aws

Panduan Developer

AWS SDK for SAP ABAP

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

AWS SDK for SAP ABAP: Panduan Developer

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

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau mungkin tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

Apa itu AWS SDK untuk SAP ABAP? 1
Fitur dari AWS SDK untuk SAP ABAP 1
Maintenance 1
Referensi API 2
Harga 2
Sumber daya 2
Memulai 3
Mendaftar untuk Akun AWS 3
Langkah 1: Siapkan AWS akun 3
Peran IAM untuk pengguna SAP 4
Autentikasi 5
Langkah 2: Instal SDK 6
Langkah 3: Konfigurasikan SDK 7
Langkah 4: Pengaturan fungsional 8
Langkah 5: Otorisasi Pengguna SAP 10
Langkah 6: Tulis kodenya 12
Langkah 7: Jalankan aplikasi 15
Mengatur 17
Prasyarat SAP 17
SDK for SAP ABAP 17
SDK untuk SAP ABAP - edisi BTP 21
Instalasi AWS SDK untuk SAP ABAP 22
Unduh SDK 22
Verifikasi file 22
AWS Transportasi SDK 23
Menginstal SDK - edisi BTP 27
Instal SDK untuk SAP ABAP - edisi BTP 27
Modul 27
Patching SDK untuk SAP ABAP - edisi BTP 28
Melakukan konfigurasi 29
Pengaturan global 30
Pengaturan teknis 31
Konfigurasikan skenario 31
Konfigurasi aplikasi 32

Profil SDK 32
Penyelesai sumber daya logis 34
Contoh 34
Pengaturan runtime 35
Log dan lacak 35
OPT-IN: telemetri yang disempurnakan 35
Skenario aktif 35
Skenario konektivitas tingkat lanjut 36
Koneksi melalui server proxy 36
Koneksi melalui paket memeriksa firewall 37
Titik akhir Gateway 37
Titik akhir antarmuka kustom 37
Perutean lanjutan 39
Per-service penggantian server proxy 39
Mengakses titik akhir di beberapa Wilayah 41
Pengaturan penyedia layanan 41
Segarkan, lacak, dan telemetri 42
Penyegaran sistem SAP 42
Jejak 43
Telemetri 44
Menggunakan SDK 45
Representasi data 45
Jenis data 46
AWS tipe data 48
Contoh program 49
Prasyarat 49
Kode 50
Bagian kode 50
Konsep 53
Kelas API 53
Objek tambahan 54
Kelas struktur 54
Array 56
Peta 57
Fungsi tingkat yang lebih tinggi 57
Fitur 1

Konfigurasi terprogram 58
Pelayan 59
Paginator 60
Coba lagi perilaku 61
Presigners 62
Rantai peran IAM lintas akun 63
Membangun produk 63
Menetapkan ID produk 63
Sesuaikan permintaan HTTP ke AWS 64
Menerapkan peningkatan 64
Filter peningkatan 64
Kode peningkatan 64
Batasan 65
Contoh kode 67
ACM 69
Tindakan 69
API Gateway 77
Tindakan 69
Pengontrol Pemulihan Aplikasi 86
Tindakan 69
Aurora 90
Tindakan 69
Auto Scaling 96
Tindakan 69
Runtime Amazon Bedrock 105
Inti Gambar Stabil 106
Runtime Agen Batuan Dasar Amazon 108
Tindakan 69
CloudFront 110
Tindakan 69
CloudWatch 112
Tindakan 69
Skenario 120
CloudWatch Log 122
Tindakan 69
Penyedia Identitas Amazon Cognito 124

Tindakan 69
Amazon Comprehend 130
Tindakan 69
AWS Config 144
Tindakan 69
AWS Control Tower 146
Tindakan 69
Firehose 158
Tindakan 69
DynamoDB 160
Hal-hal mendasar 160
Tindakan 69
Amazon EC2 174
Tindakan 69
Amazon ECR 196
Tindakan 69
Amazon EMR 203
Tindakan 69
EventBridge Penjadwal 209
Tindakan 69
AWS Glue 214
Tindakan 69
HealthImaging 226
Tindakan 69
HealthLake 244
Tindakan 69
IAM 257
Tindakan 69
AWS IoT 281
Tindakan 69
AWS IoT data 293
Tindakan 69
AWS IoT SiteWise 295
Memulai 295
Tindakan 69
Amazon Keyspaces 303

Tindakan 69
Kinesis 311
Hal-hal mendasar 160
Tindakan 69
AWS KMS 321
Memulai 295
Tindakan 69
Lambda 340
Hal-hal mendasar 160
Tindakan 69
Organizations 354
Tindakan 69
Amazon Pinpoint 359
Tindakan 69
API SMS dan Suara Amazon Pinpoint 364
Tindakan 69
Amazon Polly 374
Tindakan 69
Amazon RDS 382
Tindakan 69
Amazon Redshift 387
Tindakan 69
Amazon Rekognition 395
Tindakan 69
Amazon S3 409
Hal-hal mendasar 160
Tindakan 69
Skenario 120
Kontrol Amazon S3 434
Tindakan 69
SageMaker AI 442
Tindakan 69
Skenario 120
Secrets Manager 460
Tindakan 69
Amazon SES 462

Tindakan 69
Amazon SES API v2 479
Tindakan 69
Amazon SNS 489
Tindakan 69
Skenario 120
Amazon SQS 500
Tindakan 69
Skenario 120
Step Functions 509
Tindakan 69
Systems Manager 518
Tindakan 69
Amazon Textract 527
Tindakan 69
Skenario 120
Amazon Transcribe 538
Tindakan 69
Amazon Translate 546
Tindakan 69
Skenario 120
MCP Server 555
Manfaat 555
Menyiapkan server MCP 556
Apa yang dapat dilakukan oleh server MCP 557
Pertimbangan penting 558
Keamanan 559
Otentikasi sistem 560
Autentikasi metadata 560
Otentikasi kunci akses rahasia 561
Otentikasi berbasis sertifikat menggunakan Peran IAM Di Mana Saja 561
Otentikasi profil sumber untuk akses lintas akun 562
Langkah berikutnya 562
Praktik terbaik untuk Keamanan IAM 562
Praktik terbaik untuk profil instans Amazon EC2 563
Peran IAM untuk pengguna SAP 564

Pertimbangan Keamanan Profil Sumber 567
Otorisasi SAP 568
Otorisasi untuk konfigurasi 568
Otorisasi SAP untuk pengguna akhir 569
Operasi yang aman 570
Enkripsi Data Saat Istirahat 570
Enkripsi Data Dalam Transit 571
Penggunaan API 2
Kunci Rahasia (SSF) 571
Prasyarat 571
Prosedur 572
IAM Roles Anywhere 574
Prasyarat 574
Prosedur 574
Profil Sumber 63
Prasyarat 578
Prosedur 578
Toko Kredensyal 21
Langkah konfigurasi 581
Menggunakan SAP Credential Store dengan SDK 583
Pemecahan Masalah 587
Kegagalan impor 587
Kendala lokasi yang tidak ditentukan 587
Kesalahan SSL 588
Konfigurasi profil 589
Otorisasi IAM 590
Otorisasi untuk tindakan 590
Skenario aktif 35
Karakter-karakter khusus 591
Konektivitas 591
Topik tambahan 592
Rilis 592
Strategi rilis 592
Praktik terbaik 562
Menambal SDK untuk SAP ABAP 593
Memasang modul tambahan 594

Menghapus instalasi SDK untuk SAP ABAP 594
Lisensi SAP 595
Riwayat dokumen 596
..... dxcviii

Apa itu AWS SDK untuk SAP ABAP?

AWS SDK untuk SAP ABAP menyediakan antarmuka untuk layanan yang ditawarkan oleh AWS dalam bahasa ABAP. Dengan menggunakan SDK, Anda dapat mengimplementasikan ABAP BADIS, laporan, transaksi, layanan OData, dan artefak ABAP lainnya Layanan AWS, seperti Amazon Simple Storage Service (Amazon S3),, Amazon DynamoDB Amazon Translate, dan lainnya. Anda juga dapat mengembangkan untuk ABAP-based sistem, mulai dari SAP NetWeaver 7.4 dan di lingkungan SAP Business Technology Platform. Untuk informasi selengkapnya, lihat Menginstal AWS SDK untuk SAP ABAP - edisi BTP.

Topik

- Fitur dari AWS SDK untuk SAP ABAP
- Pemeliharaan dan dukungan untuk versi utama SDK
- Referensi API
- Harga
- Sumber daya tambahan

Fitur dari AWS SDK untuk SAP ABAP

AWS SDK untuk SAP ABAP telah dirancang agar terasa akrab dan alami bagi pengembang SAP. Misalnya, sementara semua Layanan AWS menggunakan `false` string `true` dan untuk mewakili data Boolean dalam struktur XML dan JSON, SDK untuk SAP ABAP mengonversinya menjadi nilai karakter tunggal ABAP-native`'X'.' '` SDK untuk SAP ABAP menggunakan konstruksi ABAP asli sebanyak mungkin, termasuk dalam tipe data dan format stempel waktu. Akibatnya, programmer ABAP tidak perlu khawatir tentang serialisasi JSON dan XML yang mendasarinya atau dengan format kawat protokol API.

Pemeliharaan dan dukungan untuk versi utama SDK

Untuk informasi tentang pemeliharaan dan dukungan untuk versi utama SDK dan dependensi yang mendasarinya, lihat berikut di Panduan Referensi SDK dan Alat AWS:

- AWS Kebijakan pemeliharaan SDK dan alat
- AWS SDK dan alat serta matriks dukungan versi

Referensi API

Untuk melihat daftar lengkap AWS SDK untuk SAP ABAP API, lihat AWS SDK untuk SAP ABAP - Panduan Referensi API.

Untuk melihat daftar modul lengkap AWS SDK untuk SAP ABAP TLA, lihat AWS SDK untuk SAP ABAP - Daftar Modul.

Untuk melihat daftar modul lengkap SDK untuk SAP ABAP - TLA pratinjau pengembang edisi BTP, lihat AWS SDK untuk SAP ABAP - edisi BTP - Daftar Modul.

Harga

AWS SDK untuk SAP ABAP tersedia untuk Anda tanpa biaya tambahan. Anda hanya membayar untuk AWS sumber daya dan layanan yang Anda gunakan dengan SDK.

Sumber daya tambahan

Selain panduan ini, sumber daya online berikut tersedia untuk SDK untuk SAP ABAP.

- AWS SDK untuk SAP ABAP Knowledge MCP Server — Hubungkan AI-enabled IDE Anda untuk menghasilkan kode ABAP yang akurat untuk integrasi. AWS
- AWS Perpustakaan Contoh Kode SDK
- SAP pada AWS dokumentasi
- AWS blog pengembang
- AWS forum pengembang
- @awsdevelopers (Twitter)

Memulai dengan AWS SDK untuk SAP ABAP

Bagian ini menjelaskan cara memulai dengan SDK. Ini mencakup informasi tentang menginstal SDK, melakukan konfigurasi dasar, dan membuat contoh kode Hello World yang menerjemahkan frasa dari satu bahasa ke bahasa lain. Jika Anda baru mengenal AWS SDK, sebaiknya lakukan langkah-langkah ini di lingkungan kotak pasir.

Langkah-langkah

- Mendaftar untuk Akun AWS
- Langkah 1: Siapkan AWS akun
- Langkah 2: Instal SDK
- Langkah 3: Konfigurasikan SDK
- Langkah 4: Pengaturan fungsional
- Langkah 5: Otorisasi Pengguna SAP
- Langkah 6: Tulis kodenya
- Langkah 7: Jalankan aplikasi

Mendaftar untuk Akun AWS

Untuk memulai AWS, Anda membutuhkan Akun AWS. Ada dua cara untuk mendaftar AWS. Jika Anda ingin kontrol penuh atas konfigurasi akun dan di mana konten Anda disimpan, atau jika Anda memiliki beban kerja yang diatur seperti HIPAA atau FedRAMP, gunakan Daftar untuk (lanjutan). AWS Jika Anda adalah pengguna baru dan ingin menggunakan login yang sudah Anda miliki untuk masuk AWS dan membuat AWS lingkungan dengan default yang telah dikonfigurasi sebelumnya di pilih Wilayah AWS, gunakan Daftar untuk. AWS

Langkah 1: Siapkan AWS akun

Untuk memulai dengan SDK untuk SAP ABAP, Anda harus memiliki yang aktif. Akun AWS Anda memerlukan Akun AWS bahkan jika sistem SAP Anda di-host di tempat, di SAP Business Technology Platform (BTP) atau dengan penyedia cloud lain.

Jika sistem SAP Anda berjalan di AWS Cloud, maka Anda akan melakukan panggilan ke AWS layanan di Anda Akun AWS.

Topik

- Peran IAM untuk pengguna SAP
- Autentikasi

Peran IAM untuk pengguna SAP

- Buat peran IAM dengan instruksi yang disediakan dalam Panduan AWS Identity and Access Management Pengguna. Untuk informasi selengkapnya, lihat Membuat peran untuk mendelegasikan izin ke layanan. AWS Perhatikan Nama Sumber Daya Amazon (ARN) dari peran IAM untuk digunakan nanti.
- Pilih Amazon EC2 sebagai kasus penggunaan.
- Gunakan `SapDemoTranslate` sebagai nama peran.
- Lampirkan `TranslateReadOnly` profil ke peran.
- Peran harus memiliki entitas berikut untuk memungkinkan sistem SAP untuk mengambil peran. Ganti *"111122223333"* dengan nomor akun AWS Anda.

  JSON

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

  Contoh ini menunjukkan bahwa setiap prinsipal dari Akun AWS *"111122223333"* dapat mengambil peran. Ini adalah izin luas yang cocok untuk pembuktian konsep. Anda dapat menggunakan prinsipal yang lebih sempit untuk produksi, seperti contoh berikut.

  - Pengguna tertentu — ketika sistem SAP menggunakan salah satu dari berikut ini:

    - SSF-encrypted kredensitas dari sistem SAP lokal
    - Kredensi dari layanan SAP Credential Store di SAP BTP, lingkungan ABAP
- Peran khusus - ketika sistem SAP ada di Amazon EC2 dan ada profil instance.
- Amazon EC2 — ketika sistem SAP ada di Amazon EC2 dan tidak ada profil instans.

Untuk informasi selengkapnya, lihat Praktik terbaik untuk Keamanan IAM.

Autentikasi

Otentikasi tergantung pada di mana sistem SAP Anda di-host.

Lokasi

- Pada AWS Cloud
- On-premises, SAP BTP atau cloud lainnya

Pada AWS Cloud

Pastikan bahwa instans EC2 tempat sistem SAP Anda berjalan memiliki profil instans dengan izin berikut.

JSON

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

Tambahkan ARN yang Anda catat di langkah sebelumnya.

Izin ini memungkinkan sistem SAP Anda untuk mengambil `SapDemoTranslate` peran atas nama pengguna ABAP.

On-premises, SAP BTP atau cloud lainnya

Jika sistem SAP Anda berada di tempat, di SAP BTP atau di cloud lain, gunakan langkah-langkah berikut untuk membuat koneksi untuk otentikasi menggunakan kunci akses rahasia.

1. Buat pengguna IAM. Untuk informasi selengkapnya, lihat Membuat pengguna IAM (konsol).
2. Gunakan `SapDemoSID` sebagai nama pengguna IAM. `SID`adalah ID sistem sistem SAP Anda.
3. Tetapkan `SapDemoTranslate` peran untuk pengguna ini.

Pertahankan `access_key` dan`secret_access_key`. Anda harus mengonfigurasi kredensi ini di sistem SAP Anda.

> **Note**
>
> Jika sistem SAP Anda berada di tempat, di SAP BTP atau di cloud lain, Anda dapat mengautentikasi menggunakan salah satu opsi berikut.
>
> - Otentikasi kunci akses rahasia menggunakan SSF atau SAP Credential Store
> - Menggunakan sertifikat dengan Peran IAM Di Mana Saja

Langkah 2: Instal SDK

Lihat tab berikut untuk petunjuk penginstalan.

SDK for SAP ABAP

Impor SDK untuk transportasi SAP ABAP di sistem SAP Anda. Anda dapat mengimpor transportasi ke klien mana pun. Untuk informasi selengkapnya, lihat Menginstal SDK untuk SAP ABAP.

SDK for SAP ABAP - BTP edition

Instal SDK untuk SAP ABAP - edisi BTP menggunakan aplikasi Deploy Product. Untuk informasi selengkapnya, lihat Menginstal SDK untuk SAP ABAP - edisi BTP.

Langkah 3: Konfigurasikan SDK

Sebelum mengonfigurasi SDK, pastikan Anda memiliki otorisasi yang diperlukan. Untuk informasi selengkapnya, lihat otorisasi SAP.

Lihat tab berikut untuk petunjuk konfigurasi.

SDK for SAP ABAP

Jalankan `/AWS1/IMG` transaksi untuk membuka Panduan Implementasi untuk SDK untuk SAP ABAP. Untuk menjalankan transaksi ini, masukkan `/n/AWS1/IMG` di bilah perintah sistem SAP Anda, lalu pilih Enter.

Lengkapi konfigurasi berikut.

- Pergi ke Prasyarat Teknis.
  - Tinjau Parameter yang direkomendasikan dan konektivitas HTTPS.
- Buka Pengaturan Global → Konfigurasi Skenario.
  - Ubah pengaturan, sesuai dengan rekomendasi dalam pengaturan Global.
- Buka Pengaturan Global → Pengaturan Teknis.
  - Ubah pengaturan, sesuai dengan rekomendasi dalam pengaturan Global.
- Buka Pengaturan Runtime → Log Dan Trace.
  - Pilih Entri Baru.
    - Tingkat jejak: Tidak Ada Jejak.
    - Garis Pembuangan Maksimum:`100`.
    - OPT-IN: enh telemetri: Kosongkan ini.
  - Pilih Simpan.
- Buka Pengaturan Runtime → Skenario Aktif.
  - Di bawah Skenario Baru, pilih`DEFAULT`.
  - Pilih Perubahan Skenario Komit.
  - Terima prompt.

Prasyarat untuk Sistem On-Premises

Jika sistem SAP Anda berjalan di tempat atau di cloud lain, maka kredensialnya harus disimpan di database SAP Anda. Kredensialnya dienkripsi menggunakan SAP SSF dan memerlukan pustaka kriptografi yang dikonfigurasi, seperti SAP. CommonCryptoLib

Langkah-langkah untuk mengkonfigurasi SSF untuk SDK untuk SAP ABAP dijelaskan dalam transaksi. `/AWS1/IMG`

> **Note**
>
> Prasyarat sebelumnya tidak berlaku jika sistem SAP Anda berjalan di Amazon EC2. Sistem SAP yang berjalan di Amazon EC2 mengambil kredensional yang berumur pendek dan berputar secara otomatis dari metadata instans Amazon EC2.

SDK for SAP ABAP - BTP edition

Buka lingkungan ABAP Anda di browser web, dan arahkan ke aplikasi Konfigurasi Bisnis Kustom.

Lengkapi konfigurasi berikut.

- Pergi ke Konfigurasi Skenario.
  - Ubah pengaturan, sesuai dengan rekomendasi dalam pengaturan Global.
- Buka Pengaturan Teknis.
  - Ubah pengaturan, sesuai dengan rekomendasi dalam pengaturan Global.

Langkah 4: Pengaturan fungsional

Lihat tab berikut untuk petunjuk penyiapan.

SDK for SAP ABAP

Jalankan transaksi `/AWS1/IMG` (masukkan `/n/AWS1/IMG` di bilah perintah, dan pilih Enter) untuk membuka panduan implementasi AWS SDK.

- Buka Konfigurasi Aplikasi → Profil SDK.
  - Pilih Entri Baru.
    - Profil:`DEMO`.
    - Keterangan:`Demo profile`.

    - Pilih Simpan.
- Sorot entri yang Anda buat dan klik cabang pohon Otentikasi Dan Pengaturan.
  - Pilih Entri Baru.
    - SID: ID sistem dari sistem SAP yang Anda gunakan saat ini.
    - Klien: Klien dari sistem SAP yang Anda gunakan saat ini.
    - ID Skenario: Daftar tarik-turun tempat Anda akan menemukan skenario DEFAULT yang dibuat oleh administrator Basis Anda.
    - AWS Wilayah: masukkan AWS Wilayah tempat Anda ingin melakukan panggilan. Jika sistem SAP Anda berjalan AWS, masukkan AWS Wilayah tempat ia berjalan.
    - Metode otentikasi:
      - Pilih Peran Instance melalui Metadata jika sistem SAP Anda berjalan di Amazon EC2.
      - Pilih Kredensial dari SSF Storage jika sistem SAP Anda berjalan di tempat atau di cloud lain.
        - Pilih Tetapkan Kredensial.
        - Masukkan ID Kunci Akses dan Kunci Akses Rahasia yang Anda buat di langkah sebelumnya.
  - Biarkan Nonaktifkan peran IAM tetap kosong.
  - Pilih Simpan.
- Klik pada cabang pohon Pemetaan Peran IAM.
  - Pilih Entri Baru.
    - Masukkan Nomor urutan: 010.
    - Masukkan peran IAM Logis: TESTUSER.
    - Masukkan ARN Peran IAM: masukkan arn:aws: dari peran IAM yang berisi kebijakan yang dibuat `TranslateRead0nly` pada langkah sebelumnya.

SDK for SAP ABAP - BTP edition

Siapkan otentikasi menggunakan SAP Credential Store. Untuk informasi selengkapnya, lihat Menggunakan SAP Credential Store.

Buka lingkungan ABAP Anda di browser web, dan arahkan ke aplikasi Konfigurasi Bisnis Kustom.

- Buka Profil SDK

    - Pilih Edit untuk membuat profil baru.
        - Profil:`DEMO`.
        - Keterangan:`Demo profile`.
- Pilih tombol panah kanan di sebelah entri yang dibuat untuk menavigasi ke tab Otentikasi dan Pengaturan.

  Pilih Entri Baru.
    - SID: ID sistem dari sistem SAP yang Anda gunakan saat ini.
    - Klien: Klien dari sistem SAP yang Anda gunakan saat ini.
    - ID Skenario: Daftar tarik-turun tempat Anda akan menemukan skenario DEFAULT yang dibuat oleh administrator Basis Anda.
    - AWS Wilayah: masukkan AWS Wilayah tempat Anda ingin melakukan panggilan. Jika sistem SAP Anda berjalan AWS, masukkan AWS Wilayah tempat ia berjalan.
    - Metode Otentikasi: Pilih Kredensial dari SAP Credential Store.
    - Masukkan nama Namespace dan Key dari kredensial yang disimpan di SAP Credentials Store.
    - Masukkan nama Pengaturan Komunikasi yang dibuat untuk menjalin komunikasi antara SDK untuk SAP ABAP - edisi BTP dan SAP Credential Store.
    - Biarkan Nonaktifkan peran IAM tetap kosong.
- Right-click pada tombol panah kanan di sebelah entri yang dibuat untuk menavigasi ke tab Pemetaan Peran IAM.

  Pilih Entri Baru.
    - Masukkan Nomor urutan: 010.
    - Masukkan peran IAM Logis: TESTUSER.
    - Masukkan ARN Peran IAM: masukkan arn:aws: dari peran IAM yang berisi kebijakan yang dibuat `TranslateRead0nly` pada langkah sebelumnya.

Langkah 5: Otorisasi Pengguna SAP

Pengguna SAP tidak diizinkan untuk menggunakan AWS fungsionalitas secara default. Pengguna harus secara eksplisit diotorisasi menggunakan otorisasi SAP. Lihat tab berikut untuk lebih jelasnya.

SDK for SAP ABAP

Buat peran PFCG

- Pergi ke transaksi `PFCG`
- Masukkan nama peran `ZAWS_SDK_DEMO_TESTUSER` dan pilih Buat Peran Tunggal.
  - Keterangan:`Role for demo AWS SDK functionality`.
  - Buka tab Otorisasi.
  - Pilih Ubah Data Otorisasi dan terima pop-up informasi.
  - Pada pop-up Pilih Template, pilih Jangan pilih templat.
  - Pilih Tambahkan Secara Manual dari bilah alat.
  - Tambahkan objek otorisasi berikut:
    - `/AWS1/LROL`
    - `/AWS1/SESS`
  - Di pohon otorisasi, masukkan:
    - Profil untuk mengakses AWS API: `DEMO`
    - Peran IAM Logis: `TESTUSER`
  - Pilih Simpan.
  - Pilih Hasilkan.
  - Pilih Kembali.
  - Pilih Simpan untuk menyimpan peran.

Tetapkan peran PFCG ke pengguna SAP

Setiap pengguna yang memiliki `ZAWS_SDK_DEMO_TESTUSER` peran yang ditetapkan akan diberi wewenang untuk menggunakan fungsi AWS SDK dengan pengaturan yang dikonfigurasi di profil `DEMO` SDK. Pengguna yang berwenang juga akan menganggap peran IAM yang dipetakan ke peran IAM `TESTUSER` logis dalam profil itu.

- Jalankan transaksi`SU01`.
  - Masukkan ID pengguna SAP yang akan menguji fungsionalitas AWS SDK.
  - Pilih Ubah.
  - Buka tab Peran dan tetapkan `ZAWS_SDK_DEMO_TESTUSER` peran ke pengguna.
  - Pilih Simpan.

SDK for SAP ABAP - BTP edition

Buat peran Bisnis

- Buka lingkungan ABAP Anda di browser web, dan arahkan ke aplikasi Pertahankan Peran Bisnis.
- Pilih Buat dari Template, dan masukkan detail berikut.
  - Template - Pilih**`/AWS1/RT_BTP_ENDUSER`**.
  - ID Peran Bisnis Baru - Masukkan ID.
  - Deskripsi Peran Bisnis Baru — Masukkan deskripsi.
- Pilih OK untuk melihat halaman untuk peran bisnis.
- Di bawah tab Rincian Peran Umum, buka Kategori Akses, dan atur bidang Tulis, Baca, Nilai Bantuan sebagai Dibatasi.
- Pilih Pertahankan Pembatasan, dan perluas Jenis Pembatasan yang Ditugaskan dari panel navigasi kiri. Perbarui bidang berikut di bagian Pembatasan dan Nilai.
  - Di bawah Pilih Sesi SDK, pilih ikon pensil di samping Profil SDK, dan arahkan ke tab Ranges. Masukkan**`DEMO`**, dan pilih Tambah.
  - Di bawah Pilih Peran IAM Logis, pilih ikon pensil di samping Peran IAM Logis, dan arahkan ke tab Ranges. Masukkan**`TESTUSER`**, dan pilih Tambah.

    Pilih ikon pensil di sebelah Profil SDK, dan arahkan ke tab Ranges. Masukkan**`DEMO`**, dan pilih Tambah
- Arahkan kembali ke template Peran Bisnis, dan buka tab Pengguna Bisnis. Pilih Tambah untuk menetapkan Peran Bisnis yang baru dibuat ke pengguna bisnis SAP yang akan menguji fungsionalitas SDK. Pilih Simpan.

Setiap pengguna bisnis yang ditetapkan ke Peran Bisnis yang dibuat akan diberi wewenang untuk menggunakan fungsi AWS SDK dengan setelan yang dikonfigurasi di profil `DEMO` SDK. Pengguna yang berwenang juga akan menganggap peran IAM yang dipetakan ke peran IAM `TESTUSER` logis dalam profil itu.

Langkah 6: Tulis kodenya

Lihat tab berikut untuk lebih jelasnya.

SDK for SAP ABAP

1. Buka transaksiSE38.
   - Masukkan ZDEMO_TRANSLATE_HELLO_WORLD sebagai nama program.
   - Pilih Create.
   - Masukkan AWS SDK Hello World In Any Language sebagai judul.
   - Jenis: pilih Program yang Dapat Dieksekusi.
   - Status: pilih Program Uji.
   - Pilih Simpan.
   - Simpan program sebagai Objek Lokal.

Tambahkan kode berikut.

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

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

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

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

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

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

SDK for SAP ABAP - BTP edition

1. Right-click pada paket di mana kelas ABAP perlu dibuat, lalu pilih New > ABAP class.
2. Masukkan **`ZCL_DEMO_XL8_SIMPLE`** nama Kelas, dan tambahkan deskripsi Kelas. Pilih Selanjutnya.
3. Buat atau pilih permintaan transportasi. Pilih Selesai.

Tambahkan kode berikut.

```
CLASS zcl_demo_xl8_simple DEFINITION
  PUBLIC
  FINAL
  CREATE PUBLIC .

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


CLASS zcl_demo_xl8_simple IMPLEMENTATION.
  METHOD if_oo_adt_classrun~main.

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

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

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

        out->write( |Source Phrase: { pv_text }| ).
        out->write( |Target Phrase: { lo_output->get_translatedtext( ) }| ).
      CATCH /aws1/cx_xl8unsuppedlanguage00 INTO DATA(lo_lang).
        out->write( |ERROR - Cannot translate from { lo_lang->sourcelanguagecode }
 to { lo_lang->targetlanguagecode }| ).
      CATCH cx_root INTO DATA(lo_root).
        out->write( |ERROR - { lo_root->get_text( ) }| ).
    ENDTRY.
  ENDMETHOD.
ENDCLASS.
```

Untuk detail tentang cara menulis kode ABAP yang menggunakan SDK, lihat Menggunakan. AWS SDK untuk SAP ABAP

Langkah 7: Jalankan aplikasi

Lihat tab berikut untuk lebih jelasnya.

SDK for SAP ABAP

Jalankan aplikasi di`SE38`. Jika berhasil, berikut ini akan menjadi output Anda.

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

Jika Anda kehilangan otorisasi, konfigurasi, atau prasyarat Basis, Anda mungkin mendapatkan pesan kesalahan. Lihat contoh berikut ini.

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

Jika peran SAP Anda mengizinkan Anda untuk menggunakan profil SDK dan memetakannya ke peran IAM logis sementara izin IAM Anda tidak dikonfigurasi untuk sistem SAP untuk mengambil peran IAM, berikut ini akan menjadi output Anda.

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

Dalam hal ini, tinjau izin IAM Anda dan konfigurasi kepercayaan pada peran IAM, pengguna, atau keduanya yang ditentukan. the section called “Langkah 1: Siapkan AWS akun”

SDK for SAP ABAP - BTP edition

Jalankan aplikasi di Eclipse > Run As > ABAP Application (Console). Jika berhasil, berikut ini akan menjadi output Anda.

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

Jika Anda kehilangan otorisasi, konfigurasi, atau prasyarat Basis, Anda mungkin mendapatkan pesan kesalahan. Lihat contoh berikut ini.

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

Jika peran SAP Anda mengizinkan Anda untuk menggunakan profil SDK dan memetakannya ke peran IAM logis sementara izin IAM Anda tidak dikonfigurasi untuk sistem SAP untuk mengambil peran IAM, berikut ini akan menjadi output Anda.

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

Dalam hal ini, tinjau izin IAM Anda dan konfigurasi kepercayaan pada peran IAM, pengguna, atau keduanya yang ditentukan. the section called “Langkah 1: Siapkan AWS akun”

Mengatur

Bagian ini memberikan informasi tentang cara mengatur lingkungan pengembangan Anda untuk digunakan AWS SDK untuk SAP ABAP.

Topik

- Prasyarat SAP
- Instalasi AWS SDK untuk SAP ABAP
- Menginstal AWS SDK untuk SAP ABAP - edisi BTP

Prasyarat SAP

Prasyarat berikut untuk menginstal SDK berlaku ketika sistem SAP Anda di-host. AWS

Topik

- Prasyarat untukAWSSDK for SAP ABAP
- Prasyarat untukAWSSDK untuk SAP ABAP - edisi BTP

Prasyarat untukAWSSDK for SAP ABAP

Berikut ini adalah prasyarat untuk AWS SDK untuk SAP ABAP.

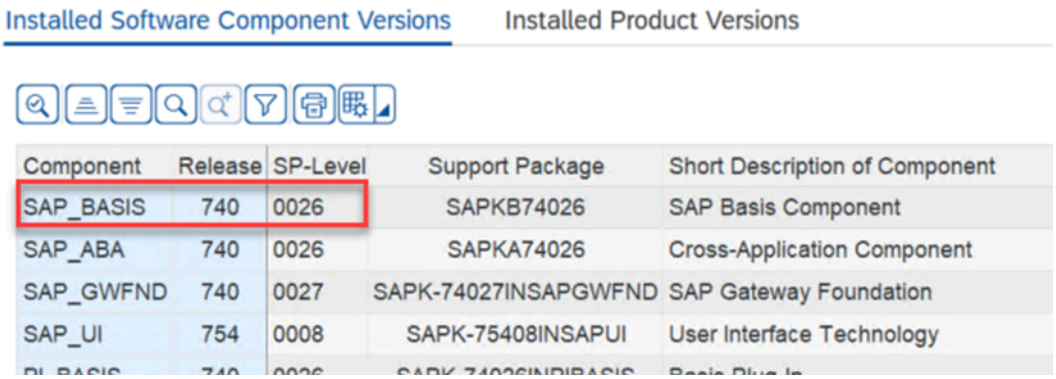
Topik

- Pelepasan dasar
- Rilis kernel
- Parameter
- Catatan
- Konektivitas keluar
- Konektivitas HTTPS
- Akses ke metadata instans Amazon EC2

Pelepasan dasar

SDK untuk SAP ABAP kompatibel dengan SAP NetWeaver 7.4 dan lebih tinggi. SDK untuk SAP ABAP tidak menyentuh tabel aplikasi SAP apa pun. Ini benar-benar agnostik tentang aplikasi, seperti SAP Enterprise Resource Planning dan SAP Landscape Transformation Replication Server.

Minimum yang didukung SP-Level untuk `SAP_BASIS 740` adalah`SP 0008`. Untuk informasi selengkapnya, lihat SAP Note 1856171 - Bidang formulir pendukung dengan nama yang sama di CL_HTTP_ENTITY (memerlukan akses portal SAP). Berdasarkan kebutuhan bisnis Anda, Anda dapat memilih yang lebih tinggi SP-Level, seperti yang ditunjukkan pada gambar berikut.



Tidak ada SP-Level persyaratan minimum untuk `SAP_BASIS 750` dan rilis yang lebih tinggi.

Rilis kernel

SDK untuk SAP ABAP dan alat yang menggunakan Internet Communication Manager (ICM) untuk konektivitas HTTP, bergantung pada kernel SAP untuk kemampuan kriptografi, HTTP, XHTML, dan JSON. Sebaiknya gunakan rilis kernel terbaru yang kompatibel dengan NetWeaver platform SAP Anda. Persyaratan minimum adalah rilis kernel 741. Untuk informasi selengkapnya, lihat SAP Note 2083594 - Versi Kernel SAP dan Tingkat Patch Kernel SAP (memerlukan akses portal SAP).

Jika Anda menggunakan rilis kernel 741 atau 742, level patch berikut diperlukan:

- 741 patchno 212
- 742 patchno 111

Parameter

Sistem SAP Anda harus mendukung Server Name Indication (SNI) seperti yang dijelaskan dalam Catatan SAP berikut (memerlukan akses portal SAP).

- SAP Note 2124480 - ICM/Web Dispatcher: Indikasi Nama Server Ekstensi TLS (SNI) sebagai klien
- SAP Note 2582368 - Pembaruan SAPSSL untuk pengiriman sisi klien ekstensi TLS SNI oleh saphttp, sapkprotp, sldreg

Konfigurasikan parameter berikut dalam `DEFAULT.PFL` file.

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

Catatan

Terapkan SAP Note berikut ke sistem Anda.

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

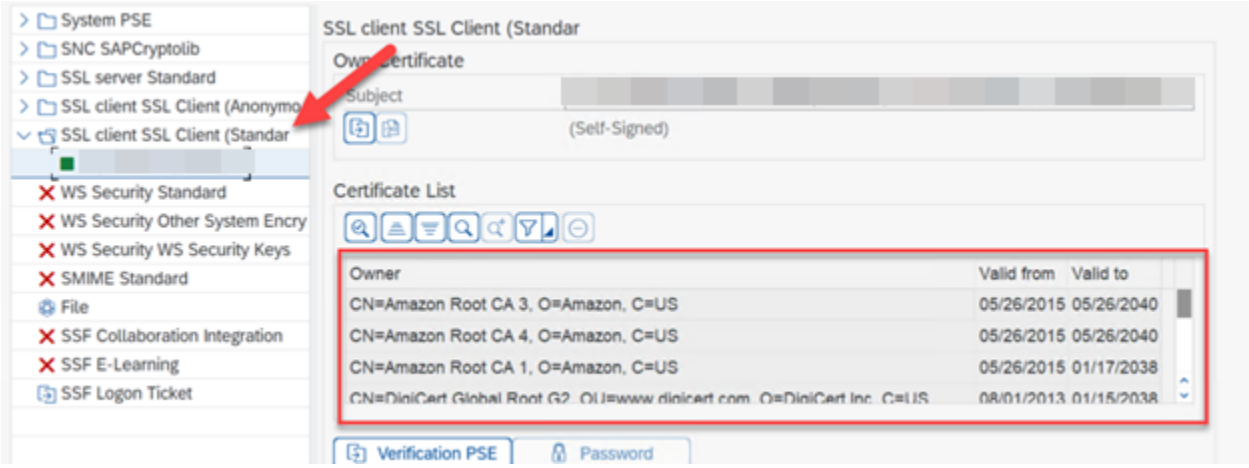
Konektivitas keluar

SDK untuk SAP ABAP adalah klien HTTPS. Sistem SAP mengirimkan pesan HTTPS keluar. Tidak ada persyaratan konektivitas masuk.

Konektivitas HTTPS

Semua panggilan AWS API dilakukan dengan saluran HTTPS terenkripsi. Sistem SAP harus diatur untuk mempercayai AWS sertifikat untuk membuat koneksi HTTPS keluar.

1. Kunjungi https://www.amazontrust.com/repository/.
2. Di bawah Root CA, unduh semua sertifikat menggunakan tautan PEM.
3. Impor sertifikat ini `SSL Client (Standard) PSE` di setiap sistem SAP Anda, seperti yang ditunjukkan pada gambar berikut. `STRUST`



Akses ke metadata instans Amazon EC2

Sistem ABAP membuat koneksi HTTP yang tidak terenkripsi ke localhost () untuk http://169.254.169.254 mengaktifkan metadata instans Amazon EC2. Saluran HTTP hanya digunakan untuk mengambil AWS kredensyal dari server lokal. Lalu lintas HTTP tetap berada di dalam host.

Metadata memungkinkan sistem SAP AWS untuk mengautentikasi dirinya sendiri dengan aman tanpa menyimpan kunci rahasia di SAP Secure Store. Fitur ini hanya berlaku untuk sistem SAP yang dihosting di Amazon EC2.

Konfigurasikan `DEFAULT.PFL` file dengan parameter berikut untuk mengaktifkan sistem SAP Anda membuat koneksi HTTP keluar yang tidak terenkripsi.

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

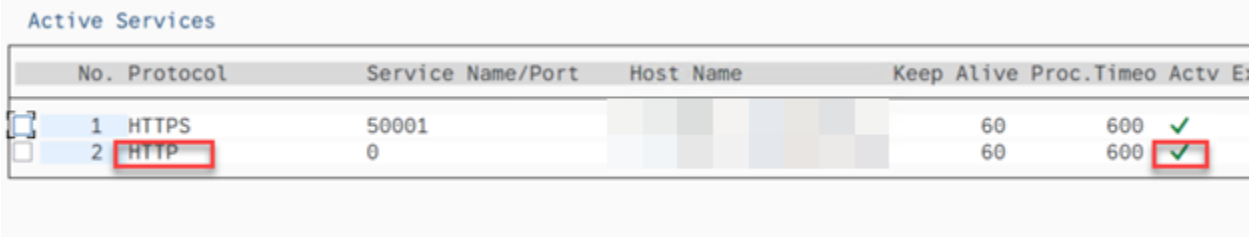
Gunakan parameter berikut untuk mengaktifkan koneksi HTTP keluar tanpa mengaktifkan koneksi masuk.

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

Verifikasi bahwa sistem SAP Anda dikonfigurasi untuk koneksi HTTP keluar dengan langkah-langkah berikut:

1. Jalankan transaksi SMICM.
2. Pergi ke Layanan Aktif.

3. Verifikasi bahwa Anda melihat tanda centang hijau di baris HTTP, di bawah kolom Aktif, seperti yang ditunjukkan pada gambar berikut.



Prasyarat untukAWSSDK untuk SAP ABAP - edisi BTP

Berikut ini adalah satu-satunya prasyarat untuk AWS SDK untuk SAP ABAP - edisi BTP.

Topik

- Portal Lanskap SAP — edisi BTP
- SAP Credential Store - edisi BTP

Portal Lanskap SAP — edisi BTP

Prasyarat ini hanya berlaku untuk AWS SDK untuk SAP ABAP - edisi BTP.

SAP Landscape Portal adalah satu-satunya mekanisme yang didukung untuk menginstal add-on di lingkungan SAP BTP. Pastikan Anda berlangganan untuk menggunakan layanan ini. Untuk informasi lebih lanjut, lihat Portal Lansekap.

SAP Credential Store - edisi BTP

Prasyarat ini hanya berlaku untuk AWS SDK untuk SAP ABAP - edisi BTP.

Dalam pratinjau pengembang, otentikasi kunci akses rahasia adalah satu-satunya mekanisme yang didukung untuk mengautentikasi AWS SDK untuk SAP ABAP - edisi BTP. SDK membaca kredensyal dari Credential Store, dan menyimpan kunci akses rahasia dengan aman.

Anda harus memenuhi prasyarat berikut.

- Berlangganan ke Credential Store.
- Credential Store ditetapkan sebagai hak atas sub-akun BTP Anda. Lihat Pengaturan awal untuk detail selengkapnya.

- Contoh layanan dengan paket standar untuk Credential Store. Lihat Membuat instance layanan untuk detail selengkapnya.

Untuk informasi selengkapnya, lihat Menggunakan SAP Credential Store.

Layanan SAP Credential Store berjalan di SAP BTP di luar sistem ABAP BTP. Lihat SAP Credential Store untuk detail selengkapnya.

Instalasi AWS SDK untuk SAP ABAP

Topik

- Unduh SDK untuk SAP ABAP
- Verifikasi SDK untuk file SAP ABAP — opsional
- AWS Transportasi SDK

Unduh SDK untuk SAP ABAP

Unduh SDK dari https://sdk-for-sapabap.aws.amazon.com/awsSdkSapabapV1/release/abapsdk-LATEST.zip https://sdk-for-sapabap.amazonaws.cn/ .

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

Ketika unduhan selesai, kami sarankan Anda unzip file yang diunduh ke dalam direktori, seperti`/tmp/awssdk`.

Verifikasi SDK untuk file SAP ABAP — opsional

Langkah opsional untuk memvalidasi tanda tangan file SDK ini membantu Anda mengonfirmasi bahwa SDK belum dirusak. Gunakan langkah-langkah berikut untuk memverifikasi file SDK Anda.

1. Unduh file SDK SIGNATURE dengan perintah berikut.

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

2. Salin kunci publik berikut, dan simpan ke file bernama `abapsdk-signing-key.pem`.

```
-----BEGIN PUBLIC KEY-----
MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAmS3oN3wKBh4HJOGaOtye
15RR5909nuw0JxOvEDCT7O9wUrXS3mjgEw6b6hvr2dLdoFr+eH4ewT5bVl6U3gDv
051sTdEJJpfLEWJJZZNK3v9fGWKyXgYe+ifmsPmf4lhNd2auzpvIy2UrlSYijCRB
BWZFW+Ux0OkILz+8vCFSXMZ6Z0qtLIlZFbGrn6A5adbwwzfOqkg9BUEZK0wB6TAi
ZTnkMdBZGCBM9K2MRKKMxtrxUn+TFcAYyh5pM9tUAb2q4XE5m7092UnZG7ur/QYl
1FSZwAhQmk8hUPgUaqOOQRC6z3TRzIGKOA/DI0cUPJMzFR4LCxEJkgh4rkRaU9V2
O7DthUpj8b7QcQaiOpnMpBf3zWLgbjNmX0hB0Eprg8/nVRHspf3zuiscJ2lMPkz0
cHOR3lMNsMLzm+d/gVkLt31R/JwAcFCkXTWvR8/VOWNGZZXdVUbefrfI/k7fP60B
bzUrIlN4poq16rc4Tk5Derg+wQ7rOWjXkXop2kiCMjbYo0ol0kS/At64PLjpz8dH
Zg25o79U9EJln+lpqZ297Ks+HoctOv2GPbeeh0s7+N0fRTyOr81EZIURLPKLVQUw
otVRzNDgLOA7eA667NrmegZfHCmqEwK9tXakZUHAcMzRPyhALc/HtmovxdStN9h1
JC4exOGqstAv1fX5QaTbMSECAwEAAQ==
-----END PUBLIC KEY-----
```

3. Verifikasi file SDK ZIP yang diunduh dengan perintah berikut. Perintah mengharuskan `openssl` itu adalah bagian dari banyak distribusi Linux.

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

4. Verifikasi bahwa output dari perintah sebelumnya adalah. `Verified OK`
5. Jika outputnya`Verification Failure`, ulangi langkah sebelumnya. Jika Anda terus menerima output yang gagal, jangan instal SDK dan kontak Dukungan.

AWS Transportasi SDK

Topik

- Daftar Isi
- Mengimpor
- Namespace

Daftar Isi

Instalasi SDK untuk SAP ABAP diselesaikan melalui ABAP Transports. Anda harus mengimpor transportasi ini ke lingkungan pengembangan atau kotak pasir Anda.

Setiap SDK untuk rilis SAP ABAP sepenuhnya menggantikan yang sebelumnya. Tidak perlu menerapkan transportasi inkremental. Transportasi dibundel dalam file ZIP. Berikut ini adalah struktur 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
.
.
.
```

`transports`Folder berisi `core` subfolder. `core`Subfolder berisi transportasi runtime inti dan subfolder untuk setiap modul, dinamai dengan singkatan tiga huruf modul. Untuk daftar modul lengkap TLAs, lihat AWS SDK untuk SAP ABAP - Daftar Modul.

AWS Transportasi SDK adalah permintaan meja kerja. Bergantung pada konfigurasi rute TMS Anda, SDK mungkin tidak secara otomatis meneruskan jaminan kualitas dan antrian produksi Anda setelah mengimpor ke sistem sebelumnya. Anda harus menambahkannya secara manual ke antrian masing-masing sistem.

Ketika proyek Anda siap untuk tahap berikutnya, AWS SDK dapat diimpor bersama dengan transportasi terpisah yang berisi Z kode Anda sendiri dengan fungsionalitas bisnis. Jika Anda menggunakan sistem kontrol perubahan, seperti SAP Change Request Management (Charm), konsultasikan dengan administrator Charm Anda untuk penanganan yang benar dari transportasi pihak ketiga.

Mengimpor

Topik

- Petunjuk kunci
- Waktu untuk mengimpor

AWS Transportasi SDK bersifat client-independent. Transportasi inti wajib dan berisi kode runtime SDK, API for AWS Security Token Service, dan API untuk Amazon Simple Storage Service. Modul SDK yang tersisa masing-masing dikirimkan dalam transportasi terpisah. Untuk menjaga ukuran SDK tetap kecil di sistem Anda, setiap modul SDK bersifat opsional. Anda dapat menginstal modul tambahan nanti, jika diperlukan untuk logika bisnis Anda.

Misalnya, jika Anda ingin menggunakan APIs untuk Amazon S3 dan Amazon Translate, impor `core` transport (berisi runtime inti, Amazon S3, dan AWS STS modul) dan `xl8` transport (berisi modul untuk) mengangkut. Amazon Translate

Untuk melihat daftar lengkap SDK untuk SAP ABAP APIs, lihat SDK untuk SAP ABAP - Panduan Referensi API.

Berikut ini adalah petunjuk utama saat mengimpor transportasi AWS SDK.

- Setiap transportasi dikirim sebagai `Knnnnnn.AWS` dan `Rnnnnnn.AWS`
  - `Knnnnnn.AWS`harus disalin ke `/usr/sap/trans/cofiles`
  - `Rnnnnnn.AWS`harus disalin ke`/usr/sap/trans/data`.
- Saat mengimpor transportasi, Anda harus memilih Abaikan Versi Komponen Tidak Valid yang ditemukan di Permintaan Transportasi Impor > Opsi > Opsi Impor.
- Semua transportasi yang diinginkan dapat diimpor secara bersamaan.
- Jika mengimpor transportasi secara terpisah, `core` transportasi harus diimpor terlebih dahulu.
- Tingkat rilis semua transportasi harus identik.

Waktu untuk mengimpor

AWS Transportasi SDK mungkin membutuhkan waktu beberapa menit untuk mengimpor. Transportasi berhasil jika STMS menunjukkan lampu hijau (`RC=0`) atau kuning (`RC=4`).

- Lampu merah (`RC=8`) menunjukkan bahwa impor memiliki kesalahan sintaks.
  - Pilih Permintaan → Tampilan → Log untuk memeriksa kesalahan impor.
  - Selama impor, jika kesalahan dilemparkan karena antarmuka yang hilang`IF_SYSTEM_UUID_RFC4122_STATIC`, maka pastikan bahwa SAP Note 2619546 diterapkan ke sistem. Untuk informasi selengkapnya, lihat Catatan.
  - Jika penyebab kesalahan tidak diketahui, hubungi Dukungan.
- Baut petir merah (`RC=12`) menunjukkan bahwa file transport belum dimuat dengan benar `/usr/sap/trans` atau tidak memiliki izin yang diperlukan.

Petunjuk kunci

Berikut ini adalah petunjuk utama saat mengimpor transportasi AWS SDK.

- Setiap transportasi dikirim sebagai `Knnnnnn.AWS` dan `Rnnnnnn.AWS`
  - `Knnnnnn.AWS`harus disalin ke `/usr/sap/trans/cofiles`
  - `Rnnnnnn.AWS`harus disalin ke`/usr/sap/trans/data`.
- Saat mengimpor transportasi, Anda harus memilih Abaikan Versi Komponen Tidak Valid yang ditemukan di Permintaan Transportasi Impor > Opsi > Opsi Impor.
- Semua transportasi yang diinginkan dapat diimpor secara bersamaan.
- Jika mengimpor transportasi secara terpisah, `core` transportasi harus diimpor terlebih dahulu.
- Tingkat rilis semua transportasi harus identik.

Waktu untuk mengimpor

AWS Transportasi SDK mungkin membutuhkan waktu beberapa menit untuk mengimpor. Transportasi berhasil jika STMS menunjukkan lampu hijau (`RC=0`) atau kuning (`RC=4`).

- Lampu merah (`RC=8`) menunjukkan impor memiliki kesalahan sintaks.
  - Pilih Permintaan → Tampilan → Log untuk memeriksa kesalahan impor.
  - Selama impor, jika kesalahan dilemparkan karena antarmuka yang hilang`IF_SYSTEM_UUID_RFC4122_STATIC`, maka pastikan bahwa SAP Note 2619546 diterapkan ke sistem. Untuk informasi selengkapnya, lihat Catatan.
  - Jika penyebab kesalahan tidak diketahui, hubungi Dukungan.
- Baut petir merah (`RC=12`) menunjukkan file transport belum dimuat dengan benar `/usr/sap/trans` atau tidak memiliki izin yang diperlukan.

Namespace

SDK untuk SAP ABAP menggunakan `/AWS1/` namespace dan tidak memodifikasi objek SAP atau objek lain di sistem Anda dengan pengecualian berikut.

- AWS `auth`objek berada di Kelas Objek Auth. Kelas Objek Auth terbatas pada empat karakter dan tidak mendukung ruang nama. SDK untuk SAP ABAP menggunakan Auth Object Class adalah. `YAW1` Jika Anda sudah memiliki kelas objek autentikasi `YAW1` dalam transaksi`SU21`, hubungi Dukungan sebelum instalasi.

Menginstal AWS SDK untuk SAP ABAP - edisi BTP

Edisi BTP ada dalam pratinjau pengembang, dan dapat diinstal dengan bergabung dengan pratinjau. Untuk menginstal SDK, isi formulir partisipasi di AWS SDK untuk SAP ABAP - BTP edition developer preview.

Sebelum menginstal SDK untuk SAP ABAP - edisi BTP, pastikan Anda memenuhi prasyarat yang diperlukan. Untuk informasi selengkapnya, lihat SAP Landscape Portal dan SAP Credential Store.

Topik

- Instal SDK untuk SAP ABAP - edisi BTP
- Modul
- Patching SDK untuk SAP ABAP - edisi BTP

Instal SDK untuk SAP ABAP - edisi BTP

1. Buka instans SAP Landscape Portal Anda, dan luncurkan aplikasi Deploy Product fiori.
2. Di Produk, pilih **`/AWS1/SDK_OMNI`**di bawah Produk Mitra.

   Hubungi Dukungan jika Anda tidak melihat *`/AWS1/SDK_OMNI`* setelah diterima di pratinjau pengembang.
3. Di Target Version, pilih versi SDK untuk SAP ABAP - edisi BTP yang ingin Anda instal di sistem Anda.
4. Di Sistem yang Tersedia, centang kotak centang untuk semua tempat Anda ingin menginstal SDK. SIDs
5. Pilih Menyebarkan, masukkan detail penjadwalan, dan pilih Jadwal. Anda dapat memantau kemajuan dalam Status Penerapan Versi Produk.

   Instalasi mungkin memakan waktu 30-45 menit, dan termasuk downtime sistem. Untuk detail selengkapnya, lihat Menerapkan Produk.

Modul

Modul berikut disertakan dalam pratinjau pengembang AWS SDK untuk SAP ABAP - edisi BTP.

- Amazon API Gateway [agw]

- Amazon Athena [] `ath`
- Runtime Amazon Bedrock [] `bdr`
- Amazon Comprehend [] `cpd`
- Amazon EventBridge [`evb`]
- Amazon Forecast [`fcs`]
- Amazon Kinesis [] `kns`
- Amazon Data Firehose [] `frh`
- Amazon SageMaker AI [`sgm`]
- Layanan Pemberitahuan Sederhana Amazon [`sns`]
- Layanan Antrian Sederhana Amazon [] `sqs`
- Layanan Penyimpanan Sederhana Amazon [`s3`]
- AWS Systems Manager [`ssm`]
- Amazon Texttract [] `tex`
- Amazon Transcribe [] `tnb`
- Amazon Translate [`xl8`]
- AWS CloudTrail [`trl`]
- AWS IoT [`iot`]
- AWS KMS [`kms`]
- AWS Lambda [`lmd`]
- AWS Secrets Manager [`smr`]
- AWS Security Token Service [`sts`]
- AWS Transfer Family [`trn`]
- Peran IAM Di Mana Saja [] `rla`
- API Data Pergeseran Merah Amazon [] `rsd`

Patching SDK untuk SAP ABAP - edisi BTP

Proses patching untuk SDK untuk SAP ABAP - edisi BTP mirip dengan proses instalasi. Jika Anda menginstal SDK pada sistem yang memiliki versi lama yang sudah diinstal, maka SDK ditambal ke versi baru pilihan Anda.

Melakukan konfigurasiAWS SDK untuk SAP ABAP

Sebelum menggunakanAWS SDK untuk SAP ABAP, Anda harus mengonfigurasi SDK dengan pengaturan teknis dan fungsional yang diperlukan untuk operasi SDK. Beberapa pengaturan dapat diangkut dan beberapa pengaturan runtime. Banyak pengaturan secara langsung analog dengan pengaturan yang ditentukan dalam `.INI` file untuk SDK lain.

Konfigurasi SDK, kecuali untuk pengaturan Runtime, harus diselesaikan di lingkungan pengembangan Anda. Anda dapat mengangkut konfigurasi ke QA dan produksi mengikuti transportasi biasa dan mengubah aturan kontrol. Konfigurasi yang dapat diangkut tidak disarankan untuk lingkungan produksi.

Jika Anda tidak memiliki izin untuk mengonfigurasi AWS SDK, lihat otorisasi SAP.

Mengkonfigurasi AWS SDK untuk SAP ABAP

Untuk menjalankan transaksi konfigurasi, masukkan `/n/AWS1/IMG` di bilah perintah SAPGUI.

Mengkonfigurasi AWS SDK untuk SAP ABAP - edisi BTP

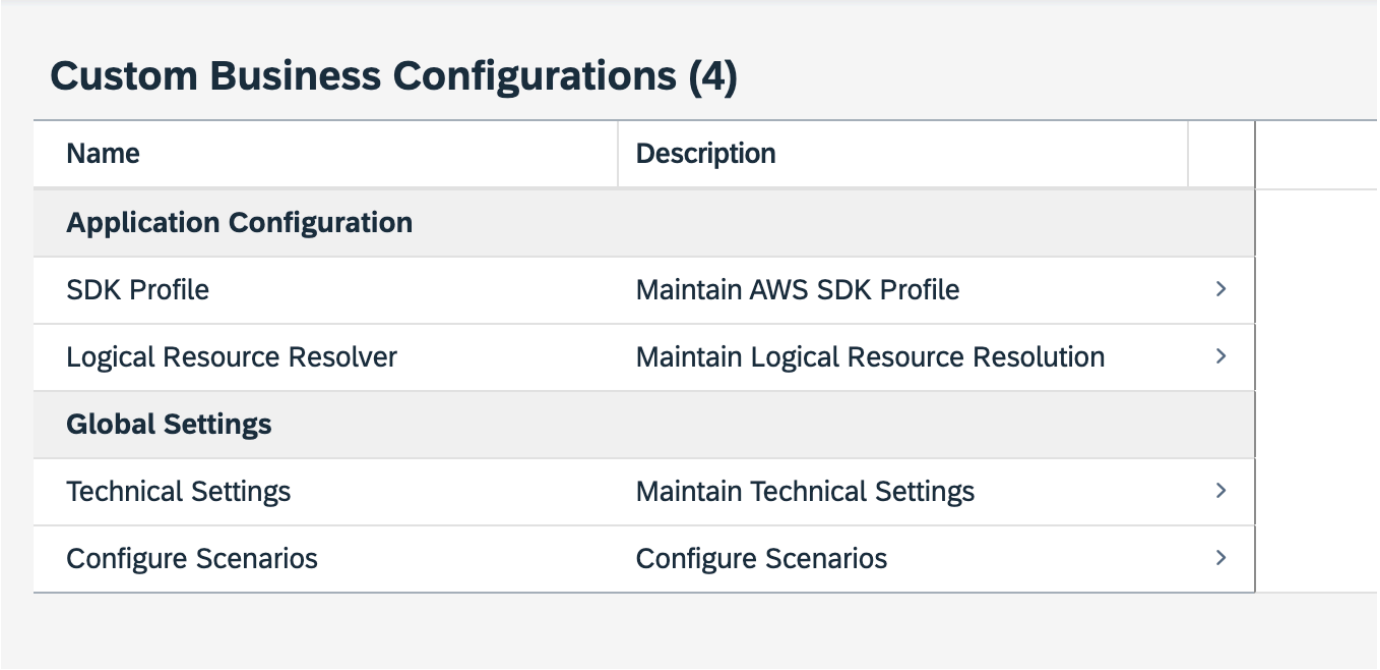
Gunakan langkah-langkah berikut untuk mengkonfigurasi SDK untuk SAP ABAP - edisi BTP.

1. Buka lingkungan ABAP Anda di browser web.
2. Arahkan ke aplikasi Konfigurasi Bisnis Kustom.

Untuk membuat permintaan penyesuaian menggunakan aplikasi Ekspor Kustomisasi Transportasi, lihat Bekerja di Aplikasi Ekspor Kustomisasi Transportasi - Buat Permintaan.

Dalam aplikasi Konfigurasi Bisnis Kustom, Anda dapat mengelompokkan konfigurasi berdasarkan jenis pengaturan SDK. Gunakan langkah-langkah berikut untuk mengelompokkan konfigurasi.

1. Buka lingkungan ABAP Anda di browser web, dan arahkan ke aplikasi Konfigurasi Bisnis Kustom.
2. Pilih Pengaturan > Grup, dan pilih Grup Konfigurasi dari daftar drop-down. Pilih OK.
3. Konfigurasi sekarang tersedia dalam struktur hierarkis seperti yang ditampilkan pada gambar. Untuk menyimpan tampilan, lihat Tampilan (Manajemen Varian) - Komponen.



Bagian ini mencakup topik-topik berikut.

Topik

- Pengaturan global
- Konfigurasi aplikasi
- Pengaturan runtime
- Skenario konektivitas tingkat lanjut
- Pengaturan penyedia layanan
- Segarkan, lacak, dan topik telemetri untuk AWS SDK untuk SAP ABAP

Pengaturan global

Gunakan transakasi `/n/AWS1/IMG` IMG untuk AWS SDK untuk SAP ABAP, dan aplikasi Custom Business Configuration untuk AWS SDK untuk SAP ABAP - edisi BTP untuk mengkonfigurasi pengaturan global. Topik ini menggunakan IMG dan Konfigurasi Bisnis Kustom secara bergantian.

Bagian ini mencakup topik-topik berikut.

Topik

- Pengaturan teknis
- Konfigurasikan skenario

Pengaturan teknis

Pengaturan `/AWS1/IMG` transaksi global mempengaruhi perilaku seluruh SDK. Pengaturan ini umumnya dikonfigurasi oleh administrator Basis. Anda dapat mengatur nilai-nilai ini ke pengaturan yang disarankan berikut.

- Pilih Entri Baru.
  - Regionalisasi S3: Akses bucket us-east-1 dengan menggunakan s3.amazonaws.com.
  - Regionalisasi STS: Akses STS dengan menggunakan titik akhir global.
  - Nonaktifkan EC2 metadata: Biarkan bidang ini kosong. Bidang ini hanya-baca dalam edisi BTP, dan diatur ke 'Ya' secara default.
  - Mode Akhir Metadata: Gunakan IPv4 titik akhir metadata. Bidang ini hanya-baca dalam edisi BTP, dan diperbarui secara otomatis.
  - Metadata Endpt URL: Biarkan bidang ini kosong. Bidang ini hanya baca dalam edisi BTP.
- Pilih Simpan.

Konfigurasikan skenario

Skenario memungkinkan AWS SDK untuk beralih pengaturan secara lebih efisien selama pengujian bencana multi-wilayah atau skenario pengujian pemulihan bencana. Anda mungkin tidak memerlukan fitur ini, dan sebagai gantinya hanya perlu mengkonfigurasi skenario DEFAULT berikut.

- Pilih Entri Baru.
  - ID Skenario: `DEFAULT`
  - Deskripsi Skenario: Skenario Default
- Pilih Simpan.

Jika Anda memiliki pengaturan pemulihan bencana Multi-wilayah atau kasus unik lainnya yang memerlukan perubahan pengaturan cepat, maka Anda dapat mengonfigurasi beberapa skenario.

- `DEFAULT`- Operasi standar.
- `DR`- Konfigurasi khusus jika bencana memerlukan pemindahan seluruh sistem ke Wilayah lain.
- `DR_TEST`- Konfigurasi khusus untuk mensimulasikan bencana, misalnya, dalam klon produksi sementara.

Konfigurasi aplikasi

Mengkonfigurasi SDK untuk SAP ABAP mirip dengan mengkonfigurasi aplikasi berbasis ABAP lainnya. Ini diatur ke dalam profil yang berbeda untuk mengelompokkan pengaturan berbagai skenario. Profil ABAP SDK mendefinisikan pengaturan yang diperlukan untuk skenario aplikasi tertentu. Misalnya, jika transaksi`ZVA01`,`ZVA02`, dan transaksi `ZVA03` terkait faktur ditingkatkan dan berjalan Layanan AWS, seperti Amazon S3, dan AWS Lambda Amazon SageMaker AI, maka profil SDK yang disebut dapat dibuat. `ZINVOICE` Profil ini dapat mengelompokkan pengaturan teknis, otorisasi SAP, dan pemetaan peran IAM untuk fungsionalitas terkait faktur.

Gunakan `/n/AWS1/IMG` transakasi untuk AWS SDK untuk SAP ABAP, dan aplikasi Custom Business Configuration untuk AWS SDK untuk SAP ABAP - edisi BTP untuk mengkonfigurasi pengaturan global. Topik ini menggunakan IMG dan Konfigurasi Bisnis Kustom secara bergantian.

Topik

- Profil SDK
- Penyelesai sumber daya logis
- Contoh

Profil SDK

Profil ABAP SDK mendefinisikan hal berikut untuk setiap SID dan klien.

> **Note**
>
> Klien selalu 100 di SAP BTP, lingkungan ABAP.

- AWS Region default untuk semua panggilan API. Misalnya, jika sistem SAP Anda berjalan di `us-east-1` Wilayah, kemungkinan AWS sumber daya Anda yang lain juga berada di Wilayah yang

sama, dan ini harus menjadi Wilayah default Anda. Kode ABAP Anda dapat mengganti Region default.

- Metode otentikasi
  - Untuk sistem SAP yang berjalan di Amazon EC2, kami sangat menyarankan untuk memilih metadata peran instans untuk mendapatkan manfaat dari kredensyal yang berumur pendek dan berputar secara otomatis.
  - Untuk sistem SAP yang berjalan di tempat atau di cloud lain, Anda harus memilih kredensyal dari penyimpanan SSF.
  - Untuk sistem ABAP yang berjalan pada SAP BTP, Anda harus memilih kredensyal dari SAP Credential Store. Untuk informasi selengkapnya, lihat Menggunakan SAP Credential Store untuk otentikasi.
  - Untuk skenario rantai peran lintas akun, pilih Profil Sumber dan tentukan ID profil sumber. Ini memungkinkan resolusi otomatis dan eksekusi rantai asumsi peran. Untuk informasi selengkapnya, lihat Menggunakan Profil Sumber untuk Akses Lintas Akun.
- Pemetaan peran IAM logis ke peran IAM.
  - Pemetaan ini diurutkan dalam urutan prioritas menurun.
  - Peran IAM dengan prioritas tertinggi di mana pengguna diberi wewenang dalam peran PFCG akan secara otomatis dipilih untuk pengguna.
- Pemetaan opsional layanan ke titik akhir kustom. Konfigurasi ini dibahas dalam skenario konektivitas lanjutan

> **Note**
>
> Peran PFGC disebut Peran Bisnis di SAP BTP, lingkungan ABAP.

Ketika program ABAP ingin terhubung ke Layanan AWS, itu akan menentukan profil ABAP SDK yang menarik pengaturan yang diperlukan. An `AUTHORIZATION-CHECK` akan dilakukan untuk mengonfirmasi bahwa pengguna memiliki izin untuk mengakses profil SDK. Administrator Keamanan SAP Anda dapat menentukan peran PFCG yang memberi Anda akses ke pengguna yang sesuai.

Penyelesai sumber daya logis

Resolver sumber daya logis memungkinkan Anda dengan tempat standar untuk menyimpan nama sumber daya. Ini dikirimkan dengan SDK untuk SAP ABAP. Tindakannya mirip dengan cara `FILE` transaksi memetakan nama file logis ke nama file fisik.

Sumber daya logis mendefinisikan konsep AWS sumber daya, seperti bucket Amazon S3 yang menyimpan faktur kami. Sumber daya logis ini, misalnya, dapat diberi nama `ZINVOICES_OUTBOUND` dan dapat dipetakan ke nama bucket fisik yang berbeda, tergantung pada apakah sistem SAP adalah pengembangan, QA, atau produksi.

SDK untuk SAP ABAP diatur sedemikian rupa sehingga sistem QA menyelesaikan sumber daya logis ke sumber daya fisik QA, bahkan setelah sistem refresh dari produksi. Pemetaan sumber daya untuk SEMUA sistem didefinisikan dalam sistem SAP pengembangan Anda dan diangkut ke depan. Pendekatan ini berbeda dari pengaturan biasa dalam sistem SAP di mana pemetaan ditangani sebagai data master dan diatur di setiap sistem. Keuntungan dari penyelesai sumber daya logis yang ditawarkan oleh SDK untuk SAP ABAP adalah bahwa kemungkinan transportasi yang salah setelah penyegaran sistem hampir tidak ada.

Contoh

Ada empat bucket Amazon S3 terpisah - masing-masing untuk pengembangan, produksi, dan QA, serta bucket QA kedua untuk pengujian regresi.

Ketika SDK menyelesaikan sumber daya logis seperti `ZINVOICE_OUTBOUND` sumber daya fisik, ia memeriksa `SY-SYSID` dan `SY-MANDT` menanyakan SID dan klien mana yang saya jalankan? , dan secara otomatis memilih sumber daya fisik yang benar.

Jika pemetaan sumber daya dalam produksi perlu diubah, Anda harus mengubah pemetaan dalam sistem pengembangan dan mengangkutnya ke depan. `IMG` Ini memastikan bahwa penugasan kembali AWS sumber daya ke sistem SAP dapat mengubah kontrol seperti halnya transportasi lainnya.

> **Note**
>
> Karena konfigurasi SDK bergantung pada klien, penugasan kembali sumber daya diangkut dalam permintaan penyesuaian, dan transportasi harus diimpor ke setiap klien.

Pengaturan runtime

Bagian ini mencakup topik-topik berikut.

> **Note**
>
> Pengaturan ini tidak dapat diangkut dan bersifat lokal untuk setiap sistem SAP.

Topik

- Log dan lacak
- OPT-IN: telemetri yang disempurnakan
- Skenario aktif

Log dan lacak

Anda dapat mengaktifkan jejak untuk tujuan debugging. Disarankan untuk menjaga tingkat jejak di No Trace, kecuali mendiagnosis masalah teknis. Untuk informasi selengkapnya, lihat operasi aman.

Pengaturan ini tidak berlaku untuk SDK untuk SAP ABAP - edisi BTP.

OPT-IN: telemetri yang disempurnakan

Semua SDKs mengirim informasi telemetri AWS untuk tujuan dukungan. Anda dapat memilih telemetri yang disempurnakan. Ini sangat berguna ketika Anda menghubungi Dukungan untuk mengidentifikasi sumber panggilan API tertentu. Untuk informasi lebih lanjut, lihat Jejak dan Telemetri.

Pengaturan ini tidak berlaku untuk SDK untuk SAP ABAP - edisi BTP.

Skenario aktif

Aktifkan `DEFAULT` skenario Anda dalam transaksi ini. Aktivasi ini diperlukan hanya sekali untuk setiap sistem dan tidak boleh diubah kecuali sistem sedang mengalami pemulihan bencana Multi-wilayah. Dalam pengaturan Multi-region, Anda dapat menggunakan pengaturan ini untuk mengalihkan sistem SAP Anda ke lingkungan pemulihan bencana atau skenario uji pemulihan bencana.

Skenario konektivitas tingkat lanjut

AWS SDK untuk SAP ABAP mengkonsumsi Layanan AWS dengan membuat panggilan HTTPS ke titik AWS akhir. Secara umum, AWS titik akhir dapat diakses melalui internet. Sistem SAP harus dapat menjangkau internet untuk membuat koneksi keluar ini. SDK untuk SAP ABAP tidak pernah memerlukan koneksi masuk dari internet ke sistem SAP.

Skenario berikut menawarkan berbagai cara untuk membuat koneksi keluar.

Skenario

- Koneksi melalui server proxy
- Koneksi melalui paket memeriksa firewall
- Titik akhir Gateway
- Titik akhir antarmuka kustom
- Perutean lanjutan
- Per-service penggantian server proxy
- Mengakses titik akhir di beberapa Wilayah

Koneksi melalui server proxy

Untuk membuat koneksi melalui server proxy, gunakan langkah-langkah berikut.

1. Di SDK, buka Transaksi **`SICF`**.
2. Pilih Eksekusi.
3. Di menu, pilih Klien > Server proxy.
4. Setel pengaturan Proxy sebagai Aktif.
5. Di bidang untuk Tidak Ada Proksi untuk Alamat Berikut, cantumkan pengecualian apa pun yang dipisahkan dengan titik koma.
6. Di bidang Protokol HTTP dan Protokol HTTPS, tentukan detail koneksi untuk server proxy Anda.

SDK tidak mengetahui server proxy, dan tidak memerlukan pengaturan apa pun untuk menggunakan konfigurasi server proxy sistem SAP.

> **Note**
>
> Jika Anda menggunakan otentikasi metadata instans Amazon EC2, maka sistem SAP tidak dapat menggunakan server proxy untuk mengakses metadata instans lokal di. `http://169.254.169.254` Anda harus menyertakan `169.254.169.254` di bidang untuk Tidak Ada Proxy untuk Alamat Berikut.

> **Note**
>
> Anda dapat mengganti perilaku server proxy berdasarkan per-layanan di bagian Perutean Lanjutan. `/AWS1/IMG` Untuk informasi selengkapnya, lihat the section called “Per-service penggantian server proxy”.

Koneksi melalui paket memeriksa firewall

Anda dapat mengkonfigurasi paket memeriksa firewall untuk koneksi keluar. Firewall ini mendekripsi lalu lintas SSL, dan kemudian mengenkripsi ulang sebelum meneruskannya ke titik akhir. Konfigurasi ini biasanya membutuhkan firewall untuk mengeluarkan sertifikat sendiri ke sistem SAP yang Layanan AWS menggunakan file. Anda harus menginstal sertifikat CA firewall Anda di`STRUST`. Untuk informasi selengkapnya, lihat konektivitas HTTPS.

Titik akhir Gateway

Beberapa Layanan AWS menawarkan titik akhir gateway untuk menyediakan VPC dengan akses kinerja tinggi tanpa internet. Titik akhir ini transparan ke SDK untuk SAP ABAP, dan tidak memerlukan konfigurasi apa pun.

Untuk informasi selengkapnya, lihat Titik akhir Gateway.

Titik akhir antarmuka kustom

Jika Anda perlu mengganti resolusi titik akhir default dengan titik akhir khusus, Anda dapat menggunakan titik akhir antarmuka untuk memberikan VPC Anda akses berkinerja tinggi tanpa internet. Untuk informasi selengkapnya, lihat Mengkonfigurasi titik akhir antarmuka.

Saat tidak menggunakan DNS pribadi, titik akhir ini memiliki alamat DNS sendiri, dan program ABAP harus secara eksplisit mengganti logika resolusi titik akhir yang biasa. Untuk informasi selengkapnya,

lihat AWS re:Post — Mengapa saya tidak dapat menyelesaikan nama domain layanan untuk titik akhir VPC antarmuka?

Dalam contoh berikut, titik akhir antarmuka dibuat untuk AWS STS dan Amazon Translate. Sistem SAP tidak menggunakan DNS pribadi, dan memanggil layanan dengan endpoint khusus. Sumber daya logis yang didefinisikan dalam `/AWS1/IMG` mewakili alamat titik akhir antarmuka fisik, sepertivpce-0123456789abcdef-hd52vxz.translate.us-west-2.vpce.amazonaws.com. Ini menghindari hard coding DNS dalam kode.

Dalam kode berikut, sumber daya logis di pertama-tama `/AWS1/IMG` diselesaikan ke nama titik akhir fisik. Mereka kemudian diberikan ke metode pabrik kelas AWS sesi (yang digunakan AWS STS untuk mengambil peran IAM) dan menerjemahkan kelas API.

```
" This example assumes we have defined our logical endpoints in /AWS1/IMG
" as logical resources so that we don't hardcode our endpoints in code.
" The endpoints may be different in Dev, QA and Prod environments.
DATA(lo_config) = /aws1/cl_rt_config=>create( 'DEMO' ).
DATA(lo_resolver) = /aws1/cl_rt_lresource_resolver=>create( lo_config ).

" logical resource STS_ENDPOINT should resolve to the interface endpoint
" for example vpce-0123456789-abcdefg.sts.us-west-2.vpce.amazonaws.com
DATA(lv_sts_endpoint) = lo_resolver->resolve_lresource( 'STS_ENDPOINT' ).

" logical resource XL8_ENDPOINT should resolve to the interface endpoint
" e.g. vpce-0123456789abcdefg-12345567.translate.us-west-2.vpce.amazonaws.com
DATA(lv_xl8_endpoint) = lo_resolver->resolve_lresource( 'XL8_ENDPOINT' ).

" the session itself uses the sts service to assume a role, so the
" session creation process requires a custom endpoint, specified here
DATA(lo_session) = /aws1/cl_rt_session_aws=>create(
  iv_profile_id = 'DEMO'
  iv_custom_sts_endpoint = |https://{ lv_sts_endpoint }|
).

" now we create an API object, and override the default endpoint with
" the custom endpoint
DATA(lo_xl8)     = /aws1/cl_xl8_factory=>create(
  io_session = lo_session
  iv_custom_endpoint = |https://{ lv_xl8_endpoint }| " provide custom endpoint
).
" now calls to lo_xl8 go to custom endpoint...
```

Seperti yang ditunjukkan dalam contoh, metode apa pun memanggil `go_xl8` pergi ke titik akhir`https://vpce-0123456789abcdefg-12345567.translate.us-west-2.vpce.amazonaws.com`. Hal ini juga memungkinkan untuk menentukan titik akhir kustom routing dalam konfigurasi IMG bukan dalam kode, seperti yang ditunjukkan pada bagian berikutnya.

Perutean lanjutan

Pada bagian sebelumnya kami menunjukkan bagaimana titik akhir kustom dapat ditentukan dalam `iv_custom_endpoint` argumen metode pabrik untuk modul SDK. Karena jumlah program ABAP yang menggunakan SDK meningkat, ini bisa menjadi sulit untuk dikelola. Dimungkinkan untuk mengonfigurasi pemetaan dari titik akhir Layanan AWS ke kustom di Profil SDK. Untuk setiap SID, klien, dan skenario, singkatan tiga huruf layanan (TLA) dapat dipetakan ke URL titik akhir:

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

Dengan konfigurasi ini, Anda tidak perlu menentukan `iv_custom_endpoint` dalam panggilan metode pabrik. Titik akhir kustom dipilih secara otomatis dari tabel konfigurasi. Konfigurasi ini khusus untuk Profil SDK sehingga Anda dapat membuat beberapa profil dengan perutean berbeda sesuai dengan kebutuhan Anda. Seperti konfigurasi Profil SDK lainnya, peruteannya adalah SID dan spesifik klien sehingga perutean terpisah dapat didefinisikan untuk sistem yang berbeda.

Per-service penggantian server proxy

Secara default, SDK menggunakan pengaturan server proxy yang dikonfigurasi dalam Transaksi `SICF` (lihatthe section called “Koneksi melalui server proxy”). Pengaturan proxy `SICF` berlaku secara global untuk semua koneksi HTTP keluar dari sistem SAP. Di beberapa lingkungan, Anda mungkin memerlukan kontrol yang lebih baik atas mana yang Layanan AWS menggunakan server proxy dan yang terhubung secara langsung.

Di bagian Perutean Lanjutan`/AWS1/IMG`, Anda dapat mengonfigurasi pengaturan Gunakan Server Proxy untuk setiap layanan. Pengaturan ini mengontrol apakah rute SDK meminta layanan tersebut melalui server proxy yang ditentukan`SICF`, terlepas dari aktivasi proxy global atau pengaturan filter.

Nilai-nilai berikut tersedia:

- Default — Gunakan perilaku proxy apa pun yang dikonfigurasi`SICF`. Jika proxy aktif dan titik akhir tidak dikecualikan oleh filter, proxy digunakan. Ini adalah perilaku default.
- Selalu — Selalu rutekan permintaan untuk layanan ini melalui server proxy yang ditentukan dalam`SICF`, terlepas dari aktivasi global atau pengaturan filter.
- Jangan pernah — Jangan pernah merutekan permintaan untuk layanan ini melalui server proxy, terlepas dari aktivasi global atau pengaturan filter. Gunakan ini ketika titik akhir layanan dapat dijangkau secara langsung tanpa proxy, misalnya saat menggunakan titik akhir VPC.

Misalnya, Anda dapat mengonfigurasi Amazon S3 dan AWS STS tidak pernah menggunakan proxy (karena mereka diakses melalui titik akhir gateway VPC), sedangkan Amazon SNS selalu menggunakan proxy karena hanya dapat dijangkau melalui internet.

TLA	Gunakan Server Proxy
EC2	Default
S3	Tidak pernah
SNS	Selalu
STS	Tidak pernah

Konfigurasi ini ditentukan per Profil SDK, SID, klien, dan skenario. Anda dapat membuat beberapa profil dengan perutean proxy yang berbeda agar sesuai dengan lingkungan atau kasus penggunaan yang berbeda.

> **Note**
>
> Host dan port server proxy selalu ditentukan di tab Protokol HTTPS dari pengaturan proxy di Transaksi`SICF`. Penggantian per-layanan hanya mengontrol apakah proxy digunakan, bukan server proxy mana yang digunakan.

Mengakses titik akhir di beberapa Wilayah

AWS endpoint secara otomatis ditentukan dari default Anda Wilayah AWS yang ditentukan dalam profil SDK. Anda juga dapat menentukan wilayah secara terprogram, mengesampingkan wilayah default. Ini dapat diganti dalam `CREATE( )` metode pabrik, atau yang lebih baru dengan objek konfigurasi SDK. Untuk informasi selengkapnya, lihat Konfigurasi terprogram.

Dalam contoh berikut, `CREATE( )` metode pabrik digunakan untuk mengatur wilayah dan daftar antrian Amazon SQS di kedua dan Wilayah. `us-east-1` `us-west-2`

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

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

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

Pengaturan penyedia layanan

Administrator basis terkadang perlu mengontrol fitur SDK tertentu di seluruh sistem, dari klien. `000` Ini adalah skenario umum untuk hosting dan penyedia layanan yang mengoperasikan sistem di AWS akun mereka sendiri atas nama pelanggan mereka. AWS SDK untuk SAP ABAP mendukung pengaturan Penyedia Layanan. Pengaturan ini dikonfigurasi di klien`000`, dan memengaruhi SDK di semua klien. Pengaturan Penyedia Layanan tidak didukung dalam SDK untuk SAP ABAP - edisi BTP.

Pengaturan Penyedia Layanan dikonfigurasi dalam transaksi`/AWS1/IMG`, dan harus dikonfigurasi di klien`000`. Pengaturan Penyedia Layanan di klien lain diabaikan. Pengaturan di klien `000` berlaku di semua klien, dan menggantikan `IMG` pengaturan lain jika terjadi konflik.

Gunakan langkah-langkah berikut untuk mengonfigurasi pengaturan Penyedia Layanan di klien`000`.

1. Perluas cabang Pengaturan Penyedia Layanan dalam transaksi`/AWS1/IMG`.
2. Pilih Pagar Pembatas Penyedia Layanan
3. Pilih Entri Baru, dan sesuaikan pengaturan berdasarkan kebutuhan bisnis Anda.
    - Nonaktifkan EC2 Metadata — mencegah SDK mengakses metadata EC2 instans di semua klien, meskipun Profil SDK dikonfigurasi untuk mengautentikasi menggunakan metadata instans. EC2 SDK memunculkan pengecualian jika program ABAP mencoba mengakses metadata instance menggunakan SDK.
4. Pilih Simpan.

Segarkan, lacak, dan topik telemetri untuk AWS SDK untuk SAP ABAP

Bagian ini mencakup topik-topik berikut.

Topik

- Penyegaran sistem SAP
- Jejak
- Telemetri

Penyegaran sistem SAP

Setelah penyegaran sistem, tantangan utama bagi administrator Basis adalah memastikan bahwa sistem terpisah tidak mengakses sumber daya masing-masing. Misalnya, Anda mungkin ingin memastikan bahwa sistem QA SAP Anda tidak mengakses sumber daya, seperti bucket S3, dari lanskap Produksi Anda.

SDK untuk SAP ABAP menyediakan pendekatan sumber daya logis yang sadar keselamatan untuk tantangan ini. Seorang analis bisnis dapat mengambil langkah-langkah berikut.

1. Tentukan sumber daya logis, seperti`ZINVOICE_OUTBOUND`.
2. Memetakan semua sistem dan klien dalam sistem pengembangan.
3. Transportasi konfigurasi SEMUA sistem ke depan hingga lanskap produksi.

Langkah-langkah dasar setelah penyegaran

1. Periksa otentikasi

- Jika sistem menggunakan autentikasi Secret Access Key, kredensi terenkripsi SSF tidak valid karena disimpan dalam data master. Kredensi harus dimasukkan kembali, yang mungkin memerlukan regenerasi Kunci Akses Rahasia baru. https://console.aws.amazon.com/iam/
- Jika sistem mengautentikasi dengan metadata EC2 instance, tidak ada langkah yang diperlukan.

Periksa pengaturan jejak

- Di`/AWS1/IMG`, pastikan bahwa pengaturan jejak adalah yang Anda inginkan. Pengaturan ini tidak dapat diangkut.

Jejak

Output jejak dikontrol dalam pengaturan runtime IMG.

Level jejak yang dapat Anda gunakan adalah:

- Tidak Ada Jejak
- Melacak panggilan API
- Melacak panggilan dan payload API

  Opsi ini berisi informasi payload yang tidak terenkripsi.
- Melacak panggilan API, payload, dan transformasi XHTML internal

  Opsi ini berisi informasi payload yang tidak terenkripsi.

Jika jejak API diaktifkan, jejak ditulis ke `DIR_WORK` dalam `aws1_trace-`*`YYYY-MM-DD`*`.log` file.

Jika payload trace juga diaktifkan, file tambahan dengan judul `aws1_payload_*` dibuat untuk setiap komponen panggilan dan payload. Panjang jejak muatan dapat dibatasi dengan batas panjang yang berlaku untuk setiap jejak muatan individu gagal.

Jejak muatan terutama dimaksudkan untuk mengumpulkan informasi yang akan diberikan jika terjadi kesalahan serialisasi. Dukungan Kami menyarankan Anda memilih No Trace kecuali Anda mencoba mendiagnosis kesalahan SDK.

> **Note**
>
> Jejak payload dapat berisi informasi bisnis yang tidak terenkripsi. Sebaiknya aktifkan jejak ini hanya untuk permintaan oleh AWS Support untuk membantu Anda memecahkan masalah. Anda dapat mematikan jejak ini setelah resolusi. Jejak tidak dihapus secara otomatis, dan perlu dihapus oleh administrator sistem ketika tidak lagi diperlukan.

Pengaturan ini tidak berlaku untuk SDK untuk SAP ABAP - edisi BTP.

Telemetri

SDKs mengirim informasi telemetri ke. Dukungan SDK untuk SAP ABAP mengumpulkan informasi berikut:

- Rilis OS dan tingkat patch
- `SAP_BASIS`tingkat rilis dan patch
- Rilis SAP Kernel dan tingkat patch

Anda dapat memilih untuk mengirimkan informasi berikut ke Dukungan.

- SAP SID dan nama instance () `host_sid_nn`
- Klien SAP () `SY-MANDT`
- Kode transaksi (`SY-TCODE`) dan report (`SY-REPID`)

Informasi tambahan memungkinkan Dukungan untuk membantu Anda lebih baik. Dukungan dapat mendeteksi mengapa panggilan API tertentu dibuat dan selanjutnya dapat menemukan transaksi yang relevan dalam sistem SAP.

Telemetri terbatas pada versi SDK dan API untuk SDK untuk SAP ABAP - edisi BTP.

Menggunakan AWS SDK untuk SAP ABAP

SDK untuk SAP ABAP memiliki dua komponen utama.

- SDK Runtime (paket `/AWS1/RT`) — sekumpulan objek yang mendukung keamanan, otentikasi, penelusuran, konfigurasi, konversi data, dan fungsi lintas-API lainnya. Modul API untuk Amazon S3,, IAM Roles Anywhere AWS STS, dan Secrets Manager adalah wajib.
- APIs (paket `/AWS1/API` dan sub-paketnya) — sub-paket untuk setiap API di mana objek dari setiap API sepenuhnya independen satu sama lain, memastikan bahwa perubahan dalam satu API tidak merusak API lain. Untuk melihat daftar lengkap AWS SDK untuk SAP ABAP APIs, lihat AWS SDK untuk SAP ABAP - Panduan Referensi API.

Bagian ini mencakup topik-topik berikut.

Topik

- Representasi data dalam ABAP
- Program contoh Amazon S3
- SDK untuk konsep SAP ABAP
- AWS SDK untuk SAP ABAP fitur
- Membangun produk dengan SDK
- Sesuaikan permintaan HTTP ke AWS
- Batasan

Representasi data dalam ABAP

Bagian ini mencakup topik-topik berikut.

Topik

- Jenis data
- AWS tipe data

Jenis data

Layanan AWS memiliki seperangkat tipe data standar yang harus dipetakan ke tipe data ABAP. Lihat tabel berikut untuk detail selengkapnya.

AWS tipe data	Tipe data ABAP	Komentar
boolean	C	Karakter tunggal "X" dan " "
String	STRING	
Byte	INT2	INT2 memiliki jangkauan yang lebih besar dari 0-255. Sebagian besar Layanan AWS akan memotong luapan tetapi perilaku ini tidak didefinisikan secara formal.
Pendek	INT2	
Bilangan Bulat	INT4	
Long	DEC19	INT8 tidak tersedia sampai ABAP 750. DEC19 digunakan untuk kompatibilitas dan konsistensi di semua platform ABAP yang didukung.
blob	XSTRING	Merupakan data biner
Desimal	STRING	Sementara ABAP mendukung DECFLOATs, itu tidak dapat mewakili nilai-nilai seperti NaN, Infinity dan -Infinity . AWS SDK merepresentasikannya secara internal sebagai STRINGs, dan mengubahnya menjadi saat runtime. DECFLOAT1
Dua Kali	STRING	

AWS tipe data	Tipe data ABAP	Komentar
		6 Jika NaN, Infinity atau +Infinity diwakili, pengemban g dapat memprosesnya sebagai tanggapan terhadap serangkaian pengecualian atau pemetaan khusus.
BigInteger	STRING	Nilai-nilai ini mewakili angka panjang tak terhingga yang tidak dapat direpresentasikan dalam ABAP, dan STRINGs digunakan sebagai pengganti BigInteger.
BigDecimal	STRING	
Stempel Waktu	`TZNTSTMPS`	`TZNTSTMPS` memungkinkan pemrosesan dengan fungsi stempel waktu ABAP asli.

Layanan AWS juga mengembalikan tipe data agregat berikut.

AWS tipe data	Tipe data ABAP	Komentar
Struktur	Kelas	
Union	Kelas	Sebuah serikat adalah sama dengan struktur, kecuali bahwa serikat pekerja tidak akan pernah memiliki lebih dari satu set bidang. Semua bidang lainnya akan disetel ke No Value.
Array	TABEL STANDAR	

AWS tipe data	Tipe data ABAP	Komentar
Hash	TABEL HASH	Tabel hash hanya akan memiliki dua kolom: KEY (string) dan VALUE (class).

AWS tipe data

Pendekatan berikut telah diintegrasikan untuk mendukung Layanan AWS di ABAP.

- Tipe AWS data tertentu tidak dapat direpresentasikan dalam ABAP. Misalnya, tipe `float` data di ABAP tidak mendukung`NaN`,`Infinity`, atau `-Infinity` nilai. Oleh karena itu, tipe `float` data direpresentasikan sebagai `STRING` dan diterjemahkan ke `DECFLOAT16` saat runtime.
- AWS data direpresentasikan pada kawat sebagai JSON atau XHTML, dan nilainya opsional. Misalnya, lihat contoh berikut yang dikembalikan oleh Layanan AWS di JSON.

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

  Jika Ana tidak memiliki nama tengah, layanan mengembalikan output berikut.

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

  ABAP tidak membedakan antara string dengan panjang 0 dan string yang tidak memiliki nilai. Bahasa lain mungkin menetapkan nilai NULL ke string atau membungkus string dalam konstruksi (seperti, pembungkus `Optional`<> Java). Ini tidak didukung di ABAP. Oleh karena itu, SDK untuk SAP ABAP memfasilitasi perbedaan nilai dengan menyediakan varian metode pengambil.

Program contoh Amazon S3

Bagian ini memandu Anda melalui program contoh sederhana untuk membuat daftar konten bucket Amazon S3 dengan menelepon. `ListObjectsV2`

Topik

- Prasyarat
- Kode
- Bagian kode

Prasyarat

Anda harus memenuhi prasyarat berikut untuk menjalankan program contoh ini.

- Anda memiliki ember Amazon S3. Dalam tutorial ini, ember diberi nama`demo-invoices.customer.com`.
- Transaksi`/AWS1/IMG`:
  - Memiliki profil SDK yang ditentukan bernama`DEMO_S3`.
    - Di profil SDK, peran IAM logis `TESTUSER` harus dipetakan ke peran IAM, seperti `arn:aws:iam::111122223333:role/SapDemoFinance` yang memberikan `s3:ListBucket` izin untuk mencantumkan konten bucket Amazon S3 Anda.
  - Memiliki sumber daya logis bernama `DEMO_BUCKET` yang dipetakan ke bucket Amazon S3 Anda dengan SID dan klien sistem SAP Anda.
- Pengguna Anda memiliki peran PFCG yang:
  - Mengotorisasi pengguna untuk mengakses profil `DEMO_S3` SDK melalui objek auth -. `/AWS1/SESS`
  - Mengotorisasi pengguna untuk `TESTUSER` akses peran IAM logis melalui objek auth -. `/AWS1/LROL`
- Sistem SAP Anda dapat mengautentikasi dirinya sendiri untuk AWS menggunakan metode yang ditentukan dalam profil SDK.
- Profil EC2 instans Amazon Anda memberi sistem SAP Anda hak untuk peran IAM yang `arn:aws:iam::111122223333:role/SapDemoFinance` dipetakan `sts:assumeRole` di profil SDK.

Kode

Blok kode berikut menunjukkan seperti apa kode Anda nantinya.

```
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.
```

Bagian kode

Berikut ini adalah ulasan kode di bagian.

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

Pengguna tidak dapat menentukan nama bucket fisik. Mereka menentukan bucket logis dan administrator sistem (khususnya analis bisnis) berkoordinasi dengan AWS administrator memetakan bucket logis ke bucket fisik. `/AWS1/IMG` Dalam sebagian besar skenario bisnis, pengguna tidak memiliki kesempatan untuk memilih bucket logika — ID sumber daya logis dikodekan keras dalam kode atau dikonfigurasi dalam tabel konfigurasi khusus.

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

Baris ini menetapkan sesi keamanan dan menyatakan bahwa program ABAP ini mengharapkan untuk menggunakan profil SDK. `DEMO_S3` Panggilan ini adalah koneksi ke konfigurasi SDK dan menarik Wilayah default, pengaturan otentikasi, dan Peran IAM yang diinginkan. Panggilan ke `AUTHORIZATION-CHECK` dilakukan secara otomatis untuk memastikan bahwa objek otorisasi `/AWS1/SESS` terpenuhi. Selain itu, `AUTHORIZATION-CHECK` panggilan akan dilakukan untuk menentukan peran IAM logis yang paling kuat (nomor urut lebih rendah) yang diizinkan pengguna, berdasarkan objek otorisasi. `/AWS1/LROL` SDK akan mengasumsikan bahwa peran IAM dipetakan ke peran IAM logis untuk SID dan klien. Kemudian, objek sesi mengaktifkan penelusuran berdasarkan pengaturan jejak di. `IMG`

Jika pengguna tidak diizinkan untuk profil SDK yang diminta atau untuk peran IAM logis apa pun yang tersedia, pengecualian akan diajukan.

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

Baris ini menyelesaikan sumber daya logis ke nama bucket fisik. Jika sumber daya logis tidak dapat diselesaikan karena konfigurasi tidak memiliki pemetaan untuk kombinasi SID/klien ini, pengecualian akan dimunculkan.

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

Baris ini membuat objek API untuk Amazon S3 menggunakan `create()` metode. `/aws1/cl_s3_factory` Objek yang dikembalikan adalah tipe `/aws1/if_s3` yang merupakan antarmuka untuk Amazon S3 API. Objek API terpisah harus dibuat untuk setiap layanan. Misalnya, jika program ABAP menggunakan Amazon S3 AWS Lambda,, dan DynamoDB, maka program tersebut akan membuat objek API dari,, dan. `/aws1/cl_s3_factory` `/aws1/cl_lmd_factory` `/aws1/cl_dyn_factory`

Ada beberapa parameter opsional ke konstruktor yang memungkinkan Anda menentukan Wilayah jika Anda ingin mengganti Wilayah default yang dikonfigurasi. IMG Dengan cara ini, bisa ada dua contoh/aws1/if_s3, satu untuk us-east-1 dan satu untukus-west-2, jika Anda ingin menyalin objek dari ember di satu Wilayah ke ember di Wilayah lain. Demikian pula, Anda dapat membuat dua objek sesi keamanan yang berbeda dan menggunakannya untuk membuat dua instance terpisah/aws1/cl_s3, jika Anda memerlukan laporan untuk membaca dari bucket terkait keuangan dan menulis objek ke bucket terkait logistik.

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

Baris ini adalah panggilan untukListObjectsV2. Hal ini membutuhkan argumen masukan sederhana dan mengembalikan satu objek. Objek-objek ini dapat mewakili data JSON dan XHTML yang dalam, de-serialisasi ke dalam konstruksi berorientasi objek ABAP. Ini bisa sangat rumit dalam beberapa kasus. Sekarang, Anda hanya perlu memproses output untuk membuat daftar isi ember.

```
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.
```

Data diakses menggunakan metode GET...() gaya yang menyembunyikan representasi internal data. GET_CONTENTS( )mengembalikan tabel ABAP dan setiap baris itu sendiri berisi objek yang mewakili entri Amazon S3 tunggal. Dalam kebanyakan kasus, AWS SDK mengambil pendekatan berorientasi objek ini dan semua data direpresentasikan sebagai objek dan tabel. LastModifiedBidang direpresentasikan sebagai stempel waktu yang dapat dikonversi ke tanggal dengan CONVERT TIME STAMP perintah ABAP-native. GET_SIZE() mengembalikan operasi matematika dan pemformatan INT4 yang mudah.

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

```
      DATA(lv_msg) = lo_ex->if_message~get_text(  ).
      MESSAGE lv_msg TYPE 'I'.
```

Semua kesalahan — koneksi, klien 4xx, server 5xx, atau kesalahan ABAP apa pun, seperti kesalahan otorisasi atau konfigurasi, direpresentasikan sebagai pengecualian. Anda dapat menangani setiap pengecualian secara terpisah. Anda memiliki pilihan apakah pengecualian harus ditangani sebagai kesalahan informasi, coba lagi, peringatan, pembuangan singkat, atau jenis penanganan lainnya.

SDK untuk konsep SAP ABAP

Bagian ini mencakup konsep dasar AWS SDK untuk SAP ABAP.

Topik

- Kelas API
- Objek tambahan
- Kelas struktur
- Array
- Peta
- Fungsi tingkat yang lebih tinggi

Kelas API

Masing-masing Layanan AWS diberi akronim tiga huruf atau`TLA`. Layanan ini diwakili oleh antarmuka dalam `/AWS1/IF_<TLA>` format. Kami akan menyebutnya antarmuka layanan. Kelas API ada dalam `/AWS1/API_<TLA>` paket. Antarmuka layanan terdiri dari satu metode untuk setiap AWS operasi (kami akan memanggil metode ini Metode Operasi). Untuk melihat daftar modul lengkap AWS SDK untuk SAP ABAP TLAs, lihat AWS SDK untuk SAP ABAP - Daftar Modul.

Setiap metode operasi memiliki beberapa `IMPORTING` argumen dan paling banyak satu `RETURNING` argumen. Seringkali, argumen ini akan menjadi objek dengan konstruktor yang rumit dan serangkaian `GET…( )` metode yang panjang. Dalam banyak kasus, objek akan berisi objek bersarang, referensi rekursif, tabel objek, tabel tabel, dan sebagainya. Ini karena Layanan AWS melewati struktur XHTML dan JSON yang dalam, yang tidak dapat diwakili oleh serangkaian argumen datar.

`RETURNING`Argumen selalu kelas, bahkan jika kelas hanya berisi satu atribut.

Objek tambahan

Selain berisi kelas API utama, setiap paket API berisi berbagai repositori terkait dan objek kamus data.

- Kelas untuk setiap objek tipe struktur.
- Sebuah kelas untuk setiap tipe data primitif yang muncul dalam tabel. Misalnya, jika layanan mengembalikan tabel string, ABAP API akan mewakilinya sebagai tabel objek, di mana setiap objek adalah kelas pembungkus yang merangkum string. Ini agar kelas pembungkus dapat menyembunyikan detail mewakili string nol yang tidak dapat direpresentasikan secara asli di ABAP.
- Kelas pengecualian untuk kesalahan tertentu yang ditentukan oleh layanan.
- Elemen data untuk setiap tipe data primitif. Setiap tipe data memiliki elemen datanya sendiri agar dapat mendokumentasikan diri.
- Objek tambahan untuk pemrosesan internal, seperti transformasi XSLT untuk serialisasi dan de-serialisasi muatan XHTML dan JSON.

Kelas struktur

Sebagian besar AWS data, dikirim dan diterima oleh layanan, diwakili oleh AWS SDK sebagai kelas. Kelas-kelas ini mewakili struktur data dan menyembunyikan detail internal penyimpanan. Secara khusus, kelas menyembunyikan cara SDK mewakili bidang ini tidak memiliki nilai.

Untuk setiap bidang dalam kelas struktur, ada tiga metode.

`GET_field( )`

`GET_field( )`Metodenya

- Mengembalikan nilai bidang, atau
- Jika bidang tidak memiliki nilai, ia mengembalikan nilai default, yang dapat Anda atur sebagai parameter opsional.

Misalnya, pertimbangkan kode berikut yang mencetak batasan lokasi bucket.

```
DATA(lo_location) = go_s3->getbucketlocation( iv_bucket = CONV string( gv_bucket ) ).
```

```
WRITE: / 'Bucket Location: ',
   lo_location->get_locationconstraint( ).
```

Jika bucket tidak memiliki batasan lokasi sama sekali (seperti dalam kasus`us-east-1`), maka `GET_LOCATIONCONSTRAINT( )` akan mengembalikan string kosong. Anda dapat mengganti perilaku ini dan menentukan nilai yang diinginkan jika bidang tidak memiliki nilai sama sekali.

```
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' ).
```

Sekarang program akan menulis `Bucket Location: assuming us-east-1` jika `getbucketlocation()` hasil tidak mengembalikan lokasi.

Dimungkinkan untuk meminta metode GET () untuk mengembalikan hasil tertentu jika nilai yang diminta benar-benar hilang, lihat contoh kode berikut.

```
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'  ).
```

Dalam hal ini, jika tidak ada kendala lokasi, `GET_LOCATIONCONSTRAINT( )` akan kembali. `NopeNopeNope`

`HAS_field( )`

`HAS_field( )`metode adalah cara untuk mengetahui apakah bidang tersebut memiliki nilai atau tidak. Lihat contoh berikut ini.

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

Jika bidang tertentu diketahui selalu memiliki nilai, tidak akan ada `HAS_field( )` metode.

`ASK_field( )`

`ASK_field( )`Metode mengembalikan nilai bidang atau menimbulkan pengecualian jika tidak memiliki nilai. Ini adalah cara yang mudah untuk memproses sejumlah bidang, dan menyelamatkan logika dan mengambil pendekatan yang berbeda jika salah satu bidang tidak memiliki nilai.

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

Perhatikan bahwa `/AWS1/CX_RT_VALUE_MISSING` ini adalah pengecualian statis dan Anda akan mendapatkan peringatan jika Anda memilih untuk tidak menangkapnya.

Praktik terbaik

Secara umum, Anda dapat menggunakan `GET_field( )` metode ini karena memperlakukan string null sebagai string kosong dan merupakan yang paling mirip ABAP dari tiga opsi. Namun, itu tidak memungkinkan Anda dengan mudah membedakan antara situasi di mana bidang memiliki nilai kosong dan di mana bidang tidak memiliki nilai. Jika logika bisnis Anda bergantung pada membedakan data yang hilang versus data kosong, maka `ASK` metode `HAS` atau memungkinkan Anda menangani kasus ini.

Array

Array direpresentasikan sebagai tabel standar objek ABAP.

Sebuah array JSON dapat berisi nilai-nilai null, seperti array berikut:. `['cat', 'dog', null, 'horse']` Ini disebut sebagai array jarang. Ini direpresentasikan dalam ABAP sebagai tabel internal referensi objek, dan `null` nilainya direpresentasikan dalam tabel sebagai nilai ABAP `null` yang sebenarnya. Saat melakukan iterasi melalui tabel jarang, Anda harus memeriksa `null` nilai untuk menghindari mengakses `null` objek dan mendapatkan pengecualian. `CX_SY_REF_IS_INITIAL` Dalam praktiknya, array jarang terjadi dalam AWS layanan.

Untuk menginisialisasi array objek, akan lebih mudah untuk menggunakan konstruksi ABAP 7.40 baru. Pertimbangkan peluncuran EC2 instans Amazon ini dengan beberapa grup keamanan yang ditetapkan:

```
ao_ec2->runinstances(
```

```
    iv_imageid                   = lo_latest_ami->get_imageid( )
    iv_instancetype              = 't2.micro'
    iv_maxcount                  = 1
    iv_mincount                  = 1
    it_securitygroupids          = VALUE /aws1/
cl_ec2secgrpidstrlist_w=>tt_securitygroupidstringlist(
                                     ( NEW /aws1/
cl_ec2secgrpidstrlist_w( 'sg-12345678' ) )
                                     ( NEW /aws1/
cl_ec2secgrpidstrlist_w( 'sg-55555555' ) )
                                     ( NEW /aws1/
cl_ec2secgrpidstrlist_w( 'sg-99999999' ) )
                                   )
    iv_subnetid                   = ao_snet->get_subnetid( )
    it_tagspecifications         = make_tag_spec( 'instance' )
)
```

Peta

Peta JSON direpresentasikan dalam ABAP sebagai `Hashed Tables` di mana setiap baris tabel hanya memiliki dua komponen.

- `KEY`— string yang `UNIQUE KEY` merupakan tabel.
- `VALUE`— objek yang mengandung nilai.

Peta adalah salah satu dari sedikit kasus di mana AWS SDK menggunakan struktur yang benar, bukan kelas. Ini diperlukan karena tabel hashed ABAP tidak dapat memiliki referensi objek sebagai bidang kunci, dan kunci AWS peta selalu string non-null.

Fungsi tingkat yang lebih tinggi

Yang Kelas API dijelaskan di bagian sebelumnya secara tepat mencerminkan AWS layanan APIs dan mewakili mereka APIs sebagai kelas ABAP yang sudah dikenal. Dalam beberapa kasus, SDK juga menyertakan fungsi tingkat yang lebih tinggi yang dibangun di atas kelas API untuk menyederhanakan operasi tertentu. Fungsi tingkat yang lebih tinggi disertakan untuk kenyamanan programmer dan tidak menggantikan kelas API tingkat rendah.

Jika SDK menyertakan fungsi tingkat yang lebih tinggi untuk modul, mereka disertakan dalam transportasi yang sama dan dapat diakses melalui kelas pabrik yang disebut`/AWS1/CL_TLA_L2_FACTORY`. Kelas pabrik menyertakan metode untuk membuat berbagai klien tingkat

yang lebih tinggi untuk modul yang didokumentasikan bersama dengan API lainnya dengan dokumentasi API.

AWS SDK untuk SAP ABAP fitur

AWS SDK untuk SAP ABAP menyediakan fitur-fitur berikut.

Topik

- Konfigurasi terprogram
- Pelayan
- Paginator
- Coba lagi perilaku
- Presigners
- Rantai peran IAM lintas akun

Konfigurasi terprogram

Gunakan transakasi `/n/AWS1/IMG` IMG untuk AWS SDK untuk SAP ABAP, dan aplikasi Custom Business Configuration untuk AWS SDK untuk SAP ABAP - edisi BTP untuk konfigurasi terprogram.

Untuk memulai konfigurasi terprogram, mulailah dengan mengambil objek konfigurasi dengan perintah. `get_config( )`

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

Setiap objek konfigurasi mengimplementasikan `/AWS1/IF_RT_CONFIG` antarmuka yang mencakup `GET` ters dan `SET` ters yang sesuai dengan. `IMG` Misalnya, wilayah default dapat diganti. Lihat contoh perintah berikut.

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

Beberapa objek konfigurasi tidak memiliki `IMG` representasi dan hanya dapat diatur secara terprogram, seperti upaya coba ulang maksimum. Lihat contoh perintah berikut.

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

Objek konfigurasi juga Layanan AWS dapat mencakup metode khusus layanan yang tidak diwakili dalam`/aws1/if_rt_config`. Misalnya, Amazon S3 dapat menangani bucket bernama `foobucket` menggunakan titik akhir `foobucket.s3.region.amazonaws.com` virtual atau `s3.region.amazonaws.com/foobucket` gaya jalur. Anda dapat menerapkan penggunaan gaya jalur dengan perintah contoh berikut.

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

Untuk informasi selengkapnya tentang konfigurasi layanan, lihat AWS SDK untuk SAP ABAP — Panduan Referensi API.

Pelayan

Saat bekerja dengan asinkron AWS APIs, Anda harus menunggu sumber daya tertentu tersedia sebelum mengambil tindakan lebih lanjut. Misalnya, `CREATETABLE()` API Amazon DynamoDB merespons langsung dengan status `CREATING` tabel. Anda dapat memulai operasi baca atau tulis hanya setelah status tabel berubah menjadi`ACTIVE`. Pelayan memberi Anda kemampuan untuk mengonfirmasi bahwa AWS sumber daya berada dalam keadaan tertentu sebelum melakukan tindakan terhadapnya.

Pelayan menggunakan operasi layanan untuk polling status sumber AWS daya sampai sumber daya mencapai status yang diinginkan atau sampai ditentukan bahwa sumber daya tidak mencapai status yang diinginkan. Ini bisa memakan waktu dan rawan kesalahan untuk menulis kode ke sumber daya polling AWS terus menerus. Pelayan membantu menyederhanakan kompleksitas ini dengan mengambil tanggung jawab melakukan jajak pendapat atas nama Anda.

Lihat contoh Amazon S3 berikut menggunakan pelayan.

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

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

" Wait until the newly created bucket becomes available
lo_s3->get_waiter( )->bucketexists(
    iv_max_wait_time = 200
```

```
    iv_bucket = |amzn-s3-demo-bucket|
).
```

- Dalam contoh ini, klien Amazon S3 digunakan untuk membuat bucket. `get_waiter()`Perintah diimplementasikan untuk menentukan kapan`bucketexists`.
- Anda harus menentukan `iv_max_wait_time` parameter untuk setiap pelayan. Ini mewakili jumlah total waktu seorang pelayan harus menunggu sebelum selesai. Dalam contoh sebelumnya, seorang pelayan dapat berlari selama 200 detik.
- Anda mungkin perlu memberikan input tambahan untuk parameter yang diperlukan. Pada contoh sebelumnya, nama bucket Amazon S3 diperlukan untuk parameter. `iv_bucket`
- `/AWS1/CX_RT_WAITER_FAILURE`pengecualian menunjukkan bahwa pelayan melebihi waktu maksimum yang ditentukan dalam `iv_max_wait_time` parameter.
- `/AWS1/CX_RT_WAITER_TIMEOUT`pengecualian menunjukkan bahwa pelayan telah berhenti karena tidak mencapai kondisi yang diinginkan.

Paginator

Beberapa Layanan AWS operasi menawarkan tanggapan halaman. Mereka diberi paginasi untuk mengembalikan jumlah data tetap dengan setiap respons. Anda perlu membuat permintaan berikutnya dengan token atau penanda untuk mengambil seluruh rangkaian hasil. Misalnya, operasi `ListObjectsV2` Amazon S3 mengembalikan hingga 1.000 objek sekaligus. Anda harus membuat permintaan berikutnya dengan token yang sesuai untuk mendapatkan halaman hasil berikutnya.

Pagination adalah proses pengiriman permintaan berturut-turut untuk mengambil di mana permintaan sebelumnya ditinggalkan. Paginator adalah iterator hasil yang disediakan oleh SDK untuk SAP ABAP. Anda dapat menggunakan paginated APIs dengan mudah, dan tanpa memahami mekanisme dasar API menggunakan token pagination.

Bekerja dengan paginator

Anda dapat membuat paginator dengan `get_paginator()` metode yang mengembalikan objek paginator. Objek paginator memanggil operasi yang dipaginasi. Objek paginator menerima parameter yang diperlukan untuk diberikan ke API yang mendasarinya. Proses ini mengembalikan objek iterator yang dapat digunakan untuk mengulangi hasil paginasi, menggunakan metode dan`has_next()`. `get_next()`

- `has_next()`mengembalikan nilai boolean yang menunjukkan jika ada lebih banyak tanggapan atau halaman yang tersedia untuk operasi yang disebut.
- `get_next()`mengembalikan respon operasi.

Contoh berikut mencantumkan semua objek dalam bucket S3 diambil dengan menggunakan paginator.

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

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

Coba lagi perilaku

SDK untuk SAP ABAP memungkinkan Anda mengonfigurasi jumlah maksimum percobaan ulang untuk permintaan Layanan AWS yang gagal karena kesalahan pelambatan atau sementara. Jumlah percobaan ulang yang diizinkan di tingkat klien layanan, yaitu berapa kali SDK mencoba ulang operasi sebelum gagal dan memunculkan pengecualian ditentukan oleh `AV_MAX_ATTEMPTS` atribut dalam objek konfigurasi layanan. Saat objek klien layanan dibuat, SDK mengonfigurasi `AV_MAX_ATTEMPTS` atribut ke nilai default 3. Objek konfigurasi layanan dapat digunakan untuk secara terprogram mengatur upaya coba ulang maksimum ke nilai yang diinginkan. Lihat contoh berikut untuk lebih jelasnya.

```
" 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**
>
> Meskipun objek konfigurasi ABAP SDK memungkinkan mode coba lagi disetel dengan `/AWS1/IF_RT_CONFIG~SET_RETRY_MODE( )` metode ini, SDK hanya mendukung mode coba lagi. `standard` Untuk informasi selengkapnya, lihat Coba lagi perilaku di AWS SDKs dan Panduan Referensi Alat.

Presigners

Anda dapat menggunakan presigned URLs untuk memberikan akses terbatas waktu ke beberapa. Layanan AWS URL presigned dapat dimasukkan dalam browser atau digunakan oleh program untuk melakukan operasi layanan. Anda dapat menggunakan URL yang telah ditandatangani beberapa kali, hingga tanggal dan waktu kedaluwarsa. Untuk informasi selengkapnya, lihat Bekerja dengan URLs SDK presigned untuk klien SAP ABAP untuk layanan yang mendukung presigner akan memiliki metode khusus yang dipanggil `GET_PRESIGNER( )` untuk membuat presigner untuk layanan tersebut. Kemudian panggil metode presigner, yang sesuai dengan metode klien API, kecuali mereka mengembalikan URL yang telah ditentukan sebelumnya alih-alih benar-benar melakukan operasi.

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

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

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

Rantai peran IAM lintas akun

Dukungan rantai peran IAM lintas akun memungkinkan akses tanpa batas ke sumber daya di beberapa AWS akun melalui konfigurasi profil sumber. Fitur ini memungkinkan Anda mengonfigurasi beberapa asumsi peran, di mana satu profil mengasumsikan peran yang kemudian mengambil peran lain, memungkinkan pola akses lintas akun yang kompleks.

Untuk informasi selengkapnya, lihat Menggunakan Profil Sumber untuk Akses Lintas Akun.

Membangun produk dengan SDK

Produk atau add-on ABAP yang dikonsumsi Layanan AWS dapat meningkatkan dan memperluas kemampuan SDK. Anda dapat membangun produk tersebut untuk digunakan dengan SDK.

Topik

- Menetapkan ID produk

Menetapkan ID produk

Disarankan agar Anda menetapkan ID produk saat membuat sesi di dalam produk atau add-on. Lihat contoh berikut untuk lebih jelasnya.

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

ID produk hanya boleh berisi huruf, angka, dan garis bawah tanpa spasi atau karakter khusus. Anda dapat mencocokkannya dengan nama teknis produk atau pengenal lainnya. Jika Anda mengembangkan beberapa produk atau add-on, ID produk harus unik untuk setiap produk. Misalnya, produk IDs untuk Invoice Analyzer, Tax Calculator, dan produk Pricing Engine dapat berupa`INVOICE_ANALYZER`,`TAX_CALCULATOR`, dan. `PRICING_ENGINE`

Menambahkan ID produk ke sesi meningkatkan telemetri yang dikirim AWS dengan setiap panggilan layanan. ID produk dan namespace objek yang membuat panggilan disertakan dalam telemetri. Dengan telemetri ini, Dukungan dapat mengidentifikasi produk yang melakukan panggilan jika pelanggan Anda menghadapi masalah dengan SDK. Ini dapat membantu memperjelas bahwa panggilan sebenarnya dilakukan oleh produk, dan bukan kode pelanggan Anda.

Sesuaikan permintaan HTTP ke AWS

AWS SDK untuk SAP ABAP Menangani proses pembuatan permintaan HTTP, mengirim payload, dan menerima respons. Anda dapat menyesuaikan perilaku atau konten permintaan HTTP untuk memenuhi persyaratan TI Anda sendiri. SDK mendefinisikan titik peningkatan `/AWS1/RT_EHN_HTTP_CLIENT` sebagai tempat sentral untuk meningkatkan komunikasi HTTP. Tempat peningkatan mendukung penambahan header HTTP ke permintaan yang dibuat. AWS

Menerapkan peningkatan

SAP menyediakan instruksi berikut untuk menerapkan tempat peningkatan:

- ABAP klasik
- BTP ABAP

Filter peningkatan

Tempat peningkatan mendukung beberapa implementasi yang dapat aktif secara bersamaan. Anda dapat memfilter eksekusi BAdi berdasarkan atribut berikut, jika Anda perlu memastikan bahwa penyempurnaan Anda hanya berjalan pada panggilan ke AWS layanan atau tindakan API tertentu:

- `TLA`- Singkatan tiga huruf dari layanan, dalam huruf besar.
- `OPERATION`- Nama tindakan API. Misalnya, operasi untuk mendapatkan objek dari bucket S3 adalah GetObject. Nama tindakan peka huruf besar/kecil dan mungkin tidak sama persis dengan nama metode ABAP.

Kode peningkatan

Peningkatan ini menyediakan metode berikut.

MODIFY_REQ_HEADERS

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

Anda dapat menambahkan dan memodifikasi header di tabel `CT_HEADERS` internal. Kami tidak menyarankan memodifikasi header, karena ini mengubah data yang digunakan layanan. AWS Setiap

header yang Anda tambahkan diabaikan oleh AWS layanan, tetapi dapat diproses oleh infrastruktur TI Anda, seperti server proxy atau middleware lainnya.

Titik peningkatan dipanggil sebelum perhitungan header otentikasi dan telemetri, jadi ini tidak dapat dimodifikasi oleh peningkatan.

Berikut ini adalah contoh implementasi.

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

Batasan

AWS SDK untuk SAP ABAP termasuk modul SDK untuk semua Layanan AWS. Beberapa modul ini mungkin memiliki keterbatasan, seperti yang dijelaskan di sini.

- Modul yang mengandalkan binding `MQTT` protokol, seperti`iotevents`, tidak akan berfungsi. `MQTT`bukan protokol berbasis HTTP dan saat ini tidak didukung oleh. AWS SDK untuk SAP ABAP
- Operasi yang mengembalikan aliran peristiwa didukung, tetapi akan menunggu hingga seluruh aliran diterima sebelum mengembalikan kontrol ke ABAP. Contohnya adalah Amazon Bedrock Agents Runtime `InvokeAgent` dan AWS Lambda `InvokeWithResponseStream`
- Operasi yang menerima aliran peristiwa tidak didukung karena keterbatasan dalam platform ABAP yang mendasarinya. Contohnya adalah Amazon Q Business `Chat` dan Amazon Lex `StartConversation`

AWS SDK untuk SAP ABAP memiliki batasan fitur berikut.

- Fitur Amazon S3 berikut belum didukung.
  - Titik akses Multi-Region
  - Enkripsi di sisi klien Amazon S3

AWS SDK untuk SAP ABAP - edisi BTP memiliki batasan berikut selama pratinjau pengembang.

- Beberapa modul mungkin tidak tersedia.
- Itu tidak dapat dihapus.

- Ini diperbarui lebih jarang.

Contoh kode SDK untuk SAP ABAP

Contoh kode dalam topik ini menunjukkan kepada Anda cara menggunakan AWS SDK untuk SAP ABAP dengan. AWS

Dasar-dasar adalah contoh kode yang menunjukkan kepada Anda bagaimana melakukan operasi penting dalam suatu layanan.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Skenario adalah contoh kode yang menunjukkan kepada Anda bagaimana menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan atau dikombinasikan dengan yang lain Layanan AWS.

Beberapa layanan berisi kategori contoh tambahan yang menunjukkan cara memanfaatkan pustaka atau fungsi khusus untuk layanan.

Layanan

- Contoh ACM menggunakan SDK untuk SAP ABAP
- Contoh API Gateway menggunakan SDK untuk SAP ABAP
- Contoh Application Recovery Controller menggunakan SDK untuk SAP ABAP
- Contoh Aurora menggunakan SDK untuk SAP ABAP
- Contoh Auto Scaling menggunakan SDK untuk SAP ABAP
- Contoh Amazon Bedrock Runtime menggunakan SDK untuk SAP ABAP
- Contoh Runtime Amazon Bedrock Agents menggunakan SDK untuk SAP ABAP
- CloudFront contoh menggunakan SDK untuk SAP ABAP
- CloudWatch contoh menggunakan SDK untuk SAP ABAP
- CloudWatch Contoh log menggunakan SDK untuk SAP ABAP
- Contoh Penyedia Identitas Amazon Cognito menggunakan SDK untuk SAP ABAP
- Amazon Comprehend contoh menggunakan SDK untuk SAP ABAP
- AWS Config contoh menggunakan SDK untuk SAP ABAP
- AWS Control Tower contoh menggunakan SDK untuk SAP ABAP

- Contoh Firehose menggunakan SDK untuk SAP ABAP
- Contoh DynamoDB menggunakan SDK untuk SAP ABAP
- Contoh Amazon EC2 menggunakan SDK untuk SAP ABAP
- Contoh Amazon ECR menggunakan SDK untuk SAP ABAP
- Contoh EMR Amazon menggunakan SDK untuk SAP ABAP
- EventBridge Contoh penjadwal menggunakan SDK untuk SAP ABAP
- AWS Glue contoh menggunakan SDK untuk SAP ABAP
- HealthImaging contoh menggunakan SDK untuk SAP ABAP
- HealthLake contoh menggunakan SDK untuk SAP ABAP
- Contoh IAM menggunakan SDK untuk SAP ABAP
- AWS IoT contoh menggunakan SDK untuk SAP ABAP
- AWS IoT data contoh menggunakan SDK untuk SAP ABAP
- AWS IoT SiteWise contoh menggunakan SDK untuk SAP ABAP
- Contoh Amazon Keyspaces menggunakan SDK untuk SAP ABAP
- Contoh Kinesis menggunakan SDK untuk SAP ABAP
- AWS KMS contoh menggunakan SDK untuk SAP ABAP
- Contoh Lambda menggunakan SDK untuk SAP ABAP
- Contoh Organizations menggunakan SDK untuk SAP ABAP
- Amazon Pinpoint contoh menggunakan SDK untuk SAP ABAP
- Amazon Pinpoint SMS dan Voice API contoh menggunakan SDK untuk SAP ABAP
- Contoh Amazon Polly menggunakan SDK untuk SAP ABAP
- Contoh Amazon RDS menggunakan SDK untuk SAP ABAP
- Contoh Amazon Redshift menggunakan SDK untuk SAP ABAP
- Contoh Rekognition Amazon menggunakan SDK untuk SAP ABAP
- Contoh Amazon S3 menggunakan SDK untuk SAP ABAP
- Contoh Kontrol Amazon S3 menggunakan SDK untuk SAP ABAP
- SageMaker Contoh AI menggunakan SDK untuk SAP ABAP
- Contoh Secrets Manager menggunakan SDK untuk SAP ABAP
- Contoh Amazon SES menggunakan SDK untuk SAP ABAP
- Contoh Amazon SES API v2 menggunakan SDK untuk SAP ABAP

- Contoh Amazon SNS menggunakan SDK untuk SAP ABAP
- Contoh Amazon SQS menggunakan SDK untuk SAP ABAP
- Contoh Step Functions menggunakan SDK untuk SAP ABAP
- Contoh Systems Manager menggunakan SDK untuk SAP ABAP
- Contoh Amazon Texttract menggunakan SDK untuk SAP ABAP
- Contoh Amazon Transcribe menggunakan SDK untuk SAP ABAP
- Contoh Amazon Translate menggunakan SDK untuk SAP ABAP

Contoh ACM menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan mengimplementasikan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan ACM.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`AddTagsToCertificate`

Contoh kode berikut menunjukkan cara menggunakan`AddTagsToCertificate`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat AddTagsToCertificatedi AWS SDK untuk referensi SAP ABAP API.

`DeleteCertificate`

Contoh kode berikut menunjukkan cara menggunakan`DeleteCertificate`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

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

- Untuk detail API, lihat DeleteCertificatedi AWS SDK untuk referensi SAP ABAP API.

`DescribeCertificate`

Contoh kode berikut menunjukkan cara menggunakan`DescribeCertificate`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat DescribeCertificatedi AWS SDK untuk referensi SAP ABAP API.

`GetCertificate`

Contoh kode berikut menunjukkan cara menggunakan`GetCertificate`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat GetCertificatedi AWS SDK untuk referensi SAP ABAP API.

`ImportCertificate`

Contoh kode berikut menunjukkan cara menggunakan`ImportCertificate`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " Only pass certificate chain if it's provided (it's optional)
        IF iv_certificate_chain IS NOT INITIAL.
```

```
        DATA(lo_result) = lo_acm->importcertificate(
          iv_certificate = iv_certificate
          iv_privatekey = iv_private_key
          iv_certificatechain = iv_certificate_chain
        ).
      ELSE.
        lo_result = lo_acm->importcertificate(
          iv_certificate = iv_certificate
          iv_privatekey = iv_private_key
        ).
      ENDIF.
      ov_certificate_arn = lo_result->get_certificatearn( ).
      MESSAGE 'Certificate imported successfully.' TYPE 'I'.
    CATCH /aws1/cx_acminvalidparameterex.
      MESSAGE 'Invalid parameter provided.' TYPE 'I'.
    CATCH /aws1/cx_acmlimitexceededex.
      MESSAGE 'Certificate limit exceeded.' TYPE 'I'.
  ENDTRY.
```

- Untuk detail API, lihat ImportCertificatedi AWS SDK untuk referensi SAP ABAP API.

`ListCertificates`

Contoh kode berikut menunjukkan cara menggunakan`ListCertificates`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
  TRY.
      oo_result = lo_acm->listcertificates(
        iv_maxitems = iv_max_items
        it_certificatestatuses = it_statuses
        io_includes = io_includes
      ).
      MESSAGE 'Certificates listed successfully.' TYPE 'I'.
    CATCH /aws1/cx_acminvalidargsex.
```

```
        MESSAGE 'Invalid arguments provided.' TYPE 'I'.
      CATCH /aws1/cx_acmvalidationex.
        MESSAGE 'Validation error occurred.' TYPE 'I'.
    ENDTRY.
```

- Untuk detail API, lihat ListCertificatesdi AWS SDK untuk referensi SAP ABAP API.

`ListTagsForCertificate`

Contoh kode berikut menunjukkan cara menggunakan`ListTagsForCertificate`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat ListTagsForCertificatedi AWS SDK untuk referensi SAP ABAP API.

`RemoveTagsFromCertificate`

Contoh kode berikut menunjukkan cara menggunakan`RemoveTagsFromCertificate`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat RemoveTagsFromCertificatedi AWS SDK untuk referensi SAP ABAP API.

`RequestCertificate`

Contoh kode berikut menunjukkan cara menggunakan`RequestCertificate`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_domain_name = 'example.com'
        " iv_validation_method = 'DNS' or 'EMAIL'
        DATA(lo_result) = lo_acm->requestcertificate(
          iv_domainname = iv_domain_name
          it_subjectalternativenames = COND #( WHEN it_alternate_domains IS NOT
 INITIAL
                                               THEN it_alternate_domains )
          iv_validationmethod = iv_validation_method
        ).
        ov_certificate_arn = lo_result->get_certificatearn( ).
        MESSAGE 'Certificate requested successfully.' TYPE 'I'.
      CATCH /aws1/cx_acminvalidparameterex.
        MESSAGE 'Invalid parameter provided.' TYPE 'I'.
      CATCH /aws1/cx_acmlimitexceededex.
        MESSAGE 'Certificate limit exceeded.' TYPE 'I'.
      CATCH /aws1/cx_acminvdomvationoptsex.
        MESSAGE 'Invalid domain validation options.' TYPE 'I'.
    ENDTRY.
```

- Untuk detail API, lihat RequestCertificatedi AWS SDK untuk referensi SAP ABAP API.

`ResendValidationEmail`

Contoh kode berikut menunjukkan cara menggunakan`ResendValidationEmail`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_certificate_arn = 'arn:aws:acm:region:123456789012:certificate/
certificate-id'
        " iv_domain = 'example.com'
        " iv_validation_domain = 'example.com'
        lo_acm->resendvalidationemail(
```

```
        iv_certificatearn = iv_certificate_arn
        iv_domain = iv_domain
        iv_validationdomain = iv_validation_domain
      ).
      MESSAGE 'Validation email resent successfully.' TYPE 'I'.
    CATCH /aws1/cx_acminvalidarnex.
      MESSAGE 'The certificate ARN is not valid.' TYPE 'I'.
    CATCH /aws1/cx_acmresourcenotfoundex.
      MESSAGE 'Certificate not found.' TYPE 'I'.
    CATCH /aws1/cx_acminvalidstateex.
      MESSAGE 'Certificate is not in a valid state.' TYPE 'I'.
    CATCH /aws1/cx_acminvdomvationoptsex.
      MESSAGE 'Invalid domain validation options.' TYPE 'I'.
  ENDTRY.
```

- Untuk detail API, lihat ResendValidationEmaildi AWS SDK untuk referensi SAP ABAP API.

Contoh API Gateway menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan API Gateway.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`CreateDeployment`

Contoh kode berikut menunjukkan cara menggunakan`CreateDeployment`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat CreateDeploymentdi AWS SDK untuk referensi SAP ABAP API.

`CreateResource`

Contoh kode berikut menunjukkan cara menggunakan`CreateResource`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat CreateResourcedi AWS SDK untuk referensi SAP ABAP API.

`CreateRestApi`

Contoh kode berikut menunjukkan cara menggunakan`CreateRestApi`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

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

- Untuk detail API, lihat CreateRestApidi AWS SDK untuk referensi SAP ABAP API.

`DeleteRestApi`

Contoh kode berikut menunjukkan cara menggunakan`DeleteRestApi`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat DeleteRestApidi AWS SDK untuk referensi SAP ABAP API.

`GetResources`

Contoh kode berikut menunjukkan cara menggunakan`GetResources`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat GetResourcesdi AWS SDK untuk referensi SAP ABAP API.

`GetRestApis`

Contoh kode berikut menunjukkan cara menggunakan`GetRestApis`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat GetRestApisdi AWS SDK untuk referensi SAP ABAP API.

`PutIntegration`

Contoh kode berikut menunjukkan cara menggunakan`PutIntegration`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_agw->putintegration(
      iv_restapiid = iv_rest_api_id
```

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

- Untuk detail API, lihat PutIntegrationdi AWS SDK untuk referensi SAP ABAP API.

`PutIntegrationResponse`

Contoh kode berikut menunjukkan cara menggunakan`PutIntegrationResponse`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

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

- Untuk detail API, lihat PutIntegrationResponsedi AWS SDK untuk referensi SAP ABAP API.

`PutMethod`

Contoh kode berikut menunjukkan cara menggunakan`PutMethod`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat PutMethoddi AWS SDK untuk referensi SAP ABAP API.

`PutMethodResponse`

Contoh kode berikut menunjukkan cara menggunakan`PutMethodResponse`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat PutMethodResponsedi AWS SDK untuk referensi SAP ABAP API.

Contoh Application Recovery Controller menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Application Recovery Controller.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`GetRoutingControlState`

Contoh kode berikut menunjukkan cara menggunakan`GetRoutingControlState`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

" Parse the comma-separated cluster endpoints
```

```
    " Expected format: "https://endpoint1.com|us-west-2,https://endpoint2.com|us-
east-1"
    SPLIT iv_cluster_endpoints AT ',' INTO TABLE lt_endpoints.

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

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

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

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

          " Try to get the routing control state
          oo_result = lo_client->getroutingcontrolstate(
            iv_routingcontrolarn = iv_routing_control_arn ).

          " If successful, return the result
          RETURN.

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

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

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

- Untuk detail API, lihat GetRoutingControlStatedi AWS SDK untuk referensi SAP ABAP API.

`UpdateRoutingControlState`

Contoh kode berikut menunjukkan cara menggunakan`UpdateRoutingControlState`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

    " Parse the comma-separated cluster endpoints
    " Expected format: "https://endpoint1.com|us-west-2,https://endpoint2.com|us-
east-1"
```

```
    SPLIT iv_cluster_endpoints AT ',' INTO TABLE lt_endpoints.

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

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

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

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

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

          " If successful, return the result
          RETURN.

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

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

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

- Untuk detail API, lihat UpdateRoutingControlStatedi AWS SDK untuk referensi SAP ABAP API.

Contoh Aurora menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan mengimplementasikan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Aurora.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`CreateDBClusterParameterGroup`

Contoh kode berikut menunjukkan cara menggunakan`CreateDBClusterParameterGroup`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat CreateDBClusterParameterGroupdi AWS SDK untuk referensi SAP ABAP API.

`DeleteDBClusterParameterGroup`

Contoh kode berikut menunjukkan cara menggunakan`DeleteDBClusterParameterGroup`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
```

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

- Untuk detail API, lihat DeleteDBClusterParameterGroupdi AWS SDK untuk referensi SAP ABAP API.

`DescribeDBClusterParameterGroups`

Contoh kode berikut menunjukkan cara menggunakan`DescribeDBClusterParameterGroups`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat DescribeDBClusterParameterGroupsdi AWS SDK untuk referensi SAP ABAP API.

`DescribeDBClusterParameters`

Contoh kode berikut menunjukkan cara menggunakan`DescribeDBClusterParameters`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

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

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

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

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

- Untuk detail API, lihat DescribeDBClusterParametersdi AWS SDK untuk referensi SAP ABAP API.

`DescribeDBEngineVersions`

Contoh kode berikut menunjukkan cara menggunakan`DescribeDBEngineVersions`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat DescribeDBEngineVersionsdi AWS SDK untuk referensi SAP ABAP API.

`DescribeOrderableDBInstanceOptions`

Contoh kode berikut menunjukkan cara menggunakan`DescribeOrderableDBInstanceOptions`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat DescribeOrderableDBInstanceOptionsdi AWS SDK untuk referensi SAP ABAP API.

`ModifyDBClusterParameterGroup`

Contoh kode berikut menunjukkan cara menggunakan`ModifyDBClusterParameterGroup`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat ModifyDBClusterParameterGroupdi AWS SDK untuk referensi SAP ABAP API.

Contoh Auto Scaling menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Auto Scaling.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`CreateAutoScalingGroup`

Contoh kode berikut menunjukkan cara menggunakan`CreateAutoScalingGroup`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

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

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

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

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

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

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

- Untuk detail API, lihat CreateAutoScalingGroupdi AWS SDK untuk referensi SAP ABAP API.

`DeleteAutoScalingGroup`

Contoh kode berikut menunjukkan cara menggunakan`DeleteAutoScalingGroup`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

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

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

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

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

- Untuk detail API, lihat DeleteAutoScalingGroupdi AWS SDK untuk referensi SAP ABAP API.

`DescribeAutoScalingGroups`

Contoh kode berikut menunjukkan cara menggunakan`DescribeAutoScalingGroups`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

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

    TRY.
        " Build group names parameter
        CREATE OBJECT lo_group_name
          EXPORTING
            iv_value = iv_group_name.
        APPEND lo_group_name TO lt_group_names.

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

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

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

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

- Untuk detail API, lihat DescribeAutoScalingGroupsdi AWS SDK untuk referensi SAP ABAP API.

`DescribeAutoScalingInstances`

Contoh kode berikut menunjukkan cara menggunakan`DescribeAutoScalingInstances`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

    TRY.
        DATA(lo_output) = ao_asc->describeautoscalinginstances(
          it_instanceids = it_instance_ids ).

        ot_output = lo_output->get_autoscalinginstances( ).

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

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

- Untuk detail API, lihat DescribeAutoScalingInstancesdi AWS SDK untuk referensi SAP ABAP API.

`DescribeScalingActivities`

Contoh kode berikut menunjukkan cara menggunakan`DescribeScalingActivities`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

TRY.
    DATA(lo_output) = ao_asc->describescalingactivities(
      iv_autoscalinggroupname = iv_group_name ).

    ot_output = lo_output->get_activities( ).

    MESSAGE 'Scaling activities retrieved successfully' TYPE 'I'.

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

- Untuk detail API, lihat DescribeScalingActivitiesdi AWS SDK untuk referensi SAP ABAP API.

`DisableMetricsCollection`

Contoh kode berikut menunjukkan cara menggunakan`DisableMetricsCollection`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

```
    TRY.
        ao_asc->disablemetricscollection(
          iv_autoscalinggroupname = iv_group_name ).

        MESSAGE 'Metrics collection disabled successfully' TYPE 'I'.

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

- Untuk detail API, lihat DisableMetricsCollectiondi AWS SDK untuk referensi SAP ABAP API.

`EnableMetricsCollection`

Contoh kode berikut menunjukkan cara menggunakan`EnableMetricsCollection`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

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

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

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

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

- Untuk detail API, lihat EnableMetricsCollectiondi AWS SDK untuk referensi SAP ABAP API.

`SetDesiredCapacity`

Contoh kode berikut menunjukkan cara menggunakan`SetDesiredCapacity`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

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

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

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

- Untuk detail API, lihat SetDesiredCapacitydi AWS SDK untuk referensi SAP ABAP API.

`TerminateInstanceInAutoScalingGroup`

Contoh kode berikut menunjukkan cara menggunakan`TerminateInstanceInAutoScalingGroup`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

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

    oo_output = lo_output->get_activity( ).

    MESSAGE 'Instance terminated successfully' TYPE 'I'.

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

- Untuk detail API, lihat TerminateInstanceInAutoScalingGroupdi AWS SDK untuk referensi SAP ABAP API.

`UpdateAutoScalingGroup`

Contoh kode berikut menunjukkan cara menggunakan`UpdateAutoScalingGroup`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

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

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

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

- Untuk detail API, lihat UpdateAutoScalingGroupdi AWS SDK untuk referensi SAP ABAP API.

Contoh Amazon Bedrock Runtime menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Amazon Bedrock Runtime.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Inti Gambar Stabil

Inti Gambar Stabil

InvokeModel

Contoh kode berikut menunjukkan cara memanggil Stability.ai Stable Image Core di Amazon Bedrock untuk menghasilkan gambar.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

Buat gambar dengan Difusi Stabil.

```
    "Stable Image Core Input Parameters should be in a format like this:
*   {
*     "prompt": "Draw a dolphin with a mustache, photorealistic",
*     "aspect_ratio": "1:1",
*     "seed": 0,
*     "output_format": "png"
*   }
    DATA: BEGIN OF ls_input,
            prompt        TYPE /aws1/rt_shape_string,
            aspect_ratio  TYPE /aws1/rt_shape_string,
            seed          TYPE /aws1/rt_shape_integer,
            output_format TYPE /aws1/rt_shape_string,
          END OF ls_input.

    ls_input-prompt = iv_prompt.
    ls_input-aspect_ratio = '1:1'.
    ls_input-seed = 0. "or better, choose a random integer.
    ls_input-output_format = 'png'.

    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-image-core-v1:1'
          iv_accept = 'application/json'
          iv_contenttype = 'application/json' ).

        "Stable Image Core Result Format:
*       {
*         "seeds": ["0"],
*         "finish_reasons": [null],
*         "images": ["iVBORw0KGgoAAAANSUhEUgAAAgAAA...."]
*       }
        DATA: BEGIN OF ls_response,
                images TYPE STANDARD TABLE OF /aws1/rt_shape_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 ).
        IF ls_response-images IS NOT INITIAL.
          DATA(lv_image) =
 cl_http_utility=>if_http_utility~decode_x_base64( ls_response-images[ 1 ] ).
        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.
```

Panggil model dasar Stability.ai Stable Image Core untuk menghasilkan gambar menggunakan klien tingkat tinggi L2.

```
    TRY.
        DATA(lo_bdr_l2_sd) = /aws1/
cl_bdr_l2_factory=>create_stable_diffusion_xl_1( lo_bdr ).
        " iv_prompt contains a prompt like 'Show me a picture of a unicorn reading
 an enterprise financial report'.
```

```
        DATA(lv_image) = lo_bdr_l2_sd->text_to_image( iv_prompt ).
      CATCH /aws1/cx_bdraccessdeniedex INTO DATA(lo_ex).
        WRITE / lo_ex->get_text( ).
        WRITE / |Don't forget to enable model access at https://
console.aws.amazon.com/bedrock/home?#/modelaccess|.

    ENDTRY.
```

- Untuk detail API, lihat InvokeModeldi AWS SDK untuk referensi SAP ABAP API.

Contoh Runtime Amazon Bedrock Agents menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Amazon Bedrock Agents Runtime.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`InvokeAgent`

Contoh kode berikut menunjukkan cara menggunakan`InvokeAgent`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
DATA(lo_result) = lo_bdz->invokeagent(
  iv_agentid      = iv_agentid
    iv_agentaliasid = iv_agentaliasid
    iv_enabletrace  = abap_true
    iv_sessionid    = CONV #( cl_system_uuid=>create_uuid_c26_static( ) )
    iv_inputtext    = |Let's play "rock, paper, scissors".  I choose rock.| ).
DATA(lo_stream) = lo_result->get_completion( ).
TRY.
    " loop while there are still events in the stream
    WHILE lo_stream->/aws1/if_rt_stream_reader~data_available( ) = abap_true.
      DATA(lo_evt) = lo_stream->read( ).
      " each /AWS1/CL_BDZRESPONSESTREAM_EV event contains exactly one member
      " all others are INITIAL.  For each event, process the non-initial
      " member if desired
      IF lo_evt->get_chunk( ) IS NOT INITIAL.
        " Process a Chunk event
        DATA(lv_xstr) = lo_evt->get_chunk( )->get_bytes( ).
        DATA(lv_answer) = /aws1/cl_rt_util=>xstring_to_string( lv_xstr ).
        " the answer says something like "I chose paper, so you lost"
      ELSEIF lo_evt->get_files( ) IS NOT INITIAL.
        " process a Files event if desired
      ELSEIF lo_evt->get_returncontrol( ) IS NOT INITIAL.
        " process a ReturnControl event if desired
      ELSEIF lo_evt->get_trace( ) IS NOT INITIAL.
        " process a Trace event if desired
      ENDIF.
    ENDWHILE.
    " the stream of events can possibly contain an exception
    " which will be raised to break the loop
    " catch /AWS1/CX_BDZACCESSDENIEDEX.
    " catch /AWS1/CX_BDZINTERNALSERVEREX.
    " catch /AWS1/CX_BDZMODELNOTREADYEX.
    " catch /AWS1/CX_BDZVALIDATIONEX.
```

```
        " catch /AWS1/CX_BDZTHROTTLINGEX.
        " catch /AWS1/CX_BDZDEPENDENCYFAILEDEX.
        " catch /AWS1/CX_BDZBADGATEWAYEX.
        " catch /AWS1/CX_BDZRESOURCENOTFOUNDEX.
        " catch /AWS1/CX_BDZSERVICEQUOTAEXCDEX.
        " catch /AWS1/CX_BDZCONFLICTEXCEPTION.
    ENDTRY.
```

- Untuk detail API, lihat InvokeAgentdi AWS SDK untuk referensi SAP ABAP API.

CloudFront contoh menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan mengimplementasikan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan. CloudFront

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`ListDistributions`

Contoh kode berikut menunjukkan cara menggunakan`ListDistributions`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat ListDistributionsdi AWS SDK untuk referensi SAP ABAP API.

`UpdateDistribution`

Contoh kode berikut menunjukkan cara menggunakan`UpdateDistribution`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

        " Create a new distribution config with the updated comment "
        " Since the config object is immutable, we need to create a new one with all
 existing values "
        DATA(lo_new_config) = NEW /aws1/cl_fntdistributionconfig(
          iv_callerreference = lo_old_config->get_callerreference( )
          io_aliases = lo_old_config->get_aliases( )
          iv_defaultrootobject = lo_old_config->get_defaultrootobject( )
          io_origins = lo_old_config->get_origins( )
          io_origingroups = lo_old_config->get_origingroups( )
          io_defaultcachebehavior = lo_old_config->get_defaultcachebehavior( )
```

```
        io_cachebehaviors = lo_old_config->get_cachebehaviors( )
        io_customerrorresponses = lo_old_config->get_customerrorresponses( )
        iv_comment = iv_comment
        io_logging = lo_old_config->get_logging( )
        iv_priceclass = lo_old_config->get_priceclass( )
        iv_enabled = lo_old_config->get_enabled( )
        io_viewercertificate = lo_old_config->get_viewercertificate( )
        io_restrictions = lo_old_config->get_restrictions( )
        iv_webaclid = lo_old_config->get_webaclid( )
        iv_httpversion = lo_old_config->get_httpversion( )
        iv_isipv6enabled = lo_old_config->get_isipv6enabled( ) ).

      " Update the distribution with the modified configuration "
      lo_fnt->updatedistribution(
        io_distributionconfig = lo_new_config
        iv_id = iv_distribution_id
        iv_ifmatch = lv_etag ).
      MESSAGE 'CloudFront distribution updated successfully.' TYPE 'I'.
    CATCH /aws1/cx_fntnosuchdistribution.
      MESSAGE 'Distribution does not exist.' TYPE 'E'.
    CATCH /aws1/cx_fntpreconditionfailed.
      MESSAGE 'Precondition failed - ETag mismatch.' TYPE 'E'.
    CATCH /aws1/cx_fntinvalidifmatchvrs.
      MESSAGE 'Invalid If-Match version.' TYPE 'E'.
  ENDTRY.
```

- Untuk detail API, lihat UpdateDistributiondi AWS SDK untuk referensi SAP ABAP API.

CloudWatch contoh menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan mengimplementasikan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan. CloudWatch

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Skenario adalah contoh kode yang menunjukkan kepada Anda bagaimana menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan atau dikombinasikan dengan yang lain Layanan AWS.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan
- Skenario

Tindakan

`DeleteAlarms`

Contoh kode berikut menunjukkan cara menggunakan`DeleteAlarms`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat DeleteAlarmsdi AWS SDK untuk referensi SAP ABAP API.

`DescribeAlarms`

Contoh kode berikut menunjukkan cara menggunakan`DescribeAlarms`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat DescribeAlarmsdi AWS SDK untuk referensi SAP ABAP API.

`DescribeAlarmsForMetric`

Contoh kode berikut menunjukkan cara menggunakan`DescribeAlarmsForMetric`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_cwt->describealarmsformetric(          " oo_result is
 returned for testing purposes. "
          iv_namespace   = iv_namespace
```

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

- Untuk detail API, lihat DescribeAlarmsForMetricdi AWS SDK untuk referensi SAP ABAP API.

`DisableAlarmActions`

Contoh kode berikut menunjukkan cara menggunakan`DisableAlarmActions`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat DisableAlarmActionsdi AWS SDK untuk referensi SAP ABAP API.

`EnableAlarmActions`

Contoh kode berikut menunjukkan cara menggunakan`EnableAlarmActions`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat EnableAlarmActionsdi AWS SDK untuk referensi SAP ABAP API.

`GetMetricStatistics`

Contoh kode berikut menunjukkan cara menggunakan`GetMetricStatistics`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat GetMetricStatisticsdi AWS SDK untuk referensi SAP ABAP API.

`ListMetrics`

Contoh kode berikut menunjukkan cara menggunakan`ListMetrics`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat ListMetricsdi AWS SDK untuk referensi SAP ABAP API.

`PutMetricAlarm`

Contoh kode berikut menunjukkan cara menggunakan`PutMetricAlarm`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat PutMetricAlarmdi AWS SDK untuk referensi SAP ABAP API.

`PutMetricData`

Contoh kode berikut menunjukkan cara menggunakan`PutMetricData`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

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

INSERT lo_metricdatum INTO TABLE lt_metricdata.

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

Masukkan satu set data ke dalam CloudWatch metrik.

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

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

```
    INSERT lo_metricdatum INTO TABLE lt_metricdata.

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

- Untuk detail API, lihat PutMetricDatadi AWS SDK untuk referensi SAP ABAP API.

Skenario

Memulai menggunakan alarm

Contoh kode berikut ini menunjukkan cara untuk melakukan:

- Membuat alarm.
- Menonaktifkan tindakan alarm.
- Menjelaskan maksud alarm.
- Menghapus alarm.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

    "Create an alarm"
    TRY.
```

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

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

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

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

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

- Untuk mengetahui hal detail mengenai API, silakan lihat topik-topik berikut di referensi API AWS SDK untuk ABAP SAP.
  - DeleteAlarms
  - DescribeAlarms
  - DisableAlarmActions
  - PutMetricAlarm

CloudWatch Contoh log menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan mengimplementasikan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Log. CloudWatch

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`GetQueryResults`

Contoh kode berikut menunjukkan cara menggunakan`GetQueryResults`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   TRY.
       oo_result = lo_cwl->getqueryresults(
         iv_queryid = iv_query_id ).

       " Display query status and result count
       DATA(lv_status) = oo_result->get_status( ).
       DATA(lt_results) = oo_result->get_results( ).
       DATA(lv_result_count) = lines( lt_results ).

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

- Untuk detail API, lihat GetQueryResultsdi AWS SDK untuk referensi SAP ABAP API.

`StartQuery`

Contoh kode berikut menunjukkan cara menggunakan`StartQuery`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

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

- Untuk detail API, lihat StartQuerydi AWS SDK untuk referensi SAP ABAP API.

Contoh Penyedia Identitas Amazon Cognito menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Amazon Cognito Identity Provider.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`AdminInitiateAuth`

Contoh kode berikut menunjukkan cara menggunakan`AdminInitiateAuth`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

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

        oo_result = lo_cgp->admininitiateauth(
```

```
          iv_userpoolid = iv_user_pool_id
          iv_clientid = iv_client_id
          iv_authflow = 'ADMIN_USER_PASSWORD_AUTH'
          it_authparameters = lt_auth_params
        ).

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

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

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

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

- Untuk detail API, lihat AdminInitiateAuthdi AWS SDK untuk referensi SAP ABAP API.

`AdminRespondToAuthChallenge`

Contoh kode berikut menunjukkan cara menggunakan`AdminRespondToAuthChallenge`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

```
            value = NEW /aws1/cl_cgpchallengerspstyp00( iv_user_name ) ) )
        ( VALUE /aws1/cl_cgpchallengerspstyp00=>ts_challengerspstype_maprow(
            key = 'SOFTWARE_TOKEN_MFA_CODE'
            value = NEW /aws1/cl_cgpchallengerspstyp00( iv_mfa_code ) ) )
      ).

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

      DATA(lo_result) = lo_cgp->adminrespondtoauthchallenge(
        iv_userpoolid = iv_user_pool_id
        iv_clientid = iv_client_id
        iv_challengename = 'SOFTWARE_TOKEN_MFA'
        it_challengeresponses = lt_challenge_responses
        iv_session = iv_session
      ).

      oo_auth_result = lo_result->get_authenticationresult( ).

      IF oo_auth_result IS BOUND.
        MESSAGE 'MFA challenge completed successfully.' TYPE 'I'.
      ELSE.
        " Another challenge might be required
        DATA(lv_next_challenge) = lo_result->get_challengename( ).
        MESSAGE |Additional challenge required: { lv_next_challenge }.| TYPE 'I'.
      ENDIF.

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

    CATCH /aws1/cx_cgpexpiredcodeex INTO DATA(lo_expired_ex).
      MESSAGE 'MFA code has expired.' TYPE 'E'.

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

- Untuk detail API, lihat AdminRespondToAuthChallengedi AWS SDK untuk referensi SAP ABAP API.

`AssociateSoftwareToken`

Contoh kode berikut menunjukkan cara menggunakan`AssociateSoftwareToken`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

    ov_secret_code = lo_result->get_secretcode( ).

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

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

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

- Untuk detail API, lihat AssociateSoftwareTokendi AWS SDK untuk referensi SAP ABAP API.

`ListUsers`

Contoh kode berikut menunjukkan cara menggunakan`ListUsers`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

    ot_users = lo_result->get_users( ).

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

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

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

- Untuk detail API, lihat ListUsersdi AWS SDK untuk referensi SAP ABAP API.

`VerifySoftwareToken`

Contoh kode berikut menunjukkan cara menggunakan`VerifySoftwareToken`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
```

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

      ov_status = lo_result->get_status( ).

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

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

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

- Untuk detail API, lihat VerifySoftwareTokendi AWS SDK untuk referensi SAP ABAP API.

Amazon Comprehend contoh menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Amazon Comprehend.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`CreateDocumentClassifier`

Contoh kode berikut menunjukkan cara menggunakan`CreateDocumentClassifier`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat CreateDocumentClassifierdi AWS SDK untuk referensi SAP ABAP API.

`DeleteDocumentClassifier`

Contoh kode berikut menunjukkan cara menggunakan`DeleteDocumentClassifier`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_cpd->deletedocumentclassifier(
      iv_documentclassifierarn = iv_classifier_arn
    ).
    MESSAGE 'Document classifier deleted.' TYPE 'I'.
  CATCH /aws1/cx_cpdinvalidrequestex.
    MESSAGE 'Invalid request.' TYPE 'E'.
  CATCH /aws1/cx_cpdtoomanyrequestsex.
    MESSAGE 'Too many requests.' TYPE 'E'.
  CATCH /aws1/cx_cpdresourcenotfoundex.
    MESSAGE 'Resource not found.' TYPE 'E'.
  CATCH /aws1/cx_cpdresourceinuseex.
    MESSAGE 'Resource in use.' TYPE 'E'.
  CATCH /aws1/cx_cpdinternalserverex.
    MESSAGE 'Internal server error occurred.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DeleteDocumentClassifierdi AWS SDK untuk referensi SAP ABAP API.

`DescribeDocumentClassificationJob`

Contoh kode berikut menunjukkan cara menggunakan`DescribeDocumentClassificationJob`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat DescribeDocumentClassificationJobdi AWS SDK untuk referensi SAP ABAP API.

`DescribeDocumentClassifier`

Contoh kode berikut menunjukkan cara menggunakan`DescribeDocumentClassifier`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
```

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

- Untuk detail API, lihat DescribeDocumentClassifierdi AWS SDK untuk referensi SAP ABAP API.

`DescribeTopicsDetectionJob`

Contoh kode berikut menunjukkan cara menggunakan`DescribeTopicsDetectionJob`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_cpd->describetopicsdetectionjob(
      iv_jobid = iv_job_id
    ).
    MESSAGE 'Topics detection job described.' TYPE 'I'.
  CATCH /aws1/cx_cpdinvalidrequestex.
    MESSAGE 'Invalid request.' TYPE 'E'.
  CATCH /aws1/cx_cpdjobnotfoundex.
    MESSAGE 'Job not found.' TYPE 'E'.
  CATCH /aws1/cx_cpdtoomanyrequestsex.
    MESSAGE 'Too many requests.' TYPE 'E'.
  CATCH /aws1/cx_cpdinternalserverex.
```

```
    MESSAGE 'Internal server error occurred.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DescribeTopicsDetectionJobdi AWS SDK untuk referensi SAP ABAP API.

`DetectDominantLanguage`

Contoh kode berikut menunjukkan cara menggunakan`DetectDominantLanguage`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat DetectDominantLanguagedi AWS SDK untuk referensi SAP ABAP API.

`DetectEntities`

Contoh kode berikut menunjukkan cara menggunakan`DetectEntities`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat DetectEntitiesdi AWS SDK untuk referensi SAP ABAP API.

`DetectKeyPhrases`

Contoh kode berikut menunjukkan cara menggunakan`DetectKeyPhrases`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
```

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

- Untuk detail API, lihat DetectKeyPhrasesdi AWS SDK untuk referensi SAP ABAP API.

`DetectPiiEntities`

Contoh kode berikut menunjukkan cara menggunakan`DetectPiiEntities`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

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

- Untuk detail API, lihat DetectPiiEntitiesdi AWS SDK untuk referensi SAP ABAP API.

`DetectSentiment`

Contoh kode berikut menunjukkan cara menggunakan`DetectSentiment`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat DetectSentimentdi AWS SDK untuk referensi SAP ABAP API.

`DetectSyntax`

Contoh kode berikut menunjukkan cara menggunakan`DetectSyntax`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat DetectSyntaxdi AWS SDK untuk referensi SAP ABAP API.

`ListDocumentClassificationJobs`

Contoh kode berikut menunjukkan cara menggunakan`ListDocumentClassificationJobs`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
```

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

- Untuk detail API, lihat ListDocumentClassificationJobsdi AWS SDK untuk referensi SAP ABAP API.

`ListDocumentClassifiers`

Contoh kode berikut menunjukkan cara menggunakan`ListDocumentClassifiers`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat ListDocumentClassifiersdi AWS SDK untuk referensi SAP ABAP API.

`ListTopicsDetectionJobs`

Contoh kode berikut menunjukkan cara menggunakan`ListTopicsDetectionJobs`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat ListTopicsDetectionJobsdi AWS SDK untuk referensi SAP ABAP API.

`StartDocumentClassificationJob`

Contoh kode berikut menunjukkan cara menggunakan`StartDocumentClassificationJob`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat StartDocumentClassificationJobdi AWS SDK untuk referensi SAP ABAP API.

`StartTopicsDetectionJob`

Contoh kode berikut menunjukkan cara menggunakan`StartTopicsDetectionJob`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat StartTopicsDetectionJobdi AWS SDK untuk referensi SAP ABAP API.

AWS Config contoh menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan mengimplementasikan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan. AWS Config

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`DeleteConfigRule`

Contoh kode berikut menunjukkan cara menggunakan`DeleteConfigRule`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat DeleteConfigRuledi AWS SDK untuk referensi SAP ABAP API.

`DescribeConfigRules`

Contoh kode berikut menunjukkan cara menggunakan`DescribeConfigRules`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat DescribeConfigRulesdi AWS SDK untuk referensi SAP ABAP API.

`PutConfigRule`

Contoh kode berikut menunjukkan cara menggunakan`PutConfigRule`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    " Create a config rule for S3 bucket public read prohibition
    lo_cfs->putconfigrule(
      io_configrule = NEW /aws1/cl_cfsconfigrule(
        iv_configrulename = iv_rule_name
        iv_description = |S3 Public Read Prohibited Bucket Rule|
        io_scope = NEW /aws1/cl_cfsscope(
          it_complianceresourcetypes = VALUE /aws1/
cl_cfscplncresrctypes_w=>tt_complianceresourcetypes(
```

```
          ( NEW /aws1/cl_cfscplncresrctypes_w( |AWS::S3::Bucket| ) )
        )
      )
      io_source = NEW /aws1/cl_cfssource(
        iv_owner = |AWS|
        iv_sourceidentifier = |S3_BUCKET_PUBLIC_READ_PROHIBITED|
      )
      iv_inputparameters = '{}'
      iv_configrulestate = |ACTIVE|
    )
  ).
  MESSAGE 'Created AWS Config rule.' TYPE 'I'.
```

- Untuk detail API, lihat PutConfigRuledi AWS SDK untuk referensi SAP ABAP API.

AWS Control Tower contoh menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan mengimplementasikan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan. AWS Control Tower

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`DisableBaseline`

Contoh kode berikut menunjukkan cara menggunakan`DisableBaseline`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

        DATA(lv_operation_id) = lo_output->get_operationidentifier( ).

        " Wait for operation to complete
        DATA lv_status TYPE /aws1/cttbaselineopstatus.
        DO 100 TIMES.
          lv_status = get_baseline_operation(
            io_ctt = io_ctt
            iv_operation_id = lv_operation_id
          ).

          DATA(lv_msg) = |Baseline operation status: { lv_status }|.
          MESSAGE lv_msg TYPE 'I'.

          IF lv_status = 'SUCCEEDED' OR lv_status = 'FAILED'.
            EXIT.
          ENDIF.

          " Wait 30 seconds
          WAIT UP TO 30 SECONDS.
        ENDDO.

        ov_operation_id = lv_operation_id.
        MESSAGE 'Baseline disabled successfully.' TYPE 'I'.
      CATCH /aws1/cx_cttconflictexception INTO DATA(lo_conflict).
        " Log conflict but don't fail - return empty operation ID
        DATA(lv_msg2) = |Conflict disabling baseline: { lo_conflict->get_text( ) }.
Skipping disable step.|.
        MESSAGE lv_msg2 TYPE 'I'.
```

```
    CLEAR ov_operation_id.
ENDTRY.
```

- Untuk detail API, lihat DisableBaselinedi AWS SDK untuk referensi SAP ABAP API.

`DisableControl`

Contoh kode berikut menunjukkan cara menggunakan`DisableControl`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
" Disable the control
DATA(lo_output) = io_ctt->disablecontrol(
  iv_controlidentifier = iv_control_arn
  iv_targetidentifier  = iv_target_identifier
).

DATA(lv_operation_id) = lo_output->get_operationidentifier( ).

" Wait for operation to complete
DATA lv_status TYPE /aws1/cttcontrolopstatus.
DO 100 TIMES.
  lv_status = get_control_operation(
    io_ctt = io_ctt
    iv_operation_id = lv_operation_id
  ).

  DATA(lv_msg) = |Control operation status: { lv_status }|.
  MESSAGE lv_msg TYPE 'I'.

  IF lv_status = 'SUCCEEDED' OR lv_status = 'FAILED'.
    EXIT.
  ENDIF.

  " Wait 30 seconds
```

```
      WAIT UP TO 30 SECONDS.
    ENDDO.

    ov_operation_id = lv_operation_id.
    MESSAGE 'Control disabled successfully.' TYPE 'I'.
```

- Untuk detail API, lihat DisableControldi AWS SDK untuk referensi SAP ABAP API.

`EnableBaseline`

Contoh kode berikut menunjukkan cara menggunakan`EnableBaseline`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    " Prepare parameters for enabling baseline
    DATA lt_parameters TYPE /aws1/
cl_cttenbdbaselineparam=>tt_enabledbaselineparameters.

    " Add Identity Center baseline parameter if provided
    IF iv_identity_center_baseline IS NOT INITIAL.
      " Create a JSON document with the baseline ARN value
      DATA(lv_json) = |\{ "IdentityCenterEnabledBaselineArn":
 "{ iv_identity_center_baseline }" \}|.
      DATA(lo_param) = NEW /aws1/cl_cttenbdbaselineparam(
        iv_key = 'IdentityCenterEnabledBaselineArn'
        io_value = /aws1/cl_rt_document=>from_json_str( lv_json )
      ).
      APPEND lo_param TO lt_parameters.
    ENDIF.

    " Enable the baseline
    DATA(lo_output) = io_ctt->enablebaseline(
      iv_baselineidentifier = iv_baseline_identifier
      iv_baselineversion    = iv_baseline_version
```

```
      iv_targetidentifier   = iv_target_identifier
      it_parameters         = lt_parameters
    ).

    DATA(lv_operation_id) = lo_output->get_operationidentifier( ).

    " Wait for operation to complete
    DATA lv_status TYPE /aws1/cttbaselineopstatus.
    DO 100 TIMES.
      lv_status = get_baseline_operation(
        io_ctt = io_ctt
        iv_operation_id = lv_operation_id
      ).

      DATA(lv_msg) = |Baseline operation status: { lv_status }|.
      MESSAGE lv_msg TYPE 'I'.

      IF lv_status = 'SUCCEEDED' OR lv_status = 'FAILED'.
        EXIT.
      ENDIF.

      " Wait 30 seconds
      WAIT UP TO 30 SECONDS.
    ENDDO.

    ov_enabled_baseline_arn = lo_output->get_arn( ).
    MESSAGE 'Baseline enabled successfully.' TYPE 'I'.
```

- Untuk detail API, lihat EnableBaselinedi AWS SDK untuk referensi SAP ABAP API.

`EnableControl`

Contoh kode berikut menunjukkan cara menggunakan`EnableControl`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
" Enable the control
DATA(lo_output) = io_ctt->enablecontrol(
  iv_controlidentifier = iv_control_arn
  iv_targetidentifier  = iv_target_identifier
).

DATA(lv_operation_id) = lo_output->get_operationidentifier( ).

" Wait for operation to complete
DATA lv_status TYPE /aws1/cttcontrolopstatus.
DO 100 TIMES.
  lv_status = get_control_operation(
    io_ctt = io_ctt
    iv_operation_id = lv_operation_id
  ).

  DATA(lv_msg) = |Control operation status: { lv_status }|.
  MESSAGE lv_msg TYPE 'I'.

  IF lv_status = 'SUCCEEDED' OR lv_status = 'FAILED'.
    EXIT.
  ENDIF.

  " Wait 30 seconds
  WAIT UP TO 30 SECONDS.
ENDDO.

ov_operation_id = lv_operation_id.
MESSAGE 'Control enabled successfully.' TYPE 'I'.
```

- Untuk detail API, lihat EnableControldi AWS SDK untuk referensi SAP ABAP API.

`GetBaselineOperation`

Contoh kode berikut menunjukkan cara menggunakan`GetBaselineOperation`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

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

- Untuk detail API, lihat GetBaselineOperationdi AWS SDK untuk referensi SAP ABAP API.

`GetControlOperation`

Contoh kode berikut menunjukkan cara menggunakan`GetControlOperation`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

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

- Untuk detail API, lihat GetControlOperationdi AWS SDK untuk referensi SAP ABAP API.

`ListBaselines`

Contoh kode berikut menunjukkan cara menggunakan`ListBaselines`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
DATA lt_baselines TYPE /aws1/cl_cttbaselinesummary=>tt_baselines.
DATA lv_nexttoken TYPE /aws1/cttstring.

" List all baselines using pagination
DO.
  DATA(lo_output) = io_ctt->listbaselines(
    iv_nexttoken = lv_nexttoken
  ).

  APPEND LINES OF lo_output->get_baselines( ) TO lt_baselines.

  lv_nexttoken = lo_output->get_nexttoken( ).
  IF lv_nexttoken IS INITIAL.
    EXIT.
  ENDIF.
ENDDO.

ot_baselines = lt_baselines.
MESSAGE 'Listed baselines successfully.' TYPE 'I'.
```

- Untuk detail API, lihat ListBaselinesdi AWS SDK untuk referensi SAP ABAP API.

`ListEnabledBaselines`

Contoh kode berikut menunjukkan cara menggunakan`ListEnabledBaselines`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    DATA lt_enabled_baselines TYPE /aws1/
cl_cttenbdbaselinesumm=>tt_enabledbaselines.
    DATA lv_nexttoken TYPE /aws1/cttlstenbdbaselinesnex00.

    " List all enabled baselines using pagination
    DO.
      DATA(lo_output) = io_ctt->listenabledbaselines(
        iv_nexttoken = lv_nexttoken
      ).

      APPEND LINES OF lo_output->get_enabledbaselines( ) TO lt_enabled_baselines.

      lv_nexttoken = lo_output->get_nexttoken( ).
      IF lv_nexttoken IS INITIAL.
        EXIT.
      ENDIF.
    ENDDO.

    ot_enabled_baselines = lt_enabled_baselines.
    MESSAGE 'Listed enabled baselines successfully.' TYPE 'I'.
```

- Untuk detail API, lihat ListEnabledBaselinesdi AWS SDK untuk referensi SAP ABAP API.

`ListEnabledControls`

Contoh kode berikut menunjukkan cara menggunakan`ListEnabledControls`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    DATA lt_enabled_controls TYPE /aws1/
cl_cttenabledcontrolsumm=>tt_enabledcontrols.
    DATA lv_nexttoken TYPE /aws1/cttstring.

    " List all enabled controls using pagination
    DO.
      DATA(lo_output) = io_ctt->listenabledcontrols(
        iv_targetidentifier = iv_target_identifier
        iv_nexttoken        = lv_nexttoken
      ).

      APPEND LINES OF lo_output->get_enabledcontrols( ) TO lt_enabled_controls.

      lv_nexttoken = lo_output->get_nexttoken( ).
      IF lv_nexttoken IS INITIAL.
        EXIT.
      ENDIF.
    ENDDO.

    ot_enabled_controls = lt_enabled_controls.
    MESSAGE 'Listed enabled controls successfully.' TYPE 'I'.
```

- Untuk detail API, lihat ListEnabledControlsdi AWS SDK untuk referensi SAP ABAP API.

`ListLandingZones`

Contoh kode berikut menunjukkan cara menggunakan`ListLandingZones`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    DATA lt_landing_zones TYPE /aws1/
cl_cttlandingzonesummary=>tt_landingzonesummaries.
    DATA lv_nexttoken TYPE /aws1/cttstring.

    " List all landing zones using pagination
    DO.
      DATA(lo_output) = io_ctt->listlandingzones(
        iv_nexttoken = lv_nexttoken
      ).

      APPEND LINES OF lo_output->get_landingzones( ) TO lt_landing_zones.

      lv_nexttoken = lo_output->get_nexttoken( ).
      IF lv_nexttoken IS INITIAL.
        EXIT.
      ENDIF.
    ENDDO.

    ot_landing_zones = lt_landing_zones.
    MESSAGE 'Listed landing zones successfully.' TYPE 'I'.
```

- Untuk detail API, lihat ListLandingZonesdi AWS SDK untuk referensi SAP ABAP API.

`ResetEnabledBaseline`

Contoh kode berikut menunjukkan cara menggunakan`ResetEnabledBaseline`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
" Reset the enabled baseline
DATA(lo_output) = io_ctt->resetenabledbaseline(
  iv_enabledbaselineidentifier = iv_enabled_baseline_identifier
).

DATA(lv_operation_id) = lo_output->get_operationidentifier( ).

" Wait for operation to complete
DATA lv_status TYPE /aws1/cttbaselineopstatus.
DO 100 TIMES.
  lv_status = get_baseline_operation(
    io_ctt = io_ctt
    iv_operation_id = lv_operation_id
  ).

  DATA(lv_msg) = |Baseline operation status: { lv_status }|.
  MESSAGE lv_msg TYPE 'I'.

  IF lv_status = 'SUCCEEDED' OR lv_status = 'FAILED'.
    EXIT.
  ENDIF.

  " Wait 30 seconds
  WAIT UP TO 30 SECONDS.
ENDDO.

ov_operation_id = lv_operation_id.
MESSAGE 'Baseline reset successfully.' TYPE 'I'.
```

- Untuk detail API, lihat ResetEnabledBaselinedi AWS SDK untuk referensi SAP ABAP API.

Contoh Firehose menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan mengimplementasikan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Firehose.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`PutRecord`

Contoh kode berikut menunjukkan cara menggunakan`PutRecord`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        DATA(lo_record) = NEW /aws1/cl_frhrecord( iv_data = iv_data ).

        DATA(lo_result) = lo_frh->putrecord(
          iv_deliverystreamname = iv_deliv_stream_name
          io_record             = lo_record ).

        MESSAGE 'Record sent to Firehose delivery stream.' TYPE 'I'.
      CATCH /aws1/cx_frhresourcenotfoundex.
        MESSAGE 'Delivery stream not found.' TYPE 'E'.
      CATCH /aws1/cx_frhinvalidargumentex.
```

```
        MESSAGE 'Invalid argument provided.' TYPE 'E'.
      CATCH /aws1/cx_frhserviceunavailex.
        MESSAGE 'Service temporarily unavailable.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat PutRecorddi AWS SDK untuk referensi SAP ABAP API.

`PutRecordBatch`

Contoh kode berikut menunjukkan cara menggunakan`PutRecordBatch`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

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

        IF lv_failed_count > 0.
          MESSAGE |{ lv_failed_count } records failed to send.| TYPE 'I'.
        ELSE.
          MESSAGE 'All records sent successfully to Firehose delivery stream.' TYPE
'I'.
        ENDIF.
      CATCH /aws1/cx_frhresourcenotfoundex.
        MESSAGE 'Delivery stream not found.' TYPE 'E'.
      CATCH /aws1/cx_frhinvalidargumentex.
        MESSAGE 'Invalid argument provided.' TYPE 'E'.
      CATCH /aws1/cx_frhserviceunavailex.
        MESSAGE 'Service temporarily unavailable.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat PutRecordBatchdi AWS SDK untuk referensi SAP ABAP API.

Contoh DynamoDB menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan mengimplementasikan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan DynamoDB.

Dasar-dasar adalah contoh kode yang menunjukkan kepada Anda bagaimana melakukan operasi penting dalam suatu layanan.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Hal-hal mendasar
- Tindakan

Hal-hal mendasar

Pelajari dasar-dasarnya

Contoh kode berikut ini menunjukkan cara untuk melakukan:

- Buat tabel yang dapat menyimpan data film.
- Masukkan, dapatkan, dan perbarui satu film dalam tabel tersebut.
- Tulis data film ke tabel dari file JSON sampel.
- Kueri untuk film yang dirilis pada tahun tertentu.
- Pindai film yang dirilis dalam suatu rentang tahun.
- Hapus film dari tabel, lalu hapus tabel tersebut.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    " Create an Amazon Dynamo DB table.

    TRY.
        DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
        DATA(lo_dyn) = /aws1/cl_dyn_factory=>create( lo_session ).
        DATA(lt_keyschema) = VALUE /aws1/cl_dynkeyschemaelement=>tt_keyschema(
          ( NEW /aws1/cl_dynkeyschemaelement( iv_attributename = 'year'
                                              iv_keytype = 'HASH' ) )
          ( NEW /aws1/cl_dynkeyschemaelement( iv_attributename = 'title'
                                              iv_keytype = 'RANGE' ) ) ).
        DATA(lt_attributedefinitions) = VALUE /aws1/
cl_dynattributedefn=>tt_attributedefinitions(
          ( NEW /aws1/cl_dynattributedefn( iv_attributename = 'year'
                                           iv_attributetype = 'N' ) )
          ( NEW /aws1/cl_dynattributedefn( iv_attributename = 'title'
                                           iv_attributetype = 'S' ) ) ).

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

```
        DATA(lv_error) = |"{ lo_resourceinuseex->av_err_code }" -
 { lo_resourceinuseex->av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.

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

    " Put items into the table.
    TRY.
        DATA(lo_resp_putitem) = lo_dyn->putitem(
          iv_tablename = iv_table_name
          it_item      = VALUE /aws1/
cl_dynattributevalue=>tt_putiteminputattributemap(
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
              key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s =
 'Jaws' ) ) )
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
              key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '1975' }| ) ) )
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
              key = 'rating' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '7.5' }| ) ) )
          ) ).
        lo_resp_putitem = lo_dyn->putitem(
          iv_tablename = iv_table_name
          it_item      = VALUE /aws1/
cl_dynattributevalue=>tt_putiteminputattributemap(
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
              key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s = 'Star
 Wars' ) ) )
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
              key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '1978' }| ) ) )
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
              key = 'rating' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '8.1' }| ) ) )
          ) ).
```

```
        lo_resp_putitem = lo_dyn->putitem(
          iv_tablename = iv_table_name
          it_item      = VALUE /aws1/
cl_dynattributevalue=>tt_putiteminputattributemap(
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
              key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s =
 'Speed' ) ) )
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
              key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '1994' }| ) ) )
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
              key = 'rating' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '7.9' }| ) ) )
          ) ).
        " TYPE REF TO /AWSEX/CL_AWS1_dyn_PUT_ITEM_OUTPUT
        MESSAGE '3 rows inserted into DynamoDB Table' && iv_table_name TYPE 'I'.
      CATCH /aws1/cx_dyncondalcheckfaile00.
        MESSAGE 'A condition specified in the operation could not be evaluated.'
 TYPE 'E'.
      CATCH /aws1/cx_dynresourcenotfoundex.
        MESSAGE 'The table or index does not exist' TYPE 'E'.
      CATCH /aws1/cx_dyntransactconflictex.
        MESSAGE 'Another transaction is using the item' TYPE 'E'.
    ENDTRY.

    " Get item from table.
    TRY.
        DATA(lo_resp_getitem) = lo_dyn->getitem(
          iv_tablename               = iv_table_name
          it_key                     = VALUE /aws1/cl_dynattributevalue=>tt_key(
           ( VALUE /aws1/cl_dynattributevalue=>ts_key_maprow(
             key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s =
 'Jaws' ) ) )
           ( VALUE /aws1/cl_dynattributevalue=>ts_key_maprow(
             key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n =
 '1975' ) ) )
          ) ).
        DATA(lt_attr) = lo_resp_getitem->get_item( ).
        DATA(lo_title) = lt_attr[ key = 'title' ]-value.
        DATA(lo_year) = lt_attr[ key = 'year' ]-value.
        DATA(lo_rating) = lt_attr[ key = 'year' ]-value.
        MESSAGE 'Movie name is: ' && lo_title->get_s( ) TYPE 'I'.
        MESSAGE 'Movie year is: ' && lo_year->get_n( ) TYPE 'I'.
        MESSAGE 'Movie rating is: ' && lo_rating->get_n( ) TYPE 'I'.
```

```
      CATCH /aws1/cx_dynresourcenotfoundex.
        MESSAGE 'The table or index does not exist' TYPE 'E'.
    ENDTRY.

    " Query item from table.
    TRY.
        DATA(lt_attributelist) = VALUE /aws1/
cl_dynattributevalue=>tt_attributevaluelist(
              ( NEW /aws1/cl_dynattributevalue( iv_n = '1975' ) ) ).
        DATA(lt_keyconditions) = VALUE /aws1/cl_dyncondition=>tt_keyconditions(
          ( VALUE /aws1/cl_dyncondition=>ts_keyconditions_maprow(
          key = 'year'
          value = NEW /aws1/cl_dyncondition(
            it_attributevaluelist = lt_attributelist
            iv_comparisonoperator = |EQ|
          ) ) ) ).
        DATA(lo_query_result) = lo_dyn->query(
          iv_tablename = iv_table_name
          it_keyconditions = lt_keyconditions ).
        DATA(lt_items) = lo_query_result->get_items( ).
        READ TABLE lo_query_result->get_items( ) INTO DATA(lt_item) INDEX 1.
        lo_title = lt_item[ key = 'title' ]-value.
        lo_year = lt_item[ key = 'year' ]-value.
        lo_rating = lt_item[ key = 'rating' ]-value.
        MESSAGE 'Movie name is: ' && lo_title->get_s( ) TYPE 'I'.
        MESSAGE 'Movie year is: ' && lo_year->get_n( ) TYPE 'I'.
        MESSAGE 'Movie rating is: ' && lo_rating->get_n( ) TYPE 'I'.
      CATCH /aws1/cx_dynresourcenotfoundex.
        MESSAGE 'The table or index does not exist' TYPE 'E'.
    ENDTRY.

    " Scan items from table.
    TRY.
        DATA(lo_scan_result) = lo_dyn->scan( iv_tablename = iv_table_name ).
        lt_items = lo_scan_result->get_items( ).
        " Read the first item and display the attributes.
        READ TABLE lo_query_result->get_items( ) INTO lt_item INDEX 1.
        lo_title = lt_item[ key = 'title' ]-value.
        lo_year = lt_item[ key = 'year' ]-value.
        lo_rating = lt_item[ key = 'rating' ]-value.
        MESSAGE 'Movie name is: ' && lo_title->get_s( ) TYPE 'I'.
        MESSAGE 'Movie year is: ' && lo_year->get_n( ) TYPE 'I'.
        MESSAGE 'Movie rating is: ' && lo_rating->get_n( ) TYPE 'I'.
      CATCH /aws1/cx_dynresourcenotfoundex.
```

```
        MESSAGE 'The table or index does not exist' TYPE 'E'.
    ENDTRY.

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

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

- Untuk mengetahui hal detail mengenai API, silakan lihat topik-topik berikut di referensi API AWS SDK untuk ABAP SAP.
  - BatchWriteItem
  - CreateTable
  - DeleteItem
  - DeleteTable
  - DescribeTable
  - GetItem
  - PutItem
  - Kueri
  - Scan
  - UpdateItem

Tindakan

`CreateTable`

Contoh kode berikut menunjukkan cara menggunakan`CreateTable`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat CreateTabledi AWS SDK untuk referensi SAP ABAP API.

`DeleteItem`

Contoh kode berikut menunjukkan cara menggunakan`DeleteItem`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat DeleteItemdi AWS SDK untuk referensi SAP ABAP API.

`DeleteTable`

Contoh kode berikut menunjukkan cara menggunakan`DeleteTable`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat DeleteTabledi AWS SDK untuk referensi SAP ABAP API.

`DescribeTable`

Contoh kode berikut menunjukkan cara menggunakan`DescribeTable`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat DescribeTabledi AWS SDK untuk referensi SAP ABAP API.

`GetItem`

Contoh kode berikut menunjukkan cara menggunakan`GetItem`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat GetItemdi AWS SDK untuk referensi SAP ABAP API.

`ListTables`

Contoh kode berikut menunjukkan cara menggunakan`ListTables`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat ListTablesdi AWS SDK untuk referensi SAP ABAP API.

`PutItem`

Contoh kode berikut menunjukkan cara menggunakan`PutItem`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat PutItemdi AWS SDK untuk referensi SAP ABAP API.

`Kueri`

Contoh kode berikut menunjukkan cara menggunakan`Query`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " Query movies for a given year .
        DATA(lt_attributelist) = VALUE /aws1/
cl_dynattributevalue=>tt_attributevaluelist(
            ( NEW /aws1/cl_dynattributevalue( iv_n = |{ iv_year }| ) ) ).
        DATA(lt_key_conditions) = VALUE /aws1/cl_dyncondition=>tt_keyconditions(
          ( VALUE /aws1/cl_dyncondition=>ts_keyconditions_maprow(
          key = 'year'
          value = NEW /aws1/cl_dyncondition(
          it_attributevaluelist = lt_attributelist
          iv_comparisonoperator = |EQ|
          ) ) ) ).
        oo_result = lo_dyn->query(
          iv_tablename = iv_table_name
          it_keyconditions = lt_key_conditions ).
        DATA(lt_items) = oo_result->get_items( ).
        "You can loop over the results to get item attributes.
        LOOP AT lt_items INTO DATA(lt_item).
          DATA(lo_title) = lt_item[ key = 'title' ]-value.
          DATA(lo_year) = lt_item[ key = 'year' ]-value.
        ENDLOOP.
        DATA(lv_count) = oo_result->get_count( ).
        MESSAGE 'Item count is: ' && lv_count TYPE 'I'.
      CATCH /aws1/cx_dynresourcenotfoundex.
        MESSAGE 'The table or index does not exist' TYPE 'E'.
```

```
    ENDTRY.
```

- Untuk detail API, lihat Kueri di referensi API AWS SDK untuk SAP ABAP.

Scan

Contoh kode berikut menunjukkan cara menggunakan`Scan`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " Scan movies for rating greater than or equal to the rating specified
        DATA(lt_attributelist) = VALUE /aws1/
cl_dynattributevalue=>tt_attributevaluelist(
            ( NEW /aws1/cl_dynattributevalue( iv_n = |{ iv_rating }| ) ) ).
        DATA(lt_filter_conditions) = VALUE /aws1/
cl_dyncondition=>tt_filterconditionmap(
          ( VALUE /aws1/cl_dyncondition=>ts_filterconditionmap_maprow(
          key = 'rating'
          value = NEW /aws1/cl_dyncondition(
          it_attributevaluelist = lt_attributelist
          iv_comparisonoperator = |GE|
          ) ) ) ).
        oo_scan_result = lo_dyn->scan( iv_tablename = iv_table_name
          it_scanfilter = lt_filter_conditions ).
        DATA(lt_items) = oo_scan_result->get_items( ).
        LOOP AT lt_items INTO DATA(lo_item).
          " You can loop over to get individual attributes.
          DATA(lo_title) = lo_item[ key = 'title' ]-value.
          DATA(lo_year) = lo_item[ key = 'year' ]-value.
        ENDLOOP.
        DATA(lv_count) = oo_scan_result->get_count( ).
        MESSAGE 'Found ' && lv_count && ' items' TYPE 'I'.
      CATCH /aws1/cx_dynresourcenotfoundex.
        MESSAGE 'The table or index does not exist' TYPE 'E'.
```

```
    ENDTRY.
```

- Untuk detail API, lihat Scan di Referensi API AWS SDK untuk SAP ABAP.

`UpdateItem`

Contoh kode berikut menunjukkan cara menggunakan`UpdateItem`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat UpdateItemdi AWS SDK untuk referensi SAP ABAP API.

Contoh Amazon EC2 menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Amazon EC2.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`AllocateAddress`

Contoh kode berikut menunjukkan cara menggunakan`AllocateAddress`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat AllocateAddressdi AWS SDK untuk referensi SAP ABAP API.

`AssociateAddress`

Contoh kode berikut menunjukkan cara menggunakan`AssociateAddress`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat AssociateAddressdi AWS SDK untuk referensi SAP ABAP API.

`AuthorizeSecurityGroupIngress`

Contoh kode berikut menunjukkan cara menggunakan`AuthorizeSecurityGroupIngress`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    " Create IP permissions for SSH access (port 22)
    " iv_cidr_ip = '192.0.2.0/24'
    DATA lt_ip_permissions TYPE /aws1/cl_ec2ippermission=>tt_ippermissionlist.
    DATA(lo_ip_permission) = NEW /aws1/cl_ec2ippermission(
```

```
      iv_ipprotocol = 'tcp'
      iv_fromport = 22
      iv_toport = 22
      it_ipranges = VALUE /aws1/cl_ec2iprange=>tt_iprangelist(
        ( NEW /aws1/cl_ec2iprange( iv_cidrip = iv_cidr_ip ) )
      )
    ).
    APPEND lo_ip_permission TO lt_ip_permissions.

    TRY.
        oo_result = lo_ec2->authsecuritygroupingress(             " oo_result is
 returned for testing purposes. "
          iv_groupid = iv_group_id
          it_ippermissions = lt_ip_permissions ).
        MESSAGE 'Authorized ingress rule for security group.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat AuthorizeSecurityGroupIngressdi AWS SDK untuk referensi SAP ABAP API.

`CreateKeyPair`

Contoh kode berikut menunjukkan cara menggunakan`CreateKeyPair`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_ec2->createkeypair( iv_keyname = iv_key_name ).
              " oo_result is returned for testing purposes. "
        MESSAGE 'Amazon EC2 key pair created.' TYPE 'I'.
```

```
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat CreateKeyPairdi AWS SDK untuk referensi SAP ABAP API.

`CreateSecurityGroup`

Contoh kode berikut menunjukkan cara menggunakan`CreateSecurityGroup`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat CreateSecurityGroupdi AWS SDK untuk referensi SAP ABAP API.

`CreateVpc`

Contoh kode berikut menunjukkan cara menggunakan`CreateVpc`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    " iv_cidr_block = '10.0.0.0/16'
    TRY.
        oo_result = lo_ec2->createvpc( iv_cidrblock = iv_cidr_block ).            "
 oo_result is returned for testing purposes. "
        DATA(lv_vpc_id) = oo_result->get_vpc( )->get_vpcid( ).
        MESSAGE 'Created VPC.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat CreateVpcdi AWS SDK untuk referensi SAP ABAP API.

`CreateVpcEndpoint`

Contoh kode berikut menunjukkan cara menggunakan`CreateVpcEndpoint`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    " iv_vpc_id = 'vpc-abc123'
    " iv_service_name = 'com.amazonaws.region.service'
    TRY.
        oo_result = lo_ec2->createvpcendpoint(             " oo_result is returned
 for testing purposes. "
```

```
          iv_vpcid = iv_vpc_id
          iv_servicename = iv_service_name
          it_routetableids = it_route_table_ids ).
        DATA(lv_vpc_endpoint_id) = oo_result->get_vpcendpoint( )-
>get_vpcendpointid( ).
        MESSAGE 'Created VPC endpoint.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat CreateVpcEndpointdi AWS SDK untuk referensi SAP ABAP API.

`DeleteKeyPair`

Contoh kode berikut menunjukkan cara menggunakan`DeleteKeyPair`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat DeleteKeyPairdi AWS SDK untuk referensi SAP ABAP API.

`DeleteSecurityGroup`

Contoh kode berikut menunjukkan cara menggunakan`DeleteSecurityGroup`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat DeleteSecurityGroupdi AWS SDK untuk referensi SAP ABAP API.

`DeleteVpc`

Contoh kode berikut menunjukkan cara menggunakan`DeleteVpc`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        lo_ec2->deletevpc( iv_vpcid = iv_vpc_id ).
        MESSAGE 'Deleted VPC.' TYPE 'I'.
```

```
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat DeleteVpcdi AWS SDK untuk referensi SAP ABAP API.

`DeleteVpcEndpoints`

Contoh kode berikut menunjukkan cara menggunakan`DeleteVpcEndpoints`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        lo_ec2->deletevpcendpoints( it_vpcendpointids = it_vpc_endpoint_ids ).
        MESSAGE 'Deleted VPC endpoint(s).' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat DeleteVpcEndpointsdi AWS SDK untuk referensi SAP ABAP API.

`DescribeAddresses`

Contoh kode berikut menunjukkan cara menggunakan`DescribeAddresses`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat DescribeAddressesdi AWS SDK untuk referensi SAP ABAP API.

`DescribeAvailabilityZones`

Contoh kode berikut menunjukkan cara menggunakan`DescribeAvailabilityZones`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat DescribeAvailabilityZonesdi AWS SDK untuk referensi SAP ABAP API.

`DescribeImages`

Contoh kode berikut menunjukkan cara menggunakan`DescribeImages`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_ec2->describeimages( it_imageids = it_image_ids ).
  " oo_result is returned for testing purposes. "
        DATA(lt_images) = oo_result->get_images( ).
        MESSAGE 'Retrieved information about Amazon Machine Images (AMIs).' TYPE
 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat DescribeImagesdi AWS SDK untuk referensi SAP ABAP API.

`DescribeInstanceTypes`

Contoh kode berikut menunjukkan cara menggunakan`DescribeInstanceTypes`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    " Create filters for architecture and instance type patterns
    " iv_architecture = 'x86_64'
    DATA lt_filters TYPE /aws1/cl_ec2filter=>tt_filterlist.
    APPEND NEW /aws1/cl_ec2filter(
      iv_name = 'processor-info.supported-architecture'
      it_values = VALUE /aws1/cl_ec2valuestringlist_w=>tt_valuestringlist(
        ( NEW /aws1/cl_ec2valuestringlist_w( iv_architecture ) )
      )
    ) TO lt_filters.
    " Filter for instance type patterns like '*.micro', '*.small'
    APPEND NEW /aws1/cl_ec2filter(
      iv_name = 'instance-type'
      it_values = VALUE /aws1/cl_ec2valuestringlist_w=>tt_valuestringlist(
        ( NEW /aws1/cl_ec2valuestringlist_w( '*.micro' ) )
        ( NEW /aws1/cl_ec2valuestringlist_w( '*.small' ) )
      )
    ) TO lt_filters.

    TRY.
        oo_result = lo_ec2->describeinstancetypes( it_filters = lt_filters ).
      " oo_result is returned for testing purposes. "
        DATA(lt_instance_types) = oo_result->get_instancetypes( ).
        MESSAGE 'Retrieved information about EC2 instance types.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat DescribeInstanceTypesdi AWS SDK untuk referensi SAP ABAP API.

`DescribeInstances`

Contoh kode berikut menunjukkan cara menggunakan`DescribeInstances`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

        " Retrieving details of EC2 instances. "
        DATA: lv_istance_id    TYPE /aws1/ec2string,
              lv_status        TYPE /aws1/ec2instancestatename,
              lv_instance_type TYPE /aws1/ec2instancetype,
              lv_image_id      TYPE /aws1/ec2string.
        LOOP AT oo_result->get_reservations( ) INTO DATA(lo_reservation).
          LOOP AT lo_reservation->get_instances( ) INTO DATA(lo_instance).
            lv_istance_id = lo_instance->get_instanceid( ).
            lv_status = lo_instance->get_state( )->get_name( ).
            lv_instance_type = lo_instance->get_instancetype( ).
            lv_image_id = lo_instance->get_imageid( ).
          ENDLOOP.
        ENDLOOP.
        MESSAGE 'Retrieved information about EC2 instances.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat DescribeInstancesdi AWS SDK untuk referensi SAP ABAP API.

`DescribeKeyPairs`

Contoh kode berikut menunjukkan cara menggunakan`DescribeKeyPairs`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat DescribeKeyPairsdi AWS SDK untuk referensi SAP ABAP API.

`DescribeRegions`

Contoh kode berikut menunjukkan cara menggunakan`DescribeRegions`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat DescribeRegionsdi AWS SDK untuk referensi SAP ABAP API.

`DescribeRouteTables`

Contoh kode berikut menunjukkan cara menggunakan`DescribeRouteTables`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    " Create filter for VPC ID
    " iv_vpc_id = 'vpc-abc123'
    DATA lt_filters TYPE /aws1/cl_ec2filter=>tt_filterlist.
    APPEND NEW /aws1/cl_ec2filter(
      iv_name = 'vpc-id'
      it_values = VALUE /aws1/cl_ec2valuestringlist_w=>tt_valuestringlist(
        ( NEW /aws1/cl_ec2valuestringlist_w( iv_vpc_id ) )
      )
    ) TO lt_filters.

    TRY.
        oo_result = lo_ec2->describeroutetables( it_filters = lt_filters ).
    " oo_result is returned for testing purposes. "
        DATA(lt_route_tables) = oo_result->get_routetables( ).
        MESSAGE 'Retrieved information about route tables.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
```

```
    ENDTRY.
```

- Untuk detail API, lihat DescribeRouteTablesdi AWS SDK untuk referensi SAP ABAP API.

`DescribeSecurityGroups`

Contoh kode berikut menunjukkan cara menggunakan`DescribeSecurityGroups`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        DATA lt_group_ids TYPE /aws1/cl_ec2groupidstrlist_w=>tt_groupidstringlist.
        APPEND NEW /aws1/cl_ec2groupidstrlist_w( iv_value = iv_group_id ) TO
 lt_group_ids.
        oo_result = lo_ec2->describesecuritygroups( it_groupids = lt_group_ids ).
      " oo_result is returned for testing purposes. "
        DATA(lt_security_groups) = oo_result->get_securitygroups( ).
        MESSAGE 'Retrieved information about security groups.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat DescribeSecurityGroupsdi AWS SDK untuk referensi SAP ABAP API.

`MonitorInstances`

Contoh kode berikut menunjukkan cara menggunakan`MonitorInstances`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat MonitorInstancesdi AWS SDK untuk referensi SAP ABAP API.

`RebootInstances`

Contoh kode berikut menunjukkan cara menggunakan`RebootInstances`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat RebootInstancesdi AWS SDK untuk referensi SAP ABAP API.

`ReleaseAddress`

Contoh kode berikut menunjukkan cara menggunakan`ReleaseAddress`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat ReleaseAddressdi AWS SDK untuk referensi SAP ABAP API.

`RunInstances`

Contoh kode berikut menunjukkan cara menggunakan`RunInstances`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    " Create tags for resource created during instance launch. "
    DATA lt_tagspecifications TYPE /aws1/
cl_ec2tagspecification=>tt_tagspecificationlist.
    DATA ls_tagspecifications LIKE LINE OF lt_tagspecifications.
    ls_tagspecifications = NEW /aws1/cl_ec2tagspecification(
      iv_resourcetype = 'instance'
      it_tags = VALUE /aws1/cl_ec2tag=>tt_taglist(
        ( NEW /aws1/cl_ec2tag( iv_key = 'Name' iv_value = iv_tag_value ) )
      ) ).
    APPEND ls_tagspecifications TO lt_tagspecifications.

    TRY.
        " Create/launch Amazon Elastic Compute Cloud (Amazon EC2) instance. "
        oo_result = lo_ec2->runinstances(                           " oo_result is
 returned for testing purposes. "
          iv_imageid = iv_ami_id
          iv_instancetype = 't3.micro'
          iv_maxcount = 1
          iv_mincount = 1
          it_tagspecifications = lt_tagspecifications
          iv_subnetid = iv_subnet_id ).
        MESSAGE 'EC2 instance created.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat RunInstancesdi AWS SDK untuk referensi SAP ABAP API.

`StartInstances`

Contoh kode berikut menunjukkan cara menggunakan`StartInstances`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat StartInstancesdi AWS SDK untuk referensi SAP ABAP API.

`StopInstances`

Contoh kode berikut menunjukkan cara menggunakan`StopInstances`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat StopInstancesdi AWS SDK untuk referensi SAP ABAP API.

Contoh Amazon ECR menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Amazon ECR.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`CreateRepository`

Contoh kode berikut menunjukkan cara menggunakan`CreateRepository`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_repository_name = 'my-repository'
        oo_result = lo_ecr->createrepository(
          iv_repositoryname = iv_repository_name ).
        DATA(lv_repository_uri) = oo_result->get_repository( )-
>get_repositoryuri( ).
        MESSAGE |Repository created with URI: { lv_repository_uri }| TYPE 'I'.
      CATCH /aws1/cx_ecrrepositoryalrexex.
        " If repository already exists, retrieve it
        DATA lt_repo_names TYPE /aws1/
cl_ecrrepositorynamels00=>tt_repositorynamelist.
        APPEND NEW /aws1/cl_ecrrepositorynamels00( iv_value = iv_repository_name )
 TO lt_repo_names.
        DATA(lo_describe_result) = lo_ecr->describerepositories( it_repositorynames
 = lt_repo_names ).
        DATA(lt_repos) = lo_describe_result->get_repositories( ).
        IF lines( lt_repos ) > 0.
          READ TABLE lt_repos INDEX 1 INTO DATA(lo_repo).
          oo_result = NEW /aws1/cl_ecrcrerepositoryrsp( io_repository = lo_repo ).
          MESSAGE |Repository { iv_repository_name } already exists.| TYPE 'I'.
        ENDIF.
    ENDTRY.
```

- Untuk detail API, lihat CreateRepositorydi AWS SDK untuk referensi SAP ABAP API.

`DeleteRepository`

Contoh kode berikut menunjukkan cara menggunakan`DeleteRepository`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_repository_name = 'my-repository'
        lo_ecr->deleterepository(
          iv_repositoryname = iv_repository_name
          iv_force = abap_true ).
        MESSAGE |Repository { iv_repository_name } deleted.| TYPE 'I'.
      CATCH /aws1/cx_ecrrepositorynotfndex.
        MESSAGE 'Repository not found.' TYPE 'I'.
    ENDTRY.
```

- Untuk detail API, lihat DeleteRepositorydi AWS SDK untuk referensi SAP ABAP API.

`DescribeImages`

Contoh kode berikut menunjukkan cara menggunakan`DescribeImages`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_repository_name = 'my-repository'
        " it_image_ids = VALUE #( ( NEW /aws1/cl_ecrimageidentifier( iv_imagetag =
'latest' ) ) )
        IF it_image_ids IS NOT INITIAL.
          oo_result = lo_ecr->describeimages(
```

```
            iv_repositoryname = iv_repository_name
            it_imageids = it_image_ids ).
        ELSE.
          oo_result = lo_ecr->describeimages(
            iv_repositoryname = iv_repository_name ).
        ENDIF.
        DATA(lt_image_details) = oo_result->get_imagedetails( ).
        MESSAGE |Found { lines( lt_image_details ) } images in repository.| TYPE
 'I'.
      CATCH /aws1/cx_ecrrepositorynotfndex.
        MESSAGE 'Repository not found.' TYPE 'I'.
      CATCH /aws1/cx_ecrimagenotfoundex.
        MESSAGE 'Image not found.' TYPE 'I'.
      CATCH /aws1/cx_ecrinvalidparameterex.
        MESSAGE 'Invalid parameter provided.' TYPE 'I'.
    ENDTRY.
```

- Untuk detail API, lihat DescribeImagesdi AWS SDK untuk referensi SAP ABAP API.

`DescribeRepositories`

Contoh kode berikut menunjukkan cara menggunakan`DescribeRepositories`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " it_repository_names = VALUE #( ( NEW /aws1/
cl_ecrrepositorynamels00( iv_value = 'my-repository' ) ) )
        oo_result = lo_ecr->describerepositories(
          it_repositorynames = it_repository_names ).
        DATA(lt_repositories) = oo_result->get_repositories( ).
        MESSAGE |Found { lines( lt_repositories ) } repositories.| TYPE 'I'.
      CATCH /aws1/cx_ecrrepositorynotfndex.
        MESSAGE 'Repository not found.' TYPE 'I'.
```

```
    ENDTRY.
```

- Untuk detail API, lihat DescribeRepositoriesdi AWS SDK untuk referensi SAP ABAP API.

`GetAuthorizationToken`

Contoh kode berikut menunjukkan cara menggunakan`GetAuthorizationToken`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_ecr->getauthorizationtoken( ).
    DATA(lt_auth_data) = oo_result->get_authorizationdata( ).
    IF lines( lt_auth_data ) > 0.
      READ TABLE lt_auth_data INDEX 1 INTO DATA(lo_auth_data).
      DATA(lv_token) = lo_auth_data->get_authorizationtoken( ).
      MESSAGE 'Authorization token retrieved.' TYPE 'I'.
    ENDIF.
  CATCH /aws1/cx_ecrserverexception.
    MESSAGE 'Server exception occurred.' TYPE 'I'.
ENDTRY.
```

- Untuk detail API, lihat GetAuthorizationTokendi AWS SDK untuk referensi SAP ABAP API.

`GetRepositoryPolicy`

Contoh kode berikut menunjukkan cara menggunakan`GetRepositoryPolicy`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_repository_name = 'my-repository'
        oo_result = lo_ecr->getrepositorypolicy(
          iv_repositoryname = iv_repository_name ).
        DATA(lv_policy_text) = oo_result->get_policytext( ).
        MESSAGE 'Repository policy retrieved.' TYPE 'I'.
      CATCH /aws1/cx_ecrrepositorynotfndex.
        MESSAGE 'Repository not found.' TYPE 'I'.
      CATCH /aws1/cx_ecrrepositoryplynot00.
        MESSAGE 'Repository policy not found.' TYPE 'I'.
    ENDTRY.
```

- Untuk detail API, lihat GetRepositoryPolicydi AWS SDK untuk referensi SAP ABAP API.

`PutLifeCyclePolicy`

Contoh kode berikut menunjukkan cara menggunakan`PutLifeCyclePolicy`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_repository_name = 'my-repository'
        " iv_lifecycle_policy_text = '{"rules":
[{"rulePriority":1,"description":"Expire images older than 14 days",...}]}'
```

```
        lo_ecr->putlifecyclepolicy(
          iv_repositoryname = iv_repository_name
          iv_lifecyclepolicytext = iv_lifecycle_policy_text ).
        MESSAGE |Lifecycle policy set for repository { iv_repository_name }.| TYPE
'I'.
      CATCH /aws1/cx_ecrrepositorynotfndex.
        MESSAGE 'Repository not found.' TYPE 'I'.
      CATCH /aws1/cx_ecrvalidationex.
        MESSAGE 'Invalid lifecycle policy format.' TYPE 'I'.
    ENDTRY.
```

- Untuk detail API, lihat PutLifeCyclePolicydi AWS SDK untuk referensi SAP ABAP API.

`SetRepositoryPolicy`

Contoh kode berikut menunjukkan cara menggunakan`SetRepositoryPolicy`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_repository_name = 'my-repository'
        " iv_policy_text = '{"Version":"2012-10-17",       "Statement":[...]}'
        lo_ecr->setrepositorypolicy(
          iv_repositoryname = iv_repository_name
          iv_policytext = iv_policy_text ).
        MESSAGE |Policy set for repository { iv_repository_name }.| TYPE 'I'.
      CATCH /aws1/cx_ecrrepositorynotfndex.
        MESSAGE 'Repository not found.' TYPE 'I'.
    ENDTRY.
```

- Untuk detail API, lihat SetRepositoryPolicydi AWS SDK untuk referensi SAP ABAP API.

Contoh EMR Amazon menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Amazon EMR.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`AddJobFlowSteps`

Contoh kode berikut menunjukkan cara menggunakan`AddJobFlowSteps`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " Build args list for Spark submit
    DATA lt_args TYPE /aws1/cl_emrxmlstringlist_w=>tt_xmlstringlist.
    APPEND NEW /aws1/cl_emrxmlstringlist_w( 'spark-submit' ) TO lt_args.
    APPEND NEW /aws1/cl_emrxmlstringlist_w( '--deploy-mode' ) TO lt_args.
    APPEND NEW /aws1/cl_emrxmlstringlist_w( 'cluster' ) TO lt_args.
    APPEND NEW /aws1/cl_emrxmlstringlist_w( iv_script_uri ) TO lt_args.
    APPEND LINES OF it_script_args TO lt_args.

    " Create step configuration
```

```
      DATA(lo_hadoop_jar_step) = NEW /aws1/cl_emrhadoopjarstepcfg(
        iv_jar = 'command-runner.jar'
        it_args = lt_args
      ).

      DATA(lo_step_config) = NEW /aws1/cl_emrstepconfig(
        iv_name = iv_name
        iv_actiononfailure = 'CONTINUE'
        io_hadoopjarstep = lo_hadoop_jar_step
      ).

      DATA lt_steps TYPE /aws1/cl_emrstepconfig=>tt_stepconfiglist.
      APPEND lo_step_config TO lt_steps.

      DATA(lo_result) = lo_emr->addjobflowsteps(
        iv_jobflowid = iv_cluster_id
        it_steps = lt_steps
      ).

      " Get first step ID
      DATA(lt_step_ids) = lo_result->get_stepids( ).
      READ TABLE lt_step_ids INDEX 1 INTO DATA(lo_step_id_obj).
      IF sy-subrc = 0.
        ov_step_id = lo_step_id_obj->get_value( ).
        MESSAGE |Step added with ID { ov_step_id }| TYPE 'I'.
      ENDIF.
    CATCH /aws1/cx_emrinternalservererr INTO DATA(lo_internal_error).
      DATA(lv_error) = lo_internal_error->if_message~get_text( ).
      MESSAGE lv_error TYPE 'E'.
  ENDTRY.
```

- Untuk detail API, lihat AddJobFlowStepsdi AWS SDK untuk referensi SAP ABAP API.

`DescribeCluster`

Contoh kode berikut menunjukkan cara menggunakan`DescribeCluster`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_emr->describecluster(
      iv_clusterid = iv_cluster_id
    ).
    DATA(lo_cluster) = oo_result->get_cluster( ).
    DATA(lv_cluster_name) = lo_cluster->get_name( ).
    MESSAGE |Retrieved cluster information for { lv_cluster_name }| TYPE 'I'.
  CATCH /aws1/cx_emrinternalserverex INTO DATA(lo_internal_error).
    DATA(lv_error) = lo_internal_error->if_message~get_text( ).
    MESSAGE lv_error TYPE 'E'.
  CATCH /aws1/cx_emrinvalidrequestex INTO DATA(lo_invalid_error).
    lv_error = lo_invalid_error->if_message~get_text( ).
    MESSAGE lv_error TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DescribeClusterdi AWS SDK untuk referensi SAP ABAP API.

`DescribeStep`

Contoh kode berikut menunjukkan cara menggunakan`DescribeStep`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
```

```
    oo_result = lo_emr->describestep(
      iv_clusterid = iv_cluster_id
      iv_stepid = iv_step_id
    ).
    DATA(lo_step) = oo_result->get_step( ).
    DATA(lv_step_name) = lo_step->get_name( ).
    MESSAGE |Retrieved step information for { lv_step_name }| TYPE 'I'.
  CATCH /aws1/cx_emrinternalserverex INTO DATA(lo_internal_error).
    DATA(lv_error) = lo_internal_error->if_message~get_text( ).
    MESSAGE lv_error TYPE 'E'.
  CATCH /aws1/cx_emrinvalidrequestex INTO DATA(lo_invalid_error).
    lv_error = lo_invalid_error->if_message~get_text( ).
    MESSAGE lv_error TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DescribeStepdi AWS SDK untuk referensi SAP ABAP API.

`ListSteps`

Contoh kode berikut menunjukkan cara menggunakan`ListSteps`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_emr->liststeps(
      iv_clusterid = iv_cluster_id
    ).
    DATA(lt_steps) = oo_result->get_steps( ).
    DATA(lv_step_count) = lines( lt_steps ).
    MESSAGE |Retrieved { lv_step_count } steps for cluster| TYPE 'I'.
  CATCH /aws1/cx_emrinternalserverex INTO DATA(lo_internal_error).
    DATA(lv_error) = lo_internal_error->if_message~get_text( ).
    MESSAGE lv_error TYPE 'E'.
  CATCH /aws1/cx_emrinvalidrequestex INTO DATA(lo_invalid_error).
```

```
        lv_error = lo_invalid_error->if_message~get_text( ).
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat ListStepsdi AWS SDK untuk referensi SAP ABAP API.

`RunJobFlow`

Contoh kode berikut menunjukkan cara menggunakan`RunJobFlow`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " Create instances configuration
        DATA(lo_instances) = NEW /aws1/cl_emrjobflowinstsconfig(
          iv_masterinstancetype = 'm5.xlarge'
          iv_slaveinstancetype = 'm5.xlarge'
          iv_instancecount = 3
          iv_keepjobflowalivewhennos00 = iv_keep_alive
          iv_emrmanagedmastersecgroup = iv_primary_sec_grp
          iv_emrmanagedslavesecgroup = iv_secondary_sec_grp
        ).

        DATA(lo_result) = lo_emr->runjobflow(
          iv_name = iv_name
          iv_loguri = iv_log_uri
          iv_releaselabel = 'emr-5.30.1'
          io_instances = lo_instances
          it_steps = it_steps
          it_applications = it_applications
          iv_jobflowrole = iv_job_flow_role
          iv_servicerole = iv_service_role
          iv_ebsrootvolumesize = 10
          iv_visibletoallusers = abap_true
        ).
```

```
    ov_cluster_id = lo_result->get_jobflowid( ).
    MESSAGE 'EMR cluster created successfully.' TYPE 'I'.
  CATCH /aws1/cx_emrinternalservererr INTO DATA(lo_internal_error).
    DATA(lv_error) = lo_internal_error->if_message~get_text( ).
    MESSAGE lv_error TYPE 'E'.
  CATCH /aws1/cx_emrclientexc INTO DATA(lo_client_error).
    lv_error = lo_client_error->if_message~get_text( ).
    MESSAGE lv_error TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat RunJobFlowdi AWS SDK untuk referensi SAP ABAP API.

`TerminateJobFlows`

Contoh kode berikut menunjukkan cara menggunakan`TerminateJobFlows`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    DATA lt_cluster_ids TYPE /aws1/cl_emrxmlstringlist_w=>tt_xmlstringlist.
    APPEND NEW /aws1/cl_emrxmlstringlist_w( iv_cluster_id ) TO lt_cluster_ids.

    lo_emr->terminatejobflows(
      it_jobflowids = lt_cluster_ids
    ).
    MESSAGE 'EMR cluster terminated successfully.' TYPE 'I'.
  CATCH /aws1/cx_emrinternalservererr INTO DATA(lo_internal_error).
    DATA(lv_error) = lo_internal_error->if_message~get_text( ).
    MESSAGE lv_error TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat TerminateJobFlowsdi AWS SDK untuk referensi SAP ABAP API.

EventBridge Contoh penjadwal menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Scheduler. EventBridge

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`CreateSchedule`

Contoh kode berikut menunjukkan cara menggunakan`CreateSchedule`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " Constants for time calculations
    DATA lv_start_date TYPE /aws1/scdstartdate.
    DATA lv_end_date TYPE /aws1/scdenddate.
    DATA lv_start_timestamp TYPE timestamp.
    DATA lv_end_timestamp TYPE timestamp.
    DATA lv_hours_to_run TYPE i VALUE 1.

    " Get current timestamp
```

```
        GET TIME STAMP FIELD lv_start_timestamp.

        " Add 1 hour to the current timestamp using CL_ABAP_TSTMP
        lv_end_timestamp = cl_abap_tstmp=>add(
          tstmp = lv_start_timestamp
          secs = lv_hours_to_run * 3600 ).

        " Convert timestamps to decimal format for AWS API
        lv_start_date = lv_start_timestamp.
        lv_end_date = lv_end_timestamp.

        " Prepare flexible time window configuration
        DATA lo_flexible_time_window TYPE REF TO /aws1/cl_scdflexibletimewindow.
        IF iv_use_flexible_time_win = abap_true.
          " iv_use_flexible_time_win = ABAP_TRUE
          " Example: Set MaximumWindowInMinutes to 10 for flexible window
          lo_flexible_time_window = NEW /aws1/cl_scdflexibletimewindow(
            iv_mode = 'FLEXIBLE'
            iv_maximumwindowinminutes = 10 ).
        ELSE.
          lo_flexible_time_window = NEW /aws1/cl_scdflexibletimewindow(
            iv_mode = 'OFF' ).
        ENDIF.

        " Prepare target configuration
        " Example iv_target_arn = 'arn:aws:sqs:us-east-1:123456789012:my-queue'
        " Example iv_role_arn = 'arn:aws:iam::123456789012:role/SchedulerRole'
        " Example iv_input = '{"message": "Hello from EventBridge Scheduler"}'
        DATA(lo_target) = NEW /aws1/cl_scdtarget(
          iv_arn = iv_target_arn
          iv_rolearn = iv_role_arn
          iv_input = iv_input ).

        " Set action after completion if needed
        DATA lv_action_after_completion TYPE /aws1/scdactionaftercompletion.
        IF iv_delete_after_completion = abap_true.
          " iv_delete_after_completion = ABAP_TRUE
          lv_action_after_completion = 'DELETE'.
        ELSE.
          lv_action_after_completion = 'NONE'.
        ENDIF.

        " Create the schedule
        " Example iv_name = 'my-schedule'
```

```
        " Example iv_schedule_expression = 'rate(15 minutes)'
        " Example iv_schedule_group_name = 'my-schedule-group'
        DATA(lo_result) = lo_scd->createschedule(
          iv_name = iv_name
          iv_scheduleexpression = iv_schedule_expression
          iv_groupname = iv_schedule_group_name
          io_target = lo_target
          io_flexibletimewindow = lo_flexible_time_window
          iv_startdate = lv_start_date
          iv_enddate = lv_end_date
          iv_actionaftercompletion = lv_action_after_completion ).

        ov_schedule_arn = lo_result->get_schedulearn( ).
        MESSAGE 'Schedule created successfully.' TYPE 'I'.

      CATCH /aws1/cx_scdconflictexception INTO DATA(lo_conflict_ex).
        DATA(lv_error) = |Conflict creating schedule: { lo_conflict_ex-
>if_message~get_text( ) }|.
        MESSAGE lv_error TYPE 'I'.
      CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_ex).
        DATA(lv_generic_error) = |Error creating schedule: { lo_generic_ex-
>if_message~get_text( ) }|.
        MESSAGE lv_generic_error TYPE 'I'.
    ENDTRY.
```

- Untuk detail API, lihat CreateScheduledi AWS SDK untuk referensi SAP ABAP API.

`CreateScheduleGroup`

Contoh kode berikut menunjukkan cara menggunakan`CreateScheduleGroup`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " Example iv_name = 'my-schedule-group'
```

```
        DATA(lo_result) = lo_scd->createschedulegroup(
          iv_name = iv_name ).

        ov_schedule_group_arn = lo_result->get_schedulegrouparn( ).
        MESSAGE 'Schedule group created successfully.' TYPE 'I'.

      CATCH /aws1/cx_scdconflictexception INTO DATA(lo_conflict_ex).
        DATA(lv_error) = |Conflict creating schedule group: { lo_conflict_ex-
>if_message~get_text( ) }|.
        MESSAGE lv_error TYPE 'I'.
      CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_ex).
        DATA(lv_generic_error) = |Error creating schedule group: { lo_generic_ex-
>if_message~get_text( ) }|.
        MESSAGE lv_generic_error TYPE 'I'.
    ENDTRY.
```

- Untuk detail API, lihat CreateScheduleGroupdi AWS SDK untuk referensi SAP ABAP API.

`DeleteSchedule`

Contoh kode berikut menunjukkan cara menggunakan`DeleteSchedule`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " Example iv_name = 'my-schedule'
        " Example iv_schedule_group_name = 'my-schedule-group'
        lo_scd->deleteschedule(
          iv_name = iv_name
          iv_groupname = iv_schedule_group_name ).
        MESSAGE 'Schedule deleted successfully.' TYPE 'I'.

      CATCH /aws1/cx_scdresourcenotfoundex INTO DATA(lo_not_found_ex).
        DATA(lv_error) = |Schedule not found: { lo_not_found_ex-
>if_message~get_text( ) }|.
```

```
        MESSAGE lv_error TYPE 'I'.
      CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_ex).
        DATA(lv_generic_error) = |Error deleting schedule: { lo_generic_ex-
>if_message~get_text( ) }|.
        MESSAGE lv_generic_error TYPE 'I'.
    ENDTRY.
```

- Untuk detail API, lihat DeleteScheduledi AWS SDK untuk referensi SAP ABAP API.

`DeleteScheduleGroup`

Contoh kode berikut menunjukkan cara menggunakan`DeleteScheduleGroup`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " Example iv_name = 'my-schedule-group'
        lo_scd->deleteschedulegroup(
          iv_name = iv_name ).
        MESSAGE 'Schedule group deleted successfully.' TYPE 'I'.

      CATCH /aws1/cx_scdresourcenotfoundex INTO DATA(lo_not_found_ex).
        DATA(lv_error) = |Schedule group not found: { lo_not_found_ex-
>if_message~get_text( ) }|.
        MESSAGE lv_error TYPE 'I'.
      CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_ex).
        DATA(lv_generic_error) = |Error deleting schedule group: { lo_generic_ex-
>if_message~get_text( ) }|.
        MESSAGE lv_generic_error TYPE 'I'.
    ENDTRY.
```

- Untuk detail API, lihat DeleteScheduleGroupdi AWS SDK untuk referensi SAP ABAP API.

AWS Glue contoh menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan mengimplementasikan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan. AWS Glue

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`CreateCrawler`

Contoh kode berikut menunjukkan cara menggunakan`CreateCrawler`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " iv_crawler_name = 'my-crawler'
    " iv_role_arn = 'arn:aws:iam::123456789012:role/AWSGlueServiceRole-Test'
    " iv_database_name = 'my-database'
    " iv_table_prefix = 'test_'
    " iv_s3_target = 's3://example-bucket/data/'

    DATA(lt_s3_targets) = VALUE /aws1/cl_glus3target=>tt_s3targetlist(
      ( NEW /aws1/cl_glus3target( iv_path = iv_s3_target ) ) ).

    DATA(lo_targets) = NEW /aws1/cl_glucrawlertargets(
```

```
          it_s3targets = lt_s3_targets ).

        lo_glu->createcrawler(
          iv_name = iv_crawler_name
          iv_role = iv_role_arn
          iv_databasename = iv_database_name
          iv_tableprefix = iv_table_prefix
          io_targets = lo_targets ).
        MESSAGE 'Crawler created successfully.' TYPE 'I'.
      CATCH /aws1/cx_glualreadyexistsex.
        MESSAGE 'Crawler already exists.' TYPE 'E'.
      CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
        DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
        MESSAGE lv_invalid_error TYPE 'E'.
      CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
        DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
        MESSAGE lv_timeout_error TYPE 'E'.
      CATCH /aws1/cx_gluresrcnumlmtexcdex INTO DATA(lo_limit_ex).
        DATA(lv_limit_error) = lo_limit_ex->if_message~get_longtext( ).
        MESSAGE lv_limit_error TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat CreateCrawlerdi AWS SDK untuk referensi SAP ABAP API.

`CreateJob`

Contoh kode berikut menunjukkan cara menggunakan`CreateJob`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_job_name = 'my-etl-job'
        " iv_description = 'ETL job for data transformation'
        " iv_role_arn = 'arn:aws:iam::123456789012:role/AWSGlueServiceRole-Test'
        " iv_script_location = 's3://example-bucket/scripts/my-script.py'
```

```
      DATA(lo_command) = NEW /aws1/cl_glujobcommand(
        iv_name = 'glueetl'
        iv_scriptlocation = iv_script_location
        iv_pythonversion = '3' ).

      lo_glu->createjob(
        iv_name = iv_job_name
        iv_description = iv_description
        iv_role = iv_role_arn
        io_command = lo_command
        iv_glueversion = '3.0' ).
      MESSAGE 'Job created successfully.' TYPE 'I'.
    CATCH /aws1/cx_glualreadyexistsex.
      MESSAGE 'Job already exists.' TYPE 'E'.
    CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
      DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
      MESSAGE lv_invalid_error TYPE 'E'.
    CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
      DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
      MESSAGE lv_internal_error TYPE 'E'.
    CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
      DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
      MESSAGE lv_timeout_error TYPE 'E'.
    CATCH /aws1/cx_gluresrcnumlmtexcdex INTO DATA(lo_limit_ex).
      DATA(lv_limit_error) = lo_limit_ex->if_message~get_longtext( ).
      MESSAGE lv_limit_error TYPE 'E'.
  ENDTRY.
```

- Untuk detail API, lihat CreateJobdi AWS SDK untuk referensi SAP ABAP API.

`DeleteCrawler`

Contoh kode berikut menunjukkan cara menggunakan`DeleteCrawler`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " iv_crawler_name = 'my-crawler'
    lo_glu->deletecrawler( iv_name = iv_crawler_name ).
    MESSAGE 'Crawler deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_glucrawlerrunningex.
    MESSAGE 'Crawler is currently running.' TYPE 'E'.
  CATCH /aws1/cx_gluentitynotfoundex.
    MESSAGE 'Crawler does not exist.' TYPE 'E'.
  CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
    DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
    MESSAGE lv_timeout_error TYPE 'E'.
  CATCH /aws1/cx_gluschdrtransingex.
    MESSAGE 'Scheduler is transitioning.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DeleteCrawlerdi AWS SDK untuk referensi SAP ABAP API.

`DeleteDatabase`

Contoh kode berikut menunjukkan cara menggunakan`DeleteDatabase`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " iv_database_name = 'my-database'
    lo_glu->deletedatabase( iv_name = iv_database_name ).
    MESSAGE 'Database deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_gluentitynotfoundex.
    MESSAGE 'Database does not exist.' TYPE 'E'.
  CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
    DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
    MESSAGE lv_invalid_error TYPE 'E'.
  CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
    DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
```

```
      MESSAGE lv_internal_error TYPE 'E'.
    CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
      DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
      MESSAGE lv_timeout_error TYPE 'E'.
  ENDTRY.
```

- Untuk detail API, lihat DeleteDatabasedi AWS SDK untuk referensi SAP ABAP API.

`DeleteJob`

Contoh kode berikut menunjukkan cara menggunakan`DeleteJob`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
  TRY.
      " iv_job_name = 'my-etl-job'
      lo_glu->deletejob( iv_jobname = iv_job_name ).
      MESSAGE 'Job deleted successfully.' TYPE 'I'.
    CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
      DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
      MESSAGE lv_invalid_error TYPE 'E'.
    CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
      DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
      MESSAGE lv_internal_error TYPE 'E'.
    CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
      DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
      MESSAGE lv_timeout_error TYPE 'E'.
  ENDTRY.
```

- Untuk detail API, lihat DeleteJobdi AWS SDK untuk referensi SAP ABAP API.

`DeleteTable`

Contoh kode berikut menunjukkan cara menggunakan`DeleteTable`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " iv_database_name = 'my-database'
    " iv_table_name = 'my-table'
    lo_glu->deletetable(
      iv_databasename = iv_database_name
      iv_name = iv_table_name ).
    MESSAGE 'Table deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_gluentitynotfoundex.
    MESSAGE 'Table or database does not exist.' TYPE 'E'.
  CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
    DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
    MESSAGE lv_invalid_error TYPE 'E'.
  CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
    DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
    MESSAGE lv_internal_error TYPE 'E'.
  CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
    DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
    MESSAGE lv_timeout_error TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DeleteTabledi AWS SDK untuk referensi SAP ABAP API.

`GetCrawler`

Contoh kode berikut menunjukkan cara menggunakan`GetCrawler`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " iv_crawler_name = 'my-crawler'
    oo_result = lo_glu->getcrawler( iv_name = iv_crawler_name ).
    DATA(lo_crawler) = oo_result->get_crawler( ).
    MESSAGE 'Crawler information retrieved.' TYPE 'I'.
  CATCH /aws1/cx_gluentitynotfoundex.
    MESSAGE 'Crawler does not exist.' TYPE 'I'.
  CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
    DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
    MESSAGE lv_timeout_error TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat GetCrawlerdi AWS SDK untuk referensi SAP ABAP API.

`GetDatabase`

Contoh kode berikut menunjukkan cara menggunakan`GetDatabase`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " iv_database_name = 'my-database'
    oo_result = lo_glu->getdatabase( iv_name = iv_database_name ).
    DATA(lo_database) = oo_result->get_database( ).
```

```
      MESSAGE 'Database information retrieved.' TYPE 'I'.
    CATCH /aws1/cx_gluentitynotfoundex.
      MESSAGE 'Database does not exist.' TYPE 'E'.
    CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
      DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
      MESSAGE lv_invalid_error TYPE 'E'.
    CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
      DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
      MESSAGE lv_internal_error TYPE 'E'.
    CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
      DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
      MESSAGE lv_timeout_error TYPE 'E'.
  ENDTRY.
```

- Untuk detail API, lihat GetDatabasedi AWS SDK untuk referensi SAP ABAP API.

`GetJobRun`

Contoh kode berikut menunjukkan cara menggunakan`GetJobRun`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
  TRY.
      " iv_job_name = 'my-etl-job'
      " iv_run_id = 'jr_abcd1234567890abcdef1234567890abcdef12345678'
      oo_result = lo_glu->getjobrun(
        iv_jobname = iv_job_name
        iv_runid = iv_run_id ).
      DATA(lo_job_run) = oo_result->get_jobrun( ).
      MESSAGE 'Job run information retrieved.' TYPE 'I'.
    CATCH /aws1/cx_gluentitynotfoundex.
      MESSAGE 'Job or job run does not exist.' TYPE 'E'.
    CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
      DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
```

```
      MESSAGE lv_invalid_error TYPE 'E'.
    CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
      DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
      MESSAGE lv_internal_error TYPE 'E'.
    CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
      DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
      MESSAGE lv_timeout_error TYPE 'E'.
  ENDTRY.
```

- Untuk detail API, lihat GetJobRundi AWS SDK untuk referensi SAP ABAP API.

`GetJobRuns`

Contoh kode berikut menunjukkan cara menggunakan`GetJobRuns`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
  TRY.
      " iv_job_name = 'my-etl-job'
      oo_result = lo_glu->getjobruns( iv_jobname = iv_job_name ).
      DATA(lt_job_runs) = oo_result->get_jobruns( ).
      MESSAGE 'Job runs retrieved successfully.' TYPE 'I'.
    CATCH /aws1/cx_gluentitynotfoundex.
      MESSAGE 'Job does not exist.' TYPE 'E'.
    CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
      DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
      MESSAGE lv_invalid_error TYPE 'E'.
    CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
      DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
      MESSAGE lv_internal_error TYPE 'E'.
    CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
      DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
      MESSAGE lv_timeout_error TYPE 'E'.
  ENDTRY.
```

- Untuk detail API, lihat GetJobRunsdi AWS SDK untuk referensi SAP ABAP API.

`GetTables`

Contoh kode berikut menunjukkan cara menggunakan`GetTables`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " iv_database_name = 'my-database'
    oo_result = lo_glu->gettables( iv_databasename = iv_database_name ).
    DATA(lt_tables) = oo_result->get_tablelist( ).
    MESSAGE 'Tables retrieved successfully.' TYPE 'I'.
  CATCH /aws1/cx_gluentitynotfoundex.
    MESSAGE 'Database does not exist.' TYPE 'E'.
  CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
    DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
    MESSAGE lv_invalid_error TYPE 'E'.
  CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
    DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
    MESSAGE lv_internal_error TYPE 'E'.
  CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
    DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
    MESSAGE lv_timeout_error TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat GetTablesdi AWS SDK untuk referensi SAP ABAP API.

`ListJobs`

Contoh kode berikut menunjukkan cara menggunakan`ListJobs`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_glu->listjobs( ).
    DATA(lt_job_names) = oo_result->get_jobnames( ).
    MESSAGE 'Job list retrieved successfully.' TYPE 'I'.
  CATCH /aws1/cx_gluentitynotfoundex.
    MESSAGE 'No jobs found.' TYPE 'I'.
  CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
    DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
    MESSAGE lv_invalid_error TYPE 'E'.
  CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
    DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
    MESSAGE lv_internal_error TYPE 'E'.
  CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
    DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
    MESSAGE lv_timeout_error TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat ListJobsdi AWS SDK untuk referensi SAP ABAP API.

`StartCrawler`

Contoh kode berikut menunjukkan cara menggunakan`StartCrawler`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " iv_crawler_name = 'my-crawler'
    lo_glu->startcrawler( iv_name = iv_crawler_name ).
    MESSAGE 'Crawler started successfully.' TYPE 'I'.
  CATCH /aws1/cx_glucrawlerrunningex.
    MESSAGE 'Crawler is already running.' TYPE 'I'.
  CATCH /aws1/cx_gluentitynotfoundex.
    MESSAGE 'Crawler does not exist.' TYPE 'E'.
  CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
    DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
    MESSAGE lv_timeout_error TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat StartCrawlerdi AWS SDK untuk referensi SAP ABAP API.

`StartJobRun`

Contoh kode berikut menunjukkan cara menggunakan`StartJobRun`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " iv_job_name = 'my-etl-job'
    " iv_input_database = 'my-database'
    " iv_input_table = 'my-table'
    " iv_output_bucket_url = 's3://example-output-bucket/'

    DATA lt_arguments TYPE /aws1/cl_glugenericmap_w=>tt_genericmap.
    lt_arguments = VALUE #(
      ( VALUE /aws1/cl_glugenericmap_w=>ts_genericmap_maprow(
        key = '--input_database'
        value = NEW /aws1/cl_glugenericmap_w( iv_value = iv_input_database ) ) )
      ( VALUE /aws1/cl_glugenericmap_w=>ts_genericmap_maprow(
        key = '--input_table'
```

```
            value = NEW /aws1/cl_glugenericmap_w( iv_value = iv_input_table ) ) )
          ( VALUE /aws1/cl_glugenericmap_w=>ts_genericmap_maprow(
            key = '--output_bucket_url'
            value = NEW /aws1/cl_glugenericmap_w( iv_value =
iv_output_bucket_url ) ) ) ).

        DATA(oo_result) = lo_glu->startjobrun(
          iv_jobname = iv_job_name
          it_arguments = lt_arguments ).
        ov_job_run_id = oo_result->get_jobrunid( ).
        MESSAGE 'Job run started successfully.' TYPE 'I'.
      CATCH /aws1/cx_gluconcurrentrunsex00.
        MESSAGE 'Maximum concurrent runs exceeded.' TYPE 'E'.
      CATCH /aws1/cx_gluentitynotfoundex.
        MESSAGE 'Job does not exist.' TYPE 'E'.
      CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
        DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
        MESSAGE lv_invalid_error TYPE 'E'.
      CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
        DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
        MESSAGE lv_internal_error TYPE 'E'.
      CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
        DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
        MESSAGE lv_timeout_error TYPE 'E'.
      CATCH /aws1/cx_gluresrcnumlmtexcdex INTO DATA(lo_limit_ex).
        DATA(lv_limit_error) = lo_limit_ex->if_message~get_longtext( ).
        MESSAGE lv_limit_error TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat StartJobRundi AWS SDK untuk referensi SAP ABAP API.

HealthImaging contoh menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan mengimplementasikan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan. HealthImaging

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`CopyImageSet`

Contoh kode berikut menunjukkan cara menggunakan`CopyImageSet`.

SDK for SAP ABAP

```
    TRY.
        " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
        " iv_source_image_set_id = '12345678901234567890123456789012345678901234567890'
        " iv_source_version_id = '1'
        " iv_destination_image_set_id =
'12345678901234567890123456789012345678901234567890' (optional)
        " iv_destination_version_id = '1' (optional)
        " iv_force = abap_false
        DATA(lo_source_info) = NEW /aws1/cl_migcpsrcimagesetinf00(
          iv_latestversionid = iv_source_version_id ).
        DATA(lo_copy_info) = NEW /aws1/cl_migcpimagesetinfmtion(
          io_sourceimageset = lo_source_info ).
        IF iv_destination_image_set_id IS NOT INITIAL AND
           iv_destination_version_id IS NOT INITIAL.
          DATA(lo_dest_info) = NEW /aws1/cl_migcopydstimageset(
            iv_imagesetid = iv_destination_image_set_id
            iv_latestversionid = iv_destination_version_id ).
          lo_copy_info = NEW /aws1/cl_migcpimagesetinfmtion(
            io_sourceimageset = lo_source_info
            io_destinationimageset = lo_dest_info ).
        ENDIF.
        oo_result = lo_mig->copyimageset(
          iv_datastoreid = iv_datastore_id
          iv_sourceimagesetid = iv_source_image_set_id
          io_copyimagesetinformation = lo_copy_info
          iv_force = iv_force ).
        DATA(lo_dest_props) = oo_result->get_dstimagesetproperties( ).
        DATA(lv_new_id) = lo_dest_props->get_imagesetid( ).
```

```
      MESSAGE |Image set copied with new ID: { lv_new_id }.| TYPE 'I'.
    CATCH /aws1/cx_migaccessdeniedex.
      MESSAGE 'Access denied.' TYPE 'I'.
    CATCH /aws1/cx_migconflictexception.
      MESSAGE 'Conflict error.' TYPE 'I'.
    CATCH /aws1/cx_miginternalserverex.
      MESSAGE 'Internal server error.' TYPE 'I'.
    CATCH /aws1/cx_migresourcenotfoundex.
      MESSAGE 'Image set not found.' TYPE 'I'.
    CATCH /aws1/cx_migservicequotaexcdex.
      MESSAGE 'Service quota exceeded.' TYPE 'I'.
    CATCH /aws1/cx_migthrottlingex.
      MESSAGE 'Request throttled.' TYPE 'I'.
    CATCH /aws1/cx_migvalidationex.
      MESSAGE 'Validation error.' TYPE 'I'.
  ENDTRY.
```

- Untuk detail API, lihat CopyImageSetdi AWS SDK untuk referensi SAP ABAP API.

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

`CreateDatastore`

Contoh kode berikut menunjukkan cara menggunakan`CreateDatastore`.

SDK for SAP ABAP

```
  TRY.
      " iv_datastore_name = 'my-datastore-name'
      oo_result = lo_mig->createdatastore( iv_datastorename = iv_datastore_name ).
      DATA(lv_datastore_id) = oo_result->get_datastoreid( ).
      MESSAGE 'Data store created.' TYPE 'I'.
    CATCH /aws1/cx_migaccessdeniedex.
      MESSAGE 'Access denied.' TYPE 'I'.
    CATCH /aws1/cx_migconflictexception.
      MESSAGE 'Conflict. Data store may already exist.' TYPE 'I'.
    CATCH /aws1/cx_miginternalserverex.
      MESSAGE 'Internal server error.' TYPE 'I'.
```

```
  CATCH /aws1/cx_migservicequotaexcdex.
    MESSAGE 'Service quota exceeded.' TYPE 'I'.
  CATCH /aws1/cx_migthrottlingex.
    MESSAGE 'Request throttled.' TYPE 'I'.
  CATCH /aws1/cx_migvalidationex.
    MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- Untuk detail API, lihat CreateDatastoredi AWS SDK untuk referensi SAP ABAP API.

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

`DeleteDatastore`

Contoh kode berikut menunjukkan cara menggunakan`DeleteDatastore`.

SDK for SAP ABAP

```
TRY.
    " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
    oo_result = lo_mig->deletedatastore( iv_datastoreid = iv_datastore_id ).
    MESSAGE 'Data store deleted.' TYPE 'I'.
  CATCH /aws1/cx_migaccessdeniedex.
    MESSAGE 'Access denied.' TYPE 'I'.
  CATCH /aws1/cx_migconflictexception.
    MESSAGE 'Conflict. Data store may contain resources.' TYPE 'I'.
  CATCH /aws1/cx_miginternalserverex.
    MESSAGE 'Internal server error.' TYPE 'I'.
  CATCH /aws1/cx_migresourcenotfoundex.
    MESSAGE 'Data store not found.' TYPE 'I'.
  CATCH /aws1/cx_migthrottlingex.
    MESSAGE 'Request throttled.' TYPE 'I'.
  CATCH /aws1/cx_migvalidationex.
    MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- Untuk detail API, lihat DeleteDatastoredi AWS SDK untuk referensi SAP ABAP API.

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

`DeleteImageSet`

Contoh kode berikut menunjukkan cara menggunakan`DeleteImageSet`.

SDK for SAP ABAP

```
TRY.
    " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
    " iv_image_set_id = '12345678901234567890123456789012345678901234567890'
    oo_result = lo_mig->deleteimageset(
      iv_datastoreid = iv_datastore_id
      iv_imagesetid = iv_image_set_id ).
    MESSAGE 'Image set deleted.' TYPE 'I'.
  CATCH /aws1/cx_migaccessdeniedex.
    MESSAGE 'Access denied.' TYPE 'I'.
  CATCH /aws1/cx_migconflictexception.
    MESSAGE 'Conflict error.' TYPE 'I'.
  CATCH /aws1/cx_miginternalserverex.
    MESSAGE 'Internal server error.' TYPE 'I'.
  CATCH /aws1/cx_migresourcenotfoundex.
    MESSAGE 'Image set not found.' TYPE 'I'.
  CATCH /aws1/cx_migthrottlingex.
    MESSAGE 'Request throttled.' TYPE 'I'.
  CATCH /aws1/cx_migvalidationex.
    MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- Untuk detail API, lihat DeleteImageSetdi AWS SDK untuk referensi SAP ABAP API.

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

`GetDICOMImportJob`

Contoh kode berikut menunjukkan cara menggunakan`GetDICOMImportJob`.

SDK for SAP ABAP

```
TRY.
    " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
    " iv_job_id = '12345678901234567890123456789012'
    oo_result = lo_mig->getdicomimportjob(
      iv_datastoreid = iv_datastore_id
      iv_jobid = iv_job_id ).
    DATA(lo_job_props) = oo_result->get_jobproperties( ).
    DATA(lv_job_status) = lo_job_props->get_jobstatus( ).
    MESSAGE |Job status: { lv_job_status }.| TYPE 'I'.
  CATCH /aws1/cx_migaccessdeniedex.
    MESSAGE 'Access denied.' TYPE 'I'.
  CATCH /aws1/cx_migconflictexception.
    MESSAGE 'Conflict error.' TYPE 'I'.
  CATCH /aws1/cx_miginternalserverex.
    MESSAGE 'Internal server error.' TYPE 'I'.
  CATCH /aws1/cx_migresourcenotfoundex.
    MESSAGE 'Job not found.' TYPE 'I'.
  CATCH /aws1/cx_migthrottlingex.
    MESSAGE 'Request throttled.' TYPE 'I'.
  CATCH /aws1/cx_migvalidationex.
    MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- Untuk detail API, lihat GetDICOMImportJobdi AWS SDK untuk referensi SAP ABAP API.

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

`GetDatastore`

Contoh kode berikut menunjukkan cara menggunakan`GetDatastore`.

SDK for SAP ABAP

```
TRY.
    " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
    oo_result = lo_mig->getdatastore( iv_datastoreid = iv_datastore_id ).
    DATA(lo_properties) = oo_result->get_datastoreproperties( ).
    DATA(lv_name) = lo_properties->get_datastorename( ).
    DATA(lv_status) = lo_properties->get_datastorestatus( ).
    MESSAGE 'Data store properties retrieved.' TYPE 'I'.
  CATCH /aws1/cx_migaccessdeniedex.
    MESSAGE 'Access denied.' TYPE 'I'.
  CATCH /aws1/cx_miginternalserverex.
    MESSAGE 'Internal server error.' TYPE 'I'.
  CATCH /aws1/cx_migresourcenotfoundex.
    MESSAGE 'Data store not found.' TYPE 'I'.
  CATCH /aws1/cx_migthrottlingex.
    MESSAGE 'Request throttled.' TYPE 'I'.
  CATCH /aws1/cx_migvalidationex.
    MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- Untuk detail API, lihat GetDatastoredi AWS SDK untuk referensi SAP ABAP API.

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

`GetImageFrame`

Contoh kode berikut menunjukkan cara menggunakan`GetImageFrame`.

SDK for SAP ABAP

```
TRY.
    " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
    " iv_image_set_id = '12345678901234567890123456789012345678901234567890'
    " iv_image_frame_id = '12345678901234567890123456789012345678901234567890'
    oo_result = lo_mig->getimageframe(
      iv_datastoreid = iv_datastore_id
      iv_imagesetid = iv_image_set_id
      io_imageframeinformation = NEW /aws1/cl_migimageframeinfmtion(
        iv_imageframeid = iv_image_frame_id ) ).
    DATA(lv_frame_blob) = oo_result->get_imageframeblob( ).
    MESSAGE 'Image frame retrieved.' TYPE 'I'.
  CATCH /aws1/cx_migaccessdeniedex.
    MESSAGE 'Access denied.' TYPE 'I'.
  CATCH /aws1/cx_migconflictexception.
    MESSAGE 'Conflict error.' TYPE 'I'.
  CATCH /aws1/cx_miginternalserverex.
    MESSAGE 'Internal server error.' TYPE 'I'.
  CATCH /aws1/cx_migresourcenotfoundex.
    MESSAGE 'Image frame not found.' TYPE 'I'.
  CATCH /aws1/cx_migthrottlingex.
    MESSAGE 'Request throttled.' TYPE 'I'.
  CATCH /aws1/cx_migvalidationex.
    MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- Untuk detail API, lihat GetImageFramedi AWS SDK untuk referensi SAP ABAP API.

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

`GetImageSet`

Contoh kode berikut menunjukkan cara menggunakan`GetImageSet`.

SDK for SAP ABAP

```
    TRY.
        " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
        " iv_image_set_id = '12345678901234567890123456789012345678901234567890'
        " iv_version_id = '1' (optional)
        IF iv_version_id IS NOT INITIAL.
          oo_result = lo_mig->getimageset(
            iv_datastoreid = iv_datastore_id
            iv_imagesetid = iv_image_set_id
            iv_versionid = iv_version_id ).
        ELSE.
          oo_result = lo_mig->getimageset(
            iv_datastoreid = iv_datastore_id
            iv_imagesetid = iv_image_set_id ).
        ENDIF.
        DATA(lv_state) = oo_result->get_imagesetstate( ).
        MESSAGE |Image set retrieved with state: { lv_state }.| TYPE 'I'.
      CATCH /aws1/cx_migaccessdeniedex.
        MESSAGE 'Access denied.' TYPE 'I'.
      CATCH /aws1/cx_migconflictexception.
        MESSAGE 'Conflict error.' TYPE 'I'.
      CATCH /aws1/cx_miginternalserverex.
        MESSAGE 'Internal server error.' TYPE 'I'.
      CATCH /aws1/cx_migresourcenotfoundex.
        MESSAGE 'Image set not found.' TYPE 'I'.
      CATCH /aws1/cx_migthrottlingex.
        MESSAGE 'Request throttled.' TYPE 'I'.
      CATCH /aws1/cx_migvalidationex.
        MESSAGE 'Validation error.' TYPE 'I'.
    ENDTRY.
```

- Untuk detail API, lihat GetImageSetdi AWS SDK untuk referensi SAP ABAP API.

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

`GetImageSetMetadata`

Contoh kode berikut menunjukkan cara menggunakan`GetImageSetMetadata`.

SDK for SAP ABAP

```
    TRY.
        " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
        " iv_image_set_id = '12345678901234567890123456789012345678901234567890'
        " iv_version_id = '1' (optional)
        IF iv_version_id IS NOT INITIAL.
          oo_result = lo_mig->getimagesetmetadata(
            iv_datastoreid = iv_datastore_id
            iv_imagesetid = iv_image_set_id
            iv_versionid = iv_version_id ).
        ELSE.
          oo_result = lo_mig->getimagesetmetadata(
            iv_datastoreid = iv_datastore_id
            iv_imagesetid = iv_image_set_id ).
        ENDIF.
        DATA(lv_metadata_blob) = oo_result->get_imagesetmetadatablob( ).
        MESSAGE 'Image set metadata retrieved.' TYPE 'I'.
      CATCH /aws1/cx_migaccessdeniedex.
        MESSAGE 'Access denied.' TYPE 'I'.
      CATCH /aws1/cx_migconflictexception.
        MESSAGE 'Conflict error.' TYPE 'I'.
      CATCH /aws1/cx_miginternalserverex.
        MESSAGE 'Internal server error.' TYPE 'I'.
      CATCH /aws1/cx_migresourcenotfoundex.
        MESSAGE 'Image set not found.' TYPE 'I'.
      CATCH /aws1/cx_migthrottlingex.
        MESSAGE 'Request throttled.' TYPE 'I'.
      CATCH /aws1/cx_migvalidationex.
        MESSAGE 'Validation error.' TYPE 'I'.
    ENDTRY.
```

- Untuk detail API, lihat GetImageSetMetadatadi AWS SDK untuk referensi SAP ABAP API.

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

`ListDICOMImportJobs`

Contoh kode berikut menunjukkan cara menggunakan`ListDICOMImportJobs`.

SDK for SAP ABAP

```
TRY.
    " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
    oo_result = lo_mig->listdicomimportjobs( iv_datastoreid = iv_datastore_id ).
    DATA(lt_jobs) = oo_result->get_jobsummaries( ).
    DATA(lv_count) = lines( lt_jobs ).
    MESSAGE |Found { lv_count } DICOM import jobs.| TYPE 'I'.
  CATCH /aws1/cx_migaccessdeniedex.
    MESSAGE 'Access denied.' TYPE 'I'.
  CATCH /aws1/cx_migconflictexception.
    MESSAGE 'Conflict error.' TYPE 'I'.
  CATCH /aws1/cx_miginternalserverex.
    MESSAGE 'Internal server error.' TYPE 'I'.
  CATCH /aws1/cx_migresourcenotfoundex.
    MESSAGE 'Resource not found.' TYPE 'I'.
  CATCH /aws1/cx_migthrottlingex.
    MESSAGE 'Request throttled.' TYPE 'I'.
  CATCH /aws1/cx_migvalidationex.
    MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- Untuk detail API, lihat ListDICOMImportJobsdi AWS SDK untuk referensi SAP ABAP API.

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

`ListDatastores`

Contoh kode berikut menunjukkan cara menggunakan`ListDatastores`.

SDK for SAP ABAP

```
TRY.
    oo_result = lo_mig->listdatastores( ).
    DATA(lt_datastores) = oo_result->get_datastoresummaries( ).
    DATA(lv_count) = lines( lt_datastores ).
    MESSAGE |Found { lv_count } data stores.| TYPE 'I'.
  CATCH /aws1/cx_migaccessdeniedex.
    MESSAGE 'Access denied.' TYPE 'I'.
  CATCH /aws1/cx_miginternalserverex.
    MESSAGE 'Internal server error.' TYPE 'I'.
  CATCH /aws1/cx_migthrottlingex.
    MESSAGE 'Request throttled.' TYPE 'I'.
  CATCH /aws1/cx_migvalidationex.
    MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- Untuk detail API, lihat ListDatastoresdi AWS SDK untuk referensi SAP ABAP API.

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

`ListImageSetVersions`

Contoh kode berikut menunjukkan cara menggunakan`ListImageSetVersions`.

SDK for SAP ABAP

```
TRY.
    " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
    " iv_image_set_id = '12345678901234567890123456789012345678901234567890'
    oo_result = lo_mig->listimagesetversions(
      iv_datastoreid = iv_datastore_id
      iv_imagesetid = iv_image_set_id ).
```

```
        DATA(lt_versions) = oo_result->get_imagesetpropertieslist( ).
        DATA(lv_count) = lines( lt_versions ).
        MESSAGE |Found { lv_count } image set versions.| TYPE 'I'.
      CATCH /aws1/cx_migaccessdeniedex.
        MESSAGE 'Access denied.' TYPE 'I'.
      CATCH /aws1/cx_migconflictexception.
        MESSAGE 'Conflict error.' TYPE 'I'.
      CATCH /aws1/cx_miginternalserverex.
        MESSAGE 'Internal server error.' TYPE 'I'.
      CATCH /aws1/cx_migresourcenotfoundex.
        MESSAGE 'Image set not found.' TYPE 'I'.
      CATCH /aws1/cx_migthrottlingex.
        MESSAGE 'Request throttled.' TYPE 'I'.
      CATCH /aws1/cx_migvalidationex.
        MESSAGE 'Validation error.' TYPE 'I'.
    ENDTRY.
```

- Untuk detail API, lihat ListImageSetVersionsdi AWS SDK untuk referensi SAP ABAP API.

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

`ListTagsForResource`

Contoh kode berikut menunjukkan cara menggunakan`ListTagsForResource`.

SDK for SAP ABAP

```
    TRY.
        " iv_resource_arn = 'arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012'
        oo_result = lo_mig->listtagsforresource( iv_resourcearn = iv_resource_arn ).
        DATA(lt_tags) = oo_result->get_tags( ).
        DATA(lv_count) = lines( lt_tags ).
        MESSAGE |Found { lv_count } tags for resource.| TYPE 'I'.
      CATCH /aws1/cx_migaccessdeniedex.
        MESSAGE 'Access denied.' TYPE 'I'.
      CATCH /aws1/cx_miginternalserverex.
```

```
      MESSAGE 'Internal server error.' TYPE 'I'.
    CATCH /aws1/cx_migresourcenotfoundex.
      MESSAGE 'Resource not found.' TYPE 'I'.
    CATCH /aws1/cx_migthrottlingex.
      MESSAGE 'Request throttled.' TYPE 'I'.
    CATCH /aws1/cx_migvalidationex.
      MESSAGE 'Validation error.' TYPE 'I'.
  ENDTRY.
```

- Untuk detail API, lihat ListTagsForResourcedi AWS SDK untuk referensi SAP ABAP API.

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

`SearchImageSets`

Contoh kode berikut menunjukkan cara menggunakan`SearchImageSets`.

SDK for SAP ABAP

```
  TRY.
      " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
      oo_result = lo_mig->searchimagesets(
        iv_datastoreid = iv_datastore_id
        io_searchcriteria = io_search_criteria ).
      DATA(lt_imagesets) = oo_result->get_imagesetsmetadatasums( ).
      DATA(lv_count) = lines( lt_imagesets ).
      MESSAGE |Found { lv_count } image sets.| TYPE 'I'.
    CATCH /aws1/cx_migaccessdeniedex.
      MESSAGE 'Access denied.' TYPE 'I'.
    CATCH /aws1/cx_migconflictexception.
      MESSAGE 'Conflict error.' TYPE 'I'.
    CATCH /aws1/cx_miginternalserverex.
      MESSAGE 'Internal server error.' TYPE 'I'.
    CATCH /aws1/cx_migresourcenotfoundex.
      MESSAGE 'Resource not found.' TYPE 'I'.
    CATCH /aws1/cx_migthrottlingex.
      MESSAGE 'Request throttled.' TYPE 'I'.
```

```
  CATCH /aws1/cx_migvalidationex.
    MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- Untuk detail API, lihat SearchImageSetsdi AWS SDK untuk referensi SAP ABAP API.

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

`StartDICOMImportJob`

Contoh kode berikut menunjukkan cara menggunakan`StartDICOMImportJob`.

SDK for SAP ABAP

```
TRY.
    " iv_job_name = 'import-job-1'
    " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
    " iv_role_arn = 'arn:aws:iam::123456789012:role/ImportJobRole'
    " iv_input_s3_uri = 's3://my-bucket/input/'
    " iv_output_s3_uri = 's3://my-bucket/output/'
    oo_result = lo_mig->startdicomimportjob(
      iv_jobname = iv_job_name
      iv_datastoreid = iv_datastore_id
      iv_dataaccessrolearn = iv_role_arn
      iv_inputs3uri = iv_input_s3_uri
      iv_outputs3uri = iv_output_s3_uri ).
    DATA(lv_job_id) = oo_result->get_jobid( ).
    MESSAGE |DICOM import job started with ID: { lv_job_id }.| TYPE 'I'.
  CATCH /aws1/cx_migaccessdeniedex.
    MESSAGE 'Access denied.' TYPE 'I'.
  CATCH /aws1/cx_migconflictexception.
    MESSAGE 'Conflict error.' TYPE 'I'.
  CATCH /aws1/cx_miginternalserverex.
    MESSAGE 'Internal server error.' TYPE 'I'.
  CATCH /aws1/cx_migresourcenotfoundex.
    MESSAGE 'Resource not found.' TYPE 'I'.
  CATCH /aws1/cx_migservicequotaexcdex.
    MESSAGE 'Service quota exceeded.' TYPE 'I'.
```

```
      CATCH /aws1/cx_migthrottlingex.
        MESSAGE 'Request throttled.' TYPE 'I'.
      CATCH /aws1/cx_migvalidationex.
        MESSAGE 'Validation error.' TYPE 'I'.
    ENDTRY.
```

- Untuk detail API, lihat StartDICOMImportJobdi AWS SDK untuk referensi SAP ABAP API.

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

`TagResource`

Contoh kode berikut menunjukkan cara menggunakan`TagResource`.

SDK for SAP ABAP

```
    TRY.
        " iv_resource_arn = 'arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012'
        lo_mig->tagresource(
          iv_resourcearn = iv_resource_arn
          it_tags = it_tags ).
        MESSAGE 'Resource tagged successfully.' TYPE 'I'.
      CATCH /aws1/cx_migaccessdeniedex.
        MESSAGE 'Access denied.' TYPE 'I'.
      CATCH /aws1/cx_miginternalserverex.
        MESSAGE 'Internal server error.' TYPE 'I'.
      CATCH /aws1/cx_migresourcenotfoundex.
        MESSAGE 'Resource not found.' TYPE 'I'.
      CATCH /aws1/cx_migthrottlingex.
        MESSAGE 'Request throttled.' TYPE 'I'.
      CATCH /aws1/cx_migvalidationex.
        MESSAGE 'Validation error.' TYPE 'I'.
    ENDTRY.
```

- Untuk detail API, lihat TagResourcedi AWS SDK untuk referensi SAP ABAP API.

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

`UntagResource`

Contoh kode berikut menunjukkan cara menggunakan`UntagResource`.

SDK for SAP ABAP

```
    TRY.
        " iv_resource_arn = 'arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012'
        lo_mig->untagresource(
          iv_resourcearn = iv_resource_arn
          it_tagkeys = it_tag_keys ).
        MESSAGE 'Resource untagged successfully.' TYPE 'I'.
      CATCH /aws1/cx_migaccessdeniedex.
        MESSAGE 'Access denied.' TYPE 'I'.
      CATCH /aws1/cx_miginternalserverex.
        MESSAGE 'Internal server error.' TYPE 'I'.
      CATCH /aws1/cx_migresourcenotfoundex.
        MESSAGE 'Resource not found.' TYPE 'I'.
      CATCH /aws1/cx_migthrottlingex.
        MESSAGE 'Request throttled.' TYPE 'I'.
      CATCH /aws1/cx_migvalidationex.
        MESSAGE 'Validation error.' TYPE 'I'.
    ENDTRY.
```

- Untuk detail API, lihat UntagResourcedi AWS SDK untuk referensi SAP ABAP API.

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

`UpdateImageSetMetadata`

Contoh kode berikut menunjukkan cara menggunakan`UpdateImageSetMetadata`.

SDK for SAP ABAP

```
    TRY.
        " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
        " iv_image_set_id = '12345678901234567890123456789012345678901234567890'
        " iv_latest_version_id = '1'
        " iv_force = abap_false
        oo_result = lo_mig->updateimagesetmetadata(
          iv_datastoreid = iv_datastore_id
          iv_imagesetid = iv_image_set_id
          iv_latestversionid = iv_latest_version_id
          io_updateimagesetmetupdates = io_metadata_updates
          iv_force = iv_force ).
        DATA(lv_new_version) = oo_result->get_latestversionid( ).
        MESSAGE |Image set metadata updated to version: { lv_new_version }.| TYPE
'I'.
      CATCH /aws1/cx_migaccessdeniedex.
        MESSAGE 'Access denied.' TYPE 'I'.
      CATCH /aws1/cx_migconflictexception.
        MESSAGE 'Conflict error.' TYPE 'I'.
      CATCH /aws1/cx_miginternalserverex.
        MESSAGE 'Internal server error.' TYPE 'I'.
      CATCH /aws1/cx_migresourcenotfoundex.
        MESSAGE 'Image set not found.' TYPE 'I'.
      CATCH /aws1/cx_migservicequotaexcdex.
        MESSAGE 'Service quota exceeded.' TYPE 'I'.
      CATCH /aws1/cx_migthrottlingex.
        MESSAGE 'Request throttled.' TYPE 'I'.
      CATCH /aws1/cx_migvalidationex.
        MESSAGE 'Validation error.' TYPE 'I'.
    ENDTRY.
```

- Untuk detail API, lihat UpdateImageSetMetadatadi AWS SDK untuk referensi SAP ABAP API.

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

HealthLake contoh menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan mengimplementasikan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan. HealthLake

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`CreateFhirDataStore`

Contoh kode berikut menunjukkan cara menggunakan`CreateFHIRDatastore`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_datastore_name = 'MyHealthLakeDataStore'
        oo_result = lo_hll->createfhirdatastore(
```

```
          iv_datastorename = iv_datastore_name
          iv_datastoretypeversion = 'R4'
        ).
        MESSAGE 'Data store created successfully.' TYPE 'I'.
      CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
      CATCH /aws1/cx_hllinternalserverex INTO DATA(lo_internal_ex).
        lv_error = |Internal server error: { lo_internal_ex->av_err_code }-
{ lo_internal_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_internal_ex.
      CATCH /aws1/cx_hllthrottlingex INTO DATA(lo_throttling_ex).
        lv_error = |Throttling error: { lo_throttling_ex->av_err_code }-
{ lo_throttling_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_throttling_ex.
    ENDTRY.
```

- Untuk detail API, lihat createFhirDataStore di AWS SDK untuk referensi API SAP ABAP.

`DeleteFhirDataStore`

Contoh kode berikut menunjukkan cara menggunakan`DeleteFHIRDatastore`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
        oo_result = lo_hll->deletefhirdatastore(
          iv_datastoreid = iv_datastore_id
        ).
```

```
        MESSAGE 'Data store deleted successfully.' TYPE 'I'.
      CATCH /aws1/cx_hllaccessdeniedex INTO DATA(lo_access_ex).
        DATA(lv_error) = |Access denied: { lo_access_ex->av_err_code }-
{ lo_access_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_access_ex.
      CATCH /aws1/cx_hllconflictexception INTO DATA(lo_conflict_ex).
        lv_error = |Conflict error: { lo_conflict_ex->av_err_code }-
{ lo_conflict_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_conflict_ex.
      CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
        lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_notfound_ex.
    ENDTRY.
```

- Untuk detail API, lihat DeleteFhirDataStore di AWS SDK untuk referensi API SAP ABAP.

`DescribeFHIRDatastore`

Contoh kode berikut menunjukkan cara menggunakan`DescribeFHIRDatastore`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
        oo_result = lo_hll->describefhirdatastore(
          iv_datastoreid = iv_datastore_id
        ).
        DATA(lo_datastore_properties) = oo_result->get_datastoreproperties( ).
        IF lo_datastore_properties IS BOUND.
          DATA(lv_datastore_name) = lo_datastore_properties->get_datastorename( ).
```

```
          DATA(lv_datastore_status) = lo_datastore_properties-
>get_datastorestatus( ).
          MESSAGE 'Data store described successfully.' TYPE 'I'.
        ENDIF.
      CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
        DATA(lv_error) = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_notfound_ex.
      CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        lv_error = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
    ENDTRY.
```

- Untuk detail API, lihat DescribeFhirDataStore di AWS SDK untuk referensi API SAP ABAP.

`DescribeFHIRExportJob`

Contoh kode berikut menunjukkan cara menggunakan`DescribeFHIRExportJob`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
        " iv_job_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
        oo_result = lo_hll->describefhirexportjob(
          iv_datastoreid = iv_datastore_id
          iv_jobid = iv_job_id
        ).
        DATA(lo_export_job_properties) = oo_result->get_exportjobproperties( ).
        IF lo_export_job_properties IS BOUND.
          DATA(lv_job_status) = lo_export_job_properties->get_jobstatus( ).
```

```
          MESSAGE |Export job status: { lv_job_status }.| TYPE 'I'.
        ENDIF.
      CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
        DATA(lv_error) = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_notfound_ex.
      CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        lv_error = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
    ENDTRY.
```

- Untuk detail API, lihat DescribeFHIRExportJobdi AWS SDK untuk referensi SAP ABAP API.

`DescribeFHIRImportJob`

Contoh kode berikut menunjukkan cara menggunakan`DescribeFHIRImportJob`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
        " iv_job_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
        oo_result = lo_hll->describefhirimportjob(
          iv_datastoreid = iv_datastore_id
          iv_jobid = iv_job_id
        ).
        DATA(lo_import_job_properties) = oo_result->get_importjobproperties( ).
        IF lo_import_job_properties IS BOUND.
          DATA(lv_job_status) = lo_import_job_properties->get_jobstatus( ).
          MESSAGE |Import job status: { lv_job_status }.| TYPE 'I'.
        ENDIF.
```

```
      CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
        DATA(lv_error) = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_notfound_ex.
      CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        lv_error = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
    ENDTRY.
```

- Untuk detail API, lihat DescribeFHIRImportJobdi AWS SDK untuk referensi SAP ABAP API.

`ListFHIRDatastores`

Contoh kode berikut menunjukkan cara menggunakan`ListFHIRDatastores`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_hll->listfhirdatastores( ).
        DATA(lt_datastores) = oo_result->get_datastorepropertieslist( ).
        DATA(lv_datastore_count) = lines( lt_datastores ).
        MESSAGE |Found { lv_datastore_count } data store(s).| TYPE 'I'.
      CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
      CATCH /aws1/cx_hllthrottlingex INTO DATA(lo_throttling_ex).
        lv_error = |Throttling error: { lo_throttling_ex->av_err_code }-
{ lo_throttling_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
```

```
        RAISE EXCEPTION lo_throttling_ex.
    ENDTRY.
```

- Untuk detail API, lihat ListfHirDataStores di AWS SDK untuk referensi API SAP ABAP.

`ListFHIRExportJobs`

Contoh kode berikut menunjukkan cara menggunakan`ListFHIRExportJobs`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
        IF iv_submitted_after IS NOT INITIAL.
          oo_result = lo_hll->listfhirexportjobs(
            iv_datastoreid = iv_datastore_id
            iv_submittedafter = iv_submitted_after
          ).
        ELSE.
          oo_result = lo_hll->listfhirexportjobs(
            iv_datastoreid = iv_datastore_id
          ).
        ENDIF.
        DATA(lt_export_jobs) = oo_result->get_exportjobpropertieslist( ).
        DATA(lv_job_count) = lines( lt_export_jobs ).
        MESSAGE |Found { lv_job_count } export job(s).| TYPE 'I'.
      CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
      CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
        lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
```

```
        RAISE EXCEPTION lo_notfound_ex.
    ENDTRY.
```

- Untuk detail API, lihat ListFHIRExportJobsdi AWS SDK untuk referensi SAP ABAP API.

`ListFHIRImportJobs`

Contoh kode berikut menunjukkan cara menggunakan`ListFHIRImportJobs`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
        IF iv_submitted_after IS NOT INITIAL.
          oo_result = lo_hll->listfhirimportjobs(
            iv_datastoreid = iv_datastore_id
            iv_submittedafter = iv_submitted_after
          ).
        ELSE.
          oo_result = lo_hll->listfhirimportjobs(
            iv_datastoreid = iv_datastore_id
          ).
        ENDIF.
        DATA(lt_import_jobs) = oo_result->get_importjobpropertieslist( ).
        DATA(lv_job_count) = lines( lt_import_jobs ).
        MESSAGE |Found { lv_job_count } import job(s).| TYPE 'I'.
      CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
      CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
        lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
```

```
        RAISE EXCEPTION lo_notfound_ex.
    ENDTRY.
```

- Untuk detail API, lihat ListFHIRImportJobsdi AWS SDK untuk referensi SAP ABAP API.

`ListTagsForResource`

Contoh kode berikut menunjukkan cara menggunakan`ListTagsForResource`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_resource_arn = 'arn:aws:healthlake:us-east-1:123456789012:datastore/
fhir/a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
        DATA(lo_result) = lo_hll->listtagsforresource(
          iv_resourcearn = iv_resource_arn
        ).
        ot_tags = lo_result->get_tags( ).
        DATA(lv_tag_count) = lines( ot_tags ).
        MESSAGE |Found { lv_tag_count } tag(s).| TYPE 'I'.
      CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
      CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
        lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_notfound_ex.
    ENDTRY.
```

- Untuk detail API, lihat ListTagsForResourcedi AWS SDK untuk referensi SAP ABAP API.

`StartFHIRExportJob`

Contoh kode berikut menunjukkan cara menggunakan`StartFHIRExportJob`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_job_name = 'MyExportJob'
        " iv_output_s3_uri = 's3://my-bucket/export/output/'
        " iv_kms_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012'
        " iv_data_access_role_arn = 'arn:aws:iam::123456789012:role/
HealthLakeExportRole'
        oo_result = lo_hll->startfhirexportjob(
          iv_jobname = iv_job_name
          io_outputdataconfig = NEW /aws1/cl_hlloutputdataconfig(
            io_s3configuration = NEW /aws1/cl_hlls3configuration(
              iv_s3uri = iv_output_s3_uri
              iv_kmskeyid = iv_kms_key_id
            )
          )
          iv_dataaccessrolearn = iv_data_access_role_arn
          iv_datastoreid = iv_datastore_id
        ).
        DATA(lv_job_id) = oo_result->get_jobid( ).
        MESSAGE |Export job started with ID { lv_job_id }.| TYPE 'I'.
      CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
      CATCH /aws1/cx_hllthrottlingex INTO DATA(lo_throttling_ex).
        lv_error = |Throttling error: { lo_throttling_ex->av_err_code }-
{ lo_throttling_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_throttling_ex.
      CATCH /aws1/cx_hllaccessdeniedex INTO DATA(lo_access_ex).
```

```
        lv_error = |Access denied: { lo_access_ex->av_err_code }-{ lo_access_ex-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_access_ex.
    ENDTRY.
```

- Untuk detail API, lihat StartFHIRExportJobdi AWS SDK untuk referensi SAP ABAP API.

`StartFHIRImportJob`

Contoh kode berikut menunjukkan cara menggunakan`StartFHIRImportJob`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_job_name = 'MyImportJob'
        " iv_input_s3_uri = 's3://my-bucket/import/data.ndjson'
        " iv_job_output_s3_uri = 's3://my-bucket/import/output/'
        " iv_kms_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012'
        " iv_data_access_role_arn = 'arn:aws:iam::123456789012:role/
HealthLakeImportRole'
        oo_result = lo_hll->startfhirimportjob(
          iv_jobname = iv_job_name
          io_inputdataconfig = NEW /aws1/cl_hllinputdataconfig( iv_s3uri =
 iv_input_s3_uri )
          io_joboutputdataconfig = NEW /aws1/cl_hlloutputdataconfig(
            io_s3configuration = NEW /aws1/cl_hlls3configuration(
              iv_s3uri = iv_job_output_s3_uri
              iv_kmskeyid = iv_kms_key_id
            )
          )
          iv_dataaccessrolearn = iv_data_access_role_arn
          iv_datastoreid = iv_datastore_id
```

```
        ).
        DATA(lv_job_id) = oo_result->get_jobid( ).
        MESSAGE |Import job started with ID { lv_job_id }.| TYPE 'I'.
      CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
      CATCH /aws1/cx_hllthrottlingex INTO DATA(lo_throttling_ex).
        lv_error = |Throttling error: { lo_throttling_ex->av_err_code }-
{ lo_throttling_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_throttling_ex.
      CATCH /aws1/cx_hllaccessdeniedex INTO DATA(lo_access_ex).
        lv_error = |Access denied: { lo_access_ex->av_err_code }-{ lo_access_ex-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_access_ex.
    ENDTRY.
```

- Untuk detail API, lihat StartFHIRImportJobdi AWS SDK untuk referensi SAP ABAP API.

`TagResource`

Contoh kode berikut menunjukkan cara menggunakan`TagResource`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_resource_arn = 'arn:aws:healthlake:us-east-1:123456789012:datastore/
fhir/a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
        lo_hll->tagresource(
          iv_resourcearn = iv_resource_arn
          it_tags = it_tags
```

```
        ).
        MESSAGE 'Resource tagged successfully.' TYPE 'I'.
      CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
      CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
        lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_notfound_ex.
    ENDTRY.
```

- Untuk detail API, lihat TagResourcedi AWS SDK untuk referensi SAP ABAP API.

`UntagResource`

Contoh kode berikut menunjukkan cara menggunakan`UntagResource`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_resource_arn = 'arn:aws:healthlake:us-east-1:123456789012:datastore/
fhir/a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
        lo_hll->untagresource(
          iv_resourcearn = iv_resource_arn
          it_tagkeys = it_tag_keys
        ).
        MESSAGE 'Resource untagged successfully.' TYPE 'I'.
      CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
```

```
        RAISE EXCEPTION lo_validation_ex.
      CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
        lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_notfound_ex.
    ENDTRY.
```

- Untuk detail API, lihat UntagResourcedi AWS SDK untuk referensi SAP ABAP API.

Contoh IAM menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan mengimplementasikan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan IAM.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`AttachRolePolicy`

Contoh kode berikut menunjukkan cara menggunakan`AttachRolePolicy`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_iam->attachrolepolicy(
      iv_rolename = iv_role_name
      iv_policyarn = iv_policy_arn ).
    MESSAGE 'Policy attached to role successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Role or policy does not exist.' TYPE 'E'.
  CATCH /aws1/cx_iamlimitexceededex.
    MESSAGE 'Policy attachment limit exceeded.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat AttachRolePolicydi AWS SDK untuk referensi SAP ABAP API.

`AttachUserPolicy`

Contoh kode berikut menunjukkan cara menggunakan`AttachUserPolicy`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_iam->attachuserpolicy(
      iv_username = iv_user_name
      iv_policyarn = iv_policy_arn ).
    MESSAGE 'Policy attached to user successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'User or policy does not exist.' TYPE 'E'.
  CATCH /aws1/cx_iamlimitexceededex.
    MESSAGE 'Policy attachment limit exceeded.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat AttachUserPolicydi AWS SDK untuk referensi SAP ABAP API.

`CreateAccessKey`

Contoh kode berikut menunjukkan cara menggunakan`CreateAccessKey`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_iam->createaccesskey(
      iv_username = iv_user_name ).
    MESSAGE 'Access key created successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'User does not exist.' TYPE 'E'.
  CATCH /aws1/cx_iamlimitexceededex.
    MESSAGE 'Maximum number of access keys reached.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat CreateAccessKeydi AWS SDK untuk referensi SAP ABAP API.

`CreateAccountAlias`

Contoh kode berikut menunjukkan cara menggunakan`CreateAccountAlias`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_iam->createaccountalias(
```

```
        iv_accountalias = iv_account_alias ).
      MESSAGE 'Account alias created successfully.' TYPE 'I'.
    CATCH /aws1/cx_iamentityalrdyexex.
      MESSAGE 'Account alias already exists.' TYPE 'E'.
    CATCH /aws1/cx_iamlimitexceededex.
      MESSAGE 'Account alias limit exceeded.' TYPE 'E'.
  ENDTRY.
```

- Untuk detail API, lihat CreateAccountAliasdi AWS SDK untuk referensi SAP ABAP API.

`CreatePolicy`

Contoh kode berikut menunjukkan cara menggunakan`CreatePolicy`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
  TRY.
      oo_result = lo_iam->createpolicy(
        iv_policyname = iv_policy_name
        iv_policydocument = iv_policy_document
        iv_description = iv_description ).
      MESSAGE 'Policy created successfully.' TYPE 'I'.
    CATCH /aws1/cx_iamentityalrdyexex.
      MESSAGE 'Policy already exists.' TYPE 'E'.
    CATCH /aws1/cx_iammalformedplydocex.
      MESSAGE 'Policy document is malformed.' TYPE 'E'.
    CATCH /aws1/cx_iamlimitexceededex.
      MESSAGE 'Policy limit exceeded.' TYPE 'E'.
  ENDTRY.
```

- Untuk detail API, lihat CreatePolicydi AWS SDK untuk referensi SAP ABAP API.

`CreatePolicyVersion`

Contoh kode berikut menunjukkan cara menggunakan`CreatePolicyVersion`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_iam->createpolicyversion(
      iv_policyarn = iv_policy_arn
      iv_policydocument = iv_policy_document
      iv_setasdefault = iv_set_as_default ).
    MESSAGE 'Policy version created successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Policy does not exist.' TYPE 'E'.
  CATCH /aws1/cx_iammalformedplydocex.
    MESSAGE 'Policy document is malformed.' TYPE 'E'.
  CATCH /aws1/cx_iamlimitexceededex.
    MESSAGE 'Policy version limit exceeded.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat CreatePolicyVersiondi AWS SDK untuk referensi SAP ABAP API.

`CreateRole`

Contoh kode berikut menunjukkan cara menggunakan`CreateRole`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_iam->createrole(
      iv_rolename = iv_role_name
      iv_assumerolepolicydocument = iv_assume_role_policy_document ).
    MESSAGE 'Role created successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamentityalrdyexex.
    MESSAGE 'Role already exists.' TYPE 'E'.
  CATCH /aws1/cx_iammalformedplydocex.
    MESSAGE 'Assume role policy document is malformed.' TYPE 'E'.
  CATCH /aws1/cx_iamlimitexceededex.
    MESSAGE 'Role limit exceeded.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat CreateRoledi AWS SDK untuk referensi SAP ABAP API.

`CreateServiceLinkedRole`

Contoh kode berikut menunjukkan cara menggunakan`CreateServiceLinkedRole`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_iam->listpolicyversions(
      iv_policyarn = iv_policy_arn ).
    MESSAGE 'Retrieved policy versions list.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Policy does not exist.' TYPE 'E'.
  CATCH /aws1/cx_iamservicefailureex.
    MESSAGE 'Service failure when listing policy versions.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat CreateServiceLinkedRoledi AWS SDK untuk referensi SAP ABAP API.

`CreateUser`

Contoh kode berikut menunjukkan cara menggunakan`CreateUser`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_iam->createuser(
      iv_username = iv_user_name ).
    MESSAGE 'User created successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamentityalrdyexex.
    MESSAGE 'User already exists.' TYPE 'E'.
  CATCH /aws1/cx_iamlimitexceededex.
    MESSAGE 'Limit exceeded for IAM users.' TYPE 'E'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Entity does not exist.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat CreateUserdi AWS SDK untuk referensi SAP ABAP API.

`DeleteAccessKey`

Contoh kode berikut menunjukkan cara menggunakan`DeleteAccessKey`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
```

```
    lo_iam->deleteaccesskey(
      iv_accesskeyid = iv_access_key_id
      iv_username = iv_user_name ).
    MESSAGE 'Access key deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Access key or user does not exist.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DeleteAccessKeydi AWS SDK untuk referensi SAP ABAP API.

`DeleteAccountAlias`

Contoh kode berikut menunjukkan cara menggunakan`DeleteAccountAlias`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_iam->deleteaccountalias(
      iv_accountalias = iv_account_alias ).
    MESSAGE 'Account alias deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Account alias does not exist.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DeleteAccountAliasdi AWS SDK untuk referensi SAP ABAP API.

`DeletePolicy`

Contoh kode berikut menunjukkan cara menggunakan`DeletePolicy`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_iam->deletepolicy( iv_policyarn = iv_policy_arn ).
    MESSAGE 'Policy deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Policy does not exist.' TYPE 'E'.
  CATCH /aws1/cx_iamdeleteconflictex.
    MESSAGE 'Policy cannot be deleted due to attachments.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DeletePolicydi AWS SDK untuk referensi SAP ABAP API.

`DeletePolicyVersion`

Contoh kode berikut menunjukkan cara menggunakan`DeletePolicyVersion`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_iam->deletepolicyversion(
      iv_policyarn = iv_policy_arn
      iv_versionid = iv_version_id ).
    MESSAGE 'Policy version deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Policy or version does not exist.' TYPE 'E'.
```

```
  CATCH /aws1/cx_iamdeleteconflictex.
    MESSAGE 'Cannot delete default policy version.' TYPE 'E'.
  CATCH /aws1/cx_iamlimitexceededex.
    MESSAGE 'Limit exceeded.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DeletePolicyVersiondi AWS SDK untuk referensi SAP ABAP API.

`DeleteRole`

Contoh kode berikut menunjukkan cara menggunakan`DeleteRole`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_iam->deleterole( iv_rolename = iv_role_name ).
    MESSAGE 'Role deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Role does not exist.' TYPE 'E'.
  CATCH /aws1/cx_iamdeleteconflictex.
    MESSAGE 'Role cannot be deleted due to attached resources.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DeleteRoledi AWS SDK untuk referensi SAP ABAP API.

`DeleteUser`

Contoh kode berikut menunjukkan cara menggunakan`DeleteUser`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_iam->deleteuser( iv_username = iv_user_name ).
    MESSAGE 'User deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'User does not exist.' TYPE 'E'.
  CATCH /aws1/cx_iamdeleteconflictex.
    MESSAGE 'User cannot be deleted due to attached resources.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DeleteUserdi AWS SDK untuk referensi SAP ABAP API.

`DetachRolePolicy`

Contoh kode berikut menunjukkan cara menggunakan`DetachRolePolicy`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_iam->detachrolepolicy(
      iv_rolename = iv_role_name
      iv_policyarn = iv_policy_arn ).
    MESSAGE 'Policy detached from role successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Role or policy does not exist.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DetachRolePolicydi AWS SDK untuk referensi SAP ABAP API.

`DetachUserPolicy`

Contoh kode berikut menunjukkan cara menggunakan`DetachUserPolicy`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_iam->detachuserpolicy(
      iv_username = iv_user_name
      iv_policyarn = iv_policy_arn ).
    MESSAGE 'Policy detached from user successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'User or policy does not exist.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DetachUserPolicydi AWS SDK untuk referensi SAP ABAP API.

`GenerateCredentialReport`

Contoh kode berikut menunjukkan cara menggunakan`GenerateCredentialReport`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_iam->generatecredentialreport( ).
    MESSAGE 'Credential report generation started.' TYPE 'I'.
  CATCH /aws1/cx_iamlimitexceededex.
    MESSAGE 'Report generation limit exceeded.' TYPE 'E'.
  CATCH /aws1/cx_iamservicefailureex.
    MESSAGE 'Service failure when generating credential report.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat GenerateCredentialReportdi AWS SDK untuk referensi SAP ABAP API.

`GetAccessKeyLastUsed`

Contoh kode berikut menunjukkan cara menggunakan`GetAccessKeyLastUsed`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_iam->getaccesskeylastused(
      iv_accesskeyid = iv_access_key_id ).
    MESSAGE 'Retrieved access key last used information.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Access key does not exist.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat GetAccessKeyLastUseddi AWS SDK untuk referensi SAP ABAP API.

`GetAccountAuthorizationDetails`

Contoh kode berikut menunjukkan cara menggunakan`GetAccountAuthorizationDetails`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_iam->getaccountauthdetails( ).
        MESSAGE 'Retrieved account authorization details.' TYPE 'I'.
      CATCH /aws1/cx_iamservicefailureex.
        MESSAGE 'Service failure when getting account authorization details.' TYPE
'E'.
    ENDTRY.
```

- Untuk detail API, lihat GetAccountAuthorizationDetailsdi AWS SDK untuk referensi SAP ABAP API.

`GetAccountPasswordPolicy`

Contoh kode berikut menunjukkan cara menggunakan`GetAccountPasswordPolicy`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_iam->getaccountpasswordpolicy( ).
        MESSAGE 'Retrieved account password policy.' TYPE 'I'.
      CATCH /aws1/cx_iamnosuchentityex.
        MESSAGE 'No password policy exists.' TYPE 'I'.
      CATCH /aws1/cx_iamservicefailureex.
        MESSAGE 'Service failure when getting password policy.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat GetAccountPasswordPolicydi AWS SDK untuk referensi SAP ABAP API.

`GetAccountSummary`

Contoh kode berikut menunjukkan cara menggunakan`GetAccountSummary`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_iam->getaccountsummary( ).
    MESSAGE 'Retrieved account summary.' TYPE 'I'.
  CATCH /aws1/cx_iamservicefailureex.
    MESSAGE 'Service failure when getting account summary.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat GetAccountSummarydi AWS SDK untuk referensi SAP ABAP API.

`GetCredentialReport`

Contoh kode berikut menunjukkan cara menggunakan`GetCredentialReport`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
```

```
      oo_result = lo_iam->getcredentialreport( ).
      MESSAGE 'Retrieved credential report.' TYPE 'I'.
    CATCH /aws1/cx_iamcredrptnotpresen00.
      MESSAGE 'Credential report not present.' TYPE 'E'.
    CATCH /aws1/cx_iamcredrptexpiredex.
      MESSAGE 'Credential report expired.' TYPE 'E'.
    CATCH /aws1/cx_iamcredrptnotreadyex.
      MESSAGE 'Credential report not ready.' TYPE 'E'.
    CATCH /aws1/cx_iamservicefailureex.
      MESSAGE 'Service failure when getting credential report.' TYPE 'E'.
  ENDTRY.
```

- Untuk detail API, lihat GetCredentialReportdi AWS SDK untuk referensi SAP ABAP API.

`GetPolicy`

Contoh kode berikut menunjukkan cara menggunakan`GetPolicy`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
  TRY.
      oo_result = lo_iam->getpolicy( iv_policyarn = iv_policy_arn ).
      MESSAGE 'Retrieved policy information.' TYPE 'I'.
    CATCH /aws1/cx_iamnosuchentityex.
      MESSAGE 'Policy does not exist.' TYPE 'E'.
  ENDTRY.
```

- Untuk detail API, lihat GetPolicydi AWS SDK untuk referensi SAP ABAP API.

`GetPolicyVersion`

Contoh kode berikut menunjukkan cara menggunakan`GetPolicyVersion`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_iam->getpolicyversion(
      iv_policyarn = iv_policy_arn
      iv_versionid = iv_version_id ).
    MESSAGE 'Retrieved policy version information.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Policy or version does not exist.' TYPE 'E'.
  CATCH /aws1/cx_iaminvalidinputex.
    MESSAGE 'Invalid input provided.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat GetPolicyVersiondi AWS SDK untuk referensi SAP ABAP API.

`GetRole`

Contoh kode berikut menunjukkan cara menggunakan`GetRole`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_iam->getrole( iv_rolename = iv_role_name ).
    MESSAGE 'Retrieved role information.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Role does not exist.' TYPE 'E'.
```

```
    ENDTRY.
```

- Untuk detail API, lihat GetRoledi AWS SDK untuk referensi SAP ABAP API.

`ListAccessKeys`

Contoh kode berikut menunjukkan cara menggunakan`ListAccessKeys`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_iam->listaccesskeys(
      iv_username = iv_user_name ).
    MESSAGE 'Retrieved access key list.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'User does not exist.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat ListAccessKeysdi AWS SDK untuk referensi SAP ABAP API.

`ListAccountAliases`

Contoh kode berikut menunjukkan cara menggunakan`ListAccountAliases`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_iam->listaccountaliases( ).
    MESSAGE 'Retrieved account alias list.' TYPE 'I'.
  CATCH /aws1/cx_iamservicefailureex.
    MESSAGE 'Service failure when listing account aliases.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat ListAccountAliasesdi AWS SDK untuk referensi SAP ABAP API.

`ListAttachedRolePolicies`

Contoh kode berikut menunjukkan cara menggunakan`ListAttachedRolePolicies`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_iam->listattachedrolepolicies(
      iv_rolename = iv_role_name ).
    MESSAGE 'Retrieved attached policy list for role.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Role does not exist.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat ListAttachedRolePoliciesdi AWS SDK untuk referensi SAP ABAP API.

`ListGroups`

Contoh kode berikut menunjukkan cara menggunakan`ListGroups`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_iam->listgroups( ).
    MESSAGE 'Retrieved group list.' TYPE 'I'.
  CATCH /aws1/cx_iamservicefailureex.
    MESSAGE 'Service failure when listing groups.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat ListGroupsdi AWS SDK untuk referensi SAP ABAP API.

`ListPolicies`

Contoh kode berikut menunjukkan cara menggunakan`ListPolicies`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_iam->listpolicies( iv_scope = iv_scope ).
    MESSAGE 'Retrieved policy list.' TYPE 'I'.
  CATCH /aws1/cx_iamservicefailureex.
    MESSAGE 'Service failure when listing policies.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat ListPoliciesdi AWS SDK untuk referensi SAP ABAP API.

`ListPolicyVersions`

Contoh kode berikut menunjukkan cara menggunakan`ListPolicyVersions`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_iam->listpolicyversions(
      iv_policyarn = iv_policy_arn ).
    MESSAGE 'Retrieved policy versions list.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Policy does not exist.' TYPE 'E'.
  CATCH /aws1/cx_iamservicefailureex.
    MESSAGE 'Service failure when listing policy versions.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat ListPolicyVersionsdi AWS SDK untuk referensi SAP ABAP API.

`ListRolePolicies`

Contoh kode berikut menunjukkan cara menggunakan`ListRolePolicies`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_iam->listrolepolicies(
      iv_rolename = iv_role_name ).
```

```
    MESSAGE 'Retrieved inline policy list for role.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Role does not exist.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat ListRolePoliciesdi AWS SDK untuk referensi SAP ABAP API.

`ListRoles`

Contoh kode berikut menunjukkan cara menggunakan`ListRoles`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_iam->listroles( ).
    MESSAGE 'Retrieved role list.' TYPE 'I'.
  CATCH /aws1/cx_iamservicefailureex.
    MESSAGE 'Service failure when listing roles.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat ListRolesdi AWS SDK untuk referensi SAP ABAP API.

`ListsAMLProviders`

Contoh kode berikut menunjukkan cara menggunakan`ListSAMLProviders`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_iam->listsamlproviders( ).
    MESSAGE 'Retrieved SAML provider list.' TYPE 'I'.
  CATCH /aws1/cx_iamservicefailureex.
    MESSAGE 'Service failure when listing SAML providers.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat ListSamlProviders di AWS SDK untuk referensi SAP ABAP API.

`ListUsers`

Contoh kode berikut menunjukkan cara menggunakan`ListUsers`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_iam->listusers( ).
    MESSAGE 'Retrieved user list.' TYPE 'I'.
  CATCH /aws1/cx_iamservicefailureex.
    MESSAGE 'Service failure when listing users.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat ListUsersdi AWS SDK untuk referensi SAP ABAP API.

`SetDefaultPolicyVersion`

Contoh kode berikut menunjukkan cara menggunakan`SetDefaultPolicyVersion`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_iam->setdefaultpolicyversion(
      iv_policyarn = iv_policy_arn
      iv_versionid = iv_version_id ).
    MESSAGE 'Default policy version set successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Policy or version does not exist.' TYPE 'E'.
  CATCH /aws1/cx_iaminvalidinputex.
    MESSAGE 'Invalid input provided.' TYPE 'E'.
  CATCH /aws1/cx_iamlimitexceededex.
    MESSAGE 'Limit exceeded.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat SetDefaultPolicyVersiondi AWS SDK untuk referensi SAP ABAP API.

`UpdateAccessKey`

Contoh kode berikut menunjukkan cara menggunakan`UpdateAccessKey`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_iam->updateaccesskey(
      iv_accesskeyid = iv_access_key_id
      iv_status = iv_status
```

```
        iv_username = iv_user_name ).
      MESSAGE 'Access key updated successfully.' TYPE 'I'.
    CATCH /aws1/cx_iamnosuchentityex.
      MESSAGE 'Access key or user does not exist.' TYPE 'E'.
  ENDTRY.
```

- Untuk detail API, lihat UpdateAccessKeydi AWS SDK untuk referensi SAP ABAP API.

`UpdateUser`

Contoh kode berikut menunjukkan cara menggunakan`UpdateUser`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
  TRY.
      lo_iam->updateuser(
        iv_username = iv_user_name
        iv_newusername = iv_new_user_name ).
      MESSAGE 'User updated successfully.' TYPE 'I'.
    CATCH /aws1/cx_iamnosuchentityex.
      MESSAGE 'User does not exist.' TYPE 'E'.
    CATCH /aws1/cx_iamentityalrdyexex.
      MESSAGE 'New user name already exists.' TYPE 'E'.
  ENDTRY.
```

- Untuk detail API, lihat UpdateUserdi AWS SDK untuk referensi SAP ABAP API.

AWS IoT contoh menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan mengimplementasikan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan. AWS IoT

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`AttachThingPrincipal`

Contoh kode berikut menunjukkan cara menggunakan`AttachThingPrincipal`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    CONSTANTS cv_pfl TYPE /aws1/rt_profile_id VALUE 'ZCODE_DEMO'.
    DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
    DATA(lo_iot) = /aws1/cl_iot_factory=>create( lo_session ).
    TRY.
        lo_iot->attachthingprincipal(
          iv_thingname = iv_thing_name
          iv_principal = iv_principal ).
        MESSAGE |Principal attached to IoT thing '{ iv_thing_name }'.| TYPE 'I'.
      CATCH /aws1/cx_iotresourcenotfoundex INTO DATA(lo_ex).
        MESSAGE |Resource not found when attaching principal to
'{ iv_thing_name }'.| TYPE 'I'.
        RAISE EXCEPTION lo_ex.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_svc_ex).
        MESSAGE lo_svc_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_svc_ex.
    ENDTRY.
```

- Untuk detail API, lihat AttachThingPrincipaldi AWS SDK untuk referensi SAP ABAP API.

`CreateKeysAndCertificate`

Contoh kode berikut menunjukkan cara menggunakan`CreateKeysAndCertificate`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
CONSTANTS cv_pfl TYPE /aws1/rt_profile_id VALUE 'ZCODE_DEMO'.
DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
DATA(lo_iot) = /aws1/cl_iot_factory=>create( lo_session ).
TRY.
    oo_result = lo_iot->createkeysandcertificate( iv_setasactive = abap_true ).
    MESSAGE |Certificate created: { oo_result->get_certificateid( ) }| TYPE 'I'.
  CATCH /aws1/cx_rt_service_generic INTO DATA(lo_ex).
    MESSAGE lo_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_ex.
ENDTRY.
```

- Untuk detail API, lihat CreateKeysAndCertificatedi AWS SDK untuk referensi SAP ABAP API.

`CreateThing`

Contoh kode berikut menunjukkan cara menggunakan`CreateThing`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    CONSTANTS cv_pfl TYPE /aws1/rt_profile_id VALUE 'ZCODE_DEMO'.
    DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
    DATA(lo_iot) = /aws1/cl_iot_factory=>create( lo_session ).
    TRY.
        oo_result = lo_iot->creatething(
          iv_thingname = iv_thing_name ).
        MESSAGE |IoT thing created: { oo_result->get_thingname( ) } ARN:
{ oo_result->get_thingarn( ) }| TYPE 'I'.
      CATCH /aws1/cx_iotresrcalrdyexistsex.
        MESSAGE |IoT thing '{ iv_thing_name }' already exists.| TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_ex).
        MESSAGE lo_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_ex.
    ENDTRY.
```

- Untuk detail API, lihat CreateThingdi AWS SDK untuk referensi SAP ABAP API.

`CreateTopicRule`

Contoh kode berikut menunjukkan cara menggunakan`CreateTopicRule`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    CONSTANTS cv_pfl TYPE /aws1/rt_profile_id VALUE 'ZCODE_DEMO'.
    DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
    DATA(lo_iot) = /aws1/cl_iot_factory=>create( lo_session ).
    TRY.
        " Build the SNS action that will receive messages matching the rule.
        DATA lo_sns_action TYPE REF TO /aws1/cl_iotsnsaction.
        CREATE OBJECT lo_sns_action
          EXPORTING
            iv_targetarn = iv_sns_action_arn
            iv_rolearn   = iv_role_arn.
```

```
    DATA lo_action TYPE REF TO /aws1/cl_iotaction.
    CREATE OBJECT lo_action
      EXPORTING
        io_sns = lo_sns_action.

    DATA lt_actions TYPE /aws1/cl_iotaction=>tt_actionlist.
    APPEND lo_action TO lt_actions.

    " iv_topic = 'my/iot/topic' - The MQTT topic pattern to match
    DATA lo_payload TYPE REF TO /aws1/cl_iottopicrulepayload.
    CREATE OBJECT lo_payload
      EXPORTING
        iv_sql     = |SELECT * FROM '{ iv_topic }'|
        it_actions = lt_actions.

    lo_iot->createtopicrule(
      iv_rulename         = iv_rule_name
      io_topicrulepayload = lo_payload ).
    MESSAGE |IoT topic rule created: { iv_rule_name }| TYPE 'I'.
  CATCH /aws1/cx_iotresrcalrdyexistsex.
    MESSAGE |Topic rule '{ iv_rule_name }' already exists.| TYPE 'I'.
  CATCH /aws1/cx_rt_service_generic INTO DATA(lo_ex).
    MESSAGE lo_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_ex.
ENDTRY.
```

- Untuk detail API, lihat CreateTopicRuledi AWS SDK untuk referensi SAP ABAP API.

`DeleteCertificate`

Contoh kode berikut menunjukkan cara menggunakan`DeleteCertificate`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
CONSTANTS cv_pfl TYPE /aws1/rt_profile_id VALUE 'ZCODE_DEMO'.
```

```
    DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
    DATA(lo_iot) = /aws1/cl_iot_factory=>create( lo_session ).
    TRY.
        " Certificates must be deactivated before they can be deleted.
        lo_iot->updatecertificate(
          iv_certificateid = iv_certificate_id
          iv_newstatus     = 'INACTIVE' ).
        lo_iot->deletecertificate( iv_certificateid = iv_certificate_id ).
        MESSAGE |Certificate deleted: { iv_certificate_id }| TYPE 'I'.
      CATCH /aws1/cx_iotresourcenotfoundex INTO DATA(lo_ex).
        MESSAGE |Certificate '{ iv_certificate_id }' not found.| TYPE 'I'.
        RAISE EXCEPTION lo_ex.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_svc_ex).
        MESSAGE lo_svc_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_svc_ex.
    ENDTRY.
```

- Untuk detail API, lihat DeleteCertificatedi AWS SDK untuk referensi SAP ABAP API.

`DeleteThing`

Contoh kode berikut menunjukkan cara menggunakan`DeleteThing`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    CONSTANTS cv_pfl TYPE /aws1/rt_profile_id VALUE 'ZCODE_DEMO'.
    DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
    DATA(lo_iot) = /aws1/cl_iot_factory=>create( lo_session ).
    TRY.
        lo_iot->deletething( iv_thingname = iv_thing_name ).
        MESSAGE |IoT thing deleted: { iv_thing_name }| TYPE 'I'.
      CATCH /aws1/cx_iotresourcenotfoundex INTO DATA(lo_ex).
        MESSAGE |IoT thing '{ iv_thing_name }' not found.| TYPE 'I'.
        RAISE EXCEPTION lo_ex.
```

```
  CATCH /aws1/cx_rt_service_generic INTO DATA(lo_svc_ex).
    MESSAGE lo_svc_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_svc_ex.
ENDTRY.
```

- Untuk detail API, lihat DeleteThingdi AWS SDK untuk referensi SAP ABAP API.

`DeleteTopicRule`

Contoh kode berikut menunjukkan cara menggunakan`DeleteTopicRule`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
CONSTANTS cv_pfl TYPE /aws1/rt_profile_id VALUE 'ZCODE_DEMO'.
DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
DATA(lo_iot) = /aws1/cl_iot_factory=>create( lo_session ).
TRY.
    lo_iot->deletetopicrule( iv_rulename = iv_rule_name ).
    MESSAGE |IoT topic rule deleted: { iv_rule_name }| TYPE 'I'.
  CATCH /aws1/cx_rt_service_generic INTO DATA(lo_ex).
    MESSAGE lo_ex->get_text( ) TYPE 'I'.
    RAISE EXCEPTION lo_ex.
ENDTRY.
```

- Untuk detail API, lihat DeleteTopicRuledi AWS SDK untuk referensi SAP ABAP API.

`DescribeEndpoint`

Contoh kode berikut menunjukkan cara menggunakan`DescribeEndpoint`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   CONSTANTS cv_pfl TYPE /aws1/rt_profile_id VALUE 'ZCODE_DEMO'.
   DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
   DATA(lo_iot) = /aws1/cl_iot_factory=>create( lo_session ).
   " iv_endpoint_type = 'iot:Data-ATS' - Endpoint type for data operations
   TRY.
       DATA(lo_result) = lo_iot->describeendpoint( iv_endpointtype =
iv_endpoint_type ).
       ov_endpoint_address = lo_result->get_endpointaddress( ).
       MESSAGE |Endpoint address: { ov_endpoint_address }| TYPE 'I'.
     CATCH /aws1/cx_iotthrottlingex INTO DATA(lo_throttle_ex).
       MESSAGE 'Request throttled. Please try again later.' TYPE 'I'.
       RAISE EXCEPTION lo_throttle_ex.
     CATCH /aws1/cx_rt_service_generic INTO DATA(lo_ex).
       MESSAGE lo_ex->get_text( ) TYPE 'I'.
       RAISE EXCEPTION lo_ex.
   ENDTRY.
```

- Untuk detail API, lihat DescribeEndpointdi AWS SDK untuk referensi SAP ABAP API.

`DetachThingPrincipal`

Contoh kode berikut menunjukkan cara menggunakan`DetachThingPrincipal`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    CONSTANTS cv_pfl TYPE /aws1/rt_profile_id VALUE 'ZCODE_DEMO'.
    DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
    DATA(lo_iot) = /aws1/cl_iot_factory=>create( lo_session ).
    TRY.
        lo_iot->detachthingprincipal(
          iv_thingname = iv_thing_name
          iv_principal = iv_principal ).
        MESSAGE |Principal detached from IoT thing '{ iv_thing_name }'.| TYPE 'I'.
      CATCH /aws1/cx_iotresourcenotfoundex INTO DATA(lo_ex).
        MESSAGE |Resource not found when detaching principal from
'{ iv_thing_name }'.| TYPE 'I'.
        RAISE EXCEPTION lo_ex.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_svc_ex).
        MESSAGE lo_svc_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_svc_ex.
    ENDTRY.
```

- Untuk detail API, lihat DetachThingPrincipaldi AWS SDK untuk referensi SAP ABAP API.

`ListCertificates`

Contoh kode berikut menunjukkan cara menggunakan`ListCertificates`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    CONSTANTS cv_pfl TYPE /aws1/rt_profile_id VALUE 'ZCODE_DEMO'.
    DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
    DATA(lo_iot) = /aws1/cl_iot_factory=>create( lo_session ).
    TRY.
        " Collect all certificates by following the pagination marker.
        DATA lv_marker TYPE /aws1/iotmarker.
        DATA lv_count  TYPE i.

        DO.
```

```
        oo_result = lo_iot->listcertificates( iv_marker = lv_marker ).
        lv_count  = lv_count + lines( oo_result->get_certificates( ) ).
        lv_marker = oo_result->get_nextmarker( ).
        IF lv_marker IS INITIAL.
          EXIT.
        ENDIF.
      ENDDO.

      MESSAGE |Retrieved { lv_count } IoT certificates.| TYPE 'I'.
    CATCH /aws1/cx_iotthrottlingex INTO DATA(lo_throttle_ex).
      MESSAGE 'Request throttled. Please try again later.' TYPE 'I'.
      RAISE EXCEPTION lo_throttle_ex.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_ex).
      MESSAGE lo_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_ex.
  ENDTRY.
```

- Untuk detail API, lihat ListCertificatesdi AWS SDK untuk referensi SAP ABAP API.

`ListThings`

Contoh kode berikut menunjukkan cara menggunakan`ListThings`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
  CONSTANTS cv_pfl TYPE /aws1/rt_profile_id VALUE 'ZCODE_DEMO'.
  DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
  DATA(lo_iot) = /aws1/cl_iot_factory=>create( lo_session ).
  TRY.
      " Collect all things by following the pagination token.
      DATA lv_nexttoken TYPE /aws1/iotnexttoken.
      DATA lv_count     TYPE i.

      DO.
```

```
        oo_result    = lo_iot->listthings( iv_nexttoken = lv_nexttoken ).
        lv_count     = lv_count + lines( oo_result->get_things( ) ).
        lv_nexttoken = oo_result->get_nexttoken( ).
        IF lv_nexttoken IS INITIAL.
          EXIT.
        ENDIF.
      ENDDO.

      MESSAGE |Retrieved { lv_count } IoT things.| TYPE 'I'.
    CATCH /aws1/cx_iotthrottlingex INTO DATA(lo_throttle_ex).
      MESSAGE 'Request throttled. Please try again later.' TYPE 'I'.
      RAISE EXCEPTION lo_throttle_ex.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_ex).
      MESSAGE lo_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_ex.
  ENDTRY.
```

- Untuk detail API, lihat ListThingsdi AWS SDK untuk referensi SAP ABAP API.

`SearchIndex`

Contoh kode berikut menunjukkan cara menggunakan`SearchIndex`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
  CONSTANTS cv_pfl TYPE /aws1/rt_profile_id VALUE 'ZCODE_DEMO'.
  DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
  DATA(lo_iot) = /aws1/cl_iot_factory=>create( lo_session ).
  " iv_query_string = 'thingName:MyThing*' - Fleet indexing query string
  TRY.
      oo_result = lo_iot->searchindex( iv_querystring = iv_query_string ).
      MESSAGE |Found { lines( oo_result->get_things( ) ) } IoT things matching
query.| TYPE 'I'.
    CATCH /aws1/cx_iotindexnotreadyex INTO DATA(lo_idx_ex).
```

```
        MESSAGE 'Fleet indexing is not ready. Enable indexing with
UpdateIndexingConfiguration first.' TYPE 'I'.
        RAISE EXCEPTION lo_idx_ex.
      CATCH /aws1/cx_iotthrottlingex INTO DATA(lo_throttle_ex).
        MESSAGE 'Request throttled. Please try again later.' TYPE 'I'.
        RAISE EXCEPTION lo_throttle_ex.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_ex).
        MESSAGE lo_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_ex.
    ENDTRY.
```

- Untuk detail API, lihat SearchIndexdi AWS SDK untuk referensi SAP ABAP API.

`UpdateIndexingConfiguration`

Contoh kode berikut menunjukkan cara menggunakan`UpdateIndexingConfiguration`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    CONSTANTS cv_pfl TYPE /aws1/rt_profile_id VALUE 'ZCODE_DEMO'.
    DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
    DATA(lo_iot) = /aws1/cl_iot_factory=>create( lo_session ).
    TRY.
        DATA lo_idx_conf TYPE REF TO /aws1/cl_iotthingindexingconf.
        CREATE OBJECT lo_idx_conf
          EXPORTING
            iv_thingindexingmode = 'REGISTRY'.
        lo_iot->updateindexingconfiguration(
          io_thingindexingconf = lo_idx_conf ).
        MESSAGE 'IoT thing indexing configuration updated to REGISTRY mode.' TYPE
'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_ex).
        MESSAGE lo_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_ex.
```

```
    ENDTRY.
```

- Untuk detail API, lihat UpdateIndexingConfigurationdi AWS SDK untuk referensi SAP ABAP API.

AWS IoT data contoh menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan mengimplementasikan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan. AWS IoT data

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`GetThingShadow`

Contoh kode berikut menunjukkan cara menggunakan`GetThingShadow`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        DATA(lo_result) = lo_iop->getthingshadow( iv_thingname = iv_thing_name ).
```

```
    " Convert xstring payload to JSON string
    DATA(lv_payload) = lo_result->get_payload( ).
    ov_shadow = /aws1/cl_rt_util=>xstring_to_string( lv_payload ).
    MESSAGE 'Thing shadow retrieved successfully.' TYPE 'I'.
  CATCH /aws1/cx_iopresourcenotfoundex.
    MESSAGE 'Thing shadow not found.' TYPE 'E'.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_exception).
    DATA(lv_error) = |{ lo_exception->get_text( ) }|.
    MESSAGE lv_error TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat GetThingShadowdi AWS SDK untuk referensi SAP ABAP API.

`UpdateThingShadow`

Contoh kode berikut menunjukkan cara menggunakan`UpdateThingShadow`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " Convert JSON string to xstring for payload
    DATA(lv_payload) = /aws1/cl_rt_util=>string_to_xstring( iv_shadow_state ).

    lo_iop->updatethingshadow(
      iv_thingname = iv_thing_name
      iv_payload = lv_payload ).
    MESSAGE 'Thing shadow updated successfully.' TYPE 'I'.
  CATCH /aws1/cx_iopresourcenotfoundex.
    MESSAGE 'Thing not found.' TYPE 'E'.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_exception).
    DATA(lv_error) = |{ lo_exception->get_text( ) }|.
    MESSAGE lv_error TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat UpdateThingShadowdi AWS SDK untuk referensi SAP ABAP API.

AWS IoT SiteWise contoh menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan mengimplementasikan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan. AWS IoT SiteWise

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Memulai
- Tindakan

Memulai

Halo AWS IoT SiteWise

Contoh kode berikut menunjukkan bagaimana untuk mulai menggunakan AWS IoT SiteWise.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_ios->listassetmodels( ). " oo_result is returned for testing
purposes. "
```

```
    DATA(lt_asset_models) = oo_result->get_assetmodelsummaries( ).
    MESSAGE 'Retrieved list of asset models.' TYPE 'I'.
  CATCH /aws1/cx_rt_generic.
    MESSAGE 'Unable to list asset models.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat ListAssetModelsdi AWS SDK untuk referensi SAP ABAP API.

Tindakan

`BatchPutAssetPropertyValue`

Contoh kode berikut menunjukkan cara menggunakan`BatchPutAssetPropertyValue`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_ios->batchputassetpropertyvalue(
      it_entries = it_entries
    ).
    MESSAGE 'Data sent to IoT SiteWise asset successfully.' TYPE 'I'.
  CATCH /aws1/cx_iosresourcenotfoundex.
    MESSAGE 'Asset does not exist.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat BatchPutAssetPropertyValuedi AWS SDK untuk referensi SAP ABAP API.

`CreateAsset`

Contoh kode berikut menunjukkan cara menggunakan`CreateAsset`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_ios->createasset(
      iv_assetname = iv_asset_name
      iv_assetmodelid = iv_asset_model_id
    ). " oo_result is returned for testing purposes. "
    MESSAGE 'IoT SiteWise asset created' TYPE 'I'.
  CATCH /aws1/cx_iosresourcenotfoundex.
    MESSAGE 'Asset model does not exist.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat CreateAssetdi AWS SDK untuk referensi SAP ABAP API.

`CreateAssetModel`

Contoh kode berikut menunjukkan cara menggunakan`CreateAssetModel`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_ios->createassetmodel(
      iv_assetmodelname = iv_asset_model_name
      iv_assetmodeldescription = 'This is a sample asset model description.'
      it_assetmodelproperties = it_properties
    ). " oo_result is returned for testing purposes. "
```

```
    MESSAGE 'IoT SiteWise asset model created' TYPE 'I'.
  CATCH /aws1/cx_iosresrcalrdyexistsex.
    MESSAGE 'Asset model already exists.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat CreateAssetModeldi AWS SDK untuk referensi SAP ABAP API.

`CreateGateway`

Contoh kode berikut menunjukkan cara menggunakan`CreateGateway`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_ios->creategateway(
      iv_gatewayname = iv_gateway_name
      io_gatewayplatform = NEW /aws1/cl_iosgatewayplatform(
        io_greengrassv2 = NEW /aws1/cl_iosgreengrassv2(
          iv_coredevicethingname = iv_core_device_thing_name
        )
      )
      it_tags = VALUE /aws1/cl_iostagmap_w=>tt_tagmap(
        (
          VALUE /aws1/cl_iostagmap_w=>ts_tagmap_maprow(
            key = 'Environment'
            value = NEW /aws1/cl_iostagmap_w( 'Production' )
          )
        )
      )
    ). " oo_result is returned for testing purposes. "
    MESSAGE 'IoT SiteWise gateway created' TYPE 'I'.
  CATCH /aws1/cx_iosresrcalrdyexistsex.
    MESSAGE 'Gateway already exists.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat CreateGatewaydi AWS SDK untuk referensi SAP ABAP API.

`DeleteAsset`

Contoh kode berikut menunjukkan cara menggunakan`DeleteAsset`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_ios->deleteasset(
      iv_assetid = iv_asset_id
    ).
    MESSAGE 'IoT SiteWise asset deleted.' TYPE 'I'.
  CATCH /aws1/cx_rt_generic.
    MESSAGE 'Unable to delete asset.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DeleteAssetdi AWS SDK untuk referensi SAP ABAP API.

`DeleteAssetModel`

Contoh kode berikut menunjukkan cara menggunakan`DeleteAssetModel`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        lo_ios->deleteassetmodel(
          iv_assetmodelid = iv_asset_model_id
        ).
        MESSAGE 'IoT SiteWise asset model deleted.' TYPE 'I'.
      CATCH /aws1/cx_rt_generic.
        MESSAGE 'Unable to delete asset model.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat DeleteAssetModeldi AWS SDK untuk referensi SAP ABAP API.

`DeleteGateway`

Contoh kode berikut menunjukkan cara menggunakan`DeleteGateway`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        lo_ios->deletegateway(
          iv_gatewayid = iv_gateway_id
        ).
        MESSAGE 'IoT SiteWise gateway deleted.' TYPE 'I'.
      CATCH /aws1/cx_iosresourcenotfoundex.
        MESSAGE 'Gateway does not exist.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat DeleteGatewaydi AWS SDK untuk referensi SAP ABAP API.

`DescribeGateway`

Contoh kode berikut menunjukkan cara menggunakan`DescribeGateway`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_ios->describegateway(
      iv_gatewayid = iv_gateway_id
    ). " oo_result is returned for testing purposes. "
    MESSAGE 'Retrieved gateway description.' TYPE 'I'.
  CATCH /aws1/cx_iosresourcenotfoundex.
    MESSAGE 'Gateway does not exist.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DescribeGatewaydi AWS SDK untuk referensi SAP ABAP API.

`GetAssetPropertyValue`

Contoh kode berikut menunjukkan cara menggunakan`GetAssetPropertyValue`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_ios->getassetpropertyvalue(
      iv_assetid = iv_asset_id
      iv_propertyid = iv_property_id
    ). " oo_result is returned for testing purposes. "
    MESSAGE 'Retrieved asset property value.' TYPE 'I'.
  CATCH /aws1/cx_iosresourcenotfoundex.
    MESSAGE 'Asset or property does not exist.' TYPE 'E'.
```

```
    ENDTRY.
```

- Untuk detail API, lihat GetAssetPropertyValuedi AWS SDK untuk referensi SAP ABAP API.

`ListAssetModelProperties`

Contoh kode berikut menunjukkan cara menggunakan`ListAssetModelProperties`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_ios->listassetmodelproperties(
      iv_assetmodelid = iv_asset_model_id
    ). " oo_result is returned for testing purposes. "
    DATA(lt_properties) = oo_result->get_assetmodelpropertysums( ).
    MESSAGE 'Retrieved list of asset model properties.' TYPE 'I'.
  CATCH /aws1/cx_rt_generic.
    MESSAGE 'Unable to list asset model properties.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat ListAssetModelPropertiesdi AWS SDK untuk referensi SAP ABAP API.

`ListAssetModels`

Contoh kode berikut menunjukkan cara menggunakan`ListAssetModels`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_ios->listassetmodels( ). " oo_result is returned for testing
purposes. "
        DATA(lt_asset_models) = oo_result->get_assetmodelsummaries( ).
        MESSAGE 'Retrieved list of asset models.' TYPE 'I'.
      CATCH /aws1/cx_rt_generic.
        MESSAGE 'Unable to list asset models.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat ListAssetModelsdi AWS SDK untuk referensi SAP ABAP API.

Contoh Amazon Keyspaces menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Amazon Keyspaces.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`CreateKeyspace`

Contoh kode berikut menunjukkan cara menggunakan`CreateKeyspace`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_kys->createkeyspace(
          iv_keyspacename = iv_keyspace_name ).
        MESSAGE 'Keyspace created successfully.' TYPE 'I'.
      CATCH /aws1/cx_kysconflictexception.
        MESSAGE 'Keyspace already exists.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat CreateKeyspacedi AWS SDK untuk referensi SAP ABAP API.

`CreateTable`

Contoh kode berikut menunjukkan cara menggunakan`CreateTable`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " Define schema with columns
        DATA(lt_columns) = VALUE /aws1/
cl_kyscolumndefinition=>tt_columndefinitionlist(
          ( NEW /aws1/cl_kyscolumndefinition( iv_name = 'title' iv_type = 'text' ) )
          ( NEW /aws1/cl_kyscolumndefinition( iv_name = 'year' iv_type = 'int' ) )
          ( NEW /aws1/cl_kyscolumndefinition( iv_name = 'release_date' iv_type =
 'timestamp' ) )
          ( NEW /aws1/cl_kyscolumndefinition( iv_name = 'plot' iv_type = 'text' ) )
        ).

        " Define partition keys
        DATA(lt_partition_keys) = VALUE /aws1/
cl_kyspartitionkey=>tt_partitionkeylist(
```

```
          ( NEW /aws1/cl_kyspartitionkey( iv_name = 'year' ) )
          ( NEW /aws1/cl_kyspartitionkey( iv_name = 'title' ) )
        ).

        " Create schema definition
        DATA(lo_schema) = NEW /aws1/cl_kysschemadefinition(
          it_allcolumns = lt_columns
          it_partitionkeys = lt_partition_keys ).

        " Enable point-in-time recovery
        DATA(lo_pitr) = NEW /aws1/cl_kyspointintimerec(
          iv_status = 'ENABLED' ).

        oo_result = lo_kys->createtable(
          iv_keyspacename = iv_keyspace_name
          iv_tablename = iv_table_name
          io_schemadefinition = lo_schema
          io_pointintimerecovery = lo_pitr ).
        MESSAGE 'Table created successfully.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat CreateTabledi AWS SDK untuk referensi SAP ABAP API.

`DeleteKeyspace`

Contoh kode berikut menunjukkan cara menggunakan`DeleteKeyspace`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        lo_kys->deletekeyspace(
```

```
          iv_keyspacename = iv_keyspace_name ).
        MESSAGE 'Keyspace deleted successfully.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat DeleteKeyspacedi AWS SDK untuk referensi SAP ABAP API.

`DeleteTable`

Contoh kode berikut menunjukkan cara menggunakan`DeleteTable`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        lo_kys->deletetable(
          iv_keyspacename = iv_keyspace_name
          iv_tablename = iv_table_name ).
        MESSAGE 'Table deleted successfully.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat DeleteTabledi AWS SDK untuk referensi SAP ABAP API.

`GetKeyspace`

Contoh kode berikut menunjukkan cara menggunakan`GetKeyspace`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_kys->getkeyspace(
          iv_keyspacename = iv_keyspace_name ).
        MESSAGE 'Keyspace retrieved successfully.' TYPE 'I'.
      CATCH /aws1/cx_kysresourcenotfoundex.
        MESSAGE 'Keyspace does not exist.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat GetKeyspacedi AWS SDK untuk referensi SAP ABAP API.

`GetTable`

Contoh kode berikut menunjukkan cara menggunakan`GetTable`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_kys->gettable(
          iv_keyspacename = iv_keyspace_name
          iv_tablename = iv_table_name ).
```

```
        MESSAGE 'Table information retrieved successfully.' TYPE 'I'.
      CATCH /aws1/cx_kysresourcenotfoundex.
        MESSAGE 'Table does not exist.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat GetTabledi AWS SDK untuk referensi SAP ABAP API.

`ListKeyspaces`

Contoh kode berikut menunjukkan cara menggunakan`ListKeyspaces`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_kys->listkeyspaces(
          iv_maxresults = iv_max_results ).
        MESSAGE 'Keyspaces listed successfully.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat ListKeyspacesdi AWS SDK untuk referensi SAP ABAP API.

`ListTables`

Contoh kode berikut menunjukkan cara menggunakan`ListTables`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_kys->listtables(
          iv_keyspacename = iv_keyspace_name ).
        MESSAGE 'Tables listed successfully.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat ListTablesdi AWS SDK untuk referensi SAP ABAP API.

`RestoreTable`

Contoh kode berikut menunjukkan cara menggunakan`RestoreTable`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_kys->restoretable(
          iv_sourcekeyspacename = iv_source_keyspace_name
          iv_sourcetablename = iv_source_table_name
          iv_targetkeyspacename = iv_target_keyspace_name
          iv_targettablename = iv_target_table_name
          iv_restoretimestamp = iv_restore_timestamp ).
```

```
        MESSAGE 'Table restore initiated successfully.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat RestoreTabledi AWS SDK untuk referensi SAP ABAP API.

`UpdateTable`

Contoh kode berikut menunjukkan cara menggunakan`UpdateTable`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " Add a new column to track watched movies
        DATA(lt_add_columns) = VALUE /aws1/
cl_kyscolumndefinition=>tt_columndefinitionlist(
          ( NEW /aws1/cl_kyscolumndefinition( iv_name = 'watched' iv_type =
 'boolean' ) )
        ).

        oo_result = lo_kys->updatetable(
          iv_keyspacename = iv_keyspace_name
          iv_tablename = iv_table_name
          it_addcolumns = lt_add_columns ).
        MESSAGE 'Table updated successfully.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat UpdateTabledi AWS SDK untuk referensi SAP ABAP API.

Contoh Kinesis menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan mengimplementasikan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Kinesis.

Dasar-dasar adalah contoh kode yang menunjukkan kepada Anda bagaimana melakukan operasi penting dalam suatu layanan.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Hal-hal mendasar
- Tindakan

Hal-hal mendasar

Pelajari dasar-dasarnya

Contoh kode berikut ini menunjukkan cara untuk melakukan:

- Buat aliran dan letakkan catatan di dalamnya.
- Buat iterator pecahan.
- Baca catatan, lalu bersihkan sumber daya.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   DATA lo_stream_describe_result TYPE REF TO /aws1/cl_knsdescrstreamoutput.
   DATA lo_stream_description TYPE REF TO /aws1/cl_knsstreamdescription.
   DATA lo_sharditerator TYPE REF TO /aws1/cl_knsgetsharditerator01.
   DATA lo_record_result TYPE REF TO /aws1/cl_knsputrecordoutput.

   "Create stream."
   TRY.
       lo_kns->createstream(
           iv_streamname = iv_stream_name
           iv_shardcount = iv_shard_count ).
       MESSAGE 'Stream created.' TYPE 'I'.
     CATCH /aws1/cx_knsinvalidargumentex.
       MESSAGE 'The specified argument was not valid.' TYPE 'E'.
     CATCH /aws1/cx_knslimitexceededex.
       MESSAGE 'The request processing has failed because of a limit exceeded
exception.' TYPE 'E'.
     CATCH /aws1/cx_knsresourceinuseex.
       MESSAGE 'The request processing has failed because the resource is in use.'
TYPE 'E'.
   ENDTRY.

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

   "Create record."
   TRY.
       lo_record_result = lo_kns->putrecord(
           iv_streamname = iv_stream_name
           iv_data       = iv_data
           iv_partitionkey = iv_partition_key ).
       MESSAGE 'Record created.' TYPE 'I'.
```

```
     CATCH /aws1/cx_knsinvalidargumentex.
       MESSAGE 'The specified argument was not valid.' TYPE 'E'.
     CATCH /aws1/cx_knskmsaccessdeniedex.
       MESSAGE 'You do not have permission to perform this AWS KMS action.' TYPE
'E'.
     CATCH /aws1/cx_knskmsdisabledex.
       MESSAGE 'KMS key used is disabled.' TYPE 'E'.
     CATCH /aws1/cx_knskmsinvalidstateex.
       MESSAGE 'KMS key used is in an invalid state. ' TYPE 'E'.
     CATCH /aws1/cx_knskmsnotfoundex.
       MESSAGE 'KMS key used is not found.' TYPE 'E'.
     CATCH /aws1/cx_knskmsoptinrequired.
       MESSAGE 'KMS key option is required.' TYPE 'E'.
     CATCH /aws1/cx_knskmsthrottlingex.
       MESSAGE 'The rate of requests to AWS KMS is exceeding the request quotas.'
TYPE 'E'.
     CATCH /aws1/cx_knsprovthruputexcdex.
       MESSAGE 'The request rate for the stream is too high, or the requested data
is too large for the available throughput.' TYPE 'E'.
     CATCH /aws1/cx_knsresourcenotfoundex.
       MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
   ENDTRY.

   "Create a shard iterator in order to read the record."
   TRY.
       lo_sharditerator = lo_kns->getsharditerator(
         iv_shardid = lo_record_result->get_shardid( )
         iv_sharditeratortype = iv_sharditeratortype
         iv_streamname = iv_stream_name ).
       MESSAGE 'Shard iterator created.' TYPE 'I'.
     CATCH /aws1/cx_knsinvalidargumentex.
       MESSAGE 'The specified argument was not valid.' TYPE 'E'.
     CATCH /aws1/cx_knsprovthruputexcdex.
       MESSAGE 'The request rate for the stream is too high, or the requested data
is too large for the available throughput.' TYPE 'E'.
     CATCH /aws1/cx_sgmresourcenotfound.
       MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
   ENDTRY.

   "Read the record."
   TRY.
       oo_result = lo_kns->getrecords(                    " oo_result is returned
for testing purposes. "
           iv_sharditerator   = lo_sharditerator->get_sharditerator( ) ).
```

```
        MESSAGE 'Shard iterator created.' TYPE 'I'.
      CATCH /aws1/cx_knsexpirediteratorex.
        MESSAGE 'Iterator expired.' TYPE 'E'.
      CATCH /aws1/cx_knsinvalidargumentex.
        MESSAGE 'The specified argument was not valid.' TYPE 'E'.
      CATCH /aws1/cx_knskmsaccessdeniedex.
        MESSAGE 'You do not have permission to perform this AWS KMS action.' TYPE
'E'.
      CATCH /aws1/cx_knskmsdisabledex.
        MESSAGE 'KMS key used is disabled.' TYPE 'E'.
      CATCH /aws1/cx_knskmsinvalidstateex.
        MESSAGE 'KMS key used is in an invalid state. ' TYPE 'E'.
      CATCH /aws1/cx_knskmsnotfoundex.
        MESSAGE 'KMS key used is not found.' TYPE 'E'.
      CATCH /aws1/cx_knskmsoptinrequired.
        MESSAGE 'KMS key option is required.' TYPE 'E'.
      CATCH /aws1/cx_knskmsthrottlingex.
        MESSAGE 'The rate of requests to AWS KMS is exceeding the request quotas.'
TYPE 'E'.
      CATCH /aws1/cx_knsprovthruputexcdex.
        MESSAGE 'The request rate for the stream is too high, or the requested data
is too large for the available throughput.' TYPE 'E'.
      CATCH /aws1/cx_knsresourcenotfoundex.
        MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
    ENDTRY.

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

- Untuk mengetahui hal detail mengenai API, silakan lihat topik-topik berikut di referensi API AWS SDK untuk ABAP SAP.
  - CreateStream

- DeleteStream
- GetRecords
- GetShardIterator
- PutRecord

Tindakan

`CreateStream`

Contoh kode berikut menunjukkan cara menggunakan`CreateStream`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   TRY.
       lo_kns->createstream(
           iv_streamname = iv_stream_name
           iv_shardcount = iv_shard_count ).
       MESSAGE 'Stream created.' TYPE 'I'.
     CATCH /aws1/cx_knsinvalidargumentex.
       MESSAGE 'The specified argument was not valid.' TYPE 'E'.
     CATCH /aws1/cx_knslimitexceededex.
       MESSAGE 'The request processing has failed because of a limit exceed
exception.' TYPE 'E'.
     CATCH /aws1/cx_knsresourceinuseex.
       MESSAGE 'The request processing has failed because the resource is in use.'
TYPE 'E'.
   ENDTRY.
```

- Untuk detail API, lihat CreateStreamdi AWS SDK untuk referensi SAP ABAP API.

`DeleteStream`

Contoh kode berikut menunjukkan cara menggunakan`DeleteStream`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat DeleteStreamdi AWS SDK untuk referensi SAP ABAP API.

`DescribeStream`

Contoh kode berikut menunjukkan cara menggunakan`DescribeStream`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   TRY.
```

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

- Untuk detail API, lihat DescribeStreamdi AWS SDK untuk referensi SAP ABAP API.

`GetRecords`

Contoh kode berikut menunjukkan cara menggunakan`GetRecords`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_kns->getrecords(             " oo_result is returned for
testing purposes. "
            iv_shard_iterator = iv_shard_iterator ).
        DATA(lt_records) = oo_result->get_records( ).
        MESSAGE 'Record retrieved.' TYPE 'I'.
      CATCH /aws1/cx_knsexpirediteratorex.
        MESSAGE 'Iterator expired.' TYPE 'E'.
      CATCH /aws1/cx_knsinvalidargumentex.
        MESSAGE 'The specified argument was not valid.' TYPE 'E'.
      CATCH /aws1/cx_knskmsaccessdeniedex.
        MESSAGE 'You do not have permission to perform this AWS KMS action.' TYPE
'E'.
      CATCH /aws1/cx_knskmsdisabledex.
        MESSAGE 'KMS key used is disabled.' TYPE 'E'.
```

```
      CATCH /aws1/cx_knskmsinvalidstateex.
        MESSAGE 'KMS key used is in an invalid state. ' TYPE 'E'.
      CATCH /aws1/cx_knskmsnotfoundex.
        MESSAGE 'KMS key used is not found.' TYPE 'E'.
      CATCH /aws1/cx_knskmsoptinrequired.
        MESSAGE 'KMS key option is required.' TYPE 'E'.
      CATCH /aws1/cx_knskmsthrottlingex.
        MESSAGE 'The rate of requests to AWS KMS is exceeding the request quotas.'
TYPE 'E'.
      CATCH /aws1/cx_knsprovthruputexcdex.
        MESSAGE 'The request rate for the stream is too high, or the requested data
is too large for the available throughput.' TYPE 'E'.
      CATCH /aws1/cx_knsresourcenotfoundex.
        MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat GetRecordsdi AWS SDK untuk referensi SAP ABAP API.

`ListStreams`

Contoh kode berikut menunjukkan cara menggunakan`ListStreams`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

```
    ENDTRY.
```

- Untuk detail API, lihat ListStreamsdi AWS SDK untuk referensi SAP ABAP API.

`PutRecord`

Contoh kode berikut menunjukkan cara menggunakan`PutRecord`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   TRY.
       oo_result = lo_kns->putrecord(            " oo_result is returned for
testing purposes. "
           iv_streamname = iv_stream_name
           iv_data       = iv_data
           iv_partitionkey = iv_partition_key ).
       MESSAGE 'Record created.' TYPE 'I'.
     CATCH /aws1/cx_knsinvalidargumentex.
       MESSAGE 'The specified argument was not valid.' TYPE 'E'.
     CATCH /aws1/cx_knskmsaccessdeniedex.
       MESSAGE 'You do not have permission to perform this AWS KMS action.' TYPE
'E'.
     CATCH /aws1/cx_knskmsdisabledex.
       MESSAGE 'KMS key used is disabled.' TYPE 'E'.
     CATCH /aws1/cx_knskmsinvalidstateex.
       MESSAGE 'KMS key used is in an invalid state. ' TYPE 'E'.
     CATCH /aws1/cx_knskmsnotfoundex.
       MESSAGE 'KMS key used is not found.' TYPE 'E'.
     CATCH /aws1/cx_knskmsoptinrequired.
       MESSAGE 'KMS key option is required.' TYPE 'E'.
     CATCH /aws1/cx_knskmsthrottlingex.
       MESSAGE 'The rate of requests to AWS KMS is exceeding the request quotas.'
TYPE 'E'.
     CATCH /aws1/cx_knsprovthruputexcdex.
```

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

- Untuk detail API, lihat PutRecorddi AWS SDK untuk referensi SAP ABAP API.

`RegisterStreamConsumer`

Contoh kode berikut menunjukkan cara menggunakan`RegisterStreamConsumer`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_kns->registerstreamconsumer(       " oo_result is returned
for testing purposes. "
            iv_streamarn = iv_stream_arn
            iv_consumername = iv_consumer_name ).
        MESSAGE 'Stream consumer registered.' TYPE 'I'.
      CATCH /aws1/cx_knsinvalidargumentex.
        MESSAGE 'The specified argument was not valid.' TYPE 'E'.
      CATCH /aws1/cx_sgmresourcelimitexcd.
        MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
      CATCH /aws1/cx_sgmresourceinuse.
        MESSAGE 'Resource being accessed is in use.' TYPE 'E'.
      CATCH /aws1/cx_sgmresourcenotfound.
        MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat RegisterStreamConsumerdi AWS SDK untuk referensi SAP ABAP API.

AWS KMS contoh menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan mengimplementasikan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan. AWS KMS

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Memulai
- Tindakan

Memulai

Halo AWS KMS

Contoh kode berikut menunjukkan bagaimana untuk mulai menggunakan AWS Key Management Service.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_kms->listkeys( ).
    MESSAGE 'Retrieved KMS keys list.' TYPE 'I'.
  CATCH /aws1/cx_kmskmsinternalex.
    MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat ListKeysdi AWS SDK untuk referensi SAP ABAP API.

Tindakan

`CreateAlias`

Contoh kode berikut menunjukkan cara menggunakan`CreateAlias`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_alias_name = 'alias/my-key-alias'
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        lo_kms->createalias(
          iv_aliasname = iv_alias_name
          iv_targetkeyid = iv_key_id
        ).
        MESSAGE 'Alias created successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsalreadyexistsex.
        MESSAGE 'Alias already exists.' TYPE 'E'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmsinvalidaliasnameex.
        MESSAGE 'Invalid alias name.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat CreateAliasdi AWS SDK untuk referensi SAP ABAP API.

`CreateGrant`

Contoh kode berikut menunjukkan cara menggunakan`CreateGrant`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        " iv_grantee_principal = 'arn:aws:iam::123456789012:role/my-role'
        " it_operations contains 'Encrypt', 'Decrypt', 'GenerateDataKey'
        oo_result = lo_kms->creategrant(
          iv_keyid = iv_key_id
          iv_granteeprincipal = iv_grantee_principal
          it_operations = it_operations
        ).
        MESSAGE 'Grant created successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsdisabledexception.
        MESSAGE 'The key is disabled.' TYPE 'E'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat CreateGrantdi AWS SDK untuk referensi SAP ABAP API.

`CreateKey`

Contoh kode berikut menunjukkan cara menggunakan`CreateKey`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_description = 'Created by the AWS SDK for SAP ABAP'
        oo_result = lo_kms->createkey( iv_description = iv_description ).
        MESSAGE 'KMS key created successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
      CATCH /aws1/cx_kmslimitexceededex.
        MESSAGE 'Limit exceeded for KMS resources.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat CreateKeydi AWS SDK untuk referensi SAP ABAP API.

Dekripsi

Contoh kode berikut menunjukkan cara menggunakan`Decrypt`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        " iv_ciphertext_blob contains the encrypted data
        oo_result = lo_kms->decrypt(
          iv_keyid = iv_key_id
          iv_ciphertextblob = iv_ciphertext_blob
        ).
        MESSAGE 'Text decrypted successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsdisabledexception.
        MESSAGE 'The key is disabled.' TYPE 'E'.
      CATCH /aws1/cx_kmsincorrectkeyex.
        MESSAGE 'Incorrect key for decryption.' TYPE 'E'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
```

```
    MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat Mendekripsi di AWS SDK untuk referensi SAP ABAP API.

`DeleteAlias`

Contoh kode berikut menunjukkan cara menggunakan`DeleteAlias`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " iv_alias_name = 'alias/my-key-alias'
    lo_kms->deletealias( iv_aliasname = iv_alias_name ).
    MESSAGE 'Alias deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_kmsnotfoundexception.
    MESSAGE 'Alias not found.' TYPE 'E'.
  CATCH /aws1/cx_kmskmsinternalex.
    MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DeleteAliasdi AWS SDK untuk referensi SAP ABAP API.

`DescribeKey`

Contoh kode berikut menunjukkan cara menggunakan`DescribeKey`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        oo_result = lo_kms->describekey( iv_keyid = iv_key_id ).
        DATA(lo_key) = oo_result->get_keymetadata( ).
        MESSAGE 'Retrieved key information successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat DescribeKeydi AWS SDK untuk referensi SAP ABAP API.

`DisableKey`

Contoh kode berikut menunjukkan cara menggunakan`DisableKey`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        lo_kms->disablekey( iv_keyid = iv_key_id ).
```

```
        MESSAGE 'KMS key disabled successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat DisableKeydi AWS SDK untuk referensi SAP ABAP API.

`EnableKey`

Contoh kode berikut menunjukkan cara menggunakan`EnableKey`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        lo_kms->enablekey( iv_keyid = iv_key_id ).
        MESSAGE 'KMS key enabled successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat EnableKeydi AWS SDK untuk referensi SAP ABAP API.

`EnableKeyRotation`

Contoh kode berikut menunjukkan cara menggunakan`EnableKeyRotation`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        lo_kms->enablekeyrotation( iv_keyid = iv_key_id ).
        MESSAGE 'Key rotation enabled successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsdisabledexception.
        MESSAGE 'The key is disabled.' TYPE 'E'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmsunsupportedopex.
        MESSAGE 'Operation not supported for this key.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat EnableKeyRotationdi AWS SDK untuk referensi SAP ABAP API.

`Enkripsi`

Contoh kode berikut menunjukkan cara menggunakan`Encrypt`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
```

```
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        " iv_plaintext contains the data to encrypt
        oo_result = lo_kms->encrypt(
          iv_keyid = iv_key_id
          iv_plaintext = iv_plaintext
        ).
        MESSAGE 'Text encrypted successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsdisabledexception.
        MESSAGE 'The key is disabled.' TYPE 'E'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat Encrypt in AWS SDK untuk referensi SAP ABAP API.

`GenerateDataKey`

Contoh kode berikut menunjukkan cara menggunakan`GenerateDataKey`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        " iv_keyspec = 'AES_256'
        oo_result = lo_kms->generatedatakey(
          iv_keyid = iv_key_id
          iv_keyspec = 'AES_256'
        ).
        MESSAGE 'Data key generated successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsdisabledexception.
        MESSAGE 'The key is disabled.' TYPE 'E'.
```

```
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat GenerateDataKeydi AWS SDK untuk referensi SAP ABAP API.

`GetKeyPolicy`

Contoh kode berikut menunjukkan cara menggunakan`GetKeyPolicy`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        oo_result = lo_kms->getkeypolicy(
          iv_keyid = iv_key_id
          iv_policyname = 'default'
        ).
        MESSAGE 'Retrieved key policy successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat GetKeyPolicydi AWS SDK untuk referensi SAP ABAP API.

`ListAliases`

Contoh kode berikut menunjukkan cara menggunakan`ListAliases`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_kms->listaliases( ).
        MESSAGE 'Retrieved KMS aliases list.' TYPE 'I'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat ListAliasesdi AWS SDK untuk referensi SAP ABAP API.

`ListGrants`

Contoh kode berikut menunjukkan cara menggunakan`ListGrants`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        oo_result = lo_kms->listgrants( iv_keyid = iv_key_id ).
        MESSAGE 'Retrieved grants list.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat ListGrantsdi AWS SDK untuk referensi SAP ABAP API.

`ListKeyPolicies`

Contoh kode berikut menunjukkan cara menggunakan`ListKeyPolicies`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        oo_result = lo_kms->listkeypolicies( iv_keyid = iv_key_id ).
        MESSAGE 'Retrieved key policies list.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat ListKeyPoliciesdi AWS SDK untuk referensi SAP ABAP API.

`ListKeys`

Contoh kode berikut menunjukkan cara menggunakan`ListKeys`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_kms->listkeys( ).
        MESSAGE 'Retrieved KMS keys list.' TYPE 'I'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat ListKeysdi AWS SDK untuk referensi SAP ABAP API.

`PutKeyPolicy`

Contoh kode berikut menunjukkan cara menggunakan`PutKeyPolicy`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        " iv_policy = '{"Version":"2012-10-17",        "Statement": [...]}'
        lo_kms->putkeypolicy(
          iv_keyid = iv_key_id
          iv_policyname = 'default'
          iv_policy = iv_policy
        ).
        MESSAGE 'Key policy updated successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmsmalformedplydocex.
        MESSAGE 'Malformed policy document.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat PutKeyPolicydi AWS SDK untuk referensi SAP ABAP API.

`ReEncrypt`

Contoh kode berikut menunjukkan cara menggunakan`ReEncrypt`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_source_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        " iv_destination_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/5678dcba-56cd-78ef-90ab-5678901234cd'
        " iv_ciphertext_blob contains the encrypted data
        oo_result = lo_kms->reencrypt(
          iv_sourcekeyid = iv_source_key_id
          iv_destinationkeyid = iv_destination_key_id
          iv_ciphertextblob = iv_ciphertext_blob
        ).
        MESSAGE 'Ciphertext reencrypted successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsdisabledexception.
        MESSAGE 'The key is disabled.' TYPE 'E'.
      CATCH /aws1/cx_kmsincorrectkeyex.
        MESSAGE 'Incorrect source key for decryption.' TYPE 'E'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat ReEncryptdi AWS SDK untuk referensi SAP ABAP API.

`RetireGrant`

Contoh kode berikut menunjukkan cara menggunakan`RetireGrant`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " iv_grant_token = 'AQpAM2RhZ...'
    lo_kms->retiregrant( iv_granttoken = iv_grant_token ).
    MESSAGE 'Grant retired successfully.' TYPE 'I'.
  CATCH /aws1/cx_kmsnotfoundexception.
    MESSAGE 'Grant not found.' TYPE 'E'.
  CATCH /aws1/cx_kmsinvgranttokenex.
    MESSAGE 'Invalid grant token.' TYPE 'E'.
  CATCH /aws1/cx_kmskmsinternalex.
    MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat RetireGrantdi AWS SDK untuk referensi SAP ABAP API.

`RevokeGrant`

Contoh kode berikut menunjukkan cara menggunakan`RevokeGrant`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
```

```
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        " iv_grant_id = '1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p'
        lo_kms->revokegrant(
          iv_keyid = iv_key_id
          iv_grantid = iv_grant_id
        ).
        MESSAGE 'Grant revoked successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Grant or key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmsinvalidgrantidex.
        MESSAGE 'Invalid grant ID.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat RevokeGrantdi AWS SDK untuk referensi SAP ABAP API.

`ScheduleKeyDeletion`

Contoh kode berikut menunjukkan cara menggunakan`ScheduleKeyDeletion`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        " iv_pending_window_days = 7
        oo_result = lo_kms->schedulekeydeletion(
          iv_keyid = iv_key_id
          iv_pendingwindowindays = iv_pending_window_days
        ).
        MESSAGE 'Key scheduled for deletion.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
```

```
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat ScheduleKeyDeletiondi AWS SDK untuk referensi SAP ABAP API.

`Sign`

Contoh kode berikut menunjukkan cara menggunakan`Sign`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab' (asymmetric key)
        " iv_message contains the message to sign
        " iv_signing_algorithm = 'RSASSA_PSS_SHA_256'
        oo_result = lo_kms->sign(
          iv_keyid = iv_key_id
          iv_message = iv_message
          iv_signingalgorithm = iv_signing_algorithm
        ).
        MESSAGE 'Message signed successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsdisabledexception.
        MESSAGE 'The key is disabled.' TYPE 'E'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmsinvalidkeyusageex.
        MESSAGE 'Key cannot be used for signing.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat Sign in AWS SDK untuk referensi SAP ABAP API.

`TagResource`

Contoh kode berikut menunjukkan cara menggunakan`TagResource`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    DATA lt_tags TYPE /aws1/cl_kmstag=>tt_taglist.

    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        " iv_tag_key = 'Environment'
        " iv_tag_value = 'Production'
        APPEND NEW /aws1/cl_kmstag(
          iv_tagkey = iv_tag_key
          iv_tagvalue = iv_tag_value
        ) TO lt_tags.

        lo_kms->tagresource(
          iv_keyid = iv_key_id
          it_tags = lt_tags
        ).
        MESSAGE 'Tag added to KMS key successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmstagexception.
        MESSAGE 'Invalid tag format.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat TagResourcedi AWS SDK untuk referensi SAP ABAP API.

`UpdateAlias`

Contoh kode berikut menunjukkan cara menggunakan`UpdateAlias`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_alias_name = 'alias/my-key-alias'
        " iv_target_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/5678dcba-56cd-78ef-90ab-5678901234cd'
        lo_kms->updatealias(
          iv_aliasname = iv_alias_name
          iv_targetkeyid = iv_target_key_id
        ).
        MESSAGE 'Alias updated successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Alias or key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat UpdateAliasdi AWS SDK untuk referensi SAP ABAP API.

`Verifikasi`

Contoh kode berikut menunjukkan cara menggunakan`Verify`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab' (asymmetric key)
        " iv_message contains the original message
        " iv_signature contains the signature to verify
        " iv_signing_algorithm = 'RSASSA_PSS_SHA_256'
        oo_result = lo_kms->verify(
          iv_keyid = iv_key_id
          iv_message = iv_message
          iv_signature = iv_signature
          iv_signingalgorithm = iv_signing_algorithm
        ).
        DATA(lv_valid) = oo_result->get_signaturevalid( ).
        IF lv_valid = abap_true.
          MESSAGE 'Signature is valid.' TYPE 'I'.
        ELSE.
          MESSAGE 'Signature is invalid.' TYPE 'I'.
        ENDIF.
      CATCH /aws1/cx_kmsdisabledexception.
        MESSAGE 'The key is disabled.' TYPE 'E'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinvalidsigex.
        MESSAGE 'Invalid signature.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat Verifikasi di AWS SDK untuk referensi API SAP ABAP.

Contoh Lambda menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan mengimplementasikan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Lambda.

Dasar-dasar adalah contoh kode yang menunjukkan kepada Anda bagaimana melakukan operasi penting dalam suatu layanan.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Hal-hal mendasar
- Tindakan

Hal-hal mendasar

Pelajari dasar-dasarnya

Contoh kode berikut ini menunjukkan cara untuk melakukan:

- Buat peran IAM dan fungsi Lambda, lalu unggah kode handler.
- Panggil fungsi dengan satu parameter dan dapatkan hasil.
- Perbarui kode fungsi dan konfigurasikan dengan variabel lingkungan.
- Panggil fungsi dengan parameter baru dan dapatkan hasil. Tampilkan log eksekusi yang dikembalikan.
- Buat daftar fungsi untuk akun Anda, lalu bersihkan sumber daya.

Untuk informasi selengkapnya, lihat Membuat fungsi Lambda dengan konsol.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
```

```
        "Create an AWS Identity and Access Management (IAM) role that grants AWS
 Lambda permission to write to logs."
        DATA(lv_policy_document) = `{` &&
            `"Version":"2012-10-17",       ` &&
                  `"Statement": [` &&
                    `{` &&
                      `"Effect": "Allow",` &&
                      `"Action": [` &&
                        `"sts:AssumeRole"` &&
                      `],` &&
                      `"Principal": {` &&
                        `"Service": [` &&
                          `"lambda.amazonaws.com"` &&
                        `]` &&
                      `}` &&
                    `}` &&
                  `]` &&
                `}`.
        TRY.
            DATA(lo_create_role_output) = lo_iam->createrole(
                    iv_rolename = iv_role_name
                    iv_assumerolepolicydocument = lv_policy_document
                    iv_description = 'Grant lambda permission to write to logs' ).
            DATA(lv_role_arn) = lo_create_role_output->get_role( )->get_arn( ).
            MESSAGE 'IAM role created.' TYPE 'I'.
            WAIT UP TO 10 SECONDS.            " Make sure that the IAM role is ready
 for use. "
          CATCH /aws1/cx_iamentityalrdyexex.
            DATA(lo_role) = lo_iam->getrole( iv_rolename = iv_role_name ).
            lv_role_arn = lo_role->get_role( )->get_arn( ).
          CATCH /aws1/cx_iaminvalidinputex.
            MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
          CATCH /aws1/cx_iammalformedplydocex.
            MESSAGE 'Policy document in the request is malformed.' TYPE 'E'.
        ENDTRY.

        TRY.
            lo_iam->attachrolepolicy(
                iv_rolename  = iv_role_name
                iv_policyarn = 'arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole' ).
            MESSAGE 'Attached policy to the IAM role.' TYPE 'I'.
          CATCH /aws1/cx_iaminvalidinputex.
            MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
```

```
          CATCH /aws1/cx_iamnosuchentityex.
            MESSAGE 'The requested resource entity does not exist.' TYPE 'E'.
          CATCH /aws1/cx_iamplynotattachableex.
            MESSAGE 'Service role policies can only be attached to the service-
linked role for their service.' TYPE 'E'.
          CATCH /aws1/cx_iamunmodableentityex.
            MESSAGE 'Service that depends on the service-linked role is not
 modifiable.' TYPE 'E'.
        ENDTRY.

        " Create a Lambda function and upload handler code. "
        " Lambda function performs 'increment' action on a number. "
        TRY.
            lo_lmd->createfunction(
                 iv_functionname = iv_function_name
                 iv_runtime = `python3.9`
                 iv_role = lv_role_arn
                 iv_handler = iv_handler
                 io_code = io_initial_zip_file
                 iv_description = 'AWS Lambda code example' ).
            MESSAGE 'Lambda function created.' TYPE 'I'.
          CATCH /aws1/cx_lmdcodestorageexcdex.
            MESSAGE 'Maximum total code size per account exceeded.' TYPE 'E'.
          CATCH /aws1/cx_lmdinvparamvalueex.
            MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
          CATCH /aws1/cx_lmdresourcenotfoundex.
            MESSAGE 'The requested resource does not exist.' TYPE 'E'.
        ENDTRY.

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

        "Invoke the function with a single parameter and get results."
        TRY.
            DATA(lv_json) = /aws1/cl_rt_util=>string_to_xstring(
              `{`  &&
                `"action": "increment",`  &&
                `"number": 10` &&
```

```
              `}` ).
            DATA(lo_initial_invoke_output) = lo_lmd->invoke(
                      iv_functionname = iv_function_name
                      iv_payload = lv_json ).
            ov_initial_invoke_payload = lo_initial_invoke_output->get_payload( ).
        " ov_initial_invoke_payload is returned for testing purposes. "
            DATA(lo_writer_json) = cl_sxml_string_writer=>create( type =
 if_sxml=>co_xt_json ).
            CALL TRANSFORMATION id SOURCE XML ov_initial_invoke_payload RESULT XML
 lo_writer_json.
            DATA(lv_result) = cl_abap_codepage=>convert_from( lo_writer_json-
>get_output( ) ).
            MESSAGE 'Lambda function invoked.' TYPE 'I'.
          CATCH /aws1/cx_lmdinvparamvalueex.
            MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
          CATCH /aws1/cx_lmdinvrequestcontex.
            MESSAGE 'Unable to parse request body as JSON.' TYPE 'E'.
          CATCH /aws1/cx_lmdresourcenotfoundex.
            MESSAGE 'The requested resource does not exist.' TYPE 'E'.
          CATCH /aws1/cx_lmdunsuppedmediatyp00.
            MESSAGE 'Invoke request body does not have JSON as its content type.'
 TYPE 'E'.
        ENDTRY.

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

        TRY.
```

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

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

        "Invoke the function with new parameters and get results. Display the
 execution log that's returned from the invocation."
        TRY.
            lv_json = /aws1/cl_rt_util=>string_to_xstring(
              `{`  &&
                `"action": "decrement",`  &&
                `"number": 10` &&
              `}` ).
            DATA(lo_updated_invoke_output) = lo_lmd->invoke(
                       iv_functionname = iv_function_name
                       iv_payload = lv_json ).
            ov_updated_invoke_payload = lo_updated_invoke_output->get_payload( ).
        " ov_updated_invoke_payload is returned for testing purposes. "
            lo_writer_json = cl_sxml_string_writer=>create( type =
 if_sxml=>co_xt_json ).
            CALL TRANSFORMATION id SOURCE XML ov_updated_invoke_payload RESULT XML
 lo_writer_json.
            lv_result = cl_abap_codepage=>convert_from( lo_writer_json-
>get_output( ) ).
            MESSAGE 'Lambda function invoked.' TYPE 'I'.
```

```
          CATCH /aws1/cx_lmdinvparamvalueex.
            MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
          CATCH /aws1/cx_lmdinvrequestcontex.
            MESSAGE 'Unable to parse request body as JSON.' TYPE 'E'.
          CATCH /aws1/cx_lmdresourcenotfoundex.
            MESSAGE 'The requested resource does not exist.' TYPE 'E'.
          CATCH /aws1/cx_lmdunsuppedmediatyp00.
            MESSAGE 'Invoke request body does not have JSON as its content type.'
 TYPE 'E'.
        ENDTRY.

        " List the functions for your account. "
        TRY.
            DATA(lo_list_output) = lo_lmd->listfunctions( ).
            DATA(lt_functions) = lo_list_output->get_functions( ).
            MESSAGE 'Retrieved list of Lambda functions.' TYPE 'I'.
          CATCH /aws1/cx_lmdinvparamvalueex.
            MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
        ENDTRY.

        " Delete the Lambda function. "
        TRY.
            lo_lmd->deletefunction( iv_functionname = iv_function_name ).
            MESSAGE 'Lambda function deleted.' TYPE 'I'.
          CATCH /aws1/cx_lmdinvparamvalueex.
            MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
          CATCH /aws1/cx_lmdresourcenotfoundex.
            MESSAGE 'The requested resource does not exist.' TYPE 'W'.
        ENDTRY.

        " Detach role policy. "
        TRY.
            lo_iam->detachrolepolicy(
                iv_rolename  = iv_role_name
                iv_policyarn = 'arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole' ).
            MESSAGE 'Detached policy from the IAM role.' TYPE 'I'.
          CATCH /aws1/cx_iaminvalidinputex.
            MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
          CATCH /aws1/cx_iamnosuchentityex.
            MESSAGE 'The requested resource entity does not exist.' TYPE 'W'.
          CATCH /aws1/cx_iamplynotattachableex.
            MESSAGE 'Service role policies can only be attached to the service-
linked role for their service.' TYPE 'E'.
```

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

        " Delete the IAM role. "
        TRY.
            lo_iam->deleterole( iv_rolename = iv_role_name ).
            MESSAGE 'IAM role deleted.' TYPE 'I'.
          CATCH /aws1/cx_iamnosuchentityex.
            MESSAGE 'The requested resource entity does not exist.' TYPE 'W'.
          CATCH /aws1/cx_iamunmodableentityex.
            MESSAGE 'Service that depends on the service-linked role is not
modifiable.' TYPE 'E'.
        ENDTRY.

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

- Untuk mengetahui hal detail mengenai API, silakan lihat topik-topik berikut di referensi API AWS SDK untuk ABAP SAP.
  - CreateFunction
  - DeleteFunction
  - GetFunction
  - Memohon
  - ListFunctions
  - UpdateFunctionCode
  - UpdateFunctionConfiguration

Tindakan

`CreateFunction`

Contoh kode berikut menunjukkan cara menggunakan`CreateFunction`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat CreateFunctiondi AWS SDK untuk referensi SAP ABAP API.

`DeleteFunction`

Contoh kode berikut menunjukkan cara menggunakan`DeleteFunction`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat DeleteFunctiondi AWS SDK untuk referensi SAP ABAP API.

`GetFunction`

Contoh kode berikut menunjukkan cara menggunakan`GetFunction`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat GetFunctiondi AWS SDK untuk referensi SAP ABAP API.

`Panggil`

Contoh kode berikut menunjukkan cara menggunakan`Invoke`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   TRY.
       DATA(lv_json) = /aws1/cl_rt_util=>string_to_xstring(
         `{`  &&
```

```
            `"action": "increment",`  &&
            `"number": 10` &&
          `}` ).
        oo_result = lo_lmd->invoke(                  " oo_result is returned for
testing purposes. "
                iv_functionname = iv_function_name
                iv_payload = lv_json ).
        MESSAGE 'Lambda function invoked.' TYPE 'I'.
      CATCH /aws1/cx_lmdinvparamvalueex.
        MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
      CATCH /aws1/cx_lmdinvrequestcontex.
        MESSAGE 'Unable to parse request body as JSON.' TYPE 'E'.
      CATCH /aws1/cx_lmdinvalidzipfileex.
        MESSAGE 'The deployment package could not be unzipped.' TYPE 'E'.
      CATCH /aws1/cx_lmdrequesttoolargeex.
        MESSAGE 'Invoke request body JSON input limit was exceeded by the request
payload.' TYPE 'E'.
      CATCH /aws1/cx_lmdresourceconflictex.
        MESSAGE 'Resource already exists or another operation is in progress.' TYPE
'E'.
      CATCH /aws1/cx_lmdresourcenotfoundex.
        MESSAGE 'The requested resource does not exist.' TYPE 'E'.
      CATCH /aws1/cx_lmdserviceexception.
        MESSAGE 'An internal problem was encountered by the AWS Lambda service.'
TYPE 'E'.
      CATCH /aws1/cx_lmdtoomanyrequestsex.
        MESSAGE 'The maximum request throughput was reached.' TYPE 'E'.
      CATCH /aws1/cx_lmdunsuppedmediatyp00.
        MESSAGE 'Invoke request body does not have JSON as its content type.' TYPE
'E'.
    ENDTRY.
```

- Untuk detail API, lihat Memanggil di AWS SDK untuk referensi SAP ABAP API.

`ListFunctions`

Contoh kode berikut menunjukkan cara menggunakan`ListFunctions`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat ListFunctionsdi AWS SDK untuk referensi SAP ABAP API.

`UpdateFunctionCode`

Contoh kode berikut menunjukkan cara menggunakan`UpdateFunctionCode`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat UpdateFunctionCodedi AWS SDK untuk referensi SAP ABAP API.

`UpdateFunctionConfiguration`

Contoh kode berikut menunjukkan cara menggunakan`UpdateFunctionConfiguration`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat UpdateFunctionConfigurationdi AWS SDK untuk referensi SAP ABAP API.

Contoh Organizations menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan mengimplementasikan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP with Organizations.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`AttachPolicy`

Contoh kode berikut menunjukkan cara menggunakan`AttachPolicy`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_org->attachpolicy(
      iv_policyid = iv_policy_id
      iv_targetid = iv_target_id ).
    MESSAGE 'Policy attached to target.' TYPE 'I'.
  CATCH /aws1/cx_orgaccessdeniedex.
    MESSAGE 'You do not have permission to attach the policy.' TYPE 'E'.
  CATCH /aws1/cx_orgpolicynotfoundex.
    MESSAGE 'The specified policy does not exist.' TYPE 'E'.
  CATCH /aws1/cx_orgtargetnotfoundex.
    MESSAGE 'The specified target does not exist.' TYPE 'E'.
  CATCH /aws1/cx_orgduplicateplyatta00.
    MESSAGE 'The policy is already attached to the target.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat AttachPolicydi AWS SDK untuk referensi SAP ABAP API.

`CreatePolicy`

Contoh kode berikut menunjukkan cara menggunakan`CreatePolicy`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   TRY.
       oo_result = lo_org->createpolicy(       " oo_result is returned for testing
purposes. "
         iv_name        = iv_policy_name
         iv_description = iv_policy_description
         iv_content     = iv_policy_content
         iv_type        = iv_policy_type ).
       MESSAGE 'Policy created.' TYPE 'I'.
     CATCH /aws1/cx_orgaccessdeniedex.
       MESSAGE 'You do not have permission to create a policy.' TYPE 'E'.
     CATCH /aws1/cx_orgduplicatepolicyex.
       MESSAGE 'A policy with this name already exists.' TYPE 'E'.
     CATCH /aws1/cx_orgmalformedplydocex.
       MESSAGE 'The policy content is malformed.' TYPE 'E'.
   ENDTRY.
```

- Untuk detail API, lihat CreatePolicydi AWS SDK untuk referensi SAP ABAP API.

`DeletePolicy`

Contoh kode berikut menunjukkan cara menggunakan`DeletePolicy`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        lo_org->deletepolicy(
          iv_policyid = iv_policy_id ).
        MESSAGE 'Policy deleted.' TYPE 'I'.
      CATCH /aws1/cx_orgaccessdeniedex.
        MESSAGE 'You do not have permission to delete the policy.' TYPE 'E'.
      CATCH /aws1/cx_orgpolicynotfoundex.
        MESSAGE 'The specified policy does not exist.' TYPE 'E'.
      CATCH /aws1/cx_orgpolicyinuseex.
        MESSAGE 'The policy is still attached to one or more targets.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat DeletePolicydi AWS SDK untuk referensi SAP ABAP API.

`DescribePolicy`

Contoh kode berikut menunjukkan cara menggunakan`DescribePolicy`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_org->describepolicy(     " oo_result is returned for testing
purposes. "
          iv_policyid = iv_policy_id ).
```

```
    DATA(lo_policy) = oo_result->get_policy( ).
    MESSAGE 'Retrieved policy details.' TYPE 'I'.
  CATCH /aws1/cx_orgaccessdeniedex.
    MESSAGE 'You do not have permission to describe the policy.' TYPE 'E'.
  CATCH /aws1/cx_orgpolicynotfoundex.
    MESSAGE 'The specified policy does not exist.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DescribePolicydi AWS SDK untuk referensi SAP ABAP API.

`DetachPolicy`

Contoh kode berikut menunjukkan cara menggunakan`DetachPolicy`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_org->detachpolicy(
      iv_policyid = iv_policy_id
      iv_targetid = iv_target_id ).
    MESSAGE 'Policy detached from target.' TYPE 'I'.
  CATCH /aws1/cx_orgaccessdeniedex.
    MESSAGE 'You do not have permission to detach the policy.' TYPE 'E'.
  CATCH /aws1/cx_orgpolicynotfoundex.
    MESSAGE 'The specified policy does not exist.' TYPE 'E'.
  CATCH /aws1/cx_orgtargetnotfoundex.
    MESSAGE 'The specified target does not exist.' TYPE 'E'.
  CATCH /aws1/cx_orgpolicynotattex.
    MESSAGE 'The policy is not attached to the target.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DetachPolicydi AWS SDK untuk referensi SAP ABAP API.

`ListPolicies`

Contoh kode berikut menunjukkan cara menggunakan`ListPolicies`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_org->listpolicies(       " oo_result is returned for testing
purposes. "
          iv_filter = iv_filter ).
        DATA(lt_policies) = oo_result->get_policies( ).
        MESSAGE 'Retrieved list of policies.' TYPE 'I'.
      CATCH /aws1/cx_orgaccessdeniedex.
        MESSAGE 'You do not have permission to list policies.' TYPE 'E'.
      CATCH /aws1/cx_orgawsorgsnotinuseex.
        MESSAGE 'Your account is not a member of an organization.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat ListPoliciesdi AWS SDK untuk referensi SAP ABAP API.

Amazon Pinpoint contoh menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Amazon Pinpoint.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`SendMessages`

Contoh kode berikut menunjukkan cara menggunakan`SendMessages`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

Kirim pesan email.

```
" Build the addresses map from the list of to_addresses
DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
LOOP AT it_to_addresses INTO DATA(lo_address).
  INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
    key = lo_address->get_value( )
    value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'EMAIL' )
  ) INTO TABLE lt_addresses.
ENDLOOP.

" Send the email message
DATA(lo_result) = lo_ppt->sendmessages(
  iv_applicationid = iv_app_id
  io_messagerequest = NEW /aws1/cl_pptmessagerequest(
    it_addresses = lt_addresses
    io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
      io_emailmessage = NEW /aws1/cl_pptemailmessage(
        iv_fromaddress = iv_sender
        io_simpleemail = NEW /aws1/cl_pptsimpleemail(
          io_subject = NEW /aws1/cl_pptsimpleemailpart(
            iv_charset = iv_char_set
            iv_data = iv_subject
          )
          io_htmlpart = NEW /aws1/cl_pptsimpleemailpart(
            iv_charset = iv_char_set
```

```
              iv_data = iv_html_message
            )
            io_textpart = NEW /aws1/cl_pptsimpleemailpart(
              iv_charset = iv_char_set
              iv_data = iv_text_message
            )
          )
        )
      )
    ).

    " Extract message IDs from response
    DATA(lo_message_response) = lo_result->get_messageresponse( ).
    ot_message_ids = lo_message_response->get_result( ).

    MESSAGE 'Email message sent successfully.' TYPE 'I'.
```

Kirim pesan SMS.

```
    " Build the addresses map for the destination number
    DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
    INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
      key = iv_destination_number
      value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'SMS' )
    ) INTO TABLE lt_addresses.

    " Send the SMS message
    DATA(lo_result) = lo_ppt->sendmessages(
      iv_applicationid = iv_app_id
      io_messagerequest = NEW /aws1/cl_pptmessagerequest(
        it_addresses = lt_addresses
        io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
          io_smsmessage = NEW /aws1/cl_pptsmsmessage(
            iv_body = iv_message
            iv_messagetype = iv_message_type
            iv_originationnumber = iv_origination_number
          )
        )
      )
    ).
```

```
" Extract message ID from response
DATA(lo_message_response) = lo_result->get_messageresponse( ).
DATA(lt_results) = lo_message_response->get_result( ).
LOOP AT lt_results INTO DATA(ls_result).
  IF ls_result-key = iv_destination_number.
    ov_message_id = ls_result-value->get_messageid( ).
    EXIT.
  ENDIF.
ENDLOOP.

MESSAGE 'SMS message sent successfully.' TYPE 'I'.
```

Kirim pesan email dengan template email yang ada.

```
" Build the addresses map from the list of to_addresses
DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
LOOP AT it_to_addresses INTO DATA(lo_address).
  INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
    key = lo_address->get_value( )
    value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'EMAIL' )
  ) INTO TABLE lt_addresses.
ENDLOOP.

" Send the email message using a template
DATA(lo_result) = lo_ppt->sendmessages(
  iv_applicationid = iv_app_id
  io_messagerequest = NEW /aws1/cl_pptmessagerequest(
    it_addresses = lt_addresses
    io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
      io_emailmessage = NEW /aws1/cl_pptemailmessage(
        iv_fromaddress = iv_sender
      )
    )
    io_templateconfiguration = NEW /aws1/cl_ppttemplateconf(
      io_emailtemplate = NEW /aws1/cl_ppttemplate(
        iv_name = iv_template_name
        iv_version = iv_template_version
      )
    )
  )
).
```

```
    " Extract message IDs from response
    DATA(lo_message_response) = lo_result->get_messageresponse( ).
    ot_message_ids = lo_message_response->get_result( ).

    MESSAGE 'Templated email message sent successfully.' TYPE 'I'.
```

Kirim pesan teks dengan template SMS yang ada.

```
    " Build the addresses map for the destination number
    DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
    INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
      key = iv_destination_number
      value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'SMS' )
    ) INTO TABLE lt_addresses.

    " Send the SMS message using a template
    DATA(lo_result) = lo_ppt->sendmessages(
      iv_applicationid = iv_app_id
      io_messagerequest = NEW /aws1/cl_pptmessagerequest(
        it_addresses = lt_addresses
        io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
          io_smsmessage = NEW /aws1/cl_pptsmsmessage(
            iv_messagetype = iv_message_type
            iv_originationnumber = iv_origination_number
          )
        )
        io_templateconfiguration = NEW /aws1/cl_ppttemplateconf(
          io_smstemplate = NEW /aws1/cl_ppttemplate(
            iv_name = iv_template_name
            iv_version = iv_template_version
          )
        )
      )
    ).

    " Extract message ID from response
    DATA(lo_message_response) = lo_result->get_messageresponse( ).
    DATA(lt_results) = lo_message_response->get_result( ).
    LOOP AT lt_results INTO DATA(ls_result).
      IF ls_result-key = iv_destination_number.
        ov_message_id = ls_result-value->get_messageid( ).
        EXIT.
```

```
    ENDIF.
  ENDLOOP.

  MESSAGE 'Templated SMS message sent successfully.' TYPE 'I'.
```

- Untuk detail API, lihat SendMessagesdi AWS SDK untuk referensi SAP ABAP API.

Amazon Pinpoint SMS dan Voice API contoh menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Amazon Pinpoint SMS dan Voice API.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`CreateConfigurationSet`

Contoh kode berikut menunjukkan cara menggunakan`CreateConfigurationSet`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " Create a new configuration set
        lo_pps->createconfigurationset(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-config-
set'
        ).

        MESSAGE 'Configuration set created successfully.' TYPE 'I'.

      CATCH /aws1/cx_ppsalreadyexistsex INTO DATA(lo_already_exists_ex).
        MESSAGE lo_already_exists_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_already_exists_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppslimitexceededex INTO DATA(lo_limit_exceeded_ex).
        MESSAGE lo_limit_exceeded_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_limit_exceeded_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- Untuk detail API, lihat CreateConfigurationSetdi AWS SDK untuk referensi SAP ABAP API.

`CreateConfigurationSetEventDestination`

Contoh kode berikut menunjukkan cara menggunakan`CreateConfigurationSetEventDestination`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " Create event destination for the configuration set
        lo_pps->createconfseteventdst(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-config-
set'
          iv_eventdestinationname = iv_event_destination_name    " e.g., 'my-event-
dest'
          io_eventdestination = io_event_destination
        ).

        MESSAGE 'Event destination created successfully.' TYPE 'I'.

      CATCH /aws1/cx_ppsalreadyexistsex INTO DATA(lo_already_exists_ex).
        MESSAGE lo_already_exists_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_already_exists_ex.
      CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
        MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_not_found_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppslimitexceededex INTO DATA(lo_limit_exceeded_ex).
        MESSAGE lo_limit_exceeded_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_limit_exceeded_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- Untuk detail API, lihat CreateConfigurationSetEventDestinationdi AWS SDK untuk referensi SAP ABAP API.

`DeleteConfigurationSet`

Contoh kode berikut menunjukkan cara menggunakan`DeleteConfigurationSet`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " Delete the configuration set
        lo_pps->deleteconfigurationset(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-config-
set'
        ).

        MESSAGE 'Configuration set deleted successfully.' TYPE 'I'.

      CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
        MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_not_found_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- Untuk detail API, lihat DeleteConfigurationSetdi AWS SDK untuk referensi SAP ABAP API.

`DeleteConfigurationSetEventDestination`

Contoh kode berikut menunjukkan cara menggunakan`DeleteConfigurationSetEventDestination`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " Delete the event destination
        lo_pps->deleteconfseteventdst(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-config-
set'
          iv_eventdestinationname = iv_event_destination_name    " e.g., 'my-event-
dest'
        ).

        MESSAGE 'Event destination deleted successfully.' TYPE 'I'.

      CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
        MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_not_found_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- Untuk detail API, lihat DeleteConfigurationSetEventDestinationdi AWS SDK untuk referensi SAP ABAP API.

`GetConfigurationSetEventDestinations`

Contoh kode berikut menunjukkan cara menggunakan`GetConfigurationSetEventDestinations`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " Get event destinations for the configuration set
        oo_result = lo_pps->getconfseteventdestinations(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-config-
set'
        ).

        " Process the event destinations
        LOOP AT oo_result->get_eventdestinations( ) INTO DATA(lo_event_dest).
          DATA(lv_dest_name) = lo_event_dest->get_name( ).
          DATA(lv_enabled) = lo_event_dest->get_enabled( ).

          MESSAGE |Event destination: { lv_dest_name }, Enabled: { lv_enabled }|
 TYPE 'I'.

          " Check for CloudWatch Logs destination
          DATA(lo_cloudwatch_dest) = lo_event_dest->get_cloudwatchlogsdst( ).
          IF lo_cloudwatch_dest IS NOT INITIAL.
            DATA(lv_log_group_arn) = lo_cloudwatch_dest->get_loggrouparn( ).
            MESSAGE |  CloudWatch Logs destination: { lv_log_group_arn }| TYPE 'I'.
          ENDIF.

          " Check for Kinesis Firehose destination
          DATA(lo_firehose_dest) = lo_event_dest->get_kinesisfirehosedst( ).
          IF lo_firehose_dest IS NOT INITIAL.
            DATA(lv_delivery_stream) = lo_firehose_dest->get_deliverystreamarn( ).
            MESSAGE |  Kinesis Firehose destination: { lv_delivery_stream }| TYPE
 'I'.
          ENDIF.

          " Check for SNS destination
          DATA(lo_sns_dest) = lo_event_dest->get_snsdestination( ).
          IF lo_sns_dest IS NOT INITIAL.
            DATA(lv_topic_arn) = lo_sns_dest->get_topicarn( ).
```

```
            MESSAGE |  SNS destination: { lv_topic_arn }| TYPE 'I'.
          ENDIF.
        ENDLOOP.

      CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
        MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_not_found_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- Untuk detail API, lihat GetConfigurationSetEventDestinationsdi AWS SDK untuk referensi SAP ABAP API.

`ListConfigurationSets`

Contoh kode berikut menunjukkan cara menggunakan`ListConfigurationSets`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " List all configuration sets
        oo_result = lo_pps->listconfigurationsets(
          iv_nexttoken = iv_next_token    " Optional: Token for pagination
          iv_pagesize = iv_page_size      " Optional: Number of results per page,
e.g., '10'
        ).
```

```
      " Process the configuration sets
      LOOP AT oo_result->get_configurationsets( ) INTO DATA(lo_config_set).
        DATA(lv_config_set_name) = lo_config_set->get_value( ).
        MESSAGE |Configuration set: { lv_config_set_name }| TYPE 'I'.
      ENDLOOP.

      " Check if there are more results
      DATA(lv_next_token) = oo_result->get_nexttoken( ).
      IF lv_next_token IS NOT INITIAL.
        MESSAGE |More results available. Next token: { lv_next_token }| TYPE 'I'.
      ENDIF.

    CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
      MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_bad_request_ex.
    CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
      MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_internal_error_ex.
    CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
      MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_too_many_requests_ex.
  ENDTRY.
```

- Untuk detail API, lihat ListConfigurationSetsdi AWS SDK untuk referensi SAP ABAP API.

`SendVoiceMessage`

Contoh kode berikut menunjukkan cara menggunakan`SendVoiceMessage`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
  TRY.
      " Create SSML message type object with voice parameters
```

```
      DATA(lo_ssml_message) = NEW /aws1/cl_ppsssmlmessagetype(
        iv_languagecode = iv_language_code    " e.g., 'en-US'
        iv_voiceid = iv_voice_id              " e.g., 'Matthew'
        iv_text = iv_ssml_message             " SSML formatted message text
      ).

      " Create voice message content with the SSML message
      DATA(lo_content) = NEW /aws1/cl_ppsvoicemessagecont(
        io_ssmlmessage = lo_ssml_message
      ).

      " Send the voice message
      DATA(lo_result) = lo_pps->sendvoicemessage(
        iv_originationphonenumber = iv_origination_number  " e.g., '+12065550110'
        iv_callerid = iv_caller_id                         " e.g., '+12065550199'
        iv_destinationphonenumber = iv_destination_number  " e.g., '+12065550142'
        io_content = lo_content
      ).

      " Retrieve the message ID from the response
      ov_message_id = lo_result->get_messageid( ).

      MESSAGE 'Voice message sent successfully.' TYPE 'I'.

    CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
      MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_bad_request_ex.
    CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
      MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_internal_error_ex.
    CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
      MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_too_many_requests_ex.
  ENDTRY.
```

- Untuk detail API, lihat SendVoiceMessagedi AWS SDK untuk referensi SAP ABAP API.

`UpdateConfigurationSetEventDestination`

Contoh kode berikut menunjukkan cara menggunakan`UpdateConfigurationSetEventDestination`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " Update the event destination
        lo_pps->updateconfseteventdst(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-config-
set'
          iv_eventdestinationname = iv_event_destination_name    " e.g., 'my-event-
dest'
          io_eventdestination = io_event_destination
        ).

        MESSAGE 'Event destination updated successfully.' TYPE 'I'.

      CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
        MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_not_found_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- Untuk detail API, lihat UpdateConfigurationSetEventDestinationdi AWS SDK untuk referensi SAP ABAP API.

Contoh Amazon Polly menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Amazon Polly.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`DeleteLexicon`

Contoh kode berikut menunjukkan cara menggunakan`DeleteLexicon`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_ply->deletelexicon( iv_name ).
    MESSAGE 'Lexicon deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_plylexiconnotfoundex.
    MESSAGE 'Lexicon not found.' TYPE 'E'.
  CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DeleteLexicondi AWS SDK untuk referensi SAP ABAP API.

`DescribeVoices`

Contoh kode berikut menunjukkan cara menggunakan`DescribeVoices`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " Only pass optional parameters if they have values
    IF iv_engine IS NOT INITIAL AND iv_language IS NOT INITIAL.
      oo_result = lo_ply->describevoices(
        iv_engine = iv_engine
        iv_languagecode = iv_language ).
    ELSEIF iv_engine IS NOT INITIAL.
      oo_result = lo_ply->describevoices(
        iv_engine = iv_engine ).
    ELSEIF iv_language IS NOT INITIAL.
      oo_result = lo_ply->describevoices(
        iv_languagecode = iv_language ).
    ELSE.
      oo_result = lo_ply->describevoices( ).
    ENDIF.
    MESSAGE 'Retrieved voice metadata.' TYPE 'I'.
  CATCH /aws1/cx_plyinvalidnexttokenex.
    MESSAGE 'The NextToken is invalid.' TYPE 'E'.
  CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DescribeVoicesdi AWS SDK untuk referensi SAP ABAP API.

`GetLexicon`

Contoh kode berikut menunjukkan cara menggunakan`GetLexicon`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_ply->getlexicon( iv_name ).
    DATA(lo_lexicon) = oo_result->get_lexicon( ).
    IF lo_lexicon IS BOUND.
      DATA(lv_lex_name) = lo_lexicon->get_name( ).
      MESSAGE |Retrieved lexicon: { lv_lex_name }| TYPE 'I'.
    ENDIF.
  CATCH /aws1/cx_plylexiconnotfoundex.
    MESSAGE 'Lexicon not found.' TYPE 'E'.
  CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat GetLexicondi AWS SDK untuk referensi SAP ABAP API.

`GetSpeechSynthesisTask`

Contoh kode berikut menunjukkan cara menggunakan`GetSpeechSynthesisTask`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_ply->getspeechsynthesistask( iv_task_id ).
    DATA(lo_task) = oo_result->get_synthesistask( ).
```

```
    IF lo_task IS BOUND.
      DATA(lv_status) = lo_task->get_taskstatus( ).
      MESSAGE |Task status: { lv_status }| TYPE 'I'.
    ENDIF.
  CATCH /aws1/cx_plyinvalidtaskidex.
    MESSAGE 'Invalid task ID.' TYPE 'E'.
  CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
  CATCH /aws1/cx_plysynthesistsknotf00.
    MESSAGE 'Synthesis task not found.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat GetSpeechSynthesisTaskdi AWS SDK untuk referensi SAP ABAP API.

`ListLexicons`

Contoh kode berikut menunjukkan cara menggunakan`ListLexicons`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_ply->listlexicons( ).
    DATA(lt_lexicons) = oo_result->get_lexicons( ).
    DATA(lv_count) = lines( lt_lexicons ).
    MESSAGE |Found { lv_count } lexicons| TYPE 'I'.
  CATCH /aws1/cx_plyinvalidnexttokenex.
    MESSAGE 'Invalid NextToken.' TYPE 'E'.
  CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat ListLexiconsdi AWS SDK untuk referensi SAP ABAP API.

`ListSpeechSynthesisTasks`

Contoh kode berikut menunjukkan cara menggunakan`ListSpeechSynthesisTasks`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " Only pass optional parameters if they have values
    IF iv_max_results IS NOT INITIAL AND iv_status IS NOT INITIAL.
      oo_result = lo_ply->listspeechsynthesistasks(
        iv_maxresults = iv_max_results
        iv_status = iv_status ).
    ELSEIF iv_max_results IS NOT INITIAL.
      oo_result = lo_ply->listspeechsynthesistasks(
        iv_maxresults = iv_max_results ).
    ELSEIF iv_status IS NOT INITIAL.
      oo_result = lo_ply->listspeechsynthesistasks(
        iv_status = iv_status ).
    ELSE.
      oo_result = lo_ply->listspeechsynthesistasks( ).
    ENDIF.
    DATA(lt_tasks) = oo_result->get_synthesistasks( ).
    DATA(lv_count) = lines( lt_tasks ).
    MESSAGE |Found { lv_count } synthesis tasks| TYPE 'I'.
  CATCH /aws1/cx_plyinvalidnexttokenex.
    MESSAGE 'Invalid NextToken.' TYPE 'E'.
  CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat ListSpeechSynthesisTasksdi AWS SDK untuk referensi SAP ABAP API.

`PutLexicon`

Contoh kode berikut menunjukkan cara menggunakan`PutLexicon`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_ply->putlexicon(
      iv_name = iv_name
      iv_content = iv_content ).
    MESSAGE 'Lexicon created successfully.' TYPE 'I'.
  CATCH /aws1/cx_plyinvalidlexiconex.
    MESSAGE 'Invalid lexicon.' TYPE 'E'.
  CATCH /aws1/cx_plylexiconsizeexcdex.
    MESSAGE 'Lexicon size exceeded.' TYPE 'E'.
  CATCH /aws1/cx_plymaxlexemelengthe00.
    MESSAGE 'Maximum lexeme length exceeded.' TYPE 'E'.
  CATCH /aws1/cx_plymaxlexiconsnoexc00.
    MESSAGE 'Maximum number of lexicons exceeded.' TYPE 'E'.
  CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
  CATCH /aws1/cx_plyunsuppedplsalpha00.
    MESSAGE 'Unsupported PLS alphabet.' TYPE 'E'.
  CATCH /aws1/cx_plyunsuppedplslangu00.
    MESSAGE 'Unsupported PLS language.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat PutLexicondi AWS SDK untuk referensi SAP ABAP API.

`StartSpeechSynthesisTask`

Contoh kode berikut menunjukkan cara menggunakan`StartSpeechSynthesisTask`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " Only pass optional parameters if they have values
        IF iv_lang_code IS NOT INITIAL AND iv_s3_key_prefix IS NOT INITIAL.
          oo_result = lo_ply->startspeechsynthesistask(
            iv_engine = iv_engine
            iv_outputformat = iv_audio_format
            iv_outputs3bucketname = iv_s3_bucket
            iv_outputs3keyprefix = iv_s3_key_prefix
            iv_text = iv_text
            iv_voiceid = iv_voice_id
            iv_languagecode = iv_lang_code ).
        ELSEIF iv_lang_code IS NOT INITIAL.
          oo_result = lo_ply->startspeechsynthesistask(
            iv_engine = iv_engine
            iv_outputformat = iv_audio_format
            iv_outputs3bucketname = iv_s3_bucket
            iv_text = iv_text
            iv_voiceid = iv_voice_id
            iv_languagecode = iv_lang_code ).
        ELSEIF iv_s3_key_prefix IS NOT INITIAL.
          oo_result = lo_ply->startspeechsynthesistask(
            iv_engine = iv_engine
            iv_outputformat = iv_audio_format
            iv_outputs3bucketname = iv_s3_bucket
            iv_outputs3keyprefix = iv_s3_key_prefix
            iv_text = iv_text
            iv_voiceid = iv_voice_id ).
        ELSE.
          oo_result = lo_ply->startspeechsynthesistask(
            iv_engine = iv_engine
            iv_outputformat = iv_audio_format
            iv_outputs3bucketname = iv_s3_bucket
            iv_text = iv_text
            iv_voiceid = iv_voice_id ).
```

```
    ENDIF.
    MESSAGE 'Speech synthesis task started.' TYPE 'I'.
  CATCH /aws1/cx_plyinvalids3bucketex.
    MESSAGE 'Invalid S3 bucket.' TYPE 'E'.
  CATCH /aws1/cx_plyinvalidssmlex.
    MESSAGE 'Invalid SSML.' TYPE 'E'.
  CATCH /aws1/cx_plylexiconnotfoundex.
    MESSAGE 'Lexicon not found.' TYPE 'E'.
  CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
  CATCH /aws1/cx_plytextlengthexcdex.
    MESSAGE 'Text length exceeded maximum.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat StartSpeechSynthesisTaskdi AWS SDK untuk referensi SAP ABAP API.

`SynthesizeSpeech`

Contoh kode berikut menunjukkan cara menggunakan`SynthesizeSpeech`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " Only pass optional language code if it has a value
    IF iv_lang_code IS NOT INITIAL.
      oo_result = lo_ply->synthesizespeech(
        iv_engine = iv_engine
        iv_outputformat = iv_output_fmt
        iv_text = iv_text
        iv_voiceid = iv_voice_id
        iv_languagecode = iv_lang_code ).
    ELSE.
      oo_result = lo_ply->synthesizespeech(
        iv_engine = iv_engine
```

```
            iv_outputformat = iv_output_fmt
            iv_text = iv_text
            iv_voiceid = iv_voice_id ).
        ENDIF.
        MESSAGE 'Speech synthesized successfully.' TYPE 'I'.
      CATCH /aws1/cx_plyinvalidssmlex.
        MESSAGE 'Invalid SSML.' TYPE 'E'.
      CATCH /aws1/cx_plylexiconnotfoundex.
        MESSAGE 'Lexicon not found.' TYPE 'E'.
      CATCH /aws1/cx_plyservicefailureex.
        MESSAGE 'Service failure occurred.' TYPE 'E'.
      CATCH /aws1/cx_plytextlengthexcdex.
        MESSAGE 'Text length exceeded maximum.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat SynthesizeSpeechdi AWS SDK untuk referensi SAP ABAP API.

Contoh Amazon RDS menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Amazon RDS.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`CreateDBParameterGroup`

Contoh kode berikut menunjukkan cara menggunakan`CreateDBParameterGroup`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
" iv_dbparametergroupname   = 'mydbparametergroup'
" iv_dbparametergroupfamily = 'mysql8.0'
" iv_description            = 'My custom DB parameter group for MySQL 8.0'
TRY.
    oo_result = lo_rds->createdbparametergroup(
      iv_dbparametergroupname   = iv_dbparametergroupname
      iv_dbparametergroupfamily = iv_dbparametergroupfamily
      iv_description            = iv_description ).
    MESSAGE 'DB parameter group created.' TYPE 'I'.
  CATCH /aws1/cx_rdsdbparmgralrexfault.
    MESSAGE 'DB parameter group already exists.' TYPE 'I'.
  CATCH /aws1/cx_rdsdbprmgrquotaexcd00.
    MESSAGE 'DB parameter group quota exceeded.' TYPE 'I'.
ENDTRY.
```

- Untuk detail API, lihat CreateDBParameterGroupdi AWS SDK untuk referensi SAP ABAP API.

`DeleteDBParameterGroup`

Contoh kode berikut menunjukkan cara menggunakan`DeleteDBParameterGroup`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
" iv_dbparametergroupname = 'mydbparametergroup'
TRY.
```

```
        lo_rds->deletedbparametergroup(
          iv_dbparametergroupname = iv_dbparametergroupname ).
        MESSAGE 'DB parameter group deleted.' TYPE 'I'.
      CATCH /aws1/cx_rdsdbprmgrnotfndfault.
        MESSAGE 'DB parameter group not found.' TYPE 'I'.
      CATCH /aws1/cx_rdsinvdbprmgrstatef00.
        MESSAGE 'DB parameter group is in an invalid state.' TYPE 'I'.
    ENDTRY.
```

- Untuk detail API, lihat DeleteDBParameterGroupdi AWS SDK untuk referensi SAP ABAP API.

`DescribeDBEngineVersions`

Contoh kode berikut menunjukkan cara menggunakan`DescribeDBEngineVersions`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat DescribeDBEngineVersionsdi AWS SDK untuk referensi SAP ABAP API.

`DescribeDBParameterGroups`

Contoh kode berikut menunjukkan cara menggunakan`DescribeDBParameterGroups`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
" iv_dbparametergroupname = 'mydbparametergroup'
TRY.
    oo_result = lo_rds->describedbparametergroups(
      iv_dbparametergroupname = iv_dbparametergroupname ).
    MESSAGE 'DB parameter group retrieved.' TYPE 'I'.
  CATCH /aws1/cx_rdsdbprmgrnotfndfault.
    MESSAGE 'DB parameter group not found.' TYPE 'I'.
ENDTRY.
```

- Untuk detail API, lihat DescribeDBParameterGroupsdi AWS SDK untuk referensi SAP ABAP API.

`DescribeDBParameters`

Contoh kode berikut menunjukkan cara menggunakan`DescribeDBParameters`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
" iv_dbparametergroupname = 'mydbparametergroup'
" iv_source               = 'user' (optional - filters by parameter source)
TRY.
    oo_result = lo_rds->describedbparameters(
      iv_dbparametergroupname = iv_dbparametergroupname
      iv_source               = iv_source ).
```

```
        DATA(lv_param_count) = lines( oo_result->get_parameters( ) ).
        MESSAGE |Retrieved { lv_param_count } parameters.| TYPE 'I'.
      CATCH /aws1/cx_rdsdbprmgrnotfndfault.
        MESSAGE 'DB parameter group not found.' TYPE 'I'.
    ENDTRY.
```

- Untuk detail API, lihat DescribedBParameters AWS di SDK untuk referensi API SAP ABAP.

`DescribeOrderableDBInstanceOptions`

Contoh kode berikut menunjukkan cara menggunakan`DescribeOrderableDBInstanceOptions`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat DescribeOrderableDBInstanceOptionsdi AWS SDK untuk referensi SAP ABAP API.

`ModifyDBParameterGroup`

Contoh kode berikut menunjukkan cara menggunakan`ModifyDBParameterGroup`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
" iv_dbparametergroupname = 'mydbparametergroup'
" it_parameters - table containing parameter objects with:
"   - parametername = 'max_connections'
"   - parametervalue = '100'
"   - applymethod = 'immediate' or 'pending-reboot'
TRY.
    oo_result = lo_rds->modifydbparametergroup(
      iv_dbparametergroupname = iv_dbparametergroupname
      it_parameters           = it_parameters ).
    MESSAGE 'DB parameter group modified.' TYPE 'I'.
  CATCH /aws1/cx_rdsdbprmgrnotfndfault.
    MESSAGE 'DB parameter group not found.' TYPE 'I'.
  CATCH /aws1/cx_rdsinvdbprmgrstatef00.
    MESSAGE 'DB parameter group is in an invalid state.' TYPE 'I'.
ENDTRY.
```

- Untuk detail API, lihat ModifyDBParameterGroupdi AWS SDK untuk referensi SAP ABAP API.

Contoh Amazon Redshift menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Amazon Redshift.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`CreateCluster`

Contoh kode berikut menunjukkan cara menggunakan`CreateCluster`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

Buat cluster.

```
TRY.
    " Example values: iv_cluster_identifier = 'my-redshift-cluster'
    " Example values: iv_node_type = 'ra3.4xlarge'
    " Example values: iv_master_username = 'awsuser'
    " Example values: iv_master_password = 'AwsUser1000'
    " Example values: iv_publicly_accessible = abap_true
    " Example values: iv_number_of_nodes = 2
    oo_result = lo_rsh->createcluster(
      iv_clusteridentifier = iv_cluster_identifier
      iv_nodetype = iv_node_type
      iv_masterusername = iv_master_username
      iv_masteruserpassword = iv_master_password
      iv_publiclyaccessible = iv_publicly_accessible
      iv_numberofnodes = iv_number_of_nodes
    ).
    MESSAGE 'Redshift cluster created successfully.' TYPE 'I'.
  CATCH /aws1/cx_rshclustalrdyexfault.
    MESSAGE 'Cluster already exists.' TYPE 'I'.
  CATCH /aws1/cx_rshclstquotaexcdfault.
    MESSAGE 'Cluster quota exceeded.' TYPE 'I'.
ENDTRY.
```

- Untuk detail API, lihat CreateClusterdi AWS SDK untuk referensi SAP ABAP API.

`DeleteCluster`

Contoh kode berikut menunjukkan cara menggunakan`DeleteCluster`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

Hapus klaster .

```
TRY.
    " Example values: iv_cluster_identifier = 'my-redshift-cluster'
    lo_rsh->deletecluster(
      iv_clusteridentifier = iv_cluster_identifier
      iv_skipfinalclustersnapshot = abap_true
    ).
    MESSAGE 'Redshift cluster deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_rshclustnotfoundfault.
    MESSAGE 'Cluster not found.' TYPE 'I'.
  CATCH /aws1/cx_rshinvcluststatefault.
    MESSAGE 'Invalid cluster state for deletion.' TYPE 'I'.
ENDTRY.
```

- Untuk detail API, lihat DeleteClusterdi AWS SDK untuk referensi SAP ABAP API.

`DescribeClusters`

Contoh kode berikut menunjukkan cara menggunakan`DescribeClusters`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

Jelaskan cluster.

```
TRY.
    " Example values: iv_cluster_identifier = 'my-redshift-cluster' (optional)
    oo_result = lo_rsh->describeclusters(
      iv_clusteridentifier = iv_cluster_identifier
    ).
    lt_clusters = oo_result->get_clusters( ).
    lv_cluster_count = lines( lt_clusters ).
    MESSAGE |Retrieved { lv_cluster_count } cluster(s).| TYPE 'I'.
  CATCH /aws1/cx_rshclustnotfoundfault.
    MESSAGE 'Cluster not found.' TYPE 'I'.
ENDTRY.
```

- Untuk detail API, lihat DescribeClustersdi AWS SDK untuk referensi SAP ABAP API.

`DescribeStatement`

Contoh kode berikut menunjukkan cara menggunakan`DescribeStatement`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " Example values: iv_statement_id = 'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx'
    oo_result = lo_rsd->describestatement(
      iv_id = iv_statement_id
    ).
    lv_status = oo_result->get_status( ).
    MESSAGE |Statement status: { lv_status }| TYPE 'I'.
  CATCH /aws1/cx_rsdresourcenotfoundex.
    MESSAGE 'Statement not found.' TYPE 'I'.
  CATCH /aws1/cx_rsdinternalserverex.
    MESSAGE 'Internal server error.' TYPE 'I'.
ENDTRY.
```

- Untuk detail API, lihat DescribeStatementdi AWS SDK untuk referensi SAP ABAP API.

`ExecuteStatement`

Contoh kode berikut menunjukkan cara menggunakan`ExecuteStatement`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " Example values: iv_cluster_identifier = 'redshift-cluster-movies'
        " Example values: iv_database_name = 'dev'
        " Example values: iv_user_name = 'awsuser'
        " Example values: iv_sql = 'SELECT * FROM movies WHERE year = :year'
        " Example values: it_parameter_list - SQL parameters for parameterized
queries

        " Only pass parameters if the list is not empty
        IF it_parameter_list IS NOT INITIAL.
          oo_result = lo_rsd->executestatement(
            iv_clusteridentifier = iv_cluster_identifier
            iv_database = iv_database_name
            iv_dbuser = iv_user_name
            iv_sql = iv_sql
            it_parameters = it_parameter_list
          ).
        ELSE.
          oo_result = lo_rsd->executestatement(
            iv_clusteridentifier = iv_cluster_identifier
            iv_database = iv_database_name
            iv_dbuser = iv_user_name
            iv_sql = iv_sql
          ).
        ENDIF.
```

```
      lv_statement_id = oo_result->get_id( ).
      MESSAGE |Statement executed. ID: { lv_statement_id }| TYPE 'I'.
    CATCH /aws1/cx_rsdexecutestatementex.
      MESSAGE 'Statement execution error.' TYPE 'I'.
    CATCH /aws1/cx_rsdresourcenotfoundex.
      MESSAGE 'Resource not found.' TYPE 'I'.
  ENDTRY.
```

- Untuk detail API, lihat ExecuteStatementdi AWS SDK untuk referensi SAP ABAP API.

`GetStatementResult`

Contoh kode berikut menunjukkan cara menggunakan`GetStatementResult`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

Periksa hasil pernyataan.

```
  TRY.
      " Example values: iv_statement_id = 'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx'
      " Handle pagination for large result sets

      DO.
        lo_result_page = lo_rsd->getstatementresult(
          iv_id = iv_statement_id
          iv_nexttoken = lv_next_token
        ).

        " Collect records from this page
        lt_page_records = lo_result_page->get_records( ).
        APPEND LINES OF lt_page_records TO lt_all_records.

        " Check if there are more pages
```

```
        lv_next_token = lo_result_page->get_nexttoken( ).
        IF lv_next_token IS INITIAL.
          EXIT. " No more pages
        ENDIF.
      ENDDO.

      " For the last call, set oo_result for return value
      oo_result = lo_result_page.
      lv_record_count = lines( lt_all_records ).
      MESSAGE |Retrieved { lv_record_count } record(s).| TYPE 'I'.
    CATCH /aws1/cx_rsdresourcenotfoundex.
      MESSAGE 'Statement not found or results not available.' TYPE 'I'.
    CATCH /aws1/cx_rsdinternalserverex.
      MESSAGE 'Internal server error.' TYPE 'I'.
  ENDTRY.
```

- Untuk detail API, lihat GetStatementResultdi AWS SDK untuk referensi SAP ABAP API.

`ListDatabases`

Contoh kode berikut menunjukkan cara menggunakan`ListDatabases`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
  TRY.
      " Example values: iv_cluster_identifier = 'redshift-cluster-movies'
      " Example values: iv_database_name = 'dev'
      " Example values: iv_database_user = 'awsuser'
      oo_result = lo_rsd->listdatabases(
        iv_clusteridentifier = iv_cluster_identifier
        iv_database = iv_database_name
        iv_dbuser = iv_database_user
      ).
      lt_databases = oo_result->get_databases( ).
```

```
      lv_db_count = lines( lt_databases ).
      MESSAGE |Retrieved { lv_db_count } database(s).| TYPE 'I'.
    CATCH /aws1/cx_rsddatabaseconnex.
      MESSAGE 'Database connection error.' TYPE 'I'.
    CATCH /aws1/cx_rsdresourcenotfoundex.
      MESSAGE 'Cluster not found.' TYPE 'I'.
  ENDTRY.
```

- Untuk detail API, lihat ListDatabasesdi AWS SDK untuk referensi SAP ABAP API.

`ModifyCluster`

Contoh kode berikut menunjukkan cara menggunakan`ModifyCluster`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

Memodifikasi cluster.

```
  TRY.
      " Example values: iv_cluster_identifier = 'my-redshift-cluster'
      " Example values: iv_pref_maintenance_wn = 'wed:07:30-wed:08:00'
      lo_rsh->modifycluster(
        iv_clusteridentifier = iv_cluster_identifier
        iv_preferredmaintenancewin00 = iv_pref_maintenance_wn
      ).
      MESSAGE 'Redshift cluster modified successfully.' TYPE 'I'.
    CATCH /aws1/cx_rshclustnotfoundfault.
      MESSAGE 'Cluster not found.' TYPE 'I'.
    CATCH /aws1/cx_rshinvcluststatefault.
      MESSAGE 'Invalid cluster state for modification.' TYPE 'I'.
  ENDTRY.
```

- Untuk detail API, lihat ModifyClusterdi AWS SDK untuk referensi SAP ABAP API.

Contoh Rekognition Amazon menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Amazon Rekognition.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`CompareFaces`

Contoh kode berikut menunjukkan cara menggunakan`CompareFaces`.

Untuk informasi selengkapnya, lihat Membandingkan wajah dalam gambar.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " Create S3 object reference for the source image
    DATA(lo_source_s3obj) = NEW /aws1/cl_reks3object(
      iv_bucket = iv_source_s3_bucket
      iv_name = iv_source_s3_key ).

    " Create source image object
    DATA(lo_source_image) = NEW /aws1/cl_rekimage(
```

```
        io_s3object = lo_source_s3obj ).

      " Create S3 object reference for the target image
      DATA(lo_target_s3obj) = NEW /aws1/cl_reks3object(
        iv_bucket = iv_target_s3_bucket
        iv_name = iv_target_s3_key ).

      " Create target image object
      DATA(lo_target_image) = NEW /aws1/cl_rekimage(
        io_s3object = lo_target_s3obj ).

      " Compare faces
      oo_result = lo_rek->comparefaces(
        io_sourceimage = lo_source_image
        io_targetimage = lo_target_image
        iv_similaritythreshold = iv_similarity ).

      DATA(lt_face_matches) = oo_result->get_facematches( ).
      DATA(lt_unmatched_faces) = oo_result->get_unmatchedfaces( ).

      " Get counts of matched and unmatched faces
      DATA(lv_matched_count) = lines( lt_face_matches ).
      DATA(lv_unmatched_count) = lines( lt_unmatched_faces ).

      " Output detailed comparison results
      DATA(lv_message) = |Face comparison completed: | &&
                         |{ lv_matched_count } matched face(s), | &&
                         |{ lv_unmatched_count } unmatched face(s).|.
      MESSAGE lv_message TYPE 'I'.
    CATCH /aws1/cx_rekinvalids3objectex.
      MESSAGE 'Invalid S3 object.' TYPE 'E'.
    CATCH /aws1/cx_rekinvalidparameterex.
      MESSAGE 'Invalid parameter value.' TYPE 'E'.
  ENDTRY.
```

- Untuk detail API, lihat CompareFacesdi AWS SDK untuk referensi SAP ABAP API.

`CreateCollection`

Contoh kode berikut menunjukkan cara menggunakan`CreateCollection`.

Untuk informasi selengkapnya, lihat Membuat koleksi.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_rek->createcollection(
      iv_collectionid = iv_collection_id ).
    MESSAGE 'Collection created successfully.' TYPE 'I'.
  CATCH /aws1/cx_rekresrcalrdyexistsex.
    MESSAGE 'Collection already exists.' TYPE 'E'.
  CATCH /aws1/cx_rekinvalidparameterex.
    MESSAGE 'Invalid parameter value.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat CreateCollectiondi AWS SDK untuk referensi SAP ABAP API.

`DeleteCollection`

Contoh kode berikut menunjukkan cara menggunakan`DeleteCollection`.

Untuk informasi selengkapnya, lihat Menghapus koleksi.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_rek->deletecollection(
      iv_collectionid = iv_collection_id ).
    MESSAGE 'Collection deleted successfully.' TYPE 'I'.
```

```
  CATCH /aws1/cx_rekresourcenotfoundex.
    MESSAGE 'Collection not found.' TYPE 'E'.
  CATCH /aws1/cx_rekinvalidparameterex.
    MESSAGE 'Invalid parameter value.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DeleteCollectiondi AWS SDK untuk referensi SAP ABAP API.

`DeleteFaces`

Contoh kode berikut menunjukkan cara menggunakan`DeleteFaces`.

Untuk informasi selengkapnya, lihat Menghapus wajah dari koleksi.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_rek->deletefaces(
      iv_collectionid = iv_collection_id
      it_faceids = it_face_ids ).

    DATA(lt_deleted_faces) = oo_result->get_deletedfaces( ).
    DATA(lv_deleted_count) = lines( lt_deleted_faces ).
    DATA(lv_msg6) = |{ lv_deleted_count } face(s) deleted successfully.|.
    MESSAGE lv_msg6 TYPE 'I'.
  CATCH /aws1/cx_rekresourcenotfoundex.
    MESSAGE 'Collection not found.' TYPE 'E'.
  CATCH /aws1/cx_rekinvalidparameterex.
    MESSAGE 'Invalid parameter value.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DeleteFacesdi AWS SDK untuk referensi SAP ABAP API.

`DescribeCollection`

Contoh kode berikut menunjukkan cara menggunakan`DescribeCollection`.

Untuk informasi selengkapnya, lihat Menjelaskan koleksi.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_rek->describecollection(
      iv_collectionid = iv_collection_id ).
    DATA(lv_face_count) = oo_result->get_facecount( ).
    DATA(lv_msg) = |Collection described: { lv_face_count } face(s) indexed.|.
    MESSAGE lv_msg TYPE 'I'.
  CATCH /aws1/cx_rekresourcenotfoundex.
    MESSAGE 'Collection not found.' TYPE 'E'.
  CATCH /aws1/cx_rekinvalidparameterex.
    MESSAGE 'Invalid parameter value.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DescribeCollectiondi AWS SDK untuk referensi SAP ABAP API.

`DetectFaces`

Contoh kode berikut menunjukkan cara menggunakan`DetectFaces`.

Untuk informasi selengkapnya, lihat Mendeteksi wajah dalam gambar.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " Create S3 object reference for the image
        DATA(lo_s3object) = NEW /aws1/cl_reks3object(
          iv_bucket = iv_s3_bucket
          iv_name = iv_s3_key ).

        " Create image object
        DATA(lo_image) = NEW /aws1/cl_rekimage(
          io_s3object = lo_s3object ).

        " Detect faces in the image with all attributes
        DATA(lt_attributes) = VALUE /aws1/cl_rekattributes_w=>tt_attributes( ).
        DATA(lo_attr_wrapper) = NEW /aws1/cl_rekattributes_w( iv_value = 'ALL' ).
        INSERT lo_attr_wrapper INTO TABLE lt_attributes.

        oo_result = lo_rek->detectfaces(
          io_image = lo_image
          it_attributes = lt_attributes ).

        DATA(lt_face_details) = oo_result->get_facedetails( ).
        DATA(lv_detected_count) = lines( lt_face_details ).
        DATA(lv_msg8) = |{ lv_detected_count } face(s) detected in image.|.
        MESSAGE lv_msg8 TYPE 'I'.
      CATCH /aws1/cx_rekinvalids3objectex.
        MESSAGE 'Invalid S3 object.' TYPE 'E'.
      CATCH /aws1/cx_rekinvalidparameterex.
        MESSAGE 'Invalid parameter value.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat DetectFacesdi AWS SDK untuk referensi SAP ABAP API.

`DetectLabels`

Contoh kode berikut menunjukkan cara menggunakan`DetectLabels`.

Untuk informasi selengkapnya, lihat Mendeteksi label dalam gambar.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " Create S3 object reference for the image
    DATA(lo_s3object) = NEW /aws1/cl_reks3object(
      iv_bucket = iv_s3_bucket
      iv_name = iv_s3_key ).

    " Create image object
    DATA(lo_image) = NEW /aws1/cl_rekimage(
      io_s3object = lo_s3object ).

    " Detect labels in the image
    oo_result = lo_rek->detectlabels(
      io_image = lo_image
      iv_maxlabels = iv_max_labels ).

    DATA(lt_labels) = oo_result->get_labels( ).
    DATA(lv_label_count) = lines( lt_labels ).
    DATA(lv_msg9) = |{ lv_label_count } label(s) detected in image.|.
    MESSAGE lv_msg9 TYPE 'I'.
  CATCH /aws1/cx_rekinvalids3objectex.
    MESSAGE 'Invalid S3 object.' TYPE 'E'.
  CATCH /aws1/cx_rekinvalidparameterex.
    MESSAGE 'Invalid parameter value.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DetectLabelsdi AWS SDK untuk referensi SAP ABAP API.

`DetectModerationLabels`

Contoh kode berikut menunjukkan cara menggunakan`DetectModerationLabels`.

Untuk informasi selengkapnya, lihat Mendeteksi gambar yang tidak pantas.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " Create S3 object reference for the image
    DATA(lo_s3object) = NEW /aws1/cl_reks3object(
      iv_bucket = iv_s3_bucket
      iv_name = iv_s3_key ).

    " Create image object
    DATA(lo_image) = NEW /aws1/cl_rekimage(
      io_s3object = lo_s3object ).

    " Detect moderation labels
    oo_result = lo_rek->detectmoderationlabels(
      io_image = lo_image ).

    DATA(lt_moderation_labels) = oo_result->get_moderationlabels( ).
    DATA(lv_mod_count) = lines( lt_moderation_labels ).
    DATA(lv_msg10) = |{ lv_mod_count } moderation label(s) detected.|.
    MESSAGE lv_msg10 TYPE 'I'.
  CATCH /aws1/cx_rekinvalids3objectex.
    MESSAGE 'Invalid S3 object.' TYPE 'E'.
  CATCH /aws1/cx_rekinvalidparameterex.
    MESSAGE 'Invalid parameter value.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DetectModerationLabelsdi AWS SDK untuk referensi SAP ABAP API.

`DetectText`

Contoh kode berikut menunjukkan cara menggunakan`DetectText`.

Untuk informasi selengkapnya, lihat Mendeteksi teks dalam gambar.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " Create S3 object reference for the image
    DATA(lo_s3object) = NEW /aws1/cl_reks3object(
      iv_bucket = iv_s3_bucket
      iv_name = iv_s3_key ).

    " Create image object
    DATA(lo_image) = NEW /aws1/cl_rekimage(
      io_s3object = lo_s3object ).

    " Detect text in the image
    oo_result = lo_rek->detecttext(
      io_image = lo_image ).

    DATA(lt_text_detections) = oo_result->get_textdetections( ).
    DATA(lv_text_count) = lines( lt_text_detections ).
    DATA(lv_msg11) = |{ lv_text_count } text detection(s) found.|.
    MESSAGE lv_msg11 TYPE 'I'.
  CATCH /aws1/cx_rekinvalids3objectex.
    MESSAGE 'Invalid S3 object.' TYPE 'E'.
  CATCH /aws1/cx_rekinvalidparameterex.
    MESSAGE 'Invalid parameter value.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DetectTextdi AWS SDK untuk referensi SAP ABAP API.

`IndexFaces`

Contoh kode berikut menunjukkan cara menggunakan`IndexFaces`.

Untuk informasi selengkapnya, lihat Menambahkan wajah ke koleksi.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " Create S3 object reference for the image
        DATA(lo_s3object) = NEW /aws1/cl_reks3object(
          iv_bucket = iv_s3_bucket
          iv_name = iv_s3_key ).

        " Create image object
        DATA(lo_image) = NEW /aws1/cl_rekimage(
          io_s3object = lo_s3object ).

        " Index faces in the image
        oo_result = lo_rek->indexfaces(
          iv_collectionid = iv_collection_id
          io_image = lo_image
          iv_externalimageid = iv_external_id
          iv_maxfaces = iv_max_faces ).

        DATA(lt_face_records) = oo_result->get_facerecords( ).
        DATA(lv_indexed_count) = lines( lt_face_records ).
        DATA(lv_msg2) = |{ lv_indexed_count } face(s) indexed successfully.|.
        MESSAGE lv_msg2 TYPE 'I'.
      CATCH /aws1/cx_rekresourcenotfoundex.
        MESSAGE 'Collection not found.' TYPE 'E'.
      CATCH /aws1/cx_rekinvalids3objectex.
        MESSAGE 'Invalid S3 object.' TYPE 'E'.
      CATCH /aws1/cx_rekinvalidparameterex.
        MESSAGE 'Invalid parameter value.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat IndexFacesdi AWS SDK untuk referensi SAP ABAP API.

`ListCollections`

Contoh kode berikut menunjukkan cara menggunakan`ListCollections`.

Untuk informasi selengkapnya, lihat Daftar koleksi.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_rek->listcollections(
      iv_maxresults = iv_max_results ).

    DATA(lt_collection_ids) = oo_result->get_collectionids( ).
    DATA(lv_coll_count) = lines( lt_collection_ids ).
    DATA(lv_msg7) = |{ lv_coll_count } collection(s) found.|.
    MESSAGE lv_msg7 TYPE 'I'.
  CATCH /aws1/cx_rekinvalidparameterex.
    MESSAGE 'Invalid parameter value.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat ListCollectionsdi AWS SDK untuk referensi SAP ABAP API.

`ListFaces`

Contoh kode berikut menunjukkan cara menggunakan`ListFaces`.

Untuk informasi selengkapnya, lihat Daftar wajah dalam koleksi.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_rek->listfaces(
      iv_collectionid = iv_collection_id
      iv_maxresults = iv_max_results ).

    DATA(lt_faces) = oo_result->get_faces( ).
    DATA(lv_face_count2) = lines( lt_faces ).
    DATA(lv_msg3) = |{ lv_face_count2 } face(s) found in collection.|.
    MESSAGE lv_msg3 TYPE 'I'.
  CATCH /aws1/cx_rekresourcenotfoundex.
    MESSAGE 'Collection not found.' TYPE 'E'.
  CATCH /aws1/cx_rekinvalidparameterex.
    MESSAGE 'Invalid parameter value.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat ListFacesdi AWS SDK untuk referensi SAP ABAP API.

`RecognizeCelebrities`

Contoh kode berikut menunjukkan cara menggunakan`RecognizeCelebrities`.

Untuk informasi selengkapnya, lihat Mengenali selebriti dalam sebuah gambar.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " Create S3 object reference for the image
    DATA(lo_s3object) = NEW /aws1/cl_reks3object(
      iv_bucket = iv_s3_bucket
      iv_name = iv_s3_key ).

    " Create image object
    DATA(lo_image) = NEW /aws1/cl_rekimage(
      io_s3object = lo_s3object ).
```

```
    " Recognize celebrities
    oo_result = lo_rek->recognizecelebrities(
      io_image = lo_image ).

    DATA(lt_celebrity_faces) = oo_result->get_celebrityfaces( ).
    DATA(lv_celeb_count) = lines( lt_celebrity_faces ).
    DATA(lv_msg12) = |{ lv_celeb_count } celebrity/celebrities recognized.|.
    MESSAGE lv_msg12 TYPE 'I'.
  CATCH /aws1/cx_rekinvalids3objectex.
    MESSAGE 'Invalid S3 object.' TYPE 'E'.
  CATCH /aws1/cx_rekinvalidparameterex.
    MESSAGE 'Invalid parameter value.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat RecognizeCelebritiesdi AWS SDK untuk referensi SAP ABAP API.

`SearchFaces`

Contoh kode berikut menunjukkan cara menggunakan`SearchFaces`.

Untuk informasi selengkapnya, lihat Mencari wajah (ID wajah).

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_rek->searchfaces(
      iv_collectionid = iv_collection_id
      iv_faceid = iv_face_id
      iv_facematchthreshold = iv_threshold
      iv_maxfaces = iv_max_faces ).

    DATA(lt_face_matches) = oo_result->get_facematches( ).
    DATA(lv_match_count2) = lines( lt_face_matches ).
```

```
        DATA(lv_msg5) = |Face search completed: { lv_match_count2 } match(es)
found.|.
        MESSAGE lv_msg5 TYPE 'I'.
      CATCH /aws1/cx_rekresourcenotfoundex.
        MESSAGE 'Collection or face not found.' TYPE 'E'.
      CATCH /aws1/cx_rekinvalidparameterex.
        MESSAGE 'Invalid parameter value.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat SearchFacesdi AWS SDK untuk referensi SAP ABAP API.

`SearchFacesByImage`

Contoh kode berikut menunjukkan cara menggunakan`SearchFacesByImage`.

Untuk informasi selengkapnya, lihat Mencari wajah (gambar).

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " Create S3 object reference for the image
        DATA(lo_s3object) = NEW /aws1/cl_reks3object(
          iv_bucket = iv_s3_bucket
          iv_name = iv_s3_key ).

        " Create image object
        DATA(lo_image) = NEW /aws1/cl_rekimage(
          io_s3object = lo_s3object ).

        " Search for matching faces
        oo_result = lo_rek->searchfacesbyimage(
          iv_collectionid = iv_collection_id
          io_image = lo_image
          iv_facematchthreshold = iv_threshold
          iv_maxfaces = iv_max_faces ).
```

```
        DATA(lt_face_matches) = oo_result->get_facematches( ).
        DATA(lv_match_count) = lines( lt_face_matches ).
        DATA(lv_msg4) = |Face search completed: { lv_match_count } match(es)
found.|.
        MESSAGE lv_msg4 TYPE 'I'.
      CATCH /aws1/cx_rekresourcenotfoundex.
        MESSAGE 'Collection not found.' TYPE 'E'.
      CATCH /aws1/cx_rekinvalids3objectex.
        MESSAGE 'Invalid S3 object.' TYPE 'E'.
      CATCH /aws1/cx_rekinvalidparameterex.
        MESSAGE 'Invalid parameter value.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat SearchFacesByImagedi AWS SDK untuk referensi SAP ABAP API.

Contoh Amazon S3 menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Amazon S3.

Dasar-dasar adalah contoh kode yang menunjukkan kepada Anda bagaimana melakukan operasi penting dalam suatu layanan.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Skenario adalah contoh kode yang menunjukkan kepada Anda bagaimana menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan atau dikombinasikan dengan yang lain Layanan AWS.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Hal-hal mendasar
- Tindakan
- Skenario

Hal-hal mendasar

Pelajari dasar-dasarnya

Contoh kode berikut ini menunjukkan cara untuk melakukan:

- Membuat bucket dan mengunggah file ke dalamnya.
- Mengunduh objek dari bucket.
- Menyalin objek ke subfolder di bucket.
- Membuat daftar objek dalam bucket.
- Menghapus objek bucket dan bucket tersebut.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

    " Create an Amazon Simple Storage Service (Amazon S3) bucket. "
    TRY.
        " determine our region from our session
        DATA(lv_region) = CONV /aws1/s3_bucketlocationcnstrnt( lo_session-
>get_region( ) ).
        DATA lo_constraint TYPE REF TO /aws1/cl_s3_createbucketconf.
        " When in the us-east-1 region, you must not specify a constraint
        " In all other regions, specify the region as the constraint
        IF lv_region = 'us-east-1'.
          CLEAR lo_constraint.
        ELSE.
          lo_constraint = NEW /aws1/cl_s3_createbucketconf( lv_region ).
        ENDIF.

        lo_s3->createbucket(
            iv_bucket = iv_bucket_name
            io_createbucketconfiguration  = lo_constraint ).
```

```
    MESSAGE 'S3 bucket created.' TYPE 'I'.
  CATCH /aws1/cx_s3_bucketalrdyexists.
    MESSAGE 'Bucket name already exists.' TYPE 'E'.
  CATCH /aws1/cx_s3_bktalrdyownedbyyou.
    MESSAGE 'Bucket already exists and is owned by you.' TYPE 'E'.
ENDTRY.


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

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

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

" Copy an object to a subfolder in a bucket. "
TRY.
    lo_s3->copyobject(
      iv_bucket = iv_bucket_name
      iv_key = |{ iv_copy_to_folder }/{ iv_key }|
      iv_copysource = |{ iv_bucket_name }/{ iv_key }| ).
```

```
      MESSAGE 'Object copied to a subfolder.' TYPE 'I'.
    CATCH /aws1/cx_s3_nosuchbucket.
      MESSAGE 'Bucket does not exist.' TYPE 'E'.
    CATCH /aws1/cx_s3_nosuchkey.
      MESSAGE 'Object key does not exist.' TYPE 'E'.
  ENDTRY.

  " List objects in the bucket. "
  TRY.
      DATA(lo_list) = lo_s3->listobjects(
         iv_bucket = iv_bucket_name ).
      MESSAGE 'Retrieved list of objects in S3 bucket.' TYPE 'I'.
    CATCH /aws1/cx_s3_nosuchbucket.
      MESSAGE 'Bucket does not exist.' TYPE 'E'.
  ENDTRY.
  DATA text TYPE string VALUE 'Object List - '.
  DATA lv_object_key TYPE /aws1/s3_objectkey.
  LOOP AT lo_list->get_contents( ) INTO DATA(lo_object).
    lv_object_key = lo_object->get_key( ).
    CONCATENATE lv_object_key ', ' INTO text.
  ENDLOOP.
  MESSAGE text TYPE'I'.

  " Delete the objects in a bucket. "
  TRY.
      lo_s3->deleteobject(
          iv_bucket = iv_bucket_name
          iv_key = iv_key ).
      lo_s3->deleteobject(
          iv_bucket = iv_bucket_name
          iv_key = |{ iv_copy_to_folder }/{ iv_key }| ).
      MESSAGE 'Objects deleted from S3 bucket.' TYPE 'I'.
    CATCH /aws1/cx_s3_nosuchbucket.
      MESSAGE 'Bucket does not exist.' TYPE 'E'.
  ENDTRY.


  " Delete the bucket. "
  TRY.
      lo_s3->deletebucket(
          iv_bucket = iv_bucket_name ).
      MESSAGE 'Deleted S3 bucket.' TYPE 'I'.
    CATCH /aws1/cx_s3_nosuchbucket.
      MESSAGE 'Bucket does not exist.' TYPE 'E'.
```

```
    ENDTRY.
```

- Untuk mengetahui hal detail mengenai API, silakan lihat topik-topik berikut di referensi API AWS SDK untuk ABAP SAP.
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

Tindakan

`CopyObject`

Contoh kode berikut menunjukkan cara menggunakan`CopyObject`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat CopyObjectdi AWS SDK untuk referensi SAP ABAP API.

`CreateBucket`

Contoh kode berikut menunjukkan cara menggunakan`CreateBucket`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " determine our region from our session
        DATA(lv_region) = CONV /aws1/s3_bucketlocationcnstrnt( lo_session-
>get_region( ) ).
        DATA lo_constraint TYPE REF TO /aws1/cl_s3_createbucketconf.
        " When in the us-east-1 region, you must not specify a constraint
        " In all other regions, specify the region as the constraint
        IF lv_region = 'us-east-1'.
          CLEAR lo_constraint.
        ELSE.
          lo_constraint = NEW /aws1/cl_s3_createbucketconf( lv_region ).
        ENDIF.

        lo_s3->createbucket(
            iv_bucket = iv_bucket_name
            io_createbucketconfiguration  = lo_constraint ).
        MESSAGE 'S3 bucket created.' TYPE 'I'.
      CATCH /aws1/cx_s3_bucketalrdyexists.
        MESSAGE 'Bucket name already exists.' TYPE 'E'.
      CATCH /aws1/cx_s3_bktalrdyownedbyyou.
        MESSAGE 'Bucket already exists and is owned by you.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat CreateBucketdi AWS SDK untuk referensi SAP ABAP API.

`DeleteBucket`

Contoh kode berikut menunjukkan cara menggunakan`DeleteBucket`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat DeleteBucketdi AWS SDK untuk referensi SAP ABAP API.

`DeleteBucketCors`

Contoh kode berikut menunjukkan cara menggunakan`DeleteBucketCors`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_s3->deletebucketcors(
      iv_bucket = iv_bucket_name ).
    MESSAGE 'Bucket CORS configuration deleted.' TYPE 'I'.
```

```
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DeleteBucketCorsdi AWS SDK untuk referensi SAP ABAP API.

`DeleteBucketLifecycle`

Contoh kode berikut menunjukkan cara menggunakan`DeleteBucketLifecycle`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_s3->deletebucketlifecycle(
      iv_bucket = iv_bucket_name ).
    MESSAGE 'Bucket lifecycle configuration deleted.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DeleteBucketLifecycledi AWS SDK untuk referensi SAP ABAP API.

`DeleteBucketPolicy`

Contoh kode berikut menunjukkan cara menggunakan`DeleteBucketPolicy`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_s3->deletebucketpolicy(
      iv_bucket = iv_bucket_name ).
    MESSAGE 'Bucket policy deleted.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DeleteBucketPolicydi AWS SDK untuk referensi SAP ABAP API.

`DeleteObject`

Contoh kode berikut menunjukkan cara menggunakan`DeleteObject`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat DeleteObjectdi AWS SDK untuk referensi SAP ABAP API.

`DeleteObjects`

Contoh kode berikut menunjukkan cara menggunakan`DeleteObjects`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   TRY.
       oo_result = lo_s3->deleteobjects(         " oo_result is returned for
testing purposes. "
         iv_bucket = iv_bucket_name
         io_delete = NEW /aws1/cl_s3_delete( it_objects = it_object_keys ) ).
       MESSAGE 'Objects deleted from S3 bucket.' TYPE 'I'.
     CATCH /aws1/cx_s3_nosuchbucket.
       MESSAGE 'Bucket does not exist.' TYPE 'E'.
   ENDTRY.
```

- Untuk detail API, lihat DeleteObjectsdi AWS SDK untuk referensi SAP ABAP API.

`GetBucketAcl`

Contoh kode berikut menunjukkan cara menggunakan`GetBucketAcl`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   TRY.
       oo_result = lo_s3->getbucketacl(         " oo_result is returned for testing
purposes. "
         iv_bucket = iv_bucket_name ).
       MESSAGE 'Retrieved bucket ACL.' TYPE 'I'.
     CATCH /aws1/cx_s3_nosuchbucket.
       MESSAGE 'Bucket does not exist.' TYPE 'E'.
```

```
    ENDTRY.
```

- Untuk detail API, lihat GetBucketAcldi AWS SDK untuk referensi SAP ABAP API.

`GetBucketCors`

Contoh kode berikut menunjukkan cara menggunakan`GetBucketCors`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   TRY.
       oo_result = lo_s3->getbucketcors(         " oo_result is returned for
testing purposes. "
         iv_bucket = iv_bucket_name ).
       MESSAGE 'Retrieved bucket CORS configuration.' TYPE 'I'.
     CATCH /aws1/cx_s3_nosuchbucket.
       MESSAGE 'Bucket does not exist.' TYPE 'E'.
   ENDTRY.
```

- Untuk detail API, lihat GetBucketCorsdi AWS SDK untuk referensi SAP ABAP API.

`GetBucketLifecycleConfiguration`

Contoh kode berikut menunjukkan cara menggunakan`GetBucketLifecycleConfiguration`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   TRY.
       oo_result = lo_s3->getbucketlifecycleconf(         " oo_result is returned
for testing purposes. "
         iv_bucket = iv_bucket_name ).
       MESSAGE 'Retrieved bucket lifecycle configuration.' TYPE 'I'.
     CATCH /aws1/cx_s3_nosuchbucket.
       MESSAGE 'Bucket does not exist.' TYPE 'E'.
   ENDTRY.
```

- Untuk detail API, lihat GetBucketLifecycleConfigurationdi AWS SDK untuk referensi SAP ABAP API.

`GetBucketPolicy`

Contoh kode berikut menunjukkan cara menggunakan`GetBucketPolicy`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   TRY.
       oo_result = lo_s3->getbucketpolicy(         " oo_result is returned for
testing purposes. "
         iv_bucket = iv_bucket_name ).
       DATA(lv_policy) = oo_result->get_policy( ).
       MESSAGE 'Retrieved bucket policy.' TYPE 'I'.
     CATCH /aws1/cx_s3_nosuchbucket.
       MESSAGE 'Bucket does not exist.' TYPE 'E'.
   ENDTRY.
```

- Untuk detail API, lihat GetBucketPolicydi AWS SDK untuk referensi SAP ABAP API.

`GetObject`

Contoh kode berikut menunjukkan cara menggunakan`GetObject`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat GetObjectdi AWS SDK untuk referensi SAP ABAP API.

`GetObjectAcl`

Contoh kode berikut menunjukkan cara menggunakan`GetObjectAcl`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_s3->getobjectacl(         " oo_result is returned for testing
purposes. "
          iv_bucket = iv_bucket_name
          iv_key = iv_object_key ).
        MESSAGE 'Retrieved object ACL.' TYPE 'I'.
      CATCH /aws1/cx_s3_nosuchbucket.
        MESSAGE 'Bucket does not exist.' TYPE 'E'.
      CATCH /aws1/cx_s3_nosuchkey.
        MESSAGE 'Object key does not exist.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat GetObjectAcldi AWS SDK untuk referensi SAP ABAP API.

`GetObjectLegalHold`

Contoh kode berikut menunjukkan cara menggunakan`GetObjectLegalHold`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_s3->getobjectlegalhold(         " oo_result is returned for
testing purposes. "
          iv_bucket = iv_bucket_name
          iv_key = iv_object_key ).
        MESSAGE 'Retrieved object legal hold status.' TYPE 'I'.
      CATCH /aws1/cx_s3_nosuchbucket.
        MESSAGE 'Bucket does not exist.' TYPE 'E'.
      CATCH /aws1/cx_s3_nosuchkey.
        MESSAGE 'Object key does not exist.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat GetObjectLegalHolddi AWS SDK untuk referensi SAP ABAP API.

`GetObjectLockConfiguration`

Contoh kode berikut menunjukkan cara menggunakan`GetObjectLockConfiguration`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   TRY.
       oo_result = lo_s3->getobjectlockconfiguration(         " oo_result is
returned for testing purposes. "
         iv_bucket = iv_bucket_name ).
       MESSAGE 'Retrieved object lock configuration.' TYPE 'I'.
     CATCH /aws1/cx_s3_nosuchbucket.
       MESSAGE 'Bucket does not exist.' TYPE 'E'.
   ENDTRY.
```

- Untuk detail API, lihat GetObjectLockConfigurationdi AWS SDK untuk referensi SAP ABAP API.

`HeadBucket`

Contoh kode berikut menunjukkan cara menggunakan`HeadBucket`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   TRY.
       oo_result = lo_s3->headbucket(         " oo_result is returned for testing
purposes. "
```

```
          iv_bucket = iv_bucket_name ).
        MESSAGE 'Bucket exists and you have access to it.' TYPE 'I'.
      CATCH /aws1/cx_s3_nosuchbucket.
        MESSAGE 'Bucket does not exist.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat HeadBucketdi AWS SDK untuk referensi SAP ABAP API.

`ListBuckets`

Contoh kode berikut menunjukkan cara menggunakan`ListBuckets`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_s3->listbuckets(         " oo_result is returned for testing
purposes. "
        ).
        DATA(lv_bucket_count) = lines( oo_result->get_buckets( ) ).
        MESSAGE |Retrieved { lv_bucket_count } buckets in all regions.| TYPE 'I'.
      CATCH /aws1/cx_rt_generic.
        MESSAGE 'Unable to list buckets.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat ListBucketsdi AWS SDK untuk referensi SAP ABAP API.

`ListObjectVersions`

Contoh kode berikut menunjukkan cara menggunakan`ListObjectVersions`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_s3->listobjectversions(         " oo_result is returned for
testing purposes. "
          iv_bucket = iv_bucket_name
          iv_prefix = iv_prefix ).
        MESSAGE 'Retrieved object versions.' TYPE 'I'.
      CATCH /aws1/cx_s3_nosuchbucket.
        MESSAGE 'Bucket does not exist.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat ListObjectVersionsdi AWS SDK untuk referensi SAP ABAP API.

`ListObjectsV2`

Contoh kode berikut menunjukkan cara menggunakan`ListObjectsV2`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat ListObjectsV2 di AWS SDK untuk referensi SAP ABAP API.

`PutBucketAcl`

Contoh kode berikut menunjukkan cara menggunakan`PutBucketAcl`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " Example: Grant log delivery access to a bucket
    " iv_grantwrite = 'uri=http://acs.amazonaws.com/groups/s3/LogDelivery'
    lo_s3->putbucketacl(
      iv_bucket = iv_bucket_name
      iv_grantwrite = iv_grantwrite ).
    MESSAGE 'Bucket ACL updated.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat PutBucketAcldi AWS SDK untuk referensi SAP ABAP API.

`PutBucketCors`

Contoh kode berikut menunjukkan cara menggunakan`PutBucketCors`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " Example: Allow PUT, POST, DELETE methods from http://www.example.com
    lo_s3->putbucketcors(
      iv_bucket = iv_bucket_name
      io_corsconfiguration = NEW /aws1/cl_s3_corsconfiguration(
        it_corsrules = it_cors_rules ) ).
    MESSAGE 'Bucket CORS configuration set.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat PutBucketCorsdi AWS SDK untuk referensi SAP ABAP API.

`PutBucketLifecycleConfiguration`

Contoh kode berikut menunjukkan cara menggunakan`PutBucketLifecycleConfiguration`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " Example: Expire objects with prefix 'logs/' after 30 days
    lo_s3->putbucketlifecycleconf(
      iv_bucket = iv_bucket_name
      io_lifecycleconfiguration = NEW /aws1/cl_s3_bucketlcconf(
        it_rules = it_lifecycle_rule ) ).
```

```
        MESSAGE 'Bucket lifecycle configuration set.' TYPE 'I'.
      CATCH /aws1/cx_s3_nosuchbucket.
        MESSAGE 'Bucket does not exist.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat PutBucketLifecycleConfigurationdi AWS SDK untuk referensi SAP ABAP API.

`PutBucketPolicy`

Contoh kode berikut menunjukkan cara menggunakan`PutBucketPolicy`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " Example policy JSON string
        " iv_policy = '{"Version":"2012-10-17",       "Statement":
[{"Effect":"Allow","Principal":{"AWS":"arn:aws:iam::123456789012:user/
user"},"Action":["s3:GetObject"],"Resource":["arn:aws:s3:::bucketname/*"]}]}'
        lo_s3->putbucketpolicy(
          iv_bucket = iv_bucket_name
          iv_policy = iv_policy ).
        MESSAGE 'Bucket policy set.' TYPE 'I'.
      CATCH /aws1/cx_s3_nosuchbucket.
        MESSAGE 'Bucket does not exist.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat PutBucketPolicydi AWS SDK untuk referensi SAP ABAP API.

`PutBucketVersioning`

Contoh kode berikut menunjukkan cara menggunakan`PutBucketVersioning`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " Example: Enable versioning on a bucket
    " iv_status = 'Enabled'
    lo_s3->putbucketversioning(
      iv_bucket = iv_bucket_name
      io_versioningconfiguration = NEW /aws1/cl_s3_versioningconf(
        iv_status = iv_status ) ).
    MESSAGE 'Bucket versioning enabled.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat PutBucketVersioningdi AWS SDK untuk referensi SAP ABAP API.

`PutObject`

Contoh kode berikut menunjukkan cara menggunakan`PutObject`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat PutObjectdi AWS SDK untuk referensi SAP ABAP API.

`PutObjectAcl`

Contoh kode berikut menunjukkan cara menggunakan`PutObjectAcl`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " Example: Grant read access to an AWS user
        " iv_grantread = 'emailAddress=user@example.com'
        lo_s3->putobjectacl(
          iv_bucket = iv_bucket_name
          iv_key = iv_object_key
          iv_grantread = iv_grantread ).
        MESSAGE 'Object ACL updated.' TYPE 'I'.
      CATCH /aws1/cx_s3_nosuchbucket.
        MESSAGE 'Bucket does not exist.' TYPE 'E'.
      CATCH /aws1/cx_s3_nosuchkey.
```

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

- Untuk detail API, lihat PutObjectAcldi AWS SDK untuk referensi SAP ABAP API.

`PutObjectLegalHold`

Contoh kode berikut menunjukkan cara menggunakan`PutObjectLegalHold`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " Example: Set legal hold status to ON
    " iv_status = 'ON'
    lo_s3->putobjectlegalhold(
      iv_bucket = iv_bucket_name
      iv_key = iv_object_key
      io_legalhold = NEW /aws1/cl_s3_objlocklegalhold(
        iv_status = iv_status ) ).
    MESSAGE 'Object legal hold status set.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
  CATCH /aws1/cx_s3_nosuchkey.
    MESSAGE 'Object key does not exist.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat PutObjectLegalHolddi AWS SDK untuk referensi SAP ABAP API.

`PutObjectLockConfiguration`

Contoh kode berikut menunjukkan cara menggunakan`PutObjectLockConfiguration`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " Example: Enable object lock with default retention
    " iv_enabled = 'Enabled'
    lo_s3->putobjectlockconfiguration(
      iv_bucket = iv_bucket_name
      io_objectlockconfiguration = NEW /aws1/cl_s3_objectlockconf(
        iv_objectlockenabled = iv_enabled ) ).
    MESSAGE 'Object lock configuration set.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat PutObjectLockConfigurationdi AWS SDK untuk referensi SAP ABAP API.

`PutObjectRetention`

Contoh kode berikut menunjukkan cara menggunakan`PutObjectRetention`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " Example: Set retention mode to GOVERNANCE for 30 days
    " iv_mode = 'GOVERNANCE'
    " iv_retain_date should be a timestamp in the future
```

```
      lo_s3->putobjectretention(
        iv_bucket = iv_bucket_name
        iv_key = iv_object_key
        io_retention = NEW /aws1/cl_s3_objectlockret(
          iv_mode = iv_mode
          iv_retainuntildate = iv_retain_date )
        iv_bypassgovernanceretention = abap_true ).
      MESSAGE 'Object retention set.' TYPE 'I'.
    CATCH /aws1/cx_s3_nosuchbucket.
      MESSAGE 'Bucket does not exist.' TYPE 'E'.
    CATCH /aws1/cx_s3_nosuchkey.
      MESSAGE 'Object key does not exist.' TYPE 'E'.
  ENDTRY.
```

- Untuk detail API, lihat PutObjectRetentiondi AWS SDK untuk referensi SAP ABAP API.

Skenario

Membuat URL yang telah ditetapkan sebelumnya

Contoh kode berikut menunjukkan cara membuat URL presigned untuk Amazon S3 dan mengunggah objek.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

Buat permintaan yang telah ditetapkan sebelumnya ke objek GET S3.

```
  " iv_bucket_name is the bucket name
  " iv_key is the object name like "myfile.txt"

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

  "Upload a nice Hello World file to an S3 bucket."
```

```
TRY.
    DATA(lv_contents) = cl_abap_codepage=>convert_to( 'Hello, World' ).
    lo_s3->putobject(
        iv_bucket = iv_bucket_name
        iv_key = iv_key
        iv_body = lv_contents
        iv_contenttype = 'text/plain' ).
    MESSAGE 'Object uploaded to S3 bucket.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.

" now generate a presigned URL with a 600-second expiration
DATA(lo_presigner) = lo_s3->get_presigner( iv_expires_sec = 600 ).
" the presigner getobject() method has the same signature as
" lo_s3->getobject(), but it doesn't actually make the call.
" to the service.  It just prepares a presigned URL for a future call
DATA(lo_presigned_req) = lo_presigner->getobject(
  iv_bucket = iv_bucket_name
  iv_key = iv_key ).

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

Contoh Kontrol Amazon S3 menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Amazon S3 Control.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`CreateJob`

Contoh kode berikut menunjukkan cara menggunakan`CreateJob`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_manifest_arn  = 'arn:aws:s3:::my-bucket/job-manifest.csv'
        " iv_manifest_etag = 'abc123def456'
        " iv_report_bucket = 'arn:aws:s3:::my-report-bucket'
        DATA(lo_result) = lo_s3c->createjob(
          iv_accountid            = iv_account_id
          iv_rolearn              = iv_role_arn
          iv_confirmationrequired = abap_true
          iv_priority             = 10
          iv_description          = 'Batch job for tagging objects'
          io_operation            = NEW /aws1/cl_s3cjoboperation(
            io_s3putobjecttagging = NEW /aws1/cl_s3cs3setobjecttagop(
              it_tagset = VALUE /aws1/cl_s3cs3tag=>tt_s3tagset(
                ( NEW /aws1/cl_s3cs3tag(
                    iv_key   = 'BatchTag'
                    iv_value = 'BatchValue' ) )
              )
            )
          )
          io_manifest             = NEW /aws1/cl_s3cjobmanifest(
            io_spec     = NEW /aws1/cl_s3cjobmanifestspec(
              iv_format = 'S3BatchOperations_CSV_20180820'
              it_fields = VALUE /aws1/
cl_s3cjobmanifestfield00=>tt_jobmanifestfieldlist(
                ( NEW /aws1/cl_s3cjobmanifestfield00( 'Bucket' ) )
                ( NEW /aws1/cl_s3cjobmanifestfield00( 'Key' ) )
              )
            )
```

```
            io_location = NEW /aws1/cl_s3cjobmanifestloc(
              iv_objectarn = iv_manifest_arn
              iv_etag      = iv_manifest_etag
            )
          )
          io_report               = NEW /aws1/cl_s3cjobreport(
            iv_bucket      = iv_report_bucket
            iv_format      = 'Report_CSV_20180820'
            iv_enabled     = abap_true
            iv_prefix      = 'batch-op-reports'
            iv_reportscope = 'AllTasks'
          )
        ).
        ov_job_id = lo_result->get_jobid( ).
        MESSAGE |S3 Batch Operations job created: { ov_job_id }| TYPE 'I'.
      CATCH /aws1/cx_s3cbadrequestex INTO DATA(lo_ex_bad).
        MESSAGE lo_ex_bad->get_text( ) TYPE 'I'.
        RAISE EXCEPTION TYPE /aws1/cx_rt_generic
          EXPORTING previous = lo_ex_bad.
      CATCH /aws1/cx_s3cclientexc INTO DATA(lo_ex_cli).
        MESSAGE lo_ex_cli->get_text( ) TYPE 'I'.
        RAISE EXCEPTION TYPE /aws1/cx_rt_generic
          EXPORTING previous = lo_ex_cli.
      CATCH /aws1/cx_s3cserverexc INTO DATA(lo_ex_srv).
        MESSAGE lo_ex_srv->get_text( ) TYPE 'I'.
        RAISE EXCEPTION TYPE /aws1/cx_rt_generic
          EXPORTING previous = lo_ex_srv.
    ENDTRY.
```

- Untuk detail API, lihat CreateJobdi AWS SDK untuk referensi SAP ABAP API.

`DeleteJobTagging`

Contoh kode berikut menunjukkan cara menggunakan`DeleteJobTagging`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        lo_s3c->deletejobtagging(
          iv_accountid = iv_account_id
          iv_jobid     = iv_job_id
        ).
        MESSAGE |Tags deleted from job { iv_job_id }| TYPE 'I'.
      CATCH /aws1/cx_s3cnotfoundexception INTO DATA(lo_ex_nf).
        MESSAGE lo_ex_nf->get_text( ) TYPE 'I'.
        RAISE EXCEPTION TYPE /aws1/cx_rt_generic
          EXPORTING previous = lo_ex_nf.
      CATCH /aws1/cx_s3cclientexc INTO DATA(lo_ex_cli).
        MESSAGE lo_ex_cli->get_text( ) TYPE 'I'.
        RAISE EXCEPTION TYPE /aws1/cx_rt_generic
          EXPORTING previous = lo_ex_cli.
      CATCH /aws1/cx_s3cserverexc INTO DATA(lo_ex_srv).
        MESSAGE lo_ex_srv->get_text( ) TYPE 'I'.
        RAISE EXCEPTION TYPE /aws1/cx_rt_generic
          EXPORTING previous = lo_ex_srv.
    ENDTRY.
```

- Untuk detail API, lihat DeleteJobTaggingdi AWS SDK untuk referensi SAP ABAP API.

`DescribeJob`

Contoh kode berikut menunjukkan cara menggunakan`DescribeJob`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_s3c->describejob(         " oo_result is returned for testing
purposes.
          iv_accountid = iv_account_id
          iv_jobid     = iv_job_id
        ).
```

```
        DATA(lo_job) = oo_result->get_job( ).
        DATA(lv_status) = lo_job->get_status( ).
        DATA(lv_priority) = lo_job->get_priority( ).
        DATA(lo_progress) = lo_job->get_progresssummary( ).
        IF lo_progress IS NOT INITIAL.
          MESSAGE |Job { iv_job_id }: status={ lv_status },
priority={ lv_priority }, | &&
                  |total={ lo_progress->get_totalnumberoftasks( ) }, | &&
                  |succeeded={ lo_progress->get_numberoftaskssucceeded( ) }, | &&
                  |failed={ lo_progress->get_numberoftasksfailed( ) }| TYPE 'I'.
        ELSE.
          MESSAGE |Job { iv_job_id }: status={ lv_status },
priority={ lv_priority }| TYPE 'I'.
        ENDIF.
      CATCH /aws1/cx_s3cnotfoundexception INTO DATA(lo_ex_nf).
        MESSAGE lo_ex_nf->get_text( ) TYPE 'I'.
        RAISE EXCEPTION TYPE /aws1/cx_rt_generic
          EXPORTING previous = lo_ex_nf.
      CATCH /aws1/cx_s3cclientexc INTO DATA(lo_ex_cli).
        MESSAGE lo_ex_cli->get_text( ) TYPE 'I'.
        RAISE EXCEPTION TYPE /aws1/cx_rt_generic
          EXPORTING previous = lo_ex_cli.
      CATCH /aws1/cx_s3cserverexc INTO DATA(lo_ex_srv).
        MESSAGE lo_ex_srv->get_text( ) TYPE 'I'.
        RAISE EXCEPTION TYPE /aws1/cx_rt_generic
          EXPORTING previous = lo_ex_srv.
    ENDTRY.
```

- Untuk detail API, lihat DescribeJobdi AWS SDK untuk referensi SAP ABAP API.

`GetJobTagging`

Contoh kode berikut menunjukkan cara menggunakan`GetJobTagging`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   TRY.
       oo_result = lo_s3c->getjobtagging(       " oo_result is returned for testing
purposes.
         iv_accountid = iv_account_id
         iv_jobid     = iv_job_id
       ).
       DATA(lt_tags) = oo_result->get_tags( ).
       MESSAGE |Retrieved { lines( lt_tags ) } tag(s) for job { iv_job_id }| TYPE
'I'.
     CATCH /aws1/cx_s3cnotfoundexception INTO DATA(lo_ex_nf).
       MESSAGE lo_ex_nf->get_text( ) TYPE 'I'.
       RAISE EXCEPTION TYPE /aws1/cx_rt_generic
         EXPORTING previous = lo_ex_nf.
     CATCH /aws1/cx_s3cclientexc INTO DATA(lo_ex_cli).
       MESSAGE lo_ex_cli->get_text( ) TYPE 'I'.
       RAISE EXCEPTION TYPE /aws1/cx_rt_generic
         EXPORTING previous = lo_ex_cli.
     CATCH /aws1/cx_s3cserverexc INTO DATA(lo_ex_srv).
       MESSAGE lo_ex_srv->get_text( ) TYPE 'I'.
       RAISE EXCEPTION TYPE /aws1/cx_rt_generic
         EXPORTING previous = lo_ex_srv.
   ENDTRY.
```

- Untuk detail API, lihat GetJobTaggingdi AWS SDK untuk referensi SAP ABAP API.

`PutJobTagging`

Contoh kode berikut menunjukkan cara menggunakan`PutJobTagging`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   TRY.
       lo_s3c->putjobtagging(
         iv_accountid = iv_account_id
```

```
        iv_jobid     = iv_job_id
        it_tags      = VALUE /aws1/cl_s3cs3tag=>tt_s3tagset(
          ( NEW /aws1/cl_s3cs3tag(
              iv_key   = 'Environment'
              iv_value = 'Development' ) )
          ( NEW /aws1/cl_s3cs3tag(
              iv_key   = 'Team'
              iv_value = 'DataProcessing' ) )
        )
      ).
      MESSAGE |Tags added to job { iv_job_id }| TYPE 'I'.
    CATCH /aws1/cx_s3cnotfoundexception INTO DATA(lo_ex_nf).
      MESSAGE lo_ex_nf->get_text( ) TYPE 'I'.
      RAISE EXCEPTION TYPE /aws1/cx_rt_generic
        EXPORTING previous = lo_ex_nf.
    CATCH /aws1/cx_s3ctoomanytagsex INTO DATA(lo_ex_tags).
      MESSAGE lo_ex_tags->get_text( ) TYPE 'I'.
      RAISE EXCEPTION TYPE /aws1/cx_rt_generic
        EXPORTING previous = lo_ex_tags.
    CATCH /aws1/cx_s3cclientexc INTO DATA(lo_ex_cli).
      MESSAGE lo_ex_cli->get_text( ) TYPE 'I'.
      RAISE EXCEPTION TYPE /aws1/cx_rt_generic
        EXPORTING previous = lo_ex_cli.
    CATCH /aws1/cx_s3cserverexc INTO DATA(lo_ex_srv).
      MESSAGE lo_ex_srv->get_text( ) TYPE 'I'.
      RAISE EXCEPTION TYPE /aws1/cx_rt_generic
        EXPORTING previous = lo_ex_srv.
  ENDTRY.
```

- Untuk detail API, lihat PutJobTaggingdi AWS SDK untuk referensi SAP ABAP API.

`UpdateJobPriority`

Contoh kode berikut menunjukkan cara menggunakan`UpdateJobPriority`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_s3c->updatejobpriority(   " oo_result is returned for testing
 purposes.
          iv_accountid = iv_account_id
          iv_jobid     = iv_job_id
          iv_priority  = 60
        ).
        MESSAGE |Job { oo_result->get_jobid( ) } priority updated to { oo_result-
>get_priority( ) }| TYPE 'I'.
      CATCH /aws1/cx_s3cnotfoundexception INTO DATA(lo_ex_nf).
        MESSAGE lo_ex_nf->get_text( ) TYPE 'I'.
        RAISE EXCEPTION TYPE /aws1/cx_rt_generic
          EXPORTING previous = lo_ex_nf.
      CATCH /aws1/cx_s3cbadrequestex INTO DATA(lo_ex_bad).
        MESSAGE lo_ex_bad->get_text( ) TYPE 'I'.
        RAISE EXCEPTION TYPE /aws1/cx_rt_generic
          EXPORTING previous = lo_ex_bad.
      CATCH /aws1/cx_s3cclientexc INTO DATA(lo_ex_cli).
        MESSAGE lo_ex_cli->get_text( ) TYPE 'I'.
        RAISE EXCEPTION TYPE /aws1/cx_rt_generic
          EXPORTING previous = lo_ex_cli.
      CATCH /aws1/cx_s3cserverexc INTO DATA(lo_ex_srv).
        MESSAGE lo_ex_srv->get_text( ) TYPE 'I'.
        RAISE EXCEPTION TYPE /aws1/cx_rt_generic
          EXPORTING previous = lo_ex_srv.
    ENDTRY.
```

- Untuk detail API, lihat UpdateJobPrioritydi AWS SDK untuk referensi SAP ABAP API.

`UpdateJobStatus`

Contoh kode berikut menunjukkan cara menggunakan`UpdateJobStatus`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_requested_status = 'Cancelled'
        oo_result = lo_s3c->updatejobstatus(     " oo_result is returned for testing
 purposes.
          iv_accountid          = iv_account_id
          iv_jobid              = iv_job_id
          iv_requestedjobstatus = iv_requested_status
        ).
        MESSAGE |Job { oo_result->get_jobid( ) } status updated to { oo_result-
>get_status( ) }| TYPE 'I'.
      CATCH /aws1/cx_s3cjobstatusexception INTO DATA(lo_ex_js).
        MESSAGE lo_ex_js->get_text( ) TYPE 'I'.
        RAISE EXCEPTION TYPE /aws1/cx_rt_generic
          EXPORTING previous = lo_ex_js.
      CATCH /aws1/cx_s3cnotfoundexception INTO DATA(lo_ex_nf).
        MESSAGE lo_ex_nf->get_text( ) TYPE 'I'.
        RAISE EXCEPTION TYPE /aws1/cx_rt_generic
          EXPORTING previous = lo_ex_nf.
      CATCH /aws1/cx_s3cclientexc INTO DATA(lo_ex_cli).
        MESSAGE lo_ex_cli->get_text( ) TYPE 'I'.
        RAISE EXCEPTION TYPE /aws1/cx_rt_generic
          EXPORTING previous = lo_ex_cli.
      CATCH /aws1/cx_s3cserverexc INTO DATA(lo_ex_srv).
        MESSAGE lo_ex_srv->get_text( ) TYPE 'I'.
        RAISE EXCEPTION TYPE /aws1/cx_rt_generic
          EXPORTING previous = lo_ex_srv.
    ENDTRY.
```

- Untuk detail API, lihat UpdateJobStatusdi AWS SDK untuk referensi SAP ABAP API.

SageMaker Contoh AI menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan AI. SageMaker

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Skenario adalah contoh kode yang menunjukkan kepada Anda bagaimana menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan atau dikombinasikan dengan yang lain Layanan AWS.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan
- Skenario

Tindakan

`CreateEndpoint`

Contoh kode berikut menunjukkan cara menggunakan`CreateEndpoint`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

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

```
    INSERT lo_production_variants INTO TABLE lt_production_variants.

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

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

- Untuk detail API, lihat CreateEndpointdi AWS SDK untuk referensi SAP ABAP API.

`CreateModel`

Contoh kode berikut menunjukkan cara menggunakan`CreateModel`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    DATA lo_primarycontainer TYPE REF TO /aws1/cl_sgmcontainerdefn.
```

```
    "Create an ABAP object for the container image based on input variables."
    lo_primarycontainer = NEW #( iv_image = iv_container_image
                                 iv_modeldataurl = iv_model_data_url ).

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

- Untuk detail API, lihat CreateModeldi AWS SDK untuk referensi SAP ABAP API.

`CreateTrainingJob`

Contoh kode berikut menunjukkan cara menggunakan`CreateTrainingJob`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    DATA lo_hyperparameters_w TYPE REF TO /aws1/cl_sgmhyperparameters_w.
    DATA lt_hyperparameters TYPE /aws1/cl_sgmhyperparameters_w=>tt_hyperparameters.
    DATA lt_input_data_config TYPE /aws1/cl_sgmchannel=>tt_inputdataconfig.
    DATA lo_trn_channel TYPE REF TO /aws1/cl_sgmchannel.
    DATA lo_trn_datasource TYPE REF TO /aws1/cl_sgmdatasource.
    DATA lo_trn_s3datasource TYPE REF TO /aws1/cl_sgms3datasource.
    DATA lo_val_channel TYPE REF TO /aws1/cl_sgmchannel.
    DATA lo_val_datasource TYPE REF TO /aws1/cl_sgmdatasource.
    DATA lo_val_s3datasource TYPE REF TO /aws1/cl_sgms3datasource.
    DATA lo_algorithm_specification TYPE REF TO /aws1/cl_sgmalgorithmspec.
```

```
   DATA lo_resource_config  TYPE REF TO /aws1/cl_sgmresourceconfig.
   DATA lo_output_data_config TYPE REF TO /aws1/cl_sgmoutputdataconfig.
   DATA lo_stopping_condition TYPE REF TO /aws1/cl_sgmstoppingcondition.

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

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

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

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

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

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

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

   "Create ABAP objects for training data sources."
   lo_trn_s3datasource = NEW #( iv_s3datatype = iv_trn_data_s3datatype
                                iv_s3datadistributiontype =
iv_trn_data_s3datadistribution
                                iv_s3uri = iv_trn_data_s3uri ).

   lo_trn_datasource = NEW #( io_s3datasource = lo_trn_s3datasource ).

   lo_trn_channel = NEW #( iv_channelname = 'train'
```

```
                            io_datasource = lo_trn_datasource
                            iv_compressiontype = iv_trn_data_compressiontype
                            iv_contenttype = iv_trn_data_contenttype ).

   INSERT lo_trn_channel INTO TABLE lt_input_data_config.

   "Create ABAP objects for validation data sources."
   lo_val_s3datasource = NEW #( iv_s3datatype = iv_val_data_s3datatype
                                iv_s3datadistributiontype =
iv_val_data_s3datadistribution
                                iv_s3uri = iv_val_data_s3uri ).

   lo_val_datasource = NEW #( io_s3datasource = lo_val_s3datasource ).

   lo_val_channel = NEW #( iv_channelname = 'validation'
                            io_datasource = lo_val_datasource
                            iv_compressiontype = iv_val_data_compressiontype
                            iv_contenttype = iv_val_data_contenttype ).

   INSERT lo_val_channel INTO TABLE lt_input_data_config.

   "Create an ABAP object for algorithm specification."
   lo_algorithm_specification = NEW #( iv_trainingimage = iv_training_image
                                       iv_traininginputmode =
iv_training_input_mode ).

   "Create an ABAP object for resource configuration."
   lo_resource_config = NEW #( iv_instancecount = iv_instance_count
                               iv_instancetype = iv_instance_type
                               iv_volumesizeingb = iv_volume_sizeingb ).

   "Create an ABAP object for output data configuration."
   lo_output_data_config = NEW #( iv_s3outputpath = iv_s3_output_path ).

   "Create an ABAP object for stopping condition."
   lo_stopping_condition = NEW #( iv_maxruntimeinseconds =
iv_max_runtime_in_seconds ).

   "Create a training job."
   TRY.
       oo_result = lo_sgm->createtrainingjob(    " oo_result is returned for
testing purposes. "
         iv_trainingjobname          = iv_training_job_name
         iv_rolearn                  = iv_role_arn
```

```
          it_hyperparameters           = lt_hyperparameters
          it_inputdataconfig           = lt_input_data_config
          io_algorithmspecification    = lo_algorithm_specification
          io_outputdataconfig          = lo_output_data_config
          io_resourceconfig            = lo_resource_config
          io_stoppingcondition         = lo_stopping_condition ).
        MESSAGE 'Training job created.' TYPE 'I'.
      CATCH /aws1/cx_sgmresourceinuse.
        MESSAGE 'Resource being accessed is in use.' TYPE 'E'.
      CATCH /aws1/cx_sgmresourcenotfound.
        MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
      CATCH /aws1/cx_sgmresourcelimitexcd.
        MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat CreateTrainingJobdi AWS SDK untuk referensi SAP ABAP API.

`CreateTransformJob`

Contoh kode berikut menunjukkan cara menggunakan`CreateTransformJob`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

    "Create an ABAP object for an Amazon Simple Storage Service (Amazon S3) data
source."
    lo_s3datasource = NEW #( iv_s3uri = iv_tf_data_s3uri
                             iv_s3datatype = iv_tf_data_s3datatype ).
```

```
   "Create an ABAP object for data source."
   lo_datasource = NEW #( io_s3datasource = lo_s3datasource ).

   "Create an ABAP object for transform data source."
   lo_transforminput = NEW #( io_datasource = lo_datasource
                              iv_contenttype = iv_tf_data_contenttype
                              iv_compressiontype = iv_tf_data_compressiontype ).

   "Create an ABAP object for resource configuration."
   lo_transformresources = NEW #( iv_instancecount = iv_instance_count
                                  iv_instancetype = iv_instance_type ).

   "Create an ABAP object for output data configuration."
   lo_transformoutput = NEW #( iv_s3outputpath = iv_s3_output_path ).

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

- Untuk detail API, lihat CreateTransformJobdi AWS SDK untuk referensi SAP ABAP API.

`DeleteEndpoint`

Contoh kode berikut menunjukkan cara menggunakan`DeleteEndpoint`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat DeleteEndpointdi AWS SDK untuk referensi SAP ABAP API.

`DeleteModel`

Contoh kode berikut menunjukkan cara menggunakan`DeleteModel`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat DeleteModeldi AWS SDK untuk referensi SAP ABAP API.

`DescribeTrainingJob`

Contoh kode berikut menunjukkan cara menggunakan`DescribeTrainingJob`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat DescribeTrainingJobdi AWS SDK untuk referensi SAP ABAP API.

`ListAlgorithms`

Contoh kode berikut menunjukkan cara menggunakan`ListAlgorithms`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat ListAlgorithmsdi AWS SDK untuk referensi SAP ABAP API.

`ListModels`

Contoh kode berikut menunjukkan cara menggunakan`ListModels`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat ListModelsdi AWS SDK untuk referensi SAP ABAP API.

`ListNotebookInstances`

Contoh kode berikut menunjukkan cara menggunakan`ListNotebookInstances`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat ListNotebookInstancesdi AWS SDK untuk referensi SAP ABAP API.

`ListTrainingJobs`

Contoh kode berikut menunjukkan cara menggunakan`ListTrainingJobs`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat ListTrainingJobsdi AWS SDK untuk referensi SAP ABAP API.

Skenario

Memulai dengan model dan titik akhir

Contoh kode berikut ini menunjukkan cara untuk melakukan:

- Mulai pekerjaan pelatihan dan buat model SageMaker AI.
- Buat konfigurasi titik akhir.
- Buat titik akhir, lalu bersihkan sumber daya.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    DATA lo_hyperparameters_w TYPE REF TO /aws1/cl_sgmhyperparameters_w.
    DATA lo_trn_channel TYPE REF TO /aws1/cl_sgmchannel.
    DATA lo_trn_datasource TYPE REF TO /aws1/cl_sgmdatasource.
    DATA lo_trn_s3datasource TYPE REF TO /aws1/cl_sgms3datasource.
    DATA lo_val_channel TYPE REF TO /aws1/cl_sgmchannel.
    DATA lo_val_datasource TYPE REF TO /aws1/cl_sgmdatasource.
    DATA lo_val_s3datasource TYPE REF TO /aws1/cl_sgms3datasource.
    DATA lo_algorithm_specification TYPE REF TO /aws1/cl_sgmalgorithmspec.
    DATA lo_resource_config  TYPE REF TO /aws1/cl_sgmresourceconfig.
    DATA lo_output_data_config TYPE REF TO /aws1/cl_sgmoutputdataconfig.
    DATA lo_stopping_condition TYPE REF TO /aws1/cl_sgmstoppingcondition.
    DATA lo_primarycontainer TYPE REF TO /aws1/cl_sgmcontainerdefn.
    DATA lo_production_variants TYPE REF TO /aws1/cl_sgmproductionvariant.
    DATA lo_ep_config_result TYPE REF TO /aws1/cl_sgmcreateendptcfgout.
    DATA lo_training_result TYPE REF TO /aws1/cl_sgmdescrtrnjobrsp.
    DATA lt_production_variants TYPE /aws1/
cl_sgmproductionvariant=>tt_productionvariantlist.
    DATA lt_input_data_config TYPE /aws1/cl_sgmchannel=>tt_inputdataconfig.
    DATA lt_hyperparameters TYPE /aws1/cl_sgmhyperparameters_w=>tt_hyperparameters.
    DATA lv_model_data_url TYPE /aws1/sgmurl.

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

    "Create ABAP internal table for hyperparameters based on input variables."
    "These hyperparameters are based on Amazon SageMaker built-in algorithm -
 XGBoost"
    lo_hyperparameters_w = NEW #( iv_value = iv_hp_max_depth ).
```

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

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

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

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

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

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

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

   "Create ABAP internal table for data based on input variables."
   "Training data."
   lo_trn_s3datasource = NEW #( iv_s3datatype = iv_trn_data_s3datatype
                                iv_s3datadistributiontype =
iv_trn_data_s3datadistribution
                                iv_s3uri = iv_trn_data_s3uri ).

   lo_trn_datasource = NEW #( io_s3datasource = lo_trn_s3datasource ).

   lo_trn_channel = NEW #( iv_channelname = 'train'
                           io_datasource = lo_trn_datasource
                           iv_compressiontype = iv_trn_data_compressiontype
                           iv_contenttype = iv_trn_data_contenttype ).
   INSERT lo_trn_channel INTO TABLE lt_input_data_config.

   "Validation data."
   lo_val_s3datasource = NEW #( iv_s3datatype = iv_val_data_s3datatype
```

```
                                iv_s3datadistributiontype =
iv_val_data_s3datadistribution
                                iv_s3uri = iv_val_data_s3uri ).

   lo_val_datasource = NEW #( io_s3datasource = lo_val_s3datasource ).

   lo_val_channel = NEW #( iv_channelname = 'validation'
                           io_datasource = lo_val_datasource
                           iv_compressiontype = iv_val_data_compressiontype
                           iv_contenttype = iv_val_data_contenttype ).
   INSERT lo_val_channel INTO TABLE lt_input_data_config.

   "Create an ABAP object for algorithm specification based on input variables."
   lo_algorithm_specification = NEW #( iv_trainingimage = iv_training_image
                                       iv_traininginputmode =
iv_training_input_mode ).

   "Create an ABAP object for resource configuration."
   lo_resource_config = NEW #( iv_instancecount = iv_instance_count
                               iv_instancetype = iv_instance_type
                               iv_volumesizeingb = iv_volume_sizeingb ).

   "Create an ABAP object for output data configuration."
   lo_output_data_config = NEW #( iv_s3outputpath = iv_s3_output_path ).

   "Create an ABAP object for stopping condition."
   lo_stopping_condition = NEW #( iv_maxruntimeinseconds =
iv_max_runtime_in_seconds ).

   TRY.
       lo_sgm->createtrainingjob(
         iv_trainingjobname           = iv_training_job_name
         iv_rolearn                   = iv_role_arn
         it_hyperparameters           = lt_hyperparameters
         it_inputdataconfig           = lt_input_data_config
         io_algorithmspecification    = lo_algorithm_specification
         io_outputdataconfig          = lo_output_data_config
         io_resourceconfig            = lo_resource_config
         io_stoppingcondition         = lo_stopping_condition ).
       MESSAGE 'Training job created.' TYPE 'I'.
     CATCH /aws1/cx_sgmresourceinuse.
       MESSAGE 'Resource being accessed is in use.' TYPE 'E'.
     CATCH /aws1/cx_sgmresourcenotfound.
       MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
```

```
      CATCH /aws1/cx_sgmresourcelimitexcd.
        MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
    ENDTRY.

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

    "Create ABAP object for the container image based on input variables."
    lo_primarycontainer = NEW #( iv_image = iv_training_image
                                 iv_modeldataurl = lv_model_data_url ).

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

    "Create an endpoint production variant."
    lo_production_variants = NEW #( iv_variantname = iv_ep_variant_name
                                    iv_modelname = iv_model_name
                                    iv_initialinstancecount =
iv_ep_initial_instance_count
                                    iv_instancetype = iv_ep_instance_type ).
    INSERT lo_production_variants INTO TABLE lt_production_variants.

    TRY.
        "Create an endpoint configuration."
        lo_ep_config_result = lo_sgm->createendpointconfig(
          iv_endpointconfigname = iv_ep_cfg_name
          it_productionvariants = lt_production_variants ).
```

```
        MESSAGE 'Endpoint configuration created.' TYPE 'I'.

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

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

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

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

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

- Untuk mengetahui hal detail mengenai API, silakan lihat topik-topik berikut di referensi API AWS SDK untuk ABAP SAP.
  - CreateEndpoint
  - CreateEndpointConfig
  - CreateModel
  - CreateTrainingJob
  - DeleteEndpoint
  - DeleteEndpointConfig
  - DeleteModel
  - DescribeEndpoint
  - DescribeTrainingJob

Contoh Secrets Manager menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Secrets Manager.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`BatchGetSecretValue`

Contoh kode berikut menunjukkan cara menggunakan`BatchGetSecretValue`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " iv_filter_name = 'mySecret'
        DATA(lo_result) = lo_smr->batchgetsecretvalue(
          it_filters = VALUE /aws1/cl_smrfilter=>tt_filterslisttype(
            (
              NEW /aws1/cl_smrfilter(
                iv_key = 'name'
                it_values = VALUE /aws1/
cl_smrfiltvalsstrlist_w=>tt_filtervaluesstringlist(
                  ( NEW /aws1/cl_smrfiltvalsstrlist_w( iv_value = iv_filter_name ) )
                )
              )
            )
          )
        ).
        ot_secret_values = lo_result->get_secretvalues( ).
        MESSAGE 'Secrets retrieved successfully.' TYPE 'I'.
      CATCH /aws1/cx_smrresourcenotfoundex.
        MESSAGE 'One or more requested secrets were not found.' TYPE 'E'.
      CATCH /aws1/cx_smrdecryptionfailure.
        MESSAGE 'Failed to decrypt one or more secrets.' TYPE 'E'.
      CATCH /aws1/cx_smrinvalidparameterex.
        MESSAGE 'Invalid parameter provided.' TYPE 'E'.
      CATCH /aws1/cx_smrinvalidrequestex.
        MESSAGE 'Invalid request.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat BatchGetSecretValuedi AWS SDK untuk referensi SAP ABAP API.

`GetSecretValue`

Contoh kode berikut menunjukkan cara menggunakan`GetSecretValue`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " iv_secret_name = 'MySecretName'
    DATA(lo_result) = lo_smr->getsecretvalue( iv_secretid = iv_secret_name ).
    ov_secret_value = lo_result->get_secretstring( ).
    MESSAGE 'Secret value retrieved successfully.' TYPE 'I'.
  CATCH /aws1/cx_smrresourcenotfoundex.
    MESSAGE 'The requested secret was not found.' TYPE 'E'.
  CATCH /aws1/cx_smrdecryptionfailure.
    MESSAGE 'Failed to decrypt the secret.' TYPE 'E'.
  CATCH /aws1/cx_smrinvalidparameterex.
    MESSAGE 'Invalid parameter provided.' TYPE 'E'.
  CATCH /aws1/cx_smrinvalidrequestex.
    MESSAGE 'Invalid request.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat GetSecretValuedi AWS SDK untuk referensi SAP ABAP API.

Contoh Amazon SES menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Amazon SES.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`CreateReceiptFilter`

Contoh kode berikut menunjukkan cara menggunakan`CreateReceiptFilter`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
" iv_allow = abap_true means 'Allow', abap_false means 'Block'
DATA(lv_policy) = COND /aws1/sesreceiptfilterpolicy(
  WHEN iv_allow = abap_true THEN 'Allow'
  ELSE 'Block'
).

DATA(lo_ip_filter) = NEW /aws1/cl_sesreceiptipfilter(
  iv_policy = lv_policy
  iv_cidr = iv_ip_address_or_range
).

DATA(lo_filter) = NEW /aws1/cl_sesreceiptfilter(
  iv_name = iv_filter_name
  io_ipfilter = lo_ip_filter
).

TRY.
    lo_ses->createreceiptfilter( io_filter = lo_filter ).
    MESSAGE 'Receipt filter created successfully' TYPE 'I'.
  CATCH /aws1/cx_sesalreadyexistsex INTO DATA(lo_ex1).
    DATA(lv_error) = |Filter already exists: { lo_ex1->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex1.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
    lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex_generic.
ENDTRY.
```

- Untuk detail API, lihat CreateReceiptFilterdi AWS SDK untuk referensi SAP ABAP API.

`CreateReceiptRule`

Contoh kode berikut menunjukkan cara menggunakan`CreateReceiptRule`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
" Create S3 action for copying emails to S3
DATA(lo_s3_action) = NEW /aws1/cl_sess3action(
  iv_bucketname = iv_bucket_name
  iv_objectkeyprefix = iv_prefix
).

" Create receipt action with S3 action
DATA(lo_action) = NEW /aws1/cl_sesreceiptaction(
  io_s3action = lo_s3_action
).

" Create list of actions
DATA lt_actions TYPE /aws1/cl_sesreceiptaction=>tt_receiptactionslist.
APPEND lo_action TO lt_actions.

" Create receipt rule
DATA(lo_rule) = NEW /aws1/cl_sesreceiptrule(
  iv_name = iv_rule_name
  iv_enabled = abap_true
  it_recipients = it_recipients
  it_actions = lt_actions
).

TRY.
    lo_ses->createreceiptrule(
      iv_rulesetname = iv_rule_set_name
```

```
          io_rule = lo_rule
        ).
        MESSAGE 'Receipt rule created successfully' TYPE 'I'.
      CATCH /aws1/cx_sesinvalids3confex INTO DATA(lo_ex1).
        DATA(lv_error) = |Invalid S3 configuration: { lo_ex1->get_text( ) }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_ex1.
      CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
        lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_ex_generic.
    ENDTRY.
```

- Untuk detail API, lihat CreateReceiptRuledi AWS SDK untuk referensi SAP ABAP API.

`CreateReceiptRuleSet`

Contoh kode berikut menunjukkan cara menggunakan`CreateReceiptRuleSet`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        lo_ses->createreceiptruleset( iv_rulesetname = iv_rule_set_name ).
        MESSAGE 'Receipt rule set created successfully' TYPE 'I'.
      CATCH /aws1/cx_sesalreadyexistsex INTO DATA(lo_ex1).
        DATA(lv_error) = |Rule set already exists: { lo_ex1->get_text( ) }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_ex1.
      CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
        lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_ex_generic.
    ENDTRY.
```

- Untuk detail API, lihat CreateReceiptRuleSetdi AWS SDK untuk referensi SAP ABAP API.

`CreateTemplate`

Contoh kode berikut menunjukkan cara menggunakan`CreateTemplate`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
DATA(lo_template) = NEW /aws1/cl_sestemplate(
  iv_templatename = iv_name
  iv_subjectpart = iv_subject
  iv_textpart = iv_text
  iv_htmlpart = iv_html
).

TRY.
    lo_ses->createtemplate( io_template = lo_template ).
    MESSAGE 'Template created successfully' TYPE 'I'.
  CATCH /aws1/cx_sesalreadyexistsex INTO DATA(lo_ex1).
    DATA(lv_error) = |Template already exists: { lo_ex1->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex1.
  CATCH /aws1/cx_sesinvalidtemplateex INTO DATA(lo_ex2).
    lv_error = |Invalid template: { lo_ex2->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex2.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
    lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex_generic.
ENDTRY.
```

- Untuk detail API, lihat CreateTemplatedi AWS SDK untuk referensi SAP ABAP API.

`DeleteIdentity`

Contoh kode berikut menunjukkan cara menggunakan`DeleteIdentity`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_ses->deleteidentity( iv_identity = iv_identity ).
    MESSAGE 'Identity deleted successfully' TYPE 'I'.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).
    DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex.
ENDTRY.
```

- Untuk detail API, lihat DeleteIdentitydi AWS SDK untuk referensi SAP ABAP API.

`DeleteReceiptFilter`

Contoh kode berikut menunjukkan cara menggunakan`DeleteReceiptFilter`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_ses->deletereceiptfilter( iv_filtername = iv_filter_name ).
    MESSAGE 'Receipt filter deleted successfully' TYPE 'I'.
```

```
  CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).
    DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex.
ENDTRY.
```

- Untuk detail API, lihat DeleteReceiptFilterdi AWS SDK untuk referensi SAP ABAP API.

`DeleteReceiptRule`

Contoh kode berikut menunjukkan cara menggunakan`DeleteReceiptRule`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_ses->deletereceiptrule(
      iv_rulesetname = iv_rule_set_name
      iv_rulename = iv_rule_name
    ).
    MESSAGE 'Receipt rule deleted successfully' TYPE 'I'.
  CATCH /aws1/cx_sesrulesetdoesnotexex INTO DATA(lo_ex1).
    DATA(lv_error) = |Rule set does not exist: { lo_ex1->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex1.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
    lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex_generic.
ENDTRY.
```

- Untuk detail API, lihat DeleteReceiptRuledi AWS SDK untuk referensi SAP ABAP API.

`DeleteReceiptRuleSet`

Contoh kode berikut menunjukkan cara menggunakan`DeleteReceiptRuleSet`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_ses->deletereceiptruleset( iv_rulesetname = iv_rule_set_name ).
    MESSAGE 'Receipt rule set deleted successfully' TYPE 'I'.
  CATCH /aws1/cx_sescannotdeleteex INTO DATA(lo_ex1).
    DATA(lv_error) = |Cannot delete rule set: { lo_ex1->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex1.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
    lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex_generic.
ENDTRY.
```

- Untuk detail API, lihat DeleteReceiptRuleSetdi AWS SDK untuk referensi SAP ABAP API.

`DeleteTemplate`

Contoh kode berikut menunjukkan cara menggunakan`DeleteTemplate`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        lo_ses->deletetemplate( iv_templatename = iv_template_name ).
        MESSAGE 'Template deleted successfully' TYPE 'I'.
      CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).
        DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_ex.
    ENDTRY.
```

- Untuk detail API, lihat DeleteTemplatedi AWS SDK untuk referensi SAP ABAP API.

`DescribeReceiptRuleSet`

Contoh kode berikut menunjukkan cara menggunakan`DescribeReceiptRuleSet`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_ses->describereceiptruleset(
          iv_rulesetname = iv_rule_set_name
        ).
        MESSAGE 'Receipt rule set described successfully' TYPE 'I'.
      CATCH /aws1/cx_sesrulesetdoesnotexex INTO DATA(lo_ex1).
        DATA(lv_error) = |Rule set does not exist: { lo_ex1->get_text( ) }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_ex1.
      CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
        lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_ex_generic.
    ENDTRY.
```

- Untuk detail API, lihat DescribeReceiptRuleSetdi AWS SDK untuk referensi SAP ABAP API.

`GetIdentityVerificationAttributes`

Contoh kode berikut menunjukkan cara menggunakan`GetIdentityVerificationAttributes`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   DATA lt_identities TYPE /aws1/cl_sesidentitylist_w=>tt_identitylist.
   APPEND NEW /aws1/cl_sesidentitylist_w( iv_value = iv_identity ) TO
lt_identities.

   TRY.
       DATA(lo_result) = lo_ses->getidentityverificationattrs(
         it_identities = lt_identities
       ).

       DATA(lt_attrs) = lo_result->get_verificationattributes( ).
       IF lt_attrs IS NOT INITIAL.
         LOOP AT lt_attrs ASSIGNING FIELD-SYMBOL(<ls_attr>).
           ov_status = <ls_attr>-value->get_verificationstatus( ).
           EXIT.
         ENDLOOP.
       ELSE.
         ov_status = 'NotFound'.
       ENDIF.
     CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).
       DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.
       MESSAGE lv_error TYPE 'I'.
       RAISE EXCEPTION lo_ex.
   ENDTRY.
```

- Untuk detail API, lihat GetIdentityVerificationAttributesdi AWS SDK untuk referensi SAP ABAP API.

`GetTemplate`

Contoh kode berikut menunjukkan cara menggunakan`GetTemplate`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    DATA(lo_result) = lo_ses->gettemplate( iv_templatename = iv_template_name ).
    oo_template = lo_result->get_template( ).
    MESSAGE 'Template retrieved successfully' TYPE 'I'.
  CATCH /aws1/cx_sestmpldoesnotexistex INTO DATA(lo_ex1).
    DATA(lv_error) = |Template does not exist: { lo_ex1->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex1.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
    lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex_generic.
ENDTRY.
```

- Untuk detail API, lihat GetTemplatedi AWS SDK untuk referensi SAP ABAP API.

`ListIdentities`

Contoh kode berikut menunjukkan cara menggunakan`ListIdentities`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    DATA(lo_result) = lo_ses->listidentities(
      iv_identitytype = iv_identity_type
      iv_maxitems = iv_max_items
    ).
    ot_identities = lo_result->get_identities( ).
    MESSAGE 'Identities retrieved successfully' TYPE 'I'.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).
    DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex.
ENDTRY.
```

- Untuk detail API, lihat ListIdentitiesdi AWS SDK untuk referensi SAP ABAP API.

`ListReceiptFilters`

Contoh kode berikut menunjukkan cara menggunakan`ListReceiptFilters`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    DATA(lo_result) = lo_ses->listreceiptfilters( ).
    ot_filters = lo_result->get_filters( ).
    MESSAGE 'Receipt filters retrieved successfully' TYPE 'I'.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).
    DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex.
ENDTRY.
```

- Untuk detail API, lihat ListReceiptFiltersdi AWS SDK untuk referensi SAP ABAP API.

`ListTemplates`

Contoh kode berikut menunjukkan cara menggunakan`ListTemplates`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    DATA(lo_result) = lo_ses->listtemplates( iv_maxitems = iv_max_items ).
    ot_templates = lo_result->get_templatesmetadata( ).
    MESSAGE 'Templates retrieved successfully' TYPE 'I'.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).
    DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex.
ENDTRY.
```

- Untuk detail API, lihat ListTemplatesdi AWS SDK untuk referensi SAP ABAP API.

`SendEmail`

Contoh kode berikut menunjukkan cara menggunakan`SendEmail`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
" Create message object
DATA(lo_subject) = NEW /aws1/cl_sescontent( iv_data = iv_subject ).
```

```
    DATA(lo_text_body) = NEW /aws1/cl_sescontent( iv_data = iv_text ).
    DATA(lo_html_body) = NEW /aws1/cl_sescontent( iv_data = iv_html ).
    DATA(lo_body) = NEW /aws1/cl_sesbody(
      io_text = lo_text_body
      io_html = lo_html_body
    ).
    DATA(lo_message) = NEW /aws1/cl_sesmessage(
      io_subject = lo_subject
      io_body = lo_body
    ).

    TRY.
        " Send email
        DATA(lo_result) = lo_ses->sendemail(
          iv_source = iv_source
          io_destination = io_destination
          io_message = lo_message
          it_replytoaddresses = it_reply_tos
        ).
        ov_msg_id = lo_result->get_messageid( ).
        MESSAGE 'Email sent successfully' TYPE 'I'.
      CATCH /aws1/cx_sesacctsendingpause00 INTO DATA(lo_ex1).
        DATA(lv_error) = |Account sending paused: { lo_ex1->get_text( ) }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_ex1.
      CATCH /aws1/cx_sesmessagerejected INTO DATA(lo_ex2).
        lv_error = |Message rejected: { lo_ex2->get_text( ) }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_ex2.
      CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
        lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_ex_generic.
    ENDTRY.
```

- Untuk detail API, lihat SendEmaildi AWS SDK untuk referensi SAP ABAP API.

`SendTemplatedEmail`

Contoh kode berikut menunjukkan cara menggunakan`SendTemplatedEmail`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    " Send templated email
    DATA(lo_result) = lo_ses->sendtemplatedemail(
      iv_source = iv_source
      io_destination = io_destination
      iv_template = iv_template_name
      iv_templatedata = iv_template_data
      it_replytoaddresses = it_reply_tos
    ).
    ov_msg_id = lo_result->get_messageid( ).
    MESSAGE 'Templated email sent successfully' TYPE 'I'.
  CATCH /aws1/cx_sestmpldoesnotexistex INTO DATA(lo_ex1).
    DATA(lv_error) = |Template does not exist: { lo_ex1->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex1.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
    lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex_generic.
ENDTRY.
```

- Untuk detail API, lihat SendTemplatedEmaildi AWS SDK untuk referensi SAP ABAP API.

`UpdateTemplate`

Contoh kode berikut menunjukkan cara menggunakan`UpdateTemplate`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
DATA(lo_template) = NEW /aws1/cl_sestemplate(
  iv_templatename = iv_name
  iv_subjectpart = iv_subject
  iv_textpart = iv_text
  iv_htmlpart = iv_html
).

TRY.
    lo_ses->updatetemplate( io_template = lo_template ).
    MESSAGE 'Template updated successfully' TYPE 'I'.
  CATCH /aws1/cx_sestmpldoesnotexistex INTO DATA(lo_ex1).
    DATA(lv_error) = |Template does not exist: { lo_ex1->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex1.
  CATCH /aws1/cx_sesinvalidtemplateex INTO DATA(lo_ex2).
    lv_error = |Invalid template: { lo_ex2->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex2.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
    lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex_generic.
ENDTRY.
```

- Untuk detail API, lihat UpdateTemplatedi AWS SDK untuk referensi SAP ABAP API.

`VerifyDomainIdentity`

Contoh kode berikut menunjukkan cara menggunakan`VerifyDomainIdentity`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        DATA(lo_result) = lo_ses->verifydomainidentity( iv_domain =
iv_domain_name ).
        ov_token = lo_result->get_verificationtoken( ).
        MESSAGE 'Domain verification initiated' TYPE 'I'.
      CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).
        DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_ex.
    ENDTRY.
```

- Untuk detail API, lihat VerifyDomainIdentitydi AWS SDK untuk referensi SAP ABAP API.

`VerifyEmailIdentity`

Contoh kode berikut menunjukkan cara menggunakan`VerifyEmailIdentity`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        lo_ses->verifyemailidentity( iv_emailaddress = iv_email_address ).
        MESSAGE 'Email verification initiated' TYPE 'I'.
      CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).
        DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.
        MESSAGE lv_error TYPE 'I'.
```

```
        RAISE EXCEPTION lo_ex.
    ENDTRY.
```

- Untuk detail API, lihat VerifyEmailIdentitydi AWS SDK untuk referensi SAP ABAP API.

Contoh Amazon SES API v2 menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Amazon SES API v2.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`CreateContact`

Contoh kode berikut menunjukkan cara menggunakan`CreateContact`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        lo_se2->createcontact(
          iv_contactlistname = iv_contact_list_name
```

```
        iv_emailaddress = iv_email_address ).
      MESSAGE 'Contact created successfully.' TYPE 'I'.
    CATCH /aws1/cx_se2alreadyexistsex.
      MESSAGE 'Contact already exists.' TYPE 'I'.
    CATCH /aws1/cx_se2badrequestex.
      MESSAGE 'Bad request.' TYPE 'E'.
    CATCH /aws1/cx_se2notfoundexception.
      MESSAGE 'Contact list not found.' TYPE 'E'.
  ENDTRY.
```

- Untuk detail API, lihat CreateContactdi AWS SDK untuk referensi SAP ABAP API.

`CreateContactList`

Contoh kode berikut menunjukkan cara menggunakan`CreateContactList`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
  TRY.
      lo_se2->createcontactlist(
        iv_contactlistname = iv_contact_list_name ).
      MESSAGE 'Contact list created successfully.' TYPE 'I'.
    CATCH /aws1/cx_se2alreadyexistsex.
      MESSAGE 'Contact list already exists.' TYPE 'I'.
    CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).
      MESSAGE 'Bad request - contact list limit may be reached.' TYPE 'I'.
      " Re-raise the exception so the caller can handle it
      RAISE EXCEPTION lo_bad_request.
    CATCH /aws1/cx_se2limitexceededex INTO DATA(lo_limit_exceeded).
      MESSAGE 'Limit exceeded - contact list limit reached.' TYPE 'I'.
      " Re-raise the exception so the caller can handle it
      RAISE EXCEPTION lo_limit_exceeded.
  ENDTRY.
```

- Untuk detail API, lihat CreateContactListdi AWS SDK untuk referensi SAP ABAP API.

`CreateEmailIdentity`

Contoh kode berikut menunjukkan cara menggunakan`CreateEmailIdentity`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_se2->createemailidentity(
      iv_emailidentity = iv_email_identity ).
    MESSAGE 'Email identity created successfully.' TYPE 'I'.
  CATCH /aws1/cx_se2alreadyexistsex.
    MESSAGE 'Email identity already exists.' TYPE 'I'.
  CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).
    MESSAGE lo_bad_request TYPE 'I' DISPLAY LIKE 'E'.
  CATCH /aws1/cx_se2limitexceededex INTO DATA(lo_limit_exceeded).
    MESSAGE lo_limit_exceeded TYPE 'I' DISPLAY LIKE 'E'.
ENDTRY.
```

- Untuk detail API, lihat CreateEmailIdentitydi AWS SDK untuk referensi SAP ABAP API.

`CreateEmailTemplate`

Contoh kode berikut menunjukkan cara menggunakan`CreateEmailTemplate`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    DATA(lo_template_content) = NEW /aws1/cl_se2emailtmplcontent(
      iv_subject = iv_subject
      iv_html = iv_html
      iv_text = iv_text ).

    lo_se2->createemailtemplate(
      iv_templatename = iv_template_name
      io_templatecontent = lo_template_content ).
    MESSAGE 'Email template created successfully.' TYPE 'I'.
  CATCH /aws1/cx_se2alreadyexistsex.
    MESSAGE 'Email template already exists.' TYPE 'I'.
  CATCH /aws1/cx_se2badrequestex.
    MESSAGE 'Bad request.' TYPE 'E'.
  CATCH /aws1/cx_se2limitexceededex.
    MESSAGE 'Limit exceeded.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat CreateEmailTemplatedi AWS SDK untuk referensi SAP ABAP API.

`DeleteContactList`

Contoh kode berikut menunjukkan cara menggunakan`DeleteContactList`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_se2->deletecontactlist(
      iv_contactlistname = iv_contact_list_name ).
    MESSAGE 'Contact list deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_se2notfoundexception.
    MESSAGE 'Contact list not found.' TYPE 'I'.
  CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).
    MESSAGE 'Bad request.' TYPE 'I'.
```

```
    RAISE EXCEPTION lo_bad_request.
ENDTRY.
```

- Untuk detail API, lihat DeleteContactListdi AWS SDK untuk referensi SAP ABAP API.

`DeleteEmailIdentity`

Contoh kode berikut menunjukkan cara menggunakan`DeleteEmailIdentity`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_se2->deleteemailidentity(
      iv_emailidentity = iv_email_identity ).
    MESSAGE 'Email identity deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_se2notfoundexception.
    MESSAGE 'Email identity not found.' TYPE 'I'.
  CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).
    MESSAGE 'Bad request.' TYPE 'I'.
    RAISE EXCEPTION lo_bad_request.
ENDTRY.
```

- Untuk detail API, lihat DeleteEmailIdentitydi AWS SDK untuk referensi SAP ABAP API.

`DeleteEmailTemplate`

Contoh kode berikut menunjukkan cara menggunakan`DeleteEmailTemplate`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        lo_se2->deleteemailtemplate(
          iv_templatename = iv_template_name ).
        MESSAGE 'Email template deleted successfully.' TYPE 'I'.
      CATCH /aws1/cx_se2notfoundexception.
        MESSAGE 'Email template not found.' TYPE 'I'.
      CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).
        MESSAGE 'Bad request.' TYPE 'I'.
        RAISE EXCEPTION lo_bad_request.
    ENDTRY.
```

- Untuk detail API, lihat DeleteEmailTemplatedi AWS SDK untuk referensi SAP ABAP API.

`GetEmailIdentity`

Contoh kode berikut menunjukkan cara menggunakan`GetEmailIdentity`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_se2->getemailidentity(
          iv_emailidentity = iv_email_identity ).
        MESSAGE |Identity type: { oo_result->get_identitytype( ) }, | &&
                |verified for sending: { oo_result-
>get_verifiedforsendingstatus( ) }| TYPE 'I'.
```

```
  CATCH /aws1/cx_se2notfoundexception.
    MESSAGE |Email identity { iv_email_identity } not found.| TYPE 'I'.
  CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).
    MESSAGE lo_bad_request TYPE 'I' DISPLAY LIKE 'E'.
    RAISE EXCEPTION lo_bad_request.
ENDTRY.
```

- Untuk detail API, lihat GetEmailIdentitydi AWS SDK untuk referensi SAP ABAP API.

`ListContacts`

Contoh kode berikut menunjukkan cara menggunakan`ListContacts`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_se2->listcontacts(
      iv_contactlistname = iv_contact_list_name ).
    DATA(lv_count) = lines( oo_result->get_contacts( ) ).
    MESSAGE |Retrieved { lv_count } contacts from list.| TYPE 'I'.
  CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).
    MESSAGE 'Bad request.' TYPE 'I'.
    RAISE EXCEPTION lo_bad_request.
  CATCH /aws1/cx_se2notfoundexception INTO DATA(lo_not_found).
    MESSAGE 'Contact list not found.' TYPE 'I'.
    RAISE EXCEPTION lo_not_found.
ENDTRY.
```

- Untuk detail API, lihat ListContactsdi AWS SDK untuk referensi SAP ABAP API.

`SendBulkEmail`

Contoh kode berikut menunjukkan cara menggunakan`SendBulkEmail`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " Build the default template content used for all bulk recipients
        DATA(lo_template) = NEW /aws1/cl_se2template(
          iv_templatename = iv_template_name
          iv_templatedata = iv_template_data ).

        DATA(lo_default_content) = NEW /aws1/cl_se2bulkemailcontent(
          io_template = lo_template ).

        DATA(lo_result) = lo_se2->sendbulkemail(
          iv_fromemailaddress = iv_from_address
          io_defaultcontent   = lo_default_content
          it_bulkemailentries = it_bulk_entries ).

        ot_results = lo_result->get_bulkemailentryresults( ).
        MESSAGE |Bulk email sent to { lines( it_bulk_entries ) } recipient(s).| TYPE
'I'.
      CATCH /aws1/cx_se2messagerejected INTO DATA(lo_rejected).
        MESSAGE lo_rejected TYPE 'I' DISPLAY LIKE 'E'.
        RAISE EXCEPTION lo_rejected.
      CATCH /aws1/cx_se2mailfrmdomnotver00 INTO DATA(lo_not_verified).
        MESSAGE lo_not_verified TYPE 'I' DISPLAY LIKE 'E'.
        RAISE EXCEPTION lo_not_verified.
      CATCH /aws1/cx_se2notfoundexception INTO DATA(lo_not_found).
        MESSAGE lo_not_found TYPE 'I' DISPLAY LIKE 'E'.
        RAISE EXCEPTION lo_not_found.
      CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).
        MESSAGE lo_bad_request TYPE 'I' DISPLAY LIKE 'E'.
        RAISE EXCEPTION lo_bad_request.
    ENDTRY.
```

- Untuk detail API, lihat SendBulkEmaildi AWS SDK untuk referensi SAP ABAP API.

`SendEmail`

Contoh kode berikut menunjukkan cara menggunakan`SendEmail`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

Mengirim pesan.

```
    TRY.
        " Create destination with recipient address
        DATA lt_to_addresses TYPE /aws1/
cl_se2emailaddresslist_w=>tt_emailaddresslist.
        APPEND NEW /aws1/cl_se2emailaddresslist_w( iv_value = iv_to_email_address )
 TO lt_to_addresses.
        DATA(lo_destination) = NEW /aws1/cl_se2destination(
          it_toaddresses = lt_to_addresses ).

        " Create message content
        DATA(lo_subject) = NEW /aws1/cl_se2content( iv_data = iv_subject ).
        DATA(lo_text_body) = NEW /aws1/cl_se2content( iv_data = iv_text_body ).
        DATA(lo_html_body) = NEW /aws1/cl_se2content( iv_data = iv_html_body ).
        DATA(lo_body) = NEW /aws1/cl_se2body(
          io_text = lo_text_body
          io_html = lo_html_body ).
        DATA(lo_message) = NEW /aws1/cl_se2message(
          io_subject = lo_subject
          io_body = lo_body ).

        DATA(lo_content) = NEW /aws1/cl_se2emailcontent(
          io_simple = lo_message ).

        " Send the email
        lo_se2->sendemail(
          iv_fromemailaddress = iv_from_email_address
          io_destination = lo_destination
          io_content = lo_content ).
        MESSAGE 'Email sent successfully.' TYPE 'I'.
```

```
      CATCH /aws1/cx_se2accountsuspendedex INTO DATA(lo_account_suspended).
        MESSAGE 'Account suspended.' TYPE 'I'.
        RAISE EXCEPTION lo_account_suspended.
      CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).
        MESSAGE 'Bad request.' TYPE 'I'.
        RAISE EXCEPTION lo_bad_request.
      CATCH /aws1/cx_se2messagerejected INTO DATA(lo_message_rejected).
        MESSAGE 'Message rejected - check email verification.' TYPE 'I'.
        RAISE EXCEPTION lo_message_rejected.
    ENDTRY.
```

Mengirim pesan menggunakan template.

```
    TRY.
        " Create destination with recipient address
        DATA lt_to_addresses TYPE /aws1/
cl_se2emailaddresslist_w=>tt_emailaddresslist.
        APPEND NEW /aws1/cl_se2emailaddresslist_w( iv_value = iv_to_email_address )
 TO lt_to_addresses.
        DATA(lo_destination) = NEW /aws1/cl_se2destination(
          it_toaddresses = lt_to_addresses ).

        " Create template reference
        DATA(lo_template) = NEW /aws1/cl_se2template(
          iv_templatename = iv_template_name
          iv_templatedata = iv_template_data ).

        DATA(lo_content) = NEW /aws1/cl_se2emailcontent(
          io_template = lo_template ).

        " Create list management options
        DATA(lo_list_mgmt) = NEW /aws1/cl_se2listmanagementopts(
          iv_contactlistname = iv_contact_list_name ).

        " Send the email using template
        lo_se2->sendemail(
          iv_fromemailaddress = iv_from_email_address
          io_destination = lo_destination
          io_content = lo_content
          io_listmanagementoptions = lo_list_mgmt ).
        MESSAGE 'Email sent using template successfully.' TYPE 'I'.
      CATCH /aws1/cx_se2accountsuspendedex INTO DATA(lo_account_suspended).
```

```
      MESSAGE 'Account suspended.' TYPE 'I'.
      RAISE EXCEPTION lo_account_suspended.
    CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).
      MESSAGE 'Bad request.' TYPE 'I'.
      RAISE EXCEPTION lo_bad_request.
    CATCH /aws1/cx_se2messagerejected INTO DATA(lo_message_rejected).
      MESSAGE 'Message rejected - check email verification.' TYPE 'I'.
      RAISE EXCEPTION lo_message_rejected.
  ENDTRY.
```

- Untuk detail API, lihat SendEmaildi AWS SDK untuk referensi SAP ABAP API.

Contoh Amazon SNS menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Amazon SNS.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Skenario adalah contoh kode yang menunjukkan kepada Anda bagaimana menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan atau dikombinasikan dengan yang lain Layanan AWS.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan
- Skenario

Tindakan

`CreateTopic`

Contoh kode berikut menunjukkan cara menggunakan`CreateTopic`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat CreateTopicdi AWS SDK untuk referensi SAP ABAP API.

`DeleteTopic`

Contoh kode berikut menunjukkan cara menggunakan`DeleteTopic`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat DeleteTopicdi AWS SDK untuk referensi SAP ABAP API.

`GetTopicAttributes`

Contoh kode berikut menunjukkan cara menggunakan`GetTopicAttributes`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat GetTopicAttributesdi AWS SDK untuk referensi SAP ABAP API.

`ListSubscriptions`

Contoh kode berikut menunjukkan cara menggunakan`ListSubscriptions`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
```

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

- Untuk detail API, lihat ListSubscriptionsdi AWS SDK untuk referensi SAP ABAP API.

`ListTopics`

Contoh kode berikut menunjukkan cara menggunakan`ListTopics`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat ListTopicsdi AWS SDK untuk referensi SAP ABAP API.

`Publikasikan`

Contoh kode berikut menunjukkan cara menggunakan`Publish`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   TRY.
       oo_result = lo_sns->publish(              " oo_result is returned for
testing purposes. "
         iv_topicarn = iv_topic_arn
         iv_message = iv_message ).
       MESSAGE 'Message published to SNS topic.' TYPE 'I'.
     CATCH /aws1/cx_snsnotfoundexception.
       MESSAGE 'Topic does not exist.' TYPE 'E'.
   ENDTRY.
```

Publikasikan pesan dengan atribut ke topik.

```
   TRY.
       oo_result = lo_sns->publish(              " oo_result is returned for
testing purposes. "
         iv_topicarn = iv_topic_arn
         iv_message = iv_message
         it_messageattributes = it_msg_attrs ).
       MESSAGE 'Message with attributes published to SNS topic.' TYPE 'I'.
     CATCH /aws1/cx_snsnotfoundexception.
       MESSAGE 'Topic does not exist.' TYPE 'E'.
   ENDTRY.
```

Publikasikan pesan multi-format ke suatu topik.

```
   " Build JSON message structure for multi-format message
   DATA(lv_json_message) = |\{ "default": "{ iv_default_message }", "sms":
"{ iv_sms_message }", "email": "{ iv_email_message }" \}|.

   TRY.
```

```
        oo_result = lo_sns->publish(              " oo_result is returned for
testing purposes. "
          iv_topicarn = iv_topic_arn
          iv_message = lv_json_message
          iv_subject = iv_subject
          iv_messagestructure = 'json' ).
        MESSAGE 'Multi-format message published to SNS topic.' TYPE 'I'.
      CATCH /aws1/cx_snsnotfoundexception.
        MESSAGE 'Topic does not exist.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat Publikasikan di AWS SDK untuk referensi API SAP ABAP.

`SetSubscriptionAttributes`

Contoh kode berikut menunjukkan cara menggunakan`SetSubscriptionAttributes`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        lo_sns->setsubscriptionattributes(
            iv_subscriptionarn = iv_subscription_arn
            iv_attributename  = 'FilterPolicy'
            iv_attributevalue = iv_filter_policy ).
        MESSAGE 'Added filter policy to subscription.' TYPE 'I'.
      CATCH /aws1/cx_snsnotfoundexception.
        MESSAGE 'Subscription does not exist.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat SetSubscriptionAttributesdi AWS SDK untuk referensi SAP ABAP API.

`SetTopicAttributes`

Contoh kode berikut menunjukkan cara menggunakan`SetTopicAttributes`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat SetTopicAttributesdi AWS SDK untuk referensi SAP ABAP API.

`Langganan`

Contoh kode berikut menunjukkan cara menggunakan`Subscribe`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

Berlangganan alamat email ke suatu topik.

```
TRY.
```

```
        oo_result = lo_sns->subscribe(                      "oo_result is returned
 for testing purposes."
                iv_topicarn = iv_topic_arn
                iv_protocol = 'email'
                iv_endpoint = iv_email_address
                iv_returnsubscriptionarn = abap_true ).
        MESSAGE 'Email address subscribed to SNS topic.' TYPE 'I'.
      CATCH /aws1/cx_snsnotfoundexception.
        MESSAGE 'Topic does not exist.' TYPE 'E'.
      CATCH /aws1/cx_snssubscriptionlmte00.
        MESSAGE 'Unable to create subscriptions. You have reached the maximum number
 of subscriptions allowed.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat Berlangganan di AWS SDK untuk referensi API SAP ABAP.

`Berhenti Berlangganan`

Contoh kode berikut menunjukkan cara menggunakan`Unsubscribe`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat Berhenti berlangganan di AWS SDK untuk referensi SAP ABAP API.

Skenario

Membuat dan mempublikasikan ke topik FIFO

Contoh kode berikut menunjukkan cara membuat dan mempublikasikan ke topik FIFO Amazon SNS.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

Buat topik FIFO, berlangganan antrian Amazon SQS FIFO ke topik tersebut, dan publikasikan pesan ke topik Amazon SNS.

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

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

```
    " Subscribes an endpoint to an Amazon Simple Notification Service (Amazon SNS)
 topic. "
    " Only Amazon Simple Queue Service (Amazon SQS) FIFO queues can be subscribed to
 an SNS FIFO topic. "
    TRY.
        DATA(lo_subscribe_result) = lo_sns->subscribe(
               iv_topicarn = lv_topic_arn
               iv_protocol = 'sqs'
               iv_endpoint = iv_queue_arn ).
        DATA(lv_subscription_arn) = lo_subscribe_result->get_subscriptionarn( ).
        ov_subscription_arn = lv_subscription_arn.                      "
 ov_subscription_arn is returned for testing purposes. "
        MESSAGE 'SQS queue was subscribed to SNS topic.' TYPE 'I'.
      CATCH /aws1/cx_snsnotfoundexception.
        MESSAGE 'Topic does not exist.' TYPE 'E'.
      CATCH /aws1/cx_snssubscriptionlmte00.
        MESSAGE 'Unable to create subscriptions. You have reached the maximum number
 of subscriptions allowed.' TYPE 'E'.
    ENDTRY.

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

        DATA(lo_result) = lo_sns->publish(
             iv_topicarn = lv_topic_arn
             iv_message = 'The price of your mobile plan has been increased from $19
 to $23'
             iv_subject = 'Changes to mobile plan'
             iv_messagegroupid = 'Update-2'
             iv_messagededuplicationid = 'Update-2.1'
             it_messageattributes = lt_msg_attributes ).
        ov_message_id = lo_result->get_messageid( ).                    "
 ov_message_id is returned for testing purposes. "
        MESSAGE 'Message was published to SNS topic.' TYPE 'I'.
```

```
      CATCH /aws1/cx_snsnotfoundexception.
        MESSAGE 'Topic does not exist.' TYPE 'E'.
    ENDTRY.
```

- Untuk mengetahui hal detail mengenai API, silakan lihat topik-topik berikut di referensi API AWS SDK untuk ABAP SAP.
  - CreateTopic
  - Publikasikan
  - Berlangganan

Publikasikan pesan teks SMS

Contoh kode berikut menunjukkan cara mempublikasikan pesan SMS menggunakan Amazon SNS.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    " iv_phone_number = '+12065550101' - Phone number in E.164 format
    TRY.
        oo_result = lo_sns->publish(              " oo_result is returned for
testing purposes. "
          iv_phonenumber = iv_phone_number
          iv_message = iv_message ).
        MESSAGE 'Message published to phone number.' TYPE 'I'.
      CATCH /aws1/cx_snsnotfoundexception.
        MESSAGE 'Phone number does not exist.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat Publikasikan di AWS SDK untuk referensi API SAP ABAP.

Contoh Amazon SQS menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Amazon SQS.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Skenario adalah contoh kode yang menunjukkan kepada Anda bagaimana menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan atau dikombinasikan dengan yang lain Layanan AWS.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan
- Skenario

Tindakan

`CreateQueue`

Contoh kode berikut menunjukkan cara menggunakan`CreateQueue`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

Buat antrean standar Amazon SQS.

```
    TRY.
        oo_result = lo_sqs->createqueue( iv_queuename = iv_queue_name ).        "
oo_result is returned for testing purposes. "
```

```
        MESSAGE 'SQS queue created.' TYPE 'I'.
      CATCH /aws1/cx_sqsqueuedeldrecently.
        MESSAGE 'After deleting a queue, wait 60 seconds before creating another
 queue with the same name.' TYPE 'E'.
      CATCH /aws1/cx_sqsqueuenameexists.
        MESSAGE 'A queue with this name already exists.' TYPE 'E'.
    ENDTRY.
```

Buat antrean Amazon SQS yang menunggu pesan tiba.

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

- Untuk detail API, lihat CreateQueuedi AWS SDK untuk referensi SAP ABAP API.

`DeleteMessage`

Contoh kode berikut menunjukkan cara menggunakan`DeleteMessage`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   TRY.
       lo_sqs->deletemessage(
          iv_queueurl = iv_queue_url
          iv_receipthandle = iv_receipt_handle ).
       MESSAGE 'Message deleted from SQS queue.' TYPE 'I'.
     CATCH /aws1/cx_sqsinvalididformat.
       MESSAGE 'The specified receipt handle is not valid.' TYPE 'E'.
     CATCH /aws1/cx_sqsreceipthandleisinv.
       MESSAGE 'The specified receipt handle is not valid for the current version.'
TYPE 'E'.
   ENDTRY.
```

- Untuk detail API, lihat DeleteMessagedi AWS SDK untuk referensi SAP ABAP API.

`DeleteMessageBatch`

Contoh kode berikut menunjukkan cara menggunakan`DeleteMessageBatch`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   TRY.
       oo_result = lo_sqs->deletemessagebatch(       " oo_result is returned for
testing purposes. "
          iv_queueurl = iv_queue_url
```

```
          it_entries = it_entries ).
        MESSAGE 'Messages deleted from SQS queue.' TYPE 'I'.
      CATCH /aws1/cx_sqsbtcentidsnotdist00.
        MESSAGE 'Two or more batch entries in the request have the same ID.' TYPE
'E'.
      CATCH /aws1/cx_sqsemptybatchrequest.
        MESSAGE 'The batch request does not contain any entries.' TYPE 'E'.
      CATCH /aws1/cx_sqsinvbatchentryid.
        MESSAGE 'The ID of a batch entry in a batch request is not valid.' TYPE 'E'.
      CATCH /aws1/cx_sqstoomanyentriesin00.
        MESSAGE 'The batch request contains more entries than allowed.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat DeleteMessageBatchdi AWS SDK untuk referensi SAP ABAP API.

`DeleteQueue`

Contoh kode berikut menunjukkan cara menggunakan`DeleteQueue`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat DeleteQueuedi AWS SDK untuk referensi SAP ABAP API.

`GetQueueUrl`

Contoh kode berikut menunjukkan cara menggunakan`GetQueueUrl`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat GetQueueUrldi AWS SDK untuk referensi SAP ABAP API.

`ListQueues`

Contoh kode berikut menunjukkan cara menggunakan`ListQueues`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

- Untuk detail API, lihat ListQueuesdi AWS SDK untuk referensi SAP ABAP API.

`ReceiveMessage`

Contoh kode berikut menunjukkan cara menggunakan`ReceiveMessage`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

Menerima pesan dari antrian 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.
```

Menerima pesan dari antrian Amazon SQS menggunakan dukungan polling panjang.

```
   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.
```

- Untuk detail API, lihat ReceiveMessagedi AWS SDK untuk referensi SAP ABAP API.

`SendMessage`

Contoh kode berikut menunjukkan cara menggunakan`SendMessage`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat SendMessagedi AWS SDK untuk referensi SAP ABAP API.

`SendMessageBatch`

Contoh kode berikut menunjukkan cara menggunakan`SendMessageBatch`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
```

```
        oo_result = lo_sqs->sendmessagebatch(         " oo_result is returned for
testing purposes. "
          iv_queueurl = iv_queue_url
          it_entries = it_messages ).
        MESSAGE 'Messages sent to SQS queue.' TYPE 'I'.
      CATCH /aws1/cx_sqsbtcentidsnotdist00.
        MESSAGE 'Two or more batch entries in the request have the same ID.' TYPE
'E'.
      CATCH /aws1/cx_sqsbatchreqtoolong.
        MESSAGE 'The length of all the messages put together is more than the
limit.' TYPE 'E'.
      CATCH /aws1/cx_sqsemptybatchrequest.
        MESSAGE 'The batch request does not contain any entries.' TYPE 'E'.
      CATCH /aws1/cx_sqsinvbatchentryid.
        MESSAGE 'The ID of a batch entry in a batch request is not valid.' TYPE 'E'.
      CATCH /aws1/cx_sqstoomanyentriesin00.
        MESSAGE 'The batch request contains more entries than allowed.' TYPE 'E'.
      CATCH /aws1/cx_sqsunsupportedop.
        MESSAGE 'Operation not supported.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat SendMessageBatchdi AWS SDK untuk referensi SAP ABAP API.

Skenario

Membuat dan mempublikasikan ke topik FIFO

Contoh kode berikut menunjukkan cara membuat dan mempublikasikan ke topik FIFO Amazon SNS.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

Buat topik FIFO, berlangganan antrian Amazon SQS FIFO ke topik tersebut, dan publikasikan pesan ke topik Amazon SNS.

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

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

    " Subscribes an endpoint to an Amazon Simple Notification Service (Amazon SNS)
 topic. "
    " Only Amazon Simple Queue Service (Amazon SQS) FIFO queues can be subscribed to
 an SNS FIFO topic. "
    TRY.
        DATA(lo_subscribe_result) = lo_sns->subscribe(
               iv_topicarn = lv_topic_arn
               iv_protocol = 'sqs'
               iv_endpoint = iv_queue_arn ).
        DATA(lv_subscription_arn) = lo_subscribe_result->get_subscriptionarn( ).
        ov_subscription_arn = lv_subscription_arn.                          "
 ov_subscription_arn is returned for testing purposes. "
        MESSAGE 'SQS queue was subscribed to SNS topic.' TYPE 'I'.
      CATCH /aws1/cx_snsnotfoundexception.
        MESSAGE 'Topic does not exist.' TYPE 'E'.
      CATCH /aws1/cx_snssubscriptionlmte00.
        MESSAGE 'Unable to create subscriptions. You have reached the maximum number
 of subscriptions allowed.' TYPE 'E'.
    ENDTRY.

    " Publish message to SNS topic. "
    TRY.
```

```
        DATA lt_msg_attributes TYPE /aws1/
cl_snsmessageattrvalue=>tt_messageattributemap.
        DATA ls_msg_attributes TYPE /aws1/
cl_snsmessageattrvalue=>ts_messageattributemap_maprow.
        ls_msg_attributes-key = 'Importance'.
        ls_msg_attributes-value = NEW /aws1/cl_snsmessageattrvalue( iv_datatype =
 'String'
                                                                   iv_stringvalue =
 'High' ).
        INSERT ls_msg_attributes INTO TABLE lt_msg_attributes.

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

- Untuk mengetahui hal detail mengenai API, silakan lihat topik-topik berikut di referensi API AWS SDK untuk ABAP SAP.
  - CreateTopic
  - Publikasikan
  - Berlangganan

Contoh Step Functions menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan mengimplementasikan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Step Functions.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`CreateActivity`

Contoh kode berikut menunjukkan cara menggunakan`CreateActivity`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    DATA(lo_result) = lo_sfn->createactivity(
      iv_name = iv_name
    ).
    ov_activity_arn = lo_result->get_activityarn( ).
    MESSAGE 'Activity created successfully.' TYPE 'I'.
  CATCH /aws1/cx_sfnactivityalrdyex.
    MESSAGE 'Activity already exists.' TYPE 'E'.
  CATCH /aws1/cx_sfninvalidname.
    MESSAGE 'Invalid activity name.' TYPE 'E'.
  CATCH /aws1/cx_sfnactivitylimitexcd.
    MESSAGE 'Activity limit exceeded.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat CreateActivitydi AWS SDK untuk referensi SAP ABAP API.

`CreateStateMachine`

Contoh kode berikut menunjukkan cara menggunakan`CreateStateMachine`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    DATA(lo_result) = lo_sfn->createstatemachine(
      iv_name = iv_name
      iv_definition = iv_definition
      iv_rolearn = iv_role_arn
    ).
    ov_state_machine_arn = lo_result->get_statemachinearn( ).
    MESSAGE 'State machine created successfully.' TYPE 'I'.
  CATCH /aws1/cx_sfnstatemachinealrex.
    MESSAGE 'State machine already exists.' TYPE 'E'.
  CATCH /aws1/cx_sfninvaliddefinition.
    MESSAGE 'Invalid state machine definition.' TYPE 'E'.
  CATCH /aws1/cx_sfninvalidname.
    MESSAGE 'Invalid state machine name.' TYPE 'E'.
  CATCH /aws1/cx_sfninvalidarn.
    MESSAGE 'Invalid role ARN.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat CreateStateMachinedi AWS SDK untuk referensi SAP ABAP API.

`DeleteActivity`

Contoh kode berikut menunjukkan cara menggunakan`DeleteActivity`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_sfn->deleteactivity(
      iv_activityarn = iv_activity_arn
    ).
    MESSAGE 'Activity deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_sfninvalidarn.
    MESSAGE 'Invalid activity ARN.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DeleteActivitydi AWS SDK untuk referensi SAP ABAP API.

`DeleteStateMachine`

Contoh kode berikut menunjukkan cara menggunakan`DeleteStateMachine`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_sfn->deletestatemachine(
      iv_statemachinearn = iv_state_machine_arn
    ).
    MESSAGE 'State machine deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_sfninvalidarn.
    MESSAGE 'Invalid state machine ARN.' TYPE 'E'.
```

```
    CATCH /aws1/cx_sfnvalidationex.
      MESSAGE 'Validation error occurred.' TYPE 'E'.
  ENDTRY.
```

- Untuk detail API, lihat DeleteStateMachinedi AWS SDK untuk referensi SAP ABAP API.

`DescribeExecution`

Contoh kode berikut menunjukkan cara menggunakan`DescribeExecution`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
  TRY.
      oo_result = lo_sfn->describeexecution(
        iv_executionarn = iv_execution_arn
      ).
      MESSAGE 'Execution described successfully.' TYPE 'I'.
    CATCH /aws1/cx_sfnexecdoesnotexist.
      MESSAGE 'Execution does not exist.' TYPE 'E'.
    CATCH /aws1/cx_sfninvalidarn.
      MESSAGE 'Invalid execution ARN.' TYPE 'E'.
  ENDTRY.
```

- Untuk detail API, lihat DescribeExecutiondi AWS SDK untuk referensi SAP ABAP API.

`DescribeStateMachine`

Contoh kode berikut menunjukkan cara menggunakan`DescribeStateMachine`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_sfn->describestatemachine(
      iv_statemachinearn = iv_state_machine_arn
    ).
    MESSAGE 'State machine described successfully.' TYPE 'I'.
  CATCH /aws1/cx_sfnstatemachinedoes00.
    MESSAGE 'State machine does not exist.' TYPE 'E'.
  CATCH /aws1/cx_sfninvalidarn.
    MESSAGE 'Invalid state machine ARN.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat DescribeStateMachinedi AWS SDK untuk referensi SAP ABAP API.

`GetActivityTask`

Contoh kode berikut menunjukkan cara menggunakan`GetActivityTask`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_sfn->getactivitytask(
      iv_activityarn = iv_activity_arn
    ).
    MESSAGE 'Activity task retrieved successfully.' TYPE 'I'.
```

```
  CATCH /aws1/cx_sfnactivitydoesnotex.
    MESSAGE 'Activity does not exist.' TYPE 'E'.
  CATCH /aws1/cx_sfninvalidarn.
    MESSAGE 'Invalid activity ARN.' TYPE 'E'.
  CATCH /aws1/cx_sfnactivityworkerlm00.
    MESSAGE 'Activity worker limit exceeded.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat GetActivityTaskdi AWS SDK untuk referensi SAP ABAP API.

`ListActivities`

Contoh kode berikut menunjukkan cara menggunakan`ListActivities`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    DATA(lo_result) = lo_sfn->listactivities( ).
    DATA(lt_activities) = lo_result->get_activities( ).
    LOOP AT lt_activities INTO DATA(lo_activity).
      IF lo_activity->get_name( ) = iv_name.
        ov_activity_arn = lo_activity->get_activityarn( ).
        EXIT.
      ENDIF.
    ENDLOOP.
    MESSAGE 'Activities listed successfully.' TYPE 'I'.
  CATCH /aws1/cx_sfninvalidtoken.
    MESSAGE 'Invalid pagination token.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat ListActivitiesdi AWS SDK untuk referensi SAP ABAP API.

`ListStateMachines`

Contoh kode berikut menunjukkan cara menggunakan`ListStateMachines`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    DATA(lo_result) = lo_sfn->liststatemachines( ).
    DATA(lt_state_machines) = lo_result->get_statemachines( ).
    LOOP AT lt_state_machines INTO DATA(lo_state_machine).
      IF lo_state_machine->get_name( ) = iv_name.
        ov_state_machine_arn = lo_state_machine->get_statemachinearn( ).
        EXIT.
      ENDIF.
    ENDLOOP.
    MESSAGE 'State machines listed successfully.' TYPE 'I'.
  CATCH /aws1/cx_sfninvalidtoken.
    MESSAGE 'Invalid pagination token.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat ListStateMachinesdi AWS SDK untuk referensi SAP ABAP API.

`SendTaskSuccess`

Contoh kode berikut menunjukkan cara menggunakan`SendTaskSuccess`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        lo_sfn->sendtasksuccess(
          iv_tasktoken = iv_task_token
          iv_output = iv_task_response
        ).
        MESSAGE 'Task success sent successfully.' TYPE 'I'.
      CATCH /aws1/cx_sfninvalidtoken.
        MESSAGE 'Invalid task token.' TYPE 'E'.
      CATCH /aws1/cx_sfntaskdoesnotexist.
        MESSAGE 'Task does not exist.' TYPE 'E'.
      CATCH /aws1/cx_sfninvalidoutput.
        MESSAGE 'Invalid task output.' TYPE 'E'.
      CATCH /aws1/cx_sfntasktimedout.
        MESSAGE 'Task timed out.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat SendTaskSuccessdi AWS SDK untuk referensi SAP ABAP API.

`StartExecution`

Contoh kode berikut menunjukkan cara menggunakan`StartExecution`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        DATA(lo_result) = lo_sfn->startexecution(
          iv_statemachinearn = iv_state_machine_arn
          iv_input = iv_input
        ).
        ov_execution_arn = lo_result->get_executionarn( ).
        MESSAGE 'Execution started successfully.' TYPE 'I'.
      CATCH /aws1/cx_sfnstatemachinedoes00.
        MESSAGE 'State machine does not exist.' TYPE 'E'.
      CATCH /aws1/cx_sfninvalidarn.
```

```
      MESSAGE 'Invalid state machine ARN.' TYPE 'E'.
    CATCH /aws1/cx_sfninvalidexecinput.
      MESSAGE 'Invalid execution input.' TYPE 'E'.
    CATCH /aws1/cx_sfnexeclimitexceeded.
      MESSAGE 'Execution limit exceeded.' TYPE 'E'.
  ENDTRY.
```

- Untuk detail API, lihat StartExecutiondi AWS SDK untuk referensi SAP ABAP API.

Contoh Systems Manager menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan mengimplementasikan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Systems Manager.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`CreateDocument`

Contoh kode berikut menunjukkan cara menggunakan`CreateDocument`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_ssm->createdocument(
        iv_name = iv_name
        iv_content = iv_content
        iv_documenttype = 'Command' ).
    MESSAGE 'Document created.' TYPE 'I'.
  CATCH /aws1/cx_ssmdocalreadyexists.
    MESSAGE 'Document already exists.' TYPE 'I'.
  CATCH /aws1/cx_ssminvaliddoccontent.
    MESSAGE 'Invalid document content.' TYPE 'I'.
ENDTRY.
```

- Untuk detail API, lihat CreateDocumentdi AWS SDK untuk referensi SAP ABAP API.

`CreateMaintenanceWindow`

Contoh kode berikut menunjukkan cara menggunakan`CreateMaintenanceWindow`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_ssm->createmaintenancewindow(
        iv_name = iv_name
        iv_schedule = iv_schedule
        iv_duration = iv_duration
        iv_cutoff = iv_cutoff
        iv_allowunassociatedtargets = iv_allow_unassociated_targets ).
    MESSAGE 'Maintenance window created.' TYPE 'I'.
  CATCH /aws1/cx_ssmresrclimitexcdex.
    MESSAGE 'Resource limit exceeded.' TYPE 'I'.
ENDTRY.
```

- Untuk detail API, lihat CreateMaintenanceWindowdi AWS SDK untuk referensi SAP ABAP API.

`CreateOpsItem`

Contoh kode berikut menunjukkan cara menggunakan`CreateOpsItem`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_ssm->createopsitem(
        iv_title = iv_title
        iv_source = iv_source
        iv_category = iv_category
        iv_severity = iv_severity
        iv_description = iv_description ).
    MESSAGE 'OpsItem created.' TYPE 'I'.
  CATCH /aws1/cx_ssmopsitemlimitexcdex.
    MESSAGE 'You have exceeded your open OpsItem limit.' TYPE 'I'.
  CATCH /aws1/cx_ssmopsitemalrdyexex.
    MESSAGE 'OpsItem already exists.' TYPE 'I'.
ENDTRY.
```

- Untuk detail API, lihat CreateOpsItemdi AWS SDK untuk referensi SAP ABAP API.

`DeleteDocument`

Contoh kode berikut menunjukkan cara menggunakan`DeleteDocument`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_ssm->deletedocument( iv_name = iv_name ).
    MESSAGE 'Document deleted.' TYPE 'I'.
  CATCH /aws1/cx_ssminvaliddocument.
    MESSAGE 'Invalid document.' TYPE 'I'.
  CATCH /aws1/cx_ssmassocdinstances.
    MESSAGE 'Document has associated instances.' TYPE 'I'.
ENDTRY.
```

- Untuk detail API, lihat DeleteDocumentdi AWS SDK untuk referensi SAP ABAP API.

`DeleteMaintenanceWindow`

Contoh kode berikut menunjukkan cara menggunakan`DeleteMaintenanceWindow`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_ssm->deletemaintenancewindow( iv_windowid = iv_window_id ).
    MESSAGE 'Maintenance window deleted.' TYPE 'I'.
  CATCH /aws1/cx_ssminternalservererr.
    MESSAGE 'Internal server error occurred.' TYPE 'I'.
ENDTRY.
```

- Untuk detail API, lihat DeleteMaintenanceWindowdi AWS SDK untuk referensi SAP ABAP API.

`DeleteOpsItem`

Contoh kode berikut menunjukkan cara menggunakan`DeleteOpsItem`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_ssm->deleteopsitem( iv_opsitemid = iv_ops_item_id ).
    MESSAGE 'OpsItem deleted.' TYPE 'I'.
  CATCH /aws1/cx_ssmopsiteminvparamex.
    MESSAGE 'Invalid OpsItem parameter.' TYPE 'I'.
ENDTRY.
```

- Untuk detail API, lihat DeleteOpsItemdi AWS SDK untuk referensi SAP ABAP API.

`DescribeDocument`

Contoh kode berikut menunjukkan cara menggunakan`DescribeDocument`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    DATA(lo_result) = lo_ssm->describedocument( iv_name = iv_name ).
    DATA(lo_document) = lo_result->get_document( ).
    IF lo_document IS BOUND.
      rv_status = lo_document->get_status( ).
      MESSAGE |Document status: { rv_status }| TYPE 'I'.
    ENDIF.
  CATCH /aws1/cx_ssminvaliddocument.
    MESSAGE 'Invalid document.' TYPE 'I'.
```

```
    ENDTRY.
```

- Untuk detail API, lihat DescribeDocumentdi AWS SDK untuk referensi SAP ABAP API.

`DescribeOpsItems`

Contoh kode berikut menunjukkan cara menggunakan`DescribeOpsItems`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " Create filter for OpsItem ID
        DATA(lt_filters) = VALUE /aws1/cl_ssmopsitemfilter=>tt_opsitemfilters(
          ( NEW /aws1/cl_ssmopsitemfilter(
              iv_key = 'OpsItemId'
              it_values = VALUE /aws1/
cl_ssmopsitemfiltvals_w=>tt_opsitemfiltervalues(
                ( NEW /aws1/cl_ssmopsitemfiltvals_w( iv_value = iv_ops_item_id ) )
              )
              iv_operator = 'Equal'
            ) )
        ).

        " Use paginator to get all results
        DATA(lo_paginator) = lo_ssm->get_paginator( ).
        DATA(lo_iterator) = lo_paginator->describeopsitems(
          it_opsitemfilters = lt_filters ).

        rv_found = abap_false.

        WHILE lo_iterator->has_next( ).
          DATA(lo_result) = CAST /aws1/cl_ssmdescropsitemsrsp( lo_iterator-
>get_next( ) ).
          LOOP AT lo_result->get_opsitemsummaries( ) INTO DATA(lo_item).
            DATA(lv_title) = lo_item->get_title( ).
```

```
            DATA(lv_status) = lo_item->get_status( ).
            MESSAGE |The OpsItem title is { lv_title } and the status is
 { lv_status }| TYPE 'I'.
            rv_found = abap_true.
          ENDLOOP.
        ENDWHILE.
      CATCH /aws1/cx_ssminternalservererr.
        MESSAGE 'Internal server error occurred.' TYPE 'I'.
    ENDTRY.
```

- Untuk detail API, lihat DescribeOpsItemsdi AWS SDK untuk referensi SAP ABAP API.

`ListCommandInvocations`

Contoh kode berikut menunjukkan cara menggunakan`ListCommandInvocations`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        " Use paginator to get all results
        DATA(lo_paginator) = lo_ssm->get_paginator( ).
        DATA(lo_iterator) = lo_paginator->listcommandinvocations(
          iv_instanceid = iv_instance_id ).

        DATA lv_count TYPE i VALUE 0.

        WHILE lo_iterator->has_next( ).
          DATA(lo_result) = CAST /aws1/cl_ssmlistcmdinvcsresult( lo_iterator-
>get_next( ) ).
          LOOP AT lo_result->get_commandinvocations( ) INTO DATA(lo_invocation).
            lv_count = lv_count + 1.
            DATA(lv_requested_datetime) = lo_invocation->get_requesteddatetime( ).
            MESSAGE |Command invocation requested at: { lv_requested_datetime }|
 TYPE 'I'.
          ENDLOOP.
```

```
        ENDWHILE.

        MESSAGE |{ lv_count } command invocation(s) found for instance
{ iv_instance_id }.| TYPE 'I'.
      CATCH /aws1/cx_ssminvalidinstanceid.
        MESSAGE 'Invalid instance ID.' TYPE 'I'.
      CATCH /aws1/cx_ssminvalidcommandid.
        MESSAGE 'Invalid command ID.' TYPE 'I'.
    ENDTRY.
```

- Untuk detail API, lihat ListCommandInvocationsdi AWS SDK untuk referensi SAP ABAP API.

`SendCommand`

Contoh kode berikut menunjukkan cara menggunakan`SendCommand`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        DATA(lo_result) = lo_ssm->sendcommand(
            it_instanceids = it_instance_ids
            iv_documentname = iv_document_name
            iv_timeoutseconds = 3600 ).
        DATA(lo_command) = lo_result->get_command( ).
        IF lo_command IS BOUND.
          rv_command_id = lo_command->get_commandid( ).
          MESSAGE 'Command sent successfully.' TYPE 'I'.
        ENDIF.
      CATCH /aws1/cx_ssminvaliddocument.
        MESSAGE 'Invalid document.' TYPE 'I'.
      CATCH /aws1/cx_ssminvalidinstanceid.
        MESSAGE 'Invalid instance ID.' TYPE 'I'.
    ENDTRY.
```

- Untuk detail API, lihat SendCommanddi AWS SDK untuk referensi SAP ABAP API.

`UpdateMaintenanceWindow`

Contoh kode berikut menunjukkan cara menggunakan`UpdateMaintenanceWindow`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_ssm->updatemaintenancewindow(
        iv_windowid = iv_window_id
        iv_name = iv_name
        iv_enabled = iv_enabled
        iv_schedule = iv_schedule
        iv_duration = iv_duration
        iv_cutoff = iv_cutoff
        iv_allowunassociatedtargets = iv_allow_unassociated_targets ).
    MESSAGE 'Maintenance window updated.' TYPE 'I'.
  CATCH /aws1/cx_ssmdoesnotexistex.
    MESSAGE 'Maintenance window does not exist.' TYPE 'I'.
ENDTRY.
```

- Untuk detail API, lihat UpdateMaintenanceWindowdi AWS SDK untuk referensi SAP ABAP API.

`UpdateOpsItem`

Contoh kode berikut menunjukkan cara menggunakan`UpdateOpsItem`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_ssm->updateopsitem(
        iv_opsitemid = iv_ops_item_id
        iv_title = iv_title
        iv_description = iv_description
        iv_status = iv_status ).
    MESSAGE 'OpsItem updated.' TYPE 'I'.
  CATCH /aws1/cx_ssmopsitemnotfoundex.
    MESSAGE 'OpsItem not found.' TYPE 'I'.
  CATCH /aws1/cx_ssmopsiteminvparamex.
    MESSAGE 'Invalid OpsItem parameter.' TYPE 'I'.
ENDTRY.
```

- Untuk detail API, lihat UpdateOpsItemdi AWS SDK untuk referensi SAP ABAP API.

Contoh Amazon Texttract menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Amazon Textract.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Skenario adalah contoh kode yang menunjukkan kepada Anda bagaimana menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan atau dikombinasikan dengan yang lain Layanan AWS.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan
- Skenario

Tindakan

`AnalyzeDocument`

Contoh kode berikut menunjukkan cara menggunakan`AnalyzeDocument`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat AnalyzeDocumentdi AWS SDK untuk referensi SAP ABAP API.

`DetectDocumentText`

Contoh kode berikut menunjukkan cara menggunakan`DetectDocumentText`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode 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.
```

- Untuk detail API, lihat DetectDocumentTextdi AWS SDK untuk referensi SAP ABAP API.

`GetDocumentAnalysis`

Contoh kode berikut menunjukkan cara menggunakan`GetDocumentAnalysis`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

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

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

- Untuk detail API, lihat GetDocumentAnalysisdi AWS SDK untuk referensi SAP ABAP API.

`StartDocumentAnalysis`

Contoh kode berikut menunjukkan cara menggunakan`StartDocumentAnalysis`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
"Starts the asynchronous analysis of an input document for relationships"
```

```
   "between detected items such as key-value pairs, tables, and selection
elements."

   "Create ABAP objects for feature type."
   "Add TABLES to return information about the tables."
   "Add FORMS to return detected form data."
   "To perform both types of analysis, add TABLES and FORMS to FeatureTypes."

   DATA(lt_featuretypes) = VALUE /aws1/cl_texfeaturetypes_w=>tt_featuretypes(
     ( NEW /aws1/cl_texfeaturetypes_w( iv_value = 'FORMS' ) )
     ( NEW /aws1/cl_texfeaturetypes_w( iv_value = 'TABLES' ) ) ).
   "Create an ABAP object for the Amazon S3 object."
   DATA(lo_s3object) = NEW /aws1/cl_texs3object( iv_bucket = iv_s3bucket
     iv_name   = iv_s3object ).
   "Create an ABAP object for the document."
   DATA(lo_documentlocation) = NEW /aws1/cl_texdocumentlocation( io_s3object =
lo_s3object ).

   "Start async document analysis."
   TRY.
       oo_result = lo_tex->startdocumentanalysis(      "oo_result is returned for
testing purposes."
         io_documentlocation     = lo_documentlocation
         it_featuretypes         = lt_featuretypes ).
       DATA(lv_jobid) = oo_result->get_jobid( ).

       MESSAGE 'Document analysis started.' TYPE 'I'.
     CATCH /aws1/cx_texaccessdeniedex.
       MESSAGE 'You do not have permission to perform this action.' TYPE 'E'.
     CATCH /aws1/cx_texbaddocumentex.
       MESSAGE 'Amazon Textract is not able to read the document.' TYPE 'E'.
     CATCH /aws1/cx_texdocumenttoolargeex.
       MESSAGE 'The document is too large.' TYPE 'E'.
     CATCH /aws1/cx_texidempotentprmmis00.
       MESSAGE 'Idempotent parameter mismatch exception.' TYPE 'E'.
     CATCH /aws1/cx_texinternalservererr.
       MESSAGE 'Internal server error.' TYPE 'E'.
     CATCH /aws1/cx_texinvalidkmskeyex.
       MESSAGE 'AWS KMS key is not valid.' TYPE 'E'.
     CATCH /aws1/cx_texinvalidparameterex.
       MESSAGE 'Request has non-valid parameters.' TYPE 'E'.
     CATCH /aws1/cx_texinvalids3objectex.
       MESSAGE 'Amazon S3 object is not valid.' TYPE 'E'.
     CATCH /aws1/cx_texlimitexceededex.
```

```
        MESSAGE 'An Amazon Textract service limit was exceeded.' TYPE 'E'.
      CATCH /aws1/cx_texprovthruputexcdex.
        MESSAGE 'Provisioned throughput exceeded limit.' TYPE 'E'.
      CATCH /aws1/cx_texthrottlingex.
        MESSAGE 'The request processing exceeded the limit.' TYPE 'E'.
      CATCH /aws1/cx_texunsupporteddocex.
        MESSAGE 'The document is not supported.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat StartDocumentAnalysisdi AWS SDK untuk referensi SAP ABAP API.

`StartDocumentTextDetection`

Contoh kode berikut menunjukkan cara menggunakan`StartDocumentTextDetection`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

    "Create an ABAP object for the Amazon S3 object."
    DATA(lo_s3object) = NEW /aws1/cl_texs3object( iv_bucket = iv_s3bucket
      iv_name   = iv_s3object ).
    "Create an ABAP object for the document."
    DATA(lo_documentlocation) = NEW /aws1/cl_texdocumentlocation( io_s3object =
lo_s3object ).
    "Start document analysis."
    TRY.
        oo_result = lo_tex->startdocumenttextdetection( io_documentlocation =
lo_documentlocation ).
        DATA(lv_jobid) = oo_result->get_jobid( ).            "oo_result is returned
for testing purposes."
```

```
        MESSAGE 'Document analysis started.' TYPE 'I'.
      CATCH /aws1/cx_texaccessdeniedex.
        MESSAGE 'You do not have permission to perform this action.' TYPE 'E'.
      CATCH /aws1/cx_texbaddocumentex.
        MESSAGE 'Amazon Textract is not able to read the document.' TYPE 'E'.
      CATCH /aws1/cx_texdocumenttoolargeex.
        MESSAGE 'The document is too large.' TYPE 'E'.
      CATCH /aws1/cx_texidempotentprmmis00.
        MESSAGE 'Idempotent parameter mismatch exception.' TYPE 'E'.
      CATCH /aws1/cx_texinternalservererr.
        MESSAGE 'Internal server error.' TYPE 'E'.
      CATCH /aws1/cx_texinvalidkmskeyex.
        MESSAGE 'AWS KMS key is not valid.' TYPE 'E'.
      CATCH /aws1/cx_texinvalidparameterex.
        MESSAGE 'Request has non-valid parameters.' TYPE 'E'.
      CATCH /aws1/cx_texinvalids3objectex.
        MESSAGE 'Amazon S3 object is not valid.' TYPE 'E'.
      CATCH /aws1/cx_texlimitexceededex.
        MESSAGE 'An Amazon Textract service limit was exceeded.' TYPE 'E'.
      CATCH /aws1/cx_texprovthruputexcdex.
        MESSAGE 'Provisioned throughput exceeded limit.' TYPE 'E'.
      CATCH /aws1/cx_texthrottlingex.
        MESSAGE 'The request processing exceeded the limit.' TYPE 'E'.
      CATCH /aws1/cx_texunsupporteddocex.
        MESSAGE 'The document is not supported.' TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat StartDocumentTextDetectiondi AWS SDK untuk referensi SAP ABAP API.

Skenario

Memulai dengan analisis dokumen

Contoh kode berikut ini menunjukkan cara untuk melakukan:

- Mulai analisis asinkron.
- Dapatkan analisis dokumen.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   "Create ABAP objects for feature type."
   "Add TABLES to return information about the tables."
   "Add FORMS to return detected form data."
   "To perform both types of analysis, add TABLES and FORMS to FeatureTypes."

   DATA(lt_featuretypes) = VALUE /aws1/cl_texfeaturetypes_w=>tt_featuretypes(
     ( NEW /aws1/cl_texfeaturetypes_w( iv_value = 'FORMS' ) )
     ( NEW /aws1/cl_texfeaturetypes_w( iv_value = 'TABLES' ) ) ).

   "Create an ABAP object for the Amazon Simple Storage Service (Amazon S3)
object."
   DATA(lo_s3object) = NEW /aws1/cl_texs3object( iv_bucket = iv_s3bucket
     iv_name   = iv_s3object ).

   "Create an ABAP object for the document."
   DATA(lo_documentlocation) = NEW /aws1/cl_texdocumentlocation( io_s3object =
lo_s3object ).

   "Start document analysis."
   TRY.
       DATA(lo_start_result) = lo_tex->startdocumentanalysis(
         io_documentlocation     = lo_documentlocation
         it_featuretypes         = lt_featuretypes ).
       MESSAGE 'Document analysis started.' TYPE 'I'.
     CATCH /aws1/cx_texaccessdeniedex.
       MESSAGE 'You do not have permission to perform this action.' TYPE 'E'.
     CATCH /aws1/cx_texbaddocumentex.
       MESSAGE 'Amazon Textract is not able to read the document.' TYPE 'E'.
     CATCH /aws1/cx_texdocumenttoolargeex.
       MESSAGE 'The document is too large.' TYPE 'E'.
     CATCH /aws1/cx_texidempotentprmmis00.
       MESSAGE 'Idempotent parameter mismatch exception.' TYPE 'E'.
     CATCH /aws1/cx_texinternalservererr.
```

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

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

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

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

- Untuk mengetahui hal detail mengenai API, silakan lihat topik-topik berikut di referensi API AWS SDK untuk ABAP SAP.
  - GetDocumentAnalysis

- StartDocumentAnalysis

Contoh Amazon Transcribe menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan mengimplementasikan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Amazon Transcribe.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan

Tindakan

`CreateVocabulary`

Contoh kode berikut menunjukkan cara menggunakan`CreateVocabulary`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    IF it_phrases IS NOT INITIAL.
      oo_result = lo_tnb->createvocabulary(
        iv_vocabularyname = iv_vocabulary_name
        iv_languagecode = iv_language_code
        it_phrases = it_phrases ).
    ELSEIF iv_vocab_file_uri IS NOT INITIAL.
```

```
        oo_result = lo_tnb->createvocabulary(
          iv_vocabularyname = iv_vocabulary_name
          iv_languagecode = iv_language_code
          iv_vocabularyfileuri = iv_vocab_file_uri ).
      ENDIF.
      MESSAGE 'Custom vocabulary created.' TYPE 'I'.
    CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
      MESSAGE lo_bad_request_ex TYPE 'I'.
      RAISE EXCEPTION lo_bad_request_ex.
    CATCH /aws1/cx_tnblimitexceededex INTO DATA(lo_limit_ex).
      MESSAGE lo_limit_ex TYPE 'I'.
      RAISE EXCEPTION lo_limit_ex.
    CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
      MESSAGE lo_internal_ex TYPE 'I'.
      RAISE EXCEPTION lo_internal_ex.
    CATCH /aws1/cx_tnbconflictexception INTO DATA(lo_conflict_ex).
      MESSAGE lo_conflict_ex TYPE 'I'.
      RAISE EXCEPTION lo_conflict_ex.
  ENDTRY.
```

- Untuk detail API, lihat CreateVocabularydi AWS SDK untuk referensi SAP ABAP API.

`DeleteTranscriptionJob`

Contoh kode berikut menunjukkan cara menggunakan`DeleteTranscriptionJob`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
  TRY.
      lo_tnb->deletetranscriptionjob( iv_job_name ).
      MESSAGE 'Transcription job deleted.' TYPE 'I'.
    CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
      MESSAGE lo_bad_request_ex TYPE 'I'.
      RAISE EXCEPTION lo_bad_request_ex.
```

```
  CATCH /aws1/cx_tnblimitexceededex INTO DATA(lo_limit_ex).
    MESSAGE lo_limit_ex TYPE 'I'.
    RAISE EXCEPTION lo_limit_ex.
  CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
    MESSAGE lo_internal_ex TYPE 'I'.
    RAISE EXCEPTION lo_internal_ex.
ENDTRY.
```

- Untuk detail API, lihat DeleteTranscriptionJobdi AWS SDK untuk referensi SAP ABAP API.

`DeleteVocabulary`

Contoh kode berikut menunjukkan cara menggunakan`DeleteVocabulary`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    lo_tnb->deletevocabulary( iv_vocabulary_name ).
    MESSAGE 'Vocabulary deleted.' TYPE 'I'.
  CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
    MESSAGE lo_bad_request_ex TYPE 'I'.
  CATCH /aws1/cx_tnblimitexceededex INTO DATA(lo_limit_ex).
    MESSAGE lo_limit_ex TYPE 'I'.
    RAISE EXCEPTION lo_limit_ex.
  CATCH /aws1/cx_tnbnotfoundexception INTO DATA(lo_not_found_ex).
    MESSAGE lo_not_found_ex TYPE 'I'.
  CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
    MESSAGE lo_internal_ex TYPE 'I'.
    RAISE EXCEPTION lo_internal_ex.
ENDTRY.
```

- Untuk detail API, lihat DeleteVocabularydi AWS SDK untuk referensi SAP ABAP API.

`GetTranscriptionJob`

Contoh kode berikut menunjukkan cara menggunakan`GetTranscriptionJob`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    oo_result = lo_tnb->gettranscriptionjob( iv_job_name ).
    DATA(lo_job) = oo_result->get_transcriptionjob( ).
    MESSAGE 'Retrieved transcription job details.' TYPE 'I'.
  CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
    MESSAGE lo_bad_request_ex TYPE 'I'.
    RAISE EXCEPTION lo_bad_request_ex.
  CATCH /aws1/cx_tnbnotfoundexception INTO DATA(lo_not_found_ex).
    MESSAGE lo_not_found_ex TYPE 'I'.
    RAISE EXCEPTION lo_not_found_ex.
  CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
    MESSAGE lo_internal_ex TYPE 'I'.
    RAISE EXCEPTION lo_internal_ex.
ENDTRY.
```

- Untuk detail API, lihat GetTranscriptionJobdi AWS SDK untuk referensi SAP ABAP API.

`GetVocabulary`

Contoh kode berikut menunjukkan cara menggunakan`GetVocabulary`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        oo_result = lo_tnb->getvocabulary( iv_vocabulary_name ).
        MESSAGE 'Retrieved vocabulary details.' TYPE 'I'.
      CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_tnbnotfoundexception INTO DATA(lo_not_found_ex).
        MESSAGE lo_not_found_ex TYPE 'I'.
        RAISE EXCEPTION lo_not_found_ex.
      CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
        MESSAGE lo_internal_ex TYPE 'I'.
        RAISE EXCEPTION lo_internal_ex.
    ENDTRY.
```

- Untuk detail API, lihat GetVocabularydi AWS SDK untuk referensi SAP ABAP API.

`ListTranscriptionJobs`

Contoh kode berikut menunjukkan cara menggunakan`ListTranscriptionJobs`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
    TRY.
        IF iv_job_filter IS NOT INITIAL.
          oo_result = lo_tnb->listtranscriptionjobs( iv_jobnamecontains =
iv_job_filter ).
        ELSE.
          oo_result = lo_tnb->listtranscriptionjobs( ).
        ENDIF.
        MESSAGE 'Retrieved transcription jobs list.' TYPE 'I'.
      CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
```

```
    MESSAGE lo_internal_ex TYPE 'I'.
    RAISE EXCEPTION lo_internal_ex.
ENDTRY.
```

- Untuk detail API, lihat ListTranscriptionJobsdi AWS SDK untuk referensi SAP ABAP API.

`ListVocabularies`

Contoh kode berikut menunjukkan cara menggunakan`ListVocabularies`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    IF iv_vocab_filter IS NOT INITIAL.
      oo_result = lo_tnb->listvocabularies( iv_namecontains = iv_vocab_filter ).
    ELSE.
      oo_result = lo_tnb->listvocabularies( ).
    ENDIF.
    MESSAGE 'Retrieved vocabularies list.' TYPE 'I'.
  CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
    MESSAGE lo_bad_request_ex TYPE 'I'.
    RAISE EXCEPTION lo_bad_request_ex.
  CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
    MESSAGE lo_internal_ex TYPE 'I'.
    RAISE EXCEPTION lo_internal_ex.
ENDTRY.
```

- Untuk detail API, lihat ListVocabulariesdi AWS SDK untuk referensi SAP ABAP API.

`StartTranscriptionJob`

Contoh kode berikut menunjukkan cara menggunakan`StartTranscriptionJob`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   TRY.
       DATA(lo_media) = NEW /aws1/cl_tnbmedia( iv_mediafileuri = iv_media_uri ).
       DATA(lo_settings) = NEW /aws1/cl_tnbsettings( ).
       IF iv_vocabulary_name IS NOT INITIAL.
         lo_settings = NEW /aws1/cl_tnbsettings( iv_vocabularyname =
iv_vocabulary_name ).
       ENDIF.

       oo_result = lo_tnb->starttranscriptionjob(
         iv_transcriptionjobname = iv_job_name
         io_media = lo_media
         iv_mediaformat = iv_media_format
         iv_languagecode = iv_language_code
         io_settings = lo_settings ).

       MESSAGE 'Transcription job started.' TYPE 'I'.
     CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
       MESSAGE lo_bad_request_ex TYPE 'I'.
       RAISE EXCEPTION lo_bad_request_ex.
     CATCH /aws1/cx_tnblimitexceededex INTO DATA(lo_limit_ex).
       MESSAGE lo_limit_ex TYPE 'I'.
       RAISE EXCEPTION lo_limit_ex.
     CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
       MESSAGE lo_internal_ex TYPE 'I'.
       RAISE EXCEPTION lo_internal_ex.
     CATCH /aws1/cx_tnbconflictexception INTO DATA(lo_conflict_ex).
       MESSAGE lo_conflict_ex TYPE 'I'.
       RAISE EXCEPTION lo_conflict_ex.
   ENDTRY.
```

- Untuk detail API, lihat StartTranscriptionJobdi AWS SDK untuk referensi SAP ABAP API.

`UpdateVocabulary`

Contoh kode berikut menunjukkan cara menggunakan`UpdateVocabulary`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
TRY.
    IF it_phrases IS NOT INITIAL.
      oo_result = lo_tnb->updatevocabulary(
        iv_vocabularyname = iv_vocabulary_name
        iv_languagecode = iv_language_code
        it_phrases = it_phrases ).
    ELSEIF iv_vocab_file_uri IS NOT INITIAL.
      oo_result = lo_tnb->updatevocabulary(
        iv_vocabularyname = iv_vocabulary_name
        iv_languagecode = iv_language_code
        iv_vocabularyfileuri = iv_vocab_file_uri ).
    ENDIF.
    MESSAGE 'Vocabulary updated.' TYPE 'I'.
  CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
    MESSAGE lo_bad_request_ex TYPE 'I'.
  CATCH /aws1/cx_tnblimitexceededex INTO DATA(lo_limit_ex).
    MESSAGE lo_limit_ex TYPE 'I'.
    RAISE EXCEPTION lo_limit_ex.
  CATCH /aws1/cx_tnbnotfoundexception INTO DATA(lo_not_found_ex).
    MESSAGE lo_not_found_ex TYPE 'I'.
  CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
    MESSAGE lo_internal_ex TYPE 'I'.
    RAISE EXCEPTION lo_internal_ex.
  CATCH /aws1/cx_tnbconflictexception INTO DATA(lo_conflict_ex).
    MESSAGE lo_conflict_ex TYPE 'I'.
    RAISE EXCEPTION lo_conflict_ex.
ENDTRY.
```

- Untuk detail API, lihat UpdateVocabularydi AWS SDK untuk referensi SAP ABAP API.

Contoh Amazon Translate menggunakan SDK untuk SAP ABAP

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK untuk SAP ABAP dengan Amazon Translate.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Skenario adalah contoh kode yang menunjukkan kepada Anda bagaimana menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan atau dikombinasikan dengan yang lain Layanan AWS.

Setiap contoh menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- Tindakan
- Skenario

Tindakan

`DescribeTextTranslationJob`

Contoh kode berikut menunjukkan cara menggunakan`DescribeTextTranslationJob`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   "Gets the properties associated with an asynchronous batch translation job."
   "Includes properties such as name, ID, status, source and target languages, and
input/output Amazon Simple Storage Service (Amazon S3) buckets."
```

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

- Untuk detail API, lihat DescribeTextTranslationJobdi AWS SDK untuk referensi SAP ABAP API.

`ListTextTranslationJobs`

Contoh kode berikut menunjukkan cara menggunakan`ListTextTranslationJobs`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

   DATA lo_filter TYPE REF TO /aws1/cl_xl8textxlationjobfilt.

   "Create an ABAP object for filtering using jobname."
   lo_filter = NEW #( iv_jobname = iv_jobname ).

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

```
      CATCH /aws1/cx_xl8internalserverex.
        MESSAGE 'An internal server error occurred. Retry your request.' TYPE 'E'.
      CATCH /aws1/cx_xl8invalidfilterex.
        MESSAGE 'The filter specified for the operation is not valid. Specify a
 different filter.' TYPE 'E'.
      CATCH /aws1/cx_xl8invalidrequestex.
        MESSAGE 'The request that you made is not valid.' TYPE 'E'.
      CATCH /aws1/cx_xl8toomanyrequestsex.
        MESSAGE 'You have made too many requests within a short period of time.'
 TYPE 'E'.
    ENDTRY.
```

- Untuk detail API, lihat ListTextTranslationJobsdi AWS SDK untuk referensi SAP ABAP API.

`StartTextTranslationJob`

Contoh kode berikut menunjukkan cara menggunakan`StartTextTranslationJob`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

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

    "Create an ABAP object for the input data config."
    lo_inputdataconfig = NEW #( iv_s3uri = iv_input_data_s3uri
                                iv_contenttype = iv_input_data_contenttype ).
```

```
   "Create an ABAP object for the output data config."
   lo_outputdataconfig = NEW #( iv_s3uri = iv_output_data_s3uri ).

   "Create an internal table for target languages."
   lo_targetlanguagecodes = NEW #( iv_value = iv_targetlanguagecode ).
   INSERT lo_targetlanguagecodes  INTO TABLE lt_targetlanguagecodes.

   TRY.
       oo_result = lo_xl8->starttexttranslationjob(      "oo_result is returned for
testing purposes."
        io_inputdataconfig = lo_inputdataconfig
          io_outputdataconfig = lo_outputdataconfig
          it_targetlanguagecodes = lt_targetlanguagecodes
          iv_dataaccessrolearn = iv_dataaccessrolearn
          iv_jobname = iv_jobname
          iv_sourcelanguagecode = iv_sourcelanguagecode ).
       MESSAGE 'Translation job started.' TYPE 'I'.
     CATCH /aws1/cx_xl8internalserverex.
       MESSAGE 'An internal server error occurred. Retry your request.' TYPE 'E'.
     CATCH /aws1/cx_xl8invparamvalueex.
       MESSAGE 'The value of the parameter is not valid.' TYPE 'E'.
     CATCH /aws1/cx_xl8invalidrequestex.
       MESSAGE 'The request that you made is not valid.' TYPE 'E'.
     CATCH /aws1/cx_xl8resourcenotfoundex.
       MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
     CATCH /aws1/cx_xl8toomanyrequestsex.
       MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
     CATCH /aws1/cx_xl8unsuppedlanguage00.
       MESSAGE 'Amazon Translate does not support translation from the language of
the source text into the requested target language.' TYPE 'E'.
   ENDTRY.
```

- Untuk detail API, lihat StartTextTranslationJobdi AWS SDK untuk referensi SAP ABAP API.

`StopTextTranslationJob`

Contoh kode berikut menunjukkan cara menggunakan`StopTextTranslationJob`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

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

- Untuk detail API, lihat StopTextTranslationJobdi AWS SDK untuk referensi SAP ABAP API.

`TranslateText`

Contoh kode berikut menunjukkan cara menggunakan`TranslateText`.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

```
   "Translates input text from the source language to the target language."
```

```
   TRY.
       oo_result = lo_xl8->translatetext(      "oo_result is returned for testing
purposes."
         iv_text        = iv_text
           iv_sourcelanguagecode = iv_sourcelanguagecode
           iv_targetlanguagecode = iv_targetlanguagecode ).
       MESSAGE 'Translation completed.' TYPE 'I'.
     CATCH /aws1/cx_xl8detectedlanguage00.
       MESSAGE 'The confidence that Amazon Comprehend accurately detected the
source language is low.' TYPE 'E'.
     CATCH /aws1/cx_xl8internalserverex.
       MESSAGE 'An internal server error occurred.' TYPE 'E'.
     CATCH /aws1/cx_xl8invalidrequestex.
       MESSAGE 'The request that you made is not valid.' TYPE 'E'.
     CATCH /aws1/cx_xl8resourcenotfoundex.
       MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
     CATCH /aws1/cx_xl8serviceunavailex.
       MESSAGE 'The Amazon Translate service is temporarily unavailable.' TYPE 'E'.
     CATCH /aws1/cx_xl8textsizelmtexcdex.
       MESSAGE 'The size of the text you submitted exceeds the size limit. ' TYPE
'E'.
     CATCH /aws1/cx_xl8toomanyrequestsex.
       MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
     CATCH /aws1/cx_xl8unsuppedlanguage00.
       MESSAGE 'Amazon Translate does not support translation from the language of
the source text into the requested target language. ' TYPE 'E'.
   ENDTRY.
```

- Untuk detail API, lihat TranslateTextdi AWS SDK untuk referensi SAP ABAP API.

Skenario

Memulai dengan menerjemahkan pekerjaan

Contoh kode berikut ini menunjukkan cara untuk melakukan:

- Mulai pekerjaan terjemahan batch asinkron.
- Tunggu pekerjaan asinkron selesai.
- Jelaskan pekerjaan asinkron.

SDK for SAP ABAP

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di Repositori Contoh Kode AWS.

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

    "Create an ABAP object for the input data config."
    lo_inputdataconfig = NEW #( iv_s3uri = iv_input_data_s3uri
                                iv_contenttype = iv_input_data_contenttype ).

    "Create an ABAP object for the output data config."
    lo_outputdataconfig = NEW #( iv_s3uri = iv_output_data_s3uri ).

    "Create an internal table for target languages."
    lo_targetlanguagecodes = NEW #( iv_value = iv_targetlanguagecode ).
    INSERT lo_targetlanguagecodes  INTO TABLE lt_targetlanguagecodes.

    TRY.
        DATA(lo_translationjob_result) = lo_xl8->starttexttranslationjob(
          io_inputdataconfig = lo_inputdataconfig
            io_outputdataconfig = lo_outputdataconfig
            it_targetlanguagecodes = lt_targetlanguagecodes
            iv_dataaccessrolearn = iv_dataaccessrolearn
            iv_jobname = iv_jobname
            iv_sourcelanguagecode = iv_sourcelanguagecode ).
        MESSAGE 'Translation job started.' TYPE 'I'.
      CATCH /aws1/cx_xl8internalserverex.
        MESSAGE 'An internal server error occurred. Retry your request.' TYPE 'E'.
      CATCH /aws1/cx_xl8invparamvalueex.
        MESSAGE 'The value of the parameter is not valid.' TYPE 'E'.
      CATCH /aws1/cx_xl8invalidrequestex.
        MESSAGE 'The request that you made is not valid.' TYPE 'E'.
      CATCH /aws1/cx_xl8resourcenotfoundex.
```

```
        MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
      CATCH /aws1/cx_xl8toomanyrequestsex.
        MESSAGE 'You have made too many requests within a short period of time. '
 TYPE 'E'.
      CATCH /aws1/cx_xl8unsuppedlanguage00.
        MESSAGE 'Amazon Translate does not support translation from the language of
 the source text into the requested target language.' TYPE 'E'.
    ENDTRY.

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

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

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

- Untuk mengetahui hal detail mengenai API, silakan lihat topik-topik berikut di referensi API AWS SDK untuk ABAP SAP.

- DescribeTextTranslationJob
- StartTextTranslationJob

Menggunakan AWS SDK untuk SAP ABAP Knowledge MCP Server

AWS SDK untuk SAP ABAP Knowledge MCP Server adalah sumber pengetahuan khusus yang memungkinkan asisten pengkodean AI menghasilkan kode ABAP yang akurat untuk integrasi. AWS Dengan menghubungkan IDE berkemampuan AI Anda ke server ini, Anda dapat menjelaskan apa yang ingin Anda bangun dalam bahasa alami dan menerima kode ABAP yang benar secara sintaksis yang memanggil SDK untuk SAP ABAP dengan benar. AWS

Server MCP menggunakan Model Context Protocol (MCP), standar terbuka untuk menghubungkan asisten AI ke sumber pengetahuan eksternal. Server diperbarui setiap hari selaras dengan AWS SDK untuk rilis SAP ABAP, sehingga IDE Anda selalu memiliki akses ke informasi SDK yang akurat dan terkini.

AWS SDK untuk SAP ABAP Knowledge MCP Server tersedia tanpa biaya tambahan. Anda hanya membayar AWS sumber daya dan layanan yang Anda konsumsi dalam aplikasi SAP Anda dengan AWS SDK untuk SAP ABAP.

Topik

- Manfaat
- Menyiapkan server MCP
- Apa yang dapat dilakukan oleh server MCP
- Pertimbangan penting

Manfaat

AWS SDK untuk SAP ABAP Knowledge MCP Server memberikan manfaat berikut bagi pengembang ABAP.

- Pembuatan kode yang akurat - Asisten pengkodean AI Anda memperoleh pengetahuan mendalam tentang AWS SDK untuk pola spesifik SAP ABAP, tanda tangan metode, tipe data, dan penanganan pengecualian. Ini menghilangkan kesalahan umum dan secara signifikan mengurangi kesalahan kompilasi dan waktu debugging.

- Penemuan SDK Instan - Server MCP memperlihatkan AWS SDK lengkap untuk basis pengetahuan SAP ABAP, yang mencakup 200+ layanan. AWS Anda dapat menemukan layanan, operasi, dan tipe data yang tersedia tanpa mencari dokumentasi secara manual.
- Selalu up to date — Basis pengetahuan diperbarui setiap hari selaras dengan rilis SDK. Layanan baru dan tanda tangan metode yang diperbarui segera tersedia untuk IDE Anda.
- Instalasi nol - Pengaturan hanya memerlukan menempelkan URL ke konfigurasi MCP IDE Anda. Tidak ada instalasi perangkat lunak lokal, tidak ada manajemen ketergantungan, dan tidak diperlukan proses persetujuan TI.

Menyiapkan server MCP

Untuk menghubungkan IDE Anda ke AWS SDK untuk SAP ABAP Knowledge MCP Server, tambahkan konfigurasi berikut ke pengaturan MCP IDE Anda. Format konfigurasi dan lokasi file dapat bervariasi menurut IDE; lihat dokumentasi MCP IDE Anda untuk detailnya.

Contoh berikut menunjukkan konfigurasi untuk Kiro. Tambahkan ini ke file konfigurasi MCP ruang kerja Anda `.kiro/settings/mcp.json` atau file konfigurasi tingkat pengguna Anda: `~/.kiro/settings/mcp.json`

```
{
  "mcpServers": {
    "abap-sdk-knowledge": {
      "url": "https://sdk-for-sap-abap-knowledge-mcp.global.api.aws/mcp",
      "disabled": false,
      "autoApprove": ["*"]
    }
  }
}
```

`autoApprove`Pengaturan ini memungkinkan IDE Anda untuk menggunakan alat server MCP tanpa meminta persetujuan pada setiap permintaan. Ini aman karena server MCP hanya-baca dan tidak memodifikasi data atau mengeksekusi kode apa pun.

Setelah menyimpan konfigurasi, IDE Anda akan terhubung ke server MCP secara otomatis. Tidak diperlukan restart di sebagian besar IDEs. Anda dapat memverifikasi koneksi dengan mengajukan pertanyaan kepada asisten AI Anda tentang AWS SDK untuk SAP ABAP, seperti “ AWS Layanan apa yang tersedia di?” AWS SDK untuk SAP ABAP

Jika IDE Anda tidak terhubung ke server MCP, verifikasi bahwa IDE Anda mendukung Protokol Konteks Model dan sintaks file konfigurasi sesuai dengan persyaratan IDE Anda.

> **Note**
>
> Endpoint server MCP menggunakan HTTPS dan tidak memerlukan otentikasi. IDE Anda berkomunikasi dengan server hanya ketika Anda mengajukan pertanyaan asisten AI Anda terkait dengan AWS SDK untuk penggunaan SAP ABAP. Logika bisnis milik Anda dan kode aplikasi SAP tetap berada dalam lingkungan pengembangan Anda.

Apa yang dapat dilakukan oleh server MCP

Setelah terhubung, asisten pengkodean AI Anda dapat menggunakan kemampuan berikut yang disediakan oleh server MCP.

- Penemuan layanan — Daftar dan cari semua AWS layanan yang tersedia di AWS SDK untuk SAP ABAP, termasuk metadata layanan dan nama kelas pabrik.
- Detail operasi — Ambil tanda tangan metode ABAP, parameter input, tipe pengembalian, dan jenis pengecualian untuk operasi SDK apa pun.
- Definisi tipe data — Cari definisi struktur ABAP untuk tipe data SDK, termasuk nama dan tipe bidang.
- Contoh kode - Ambil contoh kode ABAP yang berfungsi untuk operasi tertentu, yang mencakup kasus penggunaan umum untuk setiap layanan.
- Pembuatan sesi dan klien — Hasilkan kode ABAP boilerplate untuk membuat sesi SDK, menginisialisasi klien layanan, dan memanggil operasi dengan penanganan pengecualian yang benar.

Misalnya, Anda dapat meminta asisten AI Anda dengan permintaan seperti berikut ini.

- “Tulis kode ABAP untuk membaca pesan dari antrian SQS.”
- “Hasilkan kode ABAP untuk memanggil model Amazon Bedrock untuk pembuatan teks.”
- “Tunjukkan cara mengunggah file ke Amazon S3 dari ABAP.”

Asisten AI menggunakan pengetahuan server MCP untuk menghasilkan kode ABAP yang benar secara sintaksis yang mengikuti konvensi AWS SDK untuk SAP ABAP, termasuk pembuatan sesi yang tepat, inisialisasi klien, dan pola penanganan pengecualian.

Pertimbangan penting

Ingatlah hal berikut saat menggunakan AWS SDK untuk SAP ABAP Knowledge MCP Server.

- Server MCP menyediakan akses hanya-baca ke pengetahuan SDK. Itu tidak mengeksekusi kode ABAP atau terhubung ke sistem SAP.
- Server MCP hanya mencakup AWS SDK untuk SAP ABAP. Ini tidak memberikan bantuan pemrograman ABAP umum yang tidak terkait dengan AWS SDK untuk penggunaan SAP ABAP.
- Server MCP tidak mendukung pustaka ABAP kustom atau pihak ketiga.
- Server MCP memberikan dokumentasi berdasarkan versi terbaru SDK, yang mungkin tidak cocok dengan versi yang diinstal di sistem SAP Anda.
- Contoh kode yang dihasilkan adalah titik awal. Tinjau dan uji semua kode yang dihasilkan sebelum menggunakannya dalam produksi.
- Kualitas dan format kode yang dihasilkan dapat bervariasi tergantung pada kemampuan asisten AI IDE Anda.

Terpisah dari AWS SDK untuk SAP ABAP Knowledge MCP Server yang dijelaskan di halaman ini, AWS juga menawarkan AWS Knowledge MCP Server, yang menyediakan informasi AWS layanan yang luas, panduan arsitektur, dan pemecahan masalah di semua layanan. AWS Kedua server saling melengkapi dan sangat efektif bila digunakan bersama.

Keamanan di AWS SDK untuk SAP ABAP

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. Model tanggung jawab bersama menjelaskan hal ini sebagai keamanan dari cloud dan keamanan dalam cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang berjalan Layanan AWS di dalamnya AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga secara teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari Program AWS Kepatuhan Program AWS Kepatuhan . Untuk mempelajari tentang program kepatuhan yang berlaku AWS SDK untuk SAP ABAP, lihat Layanan AWS di Lingkup oleh Program Kepatuhan Layanan AWS .
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh Layanan AWS yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, yang mencakup sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang dan peraturan yang berlaku.

Bagian ini mencakup topik-topik berikut.

Topik

- Otentikasi sistem SAP aktif AWS
- Praktik terbaik untuk Keamanan IAM
- Otorisasi SAP
- Operasi yang aman
- Menggunakan Otentikasi Kunci Akses Rahasia dengan Enkripsi SSF
- Menggunakan sertifikat dengan Peran IAM Di Mana Saja
- Menggunakan Profil Sumber untuk Akses Lintas Akun
- Menggunakan SAP Credential Store

Otentikasi sistem SAP aktif AWS

Sebelum sistem SAP dapat melakukan panggilan ke AWS atas nama pengguna SAP, sistem SAP harus mengotentikasi dirinya sendiri. AWS AWS SDK untuk SAP ABAP mendukung tiga metode otentikasi dasar berikut yang dipilih dalam pengaturan profil SDK di. `IMG`

AWS SDK untuk SAP ABAP - edisi BTP hanya dapat diautentikasi dengan the section called “Otentikasi kunci akses rahasia” metode menggunakan SAP Credential Store.

Untuk skenario akses lintas akun, SDK untuk SAP ABAP juga mendukung profil sumber, yang memungkinkan rantai beberapa asumsi peran IAM di seluruh akun menggunakan salah satu metode otentikasi dasar. Untuk informasi selengkapnya, lihat the section called “Otentikasi profil sumber untuk akses lintas akun”.

Topik

- Autentikasi metadata instans Amazon EC2
- Otentikasi kunci akses rahasia
- Otentikasi berbasis sertifikat menggunakan Peran IAM Di Mana Saja
- Otentikasi profil sumber untuk akses lintas akun
- Langkah berikutnya

Autentikasi metadata instans Amazon EC2

Sistem SAP yang berjalan di Amazon EC2 dapat memperoleh kredensil berumur pendek yang berputar secara otomatis dari metadata instans Amazon EC2. Untuk informasi selengkapnya, lihat Menggunakan kredensil untuk metadata instans Amazon EC2.

Kami sangat menyarankan metode otentikasi ini saat menggunakan SDK untuk SAP ABAP. Untuk mengaktifkan, administrator Basis harus mengaktifkan komunikasi HTTP keluar. Tidak diperlukan konfigurasi Basis lebih lanjut.

> **Note**
>
> Metode otentikasi ini hanya berlaku jika sistem SAP Anda berjalan di Amazon EC2. Sistem SAP yang dihosting di tempat atau di lingkungan cloud lainnya tidak dapat mengautentikasi menggunakan metode ini.

Otentikasi kunci akses rahasia

Dengan metode ini, Anda menggunakan ID Kunci Akses dan Kunci Akses Rahasia untuk mengautentikasi sistem SAP Anda. AWS Sistem SAP masuk AWS menggunakan pengguna IAM. Untuk informasi selengkapnya, lihat Mengelola Kunci Akses untuk Pengguna IAM.

Administrator Basis menerima ID Kunci Akses dan Kunci Akses Rahasia dari administrator AWS IAM. Sistem SAP Anda harus dikonfigurasi untuk menyimpan ID Kunci Akses dan Kunci Akses Rahasia.

- Aman, simpan, dan teruskan (SSF)
  - Gunakan fungsionalitas SSF untuk mengautentikasi AWS SDK untuk SAP ABAP. Untuk informasi selengkapnya, lihat Tanda Tangan Digital dan Enkripsi.
  - Anda juga dapat menguji SSF `envelope` dan `develope` fungsionalitas dengan `SSF02` laporan. Untuk informasi selengkapnya, lihat Menguji Instalasi SSF.
  - Langkah-langkah untuk mengkonfigurasi SSF untuk SDK untuk SAP ABAP dijelaskan dalam transaksi. `/AWS1/IMG` Buka Prasyarat Teknis, lalu pilih Pengaturan Tambahan untuk Sistem Lokal. Untuk langkah-langkah konfigurasi terperinci, lihat Menggunakan Otentikasi Kunci Akses Rahasia dengan Enkripsi SSF.
- Toko Kredensi SAP
  - Gunakan SAP Credential Store untuk mengautentikasi AWS SDK untuk SAP ABAP - edisi BTP. Untuk informasi selengkapnya, lihat Apa itu SAP Credential Store?
  - Lihat Menggunakan SAP Credential Store untuk langkah-langkah konfigurasi.

Otentikasi berbasis sertifikat menggunakan Peran IAM Di Mana Saja

Sertifikat X.509 yang dikeluarkan oleh otoritas sertifikat (CA) Anda dapat digunakan untuk otentikasi dengan Roles Anywhere. AWS Identity and Access Management Sertifikat harus dikonfigurasi di`STRUST`. CA harus terdaftar dengan IAM Roles Anywhere sebagai jangkar kepercayaan, dan profil harus dibuat untuk menentukan peran dan kebijakan yang akan diasumsikan oleh IAM Roles Anywhere. Untuk informasi selengkapnya, lihat Membuat jangkar kepercayaan dan profil di AWS Identity and Access Management Peran Di Mana Saja.

Untuk langkah-langkah mendetail tentang cara menggunakan IAM Roles Anywhere with SDK for SAP ABAP, lihat Menggunakan sertifikat dengan IAM Roles Anywhere.

> **Note**
>
> Pencabutan sertifikat hanya didukung melalui penggunaan daftar pencabutan sertifikat yang diimpor. Untuk informasi lebih lanjut, lihat Pencabutan.

Otentikasi profil sumber untuk akses lintas akun

Profil sumber adalah fitur canggih yang memungkinkan Anda untuk menghubungkan beberapa asumsi peran IAM di seluruh AWS akun. Dengan metode ini, satu profil mengasumsikan peran, yang kemudian mengasumsikan peran lain, dan seterusnya, mirip dengan `source_profile` parameter di CLI AWS .

Profil sumber bekerja dengan salah satu dari tiga metode otentikasi dasar (metadata instance, kunci akses rahasia, atau berbasis sertifikat). Profil pertama dalam rantai harus menggunakan salah satu metode dasar ini, dan profil berikutnya dalam rantai menggunakan kredensil dari profil sebelumnya untuk mengambil peran berikutnya.

Ini berguna untuk skenario akses lintas akun di mana Anda perlu melintasi beberapa AWS akun untuk mencapai sumber daya target Anda. Untuk langkah-langkah konfigurasi terperinci, lihat Menggunakan Profil Sumber untuk Akses Lintas Akun.

Langkah berikutnya

Setelah mengautentikasi sistem SAP Anda AWS, SDK untuk SAP ABAP secara otomatis melakukan `sts:assumeRole` untuk mengambil peran IAM yang sesuai untuk fungsi bisnis pengguna SAP.

Praktik terbaik untuk Keamanan IAM

Administrator IAM akan bertanggung jawab atas tiga bidang utama berikut.

- Memastikan bahwa sistem SAP dapat mengautentikasi dirinya sendiri dengan metadata Amazon EC2 atau kredenal Secret Key.
- Memastikan bahwa sistem SAP memiliki izin yang dibutuhkan untuk meningkatkan dirinya sendiri. `sts:assumeRole`
- Untuk setiap peran IAM logis, membuat peran IAM untuk pengguna SAP dengan izin yang diperlukan untuk menjalankan fungsi bisnis (misalnya, izin yang diperlukan untuk Amazon S3, DynamoDB, atau layanan lainnya). Ini adalah peran yang akan diasumsikan oleh pengguna SAP.

Untuk informasi selengkapnya, lihat bagian Keamanan di SAP Lens: AWS Well-Architected Framework.

Topik

- Praktik terbaik untuk profil instans Amazon EC2
- Peran IAM untuk pengguna SAP
- Pertimbangan Keamanan Profil Sumber

Praktik terbaik untuk profil instans Amazon EC2

Instans Amazon EC2 tempat sistem SAP Anda berjalan memiliki serangkaian otorisasi berdasarkan profil instansnya. Umumnya, profil instance hanya perlu memiliki izin untuk memanggil`sts:assumeRole`, untuk memungkinkan sistem SAP untuk mengambil peran IAM khusus bisnis sesuai kebutuhan. Peningkatan ke peran lain ini memastikan bahwa program ABAP dapat mengambil peran yang memberi pengguna hak istimewa paling sedikit yang diperlukan untuk melakukan pekerjaan mereka. Misalnya, profil instance mungkin berisi pernyataan berikut.

JSON

```
{
 "Version":"2012-10-17",
 "Statement": [
   {
     "Sid": "VisualEditor0",
     "Effect": "Allow",
     "Action": "sts:AssumeRole",
     "Resource":[
        "arn:aws:iam::012345678912:role/finance-cfo",
        "arn:aws:iam::012345678912:role/finance-auditor",
        "arn:aws:iam::012345678912:role/finance-reporting"
      ]
   }
 ]
}
```

Contoh sebelumnya ini memungkinkan sistem SAP untuk mengambil peran IAM untuk CFO, AUDITOR, atau pengguna PELAPORAN. AWS SDK akan memilih peran IAM yang benar untuk pengguna berdasarkan peran PFCG pengguna di SAP.

Profil instans Amazon EC2 juga dapat digunakan untuk fungsi lainnya.

- AWS Backint Agent untuk SAP HANA
- SAP pada Ketersediaan AWS Tinggi dengan Overlay IP Address Routing

Solusi ini mungkin juga memerlukan `sts:assumeRole` izin untuk peran khusus untuk pencadangan atau failover atau mereka mungkin memerlukan izin untuk ditetapkan langsung ke profil instans.

Peran IAM untuk pengguna SAP

Program ABAP memerlukan izin untuk melakukan pekerjaan pengguna: membaca tabel DynamoDB, memanggil Amazon Texttract pada objek PDF di Amazon S3, jalankan fungsi. AWS Lambda Model keamanan yang sama digunakan di semua AWS SDKs. Anda dapat menggunakan peran IAM yang sudah ada yang digunakan untuk AWS SDK lain.

Analis bisnis SAP akan meminta administrator IAM untuk arn:aws: dari peran IAM untuk setiap peran logis yang diperlukan. Misalnya, dalam skenario keuangan, analis bisnis dapat menentukan peran IAM logis berikut.

- `CFO`
- `AUDITOR`
- `REPORTING`

Administrator IAM akan mendefinisikan peran IAM untuk setiap peran IAM logis.

`CFO`

- `arn:aws:iam::0123456789:role/finance-cfo`
- izin baca dan tulis ke bucket Amazon S3
- membaca dan menulis izin ke database DynamoDB

`AUDITOR`

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

- baca izin ke bucket Amazon S3
- membaca izin ke database DynamoDB

`REPORTING`

- `arn:aws:iam::0123456789:role/finance-reporting`
- membaca izin ke database DynamoDB
- tidak ada izin untuk ember Amazon S3

Analis bisnis akan memasukkan peran IAM ke dalam tabel pemetaan untuk memetakan peran IAM logis dengan peran IAM fisik.

Peran IAM untuk pengguna SAP harus memungkinkan `sts:assumeRole` tindakan untuk prinsipal tepercaya. Prinsipal tepercaya dapat bervariasi berdasarkan bagaimana sistem SAP diautentikasi. AWS Untuk detail selengkapnya, lihat Menentukan prinsipal.

Berikut ini adalah beberapa contoh skenario SAP yang paling umum.

- Sistem SAP berjalan di Amazon EC2 dengan profil instans yang ditetapkan - di sini, profil instans Amazon EC2 dilampirkan ke peran IAM.

  JSON

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

- Sistem SAP berjalan di Amazon EC2 tanpa profil instans — di sini, Amazon EC2 mengambil peran untuk pengguna SAP.

JSON

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

- Sistem SAP yang berjalan di tempat — Sistem SAP yang berjalan di tempat hanya dapat mengautentikasi menggunakan Kunci Akses Rahasia. Untuk informasi selengkapnya, lihat otentikasi sistem SAP aktif. AWS

  Di sini, setiap peran IAM yang diasumsikan oleh pengguna SAP harus memiliki hubungan kepercayaan yang mempercayai pengguna SAP.

  JSON

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

Pertimbangan Keamanan Profil Sumber

Saat menggunakan profil sumber:

Manajemen Peran IAM

Kritis: Peran IAM dalam rantai profil sumber harus dikelola secara ketat untuk mencegah akses yang tidak sah dan eskalasi hak istimewa:

- Menerapkan prinsip hak istimewa terkecil - Berikan hanya izin minimum yang diperlukan untuk tujuan spesifik setiap peran
- Mengaudit izin peran secara teratur - Tinjau dan perbarui kebijakan peran setiap triwulan atau saat persyaratan berubah
- Memantau penggunaan peran - Gunakan untuk melacak panggilan AssumeRole API dan mengidentifikasi pola yang tidak biasa
- Batasi hubungan kepercayaan - Batasi prinsip mana yang dapat mengambil peran masing-masing hanya kepada mereka yang benar-benar membutuhkan akses
- Gunakan kondisi dalam kebijakan kepercayaan - Tambahkan kondisi seperti IP sumber, persyaratan MFA, atau batasan berbasis waktu jika sesuai
- Tujuan peran dokumen - Menjaga dokumentasi yang jelas tentang kasus penggunaan yang dimaksudkan setiap peran dan izin yang diperlukan

Otorisasi dan Kontrol Akses

- Pastikan semua profil perantara dalam rantai memiliki kebijakan kepercayaan yang sesuai yang dikonfigurasi
- Pengguna harus memiliki `/AWS1/SESS` otorisasi untuk SEMUA profil dalam rantai, termasuk profil perantara
- Setiap peran IAM harus secara eksplisit mempercayai peran sebelumnya dalam rantai

Pengamanan Teknis

- SDK memberlakukan kedalaman rantai maksimum 5 profil untuk mencegah panggilan STS API yang berlebihan
- Referensi melingkar secara otomatis terdeteksi dan dicegah

- Metode otentikasi profil dasar divalidasi untuk memastikannya menggunakan metode standar (INST, SSF, atau RLA)

Untuk informasi selengkapnya tentang mengonfigurasi profil sumber, lihat Menggunakan Profil Sumber untuk Akses Lintas Akun.

Otorisasi SAP

Otorisasi yang diperlukan untuk mengonfigurasi SDK bergantung pada edisi SDK.

Topik

- Otorisasi untuk konfigurasi
- Otorisasi SAP untuk pengguna akhir

Otorisasi untuk konfigurasi

Lihat tab berikut untuk lebih jelasnya.

SDK for SAP ABAP

Otorisasi berikut diperlukan untuk mengkonfigurasi SDK untuk SAP ABAP.

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

    Pilih dari grup otorisasi berikut.

    - /AWS1/CFG- AWS SDK untuk SAP ABAP v1 - Config
    - /AWS1/MOD- AWS SDK untuk SAP ABAP v1 - Runtime
    - /AWS1/PFL- AWS SDK untuk SAP ABAP v1 - Profil SDK
    - /AWS1/RES- AWS SDK untuk SAP ABAP v1 - Sumber Logis
    - /AWS1/TRC- AWS SDK untuk SAP ABAP v1 - Jejak

SDK for SAP ABAP - BTP edition

Gunakan langkah-langkah berikut untuk memungkinkan SDK untuk SAP ABAP - BTP edisi akses ke konfigurasi.

1. Buat peran bisnis baru menggunakan template peran `SAP_BR_BPC_EXPERT` bisnis. Template ini menyediakan akses ke aplikasi Konfigurasi Bisnis Cutsom.
2. Di bawah Rincian Peran Umum, buka Access Categories, dan pilih Unrestricted for Read, Write, Value Help.
3. Buka tab Katalog Bisnis, dan tetapkan katalog `/AWS1/RTBTP_BCAT` bisnis untuk menyediakan akses ke konfigurasi SDK.
4. Buka tab Pengguna Bisnis, dan tetapkan pengguna bisnis untuk memberikan akses ke konfigurasi SDK.

Otorisasi SAP untuk pengguna akhir

Prasyarat: Tentukan Profil SDK

Sebelum administrator keamanan SAP dapat menentukan peran mereka, Business Analyst akan menentukan profil SDK dalam transaksi `/AWS1/IMG` untuk AWS SDK untuk SAP ABAP atau aplikasi Custom Business Configuration untuk SDK untuk SAP ABAP - edisi BTP. Biasanya, profil SDK akan diberi nama sesuai dengan fungsi bisnisnya: ZFINANCE, ZBILLING, ZMFG, ZPAYROLL, dll. Untuk setiap profil SDK, Analis Bisnis akan menentukan peran IAM logis dengan nama pendek, seperti CFO, AUDITOR, PELAPORAN. Ini akan dipetakan ke peran IAM nyata oleh administrator keamanan IAM.

Tentukan PFCG atau Peran Bisnis

> **Note**
>
> Peran PFCG disebut Peran Bisnis di SAP BTP, lingkungan ABAP.

Administrator keamanan SAP kemudian akan menambahkan objek otorisasi `/AWS1/SESS` untuk memberikan akses ke profil SDK.

Objek Auth `/AWS1/SESS`

- Bidang `/AWS1/PROF` = `ZFINANCE`

Pengguna juga harus dipetakan ke peran IAM logis untuk setiap profil SDK, tergantung pada fungsi pekerjaan mereka. Misalnya, auditor keuangan dengan akses pelaporan mungkin diberi wewenang untuk peran IAM logis yang disebut. AUDITOR

Objek Auth /AWS1/LROL

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

Sementara itu, CFO, dengan otorisasi baca/tulis, mungkin memiliki peran PFCG yang mengotorisasi mereka peran logis. CFO

Objek Auth /AWS1/LROL

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

Secara umum, pengguna harus diberi wewenang hanya untuk satu peran IAM logis per profil SDK. Jika pengguna diberi wewenang untuk lebih dari satu peran IAM (misalnya, jika CFO diotorisasi untuk keduanya CFO dan peran IAM AUDITOR logis), maka AWS SDK memutuskan ikatan dengan memastikan bahwa peran prioritas yang lebih tinggi (nomor urutan yang lebih rendah) berlaku.

Seperti semua skenario keamanan, pengguna harus diberi hak istimewa paling sedikit untuk menjalankan fungsi pekerjaan mereka. Strategi sederhana untuk mengelola peran PFCG adalah memberi nama peran PFCG Tunggal sesuai dengan profil SDK dan peran logis yang mereka otorisasi. Misalnya, peran Z_AWS_PFL_ZFINANCE_CFO memberikan akses ke profil ZFINANCE dan peran IAM logis. CFO Peran tunggal ini kemudian dapat ditugaskan ke peran komposit yang mendefinisikan fungsi pekerjaan. Setiap perusahaan memiliki strategi mereka sendiri untuk manajemen peran, dan kami mendorong Anda untuk menentukan strategi PFCG yang sesuai untuk Anda.

Operasi yang aman

Enkripsi Data Saat Istirahat

AWS Kunci Akses Rahasia digunakan untuk mengautentikasi SDK. Mereka dienkripsi menggunakan fungsi SSF atau Credential Store oleh SAP.

Enkripsi Data Dalam Transit

Semua panggilan ke Layanan AWS dienkripsi dengan HTTPS. SAP ICM mengelola koneksi HTTPS. AWS sertifikat harus dipercaya dalam STRUST.

Penggunaan API

Ketika pengguna ABAP mengasumsikan peran menggunakan`sts:assumeRole`, nama sesi diberi judul`USERID-SID-MANDT`, di mana:

- `USERID`adalah pengguna ABAP dari `SY-UNAME` variabel.
- `SID`adalah ID sistem ABAP dari `SY-SYSID` variabel.
- `MANDT`adalah klien ABAP dari `SY-MANDT` variabel.

Nama sesi muncul CloudTrail sebagai nama pengguna. Ini memastikan bahwa panggilan API dari pengguna ABAP dapat ditelusuri kembali ke sistem, klien, dan pengguna yang memulai panggilan. Untuk informasi selengkapnya, lihat Apa itu AWS CloudTrail?

Menggunakan Otentikasi Kunci Akses Rahasia dengan Enkripsi SSF

Sistem SAP lokal (atau sistem yang berjalan di cloud lain) dapat diautentikasi AWS dengan menggunakan otentikasi kunci akses rahasia dengan. AWS Identity and Access ManagementSecure Store and Forward Mechanism (SSF) SAP digunakan untuk mengenkripsi dan menyimpan AWS kredensil dengan aman (Access Key ID dan Secret Access Key) dari pengguna IAM. Sistem SAP masuk AWS menggunakan pengguna IAM, lihat Mengelola Kunci Akses untuk Pengguna IAM untuk informasi.

Prasyarat

Prasyarat berikut harus dipenuhi sebelum memulai konfigurasi:

- Peran IAM untuk pengguna SAP harus dibuat oleh administrator IAM. Peran harus memiliki izin untuk memanggil yang diperlukan Layanan AWS. Untuk informasi selengkapnya, lihat Praktik terbaik untuk Keamanan IAM.
- Buat otorisasi untuk menjalankan `/AWS1/IMG` transaksi. Untuk informasi selengkapnya, lihat Otorisasi untuk konfigurasi.

Prosedur

Ikuti petunjuk ini untuk mengonfigurasi penyimpanan kredensi terenkripsi SSF:

Langkah-langkah

- Langkah 1 - Tentukan aplikasi SSF untuk Credential Storage
- Langkah 2 - Mengatur parameter enkripsi untuk aplikasi SSF
- Langkah 3 - Buat PSE untuk Aplikasi SSF
- Langkah 4 — Tetapkan aplikasi SSF ke AWS SDK untuk SAP ABAP
- Langkah 5 - Konfigurasikan profil SDK untuk menggunakan kredensil terenkripsi SSF

Langkah 1 - Tentukan aplikasi SSF untuk Credential Storage

1. Jalankan kode transaksi `SE16` untuk menentukan aplikasi SSF.
2. Masukkan nama `SSFAPPLIC` tabel, dan pilih Entri Baru.
3. Masukkan detail berikut:
   - APPLIC: `ZAWS1` (nama untuk aplikasi SSF).
   - DESKRIPSI: `SSF Encryption for the AWS SDK untuk SAP ABAP` (deskripsi).
   - Pilih `Selected(X)` opsi untuk bidang yang tersisa.
4. Pilih Simpan.

Langkah 2 - Mengatur parameter enkripsi untuk aplikasi SSF

1. Jalankan kode transaksi `/n/AWS1/IMG` untuk meluncurkan Panduan Implementasi (IMG) untuk AWS SDK untuk SAP ABAP.
2. Perluas AWS SDK untuk SAP ABAP Setelan simpul IMG > Prasyarat Teknis > Pengaturan Tambahan untuk sistem Lokal.
3. Jalankan aktivitas IMG Set SSF Parameters.
4. Pilih Entri Baru, dan pilih aplikasi SSF yang dibuat pada langkah sebelumnya. Pilih Simpan.
5. Ubah algoritma hash ke SHA256(atau lebih tinggi), dan algoritma enkripsi ke AES256-CBC. Pertahankan pengaturan lain sebagai default, dan pilih Simpan.

Pengaturan enkripsi ini akan digunakan untuk mengenkripsi AWS kredensil dengan aman.

Langkah 3 - Buat PSE untuk Aplikasi SSF

1. Jalankan `/n/AWS1/IMG` transaksi, lalu pilih AWS SDK untuk SAP ABAP Setelan > Prasyarat Teknis > Pengaturan Tambahan untuk sistem Lokal.
2. Jalankan aktivitas IMG Buat PSE untuk Aplikasi SSF, yang akan mengarahkan Anda ke aplikasi. `STRUST transaction` Pilih Edit.
3. Pilih kanan aplikasi SSF yang dibuat dithe section called “Langkah 1”, dan pilih Buat. Pertahankan semua pengaturan default lainnya, dan pilih Lanjutkan. Pastikan Anda memilih `RSA` dan bukan DSA sebagai algoritma.

Langkah 4 — Tetapkan aplikasi SSF ke AWS SDK untuk SAP ABAP

1. Jalankan `/n/AWS1/IMG` transaksi, lalu pilih AWS SDK untuk SAP ABAP Setelan > Prasyarat Teknis > Pengaturan Tambahan untuk sistem Lokal.
2. Jalankan aktivitas IMG Tetapkan aplikasi SSF ke file. AWS SDK untuk SAP ABAP
3. Pilih Entri Baru dan masukkan aplikasi SSF yang dibuat di. the section called “Langkah 1” Pilih Simpan.

Langkah 5 - Konfigurasikan profil SDK untuk menggunakan kredensil terenkripsi SSF

1. Jalankan `/n/AWS1/IMG` transaksi, dan pilih AWS SDK untuk SAP ABAP Settings > Application Configurations.
2. Jalankan Profil SDK aktivitas IMG.
3. Pilih Entri Baru. Masukkan nama profil dan deskripsi. Pilih Simpan.
4. Sorot entri yang Anda buat dan klik cabang pohon Otentikasi Dan Pengaturan.
5. Pilih Entri Baru dan masukkan detail berikut:
   - SID: ID sistem dari sistem SAP.
   - Klien: Klien dari sistem SAP.
   - ID Skenario: Pilih `DEFAULT` skenario yang dibuat oleh administrator Basis Anda.
   - AWS Wilayah: AWS Wilayah tempat Anda ingin melakukan panggilan ke.
   - Metode Otentikasi: Pilih **`Credentials from SSF Storage`** dari dropdown dan pilih Simpan. Pilih Set Credentials dan masukkan Access Key ID dan Secret Access Key dari pengguna IAM.
   - Nonaktifkan peran IAM: Simpan ini sebagai default yaitu tidak dicentang.

- Pilih Simpan.

6. Klik pada cabang pohon Pemetaan Peran IAM. Pilih Entri Baru. Masukkan nomor urut, nama untuk peran IAM logis dan ARN Peran IAM yang disediakan oleh Administrator IAM. AWS Pilih Simpan.

Untuk informasi selengkapnya, lihat Konfigurasi aplikasi.

Menggunakan sertifikat dengan Peran IAM Di Mana Saja

Sistem SAP dapat diautentikasi AWS dengan menggunakan otentikasi berbasis sertifikat dengan Roles Anywhere. AWS Identity and Access Management Anda harus menyiapkan sertifikat`STRUST`, dan mengonfigurasi profil SDK di`/AWS1/IMG`.

Prasyarat

Prasyarat berikut harus dipenuhi sebelum memulai pengaturan untuk sertifikasi.

- X.509 Sertifikat yang dikeluarkan oleh otoritas sertifikat (CA) Anda harus memenuhi persyaratan berikut.
  - Sertifikat penandatanganan harus berupa sertifikat v3.
  - Rantai tidak boleh melebihi 5 sertifikat.
  - Sertifikat harus mendukung algoritma RSA atau ECDSA.
- Daftarkan CA Anda dengan IAM Roles Anywhere sebagai jangkar kepercayaan, dan buat profil untuk menentukan Peran IAM Anywhere. roles/policies Untuk informasi selengkapnya, lihat Membuat jangkar kepercayaan dan profil di AWS Identity and Access Management Peran Di Mana Saja.
- Peran IAM untuk pengguna SAP harus dibuat oleh administrator IAM. Peran harus memiliki izin untuk memanggil yang diperlukanLayanan AWS. Untuk informasi selengkapnya, lihat Praktik terbaik untuk Keamanan IAM.
- Buat otorisasi untuk menjalankan `/AWS1/IMG` transaksi. Untuk informasi selengkapnya, lihat Otorisasi untuk konfigurasi.

Prosedur

Ikuti petunjuk ini untuk menyiapkan otentikasi berbasis sertifikat.

Langkah-langkah

- Langkah 1 - Tentukan aplikasi SSF dengan menggunakan SAP Secure Store and Forward (SSF)
- Langkah 2 - Tetapkan parameter SSF
- Langkah 3 - Buat PSE dan permintaan sertifikat
- Langkah 4 - Impor respon sertifikat ke PSE yang relevan
- Langkah 5 - Mengonfigurasi profil SDK untuk menggunakan Peran IAM Di Mana Saja

Langkah 1 - Tentukan aplikasi SSF dengan menggunakan SAP Secure Store and Forward (SSF)

1. Jalankan kode transaksi `SE16` untuk menentukan aplikasi SSF.
2. Masukkan nama `SSFAPPLIC` tabel, dan pilih Entri Baru.
3. Masukkan nama untuk aplikasi SSF di `APPLIC` file, deskripsi di `DESCRIPT` file, dan pilih `Selected (X)` opsi untuk bidang yang tersisa.

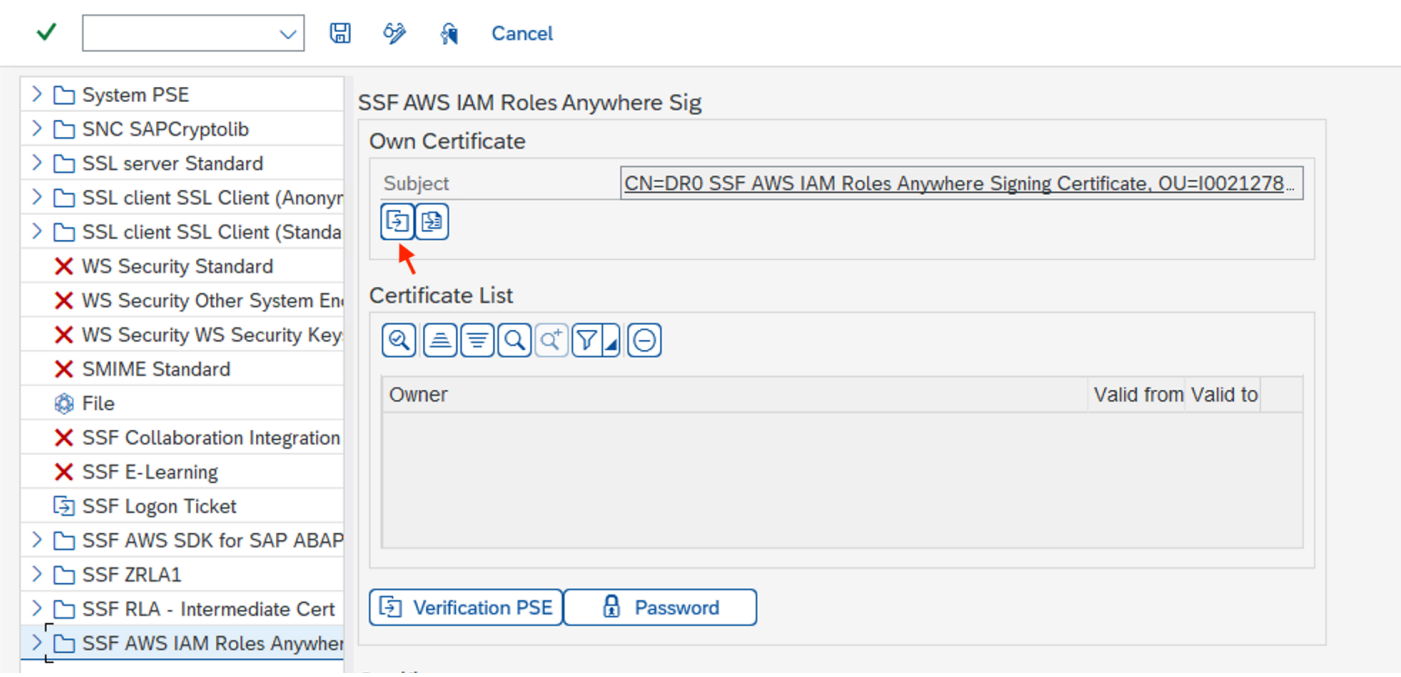
Langkah 2 - Tetapkan parameter SSF

1. Jalankan `/n/AWS1/IMG` to launch AWS SDK untuk SAP ABAP Implementation Guide (IMG).
2. Pilih AWS SDK untuk SAP ABAPPengaturan > Prasyarat Teknis > Pengaturan Tambahan untuk Sistem. On-Premises
3. Jalankan aktivitas IMG Set SSF Parameters.
4. Pilih Entri Baru, dan pilih aplikasi SSF yang dibuat pada langkah sebelumnya. Pilih Simpan.
5. Ubah algoritma hash ke SHA256, dan algoritma enkripsi ke. AES256-CBC Pertahankan pengaturan lain sebagai default, dan pilih Simpan.

Langkah 3 - Buat PSE dan permintaan sertifikat

1. Jalankan `/n/AWS1/IMG` transaksi, dan pilih AWS SDK untuk SAP ABAPPengaturan > Prasyarat Teknis > Pengaturan Tambahan untuk sistem. On-Premises
2. Jalankan aktivitas `Create PSE for SSF Application` IMG.
3. Pilih Edit untuk `STRUST` transaksi.
4. Right-select aplikasi SSF yang dibuat dithe section called “Langkah 1”, dan pilih Buat. Pertahankan semua pengaturan default lainnya, dan pilih Lanjutkan.

5. Pilih Buat Permintaan Sertifikat. Lihat gambar berikut. Pertahankan opsi default, dan pilih Lanjutkan. Salin atau ekspor permintaan sertifikat yang dihasilkan, dan berikan ke CA Anda. CA Anda memverifikasi permintaan, dan merespons dengan sertifikat kunci publik yang ditandatangani.



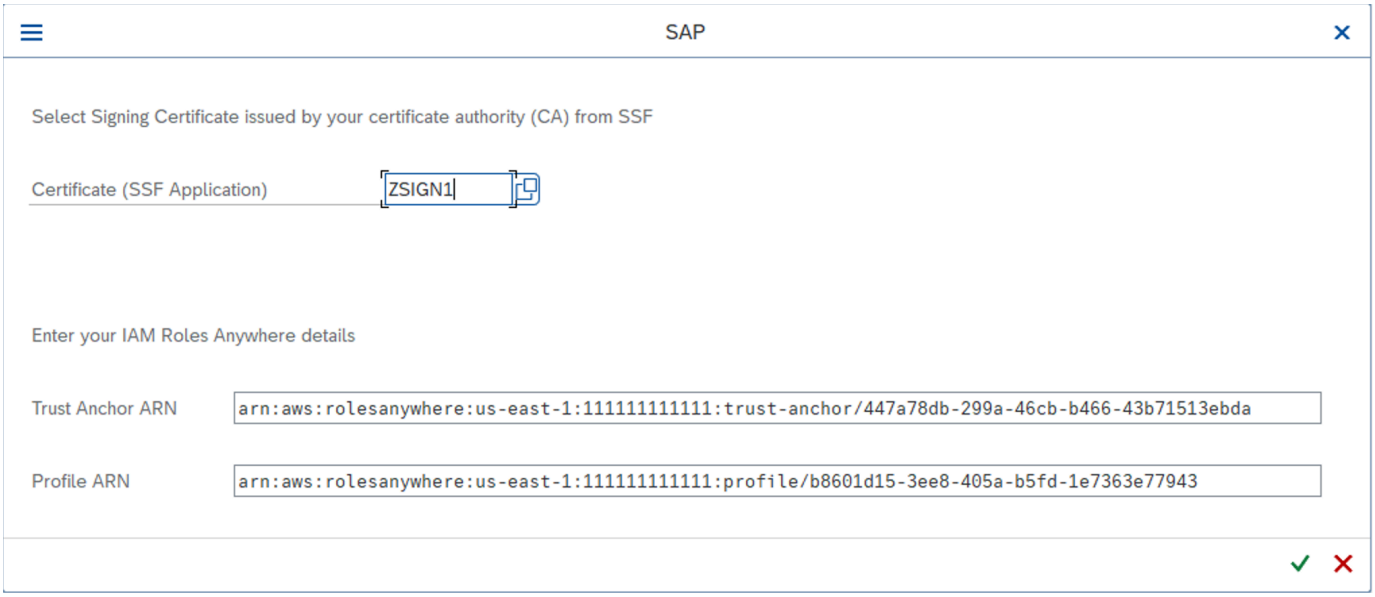
Proses penandatanganan bervariasi berdasarkan CA Anda, dan teknologi yang digunakan oleh mereka. Lihat Menerbitkan sertifikat entitas akhir pribadi dengan AWS Private Certificate Authority untuk contoh.

Langkah 4 - Impor respon sertifikat ke PSE yang relevan

1. Jalankan `/n/AWS1/IMG` transaksi, dan pilih AWS SDK untuk SAP ABAPPengaturan > Prasyarat Teknis > Pengaturan Tambahan untuk sistem. On-Premises
2. Jalankan aktivitas `Create PSE for SSF Application` IMG.
3. Pilih Edit untuk `STRUST` transaksi.
4. Pilih aplikasi SSF, lalu pilih Import Certificate Response terletak di bagian PSE di bawah subjek. Salin dan tempel respons sertifikat ke dalam kotak teks atau impor file dari sistem file. Pilih Lanjutkan > Simpan.
5. Rincian sertifikat dapat dilihat dengan memilih subjek dua kali. Informasi ditampilkan di bagian sertifikat.

Langkah 5 - Mengonfigurasi profil SDK untuk menggunakan Peran IAM Di Mana Saja

1. Jalankan `/n/AWS1/IMG` transaksi, lalu pilih AWS SDK untuk SAP ABAPSettings > Application Configurations.
2. Buat profil SDK baru, dan beri nama.
3. Pilih IAM Roles Anywhere sebagai metode otentikasi.
   - Di panel kiri, pilih Otentikasi dan Pengaturan.
   - Buat entri baru, dan masukkan informasi untuk sistem SAP Anda, danWilayah AWS.
   - Pilih IAM Roles Anywhere untuk metode otentikasi, dan pilih Simpan.
   - Pilih Masukkan Detail, dan di jendela pop-up, pilih aplikasi SSF yang dibuat di. the section called "Langkah 1" Masukkan ARN Trust Anchor, dan Profil ARN yang dibuat di. the section called "Prasyarat" Lihat gambar berikut. Pilih Lanjutkan.



4. Di panel kiri, pilih Pemetaan Peran IAM. Masukkan nama, dan berikan ARN peran IAM yang disediakan oleh administrator IAM Anda.

Untuk informasi selengkapnya, lihat Konfigurasi aplikasi.

Menggunakan Profil Sumber untuk Akses Lintas Akun

Profil sumber memungkinkan sistem SAP untuk mengakses AWS sumber daya di beberapa akun dengan merantai asumsi peran IAM. Satu profil mengasumsikan peran, yang kemudian mengasumsikan peran lain, dan seterusnya, mirip dengan `source_profile` parameter di CLI

AWS . Ini berguna untuk skenario akses lintas akun di mana Anda perlu melintasi beberapa AWS akun untuk mencapai sumber daya target Anda.

Contoh: Sistem SAP Anda berjalan di Akun A (11111111111111) dan perlu mengakses bucket Amazon S3 di Akun C (333333333333). Anda mengkonfigurasi tiga profil:

1. `DEV_BASE`mendapatkan kredensi dasar dari metadata instans Amazon EC2 dan mengasumsikan Peran P di Akun A
2. `SHARED_SERVICES`menggunakan `DEV_BASE` kredensil untuk mengambil Peran Q di Akun B (222222222222)
3. `PROD_S3_ACCESS`menggunakan `SHARED_SERVICES` kredensil untuk mengambil Peran R di Akun C

Saat aplikasi Anda menggunakan`PROD_S3_ACCESS`, SDK secara otomatis mengeksekusi chain: get credentials from instance metadata → asumsikan Role P → asumsikan Role Q → asumsikan Role R.

Prasyarat

Prasyarat berikut harus dipenuhi sebelum mengonfigurasi profil sumber:

- Peran IAM untuk setiap langkah dalam rantai harus dibuat oleh administrator IAM. Setiap peran harus memiliki:
  - Izin untuk memanggil yang diperlukan Layanan AWS
  - Hubungan kepercayaan dikonfigurasi untuk memungkinkan peran sebelumnya dalam rantai untuk mengasumsikannya

  Untuk informasi selengkapnya, lihat Praktik terbaik untuk Keamanan IAM.
- Buat otorisasi untuk menjalankan `/AWS1/IMG` transaksi. Untuk informasi selengkapnya, lihat Otorisasi untuk konfigurasi.
- Pengguna harus memiliki `/AWS1/SESS` otorisasi untuk SEMUA profil dalam rantai, termasuk profil perantara.

Prosedur

Ikuti petunjuk ini untuk mengonfigurasi profil sumber.

Langkah-langkah

- Langkah 1 - Konfigurasikan profil dasar
- Langkah 2 - Konfigurasikan profil berantai

Langkah 1 - Konfigurasikan profil dasar

Profil dasar adalah profil pertama dalam rantai dan harus menggunakan metode otentikasi standar.

1. Jalankan `/n/AWS1/IMG` transaksi untuk meluncurkan AWS SDK untuk SAP ABAP Implementation Guide (IMG).
2. Pilih AWS SDK untuk Pengaturan SAP ABAP > Konfigurasi Aplikasi > Profil SDK.
3. Buat profil baru untuk digunakan sebagai profil dasar Anda dengan memilih Entri Baru dan masukkan nama profil dan deskripsi. Pilih Simpan.

   > **Note**
   >
   > Jika Anda menggunakan profil yang sudah ada yang sudah dikonfigurasi dengan metode otentikasi standar (INST, SSF, atau RLA), Anda dapat melewati langkah-langkah yang tersisa di bagian ini dan melanjutkan langsung ke. the section called “Langkah 2”

4. Pilih profil yang Anda buat, lalu pilih Otentikasi dan Pengaturan > Entri Baru, dan masukkan detail berikut:
   - SID: ID sistem sistem SAP
   - Klien: Klien dari sistem SAP
   - ID Skenario: Pilih `DEFAULT` skenario yang dibuat oleh administrator Basis
   - AWS Wilayah: AWS Wilayah tempat Anda ingin melakukan panggilan
   - Metode Otentikasi: Pilih salah satu dari berikut ini:
     - Peran Instans melalui Metadata untuk sistem SAP yang berjalan di Amazon EC2
     - Kredensil dari SSF Storage untuk sistem cloud lokal atau lainnya
     - Peran IAM Di Mana Saja untuk otentikasi berbasis sertifikat

   Pilih Simpan.
5. Pilih Pemetaan Peran IAM > Entri Baru, dan masukkan:
   - Nomor urut: 1
   - Peran IAM Logis: Nama deskriptif (mis.,) `DEV_BASE_ROLE`

- Peran IAM ARN: ARN peran IAM di akun pertama (mis.,) `arn:aws:iam::111111111111:role/DevBaseRole`

Pilih Simpan.

Langkah 2 - Konfigurasikan profil berantai

Konfigurasikan setiap profil perantara dan akhir dalam rantai.

Untuk **`SHARED_SERVICES`** profil (rantai dari**`DEV_BASE`**):

1. Jalankan `/n/AWS1/IMG` transaksi.
2. Pilih AWS SDK untuk Pengaturan SAP ABAP > Konfigurasi Aplikasi > Profil SDK.
3. Pilih Entri Baru. Masukkan nama profil (mis.,`SHARED_SERVICES`) dan deskripsi. Pilih Simpan.
4. Pilih profil yang Anda buat, lalu pilih Otentikasi dan Pengaturan > Entri Baru dan masukkan detail berikut:
   - SID: ID sistem sistem SAP
   - Klien: Klien dari sistem SAP
   - ID Skenario: Pilih `DEFAULT` skenario yang dibuat oleh administrator Basis
   - AWS Wilayah: AWS Wilayah tempat Anda ingin melakukan panggilan
   - Metode Otentikasi: Pilih Profil Sumber dari dropdown
   - ID Profil Sumber: Masukkan ID profil profil dasar (mis.,`DEV_BASE`)

   Pilih Simpan.
5. Pilih Pemetaan Peran IAM > Entri Baru, dan masukkan:
   - Nomor urut: 1
   - Peran IAM Logis: Nama deskriptif (mis.,) `SHARED_ROLE`
   - Peran IAM ARN: `arn:aws:iam::222222222222:role/SharedServicesRole`

   Pilih Simpan.

Untuk **`PROD_S3_ACCESS`** profil (rantai dari**`SHARED_SERVICES`**):

Ulangi langkah yang sama sepertiSHARED_SERVICES, tetapi:

- Gunakan `PROD_S3_ACCESS` sebagai nama

- Setel ID Profil Sumber ke `SHARED_SERVICES`
- Gunakan `PROD_S3_ROLE` dan `arn:aws:iam::333333333333:role/ProdS3AccessRole` dalam Pemetaan Peran IAM

Untuk praktik terbaik keamanan termasuk manajemen peran IAM, konfigurasi kebijakan kepercayaan, dan persyaratan otorisasi, lihat Praktik terbaik untuk Keamanan IAM.

Menggunakan SAP Credential Store

SAP Credential Store digunakan di SAP Business Technology Platform untuk menyimpan kredensi dengan aman untuk otentikasi kunci akses rahasia. AWS Anda harus berlangganan untuk menggunakan layanan ini.

Petunjuk berikut mengasumsikan bahwa Anda telah mengonfigurasi profil SDK. Untuk informasi selengkapnya, lihat Mengkonfigurasi. AWS SDK untuk SAP ABAP

Sebelum memulai konfigurasi, pastikan Anda memenuhi prasyarat. Untuk informasi selengkapnya, lihat SAP Credential Store.

Topik

- Langkah konfigurasi
- Menggunakan SAP Credential Store dengan SDK

Langkah konfigurasi

Langkah-langkah

- Langkah 1: Konfigurasikan pengaturan untuk otentikasi
- Langkah 2: Buat kunci layanan
- Langkah 3: Ubah kunci layanan menjadi .p12 format
- Langkah 4: Connect ke SAP BTP, lingkungan ABAP

Langkah 1: Konfigurasikan pengaturan untuk otentikasi

Gunakan langkah-langkah berikut untuk mengonfigurasi pengaturan Credential Store untuk otentikasi.

1. Arahkan ke tab Pengaturan pada instance SAP Credential Store.
2. Pilih Edit Konfigurasi:
   - Pilih Mutual TLS sebagai Jenis Otentikasi Default.
   - Pilih Dinonaktifkan untuk Status Enkripsi Muatan. Payload dienkripsi dalam perjalanan dengan HTTPS. Namun, payload saat ini tidak dapat dienkripsi ganda.
3. Pilih Simpan.

Langkah 2: Buat kunci layanan

Gunakan langkah-langkah berikut untuk membuat kunci layanan untuk Credential Store.

1. Di panel kiri aplikasi SAP Credential Store, navigasikan ke Service Keys.
2. Pilih Buat Kunci Layanan.
3. Masukkan nama untuk kunci layanan, dan pilih Buat.

Kunci layanan dibuat berdasarkan jenis otentikasi yang dipilih. Unduh kunci layanan, dan jaga agar tetap aman untuk penggunaan nanti.

Langkah 3: Ubah kunci layanan menjadi `.p12` format

Sertifikat klien dalam `.p12` format diperlukan untuk membuat pengguna keluar untuk sistem komunikasi. Gunakan langkah-langkah berikut untuk menghasilkan sertifikat dari detail `.p12` sertifikat yang disediakan di kunci Layanan Toko Kredenal.

1. Unduh sertifikat SAP Cloud Root CA (diperlukan oleh SAP) dari SAP Trust Center Services.
2. Buka sertifikat SAP Cloud Root CA dalam format file teks apa pun. Di akhir file, tekan Enter, dan salin-tempel sertifikat dari bidang sertifikat kunci layanan. Ganti karakter baris baru `\n` dengan baris baru yang sebenarnya (Enter), dan simpan seluruh sertifikat dalam format `.cer` file.
3. Salin kunci dari bidang kunci kunci layanan. Kunci pribadi ini harus diperlakukan sebagai data sensitif. Tempel dalam file teks, dan ganti karakter baris baru `\n` dengan baris baru yang sebenarnya (Enter). Simpan kunci pribadi dalam file teks.
4. Dengan sertifikat dan kunci pribadi yang dihasilkan pada langkah sebelumnya, jalankan perintah berikut untuk menghasilkan `.p12` sertifikat.

```
openssl pkcs12 -export -out <.p12_filename> -inkey <private_key.key> -in
 <certificate.cer>
```

Perintah tersebut memerlukan verifikasi kata sandi ekspor. Simpan kata sandi untuk penggunaan lebih lanjut.

Hapus file `.key` teks yang disimpan di kunci pribadi Anda.

Langkah 4: Connect ke SAP BTP, lingkungan ABAP

Konfigurasikan SAP BTP, lingkungan ABAP untuk terhubung dengan SAP Credential Store.

Topik

- Sistem komunikasi
- Pengaturan komunikasi

Sistem komunikasi

Gunakan langkah-langkah berikut untuk membuat sistem komunikasi yang memungkinkan komunikasi dari SAP BTP, lingkungan ABAP ke SAP Credential Store.

1. Buka landasan peluncuran Fiori dari sistem lingkungan ABAP.
2. Pilih ubin Sistem Komunikasi untuk membuka aplikasi.
3. Pilih Baru.
4. Masukkan nama dan ID untuk sistem komunikasi, dan pilih Buat. Misalnya, Anda dapat memberi nama sistem`ZSAP_CREDSTORE`.
5. Masukkan informasi lain yang diperlukan:
   - Nama host: Salin nama host dari URL Kunci Layanan. Misalnya, jika URL-nya`https://credstore.mesh.cf.us10.hana.ondemand.com/api/v1/credentials`, maka nama host adalah`credstore.mesh.cf.us10.hana.ondemand.com`.
   - Pengguna untuk Komunikasi Keluar: Pilih + untuk menambahkan pengguna baru.
     a. Pilih SSL Client Certificate sebagai mekanisme otentikasi.
     b. Pilih Unggah Sertifikat Baru:
        - Jelajahi `.p12` sertifikat yang dihasilkan pada langkah sebelumnya.
        - Masukkan deskripsi.
        - Masukkan kata sandi ekspor yang digunakan untuk menghasilkan `.p12` sertifikat.
        - Pilih Unggah.

c. Pilih Buat untuk membuat pengguna keluar.

6. Pilih Simpan.
7. Hapus kunci layanan yang diunduh pada langkah sebelumnya.

Pengaturan komunikasi

Gunakan langkah-langkah berikut untuk membuat pengaturan komunikasi untuk menyediakan skenario komunikasi untuk komunikasi keluar.

1. Buka landasan peluncuran Fiori dari sistem lingkungan ABAP.
2. Pilih ubin Pengaturan Komunikasi untuk membuka aplikasi.
3. Pilih Baru.
4. Pilih skenario komunikasi`/AWS1/CRED_COMM_SCENARIO`, dan masukkan nama untuk pengaturan komunikasi. Misalnya, `Z_AWS_SDK_TO_SAP_CREDSTORE`.
5. Pilih Buat.
6. Di bidang Sistem Komunikasi, telusuri Sistem Komunikasi yang dibuat pada langkah sebelumnya. Informasi lainnya adalah pemilihan pos yang diisi secara otomatis dari sistem.
7. Pilih Simpan.
8. Pilih Periksa Koneksi untuk menguji koneksi Anda.

Setelah pengaturan ini selesai, lingkungan ABAP dapat menggunakan pengaturan komunikasi untuk menggunakan layanan SAP Credential Store melalui layanan keluar (HTTP).

Menggunakan SAP Credential Store dengan SDK

Langkah-langkah

- Langkah 1: Buat namespace dan kredensyal
- Langkah 2: Konfigurasikan aplikasi Konfigurasi Bisnis Kustom

Langkah 1: Buat namespace dan kredensyal

Buat namespace dan kredensi di SAP Credential Store dengan SAP Help — Create, Edit, dan Delete a Credential.

Masukkan detail berikut untuk membuat kredensi tipe Key.

- Namespace — Masukkan nama untuk namespace, dan kelompokkan kredenal terkait bersama-sama.
- Nama — Masukkan nama untuk kunci. Kami merekomendasikan`aws-0123456789012-username`, di mana:
  - `0123456789012`adalah Akun AWS ID yang kredensialnya memberikan akses
  - `username`adalah nama pengguna IAM yang menjadi milik kredensialnya
- Nilai - Masukkan kunci akses rahasia yang dikodekan basis-64. Gunakan perintah berikut untuk basis-64 menyandikan kunci akses rahasia Anda.

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

  Perintah membaca kunci akses rahasia dari input standar, dan meneruskannya ke base64 tanpa baris baru yang tertinggal. Ini mengeluarkan kunci akses rahasia yang dikodekan basis-64 ke layar. Hapus atau tutup terminal Anda setelah menyalin nilainya ke SAP Credential Store.
- Nama pengguna — Masukkan ID kunci akses Anda.

- Pilih Buat.

Namespace baru dengan satu kredensi dibuat, dan kredenal dapat ditambahkan, dihapus, atau dimodifikasi dalam namespace ini.

Ikuti prinsip hak istimewa terkecil untuk mengelola akses ke kredenal yang disimpan di namespace.

Langkah 2: Konfigurasikan aplikasi Konfigurasi Bisnis Kustom

Gunakan langkah-langkah berikut untuk mengonfigurasi aplikasi Konfigurasi Bisnis Kustom untuk menentukan kredensi yang akan digunakan untuk autentikasi oleh SDK.

1. Buka landasan peluncuran Fiori dari sistem lingkungan ABAP.
2. Jelajahi ubin Konfigurasi Bisnis Kustom untuk membuka aplikasi.
3. Buka Konfigurasi Bisnis Profil SDK.
4. Pilih profil SDK yang pengaturan autentikasi harus dikonfigurasi untuk SAP Credential Store.
5. Di tab Otentikasi dan Pengaturan untuk profil yang dipilih, pilih Edit, dan masukkan detail berikut:
   - Metode Otentikasi - Pilih Kredensial dari SAP Credential Store.

- Namespace — Masukkan namespace yang dibuat di SAP Credential Store. Untuk informasi selengkapnya, lihat the section called “Langkah 1: Buat namespace dan kredensyal”.
- Nama Kunci — Masukkan nama kredensyal kunci yang dibuat. Untuk informasi selengkapnya, lihat the section called “Langkah 1: Buat namespace dan kredensyal”.
- Pengaturan Komunikasi — Masukkan nama pengaturan komunikasi yang dibuat. Untuk informasi selengkapnya, lihat the section called “Pengaturan komunikasi”.

6. Pilih Terapkan untuk pergi ke layar Profil AWS SDK.
7. Pilih Pilih Transportasi untuk memilih transportasi menggunakan bantuan nilai.
8. Pilih Simpan.

Memecahkan masalah AWS SDK untuk SAP ABAP

Bagian ini menyediakan langkah-langkah pemecahan masalah untuk kemungkinan skenario kesalahan.

Topik

- Kegagalan impor
- Kendala lokasi yang tidak ditentukan
- Kesalahan SSL
- Konfigurasi profil
- Otorisasi IAM
- Otorisasi untuk melakukan tindakan yang diperlukan
- Skenario aktif
- Karakter khusus dalam kode
- Konektivitas

Kegagalan impor

Masalah - Kelas 'CL_SYSTEM_UUID' tidak berisi antarmuka 'IF_SYSTEM_UUID_ _STATIC RFC4122

Penyebab - SAP Note 0002619546 hilang di sistem Anda.

Resolusi — Pastikan bahwa SAP Note 0002619546 diterapkan ke sistem Anda.

Kendala lokasi yang tidak ditentukan

Masalah - Batasan lokasi yang tidak ditentukan tidak kompatibel untuk titik akhir `region` tertentu permintaan ini dikirim

Penyebab - Bucket Amazon S3 Anda tidak memiliki parameter AWS Wilayah. `io_createbucketconfiguration`

Resolusi — Saat membuat bucket di Wilayah mana pun, kecualius`-east-1`, tentukan Wilayah bucket Amazon S3 Anda menggunakan `io_createbucketconfiguration` parameter di. `createbucket()` Anda tidak perlu menentukan batasan untuk. `us-east-1`

Contoh berikut menunjukkan `io_createbucketconfiguration` parameter yang dikonfigurasi dengan benar.

```
createbucket(
    iv_bucket = 'amzn-s3-demo-bucket'
    io_createbucketconfiguration = NEW /aws1/cl_s3_createbucketconf( 'us-west-1' )
).
```

Kesalahan SSL

Masalah - Ketidakcocokan Nama Host Sertifikat Server SSL atau jabat tangan SSL dengan docs.aws.amazon.com: 443 gagal: SSSLERR_NO_SSL_RESPONSE

Penyebab — `icm/HTTPS/client_sni_enabled` parameter tidak diatur ke `TRUE` dalam `DEFAULT` profil.

Resolusi — Gunakan langkah-langkah berikut untuk memecahkan masalah yang diberikan atau masalah terkait SSL lainnya.

1. Buka SAPGUI dan pergi ke bilah perintah.
2. Jalankan transaksi`RZ10`.
3. Buka Profil dan pilih `DEFAULT` profil. Versi ini diisi secara otomatis.
4. Di bagian Edit Profil, pilih Pemeliharaan diperpanjang, lalu pilih Ubah.
5. Cari `icm/HTTPS/client_sni_enabled` parameternya.
   - Jika parameter ada, edit nilai Parameter dan atur ke`TRUE`.
   - Jika parameter tidak ada, buat parameter menggunakan langkah-langkah berikut.
     1. Pilih Parameter.

        > **Note**
        >
        > Pastikan Anda memilih Parameter untuk pembuatan, dan tidak mengedit (ikon pensil).

     2. Masukkan `icm/HTTPS/client_sni_enabled` di bidang Nama Parameter.
     3. Masukkan `TRUE` di bidang Nilai parameter.
     4. Pilih Simpan.

6. Simpan perubahan ini di `DEFAULT` profil, dan Keluar.

Konfigurasi profil

Masalah - Tidak dapat menemukan konfigurasi di bawah profil <profile_name>dengan skenario DEFAULT untuk<sid>: <client>

Penyebab — <profile_name>Salah atau belum dikonfigurasi.

Resolusi — Gunakan langkah-langkah berikut untuk mengkonfigurasi profil.

1. Buka SAPGUI dan jalankan transaksi. `/n/AWS1/IMG`
2. Buka Application Configuration > SDK Profile.
   - Jika profil Anda dikonfigurasi, verifikasi bahwa nama profil sudah benar.
   - Jika profil Anda tidak dikonfigurasi, ikuti langkah-langkah untuk mengonfigurasi profil.
3. Pilih Entri Baru.
   a. Masukkan Nama dan Deskripsi untuk profil.
   b. Pilih Simpan.
4. Pilih entri yang Anda buat di langkah sebelumnya, lalu pilih Otentikasi dan Pengaturan.
5. Pilih Entri Baru, masukkan detail berikut, lalu pilih Simpan.
   - SID
   - Klien
   - ID Skenario
   - AWS Wilayah
   - Metode Otentikasi
     - Pilih Peran Instance melalui Metadata untuk sistem SAP yang berjalan di. AWS
     - Pilih Kredensyal dari Penyimpanan SSF untuk sistem SAP yang berjalan di tempat atau cloud lainnya.
6. Pilih Pemetaan Peran IAM > Entri Baru, masukkan detail berikut, dan pilih Simpan.
   - Nomor urut
   - Peran IAM Logis
   - Peran IAM ARN

Otorisasi IAM

Masalah — Tidak dapat mengambil peran <iam_role_arn>atau Pengguna: <user_arn>tidak berwenang untuk melakukan: sts: AssumeRole on resource: <iam_role_arn>

Penyebab — berikut ini mungkin menjadi alasan yang mungkin untuk kesalahan ini.

- Peran IAM yang salah ARN telah ditentukan
- Pengguna IAM tidak memiliki izin untuk mengakses peran IAM
- Kurangnya hubungan kepercayaan antara peran IAM yang diasumsikan dan asumsi peran IAM atau pengguna IAM

Resolusi — Gunakan langkah-langkah berikut untuk memastikan bahwa ARN peran IAM benar.

1. Buka SAPGUI dan jalankan transaksi. `/n/AWS1/IMG`
2. Buka Application Configuration > SDK Profile, dan pilih profil yang telah dikonfigurasi dengan peran IAM Anda.
3. Pilih Pemetaan Peran IAM dan verifikasi atau perbaiki ARN peran IAM Anda.
    - Jika ARN peran IAM Anda benar, pastikan bahwa peran IAM Anda telah dikonfigurasi dengan benar. Untuk informasi selengkapnya, lihat Memecahkan masalah peran IAM.

Otorisasi untuk melakukan tindakan yang diperlukan

Masalah - Pengguna <user_arn>tidak berwenang untuk melakukan: <action>pada sumber daya: <resource_arn>

Penyebab — Pengguna tidak memiliki izin untuk melakukan tindakan.

Resolusi — `user_arn` harus diatur dengan izin yang diperlukan `resource_arn` untuk melakukan yang ditentukan`action`. Untuk informasi selengkapnya, lihat Izin yang diperlukan untuk mengakses sumber daya IAM.

Skenario aktif

Masalah - Tidak ada skenario aktif yang dikonfigurasi

Penyebab — Pengaturan skenario aktif terlewatkan.

Resolusi - Lihat pengaturan Runtime untuk mengonfigurasi skenario aktif.

Karakter khusus dalam kode

Peringatan - Karakter 0x00A0 tidak dapat menjadi bagian dari kata ABAP

> **Note**
>
> Peringatan ini dapat didahului oleh berbagai pesan kesalahan.

Penyebab - Menyalin dan menempelkan kode dari berbagai sumber dapat menyisipkan karakter khusus dalam kode Anda.

Resolusi — Saat Anda menempelkan kode apa pun di editor kode sumber ABAP, Anda melihat pop-up berikut.

Karakter ruang yang tidak pecah terdeteksi. Konversi ke spasi?

Pilih Ya untuk menjawab pertanyaan ini. Juga, kami sarankan memilih kode untuk menyalinnya, daripada menggunakan tombol salin di kotak kode.

Konektivitas

Masalah - SCLNT_HTTP (411): Sambungan langsung ke tla.region.amazonaws.com:443 gagal: NIECONN_REFUSED (-10)

Penyebab — Sistem SAP tidak memiliki konektivitas internet, dan tidak dapat membuat koneksi TCP/IP ke port 443 dari tla.region.amazonaws.com.

Resolusi — Sistem SAP harus dapat membuat koneksi ke AWS titik akhir pada port HTTPS 443, baik secara langsung atau melalui server proxy. Anda dapat menetapkan/memverifikasi konektivitas internet dengan salah satu opsi berikut.

- Koneksi keluar langsung ke internet melalui NAT atau gateway internet
- Koneksi melalui server proxy

  Untuk informasi selengkapnya, lihat Koneksi melalui server proxy.

Topik tambahan

Bagian ini mencakup topik-topik berikut.

Topik

- AWS SDK untuk SAP ABAP rilis
- Lisensi SAP

AWS SDK untuk SAP ABAP rilis

AWS SDK untuk SAP ABAP dikirimkan dalam transportasi, dan AWS SDK untuk SAP ABAP - edisi BTP dikirimkan sebagai add-on. Mekanisme untuk mengimpor transportasi dan add-on berbeda tetapi fungsi teknisnya sama. Untuk informasi selengkapnya, lihat Menyiapkan.

Topik

- Strategi rilis
- Praktik terbaik
- Menambal SDK untuk SAP ABAP
- Memasang modul tambahan
- Menghapus instalasi SDK untuk SAP ABAP

Strategi rilis

Versi 1 AWS SDK untuk SAP ABAP sering diperbarui. Patch baru dirilis setiap minggu atau harian berdasarkan rilis dan pembaruan. Layanan AWS Patch untuk Layanan AWS dapat mencakup perbaikan bug dan perubahan lain yang memperbarui tingkat patch SDK. Untuk informasi selengkapnya, lihat AWS SDKs dan Kebijakan pemeliharaan alat.

Praktik terbaik

Kami merekomendasikan untuk mempertahankan tingkat patch SDK yang sama untuk SAP ABAP untuk semua sistem SAP (pengembangan, QA, dan produksi).

Saat menambal SDK, impor versi terbaru di kotak pasir Anda. Anda kemudian dapat mengimpornya ke pengembangan, QA, dan sistem produksi, mengikuti prosedur kontrol perubahan normal Anda.

Menambal SDK untuk SAP ABAP

Setiap SDK untuk rilis SAP ABAP dikirimkan sebagai satu set transportasi kumulatif, termasuk semua perbaikan bug, fitur, dan pembaruan. Tidak ada perbedaan antara tambalan dan transportasi instalasi. Anda harus mengimpor transportasi terbaru untuk menambal SDK untuk SAP ABAP.

Karena dependensi modul `core` Runtime dan API, modul API individual tidak dapat ditambal secara terpisah. Saat menambal, Anda harus memperbarui `core` modul dan semua modul API lain yang telah diinstal, bahkan jika modul tertentu tidak lagi digunakan.

Skenario berikut menjelaskan proses penambalan:

- Skenario 1: Pembaruan Versi SDK Reguler

  Saat menambal ke versi SDK terbaru:

  - Impor semua transportasi (modul inti dan API yang diinstal) secara bersamaan
  - Tidak diperlukan urutan impor khusus
  - Misalnya, jika Anda mengimpor`core`,`ec2`, dan `lmd` mengangkut ketika Anda menginstal SDK, Anda harus mengimpor transportasi terbaru untuk, `core``ec2`, dan `lmd` saat menambal.
- Skenario 2: Menambahkan Modul API Baru

  Saat mengimpor modul API baru (misalnya, Amazon `tex` Textract) dari rilis SDK yang berbeda dari modul yang Anda instal saat ini:

  - Semua modul yang ada harus ditambal agar sesuai dengan versi rilis SDK dari modul baru
  - Impor secara bersamaan (tidak ada urutan tertentu):
    - Transportasi terbaru untuk semua modul yang ada (`core`dan modul API yang diinstal)
    - Transportasi untuk modul API baru `tex`
- Skenario 3: Memperbarui Modul API yang Ada

  Saat memperbarui modul API yang ada (misalnya, Amazon Translate`xl8`) ke versi SDK terbaru, pembaruan modul individual tidak didukung karena saling ketergantungan. Anda harus menambal seluruh SDK dengan memperbarui `core` modul dan semua modul API yang diinstal bersama-sama, mengikuti proses yang diuraikan dalam. Regular SDK Version Update (p. 593)

Memasang modul tambahan

Impor transport untuk modul baru pada tingkat patch yang sama dengan modul yang ada `core` dan modul untuk menginstal modul API tambahan di sistem SAP Anda. Ikuti panduan the section called “Menambal SDK untuk SAP ABAP” jika Anda ingin mengimpor versi modul yang lebih baru. Ini memastikan bahwa level patch kompatibel di semua modul SDK.

Menghapus instalasi SDK untuk SAP ABAP

Untuk menghapus SDK untuk SAP ABAP, Anda harus mengunduh kit transportasi penghapusan dari release/uninstall-abapsdk-LATEST.zip. https://sdk-for-sapabap.aws.amazon.com/awsSdkSapabapV1/

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

Anda dapat mengunduh file tanda tangan dari https://sdk-for-sapabap.aws.amazon.com/awsSdkSapabapV1/Release/Uninstall-ABAPSDK-Latest.sig. Untuk memvalidasi file, lihat Verifikasi SDK untuk SAP ABAP.

Untuk setiap modul SDK yang diinstal pada sistem SAP Anda, transport penghapusan yang sesuai harus diimpor dari file ZIP sebelumnya. Anda dapat menghapus satu modul tanpa menghapus seluruh SDK. Anda dapat melakukannya dengan mengimpor hanya transport penghapusan untuk modul yang ingin Anda hapus. Jika Anda menghapus seluruh SDK dengan semua modulnya, maka transport penghapusan inti harus diimpor terakhir.

Kami menyarankan Anda menguji penghapusan instalasi di kotak pasir sebelum mencoba dalam pengembangan, QA atau sistem produksi.

Pertimbangan-pertimbangan

Sebelum mencopot pemasangan SDK, lihat pertimbangan berikut.

- Pengaturan konfigurasi SDK dari akan hilang. `IMG`Harus dikonfigurasi ulang pada instalasi.
- Jika Anda memiliki program Z yang mengandalkan SDK, mereka akan menghasilkan kesalahan sintaks setelah penghapusan SDK.
- Peran PFCG atau Bisnis yang berisi referensi otorisasi SDK akan memiliki otorisasi yang tidak valid setelah penghapusan SDK. Hapus referensi otorisasi SDK dari peran PFCG sebelum mencopot pemasangan SDK.

> **Note**
>
> AWS SDK untuk SAP ABAP - edisi BTP tidak dapat dihapus selama pratinjau pengembang.

Lisensi SAP

Penggunaan perangkat lunak SAP tunduk pada persyaratan SAP. Anda bertanggung jawab untuk mematuhi persyaratan lisensi SAP, termasuk distribusi perangkat lunak dan ketentuan lisensi tidak langsung. Setiap informasi yang diberikan bukanlah nasihat hukum, dan tidak boleh diandalkan untuk tujuan kepatuhan lisensi. Jika Anda memiliki pertanyaan tentang lisensi atau hak Anda atas perangkat lunak SAP, konsultasikan dengan tim hukum Anda, SAP, dan/atau pengecer SAP Anda.

Pertanyaan: Apakah SDK untuk penggunaan SAP ABAP memengaruhi lisensi SAP saya?

Jawaban: AWS SDK untuk SAP ABAP memungkinkan Anda untuk mengkonsumsi Layanan AWS dengan kode ABAP Anda sendiri. Ini digunakan dalam skenario integrasi antara sistem SAP dan Layanan AWS. Setiap skenario di mana data dari sistem SAP dikirim ke sistem pihak ketiga (non-SAP), atau dibuat oleh sistem itu, mungkin memiliki implikasi untuk lisensi tidak langsung. SAP memiliki beberapa pendekatan untuk mendefinisikan akses tidak langsung, seperti perhitungan berbasis pengguna dan perhitungan berbasis hasil. Metodologi untuk menentukan akses tidak langsung tergantung pada kontrak Anda dengan SAP. Anda harus mengetahui panduan yang diberikan dalam kontrak Anda dengan SAP, dan Anda dapat mendiskusikan hal ini lebih lanjut dengan SAP atau reseller mereka.

Pada tahun 2018, SAP merilis dua dokumen — Indirect Access Guide for SAP Installed Base Customers dan SAP ERP Pricing for Digital Age — Addressing Indirect/Digital Access. Dokumen-dokumen ini dapat ditemukan di situs web SAP, dan merupakan contoh pendekatan lisensi tidak langsung. Namun, dokumen-dokumen ini tidak mencerminkan perjanjian khusus Anda dengan SAP.

Riwayat dokumen untuk AWS SDK untuk SAP ABAP Panduan Developer

Tabel berikut menjelaskan rilis dokumentasi untuk AWS SDK untuk SAP ABAP.

Perubahan	Deskripsi	Tanggal
Konten baru	Menambahkan penggantian server Per-service proxy.	15 Mei 2026
Konten baru	Ditambahkan Menggunakan AWS SDK untuk SAP ABAP Knowledge MCP Server.	April 24, 2026
Konten baru	Ditambahkan Menggunakan Profil Sumber untuk Cross-Account Akses.	Maret 4, 2026
Konten baru	Ditambahkan Menggunakan Otentikasi Kunci Akses Rahasia dengan Enkripsi SSF.	Januari 12, 2025
Konten baru	Pratinjau pengembang SDK untuk SAP ABAP - edisi BTP.	31 Mei 2024
Konten baru	Ditambahkan Menggunakan sertifikat dengan Peran IAM Di Mana Saja.	1 Desember 2023
Konten baru	Menambahkan produk Building dengan SDK.	1 Desember 2023
Konten baru	Menambahkan perilaku Coba lagi.	1 Desember 2023
Konten baru	Menambahkan lisensi SAP.	September 22, 2023

Rilis publik	Peluncuran awal Panduan AWS SDK untuk SAP ABAP Pengembang.	30 Juni 2023
Konten baru	Menambahkan AWS SDK untuk SAP ABAP fitur.	30 Mei 2023
Konten baru	Menambahkan Pemecahan Masalah AWS SDK untuk SAP ABAP.	17 Februari 2023
Pratinjau pengembang	Pratinjau pengembang Panduan AWS SDK untuk SAP ABAP Pengembang.	17 November 2022

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.