aws

개발자 가이드

AWS SDK for SAP ABAP

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

AWS SDK for SAP ABAP: 개발자 가이드

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

Amazon의 상표 및 브랜드 디자인은 Amazon 외 제품 또는 서비스와 관련하여 고객에게 혼동을 일으킬 수 있는 방식이나 Amazon 브랜드 이미지를 떨어뜨리는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon과 제휴 관계이거나 관련이 있거나 후원 관계와 관계없이 해당 소유자의 자산입니다.

Table of Contents

AWS SDK for SAP ABAP란 무엇인가요? 1
의 기능 AWS SDK for SAP ABAP 1
정비 1
API 참조 2
요금 2
리소스 2
시작하기 3
1단계: AWS 계정 준비 3
SAP 사용자의 IAM 역할 3
Authentication 4
2단계: SDK 설치 6
3단계: SDK 구성 6
4단계: 기능 설정 8
5단계: SAP 사용자 권한 부여 10
6단계: 코드 작성 12
7단계: 애플리케이션 실행 14
설정 16
SAP 사전 조건 16
SDK for SAP ABAP 16
SDK for SAP ABAP - BTP 에디션 20
설치 AWS SDK for SAP ABAP 21
SDK 다운로드 21
파일 확인 21
AWS SDK 전송 22
SDK 설치 - BTP 에디션 25
SDK for SAP ABAP 설치 - BTP 에디션 26
모듈 26
SDK for SAP ABAP 패치 적용 - BTP 에디션 27
구성 28
글로벌 설정 29
기술 설정 30
시나리오 구성 30
애플리케이션 구성 31
SDK 프로필 31

논리적 리소스 해석기 ... 32
예제 ... 33
런타임 설정 ... 33
로그 및 추적 ... 34
옵트인: 향상된 원격 측정 ... 34
활성 시나리오 ... 34
고급 연결 시나리오 ... 34
프록시 서버를 통한 연결 ... 35
패킷 검사 방화벽을 통한 연결 ... 35
게이트웨이 엔드포인트 ... 35
사용자 지정 인터페이스 엔드포인트 ... 36
고급 라우팅 ... 37
다중 리전 엔드포인트에 액세스 ... 38
서비스 공급자 설정 ... 38
새로 고침, 추적, 원격 측정 ... 39
SAP 시스템 새로 고침 ... 39
추적 ... 40
원격 측정 ... 41
SDK 사용 ... 42
데이터 표시 ... 42
데이터 타입 ... 42
AWS 데이터 형식 ... 44
예제 프로그램 ... 45
사전 조건 ... 45
코드 ... 46
코드 섹션 ... 47
개념 ... 49
API 클래스 ... 49
추가 객체 ... 50
구조 클래스 ... 50
배열 ... 52
맵 ... 53
상위 수준 함수 ... 53
특성 ... 1
프로그래밍 방식 구성 ... 54
Waiters ... 55

페이지네이터 56
재시도 동작 57
프리서명자 57
교차 계정 IAM 역할 체인 58
제품 구축 58
제품 ID 설정 58
HTTP 요청을 로 사용자 지정 AWS 59
개선 사항 구현 59
개선 사항 필터링 59
개선 사항 코딩 60
제한 사항 60
코드 예제 62
ACM 64
작업 64
API Gateway 72
작업 64
Application Recovery Controller 80
작업 64
Aurora 85
작업 64
Auto Scaling 91
작업 64
Amazon Bedrock 런타임 100
Anthropic Claude 101
Stable Diffusion 104
Amazon Bedrock Agents Runtime 106
작업 64
CloudFront 108
작업 64
CloudWatch 110
작업 64
시나리오 118
CloudWatch Logs 120
작업 64
Amazon Cognito 자격 증명 공급자 123
작업 64

Amazon Comprehend 128
작업 64
AWS Config 142
작업 64
AWS Control Tower 144
작업 64
Firehose 156
작업 64
DynamoDB 158
기본 사항 158
작업 64
Amazon EC2 172
작업 64
Amazon ECR 194
작업 64
Amazon EMR 201
작업 64
EventBridge 스케줄러 207
작업 64
AWS Glue 212
작업 64
HealthImaging 224
작업 64
HealthLake 242
작업 64
IAM 255
작업 64
AWS IoT SiteWise 279
시작하기 279
작업 64
Amazon Keyspaces 287
작업 64
Kinesis 295
기본 사항 158
작업 64
AWS KMS 304

시작하기 279
작업 64
Lambda 324
기본 사항 158
작업 64
조직 338
작업 64
Amazon Pinpoint 343
작업 64
Amazon Pinpoint SMS 및 음성 API 348
작업 64
Amazon Polly 357
작업 64
Amazon RDS 366
작업 64
Amazon Redshift 371
작업 64
Amazon Rekognition 378
작업 64
Amazon S3 393
기본 사항 158
작업 64
시나리오 118
SageMaker AI 418
작업 64
시나리오 118
Secrets Manager 435
작업 64
Amazon SES 437
작업 64
Amazon SES API v2 454
작업 64
Amazon SNS 462
작업 64
시나리오 118
Amazon SQS 473

작업 64
시나리오 118
단계 함수 482
작업 64
Systems Manager 491
작업 64
Amazon Textract 500
작업 64
시나리오 118
Amazon Transcribe 511
작업 64
Amazon Translate 519
작업 64
시나리오 118
보안 528
시스템 인증 528
메타데이터 인증 529
비밀 액세스 키 인증 529
IAM Roles Anywhere를 사용한 인증서 기반 인증 530
교차 계정 액세스를 위한 소스 프로필 인증 530
다음 단계 531
IAM 보안 모범 사례 531
Amazon EC2 인스턴스 프로파일 모범 사례 531
SAP 사용자의 IAM 역할 532
소스 프로필 보안 고려 사항 535
SAP 권한 부여 536
구성을 위한 권한 부여 536
최종 사용자를 위한 SAP 권한 부여 537
보안 운영 539
저장 데이터 암호화 539
전송 중 데이터 암호화 539
API 사용 2
비밀 키(SSF) 539
사전 조건 540
절차 540
IAM Roles Anywhere 542

사전 조건 542
절차 543
소스 프로파일 58
사전 조건 546
절차 546
자격 증명 스토어 20
구성 단계 549
SDK에서 SAP 자격 증명 스토어 사용 551
문제 해결 555
가져오기 실패 555
지정되지 않은 위치 제약 555
SSL 오류 556
프로파일 구성 557
IAM 권한 부여 558
작업을 위한 권한 부여 558
활성 시나리오 34
특수 문자 559
연결 559
추가 주제 560
출시 560
릴리스 전략 560
모범 사례 531
SDK for SAP ABAP 패치 561
추가 모듈 설치 562
SDK for SAP ABAP 제거 562
SAP 라이선싱 563
문서 이력 564
..... dlxv

AWS SDK for SAP ABAP란 무엇인가요?

AWS SDK for SAP ABAP 는 AWS 가 ABAP 언어로 제공하는 서비스에 대한 인터페이스를 제공합니다. SDK를 사용하면 ABAP BADIs, 보고서, 트랜잭션, OData 서비스 및 Amazon Simple Storage Service(Amazon S3), Amazon DynamoDB AWS 서비스, 등과 같은 기타 ABAP 아티팩트 Amazon Translate를 구현할 수 있습니다. SAP NetWeaver 7.4부터 SAP Business Technology Platform 환경에서 ABAP 기반 시스템용를 개발할 수도 있습니다. 자세한 내용은 AWS SDK for SAP ABAP 설치 - BTP 에디션을 참조하세요.

주제

- 의 기능 AWS SDK for SAP ABAP
- SDK 메이저 버전에 대한 유지 관리 및 지원
- API 참조
- 요금
- 추가 리소스

의 기능 AWS SDK for SAP ABAP

AWS SDK for SAP ABAP 는 SAP 개발자에게 친숙하고 자연스럽게 느껴지도록 설계되었습니다. 예를 들어, 모두 `true`및 `false`문자열을 AWS 서비스 사용하여 XML 및 JSON 구조의 부울 데이터를 나타내지만 SAP ABAP용 SDK는 이를 ABAP 네이티브`'X'` 및`' '` 단일 문자 값으로 변환합니다. SDK for SAP ABAP는 데이터 유형 및 타임스탬프 형식을 포함하여 기본 ABAP 구문을 최대한 많이 사용합니다. 따라서 ABAP 프로그래머는 기본 JSON 및 XML 직렬화나 API 프로토콜의 연결 형식에 대해 걱정할 필요가 없습니다.

SDK 메이저 버전에 대한 유지 관리 및 지원

SDK 메이저 버전 및 기본 종속성의 유지 관리 및 지원에 대한 자세한 내용은 AWS SDK 및 도구 참조 안내서에서 다음 내용을 참조하세요.

- AWS SDKs 및 도구 유지 관리 정책
- AWS SDKs 및 도구 및 버전 지원 매트릭스

API 참조

API의 AWS SDK for SAP ABAP APIs. AWS SDK for SAP ABAP

AWS SDK for SAP ABAP TLAs의 전체 모듈 목록을 보려면 AWS SDK for SAP ABAP - 모듈 목록을 참조하세요.

SDK for SAP ABAP - BTP 에디션 개발자 미리 보기 TLAs의 전체 모듈 목록을 보려면 AWS SDK for SAP ABAP - BTP 에디션 - 모듈 목록을 참조하세요.

요금

AWS SDK for SAP ABAP 는 추가 비용 없이 사용할 수 있습니다. SDK에서 사용하는 AWS 리소스 및 서비스에 대해서만 비용을 지불하면 됩니다.

추가 리소스

이 안내서 외에도 SDK for SAP ABAP에 대한 다음의 온라인 리소스를 사용할 수 있습니다.

- SAP on AWS 설명서
- AWS 개발자 블로그
- AWS 개발자 포럼
- AWS SDK 코드 예제 라이브러리
- @awsdevelopers(Twitter)

시작하기 AWS SDK for SAP ABAP

이 섹션에서는 SDK를 시작하는 방법을 설명합니다. 여기에는 SDK 설치, 기본 구성 수행, 문구를 한 언어에서 다른 언어로 번역하는 Hello World 코드 예제 만들기에 대한 정보가 포함됩니다. AWS SDK를 처음 사용하는 경우 샌드박스 환경에서이 단계를 수행하는 것이 좋습니다.

단계(Steps)

- 1단계: AWS 계정 준비
- 2단계: SDK 설치
- 3단계: SDK 구성
- 4단계: 기능 설정
- 5단계: SAP 사용자 권한 부여
- 6단계: 코드 작성
- 7단계: 애플리케이션 실행

1단계: AWS 계정 준비

SDK for SAP ABAP를 시작하려면 활성 AWS 계정 (이)가 있어야 합니다. SAP 시스템이 온프레미스, SAP Business Technology Platform(BTP) 또는 다른 클라우드 공급자와 호스팅되는 AWS 계정 경우에 도가 필요합니다.

SAP 시스템이 AWS 클라우드에서 실행 중인 경우에서 서비스를 호출 AWS 합니다 AWS 계정.

주제

- SAP 사용자의 IAM 역할
- Authentication

SAP 사용자의 IAM 역할

- AWS Identity and Access Management 사용자 설명서에 제공된 지침에 따라 IAM 역할을 생성합니다. 자세한 내용은 AWS 서비스에 권한을 위임할 역할 생성을 참조하세요. 나중에 사용할 수 있도록 IAM 역할의 Amazon 리소스 이름(ARN)을 적어 둡니다.
- 사용 사례로 Amazon EC2를 선택합니다.
- 역할 이름으로 `SapDemoTranslate`을(를) 사용합니다.

- `TranslateReadOnly` 프로파일을 역할에 추가합니다.
- SAP 시스템이 역할을 수임할 수 있으려면 역할에 다음 엔티티가 있어야 합니다. *"111122223333"*를 AWS 계정 번호로 바꿉니다.

  JSON

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

  이 예제는의 AWS 계정 모든 보안 주체가 역할을 수임할 *"111122223333"* 수 있음을 보여줍니다. 이는 개념 증명에 적합한 광범위한 권한입니다. 다음 예와 같이 프로덕션에 더 좁은 보안 주체를 사용할 수 있습니다.

  - 특정 사용자 - SAP 시스템에서 다음 중 하나를 사용하는 경우:
    - 온프레미스 SAP 시스템의 SSF 암호화 자격 증명
    - SAP BTP의 SAP Credential Store 서비스, ABAP 환경의 자격 증명
  - 특정 역할 — SAP 시스템이 Amazon EC2에 있고 인스턴스 프로파일이 있는 경우.
  - Amazon EC2 — SAP 시스템이 Amazon EC2에 있고 인스턴스 프로파일이 없는 경우.

자세한 내용은 IAM 보안 모범 사례를 참조하세요.

Authentication

인증은 SAP 시스템이 호스팅되는 위치에 따라 달라집니다.

위치

- AWS 클라우드에서

- 온프레미스, SAP BTP 또는 기타 클라우드

AWS 클라우드에서

SAP 시스템이 실행 중인 EC2 인스턴스에 다음 권한이 있는 인스턴스 프로파일이 있는지 확인합니다.

JSON

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

이전 단계에서 기록해 둔 ARN을 추가합니다.

이 권한을 사용하면 SAP 시스템이 ABAP 사용자를 대신하여 `SapDemoTranslate` 역할을 수임할 수 있습니다.

온프레미스, SAP BTP 또는 기타 클라우드

SAP 시스템이 온프레미스, SAP BTP 또는 다른 클라우드에 있는 경우 다음 단계에 따라 보안 액세스 키를 사용하여 인증을 위한 연결을 설정합니다.

1. IAM 사용자를 생성합니다. 자세한 정보는 IAM 사용자(콘솔) 생성을 참조하세요.
2. IAM 사용자의 이름으로 `SapDemoSID`을(를) 사용합니다. `SID`은(는) SAP 시스템의 시스템 ID입니다.
3. 이 사용자에게 `SapDemoTranslate` 역할을 할당합니다.

`access_key` 및를 보관합니다`secret_access_key`. SAP 시스템에서 이러한 자격 증명을 구성해야 합니다.

> **Note**
>
> SAP 시스템이 온프레미스, SAP BTP 또는 다른 클라우드에 있는 경우 다음 옵션 중 하나를 사용하여 인증할 수 있습니다.
>
> - SSF 또는 SAP 자격 증명 스토어를 사용한 보안 액세스 키 인증
> - IAM Roles Anywhere에서 인증서 사용

2단계: SDK 설치

설치 지침은 다음 탭을 참조하세요.

SDK for SAP ABAP

SAP 시스템에서 SAP ABAP 전송을 위한 SDK를 가져옵니다. 모든 클라이언트로 전송을 가져올 수 있습니다. 자세한 내용은 SDK for SAP ABAP 설치를 참조하세요.

SDK for SAP ABAP - BTP edition

제품 배포 애플리케이션을 사용하여 SDK for SAP ABAP - BTP 에디션을 설치합니다. 자세한 내용은 SDK for SAP ABAP 설치 - BTP 에디션을 참조하세요.

3단계: SDK 구성

SDK를 구성하기 전에 필요한 권한이 있는지 확인합니다. 자세한 내용은 SAP 권한 부여를 참조하세요.

구성 지침은 다음 탭을 참조하세요.

SDK for SAP ABAP

`/AWS1/IMG` 트랜잭션을 실행하여 SDK for SAP ABAP 구현 안내서를 엽니다. 이 트랜잭션을 실행하려면 SAP 시스템의 명령 모음에 `/n/AWS1/IMG` 을(를) 입력한 다음 입력을 선택합니다.

다음 구성을 완료합니다.

- 기술 사전 조건으로 이동합니다.
  - 권장 파라미터 및 HTTPS 연결을 검토합니다.
- 글로벌 설정 → 시나리오 구성으로 이동합니다.
  - 글로벌 설정의 권장 사항에 따라 설정을 변경합니다.

- 글로벌 설정 → 기술 설정으로 이동합니다.
  - 글로벌 설정의 권장 사항에 따라 설정을 변경합니다.
- 런타임 설정 → 로그 및 추적으로 이동합니다.
  - 새 항목을 선택합니다.
    - 추적 수준: 추적 없음.
    - 최대 덤프 라인: `100`.
    - 옵트인: enh 원격 분석: 이 항목은 비워둡니다.
  - 저장을 선택합니다.
- 런타임 설정 → 활성 시나리오로 이동합니다.
  - 새 시나리오에서 `DEFAULT`을(를) 선택합니다.
  - 시나리오 변경 커밋을 선택합니다.
  - 프롬프트를 수락합니다.

온프레미스 시스템의 사전 조건

SAP 시스템이 온프레미스 또는 다른 클라우드에서 실행되는 경우 보안 인증을 SAP 데이터베이스에 저장해야 합니다. 보안 인증은 SAP SSF를 사용하여 암호화되며 SAP의 CommonCryptoLib과 같이 구성된 암호화 라이브러리가 필요합니다.

SDK for SAP ABAP용 SSF를 구성하는 단계는 `/AWS1/IMG` 트랜잭션에 설명되어 있습니다.

> **Note**
>
> SAP 시스템이 Amazon EC2에서 실행 중인 경우에는 위의 사전 조건이 적용되지 않습니다. Amazon EC2에서 실행되는 SAP 시스템은 Amazon EC2 인스턴스 메타데이터에서 단기 자동 교체 보안 인증을 검색합니다.

SDK for SAP ABAP - BTP edition

웹 브라우저에서 ABAP 환경을 열고 사용자 지정 비즈니스 구성 애플리케이션으로 이동합니다.

다음 구성을 완료합니다.

- 시나리오 구성으로 이동합니다.
  - 글로벌 설정의 권장 사항에 따라 설정을 변경합니다.

- 기술 설정으로 이동합니다.
  - 글로벌 설정의 권장 사항에 따라 설정을 변경합니다.

4단계: 기능 설정

설정 지침은 다음 탭을 참조하세요.

SDK for SAP ABAP

트랜잭션을 실행/AWS1/IMG (명령 표시줄/n/AWS1/IMG에를 입력하고 Enter를 선택)하여 AWS SDK 구현 안내서를 엽니다.

- 애플리케이션 구성 → SDK 프로필로 이동합니다.
  - 새 항목을 선택합니다.
    - 프로파일: DEMO.
    - 설명: Demo profile.
    - 저장을 선택합니다.
- 생성한 항목을 강조 표시하고 인증 및 설정 트리 브랜치를 클릭합니다.
  - 새 항목을 선택합니다.
    - SID: 현재 사용 중인 SAP 시스템의 시스템 ID입니다.
    - 클라이언트: 현재 사용 중인 SAP 시스템의 클라이언트입니다.
    - 시나리오 ID: 기본 관리자가 생성한 DEFAULT 시나리오를 찾을 수 있는 드롭다운 목록입니다.
    - AWS 리전: 호출할 AWS 리전을 입력합니다. SAP 시스템이 실행 중인 경우 실행 중인 AWS 리전을 AWS입력합니다.
    - 인증 방법:
      - SAP 시스템이 Amazon EC2에서 실행 중인 경우 메타데이터를 통한 인스턴스 역할을 선택합니다.
      - SAP 시스템이 온프레미스나 다른 클라우드에서 실행 중인 경우 SSF 스토리지 보안 인증을 선택합니다.
        - 보안 인증 설정을 선택합니다.
        - 이전 단계에서 생성한 액세스 키 ID와 비밀 액세스 키를 입력합니다.
  - IAM 역할 비활성화를 공란으로 유지합니다.
  - 저장을 선택합니다.

- IAM 역할 매핑 트리 브랜치를 클릭합니다.
  - 새 항목을 선택합니다.
    - 시퀀스 번호 입력: 010.
    - 논리적 IAM 역할 입력: TESTUSER
    - IAM 역할 ARN 입력: 이전 단계에서 생성한 `TranslateReadOnly` 정책이 포함된 IAM 역할의 arn:aws:를 입력합니다.

SDK for SAP ABAP - BTP edition

SAP 자격 증명 스토어를 사용하여 인증을 설정합니다. 자세한 내용은 SAP 자격 증명 스토어 사용을 참조하세요.

웹 브라우저에서 ABAP 환경을 열고 사용자 지정 비즈니스 구성 애플리케이션으로 이동합니다.

- SDK 프로파일로 이동합니다.
  - 편집을 선택하여 새 프로필을 생성합니다.
    - 프로파일: `DEMO`.
    - 설명: `Demo profile`.
- 생성된 항목 옆의 오른쪽 화살표 키를 선택하여 인증 및 설정 탭으로 이동합니다.

  새 항목을 선택합니다.
  - SID: 현재 사용 중인 SAP 시스템의 시스템 ID입니다.
  - 클라이언트: 현재 사용 중인 SAP 시스템의 클라이언트입니다.
  - 시나리오 ID: 기본 관리자가 생성한 DEFAULT 시나리오를 찾을 수 있는 드롭다운 목록입니다.
  - AWS 리전: 호출할 AWS 리전을 입력합니다. SAP 시스템이 실행 중인 경우 실행 중인 AWS 리전을 AWS입력합니다.
  - 인증 방법: SAP 자격 증명 스토어에서 자격 증명을 선택합니다.
  - SAP 자격 증명 스토어에 저장된 자격 증명의 네임스페이스와 키 이름을 입력합니다.
  - SDK for SAP ABAP - BTP 에디션과 SAP 자격 증명 저장소 간의 통신을 설정하기 위해 생성된 통신 계약의 이름을 입력합니다.
  - IAM 역할 비활성화를 공란으로 유지합니다.
- 생성된 항목 옆의 오른쪽 화살표 키를 마우스 오른쪽 버튼으로 클릭하여 IAM 역할 매핑 탭으로 이동합니다.

  새 항목을 선택합니다.

- 시퀀스 번호 입력: 010.
- 논리적 IAM 역할 입력: TESTUSER
- IAM 역할 ARN 입력: 이전 단계에서 생성한 `TranslateReadOnly` 정책이 포함된 IAM 역할의 arn:aws:를 입력합니다.

5단계: SAP 사용자 권한 부여

SAP 사용자는 기본적으로 AWS 기능을 사용할 권한이 없습니다. 사용자는 SAP 인증을 사용하여 명시적으로 권한을 부여받아야 합니다. 자세한 내용은 다음 탭을 참조하세요.

SDK for SAP ABAP

PFCG 역할 생성

- PFCG 트랜잭션으로 이동
- 역할 이름 ZAWS_SDK_DEMO_TESTUSER을(를) 입력하고 단일 역할 생성을 선택합니다.
  - 설명: `Role for demo AWS SDK functionality`.
  - 권한 승인 탭으로 이동합니다.
  - 권한 승인 데이터 변경을 선택하고 정보 팝업을 수락합니다.
  - 템플릿 선택 팝업에서 템플릿 선택 안함을 선택합니다.
  - 도구 모음에서 수동 추가를 선택합니다.
  - 다음 권한 부여 객체를 추가합니다.
    - `/AWS1/LROL`
    - `/AWS1/SESS`
  - 인증 트리에 다음을 입력합니다.
    - AWS API에 액세스하기 위한 프로파일: `DEMO`
    - 논리적 IAM 역할: `TESTUSER`
  - 저장을 선택합니다.
  - 생성을 선택합니다.
  - 뒤로를 선택합니다.
  - 저장을 선택하여 역할을 저장합니다.

SAP 사용자에게 PFCG 역할 할당

ZAWS_SDK_DEMO_TESTUSER 역할이 할당된 모든 사용자는 AWS SDK 프로파일에 구성된 설정으로 DEMO SDK 함수를 사용할 수 있는 권한이 부여됩니다. 또한 승인된 사용자는 해당 프로필의 TESTUSER 논리적 IAM 역할에 매핑된 IAM 역할을 수임하게 됩니다.

- SU01 트랜잭션을 실행합니다.
  - AWS SDK 기능을 테스트할 SAP 사용자의 사용자 ID를 입력합니다.
  - 변경을 선택합니다.
  - 역할 탭으로 이동하여 사용자에게 해당 ZAWS_SDK_DEMO_TESTUSER 역할을 할당합니다.
  - 저장을 선택합니다.

SDK for SAP ABAP - BTP edition

비즈니스 역할 생성

- 웹 브라우저에서 ABAP 환경을 열고 비즈니스 역할 유지 관리 애플리케이션으로 이동합니다.
- 템플릿에서 생성을 선택하고 다음 세부 정보를 입력합니다.
  - 템플릿 -를 선택합니다**/AWS1/RT_BTP_ENDUSER**.
  - 새 비즈니스 역할 ID - ID를 입력합니다.
  - 새 비즈니스 역할 설명 - 설명을 입력합니다.
- 확인을 선택하여 비즈니스 역할에 대한 페이지를 확인합니다.
- 일반 역할 세부 정보 탭에서 액세스 범주로 이동하여 쓰기, 읽기, 값 도움말 필드를 제한됨으로 설정합니다.
- 제한 유지를 선택하고 왼쪽 탐색 창에서 할당된 제한 유형을 확장합니다. 제한 및 값 섹션에서 다음 필드를 업데이트합니다.
  - SDK 세션 선택에서 SDK 프로파일 옆의 연필 아이콘을 선택하고 범위 탭으로 이동합니다. **DEMO**를 입력하고 추가를 선택합니다.
  - 논리적 IAM 역할 선택에서 논리적 IAM 역할 옆의 연필 아이콘을 선택하고 범위 탭으로 이동합니다. **TESTUSER**를 입력하고 추가를 선택합니다.

    SDK 프로파일 옆의 연필 아이콘을 선택하고 범위 탭으로 이동합니다. **DEMO**를 입력하고 추가를 선택합니다.
- 비즈니스 역할 템플릿으로 돌아가서 비즈니스 사용자 탭을 엽니다. 추가를 선택하여 새로 생성된 비즈니스 역할을 SDK 기능을 테스트할 SAP 비즈니스 사용자에게 할당합니다. 저장을 선택합니다.

생성된 비즈니스 역할에 할당된 모든 비즈니스 사용자는 AWS SDK 프로필에 구성된 설정으로 DEMO SDK 함수를 사용할 수 있는 권한이 부여됩니다. 또한 승인된 사용자는 해당 프로필의 TESTUSER 논리적 IAM 역할에 매핑된 IAM 역할을 수임하게 됩니다.

6단계: 코드 작성

자세한 내용은 다음 탭을 참조하세요.

SDK for SAP ABAP

1. SE38 트랜잭션을 엽니다.
   - 프로그램 이름으로 ZDEMO_TRANSLATE_HELLO_WORLD을(를) 입력합니다.
   - Create을(를) 선택합니다.
   - 제목으로 AWS SDK Hello World In Any Language을(를) 입력합니다.
   - 유형: 실행 프로그램을 선택합니다.
   - 상태: 테스트 프로그램을 선택합니다.
   - 저장을 선택합니다.
   - 프로그램을 로컬 객체로 저장합니다.

   다음 코드를 추가합니다.

```
*&---------------------------------------------------------------------*
*& 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. ABAP 클래스를 생성해야 하는 패키지를 마우스 오른쪽 버튼으로 클릭한 다음 새로 만들기 > ABAP 클래스를 선택합니다.
2. 클래스 이름**ZCL_DEMO_XL8_SIMPLE**에를 입력하고 클래스 설명을 추가합니다. 다음을 선택합니다.
3. 전송 요청을 생성하거나 선택합니다. 마침을 선택합니다.

다음 코드를 추가합니다.

```
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.
```

SDK를 사용하는 ABAP 코드를 작성하는 방법에 대한 자세한 내용은 사용을 AWS SDK for SAP ABAP 참조하세요.

7단계: 애플리케이션 실행

자세한 내용은 다음 탭을 참조하세요.

SDK for SAP ABAP

SE38에서 애플리케이션을 실행합니다. 성공하면 다음과 같은 결과가 출력됩니다.

```
Source Phrase: Hello, World
```

```
Target Phrase: Hola, mundo
```

권한 부여, 구성 또는 기본 사전 조건이 누락된 경우 오류 메시지가 표시될 수 있습니다. 다음 예를 참조하세요.

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

SAP 시스템에서 IAM 역할을 수임하도록 IAM 권한이 구성되어 있지 않은 상태에서 SAP 역할이 SDK 프로파일을 사용하고 이를 논리적 IAM 역할에 매핑할 수 있는 권한을 부여하는 경우 다음과 같은 결과가 출력됩니다.

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

이 경우 the section called “1단계: AWS 계정 준비”에 정의된 IAM 역할, 사용자 또는 둘 다에 대한 IAM 권한 및 신뢰 구성을 검토합니다.

SDK for SAP ABAP - BTP edition

Eclipse > Run As > ABAP Application(콘솔)에서 애플리케이션을 실행합니다. 성공하면 다음과 같은 결과가 출력됩니다.

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

권한 부여, 구성 또는 기본 사전 조건이 누락된 경우 오류 메시지가 표시될 수 있습니다. 다음 예를 참조하세요.

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

SAP 시스템에서 IAM 역할을 수임하도록 IAM 권한이 구성되어 있지 않은 상태에서 SAP 역할이 SDK 프로파일을 사용하고 이를 논리적 IAM 역할에 매핑할 수 있는 권한을 부여하는 경우 다음과 같은 결과가 출력됩니다.

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

이 경우 the section called “1단계: AWS 계정 준비”에 정의된 IAM 역할, 사용자 또는 둘 다에 대한 IAM 권한 및 신뢰 구성을 검토합니다.

설정

이 섹션에서는 AWS SDK for SAP ABAP을(를) 사용하기 위한 개발 환경을 설정하는 방법에 대한 정보를 제공합니다.

주제

- SAP 사전 조건
- 설치 AWS SDK for SAP ABAP
- AWS SDK for SAP ABAP 설치 - BTP 에디션

SAP 사전 조건

SDK 설치를 위한 다음 사전 조건은 SAP 시스템이 호스팅되는 경우에 적용됩니다 AWS.

주제

- AWS SDK for SAP ABAP의 사전 조건
- AWS SDK for SAP ABAP - BTP 에디션의 사전 조건

AWS SDK for SAP ABAP의 사전 조건

다음은 AWS SDK for SAP ABAP의 사전 조건입니다.

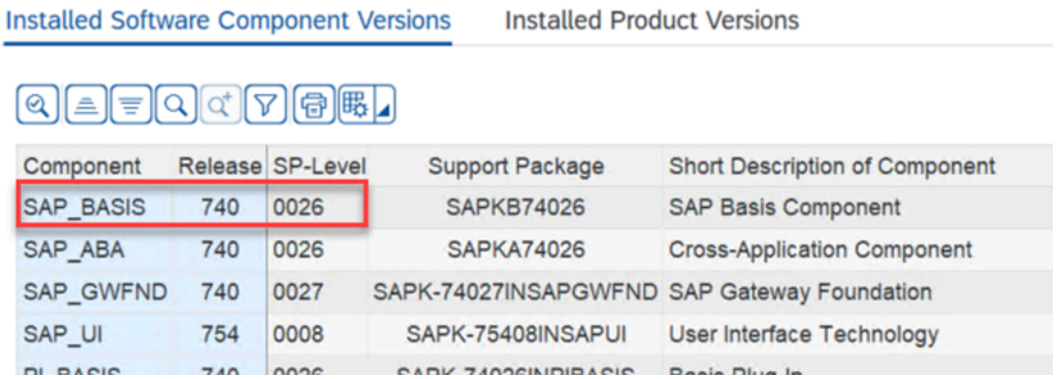
주제

- 기본 릴리스
- 커널 릴리스
- 파라미터
- Notes
- 아웃바운드 연결
- HTTPS 연결
- Amazon EC2 인스턴스 메타데이터 액세스

기본 릴리스

SDK for SAP ABAP는 SAP NetWeaver 7.4 이상과 호환됩니다. SDK for SAP ABAP는 어떤 SAP 애플리케이션 테이블에도 영향을 미치지 않습니다. SAP Enterprise Resource Planning 및 SAP Landscape Transformation Replication Server와 같은 애플리케이션에 전혀 구애받지 않습니다.

`SAP_BASIS 740`에 대해 지원되는 최소 SP 수준은 `SP 0008`입니다. 자세한 내용은 SAP Note 1856171 - CL_HTTP_ENTITY에서 동일한 이름의 지원 양식 필드 지원(SAP 포털 액세스 필요)을 참조하세요. 비즈니스 요구 사항에 따라 다음 이미지와 같이 더 높은 SP 수준을 선택할 수 있습니다.



`SAP_BASIS 750` 이상 릴리스에는 최소 SP 수준 요구 사항이 없습니다.

커널 릴리스

SDK for SAP ABAP와 HTTP 연결을 위해 인터넷 통신 관리자(ICM)를 사용하는 도구는 암호화, HTTP, XML 및 JSON 기능을 위해 SAP 커널을 사용합니다. SAP NetWeaver 플랫폼과 호환되는 최신 커널 릴리스를 사용하는 것이 좋습니다. 최소 요구 사항은 커널 릴리스 741입니다. 자세한 내용은 SAP Note 2083594 - SAP 커널 버전 및 SAP 커널 패치 수준(SAP 포털 액세스 필요)을 참조하세요.

커널 릴리스 741 또는 742를 사용하는 경우 다음과 같은 패치 수준이 필요합니다.

- 741 패치 번호 212
- 742 패치 번호 111

파라미터

SAP 시스템은 다음 SAP Notes(SAP 포털 액세스 필요)에 설명된 대로 서버 이름 표시(SNI)를 지원해야 합니다.

- SAP Note 2124480 - ICM/Web Dispatcher: 클라이언트로서의 TLS 확장 서버 이름 표시(SNI)

- SAP Note 2582368 - saphttp, sapkprotp, sldreg를 통한 TLS 확장 SNI의 클라이언트 측 전송을 위한 SapSSL 업데이트

DEFAULT.PFL 파일에서 다음 파라미터를 구성합니다.

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

Notes

다음 SAP Note를 시스템에 적용합니다.

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

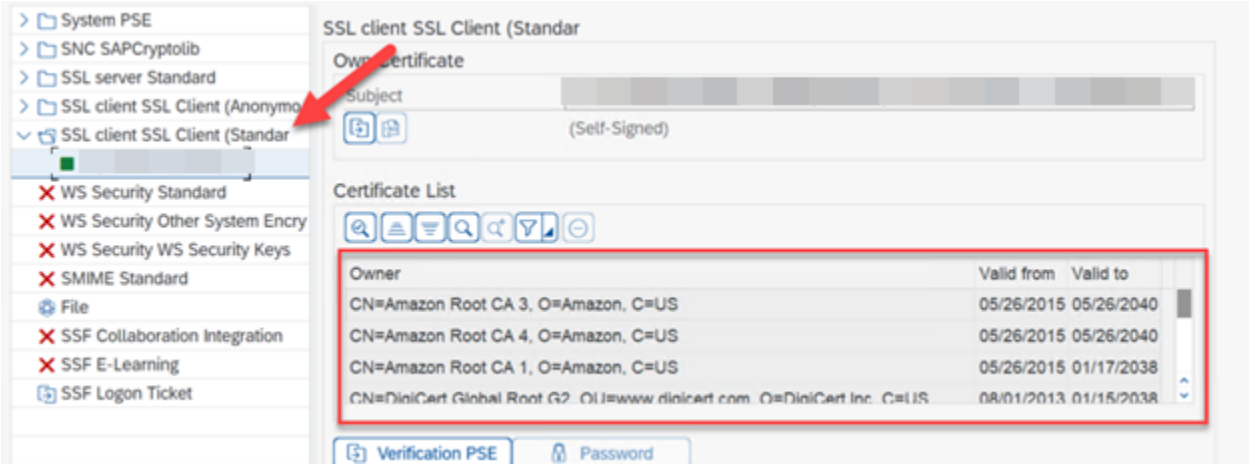
아웃바운드 연결

SDK for SAP ABAP는 HTTPS 클라이언트입니다. SAP 시스템은 HTTPS 메시지를 아웃바운드로 전송합니다. 인바운드 연결이 필요하지 않습니다.

HTTPS 연결

모든 AWS API 호출은 암호화된 HTTPS 채널을 사용하여 이루어집니다. 아웃바운드 HTTPS 연결을 설정하려면 AWS 인증서를 신뢰하도록 SAP 시스템을 설정해야 합니다.

1. https://www.amazontrust.com/repository/로 이동합니다.
2. 루트 CA에서 PEM 링크를 사용하여 모든 인증서를 다운로드합니다.
3. 다음 이미지에 표시된 대로 각 SAP 시스템에서 SSL Client (Standard) PSE의 STRUST에 위치한 인증서를 가져옵니다.



Amazon EC2 인스턴스 메타데이터 액세스

ABAP 시스템은 localhost(http://169.254.169.254)에 암호화되지 않은 HTTP 연결을 만들어 Amazon EC2 인스턴스 메타데이터를 활성화합니다. HTTP 채널은 로컬 서버에서 AWS 자격 증명을 검색하는 데만 사용됩니다. HTTP 트래픽은 호스트 내에 유지됩니다.

메타데이터를 사용하면의 SAP 시스템이 SAP Secure Store에 보안 키를 저장하지 않고도 안전하게 인증 AWS 할 수 있습니다. 이 기능은 Amazon EC2에서 호스팅되는 SAP 시스템에만 적용됩니다.

다음 파라미터로 DEFAULT.PFL 파일을 구성하여 SAP 시스템이 암호화되지 않은 아웃바운드 HTTP 연결을 만들 수 있도록 합니다.

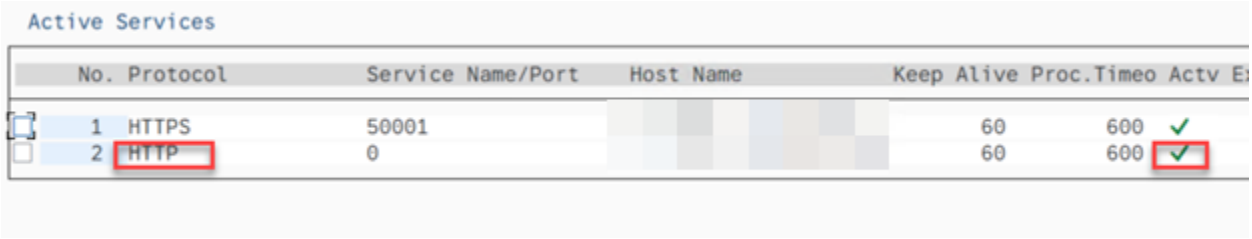
```
icm/server_port_<xx> = PROT=HTTP,PORT=8000,TIMEOUT=60,PROCTIMEOUT=600
```

다음 파라미터를 사용하면 인바운드 연결을 활성화하지 않고도 아웃바운드 HTTP 연결을 활성화할 수 있습니다.

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

다음 단계에 따라 SAP 시스템이 아웃바운드 HTTP 연결에 맞게 구성되어 있는지 확인합니다.

1. SMICM 트랜잭션을 실행합니다.
2. 활성 서비스로 이동합니다.
3. 다음 이미지와 같이 HTTP 행의 활성 열 아래에 녹색 확인 표시가 있는지 확인합니다.



AWS SDK for SAP ABAP - BTP 에디션의 사전 조건

다음은 AWS SDK for SAP ABAP - BTP 에디션의 유일한 사전 조건입니다.

주제

- SAP Landscape 포털 – BTP 에디션
- SAP 자격 증명 스토어 - BTP 에디션

SAP Landscape 포털 – BTP 에디션

이 사전 조건은 AWS SDK for SAP ABAP - BTP 에디션에만 적용됩니다.

SAP Landscape 포털은 SAP BTP 환경에 추가 기능을 설치하는 유일한 지원 메커니즘입니다. 이 서비스를 사용하도록 구독되어 있는지 확인합니다. 자세한 내용은 풍경 포털을 참조하세요.

SAP 자격 증명 스토어 - BTP 에디션

이 사전 조건은 AWS SDK for SAP ABAP - BTP 에디션에만 적용됩니다.

개발자 미리 보기에서 보안 액세스 키 인증은 AWS SDK for SAP ABAP - BTP 에디션을 인증하는 유일한 지원 메커니즘입니다. SDK는 자격 증명 스토어에서 자격 증명을 읽고 보안 액세스 키를 안전하게 저장합니다.

다음 사전 조건을 충족해야 합니다.

- 자격 증명 스토어 구독.
- 자격 증명 스토어는 BTP 하위 계정에 대한 권한으로 할당됩니다. 자세한 내용은 초기 설정을 참조하세요.
- 자격 증명 스토어에 대한 표준 계획이 있는 서비스 인스턴스입니다. 자세한 내용은 서비스 인스턴스 생성을 참조하세요.

자세한 내용은 SAP 자격 증명 스토어 사용을 참조하세요.

SAP 자격 증명 스토어 서비스는 ABAP BTP 시스템 외부의 SAP BTP에서 실행됩니다. 자세한 내용은 SAP 자격 증명 스토어를 참조하세요.

설치 AWS SDK for SAP ABAP

주제

- SDK for SAP ABAP 다운로드
- SDK for SAP ABAP 파일 확인 - 선택 사항
- AWS SDK 전송

SDK for SAP ABAP 다운로드

https://sdk-for-sapabap.aws.amazon.com/awsSdkSapabapV1/release/abapsdk-LATEST.zip에서 SDK를 다운로드합니다.

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

다운로드가 완료되면 다운로드한 파일을 /tmp/awssdk와 같은 디렉토리에 압축을 푸는 것이 좋습니다.

SDK for SAP ABAP 파일 확인 - 선택 사항

SDK 파일의 서명을 검증하는 이 선택적 단계는 SDK가 변조되지 않았는지 확인하는 데 도움이 됩니다. 다음 단계에 따라 SDK 파일을 확인할 수 있습니다.

1. 다음 명령을 사용하여 SDK SIGNATURE 파일을 다운로드합니다.

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

2. 다음 퍼블릭 키를 복사해 abapsdk-signing-key.pem 파일에 저장합니다.

   ```
   -----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. 다음 명령을 사용하여 다운로드한 SDK ZIP 파일을 확인합니다. 명령에는 많은 Linux 배포판의 일부인 `openssl`이(가) 필요합니다.

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

4. 이전 명령의 출력이 `Verified OK`인지 확인합니다.
5. 출력이 `Verification Failure`와(과) 같으면 이전 단계를 반복합니다. 계속해서 실패한 출력을 수신하는 경우 SDK를 설치하지 말고 지원에 문의하세요.

AWS SDK 전송

주제

- 내용
- 가져오기
- 네임스페이스

내용

SDK for SAP ABAP 설치는 ABAP 전송을 통해 완료됩니다. 이러한 전송을 개발 또는 샌드박스 환경으로 가져와야 합니다.

각 SDK for SAP ABAP 릴리스는 이전 버전을 완전히 대체합니다. 증분 전송을 적용할 필요가 없습니다. 전송은 ZIP 파일 번들로 제공됩니다. ZIP 파일의 구조는 다음과 같습니다.

```
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` 폴더에는 하위 폴더 `core`가 있습니다. `core` 하위 폴더에는 코어 런타임 전송과 모듈의 세 글자 약어로 이름이 지정된 각 모듈의 하위 폴더가 포함되어 있습니다. TLA의 전체 모듈 목록은 AWS SDK for SAP ABAP - 모듈 목록을 참조하세요.

AWS SDK 전송은 워크벤치 요청입니다. TMS 경로 구성에 따라 SDK를 이전 시스템으로 가져온 후 SDK가 품질 보증 및 프로덕션 대기열에 자동으로 전달되지 않을 수 있습니다. 각 시스템의 대기열에 수동으로 추가해야 합니다.

프로젝트가 다음 단계를 수행할 준비가 되면 비즈니스 기능이 있는 자체 Z 코드가 포함된 별도의 전송과 함께 AWS SDK를 가져올 수 있습니다. SAP 변경 요청 관리(CharM)와 같은 변경 제어 시스템을 사용하는 경우 타사 전송을 올바르게 처리하도록 ChaRM 관리자에게 문의하세요.

가져오기

주제

- 키 포인터
- 가져오기 소요 시간

AWS SDK 전송은 클라이언트와 독립적입니다. 코어 전송은 필수이며 SDK 런타임 코드, 용 API AWS Security Token Service및 Amazon Simple Storage Service용 API를 포함합니다. 나머지 SDK 모듈은 각각 별도의 전송으로 제공됩니다. 시스템에서 SDK의 크기를 작게 유지하기 위해 각 SDK 모듈은 선택 사항입니다. 비즈니스 로직에 필요한 경우 나중에 추가 모듈을 설치할 수 있습니다.

예를 들어 Amazon S3 및 용 APIs를 사용하려면 `core` 전송(코어 런타임, Amazon S3 및 AWS STS 모듈 포함)과 `x18` 전송(용 모듈 포함 Amazon Translate)을 Amazon Translate가져옵니다.

SDK for SAP ABAP API의 전체 목록을 보려면 SDK for SAP ABAP - API 참조 안내서를 참조하세요.

다음은 AWS SDK 전송을 가져올 때 사용되는 키 포인터입니다.

- 각 전송은 Knnnnnn.AWS 및 Rnnnnnn.AWS(으)로 제공됨
  - Knnnnnn.AWS을(를) /usr/sap/trans/cofiles(으)로 복사해야 함
  - Rnnnnnn.AWS을(를) /usr/sap/trans/data(으)로 복사해야 합니다.
- 전송을 가져올 때는 전송 요청 가져오기 > 옵션 > 가져오기 옵션에 있는 잘못된 구성 요소 버전 무시를 선택해야 합니다.
- 원하는 모든 전송을 동시에 가져올 수 있습니다.
- 전송을 개별적으로 가져오는 경우 core 전송을 먼저 가져와야 합니다.
- 모든 전송의 릴리스 수준은 동일해야 합니다.

가져오기 소요 시간

AWS SDK 전송을 가져오는 데 몇 분 정도 걸릴 수 있습니다. STMS에 녹색(RC=0) 또는 노란색(RC=4) 표시등이 표시되면 전송이 성공한 것입니다.

- 빨간색 표시등(RC=8)은 가져오기에서 구문 오류가 발생했음을 나타냅니다.
  - 요청 → 디스플레이 → 로그를 선택하여 가져오기 오류를 검사합니다.
  - 가져오는 동안 IF_SYSTEM_UUID_RFC4122_STATIC 인터페이스 누락으로 인해 오류가 발생하는 경우 SAP Note 2619546이 시스템에 적용되었는지 확인합니다. 자세한 내용은 참고를 참조하세요.
  - 오류의 원인을 알 수 없는 경우에 문의하십시오 지원.
- 빨간색 번개(RC=12)는 전송 파일이 /usr/sap/trans에 제대로 로드되지 않았거나 필요한 권한이 없음을 나타냅니다.

키 포인터

다음은 AWS SDK 전송을 가져올 때 사용되는 키 포인터입니다.

- 각 전송은 Knnnnnn.AWS 및 Rnnnnnn.AWS(으)로 제공됨
  - Knnnnnn.AWS을(를) /usr/sap/trans/cofiles(으)로 복사해야 함
  - Rnnnnnn.AWS을(를) /usr/sap/trans/data(으)로 복사해야 합니다.
- 전송을 가져올 때는 전송 요청 가져오기 > 옵션 > 가져오기 옵션에 있는 잘못된 구성 요소 버전 무시를 선택해야 합니다.
- 원하는 모든 전송을 동시에 가져올 수 있습니다.
- 전송을 개별적으로 가져오는 경우 core 전송을 먼저 가져와야 합니다.

- 모든 전송의 릴리스 수준은 동일해야 합니다.

가져오기 소요 시간

AWS SDK 전송을 가져오는 데 몇 분 정도 걸릴 수 있습니다. STMS에 녹색(RC=0) 또는 노란색(RC=4) 표시등이 표시되면 전송이 성공한 것입니다.

- 빨간색 표시등(RC=8)은 가져오기에서 구문 오류가 발생했음을 나타냅니다.
  - 요청 → 디스플레이 → 로그를 선택하여 가져오기 오류를 검사합니다.
  - 가져오는 동안 IF_SYSTEM_UUID_RFC4122_STATIC 인터페이스 누락으로 인해 오류가 발생하는 경우 SAP Note 2619546이 시스템에 적용되었는지 확인합니다. 자세한 내용은 참고를 참조하세요.
  - 오류의 원인을 알 수 없는 경우에 문의하십시오 지원.
- 빨간색 번개(RC=12)는 전송 파일이 /usr/sap/trans에 제대로 로드되지 않았거나 필요한 권한이 없음을 나타냅니다.

네임스페이스

SDK for SAP ABAP는 /AWS1/ 네임스페이스를 사용하며 다음과 같은 예외를 제외하고 시스템에 있는 SAP 객체 또는 기타 객체를 수정하지 않습니다.

- AWS auth 객체는 인증 객체 클래스에 있습니다. 인증 객체 클래스는 4자로 제한되며 네임스페이스를 지원하지 않습니다. SDK for SAP ABAP는 인증 객체 클래스는 YAW1을(를) 사용합니다. 트랜잭션 YAW1에 인증 객체 클래스가 이미 있는 경우 설치 지원 전에에 SU21문의하십시오.

AWS SDK for SAP ABAP 설치 - BTP 에디션

BTP 에디션은 개발자 미리 보기에 있으며 미리 보기에 참여하여 설치할 수 있습니다. SDK를 설치하려면 AWS SDK for SAP ABAP - BTP 에디션 개발자 미리 보기에서 참여 양식을 작성합니다.

SDK for SAP ABAP - BTP 에디션을 설치하기 전에 필수 사전 조건을 충족하는지 확인합니다. 자세한 내용은 SAP Landscape 포털 및 SAP 자격 증명 스토어를 참조하세요.

주제

- SDK for SAP ABAP 설치 - BTP 에디션
- 모듈

- SDK for SAP ABAP 패치 적용 - BTP 에디션

SDK for SAP ABAP 설치 - BTP 에디션

1. SAP Landscape 포털 인스턴스로 이동하여 제품 배포 fiori 애플리케이션을 시작합니다.
2. 제품에서 파트너 제품**/AWS1/SDK_OMNI**에서를 선택합니다.

   개발자 미리 보기에 수락된 */AWS1/SDK_OMNI* 후가 표시되지 지원 않으면에 문의하세요.
3. 대상 버전에서 시스템에 설치할 SAP ABAP용 SDK - BTP 에디션의 버전을 선택합니다.
4. 사용 가능한 시스템에서 SDK를 설치하려는 모든 SIDs의 확인란을 선택합니다.
5. 배포를 선택하고 일정 세부 정보를 입력한 다음 일정을 선택합니다. 제품 버전 배포 상태의 진행 상황을 모니터링할 수 있습니다.

   설치에는 30~45분이 걸릴 수 있으며 시스템 가동 중지 시간이 포함됩니다. 자세한 내용은 제품 배포를 참조하세요.

모듈

다음 모듈은 AWS SDK for SAP ABAP - BTP 에디션의 개발자 미리 보기에 포함되어 있습니다.

- Amazon API Gateway [agw]
- Amazon Athena [ath]
- Amazon Bedrock 런타임 [bdr]
- Amazon Comprehend [cpd]
- Amazon EventBridge [evb]
- Amazon Forecast [fcs]
- Amazon Kinesis [kns]
- Amazon Data Firehose [frh]
- Amazon SageMaker AI [sgm]
- Amazon Simple Notification Service [sns]
- Amazon Simple Queue Service [sqs]
- Amazon Simple Storage Service [s3]
- AWS Systems Manager [ssm]

- Amazon Textract [tex]
- Amazon Transcribe [tnb]
- Amazon Translate [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]
- IAM Roles Anywhere [rla]
- Amazon Redshift 데이터 API [rsd]

SDK for SAP ABAP 패치 적용 - BTP 에디션

SDK for SAP ABAP - BTP 에디션의 패치 적용 프로세스는 설치 프로세스와 유사합니다. 이전 버전이 이미 설치된 시스템에 SDK를 설치하면 선택한 새 버전에 SDK가 패치됩니다.

구성 AWS SDK for SAP ABAP

사용하기 전에 SDK 작업에 필요한 기술 및 기능 설정으로 SDK를 구성해야 AWS SDK for SAP ABAP 합니다. 일부 설정은 전송 가능하고 일부는 런타임 설정입니다. 대부분의 설정은 다른 SDK용 `.INI` 파일에 정의된 설정과 직접적으로 유사합니다.

런타임 설정을 제외한 SDK 구성은 개발 환경에서 완료해야 합니다. 일반적인 전송 및 변경 제어 규칙에 따라 구성을 QA 및 프로덕션으로 전송할 수 있습니다. 프로덕션 환경에는 전송 가능한 구성이 권장되지 않습니다.

AWS SDK를 구성할 권한이 없는 경우 SAP 권한 부여를 참조하세요.

구성 AWS SDK for SAP ABAP

구성 트랜잭션을 실행하려면 SAPGUI 명령 모음에 `/n/AWS1/IMG`을(를) 입력합니다.

AWS SDK for SAP ABAP 구성 - BTP 에디션

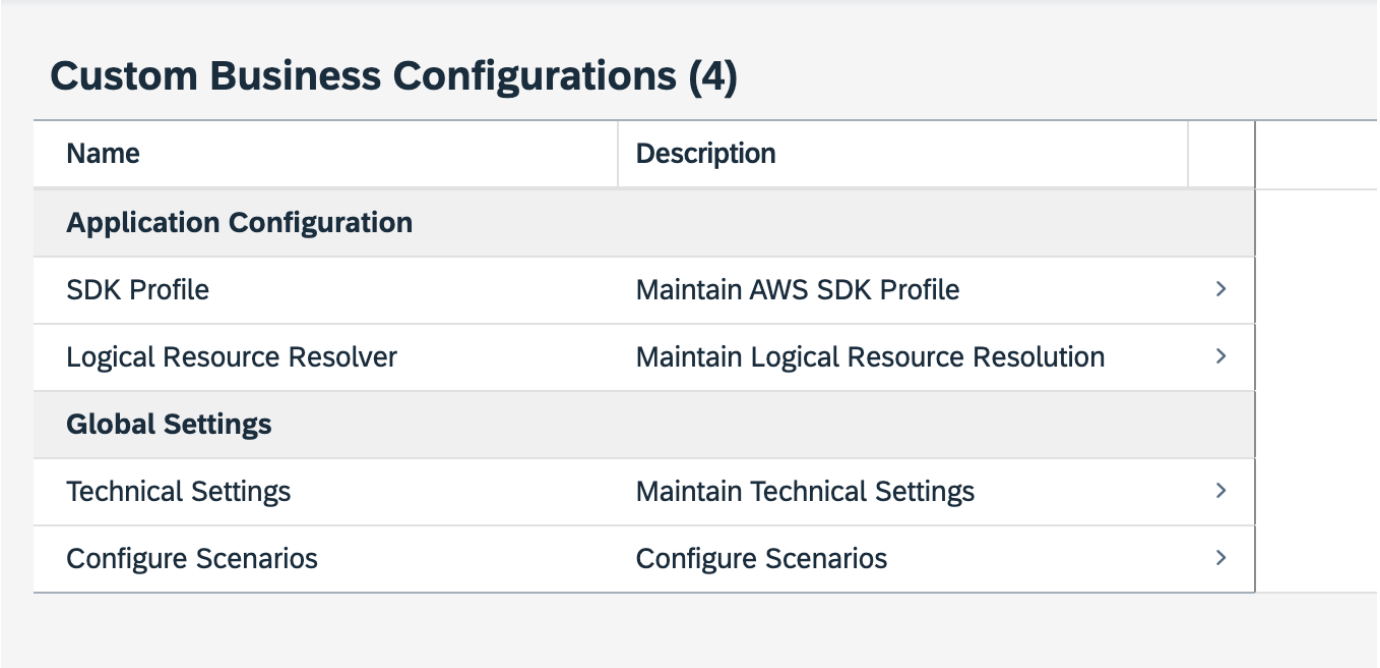
다음 단계에 따라 SDK for SAP ABAP - BTP 에디션을 구성합니다.

1. 웹 브라우저에서 ABAP 환경을 엽니다.
2. 사용자 지정 비즈니스 구성 애플리케이션으로 이동합니다.

Export Customizing Transports 애플리케이션을 사용하여 사용자 지정 요청을 생성하려면 Export Customizing Transports 앱 - 요청 생성을 참조하세요.

사용자 지정 비즈니스 구성 애플리케이션에서 SDK 설정 유형에 따라 구성을 그룹화할 수 있습니다. 다음 단계에 따라 구성을 그룹화합니다.

1. 웹 브라우저에서 ABAP 환경을 열고 사용자 지정 비즈니스 구성 애플리케이션으로 이동합니다.
2. 설정 > 그룹을 선택하고 드롭다운 목록에서 구성 그룹을 선택합니다. 확인을 선택합니다.
3. 이제 이미지에 표시된 대로 계층 구조로 구성을 사용할 수 있습니다. 뷰를 저장하려면 뷰(변형 관리) - 구성 요소를 참조하세요.



이 섹션은 다음 주제를 포함합니다.

주제

- 글로벌 설정
- 애플리케이션 구성
- 런타임 설정
- 고급 연결 시나리오
- 서비스 공급자 설정
- 에 대한 새로 고침, 추적 및 원격 측정 주제 AWS SDK for SAP ABAP

글로벌 설정

AWS SDK for SAP ABAP용 `/n/AWS1/IMG` IMG Transacation과 AWS SDK for SAP ABAP - BTP 에디션용 Custom Business Configuration 애플리케이션을 사용하여 글로벌 설정을 구성합니다. 이 주제에서는 IMG와 사용자 지정 비즈니스 구성을 상호 교환적으로 사용합니다.

이 섹션은 다음 주제를 포함합니다.

주제

- 기술 설정
- 시나리오 구성

기술 설정

/AWS1/IMG 트랜잭션의 글로벌 설정은 전체 SDK의 동작에 영향을 줍니다. 이러한 설정은 일반적으로 기본 관리자가 구성합니다. 이 값을 다음과 같은 권장 설정으로 설정할 수 있습니다.

- 새 항목을 선택합니다.
  - S3 지역화: s3.amazonaws.com을 사용하여 us-east-1 버킷에 액세스합니다.
  - STS 지역화: 글로벌 엔드포인트를 사용하여 STS에 액세스합니다.
  - EC2 메타데이터 비활성화: 이 필드를 비워 둡니다. 이 필드는 BTP 에디션에서 읽기 전용이며 기본적으로 '예'로 설정됩니다.
  - 메타데이터 엔드포인트 모드: IPv4 메타데이터 엔드포인트를 사용합니다. 이 필드는 BTP 에디션에서 읽기 전용이며 자동 업데이트됩니다.
  - 메타데이터 엔드포인트 URL: 이 필드를 비워 둡니다. 이 필드는 BTP 에디션에서 읽기 전용입니다.
- 저장을 선택합니다.

시나리오 구성

시나리오를 사용하면 AWS SDK가 다중 리전 재해 테스트 또는 재해 복구 테스트 시나리오 중에 설정을 보다 효율적으로 전환할 수 있습니다. 이 기능이 필요하지 않을 수도 있으며, 대신 다음과 같은 DEFAULT 시나리오를 구성하기만 하면 됩니다.

- 새 항목을 선택합니다.
  - 시나리오 ID: DEFAULT
  - 시나리오 설명: 기본 시나리오
- 저장을 선택합니다.

다중 리전 재해 복구 설정 또는 빠른 설정 변경이 필요한 기타 고유한 경우가 있는 경우 여러 시나리오를 구성할 수 있습니다.

- DEFAULT- 표준 작업.
- DR- 재해로 인해 전체 시스템을 다른 리전으로 이동해야 하는 경우를 대비한 특별 구성.
- DR_TEST- 임시 프로덕션 복제본과 같이 재해 시뮬레이션을 위한 특수 구성

애플리케이션 구성

SDK for SAP ABAP 구성은 다른 ABAP 기반 애플리케이션 구성과 유사합니다. 다양한 시나리오의 설정을 그룹화할 수 있도록 다양한 프로필로 구성되어 있습니다. ABAP SDK 프로필은 특정 애플리케이션 시나리오에 필요한 설정을 정의합니다. 예를 들어 트랜잭션 ZVA01, ZVA02및 ZVA03가 Amazon S3 AWS Lambda, 및 AWS 서비스와 같은 인보이스 관련 트랜잭션이 향상되고 실행되는 경우 Amazon SageMaker AI라는 SDK 프로파일을 만들 ZINVOICE 수 있습니다. 이 프로필은 인보이스 관련 기능에 대한 기술 설정, SAP 인증 및 IAM 역할 매핑을 그룹화할 수 있습니다.

AWS SDK for SAP ABAP의 /n/AWS1/IMG Transacation과 AWS SDK for SAP ABAP - BTP 에디션의 Custom Business Configuration 애플리케이션을 사용하여 전역 설정을 구성합니다. 이 주제에서는 IMG와 사용자 지정 비즈니스 구성을 상호 교환적으로 사용합니다.

주제

- SDK 프로필
- 논리적 리소스 해석기
- 예제

SDK 프로필

ABAP SDK 프로필은 각 SID 및 클라이언트에 대해 다음을 정의합니다.

> **Note**
>
> SAP BTP, ABAP 환경에서 클라이언트는 항상 100입니다.

- 모든 API 호출의 기본 AWS 리전입니다. 예를 들어 SAP 시스템이 us-east-1 리전에서 실행 중인 경우 다른 AWS 리소스도 동일한 리전에 있을 수 있으며 이는 기본 리전이어야 합니다. ABAP 코드는 기본 리전을 재정의할 수 있습니다.
- 인증 방법

  - Amazon EC2에서 실행되는 SAP 시스템의 경우, 단기 자동 순환 보안 인증을 활용하려면 인스턴스 역할 메타데이터를 선택하는 것이 좋습니다.
  - 온프레미스 또는 다른 클라우드에서 실행되는 SAP 시스템의 경우 SSF 스토리지에서 보안 인증을 선택해야 합니다.
  - SAP BTP에서 실행되는 ABAP 시스템의 경우 SAP 자격 증명 스토어에서 자격 증명을 선택해야 합니다. 자세한 내용은 인증에 SAP 자격 증명 스토어 사용을 참조하세요.
  - 교차 계정 역할 체인 시나리오의 경우 소스 프로필을 선택하고 소스 프로필 ID를 지정합니다. 이를 통해 역할 가정 체인의 자동 확인 및 실행이 가능합니다. 자세한 내용은 교차 계정 액세스를 위한 소스 프로파일 사용을 참조하세요.
- 논리적 IAM 역할을 IAM 역할에 매핑합니다.
  - 해당 매핑은 우선 순위 내림차순으로 정렬됩니다.
  - PFCG 역할에서 사용자에게 권한이 부여된 우선 순위가 가장 높은 IAM 역할이 자동으로 선택됩니다.
- 서비스를 사용자 지정 엔드포인트에 선택적으로 매핑합니다. 이 구성은 고급 연결 시나리오에서 설명합니다.

> **Note**
>
> PFGC 역할은 SAP BTP, ABAP 환경에서 비즈니스 역할이라고 합니다.

ABAP 프로그램이에 연결하려는 경우 필요한 설정을 가져오는 ABAP SDK 프로파일을 AWS 서비스지정합니다. 사용자가 SDK 프로필에 액세스할 권한이 있는지 확인하기 위해 `AUTHORIZATION-CHECK`이(가) 수행됩니다. SAP 보안 관리자는 적절한 사용자에게 액세스 권한을 부여하는 PFCG 역할을 정의할 수 있습니다.

논리적 리소스 해석기

논리적 리소스 해석기를 사용하면 표준 위치에 리소스 이름을 저장할 수 있습니다. SDK for SAP ABAP와 함께 제공됩니다. 이 동작은 `FILE` 트랜잭션이 논리적 파일 이름을 물리적 파일 이름에 매핑하는 방식과 유사합니다.

논리적 리소스는 인보이스가 들어 있는 Amazon S3 버킷과 같은 AWS 리소스의 개념을 정의합니다. 예를 들어, 이 논리적 리소스의 이름은 `ZINVOICES_OUTBOUND`일 수 있으며 SAP 시스템이 개발, QA 또는 프로덕션 단계인지에 따라 다른 물리적 버킷 이름에 매핑할 수 있습니다.

SDK for SAP ABAP는 프로덕션에서 시스템을 새로 고친 후에도 QA 시스템이 논리적 리소스를 QA 물리적 리소스로 해결하도록 설정됩니다. ALL 시스템의 리소스 매핑은 개발 SAP 시스템에서 정의되고 전달됩니다. 이 접근 방식은 매핑이 마스터 데이터로 처리되고 각 시스템에서 설정되는 SAP 시스템의 일반적인 설정과는 다릅니다. SDK for SAP ABAP에서 제공하는 논리적 리소스 해석기의 장점은 시스템을 새로 고친 후 잘못된 전송이 발생할 가능성이 거의 없다는 것입니다.

예제

Amazon S3 버킷은 총 네 개로 각각 개발, 프로덕션 및 QA용으로 하나씩 존재하며 두 번째 QA 버킷은 회귀 테스트용입니다.

SDK는 `ZINVOICE_OUTBOUND`과(와) 같은 논리적 리소스를 물리적 리소스로 해결하면 `SY-SYSID` 및 `SY-MANDT`을(를) 확인하여 실행 중인 SID와 클라이언트를 묻고 올바른 물리적 리소스를 자동으로 선택합니다.

프로덕션에서 리소스의 매핑을 변경해야 하는 경우 개발 시스템의 `IMG`에서 매핑을 변경하여 전달해야 합니다. 이렇게 하면 SAP 시스템에 AWS 리소스를 재할당할 때 다른 전송과 마찬가지로 변경 제어가 적용됩니다.

> **Note**
>
> SDK 구성은 클라이언트에 따라 달라지므로 리소스 재할당은 사용자 지정 요청으로 전송되며 전송을 각 클라이언트로 가져와야 합니다.

런타임 설정

이 섹션은 다음 주제를 포함합니다.

> **Note**
>
> 이러한 설정은 전송할 수 없으며 각 SAP 시스템에서 로컬로 설정됩니다.

주제

- 로그 및 추적
- 옵트인: 향상된 원격 측정

- 활성 시나리오

로그 및 추적

디버깅 용도로 추적을 활성화할 수 있습니다. 기술적인 문제를 진단하지 않는 한 추적 수준을 추적 없음으로 유지하는 것이 좋습니다. 자세한 정보는 보안 작업을 참조하세요.

이러한 설정은 SDK for SAP ABAP - BTP 에디션에는 적용되지 않습니다.

옵트인: 향상된 원격 측정

모든 SDKs 지원을 위해 AWS 에 원격 측정 정보를 전송합니다. 향상된 원격 측정을 옵트인할 수 있습니다. 이는 특정 API 호출의 소스를 식별하기 지원 위해에 문의할 때 특히 유용합니다. 자세한 내용은 추적 및 원격 측정을 참조하세요.

이러한 설정은 SDK for SAP ABAP - BTP 에디션에는 적용되지 않습니다.

활성 시나리오

이 트랜잭션에서 DEFAULT 시나리오를 활성화하세요. 이 활성화는 각 시스템에 한 번만 필요하며 시스템에 다중 리전 재해 복구가 진행 중인 경우가 아니면 변경해서는 안 됩니다. 다중 리전 설정에서 이 설정을 사용하여 SAP 시스템을 재해 복구 환경 또는 재해 복구 테스트 시나리오로 전환할 수 있습니다.

고급 연결 시나리오

AWS SDK for SAP ABAP 는 AWS 엔드포인트 AWS 서비스 에 HTTPS 호출을 수행하여를 사용합니다. 일반적으로 AWS 엔드포인트는 인터넷을 통해 액세스할 수 있습니다. SAP 시스템은 인터넷에 접속하여 이러한 아웃바운드 연결을 설정할 수 있어야 합니다. SDK for SAP ABAP는 인터넷에서 SAP 시스템으로의 인바운드 연결을 필요로 하지 않습니다.

다음 시나리오는 아웃바운드 연결을 설정하는 다양한 방법을 제공합니다.

시나리오

- 프록시 서버를 통한 연결
- 패킷 검사 방화벽을 통한 연결
- 게이트웨이 엔드포인트
- 사용자 지정 인터페이스 엔드포인트

- 고급 라우팅
- 다중 리전 엔드포인트에 액세스

프록시 서버를 통한 연결

프록시 서버를 통해 연결을 설정하려면 다음 단계를 수행합니다.

1. SDK에서 **SICF** 트랜잭션으로 이동합니다.
2. 실행을 선택합니다.
3. 메뉴에서 클라이언트 > 프록시 서버를 선택합니다.
4. 프록시 설정을 활성으로 설정합니다.
5. 다음 주소의 프록시 없음 필드에 모든 예외를 세미콜론으로 구분하여 나열합니다.
6. HTTP 프로토콜 및 HTTPS 프로토콜 필드에 프록시 서버의 연결 세부 정보를 지정합니다.

SDK는 프록시 서버를 인식하지 않으므로 SAP 시스템의 프록시 서버 구성을 사용하기 위한 설정이 필요하지 않습니다.

> **Note**
>
> Amazon EC2 인스턴스 메타데이터 인증을 사용하는 경우 SAP 시스템은 프록시 서버를 사용하여 `http://169.254.169.254`의 로컬 인스턴스 메타데이터에 액세스할 수 없습니다. 다음 주소의 프록시 없음 필드에 `169.254.169.254`을(를) 포함해야 합니다.

패킷 검사 방화벽을 통한 연결

아웃바운드 연결을 위한 패킷 검사 방화벽을 구성할 수 있습니다. 이러한 방화벽은 SSL 트래픽을 해독한 다음 엔드포인트로 전달하기 전에 다시 암호화합니다. 이 구성에서는 일반적으로 방화벽이 AWS 서비스을(를) 사용하는 SAP 시스템에 자체 인증서를 발급해야 합니다. STRUST에 방화벽의 CA 인증서를 설치해야 합니다. 자세한 내용은 HTTPS 연결을 참조하세요.

게이트웨이 엔드포인트

일부는 인터넷 없이 VPC에 고성능 액세스를 제공하기 위해 게이트웨이 엔드포인트를 AWS 서비스 제공합니다. 이러한 엔드포인트는 SDK for SAP ABAP에 투명하며 구성이 필요하지 않습니다.

자세한 내용은 게이트웨이 엔드포인트를 참조하세요.

사용자 지정 인터페이스 엔드포인트

기본 엔드포인트 확인을 사용자 지정 엔드포인트로 재정의해야 하는 경우 인터페이스 엔드포인트를 사용하여 인터넷 없이 VPC에 고성능 액세스를 제공할 수 있습니다. 자세한 내용은 인터페이스 엔드포인트 구성을 참조하세요.

프라이빗 DNS를 사용하지 않는 경우 이러한 엔드포인트는 자체 DNS 주소를 가지므로 ABAP 프로그램은 일반적인 엔드포인트 확인 로직을 명시적으로 재정의해야 합니다. 자세한 내용은 AWS re:Post - 인터페이스 VPC 엔드포인트의 서비스 도메인 이름을 확인할 수 없는 이유는 무엇입니까?를 참조하세요.

다음 예제에서는 AWS STS 및에 대한 인터페이스 엔드포인트가 생성됩니다 Amazon Translate. SAP 시스템은 프라이빗 DNS를 사용하지 않으며 사용자 지정 엔드포인트로 서비스를 호출합니다. `/AWS1/IMG`에 정의된 논리적 리소스는 `vpce-0123456789abcdef-hd52vxz.translate.us-west-2.vpce.amazonaws.com`와 같은 물리적 인터페이스 엔드포인트 주소를 나타냅니다. 이렇게 하면 코드에 DNS를 하드 코딩하지 않아도 됩니다.

다음 코드에서는 `/AWS1/IMG`의 논리적 리소스가 먼저 물리적 엔드포인트 이름으로 확인됩니다. 그런 다음 AWS 세션 클래스(를 사용하여 IAM 역할을 수임)의 팩토리 메서드 AWS STS 에 제공되고 API 클래스를 번역합니다.

```
" 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...
```

예제에서 볼 수 있듯이go_xl8에 대한 모든 메서드 호출은 엔드포인트 `https://vpce-0123456789abcdefg-12345567.translate.us-west-2.vpce.amazonaws.com`로 이동합니다. 다음 섹션에 표시된 대로 코드 대신 IMG 구성에서 라우팅 사용자 지정 엔드포인트를 정의할 수도 있습니다.

고급 라우팅

이전 섹션에서는 SDK 모듈의 팩토리 메서드 `iv_custom_endpoint` 인수에 사용자 지정 엔드포인트를 지정하는 방법을 보여드렸습니다. SDK를 사용하는 ABAP 프로그램 수가 증가하면 관리하기 어려울 수 있습니다. SDK 프로파일에서에서 사용자 지정 엔드포인트 AWS 서비스 로의 매핑을 구성할 수 있습니다. 각 SID, 클라이언트 및 시나리오에 대해 서비스 TLA(3자 약어)를 엔드포인트 URL에 매핑할 수 있습니다.

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

이 구성을 사용하면 공장 메서드 호출`iv_custom_endpoint`에서를 지정할 필요가 없습니다. 사용자 지정 엔드포인트는 구성 테이블에서 자동으로 선택됩니다. 구성은 SDK 프로파일에 고유하므로 필요에 따라 서로 다른 라우팅으로 여러 프로파일을 생성할 수 있습니다. 다른 SDK 프로파일 구성과 마찬

가지로 라우팅은 SID 및 클라이언트별로 다르므로 다른 시스템에 대해 별도의 라우팅을 정의할 수 있습니다.

다중 리전 엔드포인트에 액세스

AWS 엔드포인트는 SDK 프로파일에 AWS 리전 정의된 기본값에서 자동으로 결정됩니다. 프로그래밍 방식으로 지역을 지정하여 기본 리전을 재정의할 수도 있습니다. 이는 팩토리 CREATE( ) 메서드에서 재정의하거나 나중에 SDK의 구성 객체로 재정의할 수 있습니다. 자세한 내용은 프로그래밍 방식 구성을 참조하세요.

다음 예시에서는 팩토리 CREATE( ) 메서드를 사용하여 리전을 설정하고 us-east-1 및us-west-2 리전 모두에서 Amazon SQS 대기열을 나열합니다.

```
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.
```

서비스 공급자 설정

기본 관리자는 클라이언트에서 전체 시스템 전반에서 SDK의 특정 기능을 제어해야 하는 경우가 있습니다000. 이는 고객을 대신하여 자신의 AWS 계정에서 시스템을 운영하는 호스팅 및 서비스 공급자를 위한 일반적인 시나리오입니다. AWS SDK for SAP ABAP는 서비스 공급자 설정을 지원합니다. 이러한 설정은 클라이언트에서 구성000되며 모든 클라이언트의 SDK에 영향을 미칩니다. 서비스 공급자 설정은 SDK for SAP ABAP - BTP 에디션에서 지원되지 않습니다.

서비스 공급자 설정은 트랜잭션에서 구성/AWS1/IMG되며 클라이언트에서 구성되어야 합니다000. 다른 클라이언트의 서비스 공급자 설정은 무시됩니다. 클라이언트의 설정은 모든 클라이언트에 000 적용되며 충돌 시 다른 IMG 설정을 대체합니다.

다음 단계에 따라 클라이언트에서 서비스 공급자 설정을 구성합니다000.

1. 트랜잭션에서 서비스 공급자 설정 브랜치를 확장합니다/AWS1/IMG.
2. 서비스 공급자 가드레일 선택
3. 새 항목을 선택하고 비즈니스 요구 사항에 따라 설정을 조정합니다.
    - EC2 메타데이터 비활성화 - SDK 프로파일이 EC2 인스턴스 메타데이터를 사용하여 인증하도록 구성된 경우에도 SDK가 모든 클라이언트의 EC2 인스턴스 메타데이터에 액세스하지 못하도록 합니다. ABAP 프로그램이 SDK를 사용하여 인스턴스 메타데이터에 액세스하려고 하면 SDK에서 예외가 발생합니다.
4. 저장을 선택합니다.

에 대한 새로 고침, 추적 및 원격 측정 주제 AWS SDK for SAP ABAP

이 섹션은 다음 주제를 포함합니다.

주제

- SAP 시스템 새로 고침
- 추적
- 원격 측정

SAP 시스템 새로 고침

시스템을 새로 고친 후 기본 관리자의 주요 과제는 개별 시스템이 서로의 리소스에 액세스하지 못하도록 하는 것입니다. 예를 들어 QA SAP 시스템이 프로덕션 환경의 S3 버킷과 같은 리소스에 액세스하지 못하도록 할 수 있습니다.

SDK for SAP ABAP는 이러한 문제에 대한 논리적 리소스의 안전을 고려한 접근 방식을 제공합니다. 비즈니스 분석가는 다음 단계를 수행할 수 있습니다.

1. ZINVOICE_OUTBOUND과(와) 같은 논리적 리소스를 정의합니다.

2. 개발 시스템의 모든 시스템과 클라이언트를 매핑합니다.
3. 모든 시스템의 구성을 프로덕션 환경까지 전달합니다.

새로 고침 후의 기본 단계

1. 인증 확인

- 시스템에서 비밀 액세스 키 인증을 사용하는 경우 SSF로 암호화된 보안 인증은 마스터 데이터에 저장되므로 유효하지 않습니다. 보안 인증을 다시 입력해야 하며, 이 경우 https://console.aws.amazon.com/iam/에서 새 비밀 액세스 키를 다시 생성해야 할 수 있습니다.
- 시스템이 EC2 인스턴스 메타데이터로 인증하는 경우 별도의 조치가 필요하지 않습니다.

추적 설정 확인

- `/AWS1/IMG`에서 추적 설정이 원하는 설정인지 확인합니다. 이러한 설정은 전송할 수 없습니다.

추적

추적 출력은 IMG 런타임 설정에서 제어됩니다.

사용할 수 있는 추적 수준은 다음과 같습니다.

- 추적 없음
- API 호출 추적
- API 호출 및 페이로드 추적

  이 옵션에는 암호화되지 않은 페이로드 정보가 포함되어 있습니다.
- API 호출, 페이로드, 내부 XML 변환 추적

  이 옵션에는 암호화되지 않은 페이로드 정보가 포함되어 있습니다.

API 추적이 활성화되면 추적이 `aws1_trace-`*`YYYY-MM-DD`*`.log` 파일의 `DIR_WORK`에 기록됩니다.

페이로드 추적이 추가로 활성화되면 각 호출 및 페이로드 구성 요소에 대해 `aws1_payload_*`라는 제목의 추가 파일이 생성됩니다. 각 개별 페이로드 추적 실패에 적용되는 길이 제한으로 페이로드 추적 길이를 제한할 수 있습니다.

페이로드 추적은 주로 직렬화 오류가 발생할 지원 경우에 제공할 정보를 수집하기 위한 것입니다. SDK 오류를 진단하려는 경우가 아니라면 추적 없음을 선택하는 것이 좋습니다.

> **Note**
>
> 페이로드 추적에는 암호화되지 않은 비즈니스 정보가 포함될 수 있습니다. 문제 해결에 도움이 되도록 AWS Support의 요청에 대해서만 이러한 추적을 켜는 것이 좋습니다. 문제가 해결된 후에는 이러한 추적을 끌 수 있습니다. 추적은 자동으로 삭제되지 않으므로 더 이상 필요하지 않을 경우 시스템 관리자가 추적을 제거해야 합니다.

이러한 설정은 SDK for SAP ABAP - BTP 에디션에는 적용되지 않습니다.

원격 측정

SDKs 원격 측정 정보를 로 전송합니다 지원. SDK for SAP ABAP는 다음과 같은 정보를 수집합니다.

- OS 릴리스 및 패치 수준
- SAP_BASIS 릴리스 및 패치 수준
- SAP 커널 릴리스 및 패치 수준

다음 정보를 지원(으)로 전송하도록 옵트인할 수 있습니다.

- SAP SID 및 인스턴스 이름(host_sid_nn)
- SAP 클라이언트(SY-MANDT)
- 트랜잭션 코드(SY-TCODE) 및 보고서(SY-REPID)

추가 정보를 통해 지원 가 더 나은 도움을 줄 수 있습니다. 지원 는 특정 API 호출이 이루어진 이유를 감지하고 SAP 시스템에서 관련 트랜잭션을 추가로 찾을 수 있습니다.

원격 측정은 SDK for SAP ABAP - BTP 에디션용 SDK 및 API 버전으로 제한됩니다.

사용 AWS SDK for SAP ABAP

SDK for SAP ABAP에는 두 가지 주요 구성 요소가 있습니다.

- SDK 런타임(패키지/AWS1/RT) — 보안, 인증, 추적, 구성, 데이터 변환 및 기타 API 간 기능을 뒷받침하는 객체 세트입니다. Amazon S3, AWS STS IAM Roles Anywhere 및 Secrets Manager에 대한 API 모듈은 필수입니다.
- API(패키지 /AWS1/API 및 하위 패키지) — 각 API의 개체가 서로 완전히 독립적인 각 API의 하위 패키지로, 한 API의 변경으로 인해 다른 API가 손상되지 않습니다. 전체 AWS SDK for SAP ABAP API 목록을 보려면 AWS SDK for SAP ABAP - API 참조 안내서를 참조하세요.

이 섹션은 다음 주제를 포함합니다.

주제

- ABAP에서의 데이터 표시
- Amazon S3 예제 프로그램
- SDK for SAP ABAP 개념
- AWS SDK for SAP ABAP features
- SDK로 제품 구축
- HTTP 요청을 로 사용자 지정 AWS
- 제한 사항

ABAP에서의 데이터 표시

이 섹션은 다음 주제를 포함합니다.

주제

- 데이터 타입
- AWS 데이터 형식

데이터 타입

AWS 서비스 에는 ABAP 데이터 형식에 매핑해야 하는 표준 데이터 형식 세트가 있습니다. 자세한 내용은 다음 표를 참조하세요.

AWS 데이터 유형	ABAP 데이터 유형	설명
boolean	C	단일 문자 "X" 및 " "
String	STRING	
바이트	INT2	INT2의 범위는 0~255보다 큽니다. 대부분은 오버플로를 AWS 서비스 잘라내지만이 동작은 공식적으로 정의되지 않습니다.
Short	INT2	
Integer	INT4	
Long	DEC19	ABAP 750까지 INT8을 사용할 수 있습니다. DEC19는 지원되는 모든 ABAP 플랫폼에서 호환성과 일관성을 유지하는 데 사용됩니다.
Blob	XSTRING	이진 데이터 표시
Float	STRING	ABAP는 DECFLOATs 지원하지만 NaN, Infinity 및 -Infinity와 같은 값을 나타낼 수 없습니다. AWS SDK는 내부적으로 이러한 값을 STRINGs으로 나타내며 런타임 시 이를 DECFLOAT16으로 변환합니다. NaN, Infinity 또는 +Infinity가 표시되는 경우 개발자는 특수한 예외 세트 또는 매핑에 대한 응답으로 이를 처리할 수 있습니다.
배정밀도 실수	STRING	
bigInteger	STRING	이 값은 ABAP로 표현할 수 없는 무한 길이의 숫자를 나타내

AWS 데이터 유형	ABAP 데이터 유형	설명
bigDecimal	STRING	며 bigInteger 대신 STRING이 사용됩니다.
Timestamp	TZNTSTMPS	TZNTSTMPS 을(를) 사용하면 기본 ABAP 타임스탬프 함수를 사용하여 처리할 수 있습니다.

AWS 서비스 는 다음과 같은 집계 데이터 형식도 반환합니다.

AWS 데이터 유형	ABAP 데이터 유형	설명
구조	Class	
결합	Class	결합에는 필드 세트를 두 개 이상 가질 수 없다는 점을 제외하고는 구조와 동일합니다. 다른 모든 필드는 값 없음으로 설정됩니다.
Array	STANDARD TABLE	
해시	HASHED TABLE	해시 테이블에는 KEY(문자열)와 VALUE(클래스)라는 두 개의 열만 있습니다.

AWS 데이터 형식

ABAP AWS 서비스 에서를 지원하기 위해 다음 접근 방식이 통합되었습니다.

- 특정 AWS 데이터 형식은 ABAP로 표현할 수 없습니다. 예를 들어, ABAP의 `float` 데이터 유형은 `NaN`, `Infinity` 또는 `-Infinity` 값을 지원하지 않습니다. 따라서 `float` 데이터 유형은 `STRING`(으)로 표시되며 런타임 시 `DECFLOAT16`(으)로 번역됩니다.
- AWS 데이터는 유선에 JSON 또는 XML로 표시되며 값은 선택 사항입니다. 예를 들어에서 JSON AWS 서비스 으로 반환한 다음 예제를 참조하세요.

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

Ana에 중간 이름이 없는 경우 서비스는 다음 출력을 반환합니다.

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

ABAP는 길이가 0인 문자열과 값이 없는 문자열을 구분하지 않습니다. 다른 언어에서는 문자열에 NULL 값을 할당하거나 문자열을 구문으로 래핑할 수 있습니다(예: Java의 `Optional<>` 래퍼). ABAP에서는 지원되지 않습니다. 따라서 SDK for SAP ABAP에서는 getter 메서드의 변형을 제공하여 값을 쉽게 구분할 수 있도록 합니다.

Amazon S3 예제 프로그램

이 섹션에서는 `ListObjectsV2`를 호출하여 Amazon S3 버킷의 콘텐츠를 나열하는 간단한 예제 프로그램을 안내합니다.

주제

- 사전 조건
- 코드
- 코드 섹션

사전 조건

이 예제 프로그램을 실행하려면 다음 사전 조건을 충족해야 합니다.

- Amazon S3 버킷이 있습니다. 이 자습서에서 버킷의 이름은 `demo-invoices.customer.com`입니다.

- 트랜잭션 /AWS1/IMG:
  - 이름이 DEMO_S3인 정의된 SDK 프로필이 있습니다.
    - SDK 프로파일에서 논리적 IAM 역할 TESTUSER은(는) Amazon S3 버킷의 콘텐츠를 나열할 s3:ListBucket 권한을 부여하는 arn:aws:iam::*111122223333*:role/SapDemoFinance와 같은 IAM 역할에 매핑되어야 합니다.
  - SAP 시스템의 SID 및 클라이언트를 사용하여 Amazon S3 버킷에 매핑된 DEMO_BUCKET이라는 논리적 리소스가 있습니다.
- 사용자의 PFCG 역할은 다음과 같습니다.
  - 사용자가 인증 객체 /AWS1/SESS을(를) 통해 DEMO_S3 SDK 프로필에 액세스할 수 있도록 권한을 부여합니다.
  - 인증 객체 /AWS1/LROL을(를) 통해 사용자에게 논리적 IAM 역할 TESTUSER 액세스 권한을 부여합니다.
- SAP 시스템은 SDK 프로파일에 정의된 방법을 AWS 사용하여에 대해 자체 인증할 수 있습니다.
- Amazon EC2 인스턴스 프로파일은 SDK 프로필에 매핑된 IAM 역할 arn:aws:iam::*111122223333*:role/SapDemoFinance에 대한 sts:assumeRole 권한을 SAP 시스템에 부여합니다.

코드

다음 코드 블록은 코드가 어떻게 보이는지 보여줍니다.

```
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.
```

코드 섹션

다음은 섹션별 코드를 검토한 것입니다.

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

사용자는 물리적 버킷 이름을 지정할 수 없습니다. 논리적 버킷을 지정하고 시스템 관리자(특히 비즈니스 분석가)는 AWS 관리자와 협력하여 논리적 버킷을 /AWS1/IMG의 물리적 버킷에 매핑합니다. 대부분의 비즈니스 시나리오에서 사용자는 논리적 버킷을 선택할 기회가 없습니다. 논리적 리소스 ID는 코드에 하드 코딩되어 있거나 사용자 지정 구성 테이블에 구성되어 있기 때문입니다.

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

이 행은 보안 세션을 설정하고 이 ABAP 프로그램이 DEMO_S3 SDK 프로필을 사용할 것으로 예상한다고 선언합니다. 이 호출은 SDK 구성에 대한 연결이며 기본 리전, 인증 설정 및 원하는 IAM 역할을 가져옵니다. 권한 부여 객체 /AWS1/SESS가 충족되었는지 확인하기 위해 AUTHORIZATION-CHECK(이)가 자동으로 호출됩니다. 또한 권한 부여 객체 /AWS1/LROL를 기반으로 사용자에게 권한이 부여되는 가장 강력한(하위 시퀀스 번호) 논리적 IAM 역할을 결정하기 위한 AUTHORIZATION-CHECK 호출이 이루어집니다. SDK는 IAM 역할이 SID 및 클라이언트의 논리적 IAM 역할에 매핑된 것으로 가정합니다. 그러면 세션 개체가 IMG의 추적 설정을 기반으로 추적을 활성화합니다.

사용자에게 요청된 SDK 프로필 또는 사용 가능한 논리적 IAM 역할에 대한 권한이 없는 경우 예외가 발생합니다.

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

이 행은 논리적 리소스를 물리적 버킷 이름으로 해결합니다. 구성에 이 SID/클라이언트 조합에 대한 매핑이 없어서 논리적 리소스를 확인할 수 없는 경우 예외가 발생합니다.

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

이 행은 /aws1/cl_s3_factory의 create( ) 메서드를 사용하여 Amazon S3용 API 객체를 생성합니다. 반환된 객체는 Amazon S3 API의 인터페이스 /aws1/if_s3 유형입니다. 각 서비스에 대해 별도의 API 객체를 생성해야 합니다. 예를 들어 ABAP 프로그램이 Amazon S3 AWS Lambda, 및 DynamoDB를 사용하는 경우 , /aws1/cl_lmd_factory및 /aws1/cl_s3_factory에서 API 객체를 생성합니다/aws1/cl_dyn_factory.

IMG(으)로 구성된 기본 리전을 재정의하려는 경우 생성자에는 지역을 지정할 수 있는 몇 가지 선택적 파라미터가 존재합니다. 이렇게 하면 한 리전의 버킷에서 다른 리전의 버킷으로 객체를 복사하려는 경우 /aws1/if_s3 인스턴스가 두 개(하나는 us-east-1용, 하나는 us-west-2용) 있을 수 있습니다. 마찬가지로, 재무 관련 버킷에서 읽고 물류 관련 버킷에 객체를 쓰는 보고서가 필요한 경우 두 개의 서로 다른 보안 세션 객체를 생성하여 두 개의 개별 /aws1/cl_s3 인스턴스를 만들 수 있습니다.

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

이 행은 ListObjectsV2에 대한 호출입니다. 간단한 입력 인수가 필요하며 단일 객체를 반환합니다. 이러한 객체는 ABAP 객체 지향 구조로 역직렬화된 심층 JSON 및 XML 데이터를 나타낼 수 있습니다. 경우에 따라 상당히 복잡할 수 있습니다. 이제는 출력을 처리하여 버킷의 내용을 나열하기만 하면 됩니다.

```
      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.
```

데이터의 내부 표현을 숨기는 GET...() 스타일 메서드를 사용하여 데이터에 액세스합니다. GET_CONTENTS( )은(는) ABAP 테이블을 반환하고 각 행 자체에는 단일 Amazon S3 항목을 나타내는 객체가 포함됩니다. 대부분의 경우 AWS SDK는이 객체 지향 접근 방식을 취하며 모든 데이터는 객체 및 테이블로 표시됩니다. LastModified 필드는 ABAP 네이티브 CONVERT TIME STAMP 명령을 사용하여 날짜로 변환할 수 있는 타임스탬프로 표시됩니다.는 쉬운 수학 및 형식 지정 작업을 INT4 위해를 GET_SIZE() 반환합니다.

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

연결, 4xx 클라이언트, 5xx 서버, 권한 부여 또는 구성 오류 등의 모든 ABAP 오류는 예외로 표시됩니다. 각 예외를 개별적으로 해결할 수 있습니다. 예외를 정보 오류로 처리할지, 재시도, 경고, 간단한 덤프 또는 기타 처리로 처리할지 여부를 선택할 수 있습니다.

SDK for SAP ABAP 개념

이 섹션에서는의 기본 개념을 다룹니다 AWS SDK for SAP ABAP.

주제

- API 클래스
- 추가 객체
- 구조 클래스
- 배열
- 맵
- 상위 수준 함수

API 클래스

각 에는 세 글자의 약어 또는가 할당 AWS 서비스 됩니다TLA. 서비스는 /AWS1/IF_<TLA> 형식의 인터페이스로 표시됩니다. 이를 서비스 인터페이스라고 부릅니다. API 클래스는 /AWS1/API_<TLA> 패키지에 있습니다. 서비스 인터페이스는 각 AWS 작업에 대해 하나의 메서드로 구성됩니다(이러한 메서

드를 작업 메서드라고 함). AWS SDK for SAP ABAP TLAs의 전체 모듈 목록을 보려면 AWS SDK for SAP ABAP - 모듈 목록을 참조하세요.

각 작업 메서드에는 몇 개의 IMPORTING 인수가 있으며 최대 하나의 RETURNING 인수가 있습니다. 이러한 인수는 생성자가 복잡하고 긴 GET…( ) 메서드 세트가 포함된 객체인 경우가 많습니다. 대부분의 경우 객체에는 중첩된 객체, 재귀 참조, 객체 테이블, 표 등이 포함됩니다. 이는 AWS 서비스 가 딥 XML 및 JSON 구조를 전달하기 때문에 플랫 인수 세트로 표현할 수 없습니다.

클래스에 단일 속성만 포함되어 있더라도 RETURNING 인수는 항상 클래스입니다.

추가 객체

각 API 패키지에는 기본 API 클래스를 포함하는 것 외에도 다양한 관련 리포지토리 및 데이터 사전 객체가 포함되어 있습니다.

- 각 구조 유형 객체의 클래스입니다.
- 표에 나타나는 모든 기본 데이터 유형의 클래스입니다. 예를 들어 서비스가 문자열 표를 반환하면 ABAP API는 이를 객체 표로 표현합니다. 여기서 각 객체는 문자열을 캡슐화하는 래퍼 클래스입니다. 이는 ABAP에서 기본적으로 표현할 수 없는 null 문자열을 나타내는 세부 정보를 래퍼 클래스에서 숨길 수 있도록 하기 위한 것입니다.
- 서비스에서 정의한 특정 오류에 대한 예외 클래스입니다.
- 각 기본 데이터 유형의 데이터 요소입니다. 각 데이터 유형에는 자체 문서화가 가능하도록 자체 데이터 요소가 있습니다.
- XML 및 JSON 페이로드를 직렬화 및 역직렬화하기 위한 XSLT 변환과 같은 내부 처리를 위한 추가 객체입니다.

구조 클래스

서비스에서 전송 및 수신하는 대부분의 AWS 데이터는 클래스로 AWS SDK에 의해 표현됩니다. 이러한 클래스는 데이터 구조를 나타내며 스토리지의 내부 세부 정보를 숨깁니다. 특히 클래스는 SDK가 값이 없는 필드를 나타내는 방식을 숨깁니다.

구조 클래스의 각 필드에는 세 가지 메서드가 있습니다.

GET_field( )

GET_field( ) 메서드

- 필드 값 반환 또는

- 필드에 값이 없는 경우에는 기본값을 반환하며, 이 기본값을 선택적 매개 변수로 설정할 수 있습니다.

예를 들어 버킷의 위치 제약 조건을 인쇄하는 다음 코드를 고려해 보세요.

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

us-east-1의 경우처럼 버킷에 위치 제약이 전혀 없는 경우 GET_LOCATIONCONSTRAINT( ) 는 빈 문자열을 반환합니다. 필드에 값이 전혀 없는 경우 이 동작을 재정의하고 원하는 값을 지정할 수 있습니다.

```
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' ).
```

이제 getbucketlocation( )의 결과가 위치를 반환하지 않는 경우 프로그램은 Bucket Location: assuming us-east-1을(를) 기록합니다.

요청된 값이 완전히 누락된 경우 GET() 메서드에 특정 결과를 반환하도록 요청할 수 있습니다. 다음 코드 예제를 참조하십시오.

```
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'  ).
```

이 경우 위치 제약이 없으면 GET_LOCATIONCONSTRAINT( )은(는) NopeNopeNope을(를) 반환합니다.

HAS_field( )

HAS_field( ) 메서드는 필드에 값이 있는지 여부를 확인하는 방법입니다. 다음 예를 참조하세요.

```
if NOT lo_location->HAS_LOCATIONCONSTRAINT( ).
```

```
   write: / 'There is no location constraint'.
endif.
```

특정 필드에 항상 값이 있는 것으로 알려진 경우에는 HAS_field( ) 메서드가 없습니다.

ASK_field( )

ASK_field( ) 메서드는 필드 값을 반환하거나 값이 없는 경우 예외를 발생시킵니다. 이 방법은 여러 필드를 처리하고, 값이 없는 필드에서 로직을 없애고 다른 접근 방식을 취하는 편리한 방법입니다.

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

참고로 /AWS1/CX_RT_VALUE_MISSING은(는) 정적 예외이므로 이를 포착하지 않기로 선택하면 경고가 표시됩니다.

모범 사례

일반적으로 GET_field( ) 메서드는 null 문자열을 빈 문자열로 취급하며며 세 가지 옵션 중 ABAP와 가장 유사하므로 사용할 수 있습니다. 하지만 이렇게 하면 필드에 빈 값이 있는 상황과 필드에 값이 없는 상황을 쉽게 구분할 수 없습니다. 비즈니스 로직이 누락된 데이터와 빈 데이터를 구분하는 데 의존하는 경우 HAS 또는 ASK 메서드를 사용하여 이러한 경우를 처리할 수 있습니다.

배열

배열은 ABAP 표준 객체 표로 표시됩니다.

JSON 배열에는 다음 [‘cat’, ‘dog’, null, ‘horse’] 배열과 같은 null 값이 포함될 수 있습니다. 이를 스파스 배열이라고 합니다. ABAP에서는 객체 참조의 내부 표로 표시되며, null 값은 테이블에 실제 ABAP null값으로 표시됩니다. 스파스 테이블을 반복할 때는 null 객체에 액세스하여 CX_SY_REF_IS_INITIAL 예외가 발생하지 않도록 null 값을 확인해야 합니다. 실제로 AWS 서비스에서 스파스 배열은 거의 사용되지 않습니다.

객체 배열을 초기화하려면 새 ABAP 7.40 구성을 사용하면 편리합니다. 다음과 같이 여러 보안 그룹이 할당된 Amazon EC2 인스턴스의 시작을 고려해 보십시오.

```
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' )
)
```

맵

JSON 맵은 ABAP에서 Hashed Tables(으)로 표시되며 각 테이블 행에는 구성 요소가 두 개만 존재합니다.

- KEY – 테이블의 UNIQUE KEY 문자열입니다.
- VALUE— 값을 포함하는 객체입니다.

맵은 AWS SDK에서 클래스 대신 실제 구조를 사용하는 몇 안 되는 사례 중 하나입니다. 이는 ABAP 해시 테이블이 객체 참조를 키 필드로 가질 수 없고 AWS 맵 키는 항상 null이 아닌 문자열이기 때문에 필요합니다.

상위 수준 함수

이전 섹션에서 API 클래스 설명한는 AWS 서비스 APIs하고 해당 APIs 익숙한 ABAP 클래스로 나타냅니다. 경우에 따라 SDK에는 특정 작업을 간소화하기 위해 API 클래스를 기반으로 빌드되는 상위 수준 함수도 포함되어 있습니다. 상위 수준 함수는 프로그래머 편의를 위해 포함되며 하위 수준 API 클래스를 대체하지 않습니다.

SDK에 모듈에 대한 상위 수준 함수가 포함된 경우 해당 함수는 동일한 전송에 포함되며 라는 팩토리 클래스를 통해 액세스할 수 있습니다/AWS1/CL_TLA_L2_FACTORY. 팩토리 클래스에는 API 설명서와 함께 나머지 API와 함께 문서화된 모듈에 대한 다양한 상위 수준 클라이언트를 생성하는 메서드가 포함되어 있습니다.

AWS SDK for SAP ABAP features

AWS SDK for SAP ABAP 는 다음과 같은 기능을 제공합니다.

주제

- 프로그래밍 방식 구성
- Waiters
- 페이지네이터
- 재시도 동작
- 프리서명자
- 교차 계정 IAM 역할 체인

프로그래밍 방식 구성

프로그래밍 방식 구성을 위해 AWS SDK for SAP ABAP용 `/n/AWS1/IMG` IMG Transacation과 AWS SDK for SAP ABAP - BTP 에디션용 Custom Business Configuration 애플리케이션을 사용합니다.

프로그래밍 구성을 시작하려면 먼저 `get_config( )` 명령을 사용하여 구성 객체를 검색하세요.

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

각 구성 개체는 `IMG`에 해당하는 `GETters` 및 `SETters`를 포함하는 `/AWS1/IF_RT_CONFIG` 인터페이스를 구현합니다. 예를 들어, 기본 리전을 재정의할 수 있습니다. 다음 예제 명령을 참조하세요.

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

일부 구성 객체는 `IMG` 표현이 없으며 최대 재시도 횟수와 같이 프로그래밍 방식으로만 설정할 수 있습니다. 다음 예제 명령을 참조하세요.

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

의 구성 객체에는에 표시되지 않는 서비스별 메서드도 포함될 AWS 서비스 수 있습니다`/aws1/if_rt_config`. 예를 들어, Amazon S3는 `foobucket.s3.region.amazonaws.com` 가상 엔드포인트 또는 `s3.region.amazonaws.com/foobucket` 경로 스타일을 사용하여 `foobucket` 버킷을 해결할 수 있습니다. 다음 예제 명령으로 경로 스타일 사용을 적용할 수 있습니다.

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

서비스 구성에 대한 자세한 내용은 AWS SDK for SAP ABAP – API 참조 안내서를 참조하세요.

Waiters

비동기 AWS APIs로 작업할 때는 추가 작업을 수행하기 전에 특정 리소스를 사용할 수 있을 때까지 기다려야 합니다. 예를 들어의 CREATETABLE( ) API는 테이블 상태 로 즉시 Amazon DynamoDB 응답합니다CREATING. 테이블 상태가 ACTIVE로 변경된 후에만 읽기 또는 쓰기 작업을 시작할 수 있습니다. 웨이터는 리소스에 대한 작업을 수행하기 전에 AWS 리소스가 특정 상태인지 확인할 수 있는 기능을 제공합니다.

대기자는 서비스 작업을 사용하여 AWS 리소스가 의도한 상태에 도달하거나 리소스가 원하는 상태에 도달하지 않는다고 확인될 때까지 리소스의 상태를 폴링합니다. AWS 리소스를 지속적으로 폴링하는 코드를 작성하는 것은 시간이 많이 걸리고 오류가 발생하기 쉽습니다. Waiters는 사용자를 대신하여 폴링을 수행함으로써 이러한 복잡성을 단순화하는 데 도움을 줍니다.

Waiters를 사용하는 다음 Amazon S3 예제를 참조하십시오.

```
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|
).
```

- 이 예에서는 Amazon S3 클라이언트를 사용하여 버킷을 생성합니다. 이 get_waiter( ) 명령은 bucketexists의 실행 시기를 지정하도록 구현되었습니다.
- 각 Waiters의 iv_max_wait_time 파라미터를 지정해야 합니다. 이는 Waiters가 완료까지 기다려야 하는 총 시간을 나타냅니다. 위 예에서 Waiters는 200초 동안 실행될 수 있습니다.
- 필수 파라미터를 추가로 입력해야 할 수도 있습니다. 위 예제에서는 iv_bucket 파라미터에 Amazon S3 버킷 이름이 필요합니다.

- /AWS1/CX_RT_WAITER_FAILURE 예외는 Waiters가 iv_max_wait_time 파라미터에 지정된 최대 시간을 초과했음을 나타냅니다.
- /AWS1/CX_RT_WAITER_TIMEOUT 예외는 원하는 상태에 도달하지 못해 Waiters가 중지되었음을 나타냅니다.

페이지네이터

일부 AWS 서비스 작업은 페이징된 응답을 제공합니다. 각 응답과 함께 고정된 양의 데이터를 반환하도록 페이지가 매겨집니다. 전체 결과 세트를 검색하려면 토큰이나 마커를 사용하여 후속 요청을 해야 합니다. 예를 들어, ListObjectsV2 Amazon S3 작업은 한 번에 최대 1,000개의 객체만 반환합니다. 다음 결과 페이지를 얻으려면 적절한 토큰으로 후속 요청을 해야 합니다.

페이지 매김은 이전 요청이 중단된 부분부터 다시 시작하라는 요청을 연속적으로 보내는 프로세스입니다. 페이지네이터는 SDK for SAP ABAP에서 제공하는 결과를 반복합니다. 페이지 매김 토큰을 사용하면 API의 기본 메커니즘을 이해하지 않고도 페이지 매김 API를 쉽게 사용할 수 있습니다.

페이지네이터로 작업하기

페이지네이터 객체를 반환하는 get_paginator( ) 메서드를 사용하여 페이지네이터를 만들 수 있습니다. 페이지네이터 객체는 페이지 매김 작업을 호출합니다. 페이지네이터 객체는 기본 API에 제공할 필수 매개변수를 허용합니다. 이 프로세스는 has_next( ) 및 get_next( ) 메서드를 사용하여 페이지가 매겨진 결과를 반복하는 데 사용할 수 있는 반복자 객체를 반환합니다.

- has_next( )은(는) 호출된 작업에 사용할 수 있는 응답 또는 페이지가 더 있는지 여부를 나타내는 부울 값을 반환합니다.
- get_next( )은(는) 작업 응답을 반환합니다.

다음 예에서는 페이지네이터를 사용하여 검색된 S3 버킷의 모든 객체를 나열합니다.

```
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.
```

재시도 동작

SDK for SAP ABAP를 사용하면 제한 또는 일시적 오류로 인해 실패한 AWS 서비스 에 대한 요청의 최대 재시도 횟수를 구성할 수 있습니다. 서비스 클라이언트 수준에서 허용되는 재시도 횟수, 즉 SDK가 실패하고 예외가 발생하기 전에 작업을 재시도하는 횟수는 서비스 구성 객체의 AV_MAX_ATTEMPTS 속성으로 지정됩니다. 서비스 클라이언트 객체가 생성되면 SDK는 AV_MAX_ATTEMPTS 속성을 기본값인 3으로 구성합니다. 서비스 구성 객체를 사용하여 프로그래밍 방식으로 최대 재시도 횟수를 원하는 값으로 설정할 수 있습니다. 자세한 내용은 다음 예시를 참조하십시오.

```
" 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**
>
> 구성 객체 ABAP SDK를 사용하면 /AWS1/IF_RT_CONFIG~SET_RETRY_MODE( ) 방법으로 재시도 모드를 설정할 수 있지만 SDK는 standard 재시도 모드만 지원합니다. 자세한 내용은 SDK의 재시도 동작 AWS SDKs.

프리서명자

미리 서명된 URLs 사용하여 일부에 시간 제한 액세스 권한을 부여할 수 있습니다 AWS 서비스. 미리 서명된 URL은 브라우저에 입력하거나 프로그램에서 서비스 작업을 수행하는 데 사용할 수 있습니다. 미리 서명된 URL은 만료 날짜 및 시간까지 여러 번 사용할 수 있습니다. 자세한 내용은 사전 서명자를

지원하는 서비스에 대해 미리 서명된 URLs SDK for SAP ABAP 클라이언트 작업을 참조하세요. 해당 서비스에 대한 사전 서명자를 생성하기 GET_PRESIGNER( ) 위해 라는 특별한 메서드가 있습니다. 그런 다음 작업을 실제로 수행하는 대신 미리 서명된 URL을 반환한다는 점을 제외하면 API 클라이언트의 메서드에 해당하는 프리서명자의 메서드를 호출합니다.

```
" 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( ).
```

교차 계정 IAM 역할 체인

교차 계정 IAM 역할 체인 지원을 통해 소스 프로필 구성을 통해 여러 AWS 계정의 리소스에 원활하게 액세스할 수 있습니다. 이 기능을 사용하면 하나의 프로필이 다른 역할을 수임하는 역할을 수임하여 복잡한 교차 계정 액세스 패턴을 활성화하는 여러 역할 가정을 구성할 수 있습니다.

자세한 내용은 교차 계정 액세스를 위한 소스 프로파일 사용을 참조하세요.

SDK로 제품 구축

가 사용하는 제품 또는 ABAP 추가 기능은 SDK의 기능을 개선하고 확장할 AWS 서비스 수 있습니다. 이러한 제품을 구축하여 SDK와 함께 사용할 수 있습니다.

주제

- 제품 ID 설정

제품 ID 설정

제품 또는 애드온 내에서 세션을 설정할 때는 제품 ID를 설정하는 것이 좋습니다. 자세한 내용은 다음 예시를 참조하십시오.

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

제품 ID에는 띄어쓰기 또는 특수 문자 없이 문자, 숫자 및 밑줄만 포함되어야 합니다. 이를 제품의 기술적 이름 또는 기타 식별자와 일치시킬 수 있습니다. 여러 제품 또는 애드온을 개발하는 경우 제품 ID는 제품별로 고유해야 합니다. 예를 들어 인보이스 분석기, 세금 계산기 및 가격 책정 엔진 제품의 제품 ID는 `INVOICE_ANALYZER`, `TAX_CALCULATOR` 및 `PRICING_ENGINE`일 수 있습니다.

세션에 제품 ID를 추가하면 각 서비스 호출과 AWS 함께 로 전송되는 원격 측정이 향상됩니다. 제품 ID 및 직접 호출을 수행하는 객체의 네임스페이스가 텔레메트리에 포함됩니다. 이 원격 측정을 사용하여는 고객이 SDK와 관련된 문제를 겪을 경우 호출하는 제품을 지원 식별할 수 있습니다. 이를 통해 고객의 코드가 아니라 제품이 실제로 직접 호출한 것임을 명확히 알 수 있습니다.

HTTP 요청을 로 사용자 지정 AWS

는 HTTP 요청을 생성하고, 페이로드를 전송하고, 응답을 수신하는 프로세스를 AWS SDK for SAP ABAP 처리합니다. 자체 IT 요구 사항에 맞게 HTTP 요청의 동작 또는 콘텐츠를 사용자 지정할 수 있습니다. SDK는 향상 스팟을 HTTP 통신을 개선하기 위한 중앙 장소`/AWS1/RT_EHN_HTTP_CLIENT`로 정의합니다. 개선 사항 스팟은 요청에 HTTP 헤더를 추가하는 것을 지원합니다 AWS.

개선 사항 구현

SAP는 개선 사항 스팟을 구현하기 위해 다음 지침을 제공합니다.

- 클래식 ABAP
- BTP ABAP

개선 사항 필터링

개선 스팟은 동시에 활성화할 수 있는 여러 구현을 지원합니다. 개선 사항이 특정 AWS 서비스 또는 API 작업에 대한 호출에서만 실행되도록 해야 하는 경우 다음 속성을 기반으로 BAdi 실행을 필터링할 수 있습니다.

- `TLA` - 서비스의 세 글자 약어로, 대문자로 되어 있습니다.
- `OPERATION` - API 작업 이름입니다. 예를 들어 S3 버킷에서 객체를 가져오는 작업은 GetObject입니다. 작업 이름은 대/소문자를 구분하며 ABAP 메서드 이름과 정확히 일치하지 않을 수 있습니다.

개선 사항 코딩

향상된 기능은 다음과 같은 방법을 제공합니다.

MODIFY_REQ_HEADERS

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

CT_HEADERS 내부 테이블에 헤더를 추가하고 수정할 수 있습니다. 헤더를 수정하면 AWS 서비스가 사용하는 데이터가 변경되므로 권장하지 않습니다. 추가하는 헤더는 AWS 서비스에서 무시되지만 프록시 서버 또는 기타 미들웨어와 같은 IT 인프라에서 처리할 수 있습니다.

인증 및 원격 측정 헤더를 계산하기 전에 개선 사항 스팟이 호출되므로 개선 사항으로 수정할 수 없습니다.

다음은 구현 예제입니다.

```
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.
```

제한 사항

AWS SDK for SAP ABAP 에는 모두를 위한 SDK 모듈이 포함되어 있습니다 AWS 서비스. 여기에 설명된 것처럼 이러한 모듈 중 일부에는 제한 사항이 있을 수 있습니다.

- 와 같이 MQTT 프로토콜 바인딩에 의존하는 모듈은 작동하지 iotevents않습니다. MQTT는 HTTP 기반 프로토콜이 아니며 현재에서 지원되지 않습니다 AWS SDK for SAP ABAP.
- 이벤트 스트림을 반환하는 작업은 지원되지만 전체 스트림이 수신될 때까지 기다렸다가 제어를 ABAP로 반환합니다. Amazon Bedrock Agents 런타임 InvokeAgent 및 AWS Lambda를 예로 들 수 있습니다. InvokeWithResponseStream
- 이벤트 스트림을 수신하는 작업은 기본 ABAP 플랫폼의 제한으로 인해 지원되지 않습니다. Amazon Q Business Chat 및 Amazon Lex의 예 StartConversation

AWS SDK for SAP ABAP 에는 다음과 같은 기능 제한이 있습니다.

- 다음 Amazon S3 기능은 아직 지원되지 않습니다.

- 다중 리전 액세스 포인트
- Amazon S3 클라이언트 측 암호화

AWS SDK for SAP ABAP - BTP 에디션은 개발자 미리 보기 중에 다음과 같은 제한 사항이 있습니다.

- 일부 모듈은 사용하지 못할 수 있습니다.
- 제거할 수 없습니다.
- 자주 업데이트되지 않습니다.

SDK for SAP ABAP 코드 예제

이 주제의 코드 예제에서는 AWS SDK for SAP ABAP를와 함께 사용하는 방법을 보여줍니다 AWS.

기본 사항은 서비스 내에서 필수 작업을 수행하는 방법을 보여주는 코드 예제입니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접 호출하는 방법을 보여주며, 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

시나리오는 동일한 서비스 내에서 또는 다른 AWS 서비스와 결합된 상태에서 여러 함수를 직접적으로 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예제입니다.

일부 서비스에는 서비스와 관련된 라이브러리 또는 함수를 활용하는 방법을 보여주는 추가 예제 범주가 포함되어 있습니다.

서비스

- SDK for SAP ABAP를 사용한 ACM 예제
- SDK for SAP ABAP를 사용한 API Gateway 예제
- SDK for SAP ABAP를 사용한 Application Recovery Controller 예제
- SDK for SAP ABAP를 사용한 Aurora 예제
- SDK for SAP ABAP를 사용한 Auto Scaling 예제
- SDK for SAP ABAP를 사용한 Amazon Bedrock 런타임 예
- SDK for SAP ABAP를 사용한 Amazon Bedrock Agents 런타임 예제
- SDK for SAP ABAP를 사용한 CloudFront 예제
- SDK for SAP ABAP를 사용한 CloudWatch 예
- SDK for SAP ABAP를 사용한 CloudWatch Logs 예제
- SDK for SAP ABAP를 사용한 Amazon Cognito 자격 증명 공급자 예제
- SDK for SAP ABAP를 사용한 Amazon Comprehend 예제
- AWS Config SDK for SAP ABAP를 사용한 예제
- AWS Control Tower SDK for SAP ABAP를 사용한 예제
- SDK for SAP ABAP를 사용한 Firehose 예제
- SDK for SAP ABAP를 사용한 DynamoDB 예
- SDK for SAP ABAP를 사용한 Amazon EC2 예
- SDK for SAP ABAP를 사용한 Amazon ECR 예제

- SDK for SAP ABAP를 사용한 Amazon EMR 예제
- SDK for SAP ABAP를 사용한 EventBridge 스케줄러 예제
- AWS Glue SDK for SAP ABAP를 사용한 예제
- SDK for SAP ABAP를 사용한 HealthImaging 예제
- SDK for SAP ABAP를 사용한 HealthLake 예제
- SDK for SAP ABAP를 사용한 IAM 예제
- AWS IoT SiteWise SDK for SAP ABAP를 사용한 예제
- SDK for SAP ABAP를 사용한 Amazon Keyspaces 예제
- SDK for SAP ABAP를 사용하는 Kinesis 예
- AWS KMS SDK for SAP ABAP를 사용한 예제
- SDK for SAP ABAP를 사용한 Lambda 예
- SDK for SAP ABAP를 사용한 조직 예제
- SDK for SAP ABAP를 사용한 Amazon Pinpoint 예제
- SDK for SAP ABAP를 사용한 Amazon Pinpoint SMS 및 음성 API 예제
- SDK for SAP ABAP를 사용한 Amazon Polly 예제
- SDK for SAP ABAP를 사용한 Amazon RDS 예제
- SDK for SAP ABAP를 사용한 Amazon Redshift 예제
- SDK for SAP ABAP를 사용한 Amazon Rekognition 예제
- SDK for SAP ABAP를 사용한 Amazon S3 예
- SDK for SAP ABAP를 사용한 SageMaker AI 예제
- SDK for SAP ABAP를 사용한 Secrets Manager 예제
- SDK for SAP ABAP를 사용한 Amazon SES 예제
- SDK for SAP ABAP를 사용한 Amazon SES API v2 예제
- SDK for SAP ABAP를 사용한 Amazon SNS에 대한 코드 예제
- SDK for SAP ABAP를 사용한 Amazon SQS 예
- SDK for SAP ABAP를 사용한 Step Functions 예제
- SDK for SAP ABAP를 사용한 Systems Manager 예제
- SDK for SAP ABAP를 사용한 Amazon Textract 예
- SDK for SAP ABAP를 사용한 Amazon Transcribe 예제
- SDK for SAP ABAP를 사용한 Amazon Translate 예

SDK for SAP ABAP를 사용한 ACM 예제

다음 코드 예제에서는 ACM과 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`AddTagsToCertificate`

다음 코드 예시는 `AddTagsToCertificate`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 AddTagsToCertificate를 참조하세요. AWS

`DeleteCertificate`

다음 코드 예시는 `DeleteCertificate`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DeleteCertificate를 참조하세요. AWS

`DescribeCertificate`

다음 코드 예시는 `DescribeCertificate`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DescribeCertificate를 참조하세요. AWS

`GetCertificate`

다음 코드 예시는 `GetCertificate`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 GetCertificate를 참조하세요. AWS

`ImportCertificate`

다음 코드 예시는 `ImportCertificate`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
  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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ImportCertificate를 참조하세요.

`ListCertificates`

다음 코드 예시는 `ListCertificates`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
  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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 ListCertificates를 참조하세요. AWS

`ListTagsForCertificate`

다음 코드 예시는 `ListTagsForCertificate`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 ListTagsForCertificate를 참조하세요. AWS

`RemoveTagsFromCertificate`

다음 코드 예시는 `RemoveTagsFromCertificate`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 RemoveTagsFromCertificate를 참조하세요. AWS

`RequestCertificate`

다음 코드 예시는 `RequestCertificate`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 RequestCertificate를 참조하세요. AWS

`ResendValidationEmail`

다음 코드 예시는 `ResendValidationEmail`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 ResendValidationEmail을 참조하세요. AWS

SDK for SAP ABAP를 사용한 API Gateway 예제

다음 코드 예제에서는 API Gateway와 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`CreateDeployment`

다음 코드 예시는 `CreateDeployment`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateDeployment를 참조하세요.

`CreateResource`

다음 코드 예시는 `CreateResource`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
  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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateResource를 참조하세요.

`CreateRestApi`

다음 코드 예시는 `CreateRestApi`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
  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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateRestApi를 참조하세요.

`DeleteRestApi`

다음 코드 예시는 `DeleteRestApi`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteRestApi를 참조하세요.

`GetResources`

다음 코드 예시는 `GetResources`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 GetResources를 참조하세요.

`GetRestApis`

다음 코드 예시는 `GetRestApis`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 GetRestApis를 참조하세요. AWS

`PutIntegration`

다음 코드 예시는 `PutIntegration`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 PutIntegration을 참조하세요.

PutIntegrationResponse

다음 코드 예시는 PutIntegrationResponse의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 PutIntegrationResponse를 참조하세요.

PutMethod

다음 코드 예시는 PutMethod의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 PutMethod를 참조하세요.

`PutMethodResponse`

다음 코드 예시는 `PutMethodResponse`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 PutMethodResponse를 참조하세요.

SDK for SAP ABAP를 사용한 Application Recovery Controller 예제

다음 코드 예제에서는 Application Recovery Controller와 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`GetRoutingControlState`

다음 코드 예시는 `GetRoutingControlState`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 GetRoutingControlState를 참조하세요. AWS

`UpdateRoutingControlState`

다음 코드 예시는 `UpdateRoutingControlState`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 UpdateRoutingControlState를 참조하세요. AWS

SDK for SAP ABAP를 사용한 Aurora 예제

다음 코드 예제에서는 AWS SDK for SAP ABAP를 Aurora와 함께 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`CreateDBClusterParameterGroup`

다음 코드 예시는 `CreateDBClusterParameterGroup`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateDBClusterParameterGroup을 참조하세요.

`DeleteDBClusterParameterGroup`

다음 코드 예시는 `DeleteDBClusterParameterGroup`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
  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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteDBClusterParameterGroup을 참조하세요.

`DescribeDBClusterParameterGroups`

다음 코드 예시는 `DescribeDBClusterParameterGroups`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeDBClusterParameterGroups를 참조하세요.

`DescribeDBClusterParameters`

다음 코드 예시는 `DescribeDBClusterParameters`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeDBClusterParameters를 참조하세요.

`DescribeDBEngineVersions`

다음 코드 예시는 `DescribeDBEngineVersions`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   " 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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeDBEngineVersions를 참조하세요.

`DescribeOrderableDBInstanceOptions`

다음 코드 예시는 `DescribeOrderableDBInstanceOptions`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   " 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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeOrderableDBInstanceOptions를 참조하세요.

`ModifyDBClusterParameterGroup`

다음 코드 예시는 `ModifyDBClusterParameterGroup`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ModifyDBClusterParameterGroup을 참조하세요.

SDK for SAP ABAP를 사용한 Auto Scaling 예제

다음 코드 예제에서는 Auto Scaling과 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`CreateAutoScalingGroup`

다음 코드 예시는 `CreateAutoScalingGroup`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 CreateAutoScalingGroup을 참조하세요. AWS

DeleteAutoScalingGroup

다음 코드 예시는 DeleteAutoScalingGroup의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    " 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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DeleteAutoScalingGroup을 참조하세요. AWS

DescribeAutoScalingGroups

다음 코드 예시는 DescribeAutoScalingGroups의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DescribeAutoScalingGroups를 참조하세요. AWS

`DescribeAutoScalingInstances`

다음 코드 예시는 `DescribeAutoScalingInstances`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    " 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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DescribeAutoScalingInstances를 참조하세요. AWS

`DescribeScalingActivities`

다음 코드 예시는 `DescribeScalingActivities`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
" 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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DescribeScalingActivities를 참조하세요. AWS

`DisableMetricsCollection`

다음 코드 예시는 `DisableMetricsCollection`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
" 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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DisableMetricsCollection을 참조하세요. AWS

`EnableMetricsCollection`

다음 코드 예시는 `EnableMetricsCollection`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    " 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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 EnableMetricsCollection을 참조하세요. AWS

`SetDesiredCapacity`

다음 코드 예시는 `SetDesiredCapacity`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
  " 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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 SetDesiredCapacity를 참조하세요. AWS

`TerminateInstanceInAutoScalingGroup`

다음 코드 예시는 `TerminateInstanceInAutoScalingGroup`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
" 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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 TerminateInstanceInAutoScalingGroup을 참조하세요. AWS

`UpdateAutoScalingGroup`

다음 코드 예시는 `UpdateAutoScalingGroup`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
" 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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 UpdateAutoScalingGroup을 참조하세요. AWS

SDK for SAP ABAP를 사용한 Amazon Bedrock 런타임 예

다음 코드 예제에서는 Amazon Bedrock 런타임과 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- Anthropic Claude

- Stable Diffusion

Anthropic Claude

InvokeModel

다음 코드 예제에서는 Invoke Model API를 사용하여 Anthropic Claude에 텍스트 메시지를 보내는 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

Anthropic Claude 2 파운데이션 모델을 간접 호출하여 텍스트를 생성합니다. 이 예에서는 일부 NetWeaver 버전에서는 사용할 수 없는 /US2/CL_JSON 기능을 사용합니다.

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

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

    "Leave ls_input-stop_sequences empty.
    ls_input-prompt = |\n\nHuman:\\n{ iv_prompt }\n\nAssistant:\n|.
```

```
    ls_input-max_tokens_to_sample = 2048.
    ls_input-temperature = '0.5'.
    ls_input-top_k = 250.
    ls_input-top_p = 1.

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

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

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

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

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

    ENDTRY.
```

L2 상위 수준 클라이언트를 사용해 텍스트를 생성하도록 Anthropic Claude 2 파운데이션 모델을 간접 호출합니다.

```
    TRY.
        DATA(lo_bdr_l2_claude) = /aws1/cl_bdr_l2_factory=>create_claude_2( lo_bdr ).
        " iv_prompt can contain a prompt like 'tell me a joke about Java
 programmers'.
        DATA(lv_answer) = lo_bdr_l2_claude->prompt_for_text( iv_prompt ).
      CATCH /aws1/cx_bdraccessdeniedex INTO DATA(lo_ex).
        WRITE / lo_ex->get_text( ).
        WRITE / |Don't forget to enable model access at https://
console.aws.amazon.com/bedrock/home?#/modelaccess|.

    ENDTRY.
```

L2 상위 수준 클라이언트를 사용해 텍스트를 생성하도록 Anthropic Claude 3 파운데이션 모델을 간접 호출합니다.

```
    TRY.
        " Choose a model ID from Anthropic that supports the Messages API -
 currently this is
        " Claude v2, Claude v3 and v3.5.  For the list of model ID, see:
        " https://docs.aws.amazon.com/bedrock/latest/userguide/model-ids.html

        " for the list of models that support the Messages API see:
        " https://docs.aws.amazon.com/bedrock/latest/userguide/model-parameters-
anthropic-claude-messages.html
        DATA(lo_bdr_l2_claude) = /aws1/cl_bdr_l2_factory=>create_anthropic_msg_api(
          io_bdr = lo_bdr
          iv_model_id = 'anthropic.claude-3-sonnet-20240229-v1:0' ).  " choosing
 Claude v3 Sonnet
        " iv_prompt can contain a prompt like 'tell me a joke about Java
 programmers'.
        DATA(lv_answer) = lo_bdr_l2_claude->prompt_for_text( iv_prompt = iv_prompt
                                                             iv_max_tokens = 100 ).
      CATCH /aws1/cx_bdraccessdeniedex INTO DATA(lo_ex).
        WRITE / lo_ex->get_text( ).
        WRITE / |Don't forget to enable model access at https://
console.aws.amazon.com/bedrock/home?#/modelaccess|.

    ENDTRY.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 InvokeModel을 참조하세요.

Stable Diffusion

InvokeModel

다음 코드 예제에서는 이미지 생성을 위해 Amazon Bedrock에서 Stability.ai Stable Diffusion XL 모델을 간접적으로 호출하는 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

Stable Diffusion을 사용하여 이미지를 생성합니다.

```
    "Stable Diffusion Input Parameters should be in a format like this:
*   {
*     "text_prompts": [
*       {"text":"Draw a dolphin with a mustache"},
*       {"text":"Make it photorealistic"}
*     ],
*     "cfg_scale":10,
*     "seed":0,
*     "steps":50
*   }
    TYPES: BEGIN OF prompt_ts,
             text TYPE /aws1/rt_shape_string,
           END OF prompt_ts.

    DATA: BEGIN OF ls_input,
            text_prompts TYPE STANDARD TABLE OF prompt_ts,
            cfg_scale    TYPE /aws1/rt_shape_integer,
            seed         TYPE /aws1/rt_shape_integer,
            steps        TYPE /aws1/rt_shape_integer,
          END OF ls_input.

    APPEND VALUE prompt_ts( text = iv_prompt ) TO ls_input-text_prompts.
    ls_input-cfg_scale = 10.
    ls_input-seed = 0. "or better, choose a random integer.
    ls_input-steps = 50.
```

```
    DATA(lv_json) = /ui2/cl_json=>serialize(
      data = ls_input
                pretty_name   = /ui2/cl_json=>pretty_mode-low_case ).

    TRY.
        DATA(lo_response) = lo_bdr->invokemodel(
          iv_body = /aws1/cl_rt_util=>string_to_xstring( lv_json )
          iv_modelid = 'stability.stable-diffusion-xl-v1'
          iv_accept = 'application/json'
          iv_contenttype = 'application/json' ).

        "Stable Diffusion Result Format:
*       {
*         "result": "success",
*         "artifacts": [
*           {
*             "seed": 0,
*             "base64": "iVBORw0KGgoAAAANSUhEUgAAAgAAA....
*             "finishReason": "SUCCESS"
*           }
*         ]
*       }
        TYPES: BEGIN OF artifact_ts,
                 seed         TYPE /aws1/rt_shape_integer,
                 base64       TYPE /aws1/rt_shape_string,
                 finishreason TYPE /aws1/rt_shape_string,
               END OF artifact_ts.

        DATA: BEGIN OF ls_response,
                result    TYPE /aws1/rt_shape_string,
                artifacts TYPE STANDARD TABLE OF artifact_ts,
              END OF ls_response.

        /ui2/cl_json=>deserialize(
          EXPORTING jsonx = lo_response->get_body( )
                    pretty_name = /ui2/cl_json=>pretty_mode-camel_case
          CHANGING  data  = ls_response ).
        IF ls_response-artifacts IS NOT INITIAL.
          DATA(lv_image) =
 cl_http_utility=>if_http_utility~decode_x_base64( ls_response-artifacts[ 1 ]-
base64 ).
        ENDIF.
      CATCH /aws1/cx_bdraccessdeniedex INTO DATA(lo_ex).
```

```
        WRITE / lo_ex->get_text( ).
        WRITE / |Don't forget to enable model access at https://
console.aws.amazon.com/bedrock/home?#/modelaccess|.

    ENDTRY.
```

L2 상위 수준 클라이언트를 사용하여 Stability.ai Stable Diffusion XL 파운데이션 모델을 간접적으로 호출해 이미지를 생성합니다.

```
    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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 InvokeModel을 참조하세요.

SDK for SAP ABAP를 사용한 Amazon Bedrock Agents 런타임 예제

다음 코드 예제에서는 Amazon Bedrock Agents 런타임과 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`InvokeAgent`

다음 코드 예시는 InvokeAgent의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 InvokeAgent를 참조하세요.

SDK for SAP ABAP를 사용한 CloudFront 예제

다음 코드 예제에서는 CloudFront에서 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`ListDistributions`

다음 코드 예시는 `ListDistributions`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListDistributions를 참조하세요.

`UpdateDistribution`

다음 코드 예시는 `UpdateDistribution`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 UpdateDistribution을 참조하세요. AWS

SDK for SAP ABAP를 사용한 CloudWatch 예

다음 코드 예제에서는 CloudWatch와 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접 호출하는 방법을 보여주며, 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

시나리오는 동일한 서비스 내에서 또는 다른 AWS 서비스와 결합된 상태에서 여러 함수를 직접적으로 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예제입니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업
- 시나리오

작업

`DeleteAlarms`

다음 코드 예시는 `DeleteAlarms`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteAlarms를 참조하세요.

`DescribeAlarms`

다음 코드 예시는 `DescribeAlarms`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeAlarms를 참조하세요.

`DescribeAlarmsForMetric`

다음 코드 예시는 `DescribeAlarmsForMetric`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DescribeAlarmsForMetric을 참조하세요. AWS

`DisableAlarmActions`

다음 코드 예시는 `DisableAlarmActions`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    "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.
```

- API에 대한 자세한 내용은 AWS SDK for SAP ABAP API 참조의 DisableAlarmActions를 참조하세요.

`EnableAlarmActions`

다음 코드 예시는 `EnableAlarmActions`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    "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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 EnableAlarmActions를 참조하세요.

`GetMetricStatistics`

다음 코드 예시는 `GetMetricStatistics`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 GetMetricStatistics를 참조하세요. AWS

`ListMetrics`

다음 코드 예시는 `ListMetrics`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   "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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListMetrics를 참조하세요.

`PutMetricAlarm`

다음 코드 예시는 `PutMetricAlarm`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 PutMetricAlarm을 참조하세요.

`PutMetricData`

다음 코드 예시는 `PutMetricData`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

CloudWatch 지표에 데이터 집합을 추가합니다.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 PutMetricData를 참조하세요.

시나리오

경보 시작하기

다음 코드 예제에서는 다음과 같은 작업을 수행하는 방법을 보여줍니다.

- 경보를 생성합니다.
- 경보 작업을 비활성화합니다.
- 경보를 설명합니다.
- 경보를 삭제합니다.

SDK for SAP ABAP

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 다음 주제를 참조하세요.
  - DeleteAlarms
  - DescribeAlarms
  - DisableAlarmActions
  - PutMetricAlarm

SDK for SAP ABAP를 사용한 CloudWatch Logs 예제

다음 코드 예제에서는 CloudWatch Logs와 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`GetQueryResults`

다음 코드 예시는 `GetQueryResults`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 GetQueryResults를 참조하세요. AWS

`StartQuery`

다음 코드 예시는 `StartQuery`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 StartQuery를 참조하세요.

SDK for SAP ABAP를 사용한 Amazon Cognito 자격 증명 공급자 예제

다음 코드 예제에서는 Amazon Cognito 자격 증명 공급자와 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`AdminInitiateAuth`

다음 코드 예시는 `AdminInitiateAuth`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
  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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 AdminInitiateAuth를 참조하세요. AWS

AdminRespondToAuthChallenge

다음 코드 예시는 AdminRespondToAuthChallenge의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 AdminRespondToAuthChallenge를 참조하세요. AWS

`AssociateSoftwareToken`

다음 코드 예시는 `AssociateSoftwareToken`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
  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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 AssociateSoftwareToken을 참조하세요. AWS

`ListUsers`

다음 코드 예시는 `ListUsers`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API의 ListUsers을 참조하세요.

`VerifySoftwareToken`

다음 코드 예시는 `VerifySoftwareToken`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 VerifySoftwareToken을 참조하세요. AWS

SDK for SAP ABAP를 사용한 Amazon Comprehend 예제

다음 코드 예제에서는 Amazon Comprehend와 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`CreateDocumentClassifier`

다음 코드 예시는 `CreateDocumentClassifier`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 CreateDocumentClassifier를 참조하세요. AWS

`DeleteDocumentClassifier`

다음 코드 예시는 `DeleteDocumentClassifier`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
  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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DeleteDocumentClassifier를 참조하세요. AWS

`DescribeDocumentClassificationJob`

다음 코드 예시는 `DescribeDocumentClassificationJob`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DescribeDocumentClassificationJob을 참조하세요. AWS

`DescribeDocumentClassifier`

다음 코드 예시는 `DescribeDocumentClassifier`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DescribeDocumentClassifier를 참조하세요. AWS

`DescribeTopicsDetectionJob`

다음 코드 예시는 `DescribeTopicsDetectionJob`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DescribeTopicsDetectionJob을 참조하세요. AWS

`DetectDominantLanguage`

다음 코드 예시는 `DetectDominantLanguage`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DetectDominantLanguage를 참조하세요. AWS

`DetectEntities`

다음 코드 예시는 `DetectEntities`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DetectEntities를 참조하세요.

`DetectKeyPhrases`

다음 코드 예시는 `DetectKeyPhrases`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DetectKeyPhrases를 참조하세요. AWS

`DetectPiiEntities`

다음 코드 예시는 `DetectPiiEntities`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DetectPiiEntities를 참조하세요. AWS

`DetectSentiment`

다음 코드 예시는 `DetectSentiment`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DetectSentiment를 참조하세요.

`DetectSyntax`

다음 코드 예시는 `DetectSyntax`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
  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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DetectSyntax를 참조하세요.

`ListDocumentClassificationJobs`

다음 코드 예시는 `ListDocumentClassificationJobs`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 ListDocumentClassificationJobs를 참조하세요. AWS

`ListDocumentClassifiers`

다음 코드 예시는 `ListDocumentClassifiers`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 ListDocumentClassifiers를 참조하세요. AWS

`ListTopicsDetectionJobs`

다음 코드 예시는 `ListTopicsDetectionJobs`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 ListTopicsDetectionJobs를 참조하세요. AWS

`StartDocumentClassificationJob`

다음 코드 예시는 `StartDocumentClassificationJob`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 StartDocumentClassificationJob을 참조하세요. AWS

`StartTopicsDetectionJob`

다음 코드 예시는 `StartTopicsDetectionJob`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 StartTopicsDetectionJob을 참조하세요. AWS

AWS Config SDK for SAP ABAP를 사용한 예제

다음 코드 예제에서는 SAP ABAP용 AWS SDK를와 함께 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS Config.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`DeleteConfigRule`

다음 코드 예시는 `DeleteConfigRule`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

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

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteConfigRule을 참조하세요.

`DescribeConfigRules`

다음 코드 예시는 `DescribeConfigRules`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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'.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeConfigRules를 참조하세요.

`PutConfigRule`

다음 코드 예시는 `PutConfigRule`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
" 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'.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 PutConfigRule을 참조하세요.

AWS Control Tower SDK for SAP ABAP를 사용한 예제

다음 코드 예제에서는 SAP ABAP용 AWS SDK를와 함께 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS Control Tower.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`DisableBaseline`

다음 코드 예시는 `DisableBaseline`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DisableBaseline을 참조하세요.

`DisableControl`

다음 코드 예시는 `DisableControl`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
" 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'.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DisableControl을 참조하세요.

`EnableBaseline`

다음 코드 예시는 `EnableBaseline`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    " 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'.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 EnableBaseline을 참조하세요.

`EnableControl`

다음 코드 예시는 `EnableControl`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
" 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'.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 EnableControl을 참조하세요.

GetBaselineOperation

다음 코드 예시는 GetBaselineOperation의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

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

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

- API 세부 정보는 SDK for SAP ABAP API 참조의 GetBaselineOperation을 참조하세요. AWS

`GetControlOperation`

다음 코드 예시는 `GetControlOperation`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

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

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

- API 세부 정보는 SDK for SAP ABAP API 참조의 GetControlOperation을 참조하세요. AWS

`ListBaselines`

다음 코드 예시는 `ListBaselines`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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'.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListBaselines를 참조하세요.

`ListEnabledBaselines`

다음 코드 예시는 `ListEnabledBaselines`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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'.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 ListEnabledBaselines를 참조하세요. AWS

`ListEnabledControls`

다음 코드 예시는 `ListEnabledControls`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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'.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 ListEnabledControls를 참조하세요. AWS

`ListLandingZones`

다음 코드 예시는 `ListLandingZones`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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'.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 ListLandingZones를 참조하세요. AWS

`ResetEnabledBaseline`

다음 코드 예시는 `ResetEnabledBaseline`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
" 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'.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 ResetEnabledBaseline을 참조하세요. AWS

SDK for SAP ABAP를 사용한 Firehose 예제

다음 코드 예제에서는 Firehose와 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`PutRecord`

다음 코드 예시는 `PutRecord`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 PutRecord를 참조하세요.

`PutRecordBatch`

다음 코드 예시는 `PutRecordBatch`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 PutRecordBatch를 참조하세요.

SDK for SAP ABAP를 사용한 DynamoDB 예

다음 코드 예제에서는 DynamoDB와 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

기본 사항은 서비스 내에서 필수 작업을 수행하는 방법을 보여주는 코드 예제입니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 기본 사항
- 작업

기본 사항

기본 사항 알아보기

다음 코드 예제에서는 다음과 같은 작업을 수행하는 방법을 보여줍니다.

- 영화 데이터를 저장할 수 있는 테이블을 생성합니다.
- 테이블에 하나의 영화를 추가하고 가져오고 업데이트합니다.
- 샘플 JSON 파일에서 테이블에 영화 데이터를 씁니다.
- 특정 연도에 개봉된 영화를 쿼리합니다.
- 특정 연도 범위 동안 개봉된 영화를 스캔합니다.
- 테이블에서 영화를 삭제한 다음, 테이블을 삭제합니다.

SDK for SAP ABAP

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    " 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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 다음 주제를 참조하세요.
  - BatchWriteItem
  - CreateTable
  - DeleteItem
  - DeleteTable
  - DescribeTable
  - GetItem
  - PutItem
  - Query
  - Scan
  - UpdateItem

작업

`CreateTable`

다음 코드 예시는 `CreateTable`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는AWS SDK for SAP ABAP API의 CreateTable을 참조하세요.

`DeleteItem`

다음 코드 예시는 `DeleteItem`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는AWS SDK for SAP ABAP API의 DeleteItem을 참조하세요.

`DeleteTable`

다음 코드 예시는 `DeleteTable`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는AWS SDK for SAP ABAP API의 DeleteTable을 참조하세요.

`DescribeTable`

다음 코드 예시는 `DescribeTable`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeTable을 참조하세요.

`GetItem`

다음 코드 예시는 `GetItem`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는AWS SDK for SAP ABAP API의 GetItem을 참조하세요.

`ListTables`

다음 코드 예시는 `ListTables`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는AWS SDK for SAP ABAP API의 ListTables을 참조하세요.

`PutItem`

다음 코드 예시는 `PutItem`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는AWS SDK for SAP ABAP API의 PutItem을 참조하세요.

`Query`

다음 코드 예시는 `Query`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는AWS SDK for SAP ABAP API의 Query을 참조하세요.

Scan

다음 코드 예시는 Scan의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는AWS SDK for SAP ABAP API의 Scan을 참조하세요.

`UpdateItem`

다음 코드 예시는 `UpdateItem`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API 세부 정보는AWS SDK for SAP ABAP API의 UpdateItem을 참조하세요.

SDK for SAP ABAP를 사용한 Amazon EC2 예

다음 코드 예제에서는 Amazon EC2와 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`AllocateAddress`

다음 코드 예시는 `AllocateAddress`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 AllocateAddress를 참조하세요.

`AssociateAddress`

다음 코드 예시는 `AssociateAddress`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 AssociateAddress를 참조하세요.

`AuthorizeSecurityGroupIngress`

다음 코드 예시는 `AuthorizeSecurityGroupIngress`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    " 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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 AuthorizeSecurityGroupIngress를 참조하세요. AWS

`CreateKeyPair`

다음 코드 예시는 `CreateKeyPair`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateKeyPair을 참조하세요.

`CreateSecurityGroup`

다음 코드 예시는 `CreateSecurityGroup`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateSecurityGroup을 참조하세요.

`CreateVpc`

다음 코드 예시는 `CreateVpc`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    " 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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateVpc를 참조하세요.

`CreateVpcEndpoint`

다음 코드 예시는 `CreateVpcEndpoint`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    " 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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 CreateVpcEndpoint를 참조하세요. AWS

`DeleteKeyPair`

다음 코드 예시는 `DeleteKeyPair`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteKeyPair을 참조하세요.

`DeleteSecurityGroup`

다음 코드 예시는 `DeleteSecurityGroup`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteSecurityGroup을 참조하세요.

`DeleteVpc`

다음 코드 예시는 `DeleteVpc`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteVpc를 참조하세요.

`DeleteVpcEndpoints`

다음 코드 예시는 `DeleteVpcEndpoints`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DeleteVpcEndpoints를 참조하세요. AWS

`DescribeAddresses`

다음 코드 예시는 `DescribeAddresses`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeAddresses를 참조하세요.

`DescribeAvailabilityZones`

다음 코드 예시는 `DescribeAvailabilityZones`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeAvailabilityZones를 참조하세요.

`DescribeImages`

다음 코드 예시는 `DescribeImages`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeImages를 참조하세요.

`DescribeInstanceTypes`

다음 코드 예시는 `DescribeInstanceTypes`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    " 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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeInstanceTypes.

`DescribeInstances`

다음 코드 예시는 `DescribeInstances`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeInstances를 참조하세요.

`DescribeKeyPairs`

다음 코드 예시는 `DescribeKeyPairs`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeKeyPairs를 참조하세요.

`DescribeRegions`

다음 코드 예시는 `DescribeRegions`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeRegions를 참조하세요.

`DescribeRouteTables`

다음 코드 예시는 `DescribeRouteTables`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    " 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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DescribeRouteTables를 참조하세요. AWS

`DescribeSecurityGroups`

다음 코드 예시는 `DescribeSecurityGroups`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeSecurityGroups를 참조하세요.

`MonitorInstances`

다음 코드 예시는 `MonitorInstances`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 MonitorInstances를 참조하세요.

`RebootInstances`

다음 코드 예시는 `RebootInstances`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 RebootInstances를 참조하세요.

`ReleaseAddress`

다음 코드 예시는 `ReleaseAddress`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ReleaseAddress를 참조하세요.

`RunInstances`

다음 코드 예시는 `RunInstances`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    " 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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 RunInstances를 참조하세요.

`StartInstances`

다음 코드 예시는 `StartInstances`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 StartInstances를 참조하세요.

`StopInstances`

다음 코드 예시는 `StopInstances`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 StopInstances를 참조하세요.

SDK for SAP ABAP를 사용한 Amazon ECR 예제

다음 코드 예제에서는 Amazon ECR과 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`CreateRepository`

다음 코드 예시는 `CreateRepository`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateRepository를 참조하세요.

`DeleteRepository`

다음 코드 예시는 `DeleteRepository`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteRepository를 참조하세요.

`DescribeImages`

다음 코드 예시는 `DescribeImages`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeImages를 참조하세요.

`DescribeRepositories`

다음 코드 예시는 `DescribeRepositories`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeRepositories를 참조하세요.

`GetAuthorizationToken`

다음 코드 예시는 `GetAuthorizationToken`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 GetAuthorizationToken을 참조하세요. AWS

`GetRepositoryPolicy`

다음 코드 예시는 `GetRepositoryPolicy`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 GetRepositoryPolicy를 참조하세요. AWS

`PutLifeCyclePolicy`

다음 코드 예시는 `PutLifeCyclePolicy`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 PutLifeCyclePolicy를 참조하세요. AWS

`SetRepositoryPolicy`

다음 코드 예시는 `SetRepositoryPolicy`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 SetRepositoryPolicy를 참조하세요. AWS

SDK for SAP ABAP를 사용한 Amazon EMR 예제

다음 코드 예제에서는 Amazon EMR과 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`AddJobFlowSteps`

다음 코드 예시는 `AddJobFlowSteps`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 AddJobFlowSteps를 참조하세요. AWS

`DescribeCluster`

다음 코드 예시는 `DescribeCluster`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeCluster를 참조하세요.

`DescribeStep`

다음 코드 예시는 `DescribeStep`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeStep을 참조하세요.

`ListSteps`

다음 코드 예시는 `ListSteps`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListSteps를 참조하세요.

`RunJobFlow`

다음 코드 예시는 `RunJobFlow`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 RunJobFlow를 참조하세요.

`TerminateJobFlows`

다음 코드 예시는 `TerminateJobFlows`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 TerminateJobFlows를 참조하세요. AWS

SDK for SAP ABAP를 사용한 EventBridge 스케줄러 예제

다음 코드 예제에서는 EventBridge 스케줄러와 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`CreateSchedule`

다음 코드 예시는 `CreateSchedule`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateSchedule을 참조하세요.

`CreateScheduleGroup`

다음 코드 예시는 `CreateScheduleGroup`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 CreateScheduleGroup을 참조하세요. AWS

`DeleteSchedule`

다음 코드 예시는 `DeleteSchedule`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteSchedule을 참조하세요.

`DeleteScheduleGroup`

다음 코드 예시는 `DeleteScheduleGroup`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DeleteScheduleGroup을 참조하세요. AWS

AWS Glue SDK for SAP ABAP를 사용한 예제

다음 코드 예제에서는 AWS SDK for SAP ABAP를와 함께 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS Glue.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`CreateCrawler`

다음 코드 예시는 `CreateCrawler`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateCrawler를 참조하세요.

`CreateJob`

다음 코드 예시는 `CreateJob`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateJob을 참조하세요.

`DeleteCrawler`

다음 코드 예시는 `DeleteCrawler`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteCrawler를 참조하세요.

`DeleteDatabase`

다음 코드 예시는 `DeleteDatabase`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteDatabase를 참조하세요.

`DeleteJob`

다음 코드 예시는 `DeleteJob`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteJob을 참조하세요.

`DeleteTable`

다음 코드 예시는 `DeleteTable`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는AWS SDK for SAP ABAP API의 DeleteTable을 참조하세요.

`GetCrawler`

다음 코드 예시는 `GetCrawler`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 GetCrawler를 참조하세요.

`GetDatabase`

다음 코드 예시는 `GetDatabase`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 GetDatabase를 참조하세요.

`GetJobRun`

다음 코드 예시는 `GetJobRun`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
  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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 GetJobRun을 참조하세요.

`GetJobRuns`

다음 코드 예시는 `GetJobRuns`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
  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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 GetJobRuns를 참조하세요.

`GetTables`

다음 코드 예시는 `GetTables`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 GetTables를 참조하세요.

`ListJobs`

다음 코드 예시는 `ListJobs`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListJobs를 참조하세요.

`StartCrawler`

다음 코드 예시는 `StartCrawler`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 StartCrawler를 참조하세요.

`StartJobRun`

다음 코드 예시는 `StartJobRun`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 StartJobRun을 참조하세요.

SDK for SAP ABAP를 사용한 HealthImaging 예제

다음 코드 예제에서는 HealthImaging과 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

CopyImageSet

다음 코드 예시는 CopyImageSet의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 CopyImageSet를 참조하세요. AWS

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

`CreateDatastore`

다음 코드 예시는 `CreateDatastore`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

```
  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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateDatastore를 참조하세요.

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

`DeleteDatastore`

다음 코드 예시는 `DeleteDatastore`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteDatastore를 참조하세요.

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

`DeleteImageSet`

다음 코드 예시는 `DeleteImageSet`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DeleteImageSet를 참조하세요. AWS

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

`GetDICOMImportJob`

다음 코드 예시는 `GetDICOMImportJob`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 GetDICOMImportJob을 참조하세요. AWS

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

`GetDatastore`

다음 코드 예시는 `GetDatastore`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 GetDatastore를 참조하세요.

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

GetImageFrame

다음 코드 예시는 GetImageFrame의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 GetImageFrame을 참조하세요.

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

GetImageSet

다음 코드 예시는 GetImageSet의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 GetImageSet를 참조하세요. AWS

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

GetImageSetMetadata

다음 코드 예시는 GetImageSetMetadata의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 GetImageSetMetadata를 참조하세요. AWS

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

`ListDICOMImportJobs`

다음 코드 예시는 `ListDICOMImportJobs`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 ListDICOMImportJobs를 참조하세요. AWS

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

ListDatastores

다음 코드 예시는 ListDatastores의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListDatastores를 참조하세요.

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

ListImageSetVersions

다음 코드 예시는 ListImageSetVersions의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 ListImageSetVersions를 참조하세요. AWS

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

`ListTagsForResource`

다음 코드 예시는 `ListTagsForResource`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 ListTagsForResource를 참조하세요. AWS

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

`SearchImageSets`

다음 코드 예시는 `SearchImageSets`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

```
  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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 SearchImageSets를 참조하세요. AWS

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

`StartDICOMImportJob`

다음 코드 예시는 `StartDICOMImportJob`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

```
  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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 StartDICOMImportJob을 참조하세요. AWS

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

`TagResource`

다음 코드 예시는 `TagResource`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 TagResource를 참조하세요.

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

`UntagResource`

다음 코드 예시는 `UntagResource`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 UntagResource를 참조하세요.

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

UpdateImageSetMetadata

다음 코드 예시는 UpdateImageSetMetadata의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

```
   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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 UpdateImageSetMetadata를 참조하세요. AWS

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

SDK for SAP ABAP를 사용한 HealthLake 예제

다음 코드 예제에서는 HealthLake와 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`CreateFHIRDatastore`

다음 코드 예시는 `CreateFHIRDatastore`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 CreateFHIRDatastore를 참조하세요. AWS

`DeleteFHIRDatastore`

다음 코드 예시는 `DeleteFHIRDatastore`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DeleteFHIRDatastore를 참조하세요. AWS

`DescribeFHIRDatastore`

다음 코드 예시는 `DescribeFHIRDatastore`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DescribeFHIRDatastore를 참조하세요. AWS

`DescribeFHIRExportJob`

다음 코드 예시는 `DescribeFHIRExportJob`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DescribeFHIRExportJob을 참조하세요. AWS

`DescribeFHIRImportJob`

다음 코드 예시는 `DescribeFHIRImportJob`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DescribeFHIRImportJob을 참조하세요. AWS

`ListFHIRDatastores`

다음 코드 예시는 `ListFHIRDatastores`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 ListFHIRDatastores를 참조하세요. AWS

`ListFHIRExportJobs`

다음 코드 예시는 `ListFHIRExportJobs`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 ListFHIRExportJobs를 참조하세요. AWS

`ListFHIRImportJobs`

다음 코드 예시는 `ListFHIRImportJobs`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 ListFHIRImportJobs를 참조하세요. AWS

`ListTagsForResource`

다음 코드 예시는 `ListTagsForResource`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 ListTagsForResource를 참조하세요. AWS

`StartFHIRExportJob`

다음 코드 예시는 `StartFHIRExportJob`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 StartFHIRExportJob을 참조하세요. AWS

`StartFHIRImportJob`

다음 코드 예시는 `StartFHIRImportJob`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 StartFHIRImportJob을 참조하세요. AWS

`TagResource`

다음 코드 예시는 `TagResource`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 TagResource를 참조하세요.

`UntagResource`

다음 코드 예시는 `UntagResource`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 UntagResource를 참조하세요.

SDK for SAP ABAP를 사용한 IAM 예제

다음 코드 예제에서는 AWS SDK for SAP ABAP를 IAM과 함께 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`AttachRolePolicy`

다음 코드 예시는 `AttachRolePolicy`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 AttachRolePolicy를 참조하세요.

`AttachUserPolicy`

다음 코드 예시는 `AttachUserPolicy`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 AttachUserPolicy를 참조하세요.

`CreateAccessKey`

다음 코드 예시는 `CreateAccessKey`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateAccessKey를 참조하세요.

`CreateAccountAlias`

다음 코드 예시는 `CreateAccountAlias`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateAccountAlias를 참조하세요.

`CreatePolicy`

다음 코드 예시는 `CreatePolicy`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
  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.
```

- API 세부 정보는AWS SDK for SAP ABAP API의 CreatePolicy를 참조하세요.

`CreatePolicyVersion`

다음 코드 예시는 `CreatePolicyVersion`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreatePolicyVersion을 참조하세요.

`CreateRole`

다음 코드 예시는 `CreateRole`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateRole을 참조하세요.

`CreateServiceLinkedRole`

다음 코드 예시는 `CreateServiceLinkedRole`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateServiceLinkedRole을 참조하세요.

`CreateUser`

다음 코드 예시는 `CreateUser`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateUser를 참조하세요.

`DeleteAccessKey`

다음 코드 예시는 `DeleteAccessKey`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteAccessKey를 참조하세요.

`DeleteAccountAlias`

다음 코드 예시는 `DeleteAccountAlias`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteAccountAlias를 참조하세요.

`DeletePolicy`

다음 코드 예시는 `DeletePolicy`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeletePolicy를 참조하세요.

`DeletePolicyVersion`

다음 코드 예시는 `DeletePolicyVersion`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeletePolicyVersion을 참조하세요.

`DeleteRole`

다음 코드 예시는 `DeleteRole`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteRole을 참조하세요.

`DeleteUser`

다음 코드 예시는 `DeleteUser`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteUser를 참조하세요.

`DetachRolePolicy`

다음 코드 예시는 `DetachRolePolicy`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 DetachRolePolicy를 참조하세요.

`DetachUserPolicy`

다음 코드 예시는 `DetachUserPolicy`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 DetachUserPolicy를 참조하세요.

`GenerateCredentialReport`

다음 코드 예시는 `GenerateCredentialReport`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 GenerateCredentialReport를 참조하세요.

`GetAccessKeyLastUsed`

다음 코드 예시는 `GetAccessKeyLastUsed`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 GetAccessKeyLastUsed를 참조하세요.

`GetAccountAuthorizationDetails`

다음 코드 예시는 `GetAccountAuthorizationDetails`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 GetAccountPasswordPolicy를 참조하세요.

`GetAccountPasswordPolicy`

다음 코드 예시는 `GetAccountPasswordPolicy`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 GetAccountPasswordPolicy를 참조하세요.

`GetAccountSummary`

다음 코드 예시는 `GetAccountSummary`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 GetAccountSummary를 참조하세요.

`GetCredentialReport`

다음 코드 예시는 `GetCredentialReport`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 GetCredentialReport를 참조하세요.

`GetPolicy`

다음 코드 예시는 `GetPolicy`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 GetPolicy를 참조하세요.

`GetRole`

다음 코드 예시는 `GetRole`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 GetRole을 참조하세요.

`ListAccessKeys`

다음 코드 예시는 `ListAccessKeys`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListAccessKeys를 참조하세요.

`ListAccountAliases`

다음 코드 예시는 `ListAccountAliases`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SAP ABAP용AWS SDK API 참조의 ListAccountAliases를 참조하세요.

`ListAttachedRolePolicies`

다음 코드 예시는 `ListAttachedRolePolicies`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListAttachedRolePolicies를 참조하세요.

`ListGroups`

다음 코드 예시는 `ListGroups`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는AWS SDK for SAP ABAP API의 ListGroups르 참조하세요.

`ListPolicies`

다음 코드 예시는 `ListPolicies`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListPolicies를 참조하세요.

`ListPolicyVersions`

다음 코드 예시는 `ListPolicyVersions`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListPolicyVersions를 참조하세요.

`ListRolePolicies`

다음 코드 예시는 `ListRolePolicies`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListRolePolicies를 참조하세요.

`ListRoles`

다음 코드 예시는 `ListRoles`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListRoles를 참조하세요.

`ListSAMLProviders`

다음 코드 예시는 `ListSAMLProviders`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListSAMLProviders를 참조하세요.

`ListUsers`

다음 코드 예시는 `ListUsers`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API의 ListUsers을 참조하세요.

`SetDefaultPolicyVersion`

다음 코드 예시는 `SetDefaultPolicyVersion`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 SetDefaultPolicyVersion을 참조하세요.

`UpdateAccessKey`

다음 코드 예시는 `UpdateAccessKey`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 UpdateAccessKey를 참조하세요.

`UpdateUser`

다음 코드 예시는 `UpdateUser`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는AWS SDK for SAP ABAP API의 UpdateUser을 참조하세요.

AWS IoT SiteWise SDK for SAP ABAP를 사용한 예제

다음 코드 예제에서는 SAP ABAP용 AWS SDK를와 함께 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS IoT SiteWise.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 시작하기
- 작업

시작하기

안녕하세요 AWS IoT SiteWise

다음 코드 예제에서는 AWS IoT SiteWise를 사용하여 시작하는 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 ListAssetModels를 참조하세요. AWS

작업

`BatchPutAssetPropertyValue`

다음 코드 예시는 `BatchPutAssetPropertyValue`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 BatchPutAssetPropertyValue를 참조하세요. AWS

`CreateAsset`

다음 코드 예시는 `CreateAsset`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateAsset을 참조하세요.

`CreateAssetModel`

다음 코드 예시는 `CreateAssetModel`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateAssetModel을 참조하세요.

`CreateGateway`

다음 코드 예시는 `CreateGateway`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
  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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateGateway를 참조하세요.

`DeleteAsset`

다음 코드 예시는 `DeleteAsset`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteAsset을 참조하세요.

`DeleteAssetModel`

다음 코드 예시는 `DeleteAssetModel`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteAssetModel을 참조하세요.

`DeleteGateway`

다음 코드 예시는 `DeleteGateway`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteGateway를 참조하세요.

`DescribeGateway`

다음 코드 예시는 `DescribeGateway`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeGateway를 참조하세요.

`GetAssetPropertyValue`

다음 코드 예시는 `GetAssetPropertyValue`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 GetAssetPropertyValue를 참조하세요. AWS

`ListAssetModelProperties`

다음 코드 예시는 `ListAssetModelProperties`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 ListAssetModelProperties를 참조하세요. AWS

`ListAssetModels`

다음 코드 예시는 `ListAssetModels`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 ListAssetModels를 참조하세요. AWS

SDK for SAP ABAP를 사용한 Amazon Keyspaces 예제

다음 코드 예제에서는 Amazon Keyspaces와 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`CreateKeyspace`

다음 코드 예시는 `CreateKeyspace`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateKeyspace를 참조하세요.

`CreateTable`

다음 코드 예시는 `CreateTable`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는AWS SDK for SAP ABAP API의 CreateTable을 참조하세요.

`DeleteKeyspace`

다음 코드 예시는 `DeleteKeyspace`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteKeyspace를 참조하세요.

`DeleteTable`

다음 코드 예시는 `DeleteTable`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는AWS SDK for SAP ABAP API의 DeleteTable을 참조하세요.

`GetKeyspace`

다음 코드 예시는 `GetKeyspace`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 GetKeyspace를 참조하세요.

`GetTable`

다음 코드 예시는 `GetTable`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 GetTable을 참조하세요.

`ListKeyspaces`

다음 코드 예시는 `ListKeyspaces`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListKeyspaces를 참조하세요.

`ListTables`

다음 코드 예시는 `ListTables`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는AWS SDK for SAP ABAP API의 ListTables을 참조하세요.

`RestoreTable`

다음 코드 예시는 `RestoreTable`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 RestoreTable을 참조하세요.

`UpdateTable`

다음 코드 예시는 `UpdateTable`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 UpdateTable을 참조하세요.

SDK for SAP ABAP를 사용하는 Kinesis 예

다음 코드 예제에서는 Kinesis와 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

기본 사항은 서비스 내에서 필수 작업을 수행하는 방법을 보여주는 코드 예제입니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 기본 사항
- 작업

기본 사항

기본 사항 알아보기

다음 코드 예제에서는 다음과 같은 작업을 수행하는 방법을 보여줍니다.

- 스트림을 생성하고 그 안에 레코드를 넣습니다.
- 샤드 반복자를 생성합니다.
- 레코드를 읽은 다음 리소스를 정리합니다.

SDK for SAP ABAP

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 다음 주제를 참조하세요.
  - CreateStream
  - DeleteStream

- GetRecords
- GetShardIterator
- PutRecord

작업

`CreateStream`

다음 코드 예시는 `CreateStream`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateStream을 참조하세요.

`DeleteStream`

다음 코드 예시는 `DeleteStream`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteStream을 참조하세요.

`DescribeStream`

다음 코드 예시는 `DescribeStream`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeStream을 참조하세요.

`GetRecords`

다음 코드 예시는 `GetRecords`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    TRY.
        oo_result = lo_kns->getrecords(             " oo_result is returned for
testing purposes. "
            iv_sharditerator = iv_shard_iterator ).
        DATA(lt_records) = oo_result->get_records( ).
        MESSAGE 'Record retrieved.' TYPE 'I'.
      CATCH /aws1/cx_knsexpirediteratorex.
        MESSAGE 'Iterator expired.' TYPE 'E'.
      CATCH /aws1/cx_knsinvalidargumentex.
        MESSAGE 'The specified argument was not valid.' TYPE 'E'.
      CATCH /aws1/cx_knskmsaccessdeniedex.
        MESSAGE 'You do not have permission to perform this AWS KMS action.' TYPE
'E'.
      CATCH /aws1/cx_knskmsdisabledex.
        MESSAGE 'KMS key used is disabled.' TYPE 'E'.
      CATCH /aws1/cx_knskmsinvalidstateex.
        MESSAGE 'KMS key used is in an invalid state. ' TYPE 'E'.
      CATCH /aws1/cx_knskmsnotfoundex.
        MESSAGE 'KMS key used is not found.' TYPE 'E'.
```

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

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 GetRecords를 참조하세요.

`ListStreams`

다음 코드 예시는 `ListStreams`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListStreams을 참조하세요.

PutRecord

다음 코드 예시는 PutRecord의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 PutRecord를 참조하세요.

`RegisterStreamConsumer`

다음 코드 예시는 `RegisterStreamConsumer`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 RegisterStreamConsumer를 참조하세요.

AWS KMS SDK for SAP ABAP를 사용한 예제

다음 코드 예제에서는 AWS SDK for SAP ABAP를와 함께 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS KMS.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 시작하기
- 작업

시작하기

안녕하세요 AWS KMS

다음 코드 예제에서는 AWS Key Management Service를 사용하여 시작하는 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListKeys를 참조하세요.

작업

`CreateAlias`

다음 코드 예시는 `CreateAlias`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateAlias를 참조하세요.

`CreateGrant`

다음 코드 예시는 `CreateGrant`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateGrant를 참조하세요.

`CreateKey`

다음 코드 예시는 `CreateKey`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateKey를 참조하세요.

`Decrypt`

다음 코드 예시는 `Decrypt`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 Decrypt를 참조하세요.

`DeleteAlias`

다음 코드 예시는 `DeleteAlias`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteAlias를 참조하세요.

`DescribeKey`

다음 코드 예시는 `DescribeKey`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeKey를 참조하세요.

`DisableKey`

다음 코드 예시는 `DisableKey`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DisableKey를 참조하세요.

`EnableKey`

다음 코드 예시는 `EnableKey`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 EnableKey를 참조하세요.

`EnableKeyRotation`

다음 코드 예시는 `EnableKeyRotation`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 EnableKeyRotation을 참조하세요.

`Encrypt`

다음 코드 예시는 `Encrypt`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 암호화를 참조하세요.

`GenerateDataKey`

다음 코드 예시는 `GenerateDataKey`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 GenerateDataKey를 참조하세요.

`GetKeyPolicy`

다음 코드 예시는 `GetKeyPolicy`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 GetKeyPolicy를 참조하세요. AWS

`ListAliases`

다음 코드 예시는 `ListAliases`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListAliases를 참조하세요.

`ListGrants`

다음 코드 예시는 `ListGrants`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListGrants를 참조하세요.

`ListKeyPolicies`

다음 코드 예시는 `ListKeyPolicies`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListKeyPolicies를 참조하세요.

`ListKeys`

다음 코드 예시는 `ListKeys`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListKeys를 참조하세요.

`PutKeyPolicy`

다음 코드 예시는 `PutKeyPolicy`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 PutKeyPolicy를 참조하세요.

`ReEncrypt`

다음 코드 예시는 `ReEncrypt`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ReEncrypt를 참조하세요.

`RetireGrant`

다음 코드 예시는 `RetireGrant`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 RetireGrant를 참조하세요.

`RevokeGrant`

다음 코드 예시는 `RevokeGrant`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 RevokeGrant를 참조하세요. AWS

`ScheduleKeyDeletion`

다음 코드 예시는 `ScheduleKeyDeletion`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 ScheduleKeyDeletion을 참조하세요. AWS

`Sign`

다음 코드 예시는 `Sign`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 Sign in AWS SDK for SAP ABAP API 참조를 참조하세요.

`TagResource`

다음 코드 예시는 `TagResource`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 TagResource를 참조하세요.

`UpdateAlias`

다음 코드 예시는 `UpdateAlias`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 UpdateAlias를 참조하세요.

`Verify`

다음 코드 예시는 `Verify`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 확인을 참조하세요.

SDK for SAP ABAP를 사용한 Lambda 예

다음 코드 예제에서는 Lambda와 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

기본 사항은 서비스 내에서 필수 작업을 수행하는 방법을 보여주는 코드 예제입니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 기본 사항
- 작업

기본 사항

기본 사항 알아보기

다음 코드 예제에서는 다음과 같은 작업을 수행하는 방법을 보여줍니다.

- IAM 역할과 Lambda 함수를 생성하고 핸들러 코드를 업로드합니다.
- 단일 파라미터로 함수를 간접적으로 간접 호출하고 결과를 가져옵니다.
- 함수 코드를 업데이트하고 환경 변수로 구성합니다.
- 새 파라미터로 함수를 간접적으로 간접 호출하고 결과를 가져옵니다. 반환된 실행 로그를 표시합니다.
- 계정의 함수를 나열합니다.

자세한 내용은 콘솔로 Lambda 함수 생성을 참조하세요.

SDK for SAP ABAP

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 다음 주제를 참조하세요.
  - CreateFunction
  - DeleteFunction
  - GetFunction
  - 간접 호출
  - ListFunctions
  - UpdateFunctionCode
  - UpdateFunctionConfiguration

작업

`CreateFunction`

다음 코드 예시는 `CreateFunction`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateFunction을 참조하세요.

`DeleteFunction`

다음 코드 예시는 `DeleteFunction`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteFunction을 참조하세요.

`GetFunction`

다음 코드 예시는 `GetFunction`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 GetFunction을 참조하세요.

`Invoke`

다음 코드 예시는 `Invoke`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API에 대한 세부 정보는 SAP ABAP용AWS SDK API 참조의 간접 호출를 참조하세요.

`ListFunctions`

다음 코드 예시는 `ListFunctions`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListFunctions를 참조하세요.

`UpdateFunctionCode`

다음 코드 예시는 `UpdateFunctionCode`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 UpdateFunctionCode를 참조하세요.

`UpdateFunctionConfiguration`

다음 코드 예시는 `UpdateFunctionConfiguration`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 UpdateFunctionConfiguration을 참조하세요.

SDK for SAP ABAP를 사용한 조직 예제

다음 코드 예제에서는 Organizations와 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`AttachPolicy`

다음 코드 예시는 `AttachPolicy`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 AttachPolicy를 참조하세요.

`CreatePolicy`

다음 코드 예시는 `CreatePolicy`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API 세부 정보는AWS SDK for SAP ABAP API의 CreatePolicy를 참조하세요.

`DeletePolicy`

다음 코드 예시는 `DeletePolicy`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeletePolicy를 참조하세요.

`DescribePolicy`

다음 코드 예시는 `DescribePolicy`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribePolicy를 참조하세요.

`DetachPolicy`

다음 코드 예시는 `DetachPolicy`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DetachPolicy를 참조하세요.

`ListPolicies`

다음 코드 예시는 `ListPolicies`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListPolicies를 참조하세요.

SDK for SAP ABAP를 사용한 Amazon Pinpoint 예제

다음 코드 예제에서는 Amazon Pinpoint와 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`SendMessages`

다음 코드 예시는 `SendMessages`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

이메일 메시지를 전송합니다.

```
" 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'.
```

SMS 메시지를 전송합니다.

```
  " 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'.
```

기존 이메일 템플릿을 사용하여 이메일 메시지를 전송합니다.

```
    " 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'.
```

기존 SMS 템플릿을 사용하여 문자 메시지를 전송합니다.

```
" 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'.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 SendMessages를 참조하세요.

SDK for SAP ABAP를 사용한 Amazon Pinpoint SMS 및 음성 API 예제

다음 코드 예제에서는 Amazon Pinpoint SMS 및 음성 API와 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`CreateConfigurationSet`

다음 코드 예시는 `CreateConfigurationSet`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 CreateConfigurationSet를 참조하세요. AWS

`CreateConfigurationSetEventDestination`

다음 코드 예시는 `CreateConfigurationSetEventDestination`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 CreateConfigurationSetEventDestination을 참조하세요. AWS

`DeleteConfigurationSet`

다음 코드 예시는 `DeleteConfigurationSet`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DeleteConfigurationSet를 참조하세요. AWS

`DeleteConfigurationSetEventDestination`

다음 코드 예시는 `DeleteConfigurationSetEventDestination`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DeleteConfigurationSetEventDestination을 참조하세요. AWS

`GetConfigurationSetEventDestinations`

다음 코드 예시는 `GetConfigurationSetEventDestinations`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 GetConfigurationSetEventDestinations를 참조하세요. AWS

`ListConfigurationSets`

다음 코드 예시는 `ListConfigurationSets`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 ListConfigurationSets를 참조하세요. AWS

`SendVoiceMessage`

다음 코드 예시는 `SendVoiceMessage`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 SendVoiceMessage를 참조하세요. AWS

`UpdateConfigurationSetEventDestination`

다음 코드 예시는 `UpdateConfigurationSetEventDestination`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 UpdateConfigurationSetEventDestination을 참조하세요. AWS

SDK for SAP ABAP를 사용한 Amazon Polly 예제

다음 코드 예제에서는 Amazon Polly와 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`DeleteLexicon`

다음 코드 예시는 `DeleteLexicon`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteLexicon을 참조하세요.

`DescribeVoices`

다음 코드 예시는 `DescribeVoices`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeVoices를 참조하세요.

`GetLexicon`

다음 코드 예시는 `GetLexicon`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
  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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 GetLexicon을 참조하세요. AWS

`GetSpeechSynthesisTask`

다음 코드 예시는 `GetSpeechSynthesisTask`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 GetSpeechSynthesisTask를 참조하세요. AWS

`ListLexicons`

다음 코드 예시는 `ListLexicons`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListLexicons를 참조하세요.

`ListSpeechSynthesisTasks`

다음 코드 예시는 `ListSpeechSynthesisTasks`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 ListSpeechSynthesisTasks를 참조하세요. AWS

`PutLexicon`

다음 코드 예시는 `PutLexicon`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 PutLexicon을 참조하세요.

`StartSpeechSynthesisTask`

다음 코드 예시는 `StartSpeechSynthesisTask`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 StartSpeechSynthesisTask를 참조하세요. AWS

`SynthesizeSpeech`

다음 코드 예시는 `SynthesizeSpeech`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 SynthesizeSpeech를 참조하세요. AWS

SDK for SAP ABAP를 사용한 Amazon RDS 예제

다음 코드 예제에서는 Amazon RDS와 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`CreateDBParameterGroup`

다음 코드 예시는 `CreateDBParameterGroup`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
" 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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateDBParameterGroup을 참조하세요.

`DeleteDBParameterGroup`

다음 코드 예시는 `DeleteDBParameterGroup`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
" 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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteDBParameterGroup을 참조하세요.

`DescribeDBEngineVersions`

다음 코드 예시는 `DescribeDBEngineVersions`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    " 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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeDBEngineVersions를 참조하세요.

`DescribeDBParameterGroups`

다음 코드 예시는 `DescribeDBParameterGroups`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    " 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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeDBParameterGroups를 참조하세요.

`DescribeDBParameters`

다음 코드 예시는 `DescribeDBParameters`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
" 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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeDBParameters를 참조하세요.

`DescribeOrderableDBInstanceOptions`

다음 코드 예시는 `DescribeOrderableDBInstanceOptions`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    " 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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeOrderableDBInstanceOptions를 참조하세요.

`ModifyDBParameterGroup`

다음 코드 예시는 `ModifyDBParameterGroup`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    " 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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ModifyDBParameterGroup을 참조하세요.

SDK for SAP ABAP를 사용한 Amazon Redshift 예제

다음 코드 예제에서는 Amazon Redshift와 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`CreateCluster`

다음 코드 예시는 `CreateCluster`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

클러스터를 생성합니다.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateCluster를 참조하세요.

`DeleteCluster`

다음 코드 예시는 `DeleteCluster`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

클러스터를 삭제합니다.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteCluster를 참조하세요.

`DescribeClusters`

다음 코드 예시는 `DescribeClusters`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

클러스터를 설명하세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeClusters를 참조하세요.

`DescribeStatement`

다음 코드 예시는 `DescribeStatement`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeStatement를 참조하세요.

`ExecuteStatement`

다음 코드 예시는 `ExecuteStatement`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ExecuteStatement를 참조하세요.

`GetStatementResult`

다음 코드 예시는 `GetStatementResult`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

문 결과를 확인합니다.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 GetStatementResult를 참조하세요.

`ListDatabases`

다음 코드 예시는 `ListDatabases`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListDatabases를 참조하세요.

`ModifyCluster`

다음 코드 예시는 `ModifyCluster`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

클러스터를 수정합니다.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ModifyCluster을 참조하세요.

SDK for SAP ABAP를 사용한 Amazon Rekognition 예제

다음 코드 예제에서는 Amazon Rekognition과 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`CompareFaces`

다음 코드 예시는 `CompareFaces`의 사용 방법을 보여줍니다.

자세한 내용은 이미지에 있는 얼굴 비교를 참조하세요.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CompareFaces를 참조하세요.

`CreateCollection`

다음 코드 예시는 `CreateCollection`의 사용 방법을 보여 줍니다.

자세한 내용은 컬렉션 생성을 참조하세요.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateCollection을 참조하세요.

`DeleteCollection`

다음 코드 예시는 `DeleteCollection`의 사용 방법을 보여 줍니다.

자세한 내용은 컬렉션 삭제를 참조하세요.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteCollection을 참조하세요.

`DeleteFaces`

다음 코드 예시는 `DeleteFaces`의 사용 방법을 보여 줍니다.

자세한 내용은 컬렉션에서 얼굴 삭제를 참조하세요.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteFaces를 참조하세요.

`DescribeCollection`

다음 코드 예시는 `DescribeCollection`의 사용 방법을 보여 줍니다.

자세한 내용은 컬렉션 설명을 참조하세요.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeCollection을 참조하세요.

`DetectFaces`

다음 코드 예시는 `DetectFaces`의 사용 방법을 보여 줍니다.

자세한 내용은 이미지에서 얼굴 감지를 참조하세요.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DetectFaces를 참조하세요.

`DetectLabels`

다음 코드 예시는 `DetectLabels`의 사용 방법을 보여 줍니다.

자세한 내용은 이미지에서 레이블 감지를 참조하세요.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DetectLabels를 참조하세요.

`DetectModerationLabels`

다음 코드 예시는 `DetectModerationLabels`의 사용 방법을 보여 줍니다.

자세한 내용은 부적절한 이미지 감지를 참조하세요.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DetectModerationLabels를 참조하세요. AWS

`DetectText`

다음 코드 예시는 `DetectText`의 사용 방법을 보여 줍니다.

자세한 내용은 이미지에서 텍스트 감지를 참조하세요.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DetectText를 참조하세요.

`IndexFaces`

다음 코드 예시는 `IndexFaces`의 사용 방법을 보여 줍니다.

자세한 내용은 컬렉션에 얼굴 추가를 참조하세요.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 IndexFaces를 참조하세요.

`ListCollections`

다음 코드 예시는 `ListCollections`의 사용 방법을 보여 줍니다.

자세한 내용은 컬렉션 나열을 참조하세요.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListCollections를 참조하세요.

`ListFaces`

다음 코드 예시는 `ListFaces`의 사용 방법을 보여 줍니다.

자세한 내용은 컬렉션 내 얼굴 나열을 참조하세요.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListFaces를 참조하세요.

`RecognizeCelebrities`

다음 코드 예시는 `RecognizeCelebrities`의 사용 방법을 보여 줍니다.

자세한 내용은 이미지에서 유명인 인식을 참조하세요.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 RecognizeCelebrities를 참조하세요.

`SearchFaces`

다음 코드 예시는 `SearchFaces`의 사용 방법을 보여 줍니다.

자세한 내용은 얼굴 검색(얼굴 ID)을 참조하세요.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 SearchFaces를 참조하세요.

`SearchFacesByImage`

다음 코드 예시는 `SearchFacesByImage`의 사용 방법을 보여 줍니다.

자세한 내용은 얼굴 검색(이미지)을 참조하세요.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 SearchFacesByImage를 참조하세요. AWS

SDK for SAP ABAP를 사용한 Amazon S3 예

다음 코드 예제에서는 Amazon S3에서 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

기본 사항은 서비스 내에서 필수 작업을 수행하는 방법을 보여주는 코드 예제입니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접 호출하는 방법을 보여주며, 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

시나리오는 동일한 서비스 내에서 또는 다른 AWS 서비스와 결합된 상태에서 여러 함수를 직접적으로 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예제입니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 기본 사항
- 작업
- 시나리오

기본 사항

기본 사항 알아보기

다음 코드 예제에서는 다음과 같은 작업을 수행하는 방법을 보여줍니다.

- 버킷을 만들고 버킷에 파일을 업로드합니다.
- 버킷에서 객체를 다운로드합니다.
- 버킷의 하위 폴더에 객체를 복사합니다.
- 버킷의 객체를 나열합니다.
- 버킷 객체와 버킷을 삭제합니다.

SDK for SAP ABAP

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 다음 주제를 참조하세요.
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

작업

`CopyObject`

다음 코드 예시는 `CopyObject`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CopyObject를 참조하세요.

`CreateBucket`

다음 코드 예시는 `CreateBucket`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는AWS SDK for SAP ABAP API의 CreateBucket을 참조하세요.

`DeleteBucket`

다음 코드 예시는 `DeleteBucket`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteBucket을 참조하세요.

`DeleteBucketCors`

다음 코드 예시는 `DeleteBucketCors`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DeleteBucketCors를 참조하세요. AWS

`DeleteBucketLifecycle`

다음 코드 예시는 `DeleteBucketLifecycle`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
  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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DeleteBucketLifecycle을 참조하세요. AWS

`DeleteBucketPolicy`

다음 코드 예시는 `DeleteBucketPolicy`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DeleteBucketPolicy를 참조하세요. AWS

`DeleteObject`

다음 코드 예시는 `DeleteObject`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteObject를 참조하세요.

`DeleteObjects`

다음 코드 예시는 `DeleteObjects`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteObjects를 참조하세요.

`GetBucketAcl`

다음 코드 예시는 `GetBucketAcl`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 GetBucketAcl을 참조하세요. AWS

`GetBucketCors`

다음 코드 예시는 `GetBucketCors`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 GetBucketCors를 참조하세요. AWS

`GetBucketLifecycleConfiguration`

다음 코드 예시는 `GetBucketLifecycleConfiguration`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 GetBucketLifecycleConfiguration을 참조하세요. AWS

`GetBucketPolicy`

다음 코드 예시는 `GetBucketPolicy`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 GetBucketPolicy를 참조하세요. AWS

`GetObject`

다음 코드 예시는 `GetObject`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 GetObject를 참조하세요.

`GetObjectAcl`

다음 코드 예시는 `GetObjectAcl`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 GetObjectAcl을 참조하세요. AWS

`GetObjectLegalHold`

다음 코드 예시는 `GetObjectLegalHold`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 GetObjectLegalHold를 참조하세요. AWS

`GetObjectLockConfiguration`

다음 코드 예시는 `GetObjectLockConfiguration`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 GetObjectLockConfiguration을 참조하세요. AWS

`HeadBucket`

다음 코드 예시는 `HeadBucket`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 HeadBucket을 참조하세요.

`ListObjectVersions`

다음 코드 예시는 `ListObjectVersions`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 ListObjectVersions를 참조하세요. AWS

`ListObjectsV2`

다음 코드 예시는 `ListObjectsV2`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListObjectsV2를 참조하세요.

`PutBucketAcl`

다음 코드 예시는 `PutBucketAcl`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 PutBucketAcl을 참조하세요. AWS

`PutBucketCors`

다음 코드 예시는 `PutBucketCors`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 PutBucketCors를 참조하세요. AWS

`PutBucketLifecycleConfiguration`

다음 코드 예시는 `PutBucketLifecycleConfiguration`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 PutBucketLifecycleConfiguration을 참조하세요. AWS

`PutBucketPolicy`

다음 코드 예시는 `PutBucketPolicy`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 PutBucketPolicy를 참조하세요. AWS

`PutBucketVersioning`

다음 코드 예시는 `PutBucketVersioning`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 PutBucketVersioning을 참조하세요.

`PutObject`

다음 코드 예시는 `PutObject`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
"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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 PutObject를 참조하세요.

`PutObjectAcl`

다음 코드 예시는 `PutObjectAcl`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 PutObjectAcl을 참조하세요. AWS

`PutObjectLegalHold`

다음 코드 예시는 `PutObjectLegalHold`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 PutObjectLegalHold를 참조하세요. AWS

`PutObjectLockConfiguration`

다음 코드 예시는 `PutObjectLockConfiguration`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 PutObjectLockConfiguration을 참조하세요. AWS

`PutObjectRetention`

다음 코드 예시는 `PutObjectRetention`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 PutObjectRetention을 참조하세요. AWS

시나리오

미리 서명된 URL 생성

다음 코드 예제는 Amazon S3에 대해 미리 서명된 URL을 생성하고 객체를 업로드하는 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

GET S3 객체에 대한 사전 서명된 요청을 만듭니다.

```
" 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( ).
```

SDK for SAP ABAP를 사용한 SageMaker AI 예제

다음 코드 예제에서는 SageMaker AI와 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접 호출하는 방법을 보여주며, 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

시나리오는 동일한 서비스 내에서 또는 다른 AWS 서비스와 결합된 상태에서 여러 함수를 직접적으로 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예제입니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업
- 시나리오

작업

`CreateEndpoint`

다음 코드 예시는 `CreateEndpoint`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateEndpoint를 참조하세요.

`CreateModel`

다음 코드 예시는 `CreateModel`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateModel을 참조하세요.

`CreateTrainingJob`

다음 코드 예시는 `CreateTrainingJob`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateTrainingJob을 참조하세요.

`CreateTransformJob`

다음 코드 예시는 `CreateTransformJob`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateTransformJob을 참조하세요.

`DeleteEndpoint`

다음 코드 예시는 `DeleteEndpoint`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    "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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteEndpoint를 참조하세요.

`DeleteModel`

다음 코드 예시는 `DeleteModel`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteModel을 참조하세요.

`DescribeTrainingJob`

다음 코드 예시는 `DescribeTrainingJob`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeTrainingJob을 참조하세요.

`ListAlgorithms`

다음 코드 예시는 `ListAlgorithms`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListAlgorithms를 참조하세요.

`ListModels`

다음 코드 예시는 `ListModels`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListModels를 참조하세요.

`ListNotebookInstances`

다음 코드 예시는 `ListNotebookInstances`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListNotebookInstances를 참조하세요.

`ListTrainingJobs`

다음 코드 예시는 `ListTrainingJobs`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListTrainingJobs를 참조하세요.

시나리오

모델 및 엔드포인트를 사용하여 시작하기

다음 코드 예제에서는 다음과 같은 작업을 수행하는 방법을 보여줍니다.

- 훈련 작업을 시작하고 SageMaker AI 모델을 만듭니다.
- 엔드포인트 구성을 생성합니다.
- 엔드포인트를 생성한 다음 리소스를 정리합니다.

SDK for SAP ABAP

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 다음 주제를 참조하세요.
  - CreateEndpoint
  - CreateEndpointConfig
  - CreateModel
  - CreateTrainingJob
  - DeleteEndpoint
  - DeleteEndpointConfig
  - DeleteModel
  - DescribeEndpoint
  - DescribeTrainingJob

SDK for SAP ABAP를 사용한 Secrets Manager 예제

다음 코드 예제에서는 Secrets Manager와 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`BatchGetSecretValue`

다음 코드 예시는 `BatchGetSecretValue`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 BatchGetSecretValue를 참조하세요. AWS

`GetSecretValue`

다음 코드 예시는 `GetSecretValue`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 GetSecretValue를 참조하세요.

SDK for SAP ABAP를 사용한 Amazon SES 예제

다음 코드 예제에서는 Amazon SES와 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`CreateReceiptFilter`

다음 코드 예시는 `CreateReceiptFilter`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
" 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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateReceiptFilter를 참조하세요.

`CreateReceiptRule`

다음 코드 예시는 `CreateReceiptRule`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
" 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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateReceiptRule을 참조하세요.

`CreateReceiptRuleSet`

다음 코드 예시는 `CreateReceiptRuleSet`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 CreateReceiptRuleSet를 참조하세요. AWS

`CreateTemplate`

다음 코드 예시는 `CreateTemplate`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateTemplate을 참조하세요.

`DeleteIdentity`

다음 코드 예시는 `DeleteIdentity`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteIdentity를 참조하세요.

`DeleteReceiptFilter`

다음 코드 예시는 `DeleteReceiptFilter`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteReceiptFilter를 참조하세요.

`DeleteReceiptRule`

다음 코드 예시는 `DeleteReceiptRule`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteReceiptRule을 참조하세요.

`DeleteReceiptRuleSet`

다음 코드 예시는 `DeleteReceiptRuleSet`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DeleteReceiptRuleSet를 참조하세요. AWS

`DeleteTemplate`

다음 코드 예시는 `DeleteTemplate`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteTemplate을 참조하세요.

`DescribeReceiptRuleSet`

다음 코드 예시는 `DescribeReceiptRuleSet`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DescribeReceiptRuleSet를 참조하세요. AWS

`GetIdentityVerificationAttributes`

다음 코드 예시는 `GetIdentityVerificationAttributes`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 GetIdentityVerificationAttributes를 참조하세요. AWS

`GetTemplate`

다음 코드 예시는 `GetTemplate`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 GetTemplate을 참조하세요.

`ListIdentities`

다음 코드 예시는 `ListIdentities`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListIdentities를 참조하세요.

`ListReceiptFilters`

다음 코드 예시는 `ListReceiptFilters`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 ListReceiptFilters를 참조하세요. AWS

`ListTemplates`

다음 코드 예시는 `ListTemplates`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListTemplates를 참조하세요.

`SendEmail`

다음 코드 예시는 `SendEmail`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
" 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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 SendEmail을 참조하세요.

SendTemplatedEmail

다음 코드 예시는 SendTemplatedEmail의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 SendTemplatedEmail을 참조하세요. AWS

`UpdateTemplate`

다음 코드 예시는 `UpdateTemplate`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 UpdateTemplate을 참조하세요.

`VerifyDomainIdentity`

다음 코드 예시는 `VerifyDomainIdentity`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 VerifyDomainIdentity를 참조하세요. AWS

`VerifyEmailIdentity`

다음 코드 예시는 `VerifyEmailIdentity`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 VerifyEmailIdentity를 참조하세요. AWS

SDK for SAP ABAP를 사용한 Amazon SES API v2 예제

다음 코드 예제에서는 Amazon SES API v2와 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`CreateContact`

다음 코드 예시는 `CreateContact`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateContact를 참조하세요.

`CreateContactList`

다음 코드 예시는 `CreateContactList`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
  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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateContactList를 참조하세요.

`CreateEmailIdentity`

다음 코드 예시는 `CreateEmailIdentity`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 CreateEmailIdentity를 참조하세요. AWS

`CreateEmailTemplate`

다음 코드 예시는 `CreateEmailTemplate`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 CreateEmailTemplate을 참조하세요. AWS

`DeleteContactList`

다음 코드 예시는 `DeleteContactList`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteContactList를 참조하세요.

`DeleteEmailIdentity`

다음 코드 예시는 `DeleteEmailIdentity`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DeleteEmailIdentity를 참조하세요. AWS

`DeleteEmailTemplate`

다음 코드 예시는 `DeleteEmailTemplate`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DeleteEmailTemplate을 참조하세요. AWS

`ListContacts`

다음 코드 예시는 `ListContacts`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListContacts를 참조하세요.

`SendEmail`

다음 코드 예시는 `SendEmail`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

메시지를 전송합니다.

```
    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.
```

템플릿을 사용하여 메시지를 보냅니다.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 SendEmail을 참조하세요.

SDK for SAP ABAP를 사용한 Amazon SNS에 대한 코드 예제

다음 코드 예제에서는 Amazon SNS와 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접 호출하는 방법을 보여주며, 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

시나리오는 동일한 서비스 내에서 또는 다른 AWS 서비스와 결합된 상태에서 여러 함수를 직접적으로 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예제입니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업
- 시나리오

작업

`CreateTopic`

다음 코드 예시는 `CreateTopic`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API에 대한 세부 정보는 SAP ABAP용AWS SDK API 참조의 CreateTopic을 참조하세요.

`DeleteTopic`

다음 코드 예시는 `DeleteTopic`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API에 대한 세부 정보는 SAP ABAP용AWS SDK API 참조의 DeleteTopic을 참조하세요.

`GetTopicAttributes`

다음 코드 예시는 `GetTopicAttributes`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API에 대한 세부 정보는 SAP ABAP용AWS SDK API 참조의 GetTopicAttributes를 참조하세요.

`ListSubscriptions`

다음 코드 예시는 `ListSubscriptions`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API에 대한 세부 정보는 SAP ABAP용AWS SDK API 참조의 ListSubscriptions를 참조하세요.

`ListTopics`

다음 코드 예시는 `ListTopics`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API에 대한 세부 정보는 SAP ABAP용AWS SDK API 참조의 ListTopics를 참조하세요.

`Publish`

다음 코드 예시는 `Publish`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

속성이 포함된 메시지를 주제에 게시합니다.

```
   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.
```

주제에 다중 형식 메시지를 게시합니다.

```
   " 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.
```

- API에 대한 세부 정보는 SAP ABAP용AWS SDK API 참조의 Publish를 참조하세요.

`SetSubscriptionAttributes`

다음 코드 예시는 `SetSubscriptionAttributes`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 SetSubscriptionAttributes를 참조하세요. AWS

`SetTopicAttributes`

다음 코드 예시는 `SetTopicAttributes`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API에 대한 세부 정보는 SAP ABAP용AWS SDK API 참조의 SetTopicAttributes를 참조하세요.

`Subscribe`

다음 코드 예시는 `Subscribe`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

이메일 주소로 주제 구독.

```
    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.
```

- API에 대한 세부 정보는 SAP ABAP용AWS SDK API 참조의 Subscribe를 참조하세요.

`Unsubscribe`

다음 코드 예시는 `Unsubscribe`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API에 대한 세부 정보는 SAP ABAP용AWS SDK API 참조의 Unsubscribe를 참조하세요.

시나리오

FIFO 주제에 생성 및 게시

다음 코드 예제에서는 FIFO Amazon SNS 주제를 생성하고 거기에 게시하는 방법을 보여줍니다.

SDK for SAP ABAP

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

FIFO 주제를 생성하고, 주제에 대한 Amazon SQS FIFO 대기열을 구독하고, Amazon SNS 주제에 메시지를 게시합니다.

```
    " 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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 다음 주제를 참조하세요.
  - CreateTopic
  - 게시
  - Subscribe

SMS 문자 메시지 게시

다음 코드 예제에서는 Amazon SNS를 사용하여 SMS 메시지를 게시하는 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    " 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.
```

- API에 대한 세부 정보는 SAP ABAP용AWS SDK API 참조의 Publish를 참조하세요.

SDK for SAP ABAP를 사용한 Amazon SQS 예

다음 코드 예제에서는 Amazon SQS와 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접 호출하는 방법을 보여주며, 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

시나리오는 동일한 서비스 내에서 또는 다른 AWS 서비스와 결합된 상태에서 여러 함수를 직접적으로 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예제입니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업
- 시나리오

작업

`CreateQueue`

다음 코드 예시는 `CreateQueue`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

Amazon SQS 표준 대기열을 생성합니다.

```
    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.
```

메시지가 도착하기를 기다리는 Amazon SQS 대기열을 생성합니다.

```
    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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateQueue를 참조하세요.

`DeleteMessage`

다음 코드 예시는 `DeleteMessage`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteMessage를 참조하세요.

`DeleteMessageBatch`

다음 코드 예시는 `DeleteMessageBatch`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DeleteMessageBatch를 참조하세요. AWS

`DeleteQueue`

다음 코드 예시는 `DeleteQueue`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

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

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteQueue를 참조하세요.

`GetQueueUrl`

다음 코드 예시는 `GetQueueUrl`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 GetQueueUrl을 참조하세요.

`ListQueues`

다음 코드 예시는 `ListQueues`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

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

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListQueues를 참조하세요.

`ReceiveMessage`

다음 코드 예시는 `ReceiveMessage`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

Amazon SQS 대기열에서 메시지를 수신합니다.

```
   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.
```

긴 폴링 지원을 사용하여 Amazon SQS 대기열에서 메시지를 수신합니다.

```
   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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 ReceiveMessage를 참조하세요.

SendMessage

다음 코드 예시는 SendMessage의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 SendMessage를 참조하세요.

SendMessageBatch

다음 코드 예시는 SendMessageBatch의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 SendMessageBatch를 참조하세요. AWS

시나리오

FIFO 주제에 생성 및 게시

다음 코드 예제에서는 FIFO Amazon SNS 주제를 생성하고 거기에 게시하는 방법을 보여줍니다.

SDK for SAP ABAP

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

FIFO 주제를 생성하고, 주제에 대한 Amazon SQS FIFO 대기열을 구독하고, Amazon SNS 주제에 메시지를 게시합니다.

```
    " 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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 다음 주제를 참조하세요.
  - CreateTopic
  - 게시
  - Subscribe

SDK for SAP ABAP를 사용한 Step Functions 예제

다음 코드 예제에서는 Step Functions와 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`CreateActivity`

다음 코드 예시는 `CreateActivity`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateActivity를 참조하세요.

`CreateStateMachine`

다음 코드 예시는 `CreateStateMachine`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateStateMachine을 참조하세요.

`DeleteActivity`

다음 코드 예시는 `DeleteActivity`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteActivity를 참조하세요.

`DeleteStateMachine`

다음 코드 예시는 `DeleteStateMachine`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteStateMachine을 참조하세요.

`DescribeExecution`

다음 코드 예시는 `DescribeExecution`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeExecution을 참조하세요.

`DescribeStateMachine`

다음 코드 예시는 `DescribeStateMachine`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeStateMachine을 참조하세요.

`GetActivityTask`

다음 코드 예시는 `GetActivityTask`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 GetActivityTask를 참조하세요. AWS

`ListActivities`

다음 코드 예시는 `ListActivities`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListActivities를 참조하세요.

`ListStateMachines`

다음 코드 예시는 `ListStateMachines`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 ListStateMachines를 참조하세요. AWS

`SendTaskSuccess`

다음 코드 예시는 `SendTaskSuccess`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 SendTaskSuccess를 참조하세요. AWS

`StartExecution`

다음 코드 예시는 `StartExecution`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
  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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 StartExecution을 참조하세요.

SDK for SAP ABAP를 사용한 Systems Manager 예제

다음 코드 예제에서는 Systems Manager와 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`CreateDocument`

다음 코드 예시는 `CreateDocument`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateDocument를 참조하세요.

`CreateMaintenanceWindow`

다음 코드 예시는 `CreateMaintenanceWindow`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateMaintenanceWindow를 참조하세요.

`CreateOpsItem`

다음 코드 예시는 `CreateOpsItem`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 CreateOpsItem을 참조하세요.

`DeleteDocument`

다음 코드 예시는 `DeleteDocument`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteDocument를 참조하세요.

`DeleteMaintenanceWindow`

다음 코드 예시는 `DeleteMaintenanceWindow`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
  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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteMaintenanceWindow를 참조하세요.

`DeleteOpsItem`

다음 코드 예시는 `DeleteOpsItem`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 DeleteOpsItem을 참조하세요.

`DescribeDocument`

다음 코드 예시는 `DescribeDocument`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeDocument를 참조하세요.

`DescribeOpsItems`

다음 코드 예시는 `DescribeOpsItems`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeOpsItems를 참조하세요.

`ListCommandInvocations`

다음 코드 예시는 `ListCommandInvocations`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListCommandInvocations를 참조하세요.

SendCommand

다음 코드 예시는 SendCommand의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 SendCommand를 참조하세요.

`UpdateMaintenanceWindow`

다음 코드 예시는 `UpdateMaintenanceWindow`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 UpdateMaintenanceWindow를 참조하세요.

`UpdateOpsItem`

다음 코드 예시는 `UpdateOpsItem`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 UpdateOpsItem을 참조하세요.

SDK for SAP ABAP를 사용한 Amazon Textract 예

다음 코드 예제에서는 Amazon Textract와 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접 호출하는 방법을 보여주며, 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

시나리오는 동일한 서비스 내에서 또는 다른 AWS 서비스와 결합된 상태에서 여러 함수를 직접적으로 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예제입니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

- 시나리오

작업

`AnalyzeDocument`

다음 코드 예시는 AnalyzeDocument의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   "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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 AnalyzeDocument를 참조하세요.

`DetectDocumentText`

다음 코드 예시는 `DetectDocumentText`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   "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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 DetectDocumentText를 참조하세요.

`GetDocumentAnalysis`

다음 코드 예시는 `GetDocumentAnalysis`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   "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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 GetDocumentAnalysis를 참조하세요.

`StartDocumentAnalysis`

다음 코드 예시는 `StartDocumentAnalysis`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   "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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 StartDocumentAnalysis를 참조하세요.

`StartDocumentTextDetection`

다음 코드 예시는 `StartDocumentTextDetection`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    "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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 StartDocumentTextDetection을 참조하세요.

시나리오

문서 분석 시작

다음 코드 예제에서는 다음과 같은 작업을 수행하는 방법을 보여줍니다.

- 비동기 분석을 시작합니다.

- 문서 분석을 가져옵니다.

SDK for SAP ABAP

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   "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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 다음 주제를 참조하세요.
  - GetDocumentAnalysis

- StartDocumentAnalysis

SDK for SAP ABAP를 사용한 Amazon Transcribe 예제

다음 코드 예제에서는 Amazon Transcribe와 함께 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`CreateVocabulary`

다음 코드 예시는 `CreateVocabulary`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 CreateVocabulary를 참조하세요. AWS

`DeleteTranscriptionJob`

다음 코드 예시는 `DeleteTranscriptionJob`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
  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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DeleteTranscriptionJob을 참조하세요. AWS

`DeleteVocabulary`

다음 코드 예시는 `DeleteVocabulary`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 DeleteVocabulary를 참조하세요. AWS

GetTranscriptionJob

다음 코드 예시는 GetTranscriptionJob의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 GetTranscriptionJob을 참조하세요. AWS

GetVocabulary

다음 코드 예시는 GetVocabulary의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 GetVocabulary를 참조하세요. AWS

`ListTranscriptionJobs`

다음 코드 예시는 `ListTranscriptionJobs`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 ListTranscriptionJobs를 참조하세요. AWS

`ListVocabularies`

다음 코드 예시는 `ListVocabularies`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 ListVocabularies를 참조하세요. AWS

`StartTranscriptionJob`

다음 코드 예시는 `StartTranscriptionJob`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 StartTranscriptionJob을 참조하세요. AWS

`UpdateVocabulary`

다음 코드 예시는 `UpdateVocabulary`의 사용 방법을 보여 줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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.
```

- API 세부 정보는 SDK for SAP ABAP API 참조의 UpdateVocabulary를 참조하세요. AWS

SDK for SAP ABAP를 사용한 Amazon Translate 예

다음 코드 예제에서는 Amazon Translate에서 AWS SDK for SAP ABAP를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접 호출하는 방법을 보여주며, 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

시나리오는 동일한 서비스 내에서 또는 다른 AWS 서비스와 결합된 상태에서 여러 함수를 직접적으로 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예제입니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업
- 시나리오

작업

`DescribeTextTranslationJob`

다음 코드 예시는 `DescribeTextTranslationJob`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   "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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 DescribeTextTranslationJob을 참조하세요.

`ListTextTranslationJobs`

다음 코드 예시는 `ListTextTranslationJobs`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    "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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 ListTextTranslationJobs를 참조하세요.

`StartTextTranslationJob`

다음 코드 예시는 `StartTextTranslationJob`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    "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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 StartTextTranslationJob을 참조하세요.

StopTextTranslationJob

다음 코드 예시는 StopTextTranslationJob의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   "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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 StopTextTranslationJob을 참조하세요.

`TranslateText`

다음 코드 예시는 `TranslateText`의 사용 방법을 보여줍니다.

SDK for SAP ABAP API

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
   "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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 TranslateText를 참조하세요.

시나리오

번역 작업 시작하기

다음 코드 예제에서는 다음과 같은 작업을 수행하는 방법을 보여줍니다.

- 비동기식 배치 번역 작업을 시작합니다.
- 비동기 작업이 완료될 때까지 기다립니다.

- 비동기 작업에 대해 설명합니다.

SDK for SAP ABAP

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
    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.
```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 다음 주제를 참조하세요.

- DescribeTextTranslationJob
- StartTextTranslationJob

의 보안 AWS SDK for SAP ABAP

의 클라우드 보안 AWS 이 최우선 순위입니다. AWS 고객은 보안에 가장 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처의 이점을 누릴 수 있습니다.

보안은 AWS 와 사용자 간의 공동 책임입니다. 공동 책임 모델은 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드 보안 - AWS 는 AWS 서비스 에서 실행되는 인프라를 보호할 책임이 있습니다 AWS 클라우드. AWS 또한는 안전하게 사용할 수 있는 서비스를 제공합니다. 타사 감사자는 AWS 규정 준수 프로그램 일환으로 보안의 효과를 정기적으로 테스트하고 확인합니다. 에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 규정 준수 AWS 서비스 프로그램 제공 범위 내 규정 준수 AWS SDK for SAP ABAP참조하세요.
- 클라우드의 보안 - 사용자의 책임은 AWS 서비스 사용하는에 따라 결정됩니다. 또한 귀하는 귀사의 데이터 민감도, 귀사의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 섹션은 다음 주제를 포함합니다.

주제

- 의 SAP 시스템 인증 AWS
- IAM 보안 모범 사례
- SAP 권한 부여
- 보안 운영
- SSF 암호화에서 보안 액세스 키 인증 사용
- IAM Roles Anywhere로 인증서 사용
- 교차 계정 액세스를 위한 소스 프로파일 사용
- SAP 자격 증명 스토어 사용

의 SAP 시스템 인증 AWS

SAP 시스템이 SAP 사용자를 AWS 대신하여를 호출하려면 먼저 SAP 시스템이에 대해 자체 인증해야 합니다 AWS.는의 SDK 프로파일 설정에서 선택한 다음 세 가지 기본 인증 방법을 AWS SDK for SAP ABAP 지원합니다IMG.

AWS SDK for SAP ABAP - BTP 에디션은 SAP 자격 증명 스토어를 사용하는 the section called “비밀 액세스 키 인증” 메서드로만 인증할 수 있습니다.

교차 계정 액세스 시나리오의 경우 SAP ABAP용 SDK는 소스 프로파일도 지원하므로 기본 인증 방법을 사용하여 계정 간에 여러 IAM 역할 가정을 연결할 수 있습니다. 자세한 내용은 the section called “교차 계정 액세스를 위한 소스 프로필 인증” 단원을 참조하십시오.

주제

- Amazon EC2 인스턴스 메타데이터 인증
- 비밀 액세스 키 인증
- IAM Roles Anywhere를 사용한 인증서 기반 인증
- 교차 계정 액세스를 위한 소스 프로필 인증
- 다음 단계

Amazon EC2 인스턴스 메타데이터 인증

Amazon EC2에서 실행되는 SAP 시스템은 Amazon EC2 인스턴스 메타데이터에서 수명이 짧고 자동으로 교체되는 보안 인증을 획득할 수 있습니다. 자세한 내용은 Amazon EC2 인스턴스 메타데이터의 보안 인증 사용을 참조하세요.

SDK for SAP ABAP를 사용할 때는 이 인증 방법을 적극 권장합니다. 활성화하려면 기본 관리자가 아웃바운드 HTTP 통신을 활성화해야 합니다. 추가 기본 구성이 필요하지 않습니다.

> **Note**
>
> 이 인증 방법은 SAP 시스템이 Amazon EC2에서 실행되는 경우에만 적용됩니다. 온프레미스 또는 다른 클라우드 환경에서 호스팅되는 SAP 시스템은 이 방법을 사용하여 인증할 수 없습니다.

비밀 액세스 키 인증

이 매서드를 사용하면 액세스 키 ID와 비밀 액세스 키를 사용하여 AWS에서 SAP 시스템을 인증합니다. SAP 시스템은 IAM 사용자를 AWS 사용하여에 로그인합니다. 자세한 내용은 IAM 사용자의 액세스 키 관리를 참조하세요.

기본 관리자는 AWS IAM 관리자로부터 액세스 키 ID와 비밀 액세스 키를 받습니다. 액세스 키 ID 및 보안 액세스 키를 저장하도록 SAP 시스템을 구성해야 합니다.

- 보안, 저장 및 전달(SSF)
  - SSF 기능을 사용하여 AWS SDK for SAP ABAP를 인증합니다. 자세한 내용은 디지털 서명 및 암호화를 참조하세요.
  - SSF02 보고서를 사용하여 SSF의 envelope 및 develope 기능을 테스트할 수도 있습니다. 자세한 내용은 SSF 설치 테스트를 참조하세요.
  - SDK for SAP ABAP SSF를 구성하는 단계는 /AWS1/IMG 트랜잭션에 설명되어 있습니다. 기술 사전 조건으로 이동한 다음 온프레미스 시스템의 추가 설정을 선택합니다. 자세한 구성 단계는 SSF 암호화에서 보안 액세스 키 인증 사용을 참조하세요.
- SAP 자격 증명 스토어
  - SAP Credential Store를 사용하여 AWS SDK for SAP ABAP - BTP 에디션을 인증합니다. 자세한 내용은 SAP 자격 증명 스토어란 무엇입니까?를 참조하세요.
  - 구성 단계는 SAP 자격 증명 스토어 사용을 참조하세요.

IAM Roles Anywhere를 사용한 인증서 기반 인증

인증 기관(CA)에서 발급한 X.509 인증서를 AWS Identity and Access Management Roles Anywhere를 통한 인증에 사용할 수 있습니다. STRUST에서 인증서를 구성해야 합니다. IAM Roles Anywhere에 CA를 트러스트 앵커로 등록하고 프로필을 생성하여 IAM Roles Anywhere에 적용될 역할/정책을 지정해야 합니다. 자세한 내용은 AWS Identity and Access Management Roles Anywhere에서 신뢰 앵커 및 프로파일 생성을 참조하세요.

SDK for SAP ABAP와 함께 IAM Roles Anywhere를 사용하는 방법에 대한 자세한 단계는 IAM Roles Anywhere와 함께 인증서 사용 섹션을 참조하십시오.

> **Note**
>
> 인증서 취소는 가져온 인증서 취소 목록을 사용하는 경우에만 지원됩니다. 자세한 내용은 취소 섹션을 참조하십시오.

교차 계정 액세스를 위한 소스 프로필 인증

소스 프로필은 여러 AWS 계정에서 여러 IAM 역할 가정을 연결할 수 있는 고급 기능입니다. 이 방법을 사용하면 한 프로파일이 역할을 수임한 다음 AWS CLI의 source_profile 파라미터와 유사한 다른 역할 등을 수임합니다.

소스 프로파일은 세 가지 기본 인증 방법(인스턴스 메타데이터, 보안 액세스 키 또는 인증서 기반) 중 하나에서 작동합니다. 체인의 첫 번째 프로필은 이러한 기본 방법 중 하나를 사용해야 하며, 체인의 후속 프로필은 이전 프로필의 자격 증명을 사용하여 다음 역할을 수임합니다.

이는 대상 리소스에 도달하기 위해 여러 AWS 계정을 통과해야 하는 교차 계정 액세스 시나리오에 유용합니다. 자세한 구성 단계는 교차 계정 액세스를 위한 소스 프로파일 사용을 참조하세요.

다음 단계

에서 SAP 시스템을 인증한 후 SAP ABAP용 AWS SDK는`sts:assumeRole`를 자동으로 수행하여 SAP 사용자의 비즈니스 기능에 적합한 IAM 역할을 수임합니다.

IAM 보안 모범 사례

IAM 관리자는 다음 세 가지 주요 영역을 담당합니다.

- SAP 시스템이 Amazon EC2 메타데이터 또는 비밀 키 보안 인증으로 자체 인증이 가능한지 확인합니다.
- `sts:assumeRole`에서 SAP 시스템에 자체 승격에 필요한 권한이 있는지 확인합니다.
- 각 논리적 IAM 역할에 대해 비즈니스 기능을 수행하는 데 필요한 권한(예: Amazon S3, DynamoDB 또는 기타 서비스에 필요한 권한)을 가진 SAP 사용자의 IAM 역할을 생성합니다. 이는 SAP 사용자가 수임할 수 있는 역할입니다.

자세한 내용은 SAP Lens: AWS Well-Architected Framework의 보안 장을 참조하세요.

주제

- Amazon EC2 인스턴스 프로파일 모범 사례
- SAP 사용자의 IAM 역할
- 소스 프로필 보안 고려 사항

Amazon EC2 인스턴스 프로파일 모범 사례

SAP 시스템이 실행되는 Amazon EC2 인스턴스에는 인스턴스 프로파일에 기반한 권한 부여 세트가 있습니다. 일반적으로 인스턴스 프로파일에는 `sts:assumeRole` 호출 권한만 있으면 SAP 시스템이 필요에 따라 비즈니스별 IAM 역할을 맡을 수 있습니다. 다른 역할로 승격되면 ABAP 프로그램은 사용자에게 작업 수행에 필요한 최소 권한을 부여하는 역할을 맡을 수 있습니다. 예를 들어 인스턴스 프로파일에는 다음 문장이 포함될 수 있습니다.

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"
      ]
   }
 ]
}
```

이 이전 예제에서는 SAP 시스템이 CFO, AUDITOR 또는 REPORTING 사용자의 IAM 역할을 수임하도록 허용합니다. AWS SDK는 SAP에서 사용자의 PFCG 역할에 따라 사용자에게 올바른 IAM 역할을 선택합니다.

Amazon EC2 인스턴스 프로파일은 다른 기능에도 사용할 수 있습니다.

- AWS SAP HANA용 Backint 에이전트
- 오버레이 IP 주소 라우팅을 통한 AWS 고가용성 SAP

또한 이러한 솔루션에는 백업 또는 장애 조치와 관련된 역할에 대한 `sts:assumeRole` 권한이 필요하거나 인스턴스 프로파일에 직접 할당된 권한이 필요할 수도 있습니다.

SAP 사용자의 IAM 역할

ABAP 프로그램에는 DynamoDB 테이블 읽기, Amazon S3의 PDF 객체에서 Amazon Textract 호출, AWS Lambda 함수 실행 등 사용자의 작업을 수행할 수 있는 권한이 필요합니다. 모든 AWS SDKs에서 동일한 보안 모델이 사용됩니다. 다른 AWS SDK에 사용된 기존 IAM 역할을 사용할 수 있습니다.

SAP 비즈니스 분석가는 IAM 관리자에게 필요한 각 논리적 역할에 대한 IAM 역할의 arn:aws:를 요청합니다. 예를 들어 재무 시나리오에서 비즈니스 분석가는 다음과 같은 논리적 IAM 역할을 정의할 수 있습니다.

- `CFO`
- `AUDITOR`
- `REPORTING`

IAM 관리자는 각 논리적 IAM 역할에 대해 IAM 역할을 정의합니다.

`CFO`

- `arn:aws:iam::0123456789:role/finance-cfo`
- Amazon S3 버킷에 대한 읽기 및 쓰기 권한
- DynamoDB 데이터베이스에 대한 읽기 및 쓰기 권한

`AUDITOR`

- `arn:aws:iam::0123456789:role/finance-auditor`
- Amazon S3 버킷에 대한 읽기 권한
- DynamoDB 데이터베이스에 대한 읽기 권한

`REPORTING`

- `arn:aws:iam::0123456789:role/finance-reporting`
- DynamoDB 데이터베이스에 대한 읽기 권한
- Amazon S3 버킷에 대한 권한 없음

비즈니스 분석가는 IAM 역할을 매핑 테이블에 입력하여 논리적 IAM 역할을 물리적 IAM 역할에 매핑합니다.

SAP 사용자의 IAM 역할은 신뢰할 수 있는 보안 주체에 대한 `sts:assumeRole` 작업을 허용해야 합니다. 신뢰할 수 있는 보안 주체는 SAP 시스템이 AWS에서 인증되는 방식에 따라 달라질 수 있습니다. 자세한 내용은 보안 주체 지정을 참조하세요.

다음은 가장 일반적인 SAP 시나리오의 몇 가지 예입니다.

- 인스턴스 프로필이 할당된 상태로 Amazon EC2에서 실행되는 SAP 시스템. 여기서는 Amazon EC2 인스턴스 프로필이 IAM 역할에 연결됩니다.

  JSON

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

- 인스턴스 프로필 없이 Amazon EC2에서 실행되는 SAP 시스템. 여기서는 Amazon EC2가 SAP 사용자의 역할을 맡습니다.

  JSON

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

- 온프레미스에서 실행되는 SAP 시스템. 온프레미스에서 실행되는 SAP 시스템은 비밀 액세스 키를 사용해서만 인증할 수 있습니다. 자세한 내용은 AWS에서 SAP 시스템 인증을 참조하세요.

여기서 SAP 사용자가 맡는 모든 IAM 역할은 SAP 사용자를 신뢰하는 신뢰 관계를 가져야 합니다.

JSON

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

소스 프로필 보안 고려 사항

소스 프로파일을 사용하는 경우:

IAM 역할 관리

중요: 무단 액세스 및 권한 에스컬레이션을 방지하려면 소스 프로파일 체인의 IAM 역할을 엄격하게 관리해야 합니다.

- 최소 권한 원칙 적용 - 각 역할의 특정 목적에 필요한 최소 권한만 부여합니다.
- 역할 권한 정기 감사 - 분기별 또는 요구 사항이 변경될 때 역할 정책 검토 및 업데이트
- 역할 사용량 모니터링 - AssumeRole API 호출을 추적하고 비정상적인 패턴을 식별하는 데 사용
- 신뢰 관계 제한 - 각 역할을 수임할 수 있는 보안 주체를 액세스 권한이 절대적으로 필요한 보안 주체로만 제한합니다.
- 신뢰 정책의 사용 조건 - 적절한 경우 소스 IP, MFA 요구 사항 또는 시간 기반 제한과 같은 조건 추가
- 문서 역할 목적 - 각 역할의 의도된 사용 사례 및 필요한 권한에 대한 명확한 문서 유지

권한 부여 및 액세스 제어

- 체인의 모든 중간 프로필에 적절한 신뢰 정책이 구성되어 있는지 확인합니다.
- 사용자는 중간 프로파일을 포함하여 체인의 모든 프로파일에 대한 /AWS1/SESS 권한이 있어야 합니다.
- 각 IAM 역할은 체인의 이전 역할을 명시적으로 신뢰해야 합니다.

기술적 보호

- SDK는 5개 프로필의 최대 체인 깊이를 적용하여 과도한 STS API 호출을 방지합니다.
- 순환 참조가 자동으로 감지되고 방지됩니다.
- 기본 프로필 인증 방법을 검증하여 표준 방법(INST, SSF 또는 RLA)을 사용하는지 확인합니다.

소스 프로파일 구성에 대한 자세한 내용은 교차 계정 액세스를 위한 소스 프로파일 사용을 참조하세요.

SAP 권한 부여

SDK를 구성하는 데 필요한 권한 부여는 SDK 에디션에 따라 다릅니다.

주제

- 구성을 위한 권한 부여
- 최종 사용자를 위한 SAP 권한 부여

구성을 위한 권한 부여

자세한 내용은 다음 탭을 참조하세요.

SDK for SAP ABAP

SDK for SAP ABAP를 구성하려면 다음의 권한 부여가 필요합니다.

- S_TCODE
  - TCD = /AWS1/IMG
- S_TABU_DIS
  - ACTVT = 02, 03

- DICBERCLS

  다음 권한 부여 그룹 중 하나를 선택합니다.

  - /AWS1/CFG- AWS SDK for SAP ABAP v1 - 구성
  - /AWS1/MOD- AWS SDK for SAP ABAP v1 - 런타임
  - /AWS1/PFL- AWS SDK for SAP ABAP v1 - SDK 프로파일
  - /AWS1/RES- AWS SDK for SAP ABAP v1 - 논리적 리소스
  - /AWS1/TRC- AWS SDK for SAP ABAP v1 - 추적

SDK for SAP ABAP - BTP edition

다음 단계에 따라 구성에 대한 SDK for SAP ABAP - BTP 에디션 액세스를 허용합니다.

1. 비즈니스 역할 템플릿을 사용하여 새 SAP_BR_BPC_EXPERT 비즈니스 역할을 생성합니다. 이 템플릿은 Cutsom Business Configuration 애플리케이션에 대한 액세스를 제공합니다.
2. 일반 역할 세부 정보에서 액세스 범주로 이동하여 읽기, 쓰기, 값 도움말에 대해 무제한을 선택합니다.
3. Business Catalog 탭으로 이동하여 /AWS1/RTBTP_BCAT 비즈니스 카탈로그를 할당하여 SDK 구성에 대한 액세스를 제공합니다.
4. 비즈니스 사용자 탭으로 이동하여 비즈니스 사용자에게 SDK 구성에 대한 액세스 권한을 부여합니다.

최종 사용자를 위한 SAP 권한 부여

전제 조건: SDK 프로필 정의

SAP 보안 관리자가 역할을 정의하기 전에 비즈니스 분석가는 SDK for SAP ABAP 또는 AWS SDK / AWS1/IMG for SAP ABAP - BTP 에디션에 대한 사용자 지정 비즈니스 구성 애플리케이션 트랜잭션에서 SDK 프로파일을 정의합니다. 일반적으로 SDK 프로필의 이름은 비즈니스 기능에 따라 ZFINANCE, ZBILLING, ZMFG, ZPAYROLL 등으로 지정됩니다. 비즈니스 분석가는 각 SDK 프로필에 대해 CFO, AUDITOR, REPORTING과 같은 짧은 이름을 사용하여 논리적 IAM 역할을 정의합니다. IAM 보안 관리자가 이러한 역할을 실제 IAM 역할에 매핑합니다.

PFCG 또는 비즈니스 역할 정의

> **Note**
>
> PFCG 역할은 SAP BTP, ABAP 환경에서 비즈니스 역할이라고 합니다.

그러면 SAP 보안 관리자가 권한 부여 객체 /AWS1/SESS을(를) 추가하여 SDK 프로필에 대한 액세스 권한을 부여합니다.

인증 객체 /AWS1/SESS

- 필드 /AWS1/PROF = ZFINANCE

또한 사용자는 직무에 따라 각 SDK 프로필의 논리적 IAM 역할에 매핑되어야 합니다. 예를 들어 보고 액세스 권한이 있는 재무 감사자는 AUDITOR라는 논리적 IAM 역할에 대한 권한을 부여받을 수 있습니다.

인증 객체 /AWS1/LROL

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

한편, 읽기/쓰기 권한이 있는 CFO는 CFO의 논리적 역할에 대한 권한 부여가 가능한 PFCG 역할을 맡을 수 있습니다.

인증 객체 /AWS1/LROL

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

일반적으로 사용자는 SDK 프로필당 하나의 논리적 IAM 역할에 대해서만 권한을 부여받아야 합니다. 사용자가 둘 이상의 IAM 역할에 대한 권한이 있는 경우(예: CFO가 CFO 및 AUDITOR 논리적 IAM 역할 모두에 대한 권한이 있는 경우) AWS SDK는 더 높은 우선 순위(낮은 시퀀스 번호) 역할이 적용되도록 하여 연결을 해제합니다.

모든 보안 시나리오와 마찬가지로 사용자에게 업무 기능을 수행할 수 있는 최소한의 권한을 부여해야 합니다. PFCG 역할을 관리하는 간단한 전략은 권한이 부여된 SDK 프로필 및 논리적 역할에 따라 단일 PFCG 역할의 이름을 지정하는 것입니다. 예를 들어 Z_AWS_PFL_ZFINANCE_CFO 역할은 프로필

ZFINANCE 및 논리적 IAM 역할 CFO에 대한 액세스 권한을 부여합니다. 그런 다음 이러한 단일 역할을 직무 기능을 정의하는 복합 역할에 할당할 수 있습니다. 각 회사마다 역할 관리를 위한 고유한 전략이 있으므로 자신에게 적합한 PFCG 전략을 정의하는 것이 좋습니다.

보안 운영

저장 데이터 암호화

AWS 보안 액세스 키는 SDK를 인증하는 데 사용됩니다. SAP의 SSF 또는 자격 증명 스토어 기능을 사용하여 암호화됩니다.

전송 중 데이터 암호화

에 대한 모든 호출 AWS 서비스 은 HTTPS로 암호화됩니다. SAP ICM은 HTTPS 연결을 관리합니다. AWS 인증서는 STRUST에서 신뢰할 수 있어야 합니다.

API 사용

ABAP 사용자가 `sts:assumeRole`을 사용하여 역할을 맡으면 세션 이름에는 `USERID-SID-MANDT`와 같은 제목이 붙습니다.

- `USERID`은(는) `SY-UNAME` 변수의 ABAP 사용자입니다.
- `SID`은(는) `SY-SYSID` 변수의 ABAP 시스템 ID입니다.
- `MANDT`은(는) `SY-MANDT` 변수의 ABAP 클라이언트입니다.

세션 이름은 CloudTrail에 사용자 이름으로 표시됩니다. 이렇게 하면 ABAP 사용자의 API 호출을, 호출을 시작한 시스템과 클라이언트 및 사용자까지 추적할 수 있습니다. 자세한 내용은 AWS CloudTrail란 무엇입니까?를 참조하세요.

SSF 암호화에서 보안 액세스 키 인증 사용

온프레미스 SAP 시스템(또는 다른 클라우드에서 실행되는 시스템)은에서 보안 액세스 키 인증을 AWS 사용하여 인증할 수 있습니다 AWS Identity and Access Management. SAP의 Secure Store and Forward Mechanism(SSF)은 IAM 사용자의 AWS 자격 증명(액세스 키 ID 및 보안 액세스 키)을 암호화하고 안전하게 저장하는 데 사용됩니다. SAP 시스템은 IAM 사용자를 AWS 사용하여에 로그인합니다. 자세한 내용은 IAM 사용자의 액세스 키 관리를 참조하세요.

사전 조건

구성을 시작하기 전에 다음 사전 조건을 충족해야 합니다.

- SAP 사용자의 IAM 역할은 IAM 관리자가 생성해야 합니다. 역할에는 필요한를 호출할 수 있는 권한이 있어야 합니다 AWS 서비스. 자세한 내용은 IAM 보안 모범 사례를 참조하세요.
- `/AWS1/IMG` 트랜잭션을 실행하기 위한 권한을 생성하십시오. 자세한 내용은 구성을 위한 권한 부여 섹션을 참조하십시오.

절차

다음 지침에 따라 SSF 암호화 자격 증명 스토리지를 구성합니다.

단계(Steps)

- 1단계 - 자격 증명 스토리지를 위한 SSF 애플리케이션 정의
- 2단계 - SSF 애플리케이션의 암호화 파라미터 설정
- 3단계 - SSF 애플리케이션용 PSE 생성
- 4단계 -에 SSF 애플리케이션 할당 AWS SDK for SAP ABAP
- 5단계 - SSF 암호화 자격 증명을 사용하도록 SDK 프로파일 구성

1단계 - 자격 증명 스토리지를 위한 SSF 애플리케이션 정의

1. 트랜잭션 코드를 실행`SE16`하여 SSF 애플리케이션을 정의합니다.
2. `SSFAPPLIC` 테이블 이름을 입력하고 새 항목을 선택합니다.
3. 다음 세부 정보를 입력합니다.
   - APPLIC: `ZAWS1` (SSF 애플리케이션의 이름).
   - 설명: `SSF Encryption for the AWS SDK for SAP ABAP` (설명).
   - 나머지 필드에 대해 `Selected(X)` 옵션을 선택합니다.
4. 저장을 선택합니다.

2단계 - SSF 애플리케이션의 암호화 파라미터 설정

1. 트랜잭션 코드를 실행`/n/AWS1/IMG`하여 구현 가이드(IMG)를 시작합니다 AWS SDK for SAP ABAP.

2. 온프레미스 시스템에 대한 IMG 노드 AWS SDK for SAP ABAP 설정 > 기술 사전 조건 > 추가 설정을 확장합니다.
3. SSF 파라미터 설정 IMG 활동을 실행합니다.
4. 새 항목을 선택하고 이전 단계에서 만든 SSF 애플리케이션을 선택합니다. 저장을 선택합니다.
5. 해시 알고리즘을 SHA256(또는 그 이상)으로 수정하고 암호화 알고리즘을 AES256-CBC로 수정합니다. 다른 설정은 기본값으로 유지하고 저장을 선택합니다.

이러한 암호화 설정은 AWS 자격 증명을 안전하게 암호화하는 데 사용됩니다.

3단계 - SSF 애플리케이션용 PSE 생성

1. `/n/AWS1/IMG` 트랜잭션을 실행하고 AWS SDK for SAP ABAP 설정 > 기술 사전 조건 > 온프레미스 시스템의 추가 설정을 선택합니다.
2. IMG 활동 SSF 애플리케이션용 PSE 생성을 실행하면 로 이동합니다`STRUST transaction`. 편집을 선택합니다.
3. the section called “1단계”에서 만든 SSF 애플리케이션을 마우스 오른쪽 단추로 선택하고 생성을 선택합니다. 다른 모든 기본 설정을 유지하고 계속을 선택합니다. DSARSA가 아닌를 알고리즘으로 선택해야 합니다.

4단계 -에 SSF 애플리케이션 할당 AWS SDK for SAP ABAP

1. `/n/AWS1/IMG` 트랜잭션을 실행하고 AWS SDK for SAP ABAP 설정 > 기술 사전 조건 > 온프레미스 시스템의 추가 설정을 선택합니다.
2. IMG 활동 실행에 SSF 애플리케이션을 할당합니다 AWS SDK for SAP ABAP.
3. 새 항목을 선택하고 에서 생성된 SSF 애플리케이션을 입력합니다the section called “1단계”. 저장을 선택합니다.

5단계 - SSF 암호화 자격 증명을 사용하도록 SDK 프로파일 구성

1. `/n/AWS1/IMG` 트랜잭션을 실행하고 AWS SDK for SAP ABAP 설정 > 애플리케이션 구성을 선택합니다.
2. IMG 활동 SDK 프로파일을 실행합니다.
3. 새 항목을 선택합니다. 프로필 이름과 설명을 입력합니다. 저장을 선택합니다.
4. 생성한 항목을 강조 표시하고 인증 및 설정 트리 브랜치를 클릭합니다.

5. 새 항목을 선택하고 다음 세부 정보를 입력합니다.
   - SID: SAP 시스템의 시스템 ID입니다.
   - 클라이언트: SAP 시스템의 클라이언트입니다.
   - 시나리오 ID: 기본 관리자가 생성한 DEFAULT 시나리오를 선택합니다.
   - AWS 리전: 호출하려는 AWS 리전입니다.
   - 인증 방법: 드롭다운Credentials from SSF Storage에서를 선택하고 저장을 선택합니다. 자격 증명 설정을 선택하고 IAM 사용자의 액세스 키 ID 및 보안 액세스 키를 입력합니다.
   - IAM 역할 비활성화: 이를 선택 취소된 상태로 기본값으로 유지합니다.
   - 저장을 선택합니다.
6. IAM 역할 매핑 트리 브랜치를 클릭합니다. 새 항목을 선택합니다. 시퀀스 번호, 논리적 IAM 역할의 이름 및 IAM 관리자가 제공한 AWS IAM 역할 ARN을 입력합니다. 저장을 선택합니다.

자세한 내용은 애플리케이션 구성 섹션을 참조하십시오.

IAM Roles Anywhere로 인증서 사용

SAP 시스템은 AWS Identity and Access Management Roles Anywhere AWS 에서 인증된 기반 인증을 사용하여에서 인증할 수 있습니다. STRUST에서 인증서를 설정하고 /AWS1/IMG에서 SDK 프로필을 구성해야 합니다.

사전 조건

인증서 설정을 시작하기 전에는 다음의 사전 요구 사항이 충족되어야 합니다.

- 인증 기관(CA)에서 발급한 X.509 인증서는 다음 요구 사항을 충족해야 합니다.
  - 서명 인증서는 v3 인증서여야 합니다.
  - 체인은 인증서 5개를 초과할 수 없습니다.
  - 인증서는 RSA 또는 ECDSA 알고리즘을 지원해야 합니다.
- IAM Roles Anywhere에 CA를 트러스트 앵커로 등록하고 프로필을 생성하여 IAM Roles Anywhere의 역할/정책을 지정하십시오. 자세한 내용은 AWS Identity and Access Management Roles Anywhere에서 신뢰 앵커 및 프로파일 생성을 참조하세요.
- SAP 사용자의 IAM 역할은 IAM 관리자가 생성해야 합니다. 역할에는 필요한를 호출할 수 있는 권한이 있어야 합니다 AWS 서비스. 자세한 내용은 IAM 보안 모범 사례를 참조하세요.

- /AWS1/IMG 트랜잭션을 실행하기 위한 권한을 생성하십시오. 자세한 내용은 구성을 위한 권한 부여 섹션을 참조하십시오.

절차

다음 지침에 따라 인증서 기반 인증을 설정하십시오.

단계(Steps)

- 1단계 - SAP의 Secure Store and Forward(SSF)를 사용하여 SSF 애플리케이션을 정의합니다.
- 2단계 - SSF 파라미터 설정
- 3단계 - PSE 및 인증서 요청 생성
- 4단계 - 인증서 응답을 관련 PSE로 가져오기
- 5단계 - IAM Roles Anywhere를 사용하도록 SDK 프로필 구성

1단계 - SAP의 Secure Store and Forward(SSF)를 사용하여 SSF 애플리케이션을 정의합니다.

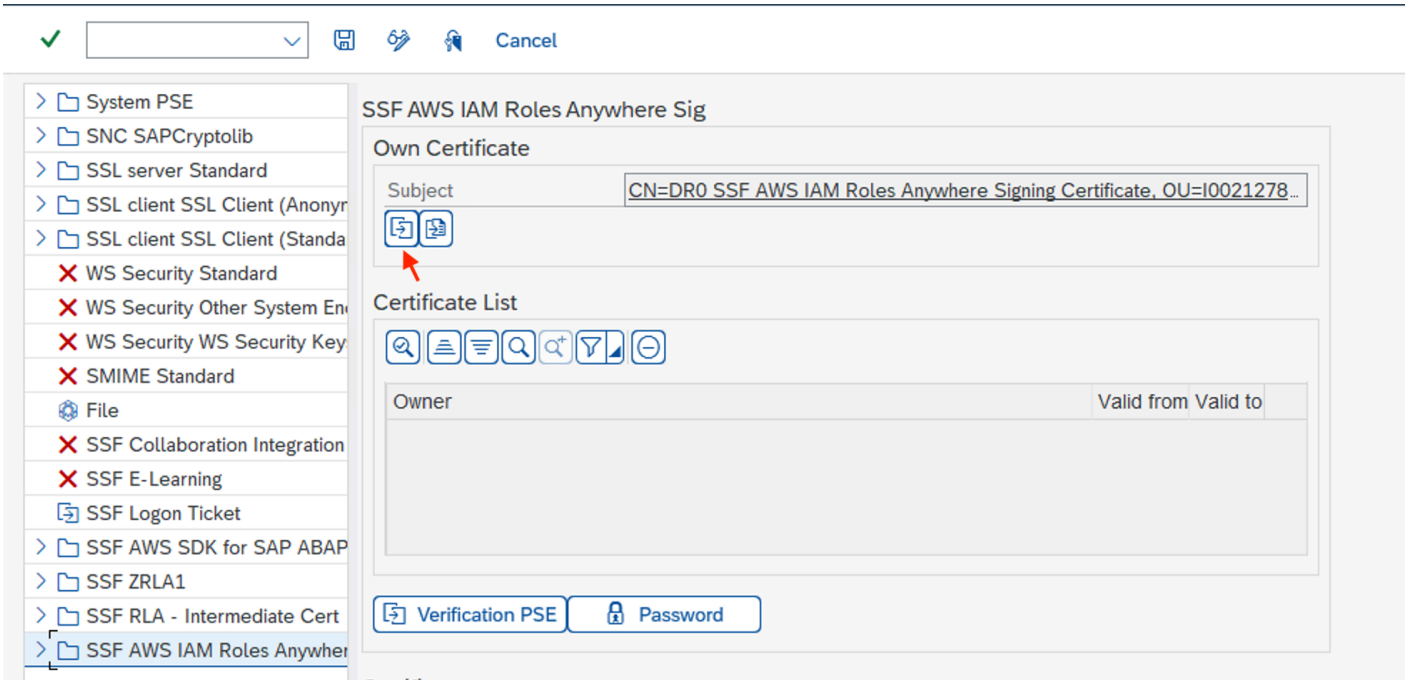
1. 트랜잭션 코드 SE16을 실행하여 SSF 애플리케이션을 정의합니다.
2. SSFAPPLIC 테이블 이름을 입력하고 새 항목을 선택합니다.
3. APPLIC 필드에 SSF 애플리케이션의 이름을 입력하고 DESCRIPT 필드에 설명을 입력한 다음 나머지 필드에서 Selected (X) 옵션을 선택합니다.

2단계 - SSF 파라미터 설정

1. 를 실행/n/AWS1/IMG하여 AWS SDK for SAP ABAP 구현 가이드(IMG)를 시작합니다.
2. AWS SDK for SAP ABAP 설정 > 기술적 사전 조건 > 온프레미스 시스템 추가 설정을 선택합니다.
3. SSF 파라미터 설정 IMG 활동을 실행합니다.
4. 새 항목을 선택하고 이전 단계에서 만든 SSF 애플리케이션을 선택합니다. 저장을 선택합니다.
5. 해시 알고리즘을 SHA256으로 수정하고 암호화 알고리즘을 AES256-CBC로 수정합니다. 다른 설정은 기본값으로 유지하고 저장을 선택합니다.

3단계 - PSE 및 인증서 요청 생성

1. /n/AWS1/IMG 트랜잭션을 실행하고, AWS SDK for SAP ABAP 설정 > 기술적 사전 조건 > 온프레미스 시스템 추가 설정을 선택합니다.
2. Create PSE for SSF Application IMG 활동을 실행합니다.
3. STRUST 트랜잭션의 편집을 선택합니다.
4. the section called “1단계”에서 만든 SSF 애플리케이션을 마우스 오른쪽 단추로 선택하고 생성을 선택합니다. 다른 모든 기본 설정을 유지하고 계속을 선택합니다.
5. 인증서 요청 생성을 선택합니다. 다음 이미지를 참조하십시오. 기본 옵션을 유지하고 계속을 선택합니다. 생성된 인증서 요청을 복사하거나 내보낸 다음 CA에 제공하십시오. CA는 요청을 확인하고 서명된 퍼블릭 키 인증서로 응답합니다.



서명 프로세스는 CA 및 CA에서 사용하는 기술에 따라 달라집니다. 예제는 Private Certificate Authority를 사용하여 AWS 프라이빗 최종 엔터티 인증서 발급을 참조하세요.

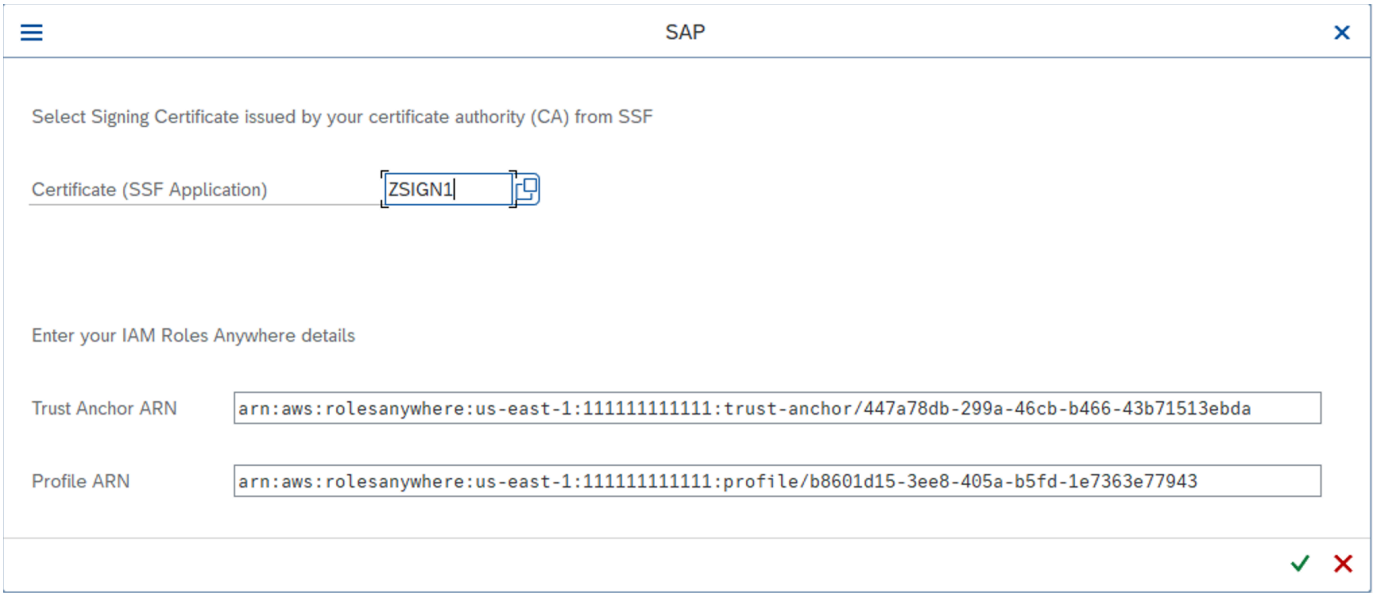
4단계 - 인증서 응답을 관련 PSE로 가져오기

1. /n/AWS1/IMG 트랜잭션을 실행하고, AWS SDK for SAP ABAP 설정 > 기술적 사전 조건 > 온프레미스 시스템 추가 설정을 선택합니다.
2. Create PSE for SSF Application IMG 활동을 실행합니다.
3. STRUST 트랜잭션의 편집을 선택합니다.

4. SSF 애플리케이션을 선택한 다음 제목 아래의 PSE 섹션에 있는 인증서 응답 가져오기를 선택합니다. 인증서 응답을 복사하여 텍스트 상자에 붙여넣거나 파일 시스템에서 파일을 가져오십시오. 계속 > 저장을 선택합니다.
5. 제목을 두 번 선택하면 인증서 세부 정보를 볼 수 있습니다. 정보는 인증서 섹션에 표시됩니다.

5단계 - IAM Roles Anywhere를 사용하도록 SDK 프로필 구성

1. `/n/AWS1/IMG` 트랜잭션을 실행하고 AWS SDK for SAP ABAP 설정 > 애플리케이션 구성을 선택합니다.
2. 새 SDK 프로필을 만들고 이름을 지정합니다.
3. 인증 방법으로 IAM Roles Anywhere를 선택합니다.
   - 왼쪽 창에서 인증 및 설정을 선택합니다.
   - 새 항목을 생성하고 SAP 시스템 및 AWS 리전에 대한 정보를 입력합니다.
   - 인증 방법으로 IAM Roles Anywhere를 선택하고 저장을 선택합니다.
   - 세부 정보 입력을 선택하고 팝업 창에서 the section called “1단계”에서 생성한 SSF 애플리케이션을 선택합니다. the section called “사전 조건”에서 생성한 트러스트 앵커 ARN 및 프로필 ARN을 입력합니다. 다음 이미지를 참조하십시오. 계속을 선택합니다.



4. 왼쪽 창에서 IAM 역할 매핑을 선택합니다. 이름을 입력하고 IAM 관리자가 제공한 IAM 역할의 ARN을 입력합니다.

자세한 내용은 애플리케이션 구성 섹션을 참조하십시오.

교차 계정 액세스를 위한 소스 프로파일 사용

소스 프로파일을 사용하면 SAP 시스템이 IAM 역할 가정을 연결하여 여러 계정의 AWS 리소스에 액세스할 수 있습니다. 한 프로필은 역할을 수임한 다음 AWS CLI의 `source_profile` 파라미터와 유사한 다른 역할 등을 수임합니다. 이는 대상 리소스에 도달하기 위해 여러 계정을 통과해야 하는 교차 AWS 계정 액세스 시나리오에 유용합니다.

예: SAP 시스템은 계정 A(111111111111)에서 실행되며 계정 C(333333333333)의 Amazon S3 버킷에 액세스해야 합니다. 세 가지 프로파일을 구성합니다.

1. `DEV_BASE`는 Amazon EC2 인스턴스 메타데이터에서 기본 자격 증명을 가져오고 계정 A에서 역할 P를 수임합니다.
2. `SHARED_SERVICES`는 자격 `DEV_BASE` 증명을 사용하여 계정 B(222222222222)에서 역할 Q를 수임합니다.
3. `PROD_S3_ACCESS`는 자격 `SHARED_SERVICES` 증명을 사용하여 계정 C에서 역할 R을 수임합니다.

애플리케이션에서 `PROD_S3_ACCESS`를 사용하는 경우 SDK는 체인을 자동으로 실행합니다. 인스턴스 메타데이터에서 자격 증명 가져오기 → 역할 P 수임 → 역할 Q 수임 → 역할 R 수임.

사전 조건

소스 프로파일을 구성하기 전에 다음 사전 조건을 충족해야 합니다.

- 체인의 각 단계에 대한 IAM 역할은 IAM 관리자가 생성해야 합니다. 각 역할에는 다음이 있어야 합니다.
  - 필요한를 호출할 수 있는 권한 AWS 서비스
  - 체인의 이전 역할이 이를 수임할 수 있도록 구성된 신뢰 관계

  자세한 내용은 IAM 보안 모범 사례를 참조하세요.
- `/AWS1/IMG` 트랜잭션을 실행하기 위한 권한을 생성하십시오. 자세한 내용은 구성을 위한 권한 부여 섹션을 참조하십시오.
- 사용자는 중간 프로파일을 포함하여 체인의 모든 프로파일에 대한 `/AWS1/SESS` 권한이 있어야 합니다.

절차

다음 지침에 따라 소스 프로파일을 구성합니다.

단계(Steps)

- 1단계 - 기본 프로파일 구성
- 2단계 - 체인 프로필 구성

1단계 - 기본 프로파일 구성

기본 프로필은 체인의 첫 번째 프로필이며 표준 인증 방법을 사용해야 합니다.

1. `/n/AWS1/IMG` 트랜잭션을 실행하여 AWS SDK for SAP ABAP 구현 가이드(IMG)를 시작합니다.
2. AWS SDK for SAP ABAP 설정 > 애플리케이션 구성 > SDK 프로필을 선택합니다.
3. 새 항목을 선택하여 기본 프로필로 사용할 새 프로필을 생성하고 프로필 이름과 설명을 입력합니다. 저장을 선택합니다.

   > **Note**
   >
   > 표준 인증 방법(INST, SSF 또는 RLA)으로 이미 구성된 기존 프로파일을 사용하는 경우이 섹션의 나머지 단계를 건너뛰고 로 바로 진행할 수 있습니다the section called “단계 2”.

4. 생성한 프로필을 선택한 다음 인증 및 설정 > 새 항목을 선택하고 다음 세부 정보를 입력합니다.
   - SID: SAP 시스템의 시스템 ID
   - 클라이언트: SAP 시스템의 클라이언트
   - 시나리오 ID: 기본 관리자가 생성한 `DEFAULT` 시나리오 선택
   - AWS 리전: 호출하려는 AWS 리전
   - 인증 방법: 다음 중 하나를 선택합니다.
     - Amazon EC2에서 실행되는 SAP 시스템용 메타데이터를 통한 인스턴스 역할
     - 온프레미스 또는 기타 클라우드 시스템을 위한 SSF 스토리지의 자격 증명
     - 인증서 기반 인증을 위한 IAM Roles Anywhere

   저장을 선택합니다.
5. IAM 역할 매핑 > 새 항목을 선택하고 다음을 입력합니다.
   - 시퀀스 번호: 1
   - 논리적 IAM 역할: 설명이 포함된 이름(예: `DEV_BASE_ROLE`)
   - IAM 역할 ARN: 첫 번째 계정에 있는 IAM 역할의 ARN(예: `arn:aws:iam::111111111111:role/DevBaseRole`)

저장을 선택합니다.

2단계 - 체인 프로필 구성

체인의 각 중간 및 최종 프로파일을 구성합니다.

SHARED_SERVICES 프로필(의 체인**DEV_BASE**):

1. /n/AWS1/IMG 트랜잭션을 실행합니다.
2. AWS SDK for SAP ABAP 설정 > 애플리케이션 구성 > SDK 프로필을 선택합니다.
3. 새 항목을 선택합니다. 프로필 이름(예: SHARED_SERVICES)과 설명을 입력합니다. 저장을 선택합니다.
4. 생성한 프로필을 선택한 다음 인증 및 설정 > 새 항목을 선택하고 다음 세부 정보를 입력합니다.
   - SID: SAP 시스템의 시스템 ID
   - 클라이언트: SAP 시스템의 클라이언트
   - 시나리오 ID: 기본 관리자가 생성한 DEFAULT 시나리오 선택
   - AWS 리전: 호출하려는 AWS 리전
   - 인증 방법: 드롭다운에서 소스 프로필을 선택합니다.
   - 소스 프로필 ID: 기본 프로필의 프로필 ID를 입력합니다(예: DEV_BASE).

   저장을 선택합니다.
5. IAM 역할 매핑 > 새 항목을 선택하고 다음을 입력합니다.
   - 시퀀스 번호: 1
   - 논리적 IAM 역할: 설명이 포함된 이름(예: SHARED_ROLE)
   - IAM 역할 ARN: arn:aws:iam::222222222222:role/SharedServicesRole

   저장을 선택합니다.

PROD_S3_ACCESS 프로필(의 체인**SHARED_SERVICES**):

와 동일한 단계를 반복SHARED_SERVICES하지만 다음을 수행합니다.

- PROD_S3_ACCESS 이름으로 사용
- 소스 프로필 ID를 로 설정 SHARED_SERVICES

- IAM 역할 매핑`arn:aws:iam::333333333333:role/ProdS3AccessRole`에서 `PROD_S3_ROLE` 및 사용

IAM 역할 관리, 신뢰 정책 구성 및 권한 부여 요구 사항을 포함한 보안 모범 사례는 IAM 보안 모범 사례를 참조하세요.

SAP 자격 증명 스토어 사용

SAP 자격 증명 스토어는 SAP Business Technology 플랫폼에서 보안 액세스 키 인증을 위한 자격 증명을 안전하게 저장하는 데 사용됩니다 AWS. 서비스를 사용하려면 구독이 있어야 합니다.

다음 지침에서는 SDK 프로필을 이미 구성했다고 가정합니다. 자세한 내용은 구성을 참조하세요 AWS SDK for SAP ABAP.

구성을 시작하기 전에 사전 조건을 충족하는지 확인합니다. 자세한 내용은 SAP 자격 증명 스토어를 참조하세요.

주제

- 구성 단계
- SDK에서 SAP 자격 증명 스토어 사용

구성 단계

단계(Steps)

- 1단계: 인증을 위한 설정 구성
- 2단계: 서비스 키 생성
- 3단계: 서비스 키를 .p12 형식으로 변환
- 4단계: SAP BTP, ABAP 환경에 연결

1단계: 인증을 위한 설정 구성

다음 단계에 따라 인증을 위한 자격 증명 스토어 설정을 구성합니다.

1. SAP Credential Store 인스턴스의 설정 탭으로 이동합니다.
2. 구성 편집을 선택합니다.
   - 상호 TLS를 기본 인증 유형으로 선택합니다.

- 페이로드 암호화 상태에 대해 비활성화됨을 선택합니다. 페이로드는 HTTPS를 사용하여 전송 중에 암호화됩니다. 그러나 현재 페이로드는 이중 암호화할 수 없습니다.

3. 저장을 선택합니다.

2단계: 서비스 키 생성

다음 단계에 따라 자격 증명 스토어에 대한 서비스 키를 생성합니다.

1. SAP Credential Store 애플리케이션의 왼쪽 창에서 서비스 키로 이동합니다.
2. 서비스 키 생성을 선택합니다.
3. 서비스 키의 이름을 입력하고 생성을 선택합니다.

서비스 키는 선택한 인증 유형을 기반으로 생성됩니다. 서비스 키를 다운로드하고 나중에 사용할 수 있도록 보안을 유지합니다.

3단계: 서비스 키를 `.p12` 형식으로 변환

통신 시스템을 위한 아웃바운드 사용자를 생성하려면 `.p12` 형식의 클라이언트 인증서가 필요합니다. 다음 단계에 따라 자격 증명 스토어 서비스 키에 제공된 `.p12` 인증서 세부 정보에서 인증서를 생성합니다.

1. SAP Trust Center Services에서 SAP Cloud Root CA 인증서(SAP에서 요구됨)를 다운로드합니다. https://support.sap.com/en/offerings-programs/support-services/trust-center-services.html
2. 모든 텍스트 파일 형식으로 SAP Cloud Root CA 인증서를 엽니다. 파일 끝에서 Enter 키를 누르고 서비스 키의 인증서 필드에서 인증서를 복사하여 붙여 넣습니다. 새 줄 문자를 실제 새 줄(Enter)\n로 바꾸고 전체 인증서를 `.cer` 파일 형식으로 저장합니다.
3. 서비스 키의 키 필드에서 키를 복사합니다. 이 프라이빗 키는 민감한 데이터로 취급되어야 합니다. 텍스트 파일에 붙여넣고 새 줄 문자를 실제 새 줄(Enter)\n로 바꿉니다. 프라이빗 키를 텍스트 파일에 저장합니다.
4. 이전 단계에서 생성된 인증서와 프라이빗 키를 사용하여 다음 명령을 실행하여 `.p12` 인증서를 생성합니다.

```
openssl pkcs12 -export -out <.p12_filename> -inkey <private_key.key> -in
 <certificate.cer>
```

명령은 내보내기 암호를 확인해야 했습니다. 나중에 사용할 수 있도록 암호를 보관합니다.

프라이빗 키에 저장된 .key 텍스트 파일을 삭제합니다.

4단계: SAP BTP, ABAP 환경에 연결

SAP 자격 증명 스토어에 연결하도록 SAP BTP, ABAP 환경을 구성합니다.

주제

- 통신 시스템
- 통신 배열

통신 시스템

다음 단계에 따라 SAP BTP, ABAP 환경에서 SAP 자격 증명 스토어로 통신할 수 있는 통신 시스템을 생성합니다.

1. ABAP 환경 시스템의 Fiori 시작패드를 엽니다.
2. Communication Systems 타일을 선택하여 애플리케이션을 엽니다.
3. New(새 파일)를 선택합니다.
4. 통신 시스템의 이름과 ID를 입력하고 생성을 선택합니다. 예를 들어 시스템 이름을 로 지정할 수 있습니다ZSAP_CREDSTORE.
5. 기타 필수 정보를 입력합니다.
   - 호스트 이름: 서비스 키 URL에서 호스트 이름을 복사합니다. 예를 들어 URL이 https://credstore.mesh.cf.us10.hana.ondemand.com/api/v1/credentials인 경우 호스트 이름은 입니다credstore.mesh.cf.us10.hana.ondemand.com.
   - 아웃바운드 통신 사용자: 새 사용자를 + 추가하려면 선택합니다.
     a. SSL 클라이언트 인증서를 인증 메커니즘으로 선택합니다.
     b. 새 인증서 업로드를 선택합니다.
        - 이전 단계에서 생성된 .p12 인증서를 찾습니다.
        - 설명을 입력합니다.
        - .p12 인증서를 생성하는 데 사용된 내보내기 암호를 입력합니다.
        - 업로드를 선택합니다.
     c. 생성을 선택하여 아웃바운드 사용자를 생성합니다.
6. 저장을 선택합니다.
7. 이전 단계에서 다운로드한 서비스 키를 삭제합니다.

통신 배열

다음 단계를 사용하여 아웃바운드 통신을 위한 통신 시나리오를 제공하는 통신 배열을 생성합니다.

1. ABAP 환경 시스템의 Fiori 시작패드를 엽니다.
2. 커뮤니케이션 계약 타일을 선택하여 애플리케이션을 엽니다.
3. New(새 파일)를 선택합니다.
4. 통신 시나리오를 `/AWS1/CRED_COMM_SCENARIO`선택하고 통신 배열의 이름을 입력합니다. 예를 들어 `Z_AWS_SDK_TO_SAP_CREDSTORE`입니다.
5. 생성을 선택합니다.
6. 통신 시스템 필드에서 이전 단계에서 생성한 통신 시스템을 찾습니다. 다른 정보는 시스템 선택 후 자동으로 채워집니다.
7. 저장을 선택합니다.
8. 연결 확인을 선택하여 연결을 테스트합니다.

이 설정이 완료되면 ABAP 환경은 통신 배열을 사용하여 아웃바운드 서비스(HTTP)를 통해 SAP 자격 증명 스토어 서비스를 사용할 수 있습니다.

SDK에서 SAP 자격 증명 스토어 사용

단계(Steps)

- 1단계: 네임스페이스 및 자격 증명(들) 생성
- 2단계: 사용자 지정 비즈니스 구성 애플리케이션 구성

1단계: 네임스페이스 및 자격 증명(들) 생성

SAP 도움말을 사용하여 SAP 자격 증명 스토어에서 네임스페이스 및 자격 증명 생성 - 자격 증명 생성, 편집 및 삭제.

다음 세부 정보를 입력하여 키 유형의 자격 증명을 생성합니다.

- 네임스페이스 - 네임스페이스의 이름을 입력하고 관련 자격 증명을 함께 그룹화합니다.
- 이름 - 키의 이름을 입력합니다. 다음과 `aws-0123456789012-username`같은 경우를 사용하는 것이 좋습니다.
  - `0123456789012`는 자격 증명이 액세스 권한을 부여하는 AWS 계정 ID입니다.
  - `username`는 자격 증명이 속한 IAM 사용자 이름입니다.

- 값 - base-64로 인코딩된 보안 액세스 키를 입력합니다. 다음 명령을 사용하여 보안 액세스 키를 base-64로 인코딩합니다.

```
xargs echo -n | base64 # just press enter, do not enter arguments on the command line
MySecretAccessKey
Ctrl-D
```

  명령은 표준 입력에서 보안 액세스 키를 읽고 후행 줄 바꿈 없이 base64에 전달합니다. base-64로 인코딩된 보안 액세스 키를 화면에 출력합니다. SAP 자격 증명 스토어에 값을 복사한 후 터미널을 지우거나 닫습니다.
- 사용자 이름 - 액세스 키 ID를 입력합니다.

- 생성을 선택합니다.

자격 증명이 하나 있는 새 네임스페이스가 생성되고이 네임스페이스 내에서 자격 증명을 추가, 삭제 또는 수정할 수 있습니다.

최소 권한 원칙에 따라 네임스페이스에 저장된 자격 증명에 대한 액세스를 관리합니다.

2단계: 사용자 지정 비즈니스 구성 애플리케이션 구성

다음 단계를 사용하여 SDK의 인증에 사용할 자격 증명을 정의하도록 사용자 지정 비즈니스 구성 애플리케이션을 구성합니다.

1. ABAP 환경 시스템의 Fiori 시작패드를 엽니다.
2. 사용자 지정 비즈니스 구성 타일을 탐색하여 애플리케이션을 엽니다.
3. SDK 프로파일 비즈니스 구성을 엽니다.
4. SAP 자격 증명 스토어에 대한 인증 설정을 구성해야 하는 SDK 프로파일을 선택합니다.
5. 선택한 프로필의 인증 및 설정 탭에서 편집을 선택하고 다음 세부 정보를 입력합니다.
   - 인증 방법 - SAP 자격 증명 스토어에서 자격 증명을 선택합니다.
   - 네임스페이스 - SAP 자격 증명 스토어에서 생성된 네임스페이스를 입력합니다. 자세한 내용은 the section called “1단계: 네임스페이스 및 자격 증명(들) 생성” 단원을 참조하십시오.
   - 키 이름 - 생성된 키 자격 증명의 이름을 입력합니다. 자세한 내용은 the section called “1단계: 네임스페이스 및 자격 증명(들) 생성” 단원을 참조하십시오.
   - 통신 배열 - 생성된 통신 배열의 이름을 입력합니다. 자세한 내용은 the section called “통신 배열” 단원을 참조하십시오.

6. 적용을 선택하여 AWS SDK 프로필 화면으로 이동합니다.
7. 전송 선택을 선택하여 값 도움말을 사용하여 전송을 선택합니다.
8. 저장을 선택합니다.

문제 해결 AWS SDK for SAP ABAP

이 섹션에서는 발생 가능한 오류 시나리오에 대한 문제 해결 단계를 제공합니다.

주제

- 가져오기 실패
- 지정되지 않은 위치 제약
- SSL 오류
- 프로파일 구성
- IAM 권한 부여
- 필수 작업 수행을 위한 권한 부여
- 활성 시나리오
- 코드의 특수 문자
- 연결

가져오기 실패

문제 - 'CL_SYSTEM_UUID' 클래스에 'IF_SYSTEM_UUID_RFC4122_STATIC' 인터페이스가 포함되어 있지 않음

원인 — 시스템에서 SAP Note 0002619546이 누락되었습니다.

해결 방법 — SAP Note 0002619546이 시스템에 적용되었는지 확인합니다.

지정되지 않은 위치 제약

문제 - 이 요청이 전송된 `region` 특정 엔드포인트에서 지정되지 않은 위치 제약이 호환되지 않음

원인 - Amazon S3 버킷에 `io_createbucketconfiguration` 파라미터의 AWS 리전이 없습니다.

해결 방법 — `us-east-1`을(를) 제외한 모든 지역에서 버킷을 생성할 때는 `createbucket()`에서 `io_createbucketconfiguration` 파라미터를 사용하여 Amazon S3 버킷의 리전을 지정합니다. `us-east-1`에 대한 제약 조건을 지정할 필요가 없습니다.

다음 예는 올바르게 구성된 io_createbucketconfiguration 파라미터를 보여줍니다.

```
createbucket(
    iv_bucket = 'amzn-s3-demo-bucket'
    io_createbucketconfiguration = NEW /aws1/cl_s3_createbucketconf( 'us-west-1' )
).
```

SSL 오류

문제 — SSL 서버 인증서 호스트 이름 불일치 또는 docs.aws.amazon.com:443과의 SSL 핸드셰이크 실패: SSSLERR_NO_SSL_RESPONSE

icm/HTTPS/client_sni_enabled원인 — DEFAULT 프로파일에서 파라미터가 TRUE(으)로 설정되어 있지 않습니다.

해결 방법 - 다음 단계를 사용하여 해당 문제 또는 기타 SSL 관련 문제를 해결합니다.

1. SAPGUI를 열고 명령 모음으로 이동합니다.
2. RZ10 트랜잭션을 실행합니다.
3. 프로파일로 이동하여 DEFAULT 프로파일을 선택합니다. 버전이 자동으로 채워집니다.
4. 프로파일 편집 섹션에서 확장 유지 관리를 선택한 다음 변경을 선택합니다.
5. icm/HTTPS/client_sni_enabled 파라미터를 검색합니다.
   - 파라미터가 있는 경우 파라미터 값을 편집하고 TRUE로 설정합니다.
   - 파라미터가 없는 경우 다음 단계에 따라 파라미터를 만듭니다.
     1. 파라미터를 선택합니다.

        > **Note**
        > 편집할 파라미터(연필 아이콘)가 아니라 생성할 파라미터를 선택했는지 확인합니다.

     2. 파라미터 이름 icm/HTTPS/client_sni_enabled을(를) 필드에 입력합니다.
     3. 파라미터 값 TRUE을(를) 필드에 입력합니다.
     4. 저장을 선택합니다.
6. 이러한 변경 내용을 DEFAULT 프로파일에 저장하고 종료합니다.

프로파일 구성

문제 — <sid>:<client>에 대한 DEFAULT 시나리오를 사용하여 <profile_name> 프로파일에서 구성을 찾을 수 없음

원인 — <profile_name>이 잘못되었거나 구성되지 않았습니다.

해결 방법 - 다음 단계를 사용하여 프로파일을 구성합니다.

1. SAPGUI를 열고 `/n/AWS1/IMG` 트랜잭션을 실행합니다.
2. 애플리케이션 구성 > SDK 프로파일로 이동합니다.
   - 프로파일이 구성된 경우 프로파일 이름이 올바른지 확인합니다.
   - 프로파일이 구성되지 않은 경우 단계에 따라 프로파일을 구성합니다.
3. 새 항목을 선택합니다.
   a. 프로파일에 대해 이름과 설명을 입력합니다.
   b. 저장을 선택합니다.
4. 이전 단계에서 만든 항목을 선택한 다음 인증 및 설정을 선택합니다.
5. 새 항목을 선택하고 다음 세부 정보를 입력한 다음 저장을 선택합니다.
   - SID
   - 클라이언트
   - 시나리오 ID
   - AWS 리전
   - 인증 방법
     - 에서 실행되는 SAP 시스템의 메타데이터를 통해 인스턴스 역할을 선택합니다 AWS.
     - 온프레미스 또는 기타 클라우드를 실행하는 SAP 시스템의 경우 SSF 스토리지에서 보안 인증을 선택합니다.
6. IAM 역할 매핑 > 새 항목을 선택하고 다음 세부 정보를 입력한 다음 저장을 선택합니다.
   - 시퀀스 번호
   - 논리적 IAM 역할
   - IAM 역할 ARN

IAM 권한 부여

문제 — <iam_role_arn> 역할을 맡을 수 없거나 사용자 <user_arn>은(는) 리소스 <iam_role_arn>에서 sts:AssumeRole을 수행할 권한이 없음

원인 — 이 오류의 가능한 원인은 다음과 같습니다.

- 잘못된 IAM 역할 ARN이 지정됨
- IAM 사용자에게 IAM 역할에 액세스하는 권한이 없음
- 수임된 IAM 역할과 수임하는 IAM 역할 또는 IAM 사용자 간의 신뢰 관계가 부족

해결 방법 - 다음 단계를 사용하여 IAM 역할 ARN이 올바른지 확인합니다.

1. SAPGUI를 열고 `/n/AWS1/IMG` 트랜잭션을 실행합니다.
2. 애플리케이션 구성 > SDK 프로파일로 이동하여 IAM 역할로 구성된 프로파일을 선택합니다.
3. IAM 역할 매핑을 선택하고 IAM 역할 ARN을 확인하거나 수정합니다.
    - IAM 역할 ARN이 올바르면 IAM 역할이 제대로 구성되었는지 확인합니다. 자세한 내용은 IAM 역할 문제 해결을 참조하세요.

필수 작업 수행을 위한 권한 부여

문제 — 사용자 <user_arn>은(는) 리소스 <resource_arn>에서 <action>을 수행할 권한이 없음

원인 — 사용자에게 작업을 수행할 권한이 없습니다.

해결 방법 — `user_arn`은(는) 지정된 `action`을(를) 수행하는 데 필요한 `resource_arn` 권한을 설정해야 합니다. 자세한 내용은 IAM 리소스에 액세스하는 데 필요한 권한을 참조하세요.

활성 시나리오

문제 — 활성 시나리오가 구성되어 있지 않음

원인 — 활성 시나리오 설정이 누락되었습니다.

해결 방법 - 활성 시나리오를 구성하려면 런타임 설정을 참조하세요.

코드의 특수 문자

경고 — 0x00A0 문자는 ABAP 단어의 일부가 될 수 없습니다

> **Note**
>
> 이 경고에 앞서 다양한 오류 메시지가 나타날 수 있습니다.

원인 — 다양한 소스에서 코드를 복사하여 붙여넣으면 코드에 특수 문자가 삽입될 수 있습니다.

해결 방법 — ABAP 소스 코드 편집기에 코드를 붙여넣으면 다음과 같은 팝업이 표시됩니다.

줄 바꿈하지 않는 공백 문자가 감지되었습니다. 공백으로 변환할까요?

이 질문에 답하려면 예를 선택합니다. 또한 코드 상자의 복사 버튼을 사용하는 대신 복사할 코드를 선택하는 것이 좋습니다.

연결

문제 — SCLNT_HTTP(411) : tla.region.amazonaws.com:443에 대한 직접 연결 실패: NIECONN_REFUSED(-10)

원인 — SAP 시스템이 인터넷에 연결되어 있지 않아 tla.region.amazonaws.com의 포트 443에 TCP/IP 연결을 설정할 수 없습니다.

해결 방법 - SAP 시스템은 직접 또는 프록시 서버를 통해 HTTPS 포트 443의 AWS 엔드포인트에 대한 연결을 설정할 수 있어야 합니다. 다음 옵션 중 하나를 사용하여 인터넷 연결을 설정/확인할 수 있습니다.

- NAT 또는 인터넷 게이트웨이를 통해 인터넷에 직접 아웃바운드 연결
- 프록시 서버를 통한 연결

  자세한 내용은 프록시 서버를 통한 연결을 참조하세요.

추가 주제

이 섹션은 다음 주제를 포함합니다.

주제

- AWS SDK for SAP ABAP 릴리스
- SAP 라이선싱

AWS SDK for SAP ABAP 릴리스

AWS SDK for SAP ABAP는 전송으로 제공되고 AWS SDK for SAP ABAP - BTP 에디션은 추가 기능으로 제공됩니다. 전송 및 추가 기능을 가져오는 메커니즘은 다르지만 기술 기능은 동일합니다. 자세한 내용은 설정을 참조하세요.

주제

- 릴리스 전략
- 모범 사례
- SDK for SAP ABAP 패치
- 추가 모듈 설치
- SDK for SAP ABAP 제거

릴리스 전략

의 버전 1 AWS SDK for SAP ABAP 은 자주 업데이트됩니다. 새 패치는 릴리스 및 업데이트에 따라 매주 또는 매일 릴리스됩니다 AWS 서비스. 에 대한 패치에는 SDK의 패치 수준을 업데이트하는 버그 수정 및 기타 변경 사항이 포함될 AWS 서비스 수 있습니다. 자세한 내용은 AWS SDKs.

모범 사례

모든 SAP 시스템(개발, QA 및 프로덕션)에 대해 동일한 수준의 SDK for SAP ABAP를 유지하는 것이 좋습니다.

SDK를 패치할 때는 샌드박스에서 최신 버전을 가져옵니다. 그런 다음 일반적인 변경 제어 절차에 따라 개발, QA 및 프로덕션 시스템으로 가져올 수 있습니다.

SDK for SAP ABAP 패치

각 SDK for SAP ABAP 릴리스는 모든 버그 수정, 기능 및 업데이트를 포함하여 누적 전송 세트로 제공됩니다. 패치와 설치 전송 사이에는 차이가 없습니다. SDK for SAP ABAP의 패치를 진행하려면 최신 전송을 가져와야 합니다.

`core` 런타임 및 API 모듈의 종속성으로 인해 개별 API 모듈은 별도로 패치할 수 없습니다. 패치 적용 시 특정 `core` 모듈을 더 이상 사용하지 않더라도 모듈과 설치된 다른 모든 API 모듈을 업데이트해야 합니다.

다음 시나리오에서는 패치 적용 프로세스를 설명합니다.

- 시나리오 1: 정기적인 SDK 버전 업데이트

  최신 SDK 버전에 패치하는 경우:

  - 모든 전송(코어 및 설치된 API 모듈)을 동시에 가져오기
  - 특정 가져오기 시퀀스가 필요하지 않음
  - 예를 들어, SDK를 설치할 때 `core`, `ec2`, `lmd` 전송을 가져온 경우, 패치를 적용할 때 `core`, `ec2`, `lmd`에 대한 최신 전송을 가져와야 합니다.
- 시나리오 2: 새 API 모듈 추가

  현재 설치된 모듈과 다른 SDK 릴리스에서 새 API 모듈(예: Amazon Textract `tex`)을 가져올 때:

  - 새 모듈의 SDK 릴리스 버전과 일치하도록 모든 기존 모듈을 패치해야 합니다.
  - 동시에 가져오기(특정 시퀀스 없음):
    - 모든 기존 모듈(`core` 및 설치된 API 모듈)의 최신 전송
    - 새 API 모듈에 대한 전송 `tex`
- 시나리오 3: 기존 API 모듈 업데이트

  기존 API 모듈(예: Amazon Translate `x18`)을 최신 SDK 버전으로 업데이트하는 경우 상호 종속성으로 인해 개별 모듈 업데이트가 지원되지 않습니다. 에 설명된 프로세스에 따라 `core` 모듈과 설치된 모든 API 모듈을 함께 업데이트하여 전체 SDK를 패치해야 합니다Regular SDK Version Update (p. 561).

추가 모듈 설치

기존 core 및 모듈과 동일한 패치 수준에서 새 모듈의 전송을 가져와 SAP 시스템에 추가 API 모듈을 설치합니다. 최신 버전의 모듈을 가져오려면 the section called “SDK for SAP ABAP 패치”의 지침을 따르세요. 이렇게 하면 패치 수준이 모든 SDK 모듈에서 호환됩니다.

SDK for SAP ABAP 제거

SDK for SAP ABAP를 제거하려면 https://sdk-for-sapabap.aws.amazon.com/awsSdkSapabapV1/release/uninstall-abapsdk-LATEST.zip에서 삭제 전송 키트를 다운로드해야 합니다.

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

https://sdk-for-sapabap.aws.amazon.com/awsSdkSapabapV1/release/uninstall-abapsdk-LATEST.sig에서 서명 파일을 다운로드할 수 있습니다. 파일을 검증하려면 SDK for SAP ABAP 확인을 참조하세요.

SAP 시스템에 설치된 각 SDK 모듈에 대해 이전 ZIP 파일에서 해당 삭제 전송을 가져와야 합니다. 전체 SDK를 제거하지 않고도 단일 모듈을 제거할 수 있습니다. 제거하려는 모듈의 삭제 전송만 가져와서 제거할 수 있습니다. 모든 모듈과 함께 전체 SDK를 제거하는 경우 코어 삭제 전송을 마지막으로 가져와야 합니다.

개발, QA 또는 프로덕션 시스템에서 제거를 시도하기 전에 샌드박스에서 제거를 테스트하는 것이 좋습니다.

고려 사항

SDK를 제거하기 전에 다음 고려 사항을 참조하세요.

- 의 SDK 구성 설정이 손실됩니다. 설치 시 다시 IMG을(를) 구성해야 합니다.
- SDK를 사용하는 Z 프로그램이 있는 경우 SDK 제거 후 구문 오류가 발생합니다.
- SDK 권한 부여 참조가 포함된 PFCG 또는 비즈니스 역할은 SDK 제거 후 잘못된 권한을 갖게 됩니다. SDK를 제거하기 전에 PFCG 역할에서 SDK 권한 부여 참조를 제거하세요.

> **Note**
>
> AWS SDK for SAP ABAP - 개발자 미리 보기 중에는 BTP 에디션을 제거할 수 없습니다.

SAP 라이선싱

SAP 소프트웨어 사용에는 SAP 약관이 적용됩니다. 귀하는 소프트웨어 배포 및 간접 라이선스 조건을 포함한 SAP 라이선스 약관을 준수할 책임이 있습니다. 제공된 모든 정보는 법률 자문이 아니므로 라이선스 규정 준수 목적으로 의존해서는 안 됩니다. SAP 소프트웨어의 라이선스 또는 소유권에 대한 질문이 있는 경우 해당 법률팀, SAP 및/또는 SAP 리셀러에게 문의하십시오.

질문: SDK for SAP ABAP 사용이 SAP 라이선스에 영향을 미치나요?

답변: 자체 ABAP 코드 AWS 서비스 로 AWS SDK for SAP ABAP 사용할 수 있습니다. SAP 시스템과 AWS 서비스간의 통합 시나리오에 사용됩니다. SAP 시스템의 데이터가 타사(비 SAP) 시스템으로 전송되거나 해당 시스템에서 생성되는 모든 시나리오는 간접 라이선스에 영향을 미칠 수 있습니다. SAP는 간접 액세스를 정의하는 데 사용자 기반 계산, 결과 기반 계산 등 다양한 접근 방식을 사용합니다. 간접 액세스를 정의하는 방법은 SAP와의 계약에 따라 달라집니다. SAP 계약에서 제공하는 지침을 숙지해야 하며, 이에 대해서는 SAP 또는 해당 리셀러와 추가로 논의할 수 있습니다.

2018년에 SAP는 SAP 설치 기반 고객을 위한 간접 액세스 안내서와 디지털 시대를 위한 SAP ERP 가격 책정 - 간접/디지털 액세스 해결이라는 두 가지 문서를 릴리스했습니다. 해당 문서는 SAP 웹 사이트에서 찾을 수 있으며 해당 문서는 간접 라이선스 접근 방식의 예입니다. 하지만 이러한 문서에는 SAP와의 특정 계약은 반영되어 있지 않습니다.

AWS SDK for SAP ABAP 개발자 안내서의 문서 기록

다음 표에서는에 대한 설명서 릴리스를 설명합니다 AWS SDK for SAP ABAP.

변경 사항	설명	날짜
새로운 내용	교차 계정 액세스를 위한 소스 프로파일 사용을 추가했습니다.	2026년 3월 4일
새로운 내용	SSF 암호화에서 보안 액세스 키 인증 사용을 추가했습니다.	2025년 1월 12일
새로운 내용	SDK for SAP ABAP - BTP 에디션의 개발자 미리 보기.	2024년 5월 31일
새로운 내용	IAM Roles Anywhere로 인증서 사용이 추가되었습니다.	2023년 12월 1일
새로운 내용	SDK로 제품 구축이 추가되었습니다.	2023년 12월 1일
새로운 내용	동작 재시도가 추가되었습니다.	2023년 12월 1일
새로운 내용	SAP 라이선스가 추가되었습니다.	2023년 9월 22일
공개 릴리스	AWS SDK for SAP ABAP 개발자 안내서의 최초 재출시.	2023년 6월 30일
새로운 내용	AWS SDK for SAP ABAP 기능이 추가되었습니다.	2023년 5월 30일
새로운 내용	AWS SDK for SAP ABAP문제 해결이 추가되었습니다.	2023년 2월 17일
개발자 미리 보기	개발자 안내서의 AWS SDK for SAP ABAP 개발자 미리 보기.	2022년 11월 17일

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.