



개발자 가이드

AWS SAP ABAP용 SDK



AWS SAP ABAP용 SDK: 개발자 가이드

Copyright © 2024 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

이게 뭐야 AWS SDK for SAP ABAP?	1
의 특징 AWS SDK for SAP ABAP	1
유지 관리	1
API 참조	2
요금	2
리소스	2
시작하기	3
1단계: 계정 준비 AWS	3
SAP 사용자의 IAM 역할	3
인증	4
2단계: SDK 설치	6
3단계: SDK 구성	6
4단계: 기능 설정	7
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
SAP ABAP용 SDK - BTP 에디션	20
설치 중 AWS SDK for SAP ABAP	21
SDK 다운로드	21
파일 확인	21
AWS SDK 트랜스포트	22
SDK 설치 - BTP 에디션	25
SAP ABAP용 SDK 설치 - BTP 에디션	26
SAP ABAP용 SDK 패치 적용 - BTP 에디션	26
구성	27
글로벌 설정	28
기술 설정	28
시나리오 구성	28
애플리케이션 구성	29
SDK 프로필	29
논리적 리소스 해석기	30

예	31
런타임 설정	31
로그 및 추적	32
옵트인: 향상된 원격 측정	32
활성 시나리오	32
새로 고침, 추적, 원격 측정	32
SAP 시스템 새로 고침	33
추적	33
원격 측정	34
고급 연결 시나리오	35
프록시 서버를 통한 연결	35
패킷 검사 방화벽을 통한 연결	36
게이트웨이 엔드포인트	36
사용자 지정 인터페이스 엔드포인트	36
다중 리전 엔드포인트에 액세스	37
SDK 사용	39
데이터 표시	39
데이터 유형	39
AWS 데이터 유형	41
예제 프로그램	42
사전 조건	42
코드	43
코드 섹션	44
개념	46
API 클래스	46
추가 객체	47
구조 클래스	47
배열	49
맵	50
상위 수준 함수	50
특성	1
프로그래밍 방식 구성	51
Waiters	52
페이지네이터	53
재시도 동작	54
제품 구축	54

제품 ID 설정	55
제한 사항	55
코드 예시	57
작업 및 시나리오	57
Amazon Bedrock 런타임	58
CloudWatch	63
DynamoDB	70
Amazon EC2	85
Kinesis	101
Lambda	111
Amazon S3	125
SageMaker	134
Amazon SNS	153
Amazon SQS	161
Amazon Textract	169
Amazon Translate	179
보안	189
시스템 인증	189
메타데이터 인증	190
비밀 액세스 키 인증	190
IAM Roles Anywhere를 사용한 인증서 기반 인증	191
다음 단계	191
IAM 보안 모범 사례	191
Amazon EC2 인스턴스 프로파일 모범 사례	192
SAP 사용자의 IAM 역할	193
SAP 권한 부여	195
구성을 위한 권한 부여	195
최종 사용자를 위한 SAP 권한 부여	196
보안 운영	198
저장 데이터 암호화	198
전송 중 데이터 암호화	198
API 사용	2
인증서 사용	198
필수 조건	199
절차	199
크리덴셜 스토어	20

구성 단계	202
SDK와 함께 SAP 자격 증명 저장소 사용	204
문제 해결	208
가져오기 실패	208
지정되지 않은 위치 제약	208
SSL 오류	209
프로파일 구성	210
IAM 권한 부여	211
작업을 위한 권한 부여	211
활성 시나리오	32
특수 문자	212
연결	212
추가 주제	213
출시	213
릴리스 전략	213
모범 사례	191
SDK for SAP ABAP 패치	214
추가 모듈 설치	214
SDK for SAP ABAP 제거	214
SAP 라이선싱	215
사용 설명서 기록	216
.....	CCXvii

이게 뭐야 AWS SDK for SAP ABAP?

AWS SDK for SAP ABAP ABAP AWS 언어로 제공되는 서비스에 대한 인터페이스를 제공합니다. SDK를 사용하면 Amazon Simple Storage Service (Amazon S3), AWS 서비스등과 같은 ABAP BADI, 보고서, 트랜잭션, OData 서비스 및 기타 ABAP 아티팩트를 구현할 수 있습니다. Amazon DynamoDB Amazon Translate 또한 SAP NetWeaver 7.4부터 시작하여 SAP 비즈니스 기술 플랫폼 환경에서 ABAP 기반 시스템용으로 개발할 수 있습니다. 자세한 내용은 [SAP ABAP용 AWS SDK 설치](#) - BTP 에디션을 참조하십시오.

주제

- [의 특징 AWS SDK for SAP ABAP](#)
- [SDK 메이저 버전에 대한 유지 관리 및 지원](#)
- [API 참조](#)
- [요금](#)
- [추가적인 리소스](#)

의 특징 AWS SDK for SAP ABAP

AWS SDK for SAP ABAP SAP 개발자에게 친숙하고 자연스럽게 느껴지도록 설계되었습니다. 예를 들어, 모두 true 및 false 문자열을 AWS 서비스 사용하여 부울 데이터를 XML 및 JSON 구조로 표현하지만 SDK for SAP ABAP는 이러한 데이터를 ABAP 네이티브 및 단일 문자 값으로 변환합니다. 'X' ' ' 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 안내서](#)에서 다음 내용을 참조하세요.

- [AWS SDK 및 도구 유지 관리 정책](#)
- [AWS SDK, 도구 및 버전 지원 매트릭스](#)

API 참조

전체 API 목록을 AWS SDK for SAP ABAP 보려면 [AWS SDK for SAP ABAP - API 참조](#) 안내서를 참조하십시오.

AWS SDK for SAP ABAP TLA의 전체 모듈 목록을 보려면 [AWS SDK for SAP ABAP - 모듈](#) 목록을 참조하십시오.

요금

AWS SDK for SAP ABAP 추가 비용 없이 사용할 수 있습니다. SDK로 사용한 AWS 리소스와 서비스에 대한 비용만 지불하면 됩니다.

추가적인 리소스

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

- [SAP \(설명서 참조 AWS\)](#)
- [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 환경에서 다음 단계를 수행하는 것이 좋습니다.

단계

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

1단계: 계정 준비 AWS

SDK for SAP ABAP를 시작하려면 활성 AWS 계정 (이)가 있어야 합니다. SAP 시스템이 온프레미스, SAP 비즈니스 기술 플랫폼 (BTP) 또는 다른 클라우드 제공업체를 통해 호스팅되는 AWS 계정 경우에도 필요합니다.

SAP 시스템이 AWS 클라우드에서 실행되는 경우 클라우드에 있는 AWS 서비스에 전화를 걸게 됩니다. AWS 계정

주제

- [SAP 사용자의 IAM 역할](#)
- [인증](#)

SAP 사용자의 IAM 역할

- AWS Identity and Access Management 사용자 설명서에 제공된 지침에 따라 IAM 역할을 생성합니다. 자세한 내용은 [AWS 서비스에 권한을 위임하기 위한 역할 생성](#)을 참조하십시오. 나중에 사용할 수 있도록 IAM 역할의 Amazon 리소스 이름(ARN)을 적어 둡니다.

- 사용 사례로 Amazon EC2를 선택합니다.
- 역할 이름으로 SapDemoTranslate을(를) 사용합니다.
- TranslateReadOnly 프로파일을 역할에 추가합니다.
- SAP 시스템이 역할을 수임할 수 있으려면 역할에 다음 엔티티가 있어야 합니다.
"111122223333"를 AWS 계정 번호로 바꿉니다.

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

이 예제는 의 모든 보안 주체가 역할을 맡을 AWS 계정 "111122223333" 수 있음을 보여줍니다. 이는 적합한 광범위한 권한입니다 proof-of-concept. 다음 예와 같이 프로덕션에 더 좁은 보안 주체를 사용할 수 있습니다.

- 특정 사용자 — SAP 시스템에서 다음 중 하나를 사용하는 경우
 - 온프레미스 SAP 시스템의 SSF로 암호화된 자격 증명
 - SAP BTP, ABAP 환경의 SAP 자격 증명 저장소 서비스에서 제공하는 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 사례](#)를 참조하세요.

인증

인증은 SAP 시스템이 호스팅되는 위치에 따라 달라집니다.

위치

- [AWS 온 클라우드](#)
- [온프레미스, SAP BTP 또는 기타 클라우드](#)

AWS 온 클라우드

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

```
{
  "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 수입할 수 있습니다.

온프레미스, SAP BTP 또는 기타 클라우드

SAP 시스템이 온프레미스, SAP BTP 또는 기타 클라우드에 있는 경우 다음 단계에 따라 비밀 액세스 키를 사용하여 인증을 위한 연결을 설정하십시오.

1. IAM 사용자를 생성합니다. 자세한 정보는 [IAM 사용자\(콘솔\) 생성](#)을 참조하세요.
2. IAM 사용자의 이름으로 SapDemoSID을(를) 사용합니다. SID은(는) SAP 시스템의 시스템 ID입니다.
3. 이 사용자에게 SapDemoTranslate 역할을 할당합니다.

및 을 access_key 보관하세요. 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 저장소를 사용한 비밀 액세스 키 인증](#)

- [어디서나 IAM 역할이 있는 인증서 사용](#)

2단계: SDK 설치

설치 지침은 다음 탭을 참조하십시오.

SDK for SAP ABAP

SAP 시스템에서 SAP ABAP 전송을 위한 SDK를 가져옵니다. 모든 클라이언트로 전송을 가져올 수 있습니다. 자세한 내용은 [SAP ABAP용 SDK 설치](#)를 참조하십시오.

SDK for SAP ABAP - BTP edition

제품 배포 애플리케이션을 사용하여 SAP ABAP용 SDK - BTP 에디션을 설치합니다. 자세한 내용은 [SAP ABAP용 SDK 설치](#) - BTP 에디션을 참조하십시오.

3단계: SDK 구성

구성 지침은 다음 탭을 참조하십시오.

SDK for SAP ABAP

/AWS1/IMG트랜잭션을 실행하여 SAP ABAP용 SDK 구현 가이드를 엽니다. 이 트랜잭션을 실행하려면 SAP 시스템의 명령 모음에 /n/AWS1/IMG 을(를) 입력한 다음 입력을 선택합니다.

다음 구성을 완료하세요.

- 기술 사전 조건으로 이동합니다.
 - 권장 프로필 파라미터와 SSL 인증서 설정을 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 DEFAULT을(를) 선택합니다.
- 시나리오 변경 커밋을 선택합니다.
- 프롬프트를 수락합니다.

SDK for SAP ABAP - BTP edition

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

다음 구성을 완료하십시오.

- 글로벌 설정 → 시나리오 구성으로 이동합니다.
 - 글로벌 설정의 권장 사항에 따라 설정을 변경합니다.
- 글로벌 설정 → 기술 설정으로 이동합니다.
 - 글로벌 설정의 권장 사항에 따라 설정을 변경합니다.

온프레미스 시스템의 사전 조건

SAP 시스템이 온프레미스 또는 다른 클라우드에서 실행되는 경우 보안 인증을 SAP 데이터베이스에 저장해야 합니다. 자격 증명은 SAP SSF를 사용하여 암호화되며 SAP와 같이 구성된 암호화 라이브러리가 필요합니다. CommonCryptoLib

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.

4단계: 기능 설정

설정 지침은 다음 탭을 참조하십시오.

SDK for SAP ABAP

트랜잭션을 실행 /AWS1/IMG (명령 표시줄에 입력하고 Enter 선택) 하여 AWS SDK 구현 가이드를 엽니다. /n/AWS1/IMG

- 애플리케이션 구성 → SDK 프로파일로 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 지역: 전화를 걸고 싶은 AWS 지역을 입력합니다. SAP 시스템이 실행 중인 경우 실행 중인 AWS 지역을 입력하십시오. AWS
 - 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 저장소를 사용하여 인증을 설정합니다. 자세한 내용은 [SAP 자격 증명 저장소 사용을 참조하십시오](#).

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

- 애플리케이션 구성 → SDK 프로필로 이동합니다.
 - 편집을 선택하여 새 프로필을 생성합니다.
 - 프로파일: DEMO.
 - 설명: Demo profile.
- 생성된 항목 옆의 오른쪽 화살표 키를 선택하고 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 지역: 전화를 걸고 싶은 AWS 지역을 입력합니다. SAP 시스템이 실행 중인 경우 실행 중인 AWS 지역을 입력하십시오. AWS
- 인증 방법: SAP 자격 증명 저장소에서 자격 증명을 선택합니다.
- SAP 자격 증명 저장소에 저장된 자격 증명의 네임스페이스 및 키 이름을 입력합니다.
- SAP ABAP용 SDK - BTP 에디션과 SAP 자격 증명 저장소 간의 통신을 설정하기 위해 만든 통신 계약의 이름을 입력합니다.
- IAM 역할 비활성화를 공란으로 유지합니다.
- 생성된 항목 옆의 화살표 키를 마우스 오른쪽 버튼으로 클릭하고 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 참조하십시오.

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_TESTUSER역할이 할당된 모든 사용자는 SDK 프로필에 DEMO 구성된 설정으로 AWS SDK 기능을 사용할 수 있습니다. 또한 승인된 사용자는 해당 프로필의 TESTUSER 논리적 IAM 역할에 매핑된 IAM 역할을 수입하게 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 — 선택합니다/AWS1/RT_BTP_ENDUSER.
 - 새 비즈니스 역할 ID — ID를 입력합니다.
 - 새 비즈니스 역할 설명 - 설명을 입력합니다.
- 확인을 선택하면 비즈니스 역할 페이지가 표시됩니다.
- 일반 역할 세부 정보 탭에서 액세스 범주로 이동하여 쓰기, 읽기, 값 도움말 필드를 제한됨으로 설정합니다.
- 제한 유지 관리를 선택하고 왼쪽 탐색 창에서 할당된 제한 유형을 확장합니다. 제한 및 값 섹션에서 다음 필드를 업데이트하십시오.
 - SDK 세션 선택에서 SDK 프로필 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 프로필 옆의 연필 아이콘을 선택하고 범위 탭으로 이동합니다. **DEMO**를 입력하고 추가를 선택합니다.

- 비즈니스 역할 템플릿으로 돌아가서 비즈니스 사용자 탭을 엽니다. 추가를 선택하여 SDK 기능을 테스트할 SAP 비즈니스 사용자에게 새로 만든 비즈니스 역할을 할당합니다. 저장을 선택합니다.

생성된 비즈니스 역할에 배정된 모든 비즈니스 사용자는 SDK 프로필에 DEMO 구성된 설정으로 AWS SDK 기능을 사용할 수 있습니다. 또한 승인된 사용자는 해당 프로필의 TESTUSER 논리적 IAM 역할에 매핑된 IAM 역할을 수임하게 됩니다.

6단계: 코드 작성

자세한 내용은 다음 탭을 참조하십시오.

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

```

SDK for SAP ABAP - BTP edition

1. ABAP 클래스를 만들어야 하는 패키지를 마우스 오른쪽 버튼으로 클릭한 다음 새로 만들기 > ABAP 클래스를 선택합니다.
2. 클래스 **ZCL_DEMO_XL8_SIMPLE** 이름을 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_xl8unsuppdedlanguage00 INTO DATA(lo_lang).
  out->write( |ERROR - Cannot translate from { lo_lang->sourcelanguagecode }
to { lo_lang->targetlanguagecode }| ).
  CATCH cx_root INTO DATA(lo_root).
  out->write( |ERROR - { lo_root->get_text( ) }| ).
ENDTRY.
ENDMETHOD.
ENDCLASS.

```

7단계: 애플리케이션 실행

자세한 내용은 다음 탭을 참조하십시오.

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: 계정 준비 AWS”](#)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 애플리케이션 (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: 계정 준비 AWS”](#)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 중 AWS SDK for SAP ABAP](#)
- [SAP ABAP용 AWS SDK 설치 - BTP 에디션](#)

SAP 사전 조건

다음과 같은 SDK 설치 전제 조건은 SAP 시스템을 호스팅할 때 적용됩니다. AWS

주제

- [SAP ABAP용 SDK의 사전 요구 사항 AWS](#)
- [SAP ABAP용 AWS SDK 사전 요구 사항 - BTP 에디션](#)

SAP ABAP용 SDK의 사전 요구 사항 AWS

다음은 SAP ABAP용 SDK의 사전 요구 사항입니다. AWS

주제

- [기본 릴리스](#)
- [커널 릴리스](#)
- [파라미터](#)
- [참고](#)
- [아웃바운드 연결](#)
- [HTTPS 연결](#)
- [Amazon EC2 인스턴스 메타데이터 액세스](#)

기본 릴리스

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

SAP_BASIS 740에 대해 지원되는 최소 SP 수준은 SP 0008입니다. 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.

Installed Software Component Versions

Installed Product Versions



Component	Release	SP-Level	Support Package	Short Description of Component
SAP_BASIS	740	0026	SAPKB74026	SAP Basis Component
SAP_ABA	740	0026	SAPKA74026	Cross-Application Component
SAP_GWFND	740	0027	SAPK-74027INSAPGWFND	SAP Gateway Foundation
SAP_UI	754	0008	SAPK-75408INSAPUI	User Interface Technology
DI_BASIS	740	0008	SAPK-74008INDIBASIS	Basic Diagnostics

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
```

참고

다음 SAP Note를 시스템에 적용합니다.

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

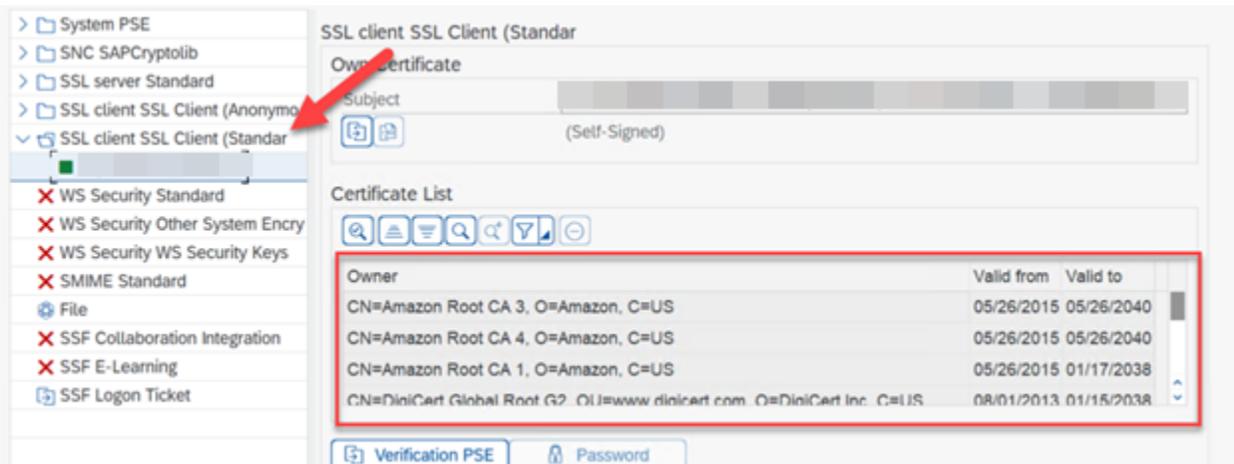
아웃바운드 연결

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 통해 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 사용하면 SAP Secure AWS Store에 비밀 키를 저장하지 않고도 SAP 시스템을 안전하게 인증할 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.

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

SAP ABAP용 AWS SDK 사전 요구 사항 - BTP 에디션

SAP ABAP용 AWS SDK - BTP 에디션의 유일한 전제 조건은 다음과 같습니다.

주제

- [SAP 랜드스케이프 포털 — BTP 에디션](#)
- [SAP 크리덴셜 스토어 — BTP 에디션](#)

SAP 랜드스케이프 포털 — BTP 에디션

이 전제 조건은 SAP ABAP용 AWS SDK - BTP 에디션에만 적용됩니다.

SAP 랜드스케이프 포털은 SAP BTP 환경에서 애드온을 설치할 수 있는 유일한 지원 메커니즘입니다. 이 서비스를 사용하려면 구독해야 합니다. 자세한 내용은 [랜드스케이프 포털](#)을 참조하십시오.

SAP 크리덴셜 스토어 — BTP 에디션

이 사전 요구 사항은 SAP ABAP용 AWS SDK - BTP 에디션에만 적용됩니다.

개발자 미리 보기에서는 비밀 액세스 키 인증이 SAP ABAP용 AWS SDK 인증 (BTP 에디션) 에 지원되는 유일한 메커니즘입니다. SDK는 자격 증명 저장소에서 자격 증명을 읽고 보안 액세스 키를 안전하게 저장합니다.

다음 사전 요구 사항을 충족해야 합니다.

- 크리덴셜 스토어 구독.
- 자격 증명 저장소가 BTP 하위 계정에 대한 자격으로 할당되었습니다. 자세한 내용은 [초기](#) 설정을 참조하십시오.
- 자격 증명 저장소용 표준 요금제를 사용하는 서비스 인스턴스. 자세한 내용은 [서비스 인스턴스 만들기](#)를 참조하십시오.

자세한 내용은 [SAP 자격 증명 저장소 사용을 참조하십시오](#).

SAP 자격 증명 저장소 서비스는 ABAP BTP 시스템 외부의 SAP BTP에서 실행됩니다. 자세한 내용은 [SAP 자격 증명 저장소를 참조하십시오](#).

설치 중 AWS SDK for SAP ABAP

주제

- [SDK for SAP ABAP 다운로드](#)
- [SDK for SAP ABAP 파일 확인 - 선택 사항](#)
- [AWS SDK 트랜스포트](#)

SDK for SAP ABAP 다운로드

sdk-for-sapabaphttps://aws.amazon.com/V1/release/abapsdk-LATEST.zip awsSdkSapabaphttps://amazonaws.cn/

```
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-----
MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICGgKCAgEAmS3oN3wKBh4HJ0Ga0tye
15RR5909nuw0Jx0vEDCT709wUrXS3mjgEw6b6hvr2dLdoFr+eH4ewT5bV16U3gDv
051sTdEJJpfLEWJJZZNK3v9fGWKyXgYe+ifmsPmf4lhNd2auzpvIy2Ur1SYijCRB
```

```
BWZFW+Ux00kILz+8vCFSXMZ6Z0qtLI1ZFbGrn6A5adbwwzf0qkg9BUEZK0wB6TAi
ZTnkMdBZGCBM9K2MRKKMxtrixUn+TFcAYyh5pM9tUAb2q4XE5m7092UnZG7ur/QY1
1FSZwAhQmk8hUPgUaq00QRC6z3TRzIGK0A/DI0cUPJMzFR4LCxEJkgh4rkRaU9V2
07DthUpj8b7QcQai0pnMpBf3zWLgjbNmX0hB0Eprg8/nVRHspf3zuisJ21MPkz0
cHOR31MNsMLzm+d/gVklT31R/JwAcFCkXTwvR8/VOWNGZZXdVUbefrfI/k7fP60B
bzUrI1N4poq16rc4Tk5Derg+wQ7r0WjXkXop2kiCMjbYo0o10kS/At64PLj pz8dH
Zg25o79U9EJ1n+1pqZ297Ks+Hoct0v2GPbeeh0s7+N0fRTy0r81EZIURLPKLVQUw
otVRzNDgLOA7eA667NrmegZfHCmqEwK9tXakZUHAcMzRPyhALc/HtmovxdStN9h1
JC4ex0GqstAv1fX5QaTbMSECAwEAAQ==
-----END PUBLIC KEY-----
```

3. 다음 명령을 사용하여 다운로드한 SDK ZIP 파일을 확인합니다. 명령에는 많은 Linux 배포판의 일부인 openssl이(가) 필요합니다.

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

4. 이전 명령의 출력이 Verified OK인지 확인합니다.
5. 출력이 Verification Failure와(과) 같으면 이전 단계를 반복합니다. 계속해서 실패한 출력을 수신하는 경우 SDK를 설치하지 말고 AWS Support에 문의하세요.

AWS SDK 트랜스포트

주제

- [내용](#)
- [가져오기](#)
- [네임스페이스](#)

내용

SDK for SAP ABAP 설치에 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 core가 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 목록](#)을 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 단계로 넘어갈 준비가 되면 비즈니스 기능을 갖춘 자체 Z 코드가 포함된 별도의 트랜스포트와 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 클라이언트에 구애받지 않습니다. 핵심 전송은 필수이며 SDK 런타임 코드, 옹 API AWS Security Token Service, Amazon 심플 스토리지 서비스용 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 S3용 API를 사용하려는 경우 core 전송 (코어 런타임 Amazon Translate, Amazon S3 및 AWS STS 모듈 포함) 및 x18 전송 (모듈 포함 Amazon Translate) 전송을 가져오십시오.

SDK for SAP ABAP API의 전체 목록을 보려면 [SDK for SAP ABAP - API 참조 안내서](#)를 참조하세요.

다음은 SDK 전송을 가져올 AWS 때의 주요 지침입니다.

- 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 트랜스퍼트를 가져오는 데 몇 분이 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_UID_RFC4122_STATIC 인터페이스 누락으로 인해 오류가 발생하는 경우 SAP Note 2619546이 시스템에 적용되었는지 확인합니다. 자세한 내용은 [참고](#)를 참조하세요.
 - 오류의 원인을 알 수 없는 경우 문의하세요. AWS Support
- 빨간색 번개(RC=12)는 전송 파일이 /usr/sap/trans에 제대로 로드되지 않았거나 필요한 권한이 없음을 나타냅니다.

키 포인터

다음은 AWS SDK 트랜스퍼트를 가져올 때의 주요 지침입니다.

- 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 트랜스포트를 가져오는 데 몇 분이 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 [참고](#)를 참조하세요.
 - 오류의 원인을 알 수 없는 경우 문의하세요. AWS Support
- 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 auth객체는 Auth 객체 클래스에 있습니다. 인증 객체 클래스는 4자로 제한되며 네임스페이스를 지원하지 않습니다. SDK for SAP ABAP는 인증 객체 클래스는 YAW1을(를) 사용합니다. 트랜잭션에 SU21 이미 인증 개체 클래스가 YAW1 있는 경우 설치 AWS Support 전에 문의하세요.

SAP ABAP용 AWS SDK 설치 - BTP 에디션

BTP 에디션은 개발자 프리뷰 버전이며, 프리뷰에 참여하여 설치할 수 있습니다. SDK를 설치하려면 [AWS SDK for SAP ABAP - BTP 에디션 개발자 프리뷰에서 참여 양식](#)을 작성하세요.

SAP ABAP용 SDK - BTP 에디션을 설치하기 전에 필수 사전 요구 사항을 충족하는지 확인하십시오. [자세한 내용은 SAP 랜드스케이프 포털 및 SAP 자격 증명 저장소를 참조하십시오.](#)

주제

- [SAP ABAP용 SDK 설치 - BTP 에디션](#)
- [SAP ABAP용 SDK 패치 적용 - BTP 에디션](#)

SAP ABAP용 SDK 설치 - BTP 에디션

1. SAP 랜드스케이프 포털 인스턴스로 이동하고 제품 배포를 선택하여 Fiori 애플리케이션을 시작합니다.
2. 제품 > 파트너 제품에서 을 선택합니다 **/AWS1/SDK_OMNI**.

개발자 미리보기에서 **/AWS1/SDK_OMNI** 수락한 후 표시되지 AWS Support 없으면 문의하세요.

3. 대상 버전에서 시스템에 설치하려는 SAP ABAP - BTP 에디션용 SDK 버전을 선택합니다.
4. SDK를 설치하려는 모든 SID의 확인란을 선택합니다.

설치에는 30~45분이 소요될 수 있으며, 여기에는 시스템 다운타임도 포함됩니다. 자세한 내용은 제품 [배포를](#) 참조하십시오.

SAP ABAP용 SDK 패치 적용 - BTP 에디션

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

구성 AWS SDK for SAP ABAP

사용하기 AWS SDK for SAP ABAP전에 SDK 작업에 필요한 기술 및 기능 설정으로 SDK를 구성해야 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 구성을 사용하지 않는 것이 좋습니다.

구성 AWS SDK for SAP ABAP

구성 트랜잭션을 실행하려면 SAPGUI 명령 모음에 /n/AWS1/IMG을(를) 입력합니다. AWS SDK를 구성할 권한이 없는 경우 [SAP 인증을](#) 참조하십시오.

SAP ABAP용 AWS SDK 구성 - BTP 에디션

다음 단계를 사용하여 SAP ABAP - BTP 에디션용 SDK를 구성하십시오.

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

사용자 지정 비즈니스 구성 애플리케이션에서 SDK 설정 유형에 따라 구성을 그룹화할 수 있습니다. 다음 단계를 사용하여 구성을 그룹화할 수 있습니다.

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 구성을 계층 구조로 사용할 수 있습니다. [뷰를 저장하려면 뷰 \(변형 관리\) - 컴포넌트를](#) 참조하십시오.

이 섹션은 다음 주제를 포함합니다.

주제

- [글로벌 설정](#)
- [애플리케이션 구성](#)
- [런타임 설정](#)
- [새로 고침, 추적 및 원격 측정 주제 AWS SDK for SAP ABAP](#)

- [고급 연결 시나리오](#)

글로벌 설정

SAP ABAP용 AWS SDK용 /n/AWS1/IMG IMG 트랜잭션과 SAP ABAP용 AWS SDK용 사용자 지정 비즈니스 구성 응용 프로그램 (BTP 에디션) 을 사용하여 글로벌 설정을 구성합니다. 이 항목에서는 IMG와 사용자 지정 비즈니스 구성을 같은 의미로 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 프로파일로 구성되어 있습니다. ABAP SDK 프로파일은 특정 애플리케이션 시나리오에 필요한 설정을 정의합니다. 예를 들어, 트랜잭션 ZVA01ZVA02, 및 ZVA03 가 Amazon S3와 같은 인보이스 관련 트랜잭션이 개선되고 Amazon SageMaker 실행되는 AWS 서비스 경우 SDK 프로파일을 ZINVOICE 호출할 수 있습니다. AWS Lambda 이 프로파일은 인보이스 관련 기능에 대한 기술 설정, SAP 인증 및 IAM 역할 매핑을 그룹화할 수 있습니다.

SAP ABAP용 AWS SDK의 경우 /n/AWS1/IMG 트랜잭션을 사용하고 SAP ABAP용 AWS SDK용 사용자 지정 비즈니스 구성 응용 프로그램 - BTP 에디션을 사용하여 글로벌 설정을 구성합니다. 이 항목에서는 IMG와 사용자 지정 비즈니스 구성을 같은 의미로 사용합니다.

주제

- [SDK 프로파일](#)
- [논리적 리소스 해석기](#)
- [예](#)

SDK 프로파일

ABAP SDK 프로파일은 각 SID 및 클라이언트에 대해 다음을 정의합니다.

Note

SAP BTP, ABAP 환경에서는 클라이언트가 항상 100입니다.

- 모든 API 호출의 기본 AWS 지역입니다. 예를 들어 SAP 시스템이 us-east-1 해당 지역에서 실행되는 경우 다른 AWS 리소스도 같은 지역에 있을 수 있으며 이 지역이 기본 지역이어야 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 저장소에서 자격 증명을 선택해야 합니다. 자세한 내용은 인증을 위한 [SAP 자격 증명 저장소 사용](#)을 참조하십시오.
- 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.

Note

SAP BTP, ABAP 환경에서는 PFCG 역할을 비즈니스 역할이라고 합니다.

ABAP 프로그램이 연결하려는 경우 필요한 설정을 가져오는 ABAP SDK 프로필을 지정합니다. AWS 서비스사용자가 SDK 프로필에 액세스할 권한이 있는지 확인하기 위해 AUTHORIZATION-CHECK이 (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 리소스의 개념을 정의합니다 (예: 청구서를 보관하는 Amazon S3 버킷). AWS 예를 들어, 이 논리적 리소스의 이름은 ZINVOICES_OUTBOUND일 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.

예

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

SDK는 ZINVOICE_OUTBOUND과(와) 같은 논리적 리소스를 물리적 리소스로 해결하면 SY-SYSID 및 SY-MANDT을(를) 확인하여 실행 중인 SID와 클라이언트를 묻고 올바른 물리적 리소스를 자동으로 선택합니다.

프로덕션에서 리소스의 매핑을 변경해야 하는 경우 개발 시스템의 IMG에서 매핑을 변경하여 전달해야 합니다. 따라서 다른 전송과 마찬가지로 SAP 시스템에 AWS 리소스를 재할당할 때도 변경 관리 대상이 됩니다.

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 SAP ABAP용 SDK - BTP 에디션에는 적용되지 않습니다.

옵트인: 향상된 원격 측정

모든 SDK는 지원 목적으로 텔레메트리 정보를 에 전송합니다. AWS 향상된 원격 측정을 옵트인할 수 있습니다. 이는 특정 API 호출의 출처를 확인하기 AWS Support 위해 문의할 때 특히 유용합니다. 자세한 내용은 [추적](#) 및 [원격 측정](#)을 참조하세요.

이러한 설정은 SAP ABAP용 SDK - 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.

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 고침 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/IMG에서 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 오류 발생 시 제공할 정보를 수집하기 AWS Support 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 Support에서 요청하는 경우에만 이러한 추적 기능을 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 SAP ABAP용 SDK - BTP 에디션에는 적용되지 않습니다.

원격 측정

SDK는 텔레메트리 정보를 로 전송합니다. AWS Support 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 AWS Support(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 AWS Support 더 나은 결과를 얻을 수 있습니다. AWS Support 특정 API 호출이 이루어진 이유를 감지하고 SAP 시스템에서 관련 트랜잭션을 추가로 찾을 수 있습니다.

텔레메트리는 SAP ABAP용 SDK (BTP 에디션) 용 SDK 및 API 버전으로 제한됩니다.

고급 연결 시나리오

AWS SDK for SAP ABAP은(는) AWS 엔드포인트에 HTTPS 호출을 수행하여 AWS 서비스(를) 소비합니다. 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](#)
- [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 인증](#)을 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 서비스를(를) 사용하는 SAP 시스템에 자체 인증서를 발급해야 합니다. STRUST에 방화벽의 CA 인증서를 설치해야 합니다. 자세한 내용은 [HTTPS 연결](#)을 참조하세요.

게이트웨이 엔드포인트

일부 AWS 서비스는(는) 인터넷 없이도 VPC에 고성능 액세스가 가능한 게이트웨이 엔드포인트를 제공합니다. 이러한 엔드포인트는 SDK for SAP ABAP에 투명하며 구성이 필요하지 않습니다.

자세한 내용은 [게이트웨이 엔드포인트](#)를 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 구성](#)을 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 엔드포인트에서 서비스 도메인 이름을 확인할 수 없는 이유는 무엇입니까?](#)를 참조하세요.

다음 예에서는 AWS STS 및 Amazon Translate 에 대한 인터페이스 엔드포인트를 생성합니다. 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/IMG의 논리적 리소스가 먼저 물리적 엔드포인트 이름으로 확인됩니다. 그런 다음 AWS 세션 클래스(AWS STS을(를) 사용하여 IAM 역할 수입)의 팩토리 메서드에 제공되고 API 클래스를 변환합니다.

```
" 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 있듯이 go_xl8에 대한 모든 메서드 호출은 엔드포인트 `https://vpce-0123456789abcdefg-12345567.translate.us-west-2.vpce.amazonaws.com`로 이동합니다.

다중 리전 엔드포인트에 액세스

AWS 엔드포인트는 SDK 프로필에 정의된 기본값 AWS 리전에 따라 자동으로 결정됩니다. 프로그래밍 방식으로 지역을 지정하여 기본 리전을 재정의할 수도 있습니다. 이는 팩토리 CREATE() 메서드에

서 재정의하거나 나중에 SDK의 구성 객체로 재정의할 수 있습니다. 자세한 내용은 [프로그래밍 방식 구성](#)을 참조하세요.

다음 예시에서는 팩토리 CREATE() 메서드를 사용하여 리전을 설정하고 us-east-1 및 us-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( ).
  ENDLIST.
ENDLOOP.
ENDREPORT.
```

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 Manager용 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 안내서](#)를 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](#)
- [제한 사항](#)

ABAP에서의 데이터 표시

이 섹션은 다음 주제를 포함합니다.

주제

- [데이터 유형](#)
- [AWS 데이터 유형](#)

데이터 유형

AWS 서비스에는 ABAP 데이터 유형에 매핑해야 하는 표준 데이터 유형 세트가 있습니다. 자세한 내용은 다음 표를 참조하세요.

AWS 데이터 유형	ABAP 데이터 유형	설명
부울	C	단일 문자 "X" 및 " "
문자열	STRING	
바이트	INT2	INT2의 범위는 0~255보다 큼니다. 대부분의 AWS 서비스은 (는) 오버플로를 자르지만 이 동작은 공식적으로 정의되지 않았습니다.
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 DECFLOAT를 지원하지 않지만 NaN, Infinity, -Infinity와 같은 값은 표현할 수 없습니다. AWS SDK는 이를 내부적으로 STRING으로 표시하고 런타임 시 DECFLOAT16으로 변환합니다. NaN, Infinity 또는 +Infinity가 표시되는 경우 개발자는 특수한 예외 세트 또는 매핑에 대한 응답으로 이를 처리할 수 있습니다.
Double	STRING	
bigInteger	STRING	이 값은 ABAP로 표현할 수 없는 무한 길이의 숫자를 나타내
bigDecimal	STRING	

AWS 데이터 유형	ABAP 데이터 유형	설명
		며 BigInteger 대신 STRING이 사용됩니다.
타임스탬프	TZNTSTMP5	TZNTSTMP5 을(를) 사용하면 기본 ABAP 타임스탬프 함수를 사용하여 처리할 수 있습니다.

AWS 서비스은(는) 다음과 같은 집계 데이터 유형도 반환합니다.

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

AWS 데이터 유형

ABAP에서 AWS 서비스을(를) 지원하기 위해 다음과 같은 접근 방식이 통합되었습니다.

- 특정 AWS 데이터 유형은 ABAP로 표시할 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 JSON 또는 XML로 전송되며 값은 선택 사항입니다. 예를 들어, JSON에서 AWS 서비스(으)로 반환한 다음 예제를 참조하십시오.

```

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`을(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 프로파일 있습니다.
 - SDK 프로파일에서 논리적 IAM 역할 TESTUSER은(는) Amazon S3 버킷의 콘텐츠를 나열할 s3:ListBucket 권한을 부여하는 `arn:aws:iam::111122223333:role/SapDemoFinance`와 같은 IAM 역할에 매핑되어야 합니다.
 - SAP 시스템의 SID 및 클라이언트를 사용하여 Amazon S3 버킷에 매핑된 DEMO_BUCKET 논리적 리소스가 있습니다.
- 사용자의 PFCG 역할은 다음과 같습니다.
 - 사용자가 인증 객체 /AWS1/SESS을(를) 통해 DEMO_S3 SDK 프로파일 에 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 사용하여 AWS로 자체 인증이 가능합니다.
- Amazon EC2 인스턴스 프로파일은 SDK 프로파일 에 매핑된 IAM 역할 `arn:aws:iam::111122223333:role/SapDemoFinance`에 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_s3_factory, /aws1/cl_lmd_factory 및 /aws1/cl_dyn_factory에서 API 객체를 생성합니다.

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 SDK는 이러한 객체 지향 접근 방식을 취하며 모든 데이터는 객체와 테이블로 표시됩니다. LastModified 필드는 ABAP 기본 CONVERT TIME STAMP 명령을 사용하여 날짜로 변환할 수 있는 타임스탬프로 표시됩니다. 쉽게 수학 및 형식 지정 작업을 수행할 수 있도록 GET_SIZE()은(는) INT4을(를) 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 섹션에서는 의 기본 개념을 다룹니다 AWS SDK for SAP ABAP.

주제

- [API 클래스](#)
- [추가 객체](#)
- [구조 클래스](#)
- [배열](#)
- [맵](#)
- [상위 수준 함수](#)

API 클래스

각 문자에는 세 글자로 된 두문자어 또는 AWS 서비스 가 지정됩니다. TLA 서비스는 /AWS1/IF_<TLA> 형식의 인터페이스로 표시됩니다. 이를 서비스 인터페이스라고 부릅니다. API 클래스는 /AWS1/API_<TLA> 패키지에 있습니다. 서비스 인터페이스는 각 작업에 대해 하나의 메서드로 구성됨

니다. 이 메서드를 AWS Operation Methods라고 부릅니다. AWS SDK for SAP ABAP TLA의 전체 모듈 목록을 보려면 [AWS SDK for SAP ABAP - 모듈 목록을](#) 참조하십시오.

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 단순한 인수 집합으로 표현할 수 없는 딥 XML 및 JSON 구조를 전달하기 때문입니다 AWS 서비스 .

클래스에 속성이 하나뿐인 경우에도 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 보내고 받는 대부분의 AWS 데이터는 AWS SDK에 의해 클래스로 표시됩니다. 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 취급하며 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 스패스 배열이라고 합니다. ABAP에서는 객체 참조의 내부 표로 표시되며, null 값은 테이블에 실제 ABAP null 값으로 표시됩니다. 스패스 테이블을 반복할 때는 null 객체에 액세스하여 CX_SY_REF_IS_INITIAL 예외가 발생하지 않도록 null 값을 확인해야 합니다. 실제로 AWS 서비스에서 스패스 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_ec2secgrpiddstringlist_w=>tt_securitygroupidstringlist(
  ( NEW /aws1/
cl_ec2secgrpiddstringlist_w( 'sg-12345678' ) )
  ( NEW /aws1/
cl_ec2secgrpiddstringlist_w( 'sg-55555555' ) )
  ( NEW /aws1/
cl_ec2secgrpiddstringlist_w( 'sg-99999999' ) )
  )
  iv_subnetid              = ao_snet->get_subnetid( )
  it_tagspecifications     = make_tag_spec( 'instance' )
)

```

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 해시된 테이블이 객체 참조를 키 필드로 사용할 수 없고 AWS 맵 키는 항상 null이 아닌 문자열이기 때문에 필요합니다.

상위 수준 함수

이전 섹션에서 [API 클래스](#) 설명한 내용은 AWS 서비스 API를 정확히 미러링하고 이러한 API를 친숙한 ABAP 클래스로 나타냅니다. 경우에 따라 SDK에는 특정 작업을 단순화하기 위해 API 클래스를 기반으로 빌드되는 상위 수준 함수도 포함되어 있습니다. 상위 수준 함수는 프로그래머 편의를 위해 포함되었으며 하위 수준 API 클래스를 대체하지 않습니다.

SDK에 모듈에 대한 상위 수준 함수가 포함된 경우 이러한 함수는 동일한 전송에 포함되며 라는 팩토리 클래스를 통해 액세스할 수 있습니다. /AWS1/CL_TLA_L2_FACTORY [팩토리 클래스에는 모듈에 사용](#)

할 다양한 상위 수준 클라이언트를 만드는 메서드가 포함되어 있으며, 이러한 메서드는 API 설명서와 함께 나머지 API와 함께 문서화되어 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](#)

프로그래밍 방식 구성

SAP ABAP용 AWS SDK에는 /n/AWS1/IMG IMG 트랜잭션을 사용하고, 프로그래밍 방식 구성에는 BTP 에디션을 SAP ABAP용 AWS SDK용 사용자 지정 비즈니스 구성 응용 프로그램을 사용하십시오.

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

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

각 구성 개체는 IMG에 해당하는 GETters 및 SETters를 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 객체에는 에는 나와 있지 않은 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 안내서](#)를 참조하십시오.

Waiters

비동기 AWS API로 작업할 때는 추가 조치를 취하기 전에 특정 리소스를 사용할 수 있을 때까지 기다려야 합니다. 예를 들어 의 `CREATETABLE()` API는 테이블 상태로 즉시 Amazon DynamoDB 응답합니다. CREATING 테이블 상태가 ACTIVE로 변경된 후에만 읽기 또는 쓰기 작업을 시작할 수 있습니다. 웨이터를 사용하면 리소스에 작업을 수행하기 전에 AWS 리소스가 특정 상태인지 확인할 수 있습니다.

웨이터는 서비스 작업을 사용하여 AWS 리소스가 의도한 상태에 도달하거나 리소스가 원하는 상태에 도달하지 못했다고 판단될 때까지 리소스 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 = |example-bucket| ).
```

```
“ Wait until the newly created bucket becomes available
```

```
lo_s3->get_waiter( )->bucketexists(
    iv_max_wait_time = 200
    iv_bucket = |example-bucket|
).
```

- 이 예에서는 Amazon S3 클라이언트를 사용하여 버킷을 생성합니다. 이 `get_waiter()` 명령은 `bucketexists`의 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 오퍼레이션은 페이징 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 서비스에 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 AWS SDK의 [재시도 동작](#) 및 도구 참조 가이드를 참조하십시오.

SDK로 제품 구축

AWS 서비스를 소비하는 제품 또는 ABAP 애드온은 SDK의 기능을 향상시키고 확장할 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_ENGINE일 수 있습니다.

세션에 제품 ID를 추가하면 각 서비스 직접 호출과 함께 AWS에 전송되는 텔레메트리가 향상됩니다. 제품 ID 및 직접 호출을 수행하는 객체의 네임스페이스가 텔레메트리에 포함됩니다. 이 텔레메트리를 통해 AWS Support는 고객이 SDK와 관련하여 문제를 겪는 경우 직접 호출을 하는 제품을 식별할 수 있습니다. 이를 통해 고객의 코드가 아니라 제품이 실제로 직접 호출한 것임을 명확히 알 수 있습니다.

제한 사항

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

- 와 같이 MQTT iotevents 프로토콜 바인딩에 의존하는 모듈은 작동하지 않습니다. MQTTHTTP 기반 프로토콜이 아니며 현재 에서 지원되지 않습니다. AWS SDK for SAP ABAP
- HTTP/2 스트리밍 기능을 사용하는 모듈은 아직 지원되지 않습니다. 이벤트 스트림과 함께 작동하는 서비스의 특정 작업은 아직 지원되지 않으며, Amazon Kinesis Video Streams와 같은 서비스의 미디어 스트리밍 작업은 작동하지 않습니다.

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 SAP ABAP용 SDK - BTP 에디션에는 개발자 프리뷰에서 다음과 같은 제한 사항이 있습니다.

- 일부 모듈은 사용하지 못할 수 있습니다.
- 제거할 수 없습니다.
- 업데이트 빈도가 낮습니다.

SDK for SAP ABAP 코드 예제

이 항목의 코드 예제는 SAP ABAP용 AWS SDK를 에서 사용하는 방법을 보여줍니다. AWS

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 호출하는 방법을 보여 주며 관련 시나리오와 교차 서비스 예시에서 컨텍스트에 맞는 작업을 볼 수 있습니다.

시나리오는 동일한 서비스 내에서 여러 함수를 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예시입니다.

교차 서비스 예시는 여러 AWS 서비스 전반에서 작동하는 샘플 애플리케이션입니다.

예제

- [SDK for SAP ABAP를 사용한 작업 및 시나리오](#)

SDK for SAP ABAP를 사용한 작업 및 시나리오

다음 코드 예제는 SAP ABAP용 AWS SDK를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다. AWS 서비스

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 호출하는 방법을 보여 주며 관련 시나리오와 교차 서비스 예시에서 컨텍스트에 맞는 작업을 볼 수 있습니다.

시나리오는 동일한 서비스 내에서 여러 함수를 직접적으로 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예제입니다.

서비스

- [SDK for SAP ABAP를 사용한 Amazon Bedrock 런타임 예](#)
- [CloudWatch SAP ABAP용 SDK를 사용하는 예](#)
- [SDK for SAP ABAP를 사용한 DynamoDB 예](#)
- [SDK for SAP ABAP를 사용한 Amazon EC2 예](#)
- [SDK for SAP ABAP를 사용하는 Kinesis 예](#)
- [SDK for SAP ABAP를 사용한 Lambda 예](#)
- [SDK for SAP ABAP를 사용한 Amazon S3 예](#)
- [SageMaker SAP ABAP용 SDK를 사용하는 예](#)

- [SDK for SAP ABAP를 사용한 Amazon SNS에 대한 코드 예제](#)
- [SDK for SAP ABAP를 사용한 Amazon SQS 예](#)
- [SDK for SAP ABAP를 사용한 Amazon Textract 예](#)
- [SDK for SAP ABAP를 사용한 Amazon Translate 예](#)

SDK for SAP ABAP를 사용한 Amazon Bedrock 런타임 예

다음 코드 예제는 Amazon Bedrock Runtime과 함께 SAP ABAP용 AWS SDK를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 호출하는 방법을 보여 주며 관련 시나리오와 교차 서비스 예시에서 컨텍스트에 맞는 작업을 볼 수 있습니다.

시나리오는 동일한 서비스 내에서 여러 함수를 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예시입니다.

각 예제에는 상황에 맞게 GitHub 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 링크가 포함되어 있습니다.

주제

- [Anthropic Claude](#)
- [Stable Diffusion](#)

Anthropic Claude

InvokeModel

다음 코드 예제는 Invoke Model API를 사용하여 Anthropic Claude에 문자 메시지를 보내는 방법을 보여줍니다.

SDK for SAP ABAP

Note

자세한 내용은 다음과 같습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배워보세요.

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

```

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

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

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

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

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

" Claude V2 Response format will be:
* {

```

```

*      "completion": "Knock Knock...",
*      "stop_reason": "stop_sequence"
*    }
DATA: BEGIN OF ls_response,
      completion TYPE string,
      stop_reason TYPE string,
END OF ls_response.

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

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

ENDTRY.

```

Anthropic Claude 2 기초 모델을 호출하여 L2 하이 레벨 클라이언트를 사용하여 텍스트를 생성합니다.

```

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.

```

- API에 대한 자세한 내용은 SAP ABAP API용 AWS SDK [InvokeModel](#) 레퍼런스를 참조하십시오.

Stable Diffusion

InvokeModel

다음 코드 예제는 Amazon Bedrock에서 Stability.ai 스테이블 디퓨전 XL을 호출하여 이미지를 생성하는 방법을 보여줍니다.

SDK for SAP ABAP

Note

자세한 내용은 다음과 같습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

안정적 확산이 적용된 이미지를 만드세요.

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

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

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

```

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

TRY.
  DATA(lo_response) = lo_bdr->invokemodel(
    iv_body = /aws1/cl_rt_util=>string_to_xstring( lv_json )
    iv_modelid = 'stability.stable-diffusion-xl-v0'
    iv_accept = 'application/json'
    iv_contenttype = 'application/json' ).

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

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

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

```

```

WRITE / |Don't forget to enable model access at https://
console.aws.amazon.com/bedrock/home?#/modelaccess|.

ENDTRY.

```

Stability.ai 스테이블 디퓨전 XL 기반 모델을 호출하여 L2 하이 레벨 클라이언트를 사용하여 이미지를 생성합니다.

```

TRY.
  DATA(lo_bdr_l2_sd) = /aws1/
cl_bdr_l2_factory=>create_stable_diffusion_10( lo_bdr ).
  " iv_prompt contains a prompt like 'Show me a picture of a unicorn reading an
enterprise financial report'.
  DATA(lv_image) = lo_bdr_l2_sd->text_to_image( iv_prompt ).
  CATCH /aws1/cx_bdraccessdeniedex INTO DATA(lo_ex).
  WRITE / lo_ex->get_text( ).
  WRITE / |Don't forget to enable model access at https://
console.aws.amazon.com/bedrock/home?#/modelaccess|.

ENDTRY.

```

- API에 대한 자세한 내용은 SAP ABAP API용AWS SDK 레퍼런스를 참조하십시오 [InvokeModel](#).

CloudWatch SAP ABAP용 SDK를 사용하는 예

다음 코드 예제는 SAP ABAP용 AWS SDK를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다. CloudWatch

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 호출하는 방법을 보여 주며 관련 시나리오와 교차 서비스 예시에서 컨텍스트에 맞는 작업을 볼 수 있습니다.

시나리오는 동일한 서비스 내에서 여러 함수를 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예시입니다.

각 예제에는 GitHub 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 링크가 포함되어 있습니다.

주제

- [작업](#)
- [시나리오](#)

작업

DeleteAlarms

다음 코드 예시에서는 DeleteAlarms을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기를 참조하십시오 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 SAP용AWS SDK ABAP API 참조를 참조하십시오 [DeleteAlarms](#).

DescribeAlarms

다음 코드 예시에서는 DescribeAlarms을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 다음과 같습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 SAP용AWS SDK ABAP API 참조를 참조하십시오 [DescribeAlarms](#).

DisableAlarmActions

다음 코드 예시에서는 DisableAlarmActions을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 다음과 같습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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.

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [DisableAlarmActions](#).

EnableAlarmActions

다음 코드 예시에서는 EnableAlarmActions을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 다음과 같습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 SAP용AWS SDK ABAP API 참조를 참조하십시오 [EnableAlarmActions](#).

ListMetrics

다음 코드 예시에서는 ListMetrics을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 다음과 같습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 SAP용AWS SDK ABAP API 참조를 참조하십시오 [ListMetrics](#).

PutMetricAlarm

다음 코드 예시에서는 PutMetricAlarm을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 다음과 같습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 SAP용AWS SDK ABAP API 참조를 참조하십시오 [PutMetricAlarm](#).

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

자세한 내용은 다음과 같습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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."
CREATE OBJECT lo_alarmname EXPORTING iv_value = iv_alarm_name.
INSERT lo_alarmname INTO TABLE lt_alarmnames.

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

"Describe alarm using the same ABAP internal table."
TRY.

```

```

        oo_result = lo_cwt->describealarms(
returned for testing purpose "                                " oo_result is
        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'.
    ENDMETHOD.

    "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'.
    ENDMETHOD.

```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 다음 주제를 참조하세요.
 - [DeleteAlarms](#)
 - [DescribeAlarms](#)
 - [DisableAlarmActions](#)
 - [PutMetricAlarm](#)

SDK for SAP ABAP를 사용한 DynamoDB 예

다음 코드 예제는 DynamoDB와 함께 SAP ABAP용 AWS SDK를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 호출하는 방법을 보여 주며 관련 시나리오와 교차 서비스 예시에서 컨텍스트에 맞는 작업을 볼 수 있습니다.

시나리오는 동일한 서비스 내에서 여러 함수를 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예시입니다.

각 예제에는 상황에 맞게 GitHub 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 링크가 포함되어 있습니다.

주제

- [작업](#)
- [시나리오](#)

작업

CreateTable

다음 코드 예시에서는 CreateTable을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기를 참조하십시오 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

```
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 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [CreateTable](#).

DeleteItem

다음 코드 예시에서는 DeleteItem을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 다음과 같습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [DeleteItem](#).

DeleteTable

다음 코드 예시에서는 DeleteTable을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 다음과 같습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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_dynresourceindex.
  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 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [DeleteTable](#).

DescribeTable

다음 코드 예시에서는 DescribeTable을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 다음과 같습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 SAP용AWS SDK ABAP API 참조를 참조하십시오 [DescribeTable](#).

GetItem

다음 코드 예시에서는 GetItem을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 다음과 같습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [GetItem](#).

ListTables

다음 코드 예시에서는 ListTables을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 다음과 같습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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( ).
    ENDLLOOP.
    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 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [ListTables](#).

PutItem

다음 코드 예시에서는 PutItem을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 다음과 같습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [PutItem](#).

Query

다음 코드 예시에서는 Query을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 다음과 같습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

```

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.

```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 [Query](#)를 참조하세요.

Scan

다음 코드 예시에서는 Scan을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

더 많은 것이 있어요 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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.
  ENDLLOOP.
  DATA(lv_count) = oo_scan_result->get_count( ).
  MESSAGE 'Found ' && lv_count && ' items' TYPE 'I'.
  CATCH /aws1/cx_dynresourcenotfoundex.
    MESSAGE 'The table or index does not exist' TYPE 'E'.
  ENDTRY.

```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 [Scan](#)을 참조하세요.

UpdateItem

다음 코드 예시에서는 UpdateItem을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

더 많은 것이 있어요 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [UpdateItem](#).

시나리오

테이블, 항목 및 쿼리 시작

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

자세한 내용은 다음과 같습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

```
" 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_dynresourceinuseex.
        MESSAGE 'The table does not exist' TYPE 'E'.
    ENDTRY.

    " Put items into the table.
    TRY.
        DATA(lo_resp_putitem) = lo_dyn->putitem(
            iv_tablename = iv_table_name
            it_item       = VALUE /aws1/
cl_dynattributevalue=>tt_putiteminputattributemap(
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
                key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s =
'Jaws' ) ) )
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
                key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '1975' }| ) ) )
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
                key = 'rating' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '7.5' }| ) ) )
            ) ).
        lo_resp_putitem = lo_dyn->putitem(
            iv_tablename = iv_table_name
            it_item       = VALUE /aws1/
cl_dynattributevalue=>tt_putiteminputattributemap(
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
                key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s = 'Star
Wars' ) ) )
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
                key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '1978' }| ) ) )

```

```

        ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
          key = 'rating' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '8.1' }| ) ) ) )
      ) ).
    lo_resp_putitem = lo_dyn->putitem(
      iv_tablename = iv_table_name
      it_item       = VALUE /aws1/
cl_dynattributevalue=>tt_putiteminputattributemap(
      ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
        key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s =
'Speed' ) ) ) )
      ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
        key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '1994' }| ) ) ) )
      ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
        key = 'rating' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '7.9' }| ) ) ) )
      ) ).
    " TYPE REF TO ZCL_AWS1_dyn_PUT_ITEM_OUTPUT
    MESSAGE '3 rows inserted into DynamoDB Table' && iv_table_name TYPE 'I'.
    CATCH /aws1/cx_dyncondalcheckfaile00.
    MESSAGE 'A condition specified in the operation could not be evaluated.'
    TYPE 'E'.
    CATCH /aws1/cx_dynresourcenotfoundex.
    MESSAGE 'The table or index does not exist' TYPE 'E'.
    CATCH /aws1/cx_dyntransactconflictex.
    MESSAGE 'Another transaction is using the item' TYPE 'E'.
    ENDTRY.

    " Get item from table.
    TRY.
      DATA(lo_resp_getitem) = lo_dyn->getitem(
        iv_tablename          = iv_table_name
        it_key                 = VALUE /aws1/cl_dynattributevalue=>tt_key(
          ( VALUE /aws1/cl_dynattributevalue=>ts_key_maprow(
            key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s =
'Jaws' ) ) ) )
          ( VALUE /aws1/cl_dynattributevalue=>ts_key_maprow(
            key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n =
'1975' ) ) ) )
          ) ).
      DATA(lt_attr) = lo_resp_getitem->get_item( ).
      DATA(lo_title) = lt_attr[ key = 'title' ]-value.
      DATA(lo_year) = lt_attr[ key = 'year' ]-value.

```

```

    DATA(lo_rating) = lt_attr[ key = 'year' ]-value.
    MESSAGE 'Movie name is: ' && lo_title->get_s( ) TYPE 'I'.
    MESSAGE 'Movie year is: ' && lo_year->get_n( ) TYPE 'I'.
    MESSAGE 'Movie rating is: ' && lo_rating->get_n( ) TYPE 'I'.
  CATCH /aws1/cx_dynresourcenotfoundex.
    MESSAGE 'The table or index does not exist' TYPE 'E'.
  ENDRY.

" Query item from table.
TRY.
  DATA(lt_attributelist) = VALUE /aws1/
cl_dynattributevalue=>tt_attributelist(
    ( NEW /aws1/cl_dynattributevalue( iv_n = '1975' ) ) ).
  DATA(lt_keyconditions) = VALUE /aws1/cl_dyncondition=>tt_keyconditions(
    ( VALUE /aws1/cl_dyncondition=>ts_keyconditions_maprow(
      key = 'year'
      value = NEW /aws1/cl_dyncondition(
        it_attributelist = lt_attributelist
        iv_comparisonoperator = |EQ|
      ) ) ) ).
  DATA(lo_query_result) = lo_dyn->query(
    iv_tablename = iv_table_name
    it_keyconditions = lt_keyconditions ).
  DATA(lt_items) = lo_query_result->get_items( ).
  READ TABLE lo_query_result->get_items( ) INTO DATA(lt_item) INDEX 1.
  lo_title = lt_item[ key = 'title' ]-value.
  lo_year = lt_item[ key = 'year' ]-value.
  lo_rating = lt_item[ key = 'rating' ]-value.
  MESSAGE 'Movie name is: ' && lo_title->get_s( ) TYPE 'I'.
  MESSAGE 'Movie year is: ' && lo_year->get_n( ) TYPE 'I'.
  MESSAGE 'Movie rating is: ' && lo_rating->get_n( ) TYPE 'I'.
  CATCH /aws1/cx_dynresourcenotfoundex.
    MESSAGE 'The table or index does not exist' TYPE 'E'.
  ENDRY.

" 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](#)

SDK for SAP ABAP를 사용한 Amazon EC2 예

다음 코드 예제는 Amazon EC2와 함께 SAP ABAP용 AWS SDK를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 호출하는 방법을 보여 주며 관련 시나리오와 교차 서비스 예시에서 컨텍스트에 맞는 작업을 볼 수 있습니다.

시나리오는 동일한 서비스 내에서 여러 함수를 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예시입니다.

각 예제에는 상황에 GitHub 맞게 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 링크가 포함되어 있습니다.

주제

- [작업](#)

작업

AllocateAddress

다음 코드 예시에서는 AllocateAddress를 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기를 참조하십시오 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 SAP용AWS SDK ABAP API 참조를 참조하십시오 [AllocateAddress](#).

AssociateAddress

다음 코드 예시에서는 AssociateAddress를 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 다음과 같습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배워보세요.

```
TRY.
```

```

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

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [AssociateAddress](#).

CreateKeyPair

다음 코드 예시에서는 CreateKeyPair을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 다음과 같습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배워보세요.

```

    TRY.
        oo_result = lo_ec2->createkeypair( iv_keyname = iv_key_name ).
        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 SAP용AWS SDK ABAP API 참조를 참조하십시오 [CreateKeyPair](#).

CreateSecurityGroup

다음 코드 예시에서는 CreateSecurityGroup을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 다음과 같습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배워보세요.

```

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

```

" oo_result is returned for testing purposes. "

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [CreateSecurityGroup](#).

DeleteKeyPair

다음 코드 예시에서는 DeleteKeyPair을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 SAP용AWS SDK ABAP API 참조를 참조하십시오 [DeleteKeyPair](#).

DeleteSecurityGroup

다음 코드 예시에서는 DeleteSecurityGroup을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 SAP용AWS SDK ABAP API 참조를 참조하십시오 [DeleteSecurityGroup](#).

DescribeAddresses

다음 코드 예시에서는 DescribeAddresses을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 SAP용AWS SDK ABAP API 참조를 참조하십시오 [DescribeAddresses](#).

DescribeAvailabilityZones

다음 코드 예시에서는 DescribeAvailabilityZones을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 SAP용AWS SDK ABAP API 참조를 참조하십시오 [DescribeAvailabilityZones](#).

DescribeInstances

다음 코드 예시에서는 DescribeInstances을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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_instance_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_instance_id = lo_instance->get_instanceid( ).
      lv_status = lo_instance->get_state( )->get_name( ).
      lv_instance_type = lo_instance->get_instancetype( ).
      lv_image_id = lo_instance->get_imageid( ).
    ENDLLOOP.
  ENDLLOOP.
  MESSAGE 'Retrieved information about EC2 instances.' TYPE 'I'.

```

```

    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
      DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
      MESSAGE lv_error TYPE 'E'.
    ENDTRY.

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [DescribeInstances](#).

DescribeKeyPairs

다음 코드 예시에서는 DescribeKeyPairs를 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 SAP용AWS SDK ABAP API 참조를 참조하십시오 [DescribeKeyPairs](#).

DescribeRegions

다음 코드 예시에서는 DescribeRegions를 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

```

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 SAP용AWS SDK ABAP API 참조를 참조하십시오 [DescribeRegions](#).

DescribeSecurityGroups

다음 코드 예시에서는 DescribeSecurityGroups을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

```

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 SAP용AWS SDK ABAP API 참조를 참조하십시오 [DescribeSecurityGroups](#).

MonitorInstances

다음 코드 예시에서는 MonitorInstances을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 SAP용AWS SDK ABAP API 참조를 참조하십시오 [MonitorInstances](#).

RebootInstances

다음 코드 예시에서는 RebootInstances을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 SAP용AWS SDK ABAP API 참조를 참조하십시오 [RebootInstances](#).

ReleaseAddress

다음 코드 예시에서는 ReleaseAddress를 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 SAP용AWS SDK ABAP API 참조를 참조하십시오 [ReleaseAddress](#).

RunInstances

다음 코드 예시에서는 RunInstances을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배워보세요.

```

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

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

```

```

        iv_imageid = iv_ami_id
        iv_instancetype = 't2.micro'
        iv_maxcount = 1
        iv_mincount = 1
        it_tagspecifications = lt_tagspecifications
        iv_subnetid = iv_subnet_id
    ).
    MESSAGE 'EC2 instance created.' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception->av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [RunInstances](#).

StartInstances

다음 코드 예시에서는 StartInstances을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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.
ENDTRY.

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [StartInstances](#).

StopInstances

다음 코드 예시에서는 StopInstances을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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.
  ENDTRY.

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [StopInstances](#).

SDK for SAP ABAP를 사용하는 Kinesis 예

다음 코드 예제는 Kinesis와 함께 SAP ABAP용 AWS SDK를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 호출하는 방법을 보여 주며 관련 시나리오와 교차 서비스 예시에서 컨텍스트에 맞는 작업을 볼 수 있습니다.

시나리오는 동일한 서비스 내에서 여러 함수를 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예시입니다.

각 예제에는 상황에 GitHub 맞게 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 링크가 포함되어 있습니다.

주제

- [작업](#)
- [시나리오](#)

작업

CreateStream

다음 코드 예시에서는 CreateStream을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기를 참조하십시오 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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_knslimitexceeddex .
    MESSAGE 'The request processing has failed because of a limit exceed
exception.' TYPE 'E'.
    CATCH /aws1/cx_knsresourceinuseex .
    MESSAGE 'The request processing has failed because the resource is in use.'
TYPE 'E'.
    ENDTRY.

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [CreateStream](#).

DeleteStream

다음 코드 예시에서는 DeleteStream을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배워보세요.

```

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

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [DeleteStream](#).

DescribeStream

다음 코드 예시에서는 DescribeStream을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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_knslimitexceedex .  
    MESSAGE 'The request processing has failed because of a limit exceed  
exception.' TYPE 'E'.  
CATCH /aws1/cx_knsresourcenotfoundex .  
    MESSAGE 'Resource being accessed is not found.' TYPE 'E'.  
ENDTRY.
```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [DescribeStream](#).

GetRecords

다음 코드 예시에서는 GetRecords을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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_knskmsstrottlingex .
    MESSAGE 'The rate of requests to AWS KMS is exceeding the request quotas.'
TYPE 'E'.
CATCH /aws1/cx_knsprovthruputexcdex .
    MESSAGE 'The request rate for the stream is too high, or the requested data
is too large for the available throughput.' TYPE 'E'.
CATCH /aws1/cx_knsresourcenotfoundex .
    MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
ENDTRY.

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [GetRecords](#).

ListStreams

다음 코드 예시에서는 ListStreams을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

```

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

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [ListStreams](#).

PutRecord

다음 코드 예시에서는 PutRecord를 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

```

TRY.
    oo_result = lo_kns->putrecord(           " oo_result is returned for
testing purposes. "

```

```

        iv_streamname = iv_stream_name
        iv_data        = iv_data
        iv_partitionkey = iv_partition_key
    ).
    MESSAGE 'Record created.' TYPE 'I'.
CATCH /aws1/cx_knsinvalidargumentex .
    MESSAGE 'The specified argument was not valid.' TYPE 'E'.
CATCH /aws1/cx_knskmsaccessdeniedex .
    MESSAGE 'You do not have permission to perform this AWS KMS action.' TYPE
'E'.
CATCH /aws1/cx_knskmsdisabledex .
    MESSAGE 'KMS key used is disabled.' TYPE 'E'.
CATCH /aws1/cx_knskmsinvalidstateex .
    MESSAGE 'KMS key used is in an invalid state. ' TYPE 'E'.
CATCH /aws1/cx_knskmsnotfoundex .
    MESSAGE 'KMS key used is not found.' TYPE 'E'.
CATCH /aws1/cx_knskmsoptinrequired .
    MESSAGE 'KMS key option is required.' TYPE 'E'.
CATCH /aws1/cx_knskmsstrottlingex .
    MESSAGE 'The rate of requests to AWS KMS is exceeding the request quotas.'
TYPE 'E'.
CATCH /aws1/cx_knsprovthruputexcdex .
    MESSAGE 'The request rate for the stream is too high, or the requested data
is too large for the available throughput.' TYPE 'E'.
CATCH /aws1/cx_knsresourcenotfoundex .
    MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
ENDTRY.

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [PutRecord](#).

RegisterStreamConsumer

다음 코드 예시에서는 RegisterStreamConsumer을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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_sgmresource-limitexcd.
    MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
CATCH /aws1/cx_sgmresource-inuse.
    MESSAGE 'Resource being accessed is in use.' TYPE 'E'.
CATCH /aws1/cx_sgmresource-notfound.
    MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
ENDTRY.

```

- API 세부 정보는 SAP용 AWS SDK ABAP API 참조를 참조하십시오 [RegisterStreamConsumer](#).

시나리오

데이터 스트림 시작하기

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

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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_knskms throttlingex .
  MESSAGE 'The rate of requests to AWS KMS is exceeding the request quotas.'
TYPE 'E'.
CATCH /aws1/cx_knsprovthruputexcdex .
  MESSAGE 'The request rate for the stream is too high, or the requested data
is too large for the available throughput.' TYPE 'E'.
CATCH /aws1/cx_knsresourcenotfoundex .
  MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
ENDTRY.

"Create a shard iterator in order to read the record."
TRY.
  lo_sharditerator = lo_kns->getsharditerator(
    iv_shardid = lo_record_result->get_shardid( )
    iv_sharditeratortype = iv_sharditeratortype
    iv_streamname = iv_stream_name
  ).
  MESSAGE 'Shard iterator created.' TYPE 'I'.
CATCH /aws1/cx_knsinvalidargumentex.
  MESSAGE 'The specified argument was not valid.' TYPE 'E'.
CATCH /aws1/cx_knsprovthruputexcdex .
  MESSAGE 'The request rate for the stream is too high, or the requested data
is too large for the available throughput.' TYPE 'E'.
CATCH /aws1/cx_sgmresourcenotfound.
  MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
ENDTRY.

"Read the record."
TRY.
  oo_result = lo_kns->getrecords(
    " oo_result is returned
for testing purposes. "

```

```

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

"Delete stream."
TRY.
    lo_kns->deletestream(
        iv_streamname = iv_stream_name
    ).
    MESSAGE 'Stream deleted.' TYPE 'I'.
    CATCH /aws1/cx_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](#)

SDK for SAP ABAP를 사용한 Lambda 예

다음 코드 예제는 Lambda와 함께 SAP ABAP용 AWS SDK를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 호출하는 방법을 보여 주며 관련 시나리오와 교차 서비스 예시에서 컨텍스트에 맞는 작업을 볼 수 있습니다.

시나리오는 동일한 서비스 내에서 여러 함수를 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예시입니다.

각 예제에는 컨텍스트에서 GitHub 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 링크가 포함되어 있습니다.

주제

- [작업](#)
- [시나리오](#)

작업

CreateFunction

다음 코드 예시에서는 CreateFunction을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기를 참조하십시오 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

```

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 SAP용AWS SDK ABAP API 참조를 참조하십시오 [CreateFunction](#).

DeleteFunction

다음 코드 예시에서는 DeleteFunction을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 SAP용AWS SDK ABAP API 참조를 참조하십시오 [DeleteFunction](#).

GetFunction

다음 코드 예시에서는 GetFunction을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 SAP용AWS SDK ABAP API 참조를 참조하십시오 [GetFunction](#).

Invoke

다음 코드 예시에서는 Invoke을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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(
testing purposes. "
        iv_functionname = iv_function_name
        iv_payload = lv_json
    ).
    MESSAGE 'Lambda function invoked.' TYPE 'I'.

```

```

CATCH /aws1/cx_lmdinvparamvalueex.
  MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
CATCH /aws1/cx_lmdinvrequestcontex.
  MESSAGE 'Unable to parse request body as JSON.' TYPE 'E'.
CATCH /aws1/cx_lmdinvalidzipfileex.
  MESSAGE 'The deployment package could not be unzipped.' TYPE 'E'.
CATCH /aws1/cx_lmdrequesttoolargeex.
  MESSAGE 'Invoke request body JSON input limit was exceeded by the request
payload.' TYPE 'E'.
CATCH /aws1/cx_lmdresourceconflictex.
  MESSAGE 'Resource already exists or another operation is in progress.' TYPE
'E'.
CATCH /aws1/cx_lmdresourcenotfoundex.
  MESSAGE 'The requested resource does not exist.' TYPE 'E'.
CATCH /aws1/cx_lmdserviceexception.
  MESSAGE 'An internal problem was encountered by the AWS Lambda service.'
TYPE 'E'.
CATCH /aws1/cx_lmdtoomanyrequestsex.
  MESSAGE 'The maximum request throughput was reached.' TYPE 'E'.
CATCH /aws1/cx_lmdunsuppmediatyp00.
  MESSAGE 'Invoke request body does not have JSON as its content type.' TYPE
'E'.
ENDTRY.

```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 [Invoke](#)를 참조하세요.

ListFunctions

다음 코드 예시에서는 ListFunctions을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

더 많은 것이 있어요 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 SAP용AWS SDK ABAP API 참조를 참조하십시오 [ListFunctions](#).

UpdateFunctionCode

다음 코드 예시에서는 UpdateFunctionCode을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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_lmdcodestorageexc dex.
    MESSAGE 'Maximum total code size per account exceeded.' TYPE 'E'.
    CATCH /aws1/cx_lmdcodeverification00.
    MESSAGE 'Code signature failed one or more validation checks for signature
mismatch or expiration.' TYPE 'E'.

```

```

CATCH /aws1/cx_lmdinvalidcodesigex.
  MESSAGE 'Code signature failed the integrity check.' TYPE 'E'.
CATCH /aws1/cx_lmdinvparamvalueex.
  MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
CATCH /aws1/cx_lmdresourceconflictex.
  MESSAGE 'Resource already exists or another operation is in progress.' TYPE
'E'.
CATCH /aws1/cx_lmdresourcenotfoundex.
  MESSAGE 'The requested resource does not exist.' TYPE 'E'.
CATCH /aws1/cx_lmdserviceexception.
  MESSAGE 'An internal problem was encountered by the AWS Lambda service.'
TYPE 'E'.
CATCH /aws1/cx_lmdtoomanyrequestsex.
  MESSAGE 'The maximum request throughput was reached.' TYPE 'E'.
ENDTRY.

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [UpdateFunctionCode](#).

UpdateFunctionConfiguration

다음 코드 예시에서는 UpdateFunctionConfiguration을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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.

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [UpdateFunctionConfiguration](#).

시나리오

함수 시작하기

다음 코드 예제에서는 다음과 같은 작업을 수행하는 방법을 보여줍니다.

- IAM 역할과 Lambda 함수를 생성하고 핸들러 코드를 업로드합니다.
- 단일 파라미터로 함수를 간접적으로 호출하고 결과를 가져옵니다.
- 함수 코드를 업데이트하고 환경 변수로 구성합니다.
- 새 파라미터로 함수를 간접적으로 호출하고 결과를 가져옵니다. 반환된 실행 로그를 표시합니다.
- 계정의 함수를 나열합니다.

자세한 내용은 [콘솔로 Lambda 함수 생성](#)을 참조하십시오.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

```

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

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

```

```
        MESSAGE 'Policy document in the request is malformed.' TYPE 'E'.
    ENDDTRY.

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

    " Create a Lambda function and upload handler code. "
    " Lambda function performs 'increment' action on a number. "
    TRY.
        lo_lmd->createfunction(
            iv_functionname = iv_function_name
            iv_runtime = `python3.9`
            iv_role = lo_create_role_output->get_role( )->get_arn( )
            iv_handler = iv_handler
            io_code = io_initial_zip_file
            iv_description = 'AWS Lambda code example'
        ).
        MESSAGE 'Lambda function created.' TYPE 'I'.
    CATCH /aws1/cx_lmdcodestorageexcdex.
        MESSAGE 'Maximum total code size per account exceeded.' TYPE 'E'.
    CATCH /aws1/cx_lmdinvparamvalueex.
        MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
    CATCH /aws1/cx_lmdresourcenotfoundex.
        MESSAGE 'The requested resource does not exist.' TYPE 'E'.
    ENDDTRY.

    " Verify the function is in Active state "
```

```

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

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

    " Update the function code and configure its Lambda environment with an
environment variable. "
    " Lambda function is updated to perform 'decrement' action also. "
    TRY.

```

```

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

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

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

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

" Delete the IAM role. "
TRY.
  lo_iam->deleterole( iv_rolename = iv_role_name ).
  MESSAGE 'IAM role deleted.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'The requested resource entity does not exist.' TYPE 'E'.
  CATCH /aws1/cx_iamunmodableentityex.
    MESSAGE 'Service that depends on the service-linked role is not
modifiable.' TYPE 'E'.
  ENDTRY.

  CATCH /aws1/cx_rt_service_generic INTO lo_exception.
    DATA(lv_error) = lo_exception->get_longtext( ).
    MESSAGE lv_error TYPE 'E'.
  ENDTRY.

```

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

- [CreateFunction](#)
- [DeleteFunction](#)
- [GetFunction](#)
- [Invoke](#)
- [ListFunctions](#)
- [UpdateFunctionCode](#)
- [UpdateFunctionConfiguration](#)

SDK for SAP ABAP를 사용한 Amazon S3 예

다음 코드 예제는 Amazon S3와 함께 SAP ABAP용 AWS SDK를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 호출하는 방법을 보여 주며 관련 시나리오와 교차 서비스 예시에서 컨텍스트에 맞는 작업을 볼 수 있습니다.

시나리오는 동일한 서비스 내에서 여러 함수를 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예시입니다.

각 예제에는 상황에 GitHub 맞게 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 링크가 포함되어 있습니다.

주제

- [작업](#)
- [시나리오](#)

작업

CopyObject

다음 코드 예시에서는 CopyObject를 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기를 참조하십시오 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

```
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 SAP용AWS SDK ABAP API 참조를 참조하십시오 [CopyObject](#).

CreateBucket

다음 코드 예시에서는 CreateBucket을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

```
TRY.
  lo_s3->createbucket(
    iv_bucket = iv_bucket_name
```

```

    ).
    MESSAGE 'S3 bucket created.' TYPE 'I'.
  CATCH /aws1/cx_s3_bucketalrddyexists.
    MESSAGE 'Bucket name already exists.' TYPE 'E'.
  CATCH /aws1/cx_s3_bktalrddyownedbyyou.
    MESSAGE 'Bucket already exists and is owned by you.' TYPE 'E'.
  ENDTRY.

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [CreateBucket](#).

DeleteBucket

다음 코드 예시에서는 DeleteBucket을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 SAP용AWS SDK ABAP API 참조를 참조하십시오 [DeleteBucket](#).

DeleteObject

다음 코드 예시에서는 DeleteObject을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 SAP용AWS SDK ABAP API 참조를 참조하십시오 [DeleteObject](#).

GetObject

다음 코드 예시에서는 GetObject을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 SAP용AWS SDK ABAP API 참조를 참조하십시오 [GetObject](#).

ListObjectsV2

다음 코드 예시에서는 ListObjectsV2을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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.

```

- API에 대한 자세한 내용은 SAP ABAP API 참조용AWS SDK의 ListObjects [V2](#)를 참조하십시오.

PutObject

다음 코드 예시에서는 PutObject을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 SAP용AWS SDK ABAP API 참조를 참조하십시오 [PutObject](#).

시나리오

버킷 및 객체 시작하기

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

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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.
  lo_s3->createbucket(
    iv_bucket = iv_bucket_name
  ).
  MESSAGE 'S3 bucket created.' TYPE 'I'.
CATCH /aws1/cx_s3_bucketalrddyexists.
  MESSAGE 'Bucket name already exists.' TYPE 'E'.
CATCH /aws1/cx_s3_bktalrddyownedbyyou.
  MESSAGE 'Bucket already exists and is owned by you.' TYPE 'E'.
ENDTRY.

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

  lo_s3->putobject(
    iv_bucket = iv_bucket_name
    iv_key = iv_key
    iv_body = lv_file_content
  ).
  MESSAGE 'Object uploaded to S3 bucket.' TYPE 'I'.
```

```
CATCH /aws1/cx_s3_nosuchbucket.
  MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.

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

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

SageMaker SAP ABAP용 SDK를 사용하는 예

다음 코드 예제는 SAP ABAP용 AWS SDK를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다. SageMaker

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 호출하는 방법을 보여 주며 관련 시나리오와 교차 서비스 예시에서 컨텍스트에 맞는 작업을 볼 수 있습니다.

시나리오는 동일한 서비스 내에서 여러 함수를 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예시입니다.

각 예제에는 GitHub 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 링크가 포함되어 있습니다.

주제

- [작업](#)
- [시나리오](#)

작업

CreateEndpoint

다음 코드 예시에서는 CreateEndpoint을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기를 참조하십시오 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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."
CREATE OBJECT lo_production_variants
EXPORTING
  iv_variantname      = iv_variant_name
  iv_modelname        = iv_model_name
  iv_initialinstancecount = iv_initial_instance_count
  iv_instancetype     = iv_instance_type.

INSERT lo_production_variants INTO TABLE lt_production_variants.

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

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

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [CreateEndpoint](#).

CreateModel

다음 코드 예시에서는 CreateModel을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

```
DATA lo_primarycontainer TYPE REF TO /aws1/cl_sgmcontainerdefn.

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

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

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [CreateModel](#).

CreateTrainingJob

다음 코드 예시에서는 CreateTrainingJob을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

 Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

```
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_sgmalgorithm-spec.
DATA lo_resource_config TYPE REF TO /aws1/cl_sgmresourceconfig.
DATA lo_output_data_config TYPE REF TO /aws1/cl_sgmoutputdataconfig.
DATA lo_stopping_condition TYPE REF TO /aws1/cl_sgmstoppingcondition.
```

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

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

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

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

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

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

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

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

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

"Create ABAP objects for training data sources."

```
CREATE OBJECT lo_trn_s3datasource  
EXPORTING  
    iv_s3datatype           = iv_trn_data_s3datatype  
    iv_s3datadistributiontype = iv_trn_data_s3datadistribution  
    iv_s3uri                = iv_trn_data_s3uri.
```

```
CREATE OBJECT lo_trn_datasource  
EXPORTING  
    io_s3datasource = lo_trn_s3datasource.
```

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

```
INSERT lo_trn_channel INTO TABLE lt_input_data_config.
```

"Create ABAP objects for validation data sources."

```
CREATE OBJECT lo_val_s3datasource  
EXPORTING  
    iv_s3datatype           = iv_val_data_s3datatype  
    iv_s3datadistributiontype = iv_val_data_s3datadistribution  
    iv_s3uri                = iv_val_data_s3uri.
```

```
CREATE OBJECT lo_val_datasource  
EXPORTING  
    io_s3datasource = lo_val_s3datasource.
```

```
CREATE OBJECT lo_val_channel
```

```

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

INSERT lo_val_channel INTO TABLE lt_input_data_config.

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

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

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

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

"Create a training job."
TRY.
  oo_result = lo_sgm->createtrainingjob( " oo_result is returned for
testing purposes. "
    iv_trainingjobname      = iv_training_job_name
    iv_rolearn              = iv_role_arn
    it_hyperparameters      = lt_hyperparameters
    it_inputdataconfig      = lt_input_data_config
    io_algorithmspecification = lo_algorithm_specification
    io_outputdataconfig     = lo_output_data_config
    io_resourceconfig       = lo_resource_config
    io_stoppingcondition    = lo_stopping_condition
  ).

```

```

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

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [CreateTrainingJob](#).

CreateTransformJob

다음 코드 예시에서는 CreateTransformJob을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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."

```

CREATE OBJECT lo_s3datasource
EXPORTING
    iv_s3uri      = iv_tf_data_s3uri
    iv_s3datatype = iv_tf_data_s3datatype.

```

"Create an ABAP object for data source."

```

CREATE OBJECT lo_datasource
EXPORTING

```

```

        io_s3datasource = lo_s3datasource.

"Create an ABAP object for transform data source."
CREATE OBJECT lo_transforminput
EXPORTING
    io_datasource      = lo_datasource
    iv_contenttype     = iv_tf_data_contenttype
    iv_compressiontype = iv_tf_data_compressiontype.

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

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

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

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [CreateTransformJob](#).

DeleteEndpoint

다음 코드 예시에서는 DeleteEndpoint을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

```
"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 SAP용AWS SDK ABAP API 참조를 참조하십시오 [DeleteEndpoint](#).

DeleteModel

다음 코드 예시에서는 DeleteModel을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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.
```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [DeleteModel](#).

DescribeTrainingJob

다음 코드 예시에서는 DescribeTrainingJob을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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'.
  ENDMETHOD.

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [DescribeTrainingJob](#).

ListAlgorithms

다음 코드 예시에서는 ListAlgorithms을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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'.
  ENDMETHOD.

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [ListAlgorithms](#).

ListModels

다음 코드 예시에서는 ListModels을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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.

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [ListModels](#).

ListNotebookInstances

다음 코드 예시에서는 ListNotebookInstances을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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.

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [ListNotebookInstances](#).

ListTrainingJobs

다음 코드 예시에서는 ListTrainingJobs을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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.

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [ListTrainingJobs](#).

시나리오

모델 및 엔드포인트를 사용하여 시작하기

다음 코드 예시는 다음과 같은 작업을 수행하는 방법을 보여줍니다.

- 교육 작업을 시작하고 모델을 SageMaker 생성하세요.
- 엔드포인트 구성을 생성합니다.
- 엔드포인트를 생성한 다음 리소스를 정리합니다.

SDK for SAP ABAP

Note

더 많은 정보가 있습니다 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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_sgmalgorithm_spec.
DATA lo_resource_config TYPE REF TO /aws1/cl_sgmresourceconfig.
DATA lo_output_data_config TYPE REF TO /aws1/cl_sgmoutputdataconfig.
DATA lo_stopping_condition TYPE REF TO /aws1/cl_sgmstoppingcondition.
DATA lo_primarycontainer TYPE REF TO /aws1/cl_sgmcontainerdefn.
DATA lo_production_variants TYPE REF TO /aws1/cl_sgmproductionvariant.
DATA lo_ep_config_result TYPE REF TO /aws1/cl_sgmcreateendptcfgout.
DATA lo_training_result TYPE REF TO /aws1/cl_sgmdescrtrnjobrsp.
DATA lt_production_variants TYPE /aws1/
cl_sgmproductionvariant=>tt_productionvariantlist.
DATA lt_input_data_config TYPE /aws1/cl_sgmchannel=>tt_inputdataconfig.
DATA lt_hyperparameters TYPE /aws1/cl_sgmhyperparameters_w=>tt_hyperparameters.
DATA lv_model_data_url TYPE /aws1/sgmurl.
```

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

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

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

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

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

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

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

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

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

CREATE OBJECT lo_trn_datasource EXPORTING io_s3datasource = lo_trn_s3datasource.
```

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

"Validation data."
CREATE OBJECT lo_val_s3datasource
  EXPORTING
    iv_s3datatype       = iv_val_data_s3datatype
    iv_s3datadistributiontype = iv_val_data_s3datadistribution
    iv_s3uri            = iv_val_data_s3uri.

CREATE OBJECT lo_val_datasource EXPORTING io_s3datasource = lo_val_s3datasource.

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

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

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

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

"Create an ABAP object for stopping condition."
```

```
CREATE OBJECT lo_stopping_condition EXPORTING iv_maxruntimeinseconds =
iv_max_runtime_in_seconds.

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

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

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

"Create an Amazon SageMaker model."
TRY.
  lo_sgm->createmodel(
```

```

        iv_executionrolearn = iv_role_arn
        iv_modelname = iv_model_name
        io_primarycontainer = lo_primarycontainer
    ).
    MESSAGE 'Model created.' TYPE 'I'.
CATCH /aws1/cx_sgmresourcecelimitexcd.
    MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
ENDTRY.

"Create an endpoint production variant."
CREATE OBJECT lo_production_variants
EXPORTING
    iv_variantname          = iv_ep_variant_name
    iv_modelname            = iv_model_name
    iv_initialinstancecount = iv_ep_initial_instance_count
    iv_instancetype         = iv_ep_instance_type.
INSERT lo_production_variants INTO TABLE lt_production_variants.

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

    "Create an endpoint."
    oo_ep_output = lo_sgm->createendpoint(          " 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_sgmresourcecelimitexcd.
    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 Amazon SNS에 대한 코드 예제

다음 코드 예제는 Amazon SNS와 함께 SAP ABAP용 AWS SDK를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 호출하는 방법을 보여 주며 관련 시나리오와 교차 서비스 예시에서 컨텍스트에 맞는 작업을 볼 수 있습니다.

시나리오는 동일한 서비스 내에서 여러 함수를 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예시입니다.

각 예제에는 상황에 GitHub 맞게 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 링크가 포함되어 있습니다.

주제

- [작업](#)
- [시나리오](#)

작업

CreateTopic

다음 코드 예시에서는 CreateTopic을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기를 참조하십시오 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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.
```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [CreateTopic](#).

DeleteTopic

다음 코드 예시에서는 DeleteTopic을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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.
```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [DeleteTopic](#).

GetTopicAttributes

다음 코드 예시에서는 GetTopicAttributes을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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.

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [GetTopicAttributes](#).

ListSubscriptions

다음 코드 예시에서는 ListSubscriptions을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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.

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [ListSubscriptions](#).

ListTopics

다음 코드 예시에서는 ListTopics을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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.
```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [ListTopics](#).

Publish

다음 코드 예시에서는 Publish을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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.
```

- API에 대한 세부 정보는 AWS SDK for SAP ABAP API 참조의 [Publish](#)를 참조하세요.

SetTopicAttributes

다음 코드 예시에서는 SetTopicAttributes을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

더 많은 것이 있어요 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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.
```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [SetTopicAttributes](#).

Subscribe

다음 코드 예시에서는 Subscribe을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배워보세요.

주제에 대한 이메일 주소를 구독하세요.

```

TRY.
    oo_result = lo_sns->subscribe(
        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_snssubscriptionlmt00.
    MESSAGE 'Unable to create subscriptions. You have reached the maximum number
of subscriptions allowed.' TYPE 'E'.
ENDTRY.

```

"oo_result is returned

- API에 대한 세부 정보는 [AWS SDK for SAP ABAP API 참조](#)의 Subscribe를 참조하세요.

Unsubscribe

다음 코드 예시에서는 Unsubscribe을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

더 많은 정보가 있습니다 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 [AWS SDK for SAP ABAP API 참조](#)의 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

더 많은 것이 있어요 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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_snstopicattrsmw=>tt_topicattributesmap.
DATA ls_tpc_attributes TYPE /aws1/
cl_snstopicattrsmw=>ts_topicattributesmap_maprow.
ls_tpc_attributes-key = 'FifoTopic'.
ls_tpc_attributes-value = NEW /aws1/cl_snstopicattrsmw( iv_value = 'true' ).
INSERT ls_tpc_attributes INTO TABLE lt_tpc_attributes.

TRY.

```

```

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

" Subscribes an endpoint to an Amazon Simple Notification Service (Amazon SNS)
topic. "
" Only Amazon Simple Queue Service (Amazon SQS) FIFO queues can be subscribed to
an SNS FIFO topic. "
        TRY.
            DATA(lo_subscribe_result) = lo_sns->subscribe(
                iv_topicarn = lv_topic_arn
                iv_protocol = 'sqs'
                iv_endpoint = iv_queue_arn
            ).
            DATA(lv_subscription_arn) = lo_subscribe_result->get_subscriptionarn( ).
            ov_subscription_arn = lv_subscription_arn.
"
ov_subscription_arn is returned for testing purposes. "
            MESSAGE 'SQS queue was subscribed to SNS topic.' TYPE 'I'.
            CATCH /aws1/cx_snsnotfoundexception.
                MESSAGE 'Topic does not exist.' TYPE 'E'.
            CATCH /aws1/cx_snssubscriptionlmte00.
                MESSAGE 'Unable to create subscriptions. You have reached the maximum number
of subscriptions allowed.' TYPE 'E'.
            ENDTRY.

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

```

```

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

```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 다음 주제를 참조하세요.
 - [CreateTopic](#)
 - [게시](#)
 - [Subscribe](#)

SDK for SAP ABAP를 사용한 Amazon SQS 예

다음 코드 예제는 Amazon SQS와 함께 SAP ABAP용 AWS SDK를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 호출하는 방법을 보여 주며 관련 시나리오와 교차 서비스 예시에서 컨텍스트에 맞는 작업을 볼 수 있습니다.

시나리오는 동일한 서비스 내에서 여러 함수를 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예시입니다.

각 예제에는 상황에 GitHub 맞게 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 링크가 포함되어 있습니다.

주제

- [작업](#)
- [시나리오](#)

작업

CreateQueue

다음 코드 예시에서는 CreateQueue을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기를 참조하십시오 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배워보세요.

Amazon SQS 표준 대기열을 생성합니다.

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

메시지가 도착하기를 기다리는 Amazon SQS 대기열을 생성합니다.

```
TRY.
    DATA lt_attributes TYPE /aws1/cl_sqsqueueattribmap_w=>tt_queueattributemap.
    DATA ls_attribute TYPE /aws1/
cl_sqsqueueattribmap_w=>ts_queueattributemap_maprow.
    ls_attribute-key = 'ReceiveMessageWaitTimeSeconds'.
    " Time in
seconds for long polling, such as how long the call waits for a message to arrive
in the queue before returning. "
```

```

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

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [CreateQueue](#).

DeleteQueue

다음 코드 예시에서는 DeleteQueue을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배워보세요.

```

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

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [DeleteQueue](#).

GetQueueUrl

다음 코드 예시에서는 GetQueueUrl을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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.
```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [GetQueueUrl](#).

ListQueues

다음 코드 예시에서는 ListQueues을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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.
```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [ListQueues](#).

ReceiveMessage

다음 코드 예시에서는 ReceiveMessage을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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.
```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [ReceiveMessage](#).

SendMessage

다음 코드 예시에서는 SendMessage을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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.
```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [SendMessage](#).

시나리오

FIFO 주제에 생성 및 게시

다음 코드 예제에서는 FIFO Amazon SNS 주제를 생성하고 거기에 게시하는 방법을 보여줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

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

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

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

```

```

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

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

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

```

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

- [CreateTopic](#)
- [게시](#)
- [Subscribe](#)

SDK for SAP ABAP를 사용한 Amazon Textract 예

다음 코드 예제는 Amazon Textract와 함께 SAP ABAP용 AWS SDK를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 호출하는 방법을 보여 주며 관련 시나리오와 교차 서비스 예시에서 컨텍스트에 맞는 작업을 볼 수 있습니다.

시나리오는 동일한 서비스 내에서 여러 함수를 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예시입니다.

각 예제에는 GitHub 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 링크가 포함되어 있습니다.

주제

- [작업](#)
- [시나리오](#)

작업

AnalyzeDocument

다음 코드 예시에서는 AnalyzeDocument을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기를 참조하십시오 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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.

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [AnalyzeDocument](#).

DetectDocumentText

다음 코드 예시에서는 DetectDocumentText을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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.

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [DetectDocumentText](#).

GetDocumentAnalysis

다음 코드 예시에서는 GetDocumentAnalysis을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

 Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

```

"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_textthrottlingex.
MESSAGE 'The request processing exceeded the limit.' TYPE 'E'.
ENDTRY.

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [GetDocumentAnalysis](#).

StartDocumentAnalysis

다음 코드 예시에서는 StartDocumentAnalysis을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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_texlimitexceeddex.
    MESSAGE 'An Amazon Textract service limit was exceeded.' TYPE 'E'.
    CATCH /aws1/cx_texprovthruputexcdex.
    MESSAGE 'Provisioned throughput exceeded limit.' TYPE 'E'.
    CATCH /aws1/cx_texthrottlingex.
    MESSAGE 'The request processing exceeded the limit.' TYPE 'E'.
    CATCH /aws1/cx_texunsupporteddocex.
    MESSAGE 'The document is not supported.' TYPE 'E'.
ENDTRY.

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [StartDocumentAnalysis](#).

StartDocumentTextDetection

다음 코드 예시에서는 StartDocumentTextDetection을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

 Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

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

"Create an ABAP object for the Amazon S3 object."
DATA(lo_s3object) = NEW /aws1/cl_texs3object( iv_bucket = iv_s3bucket
      iv_name      = iv_s3object ).
"Create an ABAP object for the document."
DATA(lo_documentlocation) = NEW /aws1/cl_texdocumentlocation( io_s3object =
lo_s3object ).
"Start document analysis."
TRY.
      oo_result = lo_tex->startdocumenttextdetection( io_documentlocation =
lo_documentlocation ).
      DATA(lv_jobid) = oo_result->get_jobid( ).           "oo_result is returned
for testing purposes."
      MESSAGE 'Document analysis started.' TYPE 'I'.
      CATCH /aws1/cx_texaccessdeniedex.
      MESSAGE 'You do not have permission to perform this action.' TYPE 'E'.
      CATCH /aws1/cx_texbaddocumentex.
      MESSAGE 'Amazon Textract is not able to read the document.' TYPE 'E'.
      CATCH /aws1/cx_texdocumenttoolargeex.
      MESSAGE 'The document is too large.' TYPE 'E'.
      CATCH /aws1/cx_texidempotentprmmis00.
      MESSAGE 'Idempotent parameter mismatch exception.' TYPE 'E'.
      CATCH /aws1/cx_texinternalservererr.
      MESSAGE 'Internal server error.' TYPE 'E'.
      CATCH /aws1/cx_texinvalidkmskeyex.
      MESSAGE 'AWS KMS key is not valid.' TYPE 'E'.
      CATCH /aws1/cx_texinvalidparameterex.
      MESSAGE 'Request has non-valid parameters.' TYPE 'E'.
      CATCH /aws1/cx_texinvalids3objectex.
      MESSAGE 'Amazon S3 object is not valid.' TYPE 'E'.
```

```

CATCH /aws1/cx_texlimitexceeddex.
  MESSAGE 'An Amazon Textract service limit was exceeded.' TYPE 'E'.
CATCH /aws1/cx_texprovthruputexcdex.
  MESSAGE 'Provisioned throughput exceeded limit.' TYPE 'E'.
CATCH /aws1/cx_texthrottlingex.
  MESSAGE 'The request processing exceeded the limit.' TYPE 'E'.
CATCH /aws1/cx_texunsupporteddocex.
  MESSAGE 'The document is not supported.' TYPE 'E'.
ENDTRY.

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [StartDocumentTextDetection](#).

시나리오

문서 분석 시작

다음 코드 예제에서는 다음과 같은 작업을 수행하는 방법을 보여줍니다.

- 비동기 분석을 시작합니다.
- 문서 분석을 가져옵니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 Translate 예

다음 코드 예제는 Amazon Translate와 함께 SAP ABAP용 AWS SDK를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 호출하는 방법을 보여 주며 관련 시나리오와 교차 서비스 예시에서 컨텍스트에 맞는 작업을 볼 수 있습니다.

시나리오는 동일한 서비스 내에서 여러 함수를 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예시입니다.

각 예제에는 상황에 GitHub 맞게 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 링크가 포함되어 있습니다.

주제

- [작업](#)

- [시나리오](#)

작업

DescribeTextTranslationJob

다음 코드 예시에서는 DescribeTextTranslationJob을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기를 참조하십시오 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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."
    EXPORTING
        iv_jobid      = iv_jobid
    ).
    MESSAGE 'Job description retrieved.' TYPE 'I'.
    CATCH /aws1/cx_xl8internalserverex .
    MESSAGE 'An internal server error occurred. Retry your request.' TYPE 'E'.
    CATCH /aws1/cx_xl8resourcenotfoundex .
    MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
    CATCH /aws1/cx_xl8toomanyrequestsex.
    MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
ENDTRY.
```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [DescribeTextTranslationJob](#).

ListTextTranslationJobs

다음 코드 예시에서는 ListTextTranslationJobs을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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_xl8textxl8tionjobfilt.

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

TRY.
  oo_result = lo_xl8->listtexttranslationjobs(      "oo_result is returned for
testing purposes."
  EXPORTING
    io_filter      = lo_filter
  ).
  MESSAGE 'Jobs retrieved.' TYPE 'I'.
CATCH /aws1/cx_xl8internalserverex .
  MESSAGE 'An internal server error occurred. Retry your request.' TYPE 'E'.
CATCH /aws1/cx_xl8invalidfilterex .
  MESSAGE 'The filter specified for the operation is not valid. Specify a
different filter.' TYPE 'E'.
CATCH /aws1/cx_xl8invalidrequestex .
  MESSAGE 'The request that you made is not valid.' TYPE 'E'.
CATCH /aws1/cx_xl8toomanyrequestsex.
  MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
ENDTRY.
```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [ListTextTranslationJobs](#).

StartTextTranslationJob

다음 코드 예시에서는 StartTextTranslationJob을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

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

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

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

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

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

TRY.
  oo_result = lo_xl8->starttexttranslationjob(      "oo_result is returned for
testing purposes."
  EXPORTING
```

```

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

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [StartTextTranslationJob](#).

StopTextTranslationJob

다음 코드 예시에서는 StopTextTranslationJob을 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

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

```
TRY.
```

```

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

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [StopTextTranslationJob](#).

TranslateText

다음 코드 예시에서는 TranslateText를 사용하는 방법을 보여 줍니다.

SDK for SAP ABAP

Note

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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."
    EXPORTING
        iv_text      = iv_text
        iv_sourcelanguagecode = iv_sourcelanguagecode
        iv_targetlanguagecode = iv_targetlanguagecode
    ).
    MESSAGE 'Translation completed.' TYPE 'I'.
    CATCH /aws1/cx_xl8detectedlanguage00 .

```

```

MESSAGE 'The confidence that Amazon Comprehend accurately detected the
source language is low.' TYPE 'E'.
CATCH /aws1/cx_xl8internalserverex .
MESSAGE 'An internal server error occurred.' TYPE 'E'.
CATCH /aws1/cx_xl8invalidrequestex .
MESSAGE 'The request that you made is not valid.' TYPE 'E'.
CATCH /aws1/cx_xl8resourcenotfoundex .
MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
CATCH /aws1/cx_xl8serviceunavailex .
MESSAGE 'The Amazon Translate service is temporarily unavailable.' TYPE 'E'.
CATCH /aws1/cx_xl8textsizefmtexcdex .
MESSAGE 'The size of the text you submitted exceeds the size limit. ' TYPE
'E'.
CATCH /aws1/cx_xl8toomanyrequestsex .
MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
CATCH /aws1/cx_xl8unsuppdedlanguage00 .
MESSAGE 'Amazon Translate does not support translation from the language of
the source text into the requested target language. ' TYPE 'E'.
ENDTRY.

```

- API 세부 정보는 SAP용AWS SDK ABAP API 참조를 참조하십시오 [TranslateText](#).

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

자세한 내용은 여기에서 확인할 수 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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."
CREATE OBJECT lo_inputdataconfig
  EXPORTING
    iv_s3uri          = iv_input_data_s3uri
    iv_contenttype = iv_input_data_contenttype.

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

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

TRY.
  DATA(lo_translationjob_result) = lo_xl8->starttexttranslationjob(
    EXPORTING
      io_inputdataconfig = lo_inputdataconfig
      io_outputdataconfig = lo_outputdataconfig
      it_targetlanguagecodes = lt_targetlanguagecodes
      iv_dataaccessrolelearn = iv_dataaccessrolelearn
      iv_jobname = iv_jobname
      iv_sourcelanguagecode = iv_sourcelanguagecode
    ).
  MESSAGE 'Translation job started.' TYPE 'I'.
CATCH /aws1/cx_xl8internalserverex .
  MESSAGE 'An internal server error occurred. Retry your request.' TYPE 'E'.
CATCH /aws1/cx_xl8invparamvalueex .
  MESSAGE 'The value of the parameter is not valid.' TYPE 'E'.
CATCH /aws1/cx_xl8invalidrequestex.
  MESSAGE 'The request that you made is not valid.' TYPE 'E'.
CATCH /aws1/cx_xl8resourcenotfoundex .
  MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.

```

```

    CATCH /aws1/cx_xl8toomanyrequestsex.
      MESSAGE 'You have made too many requests within a short period of time. '
TYPE 'E'.
    CATCH /aws1/cx_xl8unsuppdedlanguage00 .
      MESSAGE 'Amazon Translate does not support translation from the language of
the source text into the requested target language.' TYPE 'E'.
    ENDTRY.

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

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

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

```

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

- [DescribeTextTranslationJob](#)
- [StartTextTranslationJob](#)

보안 내부 AWS SDK for SAP ABAP

클라우드 AWS 보안이 최우선 과제입니다. AWS 고객은 가장 보안에 민감한 조직의 요구 사항을 충족 하도록 구축된 데이터 센터 및 네트워크 아키텍처의 혜택을 누릴 수 있습니다.

보안은 기업과 귀사 간의 공동 책임입니다. AWS [공동 책임 모델](#)은 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드 보안 — AWS AWS 서비스 클라우드에서 실행되는 인프라를 보호하는 역할을 합니다 AWS 클라우드. AWS 또한 안전하게 사용할 수 있는 서비스를 제공합니다. 적용되는 규정 준수 프로그램에 대한 자세한 내용은 규정 준수 프로그램의 [범위AWS 서비스 내 규정 준수 프로그램의AWS 서비스](#) 참조하십시오. AWS SDK for SAP ABAP
- 클라우드에서의 보안 — AWS 서비스 사용하는 항목에 따라 책임이 결정됩니다. 또한 귀하는 귀사의 데이터의 민감도, 귀사의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 섹션은 다음 주제를 포함합니다.

주제

- [SAP 시스템 인증 커짐 AWS](#)
- [IAM 보안 모범 사례](#)
- [SAP 권한 부여](#)
- [보안 운영](#)
- [IAM Roles Anywhere로 인증서 사용](#)
- [SAP 자격 증명 저장소 사용](#)

SAP 시스템 인증 커짐 AWS

SAP 시스템이 SAP 사용자를 AWS 대신하여 전화를 걸려면 먼저 SAP 시스템이 자체 인증을 받아야 합니다. AWS AWS SDK for SAP ABAP 의 SDK 프로필 설정에서 선택한 다음 세 가지 인증 방법을 지원합니다. IMG

AWS SAP ABAP용 SDK - BTP 에디션은 SAP 자격 증명 저장소를 사용하는 [the section called “비밀 액세스 키 인증”](#) 방법으로만 인증할 수 있습니다.

주제

- [Amazon EC2 인스턴스 메타데이터 인증](#)
- [비밀 액세스 키 인증](#)
- [IAM Roles Anywhere를 사용한 인증서 기반 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 사용](#)을 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 사용하여 로그인합니다. AWS 자세한 내용은 [IAM 사용자의 액세스 키 관리](#)를 참조하세요.

기본 관리자는 AWS IAM 관리자로부터 액세스 키 ID와 비밀 액세스 키를 받습니다. 액세스 키 ID와 보안 액세스 키를 저장하도록 SAP 시스템을 구성해야 합니다.

- 보안, 저장 및 전달 (SSF)
 - SSF 기능을 사용하여 SAP AWS ABAP용 SDK를 인증하십시오. 자세한 내용은 [디지털 서명 및 암호화](#)를 참조하세요.
 - SSF02 보고서를 사용하여 SSF의 envelope 및 develop 기능을 테스트할 수도 있습니다. 자세한 내용은 [SSF 설치 테스트](#)를 참조하세요.
 - SDK for SAP ABAP SSF를 구성하는 단계는 /AWS1/IMG 트랜잭션에 설명되어 있습니다. 기술 사전 조건으로 이동한 다음 온프레미스 시스템의 추가 설정을 선택합니다.

- SAP 크리덴셜 스토어
 - SAP 크리덴셜 스토어를 사용하여 SAP ABAP용 AWS SDK - BTP 에디션을 인증하세요. 자세한 내용은 SAP 자격 증명 [저장소란?](#) 을 참조하십시오.
 - 구성 단계는 [SAP 자격 증명 저장소 사용을](#) 참조하십시오.

IAM Roles Anywhere를 사용한 인증서 기반 인증

인증 기관 (CA) 에서 발급한 X.509 인증서를 롤즈 애니웨어의 인증에 사용할 수 있습니다. AWS Identity and Access Management STRUST에서 인증서를 구성해야 합니다. IAM Roles Anywhere에 CA를 트러스트 앵커로 등록하고 프로필을 생성하여 IAM Roles Anywhere에 적용될 역할/정책을 지정해야 합니다. 자세한 내용은 Roles [Anywhere에서 AWS Identity and Access Management 신뢰 앵커 및 프로필 만들기를](#) 참조하십시오.

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.

다음 단계

에서 AWS SAP 시스템을 인증한 후 SDK for SAP ABAP는 자동으로 SAP 사용자의 비즈니스 기능에 적합한 IAM 역할을 말합니다. `sts:assumeRole`

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 수입할 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](#)

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.

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

위의 예를 통해 SAP 시스템은 CFO, AUDITOR 또는 REPORTING 사용자의 IAM 역할을 맡을 수 있습니다. AWS SDK는 SAP에서 사용자의 PFCG 역할을 기반으로 사용자에게 적합한 IAM 역할을 선택합니다.

Amazon EC2 인스턴스 프로파일은 다른 기능에도 사용할 수 있습니다.

- [AWSSAP 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 간접 호출, AWS Lambda 함수 실행 등의 사용자 작업을 수행할 수 있는 권한이 필요합니다. 모든 AWS SDK에 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 지정](#)을 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.

```
{
  "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.

```
{
  "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 인증](#)을 참조하세요.

여기서 SAP 사용자가 맡는 모든 IAM 역할은 SAP 사용자를 신뢰하는 신뢰 관계를 가져야 합니다.

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

SAP 권한 부여

SDK를 구성하는 데 필요한 승인은 SDK 에디션에 따라 다릅니다.

주제

- [구성을 위한 권한 부여](#)
- [최종 사용자를 위한 SAP 권한 부여](#)

구성을 위한 권한 부여

자세한 내용은 다음 탭을 참조하십시오.

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 프로필
- /AWS1/RES- AWS SDK for SAP ABAP v1 - 논리적 리소스
- /AWS1/TRC- AWS SDK for SAP ABAP v1 - 트레이스

SDK for SAP ABAP - BTP edition

다음 단계를 사용하여 SAP ABAP용 SDK - BTP 에디션이 구성에 액세스할 수 있도록 허용하십시오.

1. 비즈니스 역할 템플릿을 사용하여 새 비즈니스 역할을 생성합니다. SAP_BR_BPC_EXPERT 이 템플릿은 사용자 지정 비즈니스 구성 애플리케이션에 대한 액세스를 제공합니다.
2. 일반 역할 세부 정보에서 액세스 범주로 이동한 다음 읽기, 쓰기, 값 도움말에서 무제한을 선택합니다.
3. 비즈니스 카탈로그 탭으로 이동하여 비즈니스 카탈로그를 할당하여 SDK 구성에 대한 액세스를 제공하십시오. /AWS1/RTBTP_BCAT
4. 비즈니스 사용자 탭으로 이동하여 SDK 구성에 대한 액세스 권한을 부여할 비즈니스 사용자를 할당하세요.

최종 사용자를 위한 SAP 권한 부여

전제 조건: SDK 프로필 정의

SAP 보안 관리자가 역할을 정의하기 전에 비즈니스 분석가는 SAP ABAP용 SDK 또는 SAP /AWS1/IMG ABAP용 AWS SDK - BTP 에디션용 맞춤형 비즈니스 구성 애플리케이션을 위한 트랜잭션에서 SDK 프로필을 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 역할을 정의하십시오.

Note

SAP BTP, ABAP 환경에서는 PFGC 역할을 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 받을 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. 사용자에게 두 개 이상의 IAM 역할 권한이 부여된 경우 (예: CFO가 AUDITOR 논리적 IAM 역할 모두에 CFO 대해 권한을 부여받은 경우) AWS SDK는 우선 순위가 더 높은 (낮은 시퀀스 번호) 역할이 적용되도록 함으로써 관계를 깨뜨립니다.

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 SDK 인증에 사용됩니다. SAP의 SSF 또는 자격 증명 저장소 기능을 사용하여 암호화됩니다.

전송 중 데이터 암호화

모든 통화는 AWS 서비스 HTTPS로 암호화됩니다. SAP ICM은 HTTPS 연결을 관리합니다. AWS 인증서는 STRUST에서 신뢰할 수 있어야 합니다.

API 사용

ABAP 사용자가 sts:assumeRole을 사용하여 역할을 맡으면 세션 이름에는 USERID-SID-MANDT와 같은 제목이 붙습니다.

- USERID은(는) SY-UNAME 변수의 ABAP 사용자입니다.
- SID은(는) SY-SYSID 변수의 ABAP 시스템 ID입니다.
- MANDT은(는) SY-MANDT 변수의 ABAP 클라이언트입니다.

세션 이름은 사용자 CloudTrail 이름으로 표시됩니다. 이렇게 하면 ABAP 사용자의 API 호출을, 호출을 시작한 시스템과 클라이언트 및 사용자까지 추적할 수 있습니다. 자세한 내용은 [AWS CloudTrail란 무엇입니까?](#)를 참조하십시오.

IAM Roles Anywhere로 인증서 사용

Roles Anywhere와 함께 인증 기반 AWS 인증을 사용하여 SAP 시스템을 인증할 수 있습니다. AWS Identity and Access Management STRUST에서 인증서를 설정하고 /AWS1/IMG에서 SDK 프로필을 구성해야 합니다.

필수 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 Roles [Anywhere에서 신뢰 앵커 및 프로필 만들기를](#) 참조하십시오. AWS Identity and Access Management
- SAP 사용자의 IAM 역할은 IAM 관리자가 생성해야 합니다. 역할에는 필요한 것을 호출할 수 있는 권한이 있어야 AWS 서비스입니다. 자세한 내용은 [IAM 보안 모범 사례](#)를 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

- [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.

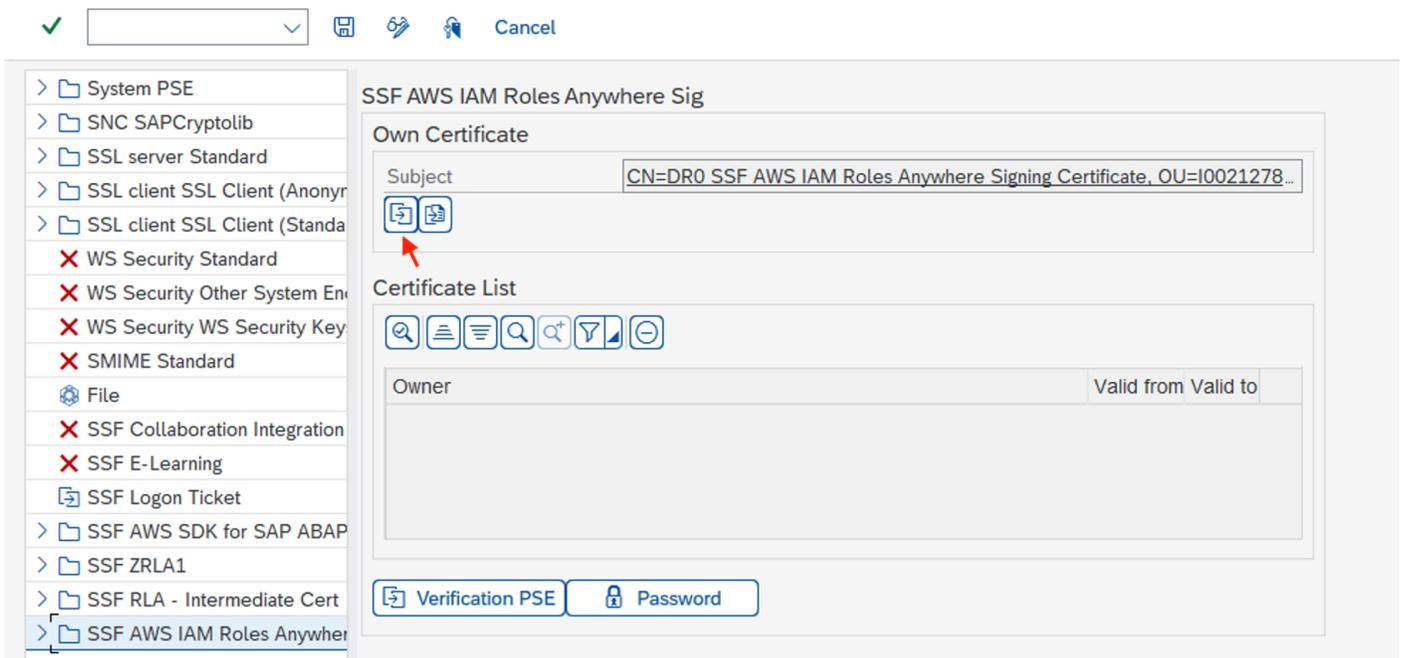
1. 트랜잭션 코드 SE16을 실행하여 SSF 애플리케이션을 정의합니다.
2. SSFAPPLIC 테이블 이름을 입력하고 새 항목을 선택합니다.
3. APPLIC 필드에 SSF 애플리케이션의 이름을 입력하고 DESCRIPT 필드에 설명을 입력한 다음 나머지 필드에서 Selected (X) 옵션을 선택합니다.

2단계 - SSF 파라미터 설정

1. 를 /n/AWS1/IMG 실행하여 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 [사설 인증 기관을 통한 AWS 사설 최종 개체 인증서 발급](#)을 참조하십시오.

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 > 기술적 사전 조건 > 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 “필수 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.

SAP 자격 증명 저장소 사용

SAP 자격 증명 저장소는 SAP 비즈니스 기술 플랫폼에서 비밀 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 AWS SDK for SAP ABAP](#) 참조하십시오.

구성을 시작하기 전에 사전 요구 사항을 충족하는지 확인하십시오. 자세한 내용은 [SAP](#) 자격 증명 저장소를 참조하십시오.

주제

- [구성 단계](#)
- [SDK와 함께 SAP 자격 증명 저장소 사용](#)

구성 단계

단계

- [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

인증을 위한 자격 증명 저장소 설정을 구성하려면 다음 단계를 사용하십시오.

1. SAP 자격 증명 저장소 인스턴스의 설정 탭으로 이동합니다.
2. 구성 편집을 선택합니다.
 - 상호 TLS를 기본 인증 유형으로 선택합니다.
 - 페이로드 암호화 상태로 [사용 안 함] 을 선택합니다. 페이로드는 전송 중에 HTTPS로 암호화됩니다. 하지만 현재 페이로드는 이중 암호화할 수 없습니다.
3. 저장을 선택합니다.

2단계: 서비스 키 생성

다음 단계를 사용하여 자격 증명 저장소의 서비스 키를 생성합니다.

1. SAP 자격 증명 저장소 애플리케이션의 왼쪽 창에서 서비스 키로 이동합니다.
2. 서비스 키 생성을 선택합니다.
3. 서비스 키의 이름을 입력하고 [Create] 를 선택합니다.

서비스 키는 선택한 인증 유형을 기반으로 생성됩니다. 서비스 키를 다운로드하고 나중에 사용할 수 있도록 안전하게 보관하세요.

3단계: 서비스 키를 .p12 형식으로 변환

통신 시스템용 아웃바운드 사용자를 생성하려면 이 .p12 형식의 클라이언트 인증서가 필요합니다. 다음 단계를 사용하여 자격 증명 저장소 서비스 .p12 키에 제공된 인증서 세부 정보에서 인증서를 생성할 수 있습니다.

1. SAP [보안 센터 서비스에서 SAP 클라우드 루트 CA 인증서 \(SAP에 필요\) 를 다운로드하세요.](#)
2. SAP 클라우드 루트 CA 인증서를 원하는 텍스트 파일 형식으로 엽니다. 파일 끝에서 Enter를 누르고 서비스 키의 인증서 필드에 있는 인증서를 복사하여 붙여넣습니다. 새 줄 문자를 실제 새 줄 (Enter) \n 으로 바꾸고 전체 인증서를 .cer 파일 형식으로 저장합니다.

3. 서비스 키의 키 필드에서 키를 복사합니다. 이 개인 키는 민감한 데이터로 취급되어야 합니다. 텍스트 파일에 붙여넣고 새 줄 문자를 실제 새 \n 줄로 바꿉니다 (Enter). 개인 키를 텍스트 파일에 저장합니다.
4. 이전 단계에서 생성한 인증서와 개인 키를 사용하여 다음 명령을 실행하여 .p12 인증서를 생성합니다.

```
openssl pkcs12 -export -out <.p12_<filename>> -inkey <private_key.key> -in
<certificate.cer>
```

명령을 실행하려면 내보내기 암호 확인이 필요했습니다. 나중에 사용할 수 있도록 암호를 보관해 두십시오.

개인 키에 저장된 .key 텍스트 파일을 삭제합니다.

4단계: SAP BTP, ABAP 환경에 연결

SAP 자격 증명 저장소와 연결하도록 SAP BTP, ABAP 환경을 구성합니다.

주제

- [통신 시스템](#)
- [커뮤니케이션 준비](#)

통신 시스템

다음 단계를 사용하여 SAP BTP, ABAP 환경에서 SAP 자격 증명 저장소로의 통신을 가능하게 하는 통신 시스템을 생성하십시오.

1. ABAP 환경 시스템의 Fiori 런치패드를 엽니다.
2. 통신 시스템 타일을 선택하여 애플리케이션을 엽니다.
3. New(새 파일)를 선택합니다.
4. 통신 시스템의 이름과 ID를 입력하고 [Create] 를 선택합니다. 예를 들어, 시스템 이름을 지정할 수 ZSAP_CREDSTORE 있습니다.
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/credentials 인 경우 호스트 이름은 입니다credstore.mesh.cf.us10.hana.ondemand.com.

- 아웃바운드 통신 사용자: 새 사용자를 + 추가하려면 선택합니다.
 - a. 인증 메커니즘으로 SSL 클라이언트 인증서를 선택합니다.
 - b. 새 인증서 업로드를 선택합니다.
 - 이전 단계에서 생성한 .p12 인증서를 찾아보십시오.
 - 설명을 입력합니다.
 - .p12인증서 생성에 사용된 내보내기 암호를 입력합니다.
 - 업로드를 선택합니다.
 - c. 생성을 선택하여 아웃바운드 사용자를 생성합니다.
- 6. 저장을 선택합니다.
- 7. 이전 단계에서 다운로드한 서비스 키를 삭제합니다.

커뮤니케이션 준비

다음 단계를 사용하여 아웃바운드 통신을 위한 통신 시나리오를 제공하는 통신 배열을 생성하십시오.

1. ABAP 환경 시스템의 Fiori 런치패드를 엽니다.
2. 통신 배열 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_CREDSTORE입니다.
5. 생성을 선택합니다.
6. 통신 시스템 필드에서 이전 단계에서 생성한 통신 시스템을 찾아보십시오. 기타 정보는 시스템 선택 후 자동으로 입력됩니다.
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 저장소 서비스를 사용할 수 있습니다.

SDK와 함께 SAP 자격 증명 저장소 사용

단계

- [1단계: 네임스페이스 및 자격 증명 생성](#)

• [2단계: 사용자 지정 비즈니스 구성 애플리케이션 구성](#)

1단계: 네임스페이스 및 자격 증명 생성

SAP 도움말 — [자격 증명 생성, 편집 및 삭제를 사용하여 SAP 자격 증명 저장소에서 네임스페이스와 자격 증명을 생성합니다.](#)

다음 세부 정보를 입력하여 키 유형의 자격 증명을 생성합니다.

- 네임스페이스 - 네임스페이스의 이름을 입력하고 관련 자격 증명을 함께 그룹화합니다.
- 이름 - 키 이름을 입력합니다. aws-0123456789012-username 권장되는 위치는 다음과 같습니다.
 - 0123456789012 자격 증명이 액세스 권한을 부여하는 AWS 계정 ID입니다.
 - username 자격 증명에 속한 IAM 사용자 이름입니다.
- 값 — base-64로 인코딩된 보안 액세스 키를 입력합니다. 다음 명령을 사용하여 비밀 액세스 키를 base-64로 인코딩합니다.

```
xargs echo -n | base64 # just press enter, do not enter arguments on the command line
MySecretAccessKey
Ctrl-D
```

이 명령은 표준 입력에서 보안 액세스 키를 읽고 뒤에 줄 바꿈이 없는 base64로 전달합니다.

base-64로 인코딩된 비밀 액세스 키를 화면에 출력합니다. 값을 SAP 자격 증명 저장소에 복사한 후 터미널을 지우거나 닫습니다.

- 사용자 이름 — 액세스 키 ID를 입력합니다.
- 생성을 선택합니다.

자격 증명이 하나인 새 네임스페이스가 생성되고 이 네임스페이스 내에서 자격 증명을 추가, 삭제 또는 수정할 수 있습니다.

최소 권한 원칙에 따라 네임스페이스에 저장된 자격 증명에 대한 액세스를 관리하세요.

2단계: 사용자 지정 비즈니스 구성 애플리케이션 구성

다음 단계를 사용하여 SDK의 인증에 사용할 자격 증명을 정의하도록 사용자 지정 비즈니스 구성 애플리케이션을 구성하십시오.

1. ABAP 환경 시스템의 Fiori 런치패드를 엽니다.
2. 사용자 지정 비즈니스 구성 타일을 찾아 애플리케이션을 엽니다.
3. SDK 프로필 비즈니스 구성을 엽니다.
4. SAP 자격 증명 저장소에 대한 인증 설정을 구성해야 하는 SDK 프로필을 선택합니다.
5. 선택한 프로필의 인증 및 설정 탭에서 편집을 선택하고 다음 세부 정보를 입력합니다.
 - 인증 방법 - SAP 자격 증명 저장소에서 자격 증명을 선택합니다.
 - 네임스페이스 - SAP 자격 증명 저장소에서 만든 네임스페이스를 입력합니다. 자세한 정보는 [the section called “1단계: 네임스페이스 및 자격 증명 생성”](#)을 참조하세요.
 - 키 이름 - 생성된 서비스 키의 이름을 입력합니다. 자세한 정보는 [the section called “2단계: 서비스 키 생성”](#)을 참조하세요.
 - 통신 배열 - 생성된 통신 배열의 이름을 입력합니다. 자세한 정보는 [the section called “커뮤니케이션 준비”](#)을 참조하세요.
6. 적용을 선택하여 AWS SDK 프로필 화면으로 이동합니다.
7. 전송 선택을 선택하여 값 도움말을 사용하여 전송을 선택합니다.
8. 저장을 선택합니다.

AWS SDK for SAP ABAP 문제 해결

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 0002619546](#)이 시스템에 적용되었는지 확인합니다.

지정되지 않은 위치 제약

문제 — 이 요청이 전송된 region 특정 엔드포인트에서 지정되지 않은 위치 제약이 호환되지 않음

원인 — Amazon S3 버킷의 io_createbucketconfiguration 파라미터에 AWS 리전이 없습니다.

해결 방법 — us-east-1을(를) 제외한 모든 지역에서 버킷을 생성할 때는 createbucket()에서 io_createbucketconfiguration 파라미터를 사용하여 Amazon S3 버킷의 리전을 지정합니다. us-east-1에 대한 제약 조건을 지정할 필요가 없습니다.

다음 예는 올바르게 구성된 `io_createbucketconfiguration` 파라미터를 보여줍니다.

```
createbucket(
  iv_bucket = 'test-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_enabled`을(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
 - AWS에서 실행 중인 SAP 시스템의 메타데이터를 통해 인스턴스 역할을 선택합니다.
 - 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
- 수입된 IAM 역할과 수입하는 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 해결](#)을 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_arn은(는) 지정된 action을(를) 수행하는 데 필요한 resource_arn 권한을 설정해야 합니다. 자세한 내용은 [IAM 리소스에 액세스하는 데 필요한 권한](#)을 참조하세요.

활성 시나리오

문제 — 활성 시나리오가 구성되어 있지 않음

원인 — 활성 시나리오 설정이 누락되었습니다.

해결 방법 - 활성 시나리오를 구성하려면 [런타임 설정](#)을 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 HTTPS 포트 443에서 AWS 엔드포인트에 직접 또는 프록시 서버를 통해 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 연결](#)을 참조하세요.

추가 주제

이 섹션은 다음 주제를 포함합니다.

주제

- [AWS SDK for SAP ABAP 보도 자료](#)
- [SAP 라이선싱](#)

AWS SDK for SAP ABAP 보도 자료

AWS SAP ABAP용 SDK는 전송 형식으로 제공되며 SAP ABAP용 AWS SDK - BTP 에디션은 애드온으로 제공됩니다. 트랜스포트와 애드온을 가져오는 메커니즘은 다르지만 기술적 기능은 동일합니다. 자세한 내용은 [설정](#)을 참조하십시오.

주제

- [릴리스 전략](#)
- [모범 사례](#)
- [SDK for SAP ABAP 패치](#)
- [추가 모듈 설치](#)
- [SDK for SAP ABAP 제거](#)

릴리스 전략

버전 1은 자주 AWS SDK for SAP ABAP 업데이트됩니다. 새 패치는 의 릴리스 및 업데이트에 따라 매주 또는 매일 AWS 서비스릴리스됩니다. 의 패치에는 SDK의 패치 수준을 업데이트하는 버그 수정 및 기타 변경 사항이 포함될 AWS 서비스 수 있습니다. 자세한 내용은 [AWS SDK 및 도구 유지 관리 정책](#)을 참조하십시오.

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 해당 모듈을 더 이상 사용하지 않더라도 core 모듈과 설치한 기타 모든 모듈의 패치를 진행해야 합니다. 예를 들어, SDK를 설치할 때 core, ec2, lmd 전송을 가져온 경우, 패치를 적용할 때 core, ec2, lmd에 대한 최신 전송을 가져와야 합니다.

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

SAP ABAP용 SDK를 제거하려면 <https://aws.amazon.com/V1/release/uninstall-abapsdk-LATEST.zip>에서 삭제 전송 키트를 다운로드해야 합니다. `sdk-for-sapabap awsSdkSapabap`

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

<https://aws.amazon.com/v1/release/uninstall-ABAPSDK-Latest.sig>에서 서명 파일을 다운로드할 수 있습니다. `sdk-for-sapabap awsSdkSapabap` 파일을 검증하려면 [SDK for SAP ABAP 확인](#)을 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 유효하지 않은 권한 부여를 받게 됩니다. SDK를 제거하기 전에 PFCG 역할에서 SDK 권한 부여 참조를 제거하세요.

Note

AWS 개발자 프리뷰 중에는 SAP ABAP용 SDK - 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: AWS 서비스을(를) 사용하면 자체 ABAP 코드로 AWS SDK for SAP ABAP을(를) 사용할 수 있습니다. SAP 시스템과 AWS 서비스 간의 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 표에는 의 설명서 릴리스가 설명되어 AWS SDK for SAP ABAP 있습니다.

변경 사항	설명	날짜
새 콘텐츠	SAP ABAP용 SDK의 개발자 프리뷰 - BTP 에디션	2024년 5월 31일
새 콘텐츠	IAM Roles Anywhere로 인증서 사용 이 추가되었습니다.	2023년 12월 1일
새 콘텐츠	SDK로 제품 구축 이 추가되었습니다.	2023년 12월 1일
새 콘텐츠	동작 재시도 가 추가되었습니다.	2023년 12월 1일
새 콘텐츠	SAP 라이선스 가 추가되었습니다.	2023년 9월 22일
공개 릴리스	개발자 가이드 최초 재출시. AWS SDK for SAP ABAP	2023년 6월 30일
새 콘텐츠	AWS SDK for SAP ABAP 기능 이 추가되었습니다.	2023년 5월 30일
새 콘텐츠	AWS SDK for SAP ABAP문제 해결 이 추가되었습니다.	2023년 2월 17일
개발자 미리 보기	개발자 가이드의 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.