開發人員指南

AWS 適用於 SAP ABAP 的 SDK

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

AWS 適用於 SAP ABAP 的 SDK: 開發人員指南

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

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能附屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

Table of Contents

什麼是 適用於 SAP ABAP 的 AWS SDK？ 1
的功能 適用於 SAP ABAP 的 AWS SDK 1
Maintenance (維護) 1
API 參考 1
定價 2
Resources 2
開始使用 3
步驟 1：準備 AWS 您的帳戶 3
SAP 使用者的 IAM 角色 3
身分驗證 4
步驟 2：安裝 SDK 6
步驟 3：設定 SDK 6
步驟 4：功能設定 7
步驟 5：授權 SAP 使用者 9
步驟 6：撰寫程式碼 11
步驟 7：執行應用程式 14
設定 16
SAP 先決條件 16
適用於 SAP ABAP 的開發套件 16
適用於 SAP ABAP 的 SDK - BTP 版本 19
安裝 適用於 SAP ABAP 的 AWS SDK 20
下載軟體開發套件 20
驗證 檔案 21
AWS SDK 傳輸 22
安裝 SDK - BTP 版本 25
安裝適用於 SAP ABAP 的 SDK - BTP 版本 25
模組 25
修補適用於 SAP ABAP 的 SDK - BTP 版本 26
設定 27
全域設定 28
技術設定 29
設定案例 29
應用程式組態 30
SDK 描述檔 30

邏輯資源解析程式 31
範例 31
執行期設定 32
日誌和追蹤 32
OPT-IN：增強型遙測 32
作用中案例 33
進階連線案例 33
透過代理伺服器的連線 33
透過封包檢查防火牆的連線 34
閘道端點 34
自訂界面端點 34
進階路由 35
存取多個區域中的端點 36
服務提供者設定 37
重新整理、追蹤和遙測 37
SAP 系統重新整理 37
追蹤 38
遙測 39
使用開發套件 40
資料表示 40
資料類型 40
AWS 資料類型 42
範例程式 43
先決條件 43
代碼 44
程式碼區段 44
概念 46
API 類別 47
其他物件 47
結構類別 47
陣列 49
地圖 50
更高層級的函數 50
功能 1
程式設計組態 51
等待程式 51

分頁程式 52
重試行為 53
預先簽章者 54
跨帳戶 IAM 角色鏈結 54
建置產品 55
設定產品 ID 55
自訂對 的 HTTP 請求 AWS 55
實作增強功能 55
篩選增強功能 56
編碼增強功能 56
限制 56
程式碼範例 58
ACM 60
動作 60
API Gateway 68
動作 60
應用程式復原控制器 76
動作 60
Aurora 81
動作 60
Auto Scaling 87
動作 60
Amazon Bedrock 執行時期 96
Anthropic Claude 97
Stable Diffusion 100
Amazon Bedrock 代理程式執行時期 102
動作 60
CloudFront 104
動作 60
CloudWatch 107
動作 60
案例 114
CloudWatch Logs 116
動作 60
Amazon Cognito 身分提供者 119
動作 60

Amazon Comprehend 124
動作 60
AWS Config 138
動作 60
AWS Control Tower 140
動作 60
Firehose 152
動作 60
DynamoDB 154
基本概念 154
動作 60
Amazon EC2 168
動作 60
Amazon ECR 190
動作 60
Amazon EMR 197
動作 60
EventBridge 排程器 203
動作 60
AWS Glue 208
動作 60
HealthImaging 221
動作 60
HealthLake 238
動作 60
IAM 252
動作 60
AWS IoT SiteWise 276
開始使用 277
動作 60
Amazon Keyspaces 285
動作 60
Kinesis 293
基本概念 154
動作 60
AWS KMS 303

開始使用 277
動作 60
Lambda 323
基本概念 154
動作 60
組織 336
動作 60
Amazon Pinpoint 341
動作 60
Amazon Pinpoint SMS 和語音 API 346
動作 60
Amazon Polly 355
動作 60
Amazon RDS 364
動作 60
Amazon Redshift 369
動作 60
Amazon Rekognition 377
動作 60
Amazon S3 391
基本概念 154
動作 60
案例 114
SageMaker AI 416
動作 60
案例 114
Secrets Manager 434
動作 60
Amazon SES 436
動作 60
Amazon SES API v2 453
動作 60
Amazon SNS 462
動作 60
案例 114
Amazon SQS 472

動作 60
案例 114
步驟函數 482
動作 60
Systems Manager 491
動作 60
Amazon Textract 500
動作 60
案例 114
Amazon Transcribe 511
動作 60
Amazon Translate 519
動作 60
案例 114
安全 528
系統身分驗證 528
中繼資料身分驗證 529
私密存取金鑰身分驗證 529
使用 IAM Roles Anywhere 的憑證型身分驗證 530
跨帳戶存取的來源描述檔身分驗證 530
下一步驟 531
IAM 安全性的最佳實務 531
Amazon EC2 執行個體描述檔的最佳實務 531
SAP 使用者的 IAM 角色 532
來源描述檔安全性考量 535
SAP 授權 535
組態的授權 536
最終使用者的 SAP 授權 537
安全操作 538
靜態資料加密 538
傳輸中資料的加密 538
API 用量 1
私密金鑰 (SSF) 538
先決條件 539
程序 539
IAM Roles Anywhere 541

先決條件 541
程序 541
來源設定檔 54
先決條件 545
程序 545
登入資料存放區 20
組態步驟 547
搭配 SDK 使用 SAP 登入資料存放區 549
疑難排解 553
匯入失敗 553
未指定的位置限制條件 553
SSL 錯誤 554
設定檔組態 554
IAM 授權 555
動作的授權 556
作用中案例 33
特殊字元 556
連線能力 557
其他主題 558
推出 558
發行策略 558
最佳實務 531
修補適用於 SAP ABAP 的 SDK 559
安裝其他模組 559
解除安裝適用於 SAP ABAP 的 SDK 560
SAP 授權 560
文件歷史紀錄 562
..... dlxiii

什麼是 適用於 SAP ABAP 的 AWS SDK？

適用於 SAP ABAP 的 AWS SDK 以 AWS ABAP 語言為 提供的服務提供界面。使用 SDK，您可以在上實作 ABAP BADIs、報告、交易、OData 服務和其他 ABAP 成品 AWS 服務，例如 Amazon Simple Storage Service (Amazon S3) Amazon DynamoDB、、 Amazon Translate等。您也可以針對 ABAP 型系統進行開發，從 SAP NetWeaver 7.4 開始，並在 SAP Business Technology Platform 環境中進行。如需詳細資訊，請參閱安裝適用於 SAP ABAP 的 AWS SDK - BTP 版本 。

主題

- 的功能 適用於 SAP ABAP 的 AWS SDK
- 開發套件主要版本的維護與支援
- API 參考
- 定價
- 其他資源

的功能 適用於 SAP ABAP 的 AWS SDK

適用於 SAP ABAP 的 AWS SDK 旨在讓 SAP 開發人員感到熟悉和自然。例如，雖然所有 都 AWS 服務 使用 `true`和 `false`字串來代表 XML 和 JSON 結構中的布林值資料，但適用於 SAP ABAP 的 SDK 會將這些資料轉換為 ABAP 原生`'X'`和`' '`單一字元值。適用於 SAP ABAP 的 SDK 會盡可能使用原生 ABAP 建構，包括資料類型和時間戳記格式。因此，ABAP 程式設計人員不需要擔心基礎 JSON 和 XML 序列化或 API 通訊協定的線路格式。

開發套件主要版本的維護與支援

如需開發套件主要版本及其基礎相依性之維護與支援的相關資訊，請參閱《AWS 開發套件及工具參考指南》中的以下內容：

- AWS SDKs和工具維護政策
- AWS SDKs、工具和版本支援矩陣

API 參考

若要查看 適用於 SAP ABAP 的 AWS SDK APIs的完整清單，請參閱 適用於 SAP ABAP 的 AWS SDK - API 參考指南。

若要查看 適用於 SAP ABAP 的 AWS SDK TLAs 模組清單，請參閱 適用於 SAP ABAP 的 AWS SDK - 模組清單。

若要查看適用於 SAP ABAP 的 SDK - BTP 版本開發人員預覽 TLAs的完整模組清單，請參閱AWS 適用於 SAP ABAP 的 SDK - BTP 版本 - 模組清單。

定價

適用於 SAP ABAP 的 AWS SDK 可供您免費使用。您只需為使用 SDK 使用 AWS 的資源和服務付費。

其他資源

除了本指南之外，下列線上資源也適用於適用於 SAP ABAP 的 SDK。

- AWS 文件上的 SAP
- AWS 開發人員部落格
- AWS 開發人員論壇
- AWS SDK 程式碼範例程式庫
- @awsdevelopers(Twitter)

入門 適用於 SAP ABAP 的 AWS SDK

本節說明如何開始使用 SDK。它包含安裝 SDK、執行基本組態，以及建立 Hello World 程式碼範例，將片語從一種語言轉譯為另一種語言的相關資訊。如果您是初次使用 AWS SDK，建議您在沙盒環境中執行這些步驟。

步驟

- 步驟 1：準備 AWS 您的帳戶
- 步驟 2：安裝 SDK
- 步驟 3：設定 SDK
- 步驟 4：功能設定
- 步驟 5：授權 SAP 使用者
- 步驟 6：撰寫程式碼
- 步驟 7：執行應用程式

步驟 1：準備 AWS 您的帳戶

若要開始使用適用於 SAP ABAP 的 SDK，您必須有作用中的 AWS 帳戶 。 AWS 帳戶 即使您的 SAP 系統是在內部部署、SAP Business Technology Platform (BTP) 上或其他雲端供應商託管，您也需要 。

如果您的 SAP 系統在 AWS 雲端上執行，則您將呼叫 AWS 中的 服務 AWS 帳戶。

主題

- SAP 使用者的 IAM 角色
- 身分驗證

SAP 使用者的 IAM 角色

- 使用 AWS Identity and Access Management 使用者指南中提供的指示建立 IAM 角色。如需詳細資訊，請參閱建立角色以將許可委派給 AWS 服務。請注意 IAM 角色的 Amazon Resource Name (ARN)，以供日後使用。
- 選取 Amazon EC2 做為使用案例。
- 使用 `SapDemoTranslate`做為角色的名稱。

- 將TranslateReadOnly設定檔連接至角色。
- 角色必須具有下列實體，才能讓 SAP 系統擔任該角色。將 *"111122223333"* 換成您的 AWS 帳戶號碼。

  JSON

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

  此範例顯示來自 AWS 帳戶 的任何委託人*"111122223333"*都可以擔任該角色。這是適用於proof-of-concept的廣泛許可。您可以使用較窄的主體進行生產，例如下列範例。

- 特定使用者 – 當 SAP 系統使用下列其中一項時：
  - 來自現場部署 SAP 系統的 SSF 加密憑證
  - SAP BTP、ABAP 環境上 SAP Credential Store 服務的登入資料
- 特定角色 – 當 SAP 系統位於 Amazon EC2 上且有執行個體描述檔時。
- Amazon EC2 – 當 SAP 系統位於 Amazon EC2 上且沒有執行個體描述檔時。

如需詳細資訊，請參閱 IAM 安全的最佳實務。

身分驗證

身分驗證取決於 SAP 系統的託管位置。

位置

- 在 AWS 雲端
- 內部部署、SAP BTP 或其他雲端

在 AWS 雲端

請確定 SAP 系統執行所在的 EC2 執行個體具有具有下列許可的執行個體描述檔。

JSON

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

新增您在上一個步驟中記下的 ARN。

此許可可讓您的 SAP 系統代表 ABAP 使用者擔任該`SapDemoTranslate`角色。

內部部署、SAP BTP 或其他雲端

如果您的 SAP 系統位於內部部署、SAP BTP 或其他雲端，請使用下列步驟建立使用私密存取金鑰進行身分驗證的連線。

1. 建立 IAM 使用者。如需詳細資訊，請參閱建立 IAM 使用者（主控台）。
2. 使用 `SapDemoSID`做為 IAM 使用者的名稱。 `SID`是 SAP 系統的系統 ID。
3. 將`SapDemoTranslate` 角色指派給此使用者。

保留 `access_key`和 `secret_access_key`。您必須在 SAP 系統中設定這些登入資料。

> **Note**
>
> 如果您的 SAP 系統位於內部部署、SAP BTP 或其他雲端，您可以使用下列其中一個選項進行身分驗證。
>
> - 使用 SSF 或 SAP 登入資料存放區的私密存取金鑰身分驗證

- 搭配 IAM Roles Anywhere 使用憑證

步驟 2：安裝 SDK

如需安裝說明，請參閱下列索引標籤。

SDK for SAP ABAP

在您的 SAP 系統中匯入適用於 SAP ABAP 傳輸的 SDK。您可以將傳輸匯入任何用戶端。如需詳細資訊，請參閱安裝適用於 SAP ABAP 的 SDK。

SDK for SAP ABAP - BTP edition

使用部署產品應用程式安裝適用於 SAP ABAP 的 SDK - BTP 版本。如需詳細資訊，請參閱安裝適用於 SAP ABAP 的 SDK - BTP 版本。

步驟 3：設定 SDK

設定 SDK 之前，請確定您擁有必要的授權。如需詳細資訊，請參閱 SAP 授權。

如需組態指示，請參閱下列索引標籤。

SDK for SAP ABAP

執行`/AWS1/IMG`交易以開啟適用於 SAP ABAP 的 SDK 實作指南。若要執行此交易，請在 SAP 系統的命令列`/n/AWS1/IMG` 中輸入 ，然後選擇 Enter。

完成下列組態。

- 前往技術先決條件。
  - 檢閱建議的參數和 HTTPS 連線。
- 前往全域設定 → 設定案例。
  - 根據全域設定中的建議變更設定。
- 前往全域設定 → 技術設定。
  - 根據全域設定中的建議變更設定。
- 前往執行期設定 → 日誌和追蹤。
  - 選取新項目。

    - 追蹤層級： 無追蹤。
    - 傾印行上限：100。
    - OPT-IN：enh 遙測：將此保留空白。
  - 選取儲存。
- 前往執行期設定 → 作用中案例。
  - 在新案例下，選取 DEFAULT。
  - 選取遞交案例變更。
  - 接受提示。

現場部署系統的先決條件

如果您的 SAP 系統正在內部部署或其他雲端中執行，則必須將登入資料存放在 SAP 資料庫中。登入資料會使用 SAP SSF 加密，並需要設定的密碼編譯程式庫，例如 SAP 的 CommonCryptoLib。

/AWS1/IMG 交易中說明設定 SSF for SDK for SAP ABAP 的步驟。

> **Note**
>
> 如果您的 SAP 系統在 Amazon EC2 上執行，則上述先決條件不適用。在 Amazon EC2 上執行的 SAP 系統會從 Amazon EC2 執行個體中繼資料擷取短期自動輪換憑證。

SDK for SAP ABAP - BTP edition

在 Web 瀏覽器中開啟您的 ABAP 環境，然後導覽至自訂業務組態應用程式。

完成下列組態。

- 前往設定案例。
  - 根據全域設定中的建議變更設定。
- 前往技術設定。
  - 根據全域設定中的建議變更設定。

步驟 4：功能設定

如需設定指示，請參閱下列索引標籤。

執行交易/AWS1/IMG (/n/AWS1/IMG在命令列中輸入 ，然後選擇 Enter) 以開啟 AWS SDK 的實作指南。

- 前往應用程式組態 → SDK 設定檔。
  - 選取新項目。
    - 設定檔：DEMO。
    - 描述： Demo profile。
    - 選取儲存。
- 反白顯示您建立的項目，然後按一下身分驗證和設定樹狀目錄分支。
  - 選取新項目。
    - SID：您目前所在的 SAP 系統 ID。
    - 用戶端：您目前所在的 SAP 系統的用戶端。
    - 案例 ID：下拉式清單，您可以在其中找到由 Basis 管理員建立的 DEFAULT 案例。
    - AWS 區域：輸入您要呼叫 AWS 的區域。如果您的 SAP 系統正在執行 AWS，請輸入正在執行 AWS 的區域。
    - 身分驗證方法：
      - 如果您的 SAP 系統在 Amazon EC2 上執行，請透過中繼資料選取執行個體角色。
      - 如果您的 SAP 系統正在內部部署或其他雲端中執行，請從 SSF Storage 選取登入資料。
        - 選取設定登入資料。
        - 輸入您在上一個步驟中建立的存取金鑰 ID 和私密存取金鑰。
  - 將停用 IAM 角色保留空白。
  - 選取儲存。
- 按一下 IAM 角色映射樹狀目錄分支。
  - 選取新項目。
    - 輸入序號：010。
    - 輸入邏輯 IAM 角色：TESTUSER。
    - 輸入 IAM 角色 ARN：輸入包含上一個步驟中建立之TranslateRead0nly政策的 IAM 角色 arn：aws：。

SDK for SAP ABAP - BTP edition

使用 SAP Credential Store 設定身分驗證。如需詳細資訊，請參閱使用 SAP 登入資料存放區。

在 Web 瀏覽器中開啟您的 ABAP 環境，然後導覽至自訂業務組態應用程式。

- 前往 SDK 設定檔。
  - 選取編輯以建立新的設定檔。
    - 設定檔：DEMO。
    - 描述： Demo profile。
- 選取所建立項目旁的向右箭頭鍵，以導覽至身分驗證和設定索引標籤。

  選取新增項目。
  - SID：您目前所在的 SAP 系統 ID。
  - 用戶端：您目前所在的 SAP 系統的用戶端。
  - 案例 ID：下拉式清單，您可以在其中找到由 Basis 管理員建立的 DEFAULT 案例。
  - AWS 區域：輸入您要呼叫 AWS 的區域。如果您的 SAP 系統正在其中執行 AWS，請輸入正在其中執行 AWS 的區域。
  - 身分驗證方法：從 SAP Credential Store 選取登入資料。
  - 輸入儲存在 SAP Credentials Store 中登入資料的命名空間和金鑰名稱。
  - 輸入建立的通訊安排名稱，以在適用於 SAP ABAP 的 SDK - BTP 版本和 SAP 登入資料存放區之間建立通訊。
  - 將停用 IAM 角色保留空白。
- 在已建立項目旁的向右箭頭鍵上按一下滑鼠右鍵，以導覽至 IAM 角色映射索引標籤。

  選取新增項目。
  - 輸入序號：010。
  - 輸入邏輯 IAM 角色：TESTUSER。
  - 輸入 IAM 角色 ARN：輸入包含上一個步驟中建立之TranslateReadOnly政策的 IAM 角色 arn：aws：。

步驟 5：授權 SAP 使用者

SAP 使用者預設無權使用 AWS 功能。使用者必須使用 SAP 授權明確授權。如需詳細資訊，請參閱下列索引標籤。

SDK for SAP ABAP

建立 PFCG 角色

- 前往交易 PFCG
- 輸入角色名稱 ZAWS_SDK_DEMO_TESTUSER ，然後選取 建立單一角色。
  - 描述：Role for demo AWS SDK functionality。
  - 前往授權索引標籤。
  - 選取變更授權資料，並接受資訊彈出式視窗。
  - 在選擇範本快顯視窗中，選取不選取範本。
  - 從工具列選取手動新增。
  - 新增下列授權物件：
    - /AWS1/LROL
    - /AWS1/SESS
  - 在授權樹狀目錄中，輸入：
    - 存取 AWS APIs設定檔： DEMO
    - 邏輯 IAM 角色： TESTUSER
  - 選取儲存。
  - 選取產生。
  - 選取返回。
  - 選取儲存以儲存角色。

將 PFCG 角色指派給 SAP 使用者

指派ZAWS_SDK_DEMO_TESTUSER 角色的任何使用者將獲授權使用 AWS SDK 函數與 DEMO SDK 描述檔中設定的設定。授權使用者也會擔任對應至該設定檔中邏輯 TESTUSER IAM 角色的 IAM 角色。

- 執行交易 SU01。
  - 輸入將測試 AWS SDK 功能的 SAP 使用者的使用者 ID。
  - 選取變更。
  - 前往角色索引標籤，並將ZAWS_SDK_DEMO_TESTUSER 角色指派給使用者。
  - 選取儲存。

SDK for SAP ABAP - BTP edition

建立業務角色

- 在 Web 瀏覽器中開啟您的 ABAP 環境，然後導覽至維護業務角色應用程式。
- 選取從範本建立，然後輸入下列詳細資訊。
  - 範本 – 選擇 **/AWS1/RT_BTP_ENDUSER**。
  - 新的業務角色 ID – 輸入 ID。
  - 新業務角色描述 – 輸入描述。
- 選取確定以查看業務角色的頁面。
- 在一般角色詳細資訊索引標籤下，前往存取類別，並將寫入、讀取、值說明欄位設定為受限制。
- 選取維護限制，然後從左側導覽窗格中展開指派的限制類型。在限制和值區段中更新下列欄位。
  - 在選擇 SDK 工作階段下，選取 SDK 設定檔旁的鉛筆圖示，然後導覽至範圍索引標籤。輸入 **DEMO**，然後選取新增。
  - 在選擇邏輯 IAM 角色下，選取邏輯 IAM 角色旁的鉛筆圖示，然後導覽至範圍索引標籤。輸入 **TESTUSER**，然後選取新增。

    選取 SDK 設定檔旁的鉛筆圖示，然後導覽至範圍索引標籤。輸入 **DEMO**，然後選取新增
- 導覽回業務角色範本，然後開啟業務使用者索引標籤。選取新增，將新建立的業務角色指派給將測試 SDK 功能的 SAP 業務使用者。選取儲存。

指派給所建立商業角色的任何商業使用者，將獲授權使用 AWS SDK 函數與 DEMO SDK 設定檔中設定的設定。授權使用者也會擔任對應至該設定檔中邏輯 TESTUSER IAM 角色的 IAM 角色。

步驟 6：撰寫程式碼

如需詳細資訊，請參閱下列索引標籤。

SDK for SAP ABAP

1. 開啟交易 SE38。
   - 輸入 ZDEMO_TRANSLATE_HELLO_WORLD做為程式名稱。
   - 選取 Create。
   - 輸入 AWS SDK Hello World In Any Language做為標題。
   - 類型：選擇可執行程式。

- 狀態：選擇測試程式。
- 選取儲存。
- 將程式儲存為本機物件。

新增下列程式碼。

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

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

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

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

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

```
  ENDTRY.
```

SDK for SAP ABAP - BTP edition

1. 在需要建立 ABAP 類別的套件上按一下滑鼠右鍵，然後選取新增 > ABAP 類別。
2. **ZCL_DEMO_XL8_SIMPLE** 針對類別名稱輸入 ，然後新增類別描述。選取下一步。
3. 建立或選擇傳輸請求。選取完成。

新增下列程式碼。

```
CLASS zcl_demo_xl8_simple DEFINITION
  PUBLIC
  FINAL
  CREATE PUBLIC .

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


CLASS zcl_demo_xl8_simple IMPLEMENTATION.
  METHOD if_oo_adt_classrun~main.

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

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

        out->write( |Source Phrase: { pv_text }| ).
        out->write( |Target Phrase: { lo_output->get_translatedtext( ) }| ).
      CATCH /aws1/cx_xl8unsuppedlanguage00 INTO DATA(lo_lang).
```

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

如需如何撰寫使用 SDK 之 ABAP 程式碼的詳細資訊，請參閱使用 適用於 SAP ABAP 的 AWS SDK。

步驟 7：執行應用程式

如需詳細資訊，請參閱下列索引標籤。

SDK for SAP ABAP

在 中執行應用程式SE38。如果成功，以下將是您的輸出。

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

如果您缺少授權、組態或 Basis 先決條件，您可能會收到錯誤訊息。請參閱以下範例。

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

如果您的 SAP 角色授權您使用 SDK 設定檔，並在您的 IAM 許可未設定為讓 SAP 系統擔任 IAM 角色時將其映射至邏輯 IAM 角色，則以下將是您的輸出。

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

在此情況下，請檢閱您在 中定義的 IAM 角色、使用者或兩者上的 IAM 許可和信任組態the section called “步驟 1：準備 AWS 您的帳戶”。

SDK for SAP ABAP - BTP edition

在 Eclipse > 執行為 > ABAP 應用程式 （主控台） 上執行應用程式。如果成功，以下將是您的輸出。

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

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

如果您缺少授權、組態或 Basis 先決條件，您可能會收到錯誤訊息。請參閱以下範例。

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

如果您的 SAP 角色授權您使用 SDK 設定檔，並在您的 IAM 許可未設定為讓 SAP 系統擔任 IAM 角色時將其映射到邏輯 IAM 角色，則以下將是您的輸出。

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

在此情況下，請檢閱您在 中定義的 IAM 角色、使用者或兩者上的 IAM 許可和信任組態the section called “步驟 1：準備 AWS 您的帳戶”。

設定

本節提供如何設定開發環境以使用 的相關資訊 適用於 SAP ABAP 的 AWS SDK。

主題

- SAP 先決條件
- 安裝 適用於 SAP ABAP 的 AWS SDK
- 安裝適用於 SAP ABAP 的 AWS SDK - BTP 版本

SAP 先決條件

當您的 SAP 系統託管時，下列安裝 SDK 的先決條件適用 AWS。

主題

- 適用於 SAP ABAP 的 AWS SDK 先決條件
- 適用於 SAP ABAP 的 AWS SDK 先決條件 - BTP 版本

適用於 SAP ABAP 的 AWS SDK 先決條件

以下是適用於 SAP ABAP 的 AWS SDK 先決條件。

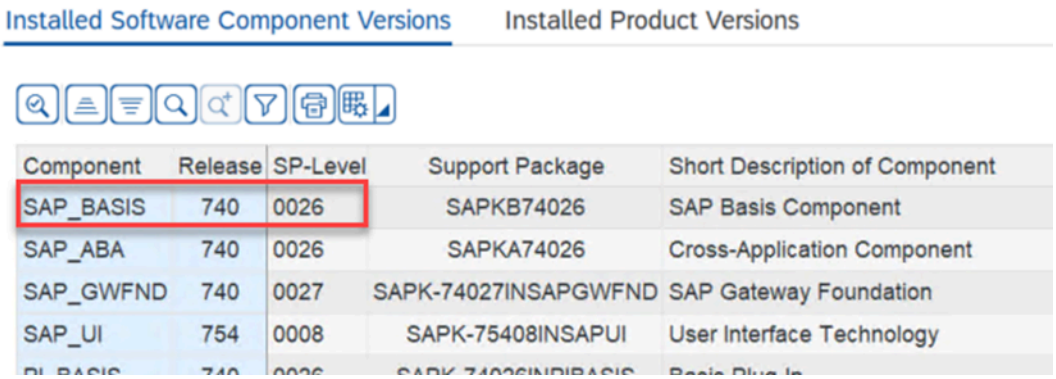
主題

- 基礎版本
- 核心版本
- 參數
- 備註
- 傳出連線
- HTTPS 連線
- 存取 Amazon EC2 執行個體中繼資料

基礎版本

適用於 SAP ABAP 的 SDK 與 SAP NetWeaver 7.4 及更高版本相容。適用於 SAP ABAP 的 SDK 不會接觸任何 SAP 應用程式資料表。它完全與應用程式無關，例如 SAP Enterprise Resource Planning 和 SAP Landscape Transformation Replication Server。

支援的 SP 層級下限SAP_BASIS 740為 SP 0008。如需詳細資訊，請參閱 CL_HTTP_ENTITY 中的 SAP Note 1856171 - 支援相同名稱的表單欄位（需要 SAP 入口網站存取權）。根據您的業務需求，您可以選擇更高的 SP 層級，如下圖所示。


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

SAP_BASIS 750 和更新版本沒有最低 SP 層級需求。

核心版本

適用於 SAP ABAP 的 SDK 和使用網際網路通訊管理員 (ICM) 進行 HTTP 連線的工具，依賴 SAP 核心提供其密碼編譯、HTTP、XML 和 JSON 功能。我們建議您使用與您的 SAP NetWeaver 平台相容的最新核心版本。最低需求是核心版本 741。如需詳細資訊，請參閱 SAP Note 2083594 - SAP 核心版本和 SAP 核心修補程式層級（需要 SAP 入口網站存取權）。

如果您使用的是核心版本 741 或 742，則需要下列修補程式層級：

- 741 patchno 212
- 742 修補程式 111

參數

您的 SAP 系統必須支援伺服器名稱指示 (SNI)，如以下 SAP 備註所述（需要 SAP 入口網站存取權）。

- SAP Note 2124480 - ICM/Web 發送器：作為用戶端的 TLS 延伸伺服器名稱指示 (SNI)

- SAP Note 2582368 - 透過 saphttp、sapkprotp、sldreg 用戶端傳送 TLS 延伸模組 SNI 的 SapSSL 更新

在 DEFAULT.PFL 檔案中設定下列參數。

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

備註

將下列 SAP 備註套用至您的系統。

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

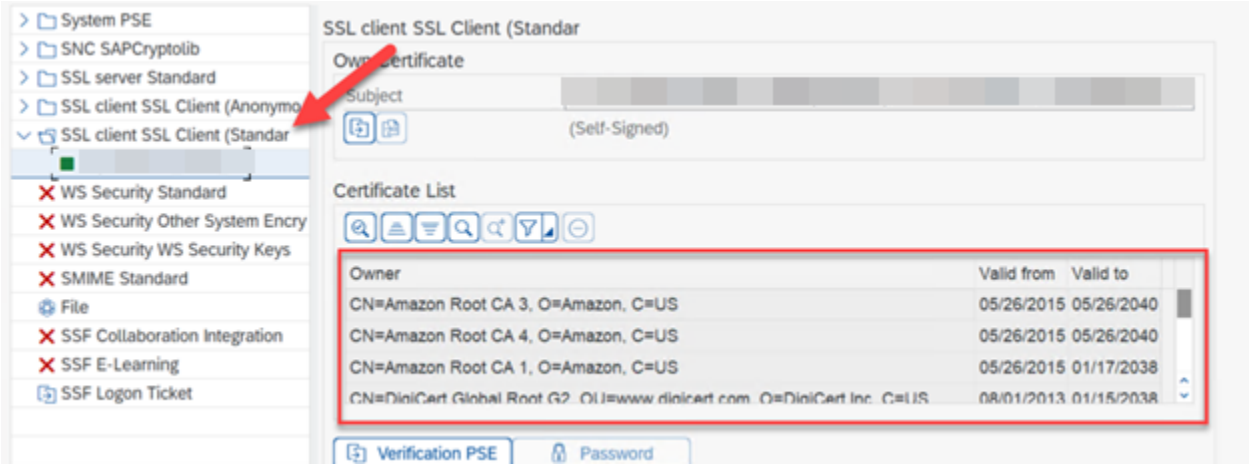
傳出連線

適用於 SAP ABAP 的 SDK 是 HTTPS 用戶端。SAP 系統會傳出 HTTPS 訊息。不需要傳入連線。

HTTPS 連線

所有 AWS API 呼叫都是使用加密的 HTTPS 通道進行。SAP 系統必須設定為信任 AWS 憑證，才能建立傳出 HTTPS 連線。

1. 前往 https://www.amazontrust.com/repository/。
2. 在根 CAs下，使用 PEM 連結下載所有憑證。
3. 在每個 SAP 系統SSL Client (Standard) PSE上的 STRUST 中匯入這些憑證，如下圖所示。



存取 Amazon EC2 執行個體中繼資料

ABAP 系統會對 localhost (http://169.254.169.254) 進行未加密的 HTTP 連線，以啟用 Amazon EC2 執行個體中繼資料。HTTP 通道僅用於從本機伺服器擷取 AWS 登入資料。HTTP 流量會保留在主機內。

中繼資料可讓 中的 SAP AWS 系統安全地進行身分驗證，而不會在 SAP 安全存放區中存放私密金鑰。此功能僅適用於 Amazon EC2 上託管的 SAP 系統。

使用下列參數設定 DEFAULT.PFL 檔案，讓您的 SAP 系統進行未加密的傳出 HTTP 連線。

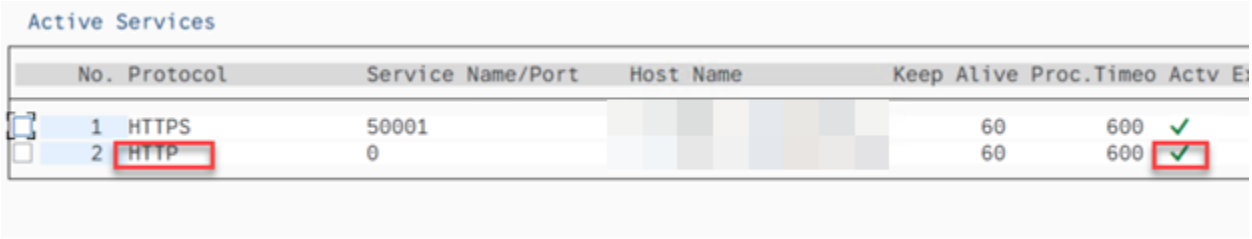
```
icm/server_port_<xx> = PROT=HTTP,PORT=8000,TIMEOUT=60,PROCTIMEOUT=600
```

使用下列參數來啟用傳出 HTTP 連線，而不啟用傳入連線。

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

請確認您的 SAP 系統已設定傳出 HTTP 連線，步驟如下：

1. 執行 SMICM 交易。
2. 前往 Active Services。
3. 確認您在 HTTP 資料列的作用中資料欄下方看到綠色核取記號，如下圖所示。



適用於 SAP ABAP 的 AWS SDK 先決條件 - BTP 版本

以下是適用於 SAP ABAP 的 AWS SDK - BTP 版本的唯一先決條件。

主題

- SAP 景觀入口網站 – BTP 版本
- SAP 登入資料存放區 – BTP 版本

SAP 景觀入口網站 – BTP 版本

此先決條件僅適用於適用於 SAP ABAP 的 AWS SDK - BTP 版本。

SAP Landscape Portal 是在 SAP BTP 環境中安裝附加元件的唯一支援機制。請確定您已訂閱使用此服務。如需詳細資訊，請參閱橫向入口網站。

SAP 登入資料存放區 – BTP 版本

此先決條件僅適用於適用於 SAP ABAP 的 AWS SDK - BTP 版本。

在開發人員預覽中，私密存取金鑰身分驗證是驗證適用於 SAP ABAP 的 AWS SDK - BTP 版本唯一支援的機制。SDK 會從登入資料存放區讀取登入資料，並安全地存放私密存取金鑰。

您必須符合下列先決條件。

- 訂閱登入資料存放區。
- 指定為 BTP 子帳戶權利的登入資料存放區。如需詳細資訊，請參閱初始設定。
- 具有 Credential Store 標準方案的服務執行個體。如需詳細資訊，請參閱建立服務執行個體。

如需詳細資訊，請參閱使用 SAP 登入資料存放區。

SAP Credential Store 服務會在 ABAP BTP 系統外部的 SAP BTP 中執行。如需詳細資訊，請參閱 SAP 登入資料存放區。

安裝 適用於 SAP ABAP 的 AWS SDK

主題

- 下載適用於 SAP ABAP 的 SDK
- 驗證適用於 SAP ABAP 的 SDK 檔案 – 選用
- AWS SDK 傳輸

下載適用於 SAP ABAP 的 SDK

從 https://sdk-for-sapabap.aws.amazon.com/awsSdkSapabapV1/release/abapsdk-LATEST.zip：// 下載 SDK。

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

下載完成時，建議您將下載的檔案解壓縮至目錄，例如 /tmp/awssdk。

驗證適用於 SAP ABAP 的 SDK 檔案 – 選用

驗證 SDK 檔案簽章的這個選用步驟可協助您確認 SDK 未遭到竄改。使用下列步驟來驗證 SDK 檔案。

1. 使用下列命令下載 SDK SIGNATURE 檔案。

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

2. 複製下列公有金鑰，並將它儲存至名為 abapsdk-signing-key.pem 的檔案。

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

3. 使用以下命令驗證下載的 SDK ZIP 檔案。 命令需要 openssl 作為許多 Linux 發行版本的一部分。

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

4. 確認上述命令的輸出為 Verified OK。

5. 如果輸出為 Verification Failure，請重複上述步驟。如果您繼續收到失敗的輸出，請勿安裝 SDK 並聯絡 支援。

AWS SDK 傳輸

主題

- 目錄
- 匯入中
- 命名空間

目錄

適用於 SAP ABAP 的 SDK 安裝是透過 ABAP Transports 完成。您必須將這些傳輸匯入您的開發或沙盒環境。

每個適用於 SAP ABAP 的 SDK 版本都會完全取代先前的版本。 不需要套用增量傳輸。 傳輸會封裝在 ZIP 檔案中。 下列是 ZIP 檔案的結構。

```
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` 資料夾包含`core`子資料夾。`core` 子資料夾包含每個模組的核心執行時間傳輸和子資料夾，以模組的三個字母縮寫命名。如需 TLAs 的完整模組清單，請參閱 適用於 SAP ABAP 的 AWS SDK - 模組清單。

AWS SDK 傳輸是工作台請求。視 TMS 路由的組態而定，軟體開發套件可能不會在匯入先前的系統之後自動轉送至您的品質保證和生產佇列。您必須手動將它們新增至每個系統的佇列。

當您的專案準備好進入下一個階段時，可以匯入 AWS SDK 以及包含您自己的Z程式碼和業務功能的個別傳輸。如果您使用的是變更控制系統，例如 SAP 變更請求管理 (ChaRM)，請洽詢您的 ChaRM 管理員，以正確處理第三方傳輸。

匯入中

主題

- 索引鍵指標
- 匯入時間

AWS SDK 傳輸與用戶端無關。核心傳輸是強制性的，並包含 SDK 執行期程式碼、 的 API AWS Security Token Service和 Amazon Simple Storage Service 的 API。 剩餘的 SDK 模組會以個別傳輸方式交付。若要將 SDK 的大小保留在您的系統中，每個 SDK 模組都是選用的。如果業務邏輯需要，您可以稍後安裝其他模組。

例如，如果您想要使用 Amazon S3 APIs Amazon Translate，並匯入`core`傳輸 （包含核心執行時間、Amazon S3 和 AWS STS 模組） 和`xl8`傳輸 （包含模組的 Amazon Translate) 傳輸。

若要查看適用於 SAP ABAP APIs 的完整清單，請參閱適用於 SAP ABAP 的 SDK - API 參考指南。

以下是匯入 AWS SDK 傳輸時的索引鍵。

- 每個傳輸都會以 `Knnnnnn.AWS`和 的形式交付 `Rnnnnnn.AWS`
  - `Knnnnnn.AWS` 必須複製到 `/usr/sap/trans/cofiles`
  - `Rnnnnnn.AWS` 必須複製到 `/usr/sap/trans/data`。
- 匯入傳輸時，您必須選取匯入傳輸請求 > 選項 > 匯入選項中找到的忽略無效元件版本。
- 您可以同時匯入所有所需的傳輸。
- 如果單獨匯入傳輸，則必須先匯入`core`傳輸。
- 所有傳輸的發行層級必須相同。

匯入時間

AWS SDK 傳輸可能需要幾分鐘的時間才能匯入。如果 STMS 顯示綠色 (RC=0) 或黃色 (RC=4) 光線，則傳輸會成功。

- 紅燈 (RC=8) 表示匯入發生語法錯誤。
  - 選取請求 → 顯示 → 日誌以檢查匯入錯誤。
  - 在匯入期間，如果因為缺少界面 而擲回錯誤`IF_SYSTEM_UUID_RFC4122_STATIC`，請確保 SAP Note 2619546 已套用至系統。如需詳細資訊，請參閱備註。
  - 如果錯誤的原因不明，請聯絡 支援。

- 紅色閃電 (RC=12) 表示傳輸檔案未正確載入至 ，/usr/sap/trans或沒有必要的許可。

索引鍵指標

以下是匯入 AWS SDK 傳輸時的索引鍵。

- 每個傳輸都會以 Knnnnnn.AWS和 的形式交付 Rnnnnnn.AWS
  - Knnnnnn.AWS 必須複製到 /usr/sap/trans/cofiles
  - Rnnnnnn.AWS 必須複製到 /usr/sap/trans/data。
- 匯入傳輸時，您必須選取匯入傳輸請求 > 選項 > 匯入選項中找到的忽略無效元件版本。
- 您可以同時匯入所有所需的傳輸。
- 如果單獨匯入傳輸，則必須先匯入core傳輸。
- 所有傳輸的發行層級必須相同。

匯入時間

AWS SDK 傳輸可能需要幾分鐘的時間才能匯入。如果 STMS 顯示綠色 (RC=0) 或黃色 (RC=4) 光線，則傳輸會成功。

- 紅燈 (RC=8) 表示匯入發生語法錯誤。
  - 選取請求 → 顯示 → 日誌以檢查匯入錯誤。
  - 在匯入期間，如果因為缺少界面 而擲出錯誤IF_SYSTEM_UUID_RFC4122_STATIC，請確保 SAP Note 2619546 已套用至系統。如需詳細資訊，請參閱備註。
  - 如果錯誤的原因不明，請聯絡 支援。
- 紅色閃電 (RC=12) 表示傳輸檔案未正確載入至 ，/usr/sap/trans或沒有必要的許可。

命名空間

適用於 SAP ABAP 的 SDK 使用 /AWS1/ 命名空間，不會修改您系統中的 SAP 物件或任何其他物件，但有下列例外。

- AWS auth 物件位於驗證物件類別中。驗證物件類別限制為四個字元，不支援命名空間。適用於 SAP ABAP 的 SDK 使用身分驗證物件類別為 YAW1。如果您在交易 YAW1 中已有身分驗證物件類別SU21，請在安裝 支援 之前聯絡 。

安裝適用於 SAP ABAP 的 AWS SDK - BTP 版本

BTP 版本處於開發人員預覽版中，可以透過加入預覽版來安裝。若要安裝 SDK，請在AWS 適用於 SAP ABAP 的 SDK - BTP 版開發人員預覽中填寫參與表單。

安裝適用於 SAP ABAP 的 SDK - BTP 版本之前，請確定您符合必要的先決條件。如需詳細資訊，請參閱 SAP 景觀入口網站和 SAP 登入資料存放區。

主題

- 安裝適用於 SAP ABAP 的 SDK - BTP 版本
- 模組
- 修補適用於 SAP ABAP 的 SDK - BTP 版本

安裝適用於 SAP ABAP 的 SDK - BTP 版本

1. 前往 SAP 景觀入口網站執行個體，並啟動部署產品 fiori 應用程式。
2. 在產品中，選取合作夥伴產品**/AWS1/SDK_OMNI**下的 。

   支援 如果您在開發人員預覽中接受*/AWS1/SDK_OMNI*之後沒有看到 ，請聯絡 。
3. 在目標版本中，選擇您要安裝在系統上的適用於 SAP ABAP 的 SDK - BTP 版本。
4. 在可用系統中，勾選您要安裝 SDK 的所有 SIDs的核取方塊。
5. 選取部署，輸入排程詳細資訊，然後選取排程。您可以在產品版本部署狀態中監控進度。

   安裝可能需要 30-45 分鐘，並包含系統停機時間。如需詳細資訊，請參閱部署產品。

模組

下列模組包含在適用於 SAP ABAP 的 AWS SDK - BTP 版本的開發人員預覽中。

- Amazon API Gateway 【agw】
- Amazon Athena 【ath】
- Amazon Bedrock 執行時間 【bdr】
- Amazon Comprehend 【cpd】
- Amazon EventBridge 【evb】
- Amazon Forecast 【fcs】

- Amazon Kinesis 【kns】
- Amazon Data Firehose 【frh】
- Amazon SageMaker AI 【sgm】
- Amazon Simple Notification Service 【sns】
- Amazon Simple Queue Service 【sqs】
- Amazon Simple Storage Service 【s3】
- AWS Systems Manager [ssm]
- Amazon Textract 【tex】
- Amazon Transcribe 【tnb】
- Amazon Translate 【xl8】
- AWS CloudTrail [trl]
- AWS IoT [iot]
- AWS KMS [kms]
- AWS Lambda [lmd]
- AWS Secrets Manager [smr]
- AWS Security Token Service [sts]
- AWS Transfer Family [trn]
- IAM Roles Anywhere 【rla】
- Amazon Redshift 資料 API 【rsd】

修補適用於 SAP ABAP 的 SDK - BTP 版本

SDK for SAP ABAP - BTP 版本的修補程序類似於安裝程序。如果您在已安裝舊版 的系統上安裝軟體開發套件，則軟體開發套件會修補至您選擇的新版本。

設定 適用於 SAP ABAP 的 AWS SDK

使用 之前 適用於 SAP ABAP 的 AWS SDK，您必須使用軟體開發套件操作所需的技術和功能設定來設定軟體開發套件。有些設定是可傳輸的，有些則是執行時間設定。許多設定與其他 SDKs `.INI` 檔案中定義的設定直接類似。

除了執行期設定之外，開發套件組態必須在您的開發環境中完成。您可以依照一般的傳輸和變更控制規則，將組態傳輸到 QA 和生產環境。生產環境不建議使用可傳輸組態。

如果您沒有設定 AWS SDK 的許可，請參閱 SAP 授權。

設定 適用於 SAP ABAP 的 AWS SDK

若要執行組態交易，請在 SAPGUI 命令列`/n/AWS1/IMG`中輸入 。

設定適用於 SAP ABAP 的 AWS SDK - BTP 版本

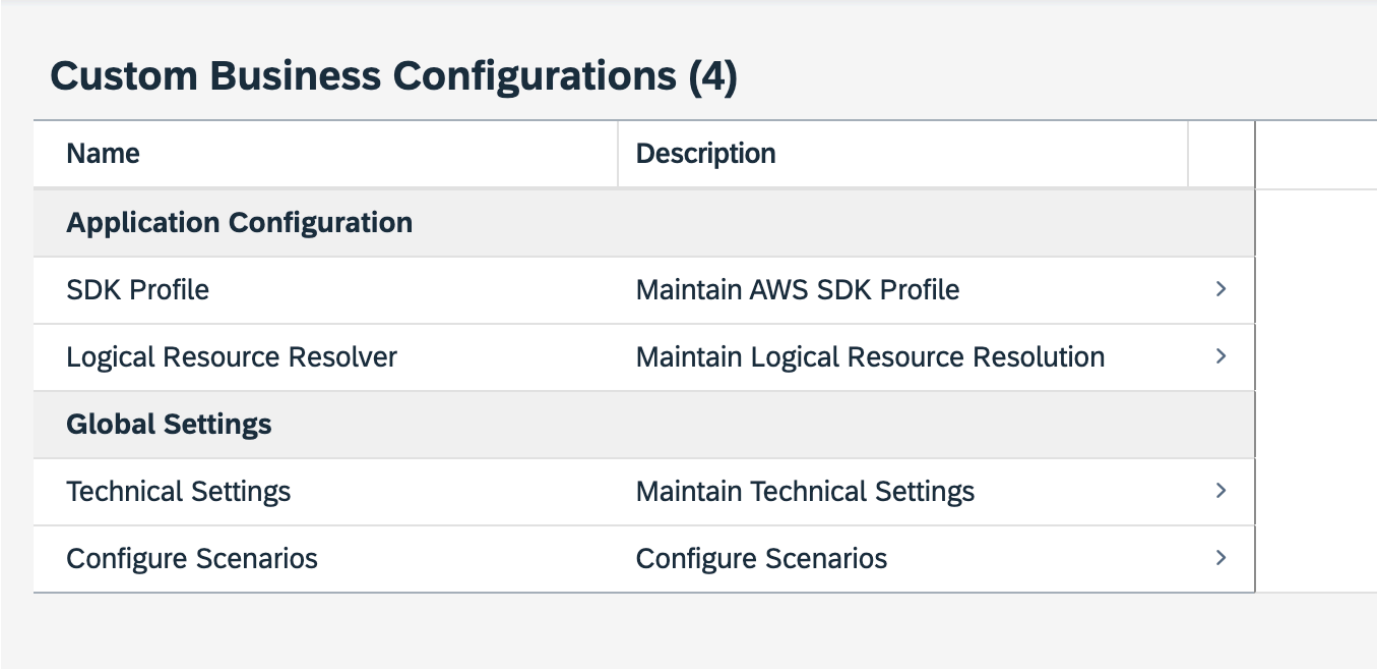
使用下列步驟來設定適用於 SAP ABAP 的 SDK - BTP 版本。

1. 在 Web 瀏覽器中開啟您的 ABAP 環境。
2. 導覽至自訂業務組態應用程式。

若要使用匯出自訂傳輸應用程式建立自訂請求，請參閱使用匯出自訂傳輸應用程式 - 建立請求。

在自訂商業組態應用程式中，您可以根據 SDK 設定的類型來分組組態。使用下列步驟將組態分組。

1. 在 Web 瀏覽器中開啟您的 ABAP 環境，然後導覽至自訂業務組態應用程式。
2. 選取設定 > 群組，然後從下拉式清單中選擇組態群組。選取 OK (確定)。
3. 這些組態現在以階層結構提供，如影像所示。若要儲存檢視，請參閱檢視 （變體管理） - 元件。



本節涵蓋下列主題。

主題

- 全域設定
- 應用程式組態
- 執行期設定
- 進階連線案例
- 服務提供者設定
- 的重新整理、追蹤和遙測主題 適用於 SAP ABAP 的 AWS SDK

全域設定

使用適用於 SAP ABAP 的 AWS SDK 的 `/n/AWS1/IMG` IMG 轉換，以及適用於適用於 SAP ABAP 的 AWS SDK 的自訂業務組態應用程式 - BTP 版本來設定全域設定。本主題會交替使用 IMG 和自訂商業組態。

本節涵蓋下列主題。

主題

- 技術設定
- 設定案例

技術設定

/AWS1/IMG 交易的全域設定會影響整個 SDK 的行為。這些設定通常由 Basis 管理員設定。您可以將這些值設定為下列建議的設定。

- 選取新增項目。
  - S3 區域化：使用 s3.amazonaws.com 存取 us-east-1 儲存貯體。
  - STS 區域化：使用全域端點存取 STS。
  - 停用 EC2 中繼資料：將此欄位保留空白。此欄位在 BTP 版本中是唯讀的，預設為「是」。
  - 中繼資料結束模式：使用 IPv4 中繼資料端點。此欄位在 BTP 版本中為唯讀，且已自動更新。
  - 中繼資料結束 URL：此欄位保持空白。此欄位在 BTP 版本中為唯讀。
- 選取 Save (儲存)。

設定案例

案例可讓 AWS SDK 在多區域災難測試或災難復原測試案例期間更有效率地切換設定。您可能不需要此功能，而只需要設定下列 DEFAULT 案例。

- 選取新增項目。
  - 案例 ID：DEFAULT
  - 案例描述：預設案例
- 選取 Save (儲存)。

如果您有多區域災難復原設定或其他需要快速變更設定的唯一案例，則可以設定多個案例。

- DEFAULT - 標準操作。
- DR - 如果災難需要將整個系統移至另一個區域，則為特殊組態。
- DR_TEST - 模擬災難的特殊組態，例如，在臨時生產複製中。

應用程式組態

設定適用於 SAP ABAP 的 SDK 類似於設定其他以 ABAP 為基礎的應用程式。它會組織成不同的設定檔，將各種案例的設定分組。ABAP SDK 設定檔會定義特定應用程式案例所需的設定。例如，如果交易 ZVA01、 ZVA02和 ZVA03是增強並執行於 上的發票相關交易， AWS 服務例如 Amazon S3 AWS Lambda和 Amazon SageMaker AI，則可以ZINVOICE建立名為 的 SDK 設定檔。此設定檔可將發票相關功能的技術設定、SAP 授權和 IAM 角色映射分組。

使用適用於 SAP ABAP 的 AWS SDK 的/n/AWS1/IMG轉換，以及適用於適用於 SAP ABAP 的 AWS SDK 的自訂業務組態應用程式 - BTP 版本來設定全域設定。本主題會交替使用 IMG 和自訂商業組態。

主題

- SDK 描述檔
- 邏輯資源解析程式
- 範例

SDK 描述檔

ABAP SDK 設定檔會為每個 SID 和用戶端定義下列項目。

> **Note**
>
> SAP BTP、ABAP 環境中的用戶端一律為 100。

- 所有 API 呼叫的預設 AWS 區域。例如，如果您的 SAP 系統在 us-east-1區域中執行，則您的其他 AWS 資源可能也位於相同的區域，這應該是您的預設區域。您的 ABAP 程式碼可以覆寫預設區域。
- 身分驗證方法
  - 對於在 Amazon EC2 上執行的 SAP 系統，強烈建議選擇執行個體角色中繼資料，以受益於短期、自動輪換的憑證。
  - 對於在內部部署或其他雲端中執行的 SAP 系統，您必須從 SSF 儲存中選擇憑證。
  - 對於在 SAP BTP 上執行的 ABAP 系統，您必須從 SAP Credential Store 選擇登入資料。如需詳細資訊，請參閱使用 SAP 登入資料存放區進行身分驗證。
  - 針對跨帳戶角色鏈結案例，選擇來源設定檔並指定來源設定檔 ID。這可自動解析和執行角色擔任鏈結。如需詳細資訊，請參閱使用來源設定檔進行跨帳戶存取。

- 邏輯 IAM 角色與 IAM 角色的映射。
  - 此映射會依遞減優先順序排序。
  - 系統會自動為使用者選取在 PFCG 角色中授權使用者的最高優先順序 IAM 角色。
- 服務到自訂端點的選用映射。此組態會在進階連線案例中討論

> **Note**
>
> PFGC 角色在 SAP BTP、ABAP 環境中稱為業務角色。

當 ABAP 程式想要連線到 時 AWS 服務，它會指定 ABAP SDK 設定檔來提取必要的設定。AUTHORIZATION-CHECK 將執行 ，以確認使用者具有存取 SDK 描述檔的許可。您的 SAP 安全管理員可以定義將存取權授予適當使用者的 PFCG 角色。

邏輯資源解析程式

邏輯資源解析程式可讓您使用標準位置來存放資源名稱。它隨附適用於 SAP ABAP 的 SDK。其動作類似於FILE交易將邏輯檔案名稱映射至實體檔案名稱的方式。

邏輯資源定義 AWS 資源的概念，例如存放發票的 Amazon S3 儲存貯體。例如，此邏輯資源可以命名，ZINVOICES_OUTBOUND而且可以對應到不同的實體儲存貯體名稱，具體取決於 SAP 系統是開發、QA 還是生產。

適用於 SAP ABAP 的 SDK 設定為 QA 系統將邏輯資源解析為 QA 實體資源，即使在從生產環境重新整理系統之後也是如此。ALL 系統的資源映射在您的開發 SAP 系統中定義並向前傳輸。 此方法與 SAP 系統的一般設定不同，其中映射處理為主資料並在每個系統中設定。SDK for SAP ABAP 提供的邏輯資源解析程式的優點是系統重新整理後發生傳輸錯誤的機率幾乎沒有。

範例

有四個不同的 Amazon S3 儲存貯體，分別用於開發、生產和 QA，以及第二個 QA 儲存貯體用於迴歸測試。

當 SDK 將類似 的邏輯資源解析ZINVOICE_OUTBOUND為實體資源時，它會檢查SY-SYSIDSY-MANDT並詢問我在哪個 SID 和用戶端中執行？ ，並自動選取正確的實體資源。

如果生產環境中資源的映射需要變更，您必須在開發系統的 IMG 中變更映射並將其轉送。這可確保將 AWS 資源重新指派給 SAP 系統與任何其他傳輸一樣受到變更控制。

> **Note**
>
> 由於 SDK 組態與用戶端相關，資源的重新指派會在 自訂請求中傳輸，且傳輸必須匯入至每個用戶端。

執行期設定

本節涵蓋下列主題。

> **Note**
>
> 這些設定無法傳輸，且是每個 SAP 系統的本機設定。

主題

- 日誌和追蹤
- OPT-IN：增強型遙測
- 作用中案例

日誌和追蹤

您可以啟用追蹤以進行偵錯。除非診斷技術問題，否則建議將追蹤層級保持在 No Trace。如需詳細資訊，請參閱安全操作。

這些設定不適用於適用於 SAP ABAP 的 SDK - BTP 版本。

OPT-IN：增強型遙測

所有 SDKs都會將遙測資訊傳送至 AWS ，以用於支援目的。您可以選擇加入增強型遙測。當您聯絡 支援 以識別特定 API 呼叫的來源時，這特別有用。如需詳細資訊，請參閱追蹤和遙測。

這些設定不適用於適用於 SAP ABAP 的 SDK - BTP 版本。

作用中案例

在此交易中啟用您的DEFAULT案例。每個系統只需要啟用一次，除非系統正在進行多區域災難復原，否則不應變更此啟用。在多區域設定中，您可以使用此設定將 SAP 系統切換到災難復原環境或災難復原測試案例。

進階連線案例

適用於 SAP ABAP 的 AWS SDK 透過對 AWS 端點進行 HTTPS 呼叫 AWS 服務 來使用 。一般而言，AWS 端點可透過網際網路存取。SAP 系統必須能夠連線到網際網路，以建立這些傳出連線。適用於 SAP ABAP 的 SDK 永遠不需要從網際網路到 SAP 系統的傳入連線。

下列案例提供建立傳出連線的不同方式。

案例

- 透過代理伺服器的連線
- 透過封包檢查防火牆的連線
- 閘道端點
- 自訂界面端點
- 進階路由
- 存取多個區域中的端點

透過代理伺服器的連線

若要透過代理伺服器建立連線，請使用下列步驟。

1. 在 SDK 中，前往交易 **SICF**。
2. 選擇 Execute (執行)。
3. 在功能表中，選擇用戶端 > Proxy 伺服器。
4. 將 Proxy 設定設為作用中。
5. 在下列地址的無代理欄位中，列出以分號分隔的任何例外狀況。
6. 在 HTTP 通訊協定和 HTTPs 通訊協定欄位中，指定代理伺服器的連線詳細資訊。

SDK 不知道代理伺服器，且不需要任何設定即可使用 SAP 系統的代理伺服器組態。

> **Note**
>
> 如果您使用 Amazon EC2 執行個體中繼資料身分驗證，則 SAP 系統無法使用代理伺服器存取位於 的本機執行個體中繼資料`http://169.254.169.254`。您必須在下列地址的無代理`169.254.169.254`欄位中包含 。

透過封包檢查防火牆的連線

您可以設定封包檢查防火牆以進行傳出連線。這些防火牆會解密 SSL 流量，然後在將其傳遞至端點之前重新加密。此組態通常需要防火牆向使用 的 SAP 系統發行自己的憑證 AWS 服務。您必須在 中安裝防火牆的 CA 憑證`STRUST`。如需詳細資訊，請參閱 HTTPS 連線。

閘道端點

有些 AWS 服務 提供閘道端點，為 VPC 提供無需網際網路的高效能存取。這些端點對適用於 SAP ABAP 的 SDK 來說是透明的，不需要任何組態。

如需詳細資訊，請參閱閘道端點。

自訂界面端點

如果您需要使用自訂端點覆寫預設端點解析度，您可以使用介面端點來提供 VPC 無需網際網路的高效能存取。如需詳細資訊，請參閱設定介面端點。

不使用私有 DNS 時，這些端點有自己的 DNS 地址，而且 ABAP 程式必須明確覆寫一般的端點解析邏輯。如需詳細資訊，請參閱 AWS re:Post – 為什麼我無法解析介面 VPC 端點的服務網域名稱？

在下列範例中，會為 AWS STS 和 建立介面端點 Amazon Translate。SAP 系統不使用私有 DNS，並使用自訂端點呼叫 服務。中定義的邏輯資源`/AWS1/IMG`代表實體界面端點地址，例如`vpce-0123456789abcdef-hd52vxz.translate.us-west-2.vpce.amazonaws.com`。這可避免對程式碼中的 DNS 進行硬式編碼。

在下列程式碼中， 中的邏輯資源`/AWS1/IMG`會先解析為實體端點名稱。然後，它們會提供給 AWS 工作階段類別 （使用 AWS STS 擔任 IAM 角色） 的原廠方法，並翻譯 API 類別。

```
" 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...
```

如範例所示，上的任何方法呼叫go_xl8都會前往端點 `https://vpce-0123456789abcdefg-12345567.translate.us-west-2.vpce.amazonaws.com`。您也可以在 IMG 組態中定義路由自訂端點，而不是在程式碼中，如下一節所示。

進階路由

在上一節中，我們示範如何在 SDK 模組的原廠方法引iv_custom_endpoint數中指定自訂端點。隨著使用 SDK 的 ABAP 程式數量增加，這可能會變得難以管理。您可以在 SDK 設定檔中設定從 AWS 服務 到自訂端點的映射。對於每個 SID、用戶端和案例，服務三個字母縮寫 (TLA) 可以映射到端點 URL：

TLA	自訂端點 URL
BDR	https://vpce-23456789abcdef012-3c4d5e6f.bedrock-runtime.us-east-1.vpce.amazonaws.com

TLA	自訂端點 URL
LMD	https://vpce-123456789abcdef01-2b3c4d5e.lambda.us-east-1.vpce.amazonaws.com
S3	https://vpce-0123456789abcdef0-1a2b3c4d.s3.us-east-1.vpce.amazonaws.com

使用此組態，您不需要在原廠方法呼叫`iv_custom_endpoint`中指定 。自訂端點會自動從組態資料表中選取。組態專屬於 SDK 設定檔，因此您可以使用不同的路由來建立多個設定檔，以滿足您的需求。如同其他 SDK 設定檔組態，路由是 SID 和用戶端特定，因此可以為不同的系統定義個別路由。

存取多個區域中的端點

AWS 端點會自動從您的 SDK AWS 區域 設定檔中定義的預設值決定。您也可以透過程式設計方式指定區域，覆寫預設區域。這可以在原廠`CREATE( )`方法中覆寫，或在稍後使用 SDK 的組態物件覆寫。如需詳細資訊，請參閱程式設計組態。

在下列範例中，原廠`CREATE( )`方法用於設定 區域，並列出 `us-east-1`和 區域中`us-west-2`的 Amazon SQS 佇列。

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

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

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

```
ENDLOOP.
```

服務提供者設定

基礎管理員有時需要從用戶端 控制整個系統 SDK 的某些功能000。這是託管和服務供應商的常見案例，這些服務供應商代表其客戶在自己的 AWS 帳戶中操作系統。 AWS 適用於 SAP ABAP 的 SDK 支援服務供應商設定。這些設定是在用戶端 中設定000，並影響所有用戶端的 SDK。SDK for SAP ABAP - BTP 版本不支援服務供應商設定。

服務提供者設定是在交易 中設定/AWS1/IMG，且必須在用戶端 中設定000。忽略其他用戶端中的服務提供者設定。用戶端中的設定會在所有用戶端中000生效，並在發生衝突時疊代其他IMG設定。

使用下列步驟在用戶端 中設定服務提供者設定000。

1. 展開交易 中的服務提供者設定分支/AWS1/IMG。
2. 選擇服務提供者護欄
3. 選取新增項目，然後根據您的業務需求調整設定。
    - 停用 EC2 中繼資料 – 防止 SDK 存取所有用戶端中的 EC2 執行個體中繼資料，即使 SDK 設定檔設定為使用 EC2 執行個體中繼資料進行身分驗證。如果 ABAP 程式嘗試使用 SDK 存取執行個體中繼資料，則 SDK 會引發例外狀況。
4. 選取 Save (儲存)。

的重新整理、追蹤和遙測主題 適用於 SAP ABAP 的 AWS SDK

本節涵蓋下列主題。

主題

- SAP 系統重新整理
- 追蹤
- 遙測

SAP 系統重新整理

系統重新整理後，Basis 管理員的主要挑戰是確保個別系統不會存取彼此的資源。例如，您可能想要確保 QA SAP 系統不會存取生產環境的資源，例如 S3 儲存貯體。

適用於 SAP ABAP 的 SDK 為這項挑戰提供了邏輯資源的安全意識方法。商業分析師可以採取下列步驟。

1. 定義邏輯資源，例如 ZINVOICE_OUTBOUND。
2. 映射開發系統中的所有系統和用戶端。
3. 將 ALL 系統的組態向前傳輸，直到生產環境。

重新整理後的基礎步驟

1. 檢查身分驗證

- 如果系統使用私密存取金鑰身分驗證，則 SSF 加密的憑證會無效，因為它們存放在主資料中。必須重新輸入登入資料，這可能需要在 https://console.aws.amazon.com/iam/：//www. 中重新產生新的私密存取金鑰。
- 如果系統使用 EC2 執行個體中繼資料進行驗證，則不需要任何步驟。

檢查追蹤設定

- 在 中/AWS1/IMG，確保追蹤設定是您想要的。這些設定無法傳輸。

追蹤

追蹤輸出是在 IMG 執行時間設定中控制。

您可以使用的追蹤層級為：

- 無追蹤
- 追蹤 API 呼叫
- 追蹤 API 呼叫和承載

  此選項包含未加密的承載資訊。
- 追蹤 API 呼叫、承載和內部 XML 轉換

  此選項包含未加密的承載資訊。

如果啟用 API 追蹤，則會將追蹤寫入 DIR_WORKaws1_trace-*YYYY-MM-DD*.log檔案中。

如果另外啟用承載追蹤，aws1_payload_*則會為每個呼叫和承載元件建立具有標題的其他檔案。承載追蹤長度可以受到限制，套用到每個個別承載追蹤的長度限制會失敗。

承載追蹤主要用於收集在序列化錯誤 支援 時提供給 的資訊。建議您選擇無追蹤，除非您嘗試診斷 SDK 錯誤。

> **Note**
>
> 承載追蹤可以包含未加密的商業資訊。我們建議您僅針對 AWS Support 的請求開啟這些追蹤，以協助您進行故障診斷。您可以在解析之後關閉這些追蹤。系統不會自動刪除追蹤，而且不再需要時，系統管理員需要移除追蹤。

這些設定不適用於適用於 SAP ABAP 的 SDK - BTP 版本。

遙測

SDKs會將遙測資訊傳送至 支援。適用於 SAP ABAP 的 SDK 會收集下列資訊：

- 作業系統版本和修補程式層級
- SAP_BASIS 版本和修補程式層級
- SAP 核心版本和修補程式層級

您可以選擇接收以下資訊 支援。

- SAP SID 和執行個體名稱 (host_sid_nn)
- SAP 用戶端 (SY-MANDT)
- 交易代碼 (SY-TCODE) 和報告 (SY-REPID)

其他資訊 支援 可讓您獲得更好的協助。 支援 可以偵測為何進行特定 API 呼叫，並可在 SAP 系統中進一步尋找相關交易。

遙測僅限於適用於 SAP ABAP 的 SDK 的 SDK 和 API 版本 - BTP 版本。

使用 適用於 SAP ABAP 的 AWS SDK

適用於 SAP ABAP 的 SDK 有兩個主要元件。

- SDK 執行期 （ 套件 /AWS1/RT) – 一組物件，其可支援安全性、身分驗證、追蹤、組態、資料轉換和其他跨 API 函數。Amazon S3、IAM Roles Anywhere AWS STS和 Secrets Manager 的 API 模組是強制性的。
- APIs（ 套件/AWS1/API及其子套件 ） – 每個 API 的子套件，其中每個 API 的物件彼此完全獨立，確保一個 API 的變更不會破壞另一個 API。若要查看 APIs 適用於 SAP ABAP 的 AWS SDK 的完整清單，請參閱 適用於 SAP ABAP 的 AWS SDK - API 參考指南。

本節涵蓋下列主題。

主題

- ABAP 中的資料表示
- Amazon S3 範例程式
- 適用於 SAP ABAP 的 SDK 概念
- 適用於 SAP ABAP 的 AWS SDK features
- 使用 SDK 建置產品
- 自訂對 的 HTTP 請求 AWS
- 限制

ABAP 中的資料表示

本節涵蓋下列主題。

主題

- 資料類型
- AWS 資料類型

資料類型

AWS 服務 有一組標準資料類型，必須映射至 ABAP 資料類型。如需詳細資訊，請參閱下表。

AWS 資料類型	ABAP 資料類型	說明
boolean	C	單一字元"X"和 " "
字串	STRING	
位元組	INT2	INT2 的範圍大於 0-255。大多數 AWS 服務 會截斷溢位，但此行為不會正式定義。
Short	INT2	
Integer	INT4	
Long	DEC19	INT8 在 ABAP 750 之前無法使用。DEC19 用於所有支援的 ABAP 平台之間的相容性和一致性。
Blob	XSTRING	代表二進位資料
Float	STRING	雖然 ABAP 支援 DECFLOATs，但無法代表 NaN、Infinity 和 -Infinity. AWS SDK 等值，在內部將其表示為 STRINGs，並在執行時間將其轉換為 DECFLOAT16。如果代表 NaN、Infinity 或 +Infinity，開發人員可能會處理這些項目，以回應一組特殊例外狀況或映射。
Double	STRING	
bigInteger	STRING	這些值代表無法在 ABAP 中表示的無限長度數字，並使用 STRINGs而不是 bigInteger。
bigDecimal	STRING	
時間戳記	TZNTSTMPS	TZNTSTMPS 可使用原生 ABAP 時間戳記函數進行處理。

AWS 服務 也會傳回下列彙總資料類型。

AWS 資料類型	ABAP 資料類型	說明
結構	類別	
UNION	類別	聯集與結構相同，只是聯集絕不會有多個欄位集。所有其他欄位都會設定為無值。
陣列	標準資料表	
雜湊	HASHED 資料表	雜湊資料表只會有兩個資料欄：KEY （字串） 和 VALUE （類別）。

AWS 資料類型

下列方法已整合為 AWS 服務 支援 ABAP。

- 某些 AWS 資料類型無法在 ABAP 中表示。例如，ABAP 中的`float`資料類型不支援 `NaN`、`Infinity`或 `-Infinity`值。因此，`float`資料類型會以 表示，`STRING`並在`DECFLOAT16`執行時間轉譯為 。
- AWS 資料在線路上以 JSON 或 XML 表示，值為選用。例如，請參閱以下 JSON AWS 服務 中 傳回的範例。

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

如果 Ana 沒有中間名，服務會傳回下列輸出。

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

```
}
```

ABAP 不會區分長度為 0 的字串和沒有值的字串。其他語言可能會將 NULL 值指派給字串，或在建構中包裝字串 （例如 Java 的`Optional<>`包裝函式）。ABAP 不支援這些項目。因此，適用於 SAP ABAP 的 SDK 透過提供 getter 方法的變體來促進值的差異。

Amazon S3 範例程式

本節將引導您完成簡單的範例程式，透過呼叫 列出 Amazon S3 儲存貯體的內容`ListObjectsV2`。

主題

- 先決條件
- 代碼
- 程式碼區段

先決條件

您必須符合下列先決條件，才能執行此範例程式。

- 您有 Amazon S3 儲存貯體。在本教學課程中，儲存貯體名為 `demo-invoices.customer.com`。
- 交易`/AWS1/IMG`：
  - 已定義名為 的 SDK 描述檔`DEMO_S3`。
    - 在 SDK 描述檔中，邏輯 IAM 角色`TESTUSER`必須映射至 IAM 角色，例如 `arn:aws:iam::`*`111122223333`*`:role/SapDemoFinance`授予列出 Amazon S3 儲存貯體內容的`s3:ListBucket`許可。
  - 具有名為 的邏輯資源`DEMO_BUCKET`，其會映射至具有 SAP 系統的 SID 和用戶端的 Amazon S3 儲存貯體。
- 您的使用者具有以下 PFCG 角色：
  - 授權使用者透過身分驗證物件 - 存取 `DEMO_S3` SDK 描述檔`/AWS1/SESS`。
  - 授權使用者透過身分驗證物件 - 進行邏輯 IAM 角色`TESTUSER`存取`/AWS1/LROL`。
- 您的 SAP 系統可以使用 SDK AWS 描述檔中定義的方法，對 進行自我驗證。
- 您的 Amazon EC2 執行個體描述檔授予 SAP 系統 SDK 描述`sts:assumeRole`檔中 IAM 角色`arn:aws:iam::`*`111122223333`*`:role/SapDemoFinance`映射的 權利。

代碼

下列程式碼區塊示範程式碼的外觀。

```
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.
```

程式碼區段

以下是章節中程式碼的檢閱。

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

使用者無法指定實體儲存貯體名稱。他們會指定邏輯儲存貯體和系統管理員（特別是商業分析師），AWS 與管理員將邏輯儲存貯體映射到 中的實體儲存貯體/AWS1/IMG。在大多數商業案例中，使用者沒有機會選擇邏輯儲存貯體 — 邏輯資源 ID 在程式碼中硬式編碼或在自訂組態資料表中設定。

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

這行會建立安全工作階段，並宣告此 ABAP 程式預期使用 DEMO_S3 SDK 描述檔。此呼叫是 SDK 組態的連線，並提取預設區域、身分驗證設定和所需的 IAM 角色。AUTHORIZATION-CHECK 系統會自動呼叫 ，以確保/AWS1/SESS滿足授權物件。此外，會根據授權物件 進行AUTHORIZATION-CHECK呼叫，以判斷使用者獲得授權的最強大（序號較低）邏輯 IAM 角色/AWS1/LROL。SDK 將假設 IAM 角色映射到 SID 和用戶端的邏輯 IAM 角色。然後，工作階段物件會根據 中的追蹤設定啟用追蹤IMG。

如果使用者未獲得請求的 SDK 設定檔或任何可用邏輯 IAM 角色的授權，則會引發例外狀況。

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

此行會將邏輯資源解析為實體儲存貯體名稱。如果因為組態沒有此 SID/用戶端組合的映射，而無法解析邏輯資源，則會引發例外狀況。

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

此行使用 的 create()方法，為 Amazon S3 建立 API 物件/aws1/cl_s3_factory。傳回的物件類型為 /aws1/if_s3，這是 Amazon S3 API 的界面。必須為每個服務建立單獨的 API 物件。例如，如果 ABAP 程式正在使用 Amazon S3 AWS Lambda和 DynamoDB，則會從 /aws1/cl_s3_factory、 /aws1/cl_lmd_factory和 建立 API 物件/aws1/cl_dyn_factory。

建構函數有一些選用參數，可讓您在想要覆寫設定的預設區域 時指定區域IMG。如此一來，如果您想要將物件從一個區域中的儲存貯體複製到另一個區域中的儲存貯體，則可能會有兩個 執行個體，/aws1/if_s3一個用於 us-west-2，us-east-1另一個用於 。同樣地，如果您需要報告從財務相關儲存貯體讀取並將物件寫入物流相關儲存貯體/aws1/cl_s3，您可以建立兩個不同的安全工作階段物件，並使用它們來建立兩個不同的 執行個體。

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

```
    ).
```

這行是對 的呼叫ListObjectsV2。它需要簡單的輸入引數，並傳回單一物件。這些物件可能代表深層 JSON 和 XML 資料，並解序列化為 ABAP 物件導向建構。在某些情況下，這可能相當複雜。現在，您只需要處理輸出來列出儲存貯體的內容。

```
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.
```

使用GET...()隱藏資料內部表示法的樣式方法來存取資料。 GET_CONTENTS( )會傳回 ABAP 資料表，且每一列本身都包含代表單一 Amazon S3 項目的物件。在大多數情況下， AWS SDK 採用這種物件導向方法，所有資料都以物件和資料表表示。 LastModified 欄位以時間戳記表示，可以使用 ABAP 原生CONVERT TIME STAMP命令轉換為日期。 會GET_SIZE()傳回 INT4，以便於數學和格式化操作。

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

所有錯誤 – 連線、4xx 用戶端、5xx 伺服器或任何 ABAP 錯誤，例如授權或組態錯誤，都會以例外狀況表示。您可以分別處理每個例外狀況。您可以選擇是否應將例外狀況視為資訊錯誤、重試、警告、簡短傾印或任何其他類型的處理。

適用於 SAP ABAP 的 SDK 概念

本節涵蓋 的基本概念 適用於 SAP ABAP 的 AWS SDK。

主題

- API 類別
- 其他物件
- 結構類別

- 陣列
- 地圖
- 更高層級的函數

API 類別

每個 AWS 服務 都會指派一個三個字母的縮寫或 TLA。服務由 /AWS1/IF_<TLA> 格式的界面表示。我們將此稱為服務界面。API 類別位於 /AWS1/API_<TLA>套件中。服務界面包含每個 AWS 操作的一個方法 （ 我們將呼叫這些方法操作方法 ）。若要查看 適用於 SAP ABAP 的 AWS SDK TLAs 模組清單，請參閱 適用於 SAP ABAP 的 AWS SDK - 模組清單。

每個操作方法都有一些IMPORTING引數，最多有一個RETURNING引數。這些引數通常是具有複雜建構函數和長組GET…( )方法的物件。在許多情況下，物件將包含巢狀物件、遞迴參考、物件資料表、資料表等。這是因為 AWS 服務 正在傳遞深度 XML 和 JSON 結構，這些結構無法以一組平面引數表示。

引RETURNING數一律是類別，即使該類別只包含單一屬性。

其他物件

除了包含主要 API 類別之外，每個 API 套件還包含各種相關的儲存庫和資料字典物件。

- 每個結構類型物件的類別。
- 出現在資料表中任何基本資料類型的類別。例如，如果服務傳回字串資料表，ABAP API 會將其表示為物件資料表，其中每個物件都是封裝字串的包裝函式類別。如此一來，包裝函式類別可以隱藏代表無法在 ABAP 中原生表示的 null 字串的詳細資訊。
- 服務定義之任何特定錯誤的例外類別。
- 每個基本資料類型的資料元素。每個資料類型都有自己的資料元素，以便自我記錄。
- 用於內部處理的其他物件，例如用於序列化和還原序列化 XML 和 JSON 承載的 XSLT 轉換。

結構類別

服務傳送和接收的大多數 AWS 資料會以 AWS類別形式表示 SDK。這些類別代表資料結構，並隱藏儲存體的內部詳細資訊。特別是，類別會隱藏 SDK 代表此欄位的方式沒有值。

對於結構類別中的每個欄位，有三種方法。

GET_field( )

GET_field( ) 方法

- 傳回 欄位的值，或
- 如果 欄位沒有值，則會傳回預設值，您可以將該值設定為選用參數。

例如，請考慮下列程式碼，以列印儲存貯體的位置限制。

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

如果儲存貯體完全沒有位置限制 （ 如 的情況us-east-1)，則 GET_LOCATIONCONSTRAINT( )將傳回空字串。如果 欄位完全沒有值，您可以覆寫此行為並指定所需的值。

```
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' ).
```

現在，Bucket Location: assuming us-east-1如果 getbucketlocation()的結果未傳回位置，程式將會寫入 。

如果請求的值完全遺失，您可以要求 GET( ) 方法傳回特定結果，請參閱下列程式碼範例。

```
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'  ).
```

在此情況下，如果沒有位置限制， GET_LOCATIONCONSTRAINT( )將傳回 NopeNopeNope。

HAS_field( )

HAS_field( ) 方法是一種了解欄位是否具有值的方法。請參閱以下範例。

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

```
endif.
```

如果已知特定欄位一律具有值，則不會有任何HAS_field( )方法。

ASK_field( )

ASK_field( ) 方法會傳回 欄位的值，如果沒有 值，則引發例外狀況。這是處理多個欄位、解決邏輯問題，以及在任何欄位沒有值時採取不同方法的便利方法。

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

請注意， /AWS1/CX_RT_VALUE_MISSING 是靜態例外狀況，如果您選擇不擷取它，則會收到警告。

最佳實務

一般而言，您可以使用 GET_field( )方法，因為它將 null 字串視為空字串，並且是三個選項中最近似 ABAP 的字串。不過，它不會讓您輕鬆地區分欄位具有空白值和欄位沒有值的情況。如果您的商業邏輯取決於區分遺失資料與空白資料，則 HAS或ASK 方法可讓您處理這些案例。

陣列

陣列表示為物件的 ABAP 標準資料表。

JSON 陣列可以包含 null 值，例如下列陣列：[‘cat’, ‘dog’, null, ‘horse’]。這稱為稀疏陣列。它在 ABAP 中表示為物件參考的內部資料表，而 null值在資料表中表示為真正的 ABAP null值。逐一查看稀疏資料表時，您必須檢查null值，以避免存取null物件並取得CX_SY_REF_IS_INITIAL例外狀況。實際上，稀疏陣列在 服務中 AWS 很少見。

若要初始化 物件陣列，使用新的 ABAP 7.40 建構很方便。請考慮啟動已指派數個安全群組的 Amazon EC2 執行個體：

```
ao_ec2->runinstances(
    iv_imageid                  = lo_latest_ami->get_imageid( )
    iv_instancetype             = 't2.micro'
    iv_maxcount                 = 1
    iv_mincount                 = 1
```

```
    it_securitygroupids          = VALUE /aws1/
cl_ec2secgrpidstrlist_w=>tt_securitygroupidstringlist(
                                     ( NEW /aws1/
cl_ec2secgrpidstrlist_w( 'sg-12345678' ) )
                                     ( NEW /aws1/
cl_ec2secgrpidstrlist_w( 'sg-55555555' ) )
                                     ( NEW /aws1/
cl_ec2secgrpidstrlist_w( 'sg-99999999' ) )
                                   )
    iv_subnetid                   = ao_snet->get_subnetid( )
    it_tagspecifications          = make_tag_spec( 'instance' )
)
```

地圖

JSON 映射以 ABAP 表示，Hashed Tables因為每個資料表列只有兩個元件。

- KEY – 字串，為資料表UNIQUE KEY的 。
- VALUE – 包含 值的物件。

映射是極少數的案例之一，其中 AWS SDK 使用真正的結構，而不是類別。這是必要的，因為 ABAP 雜湊資料表不能將物件參考作為索引鍵欄位，而 AWS 映射索引鍵一律是非空字串。

更高層級的函數

上一節API 類別中所述的 會精確反映 AWS 服務 APIs並將這些 APIs為熟悉的 ABAP 類別。在某些情況下，開發套件也包含更高層級的函數，這些函數在 API 類別之上建置，以簡化特定操作。為了方便程式設計人員，包含更高層級的函數，且不會取代較低層級的 API 類別。

如果 SDK 包含模組的更高層級函數，則它們會包含在相同的傳輸中，並且可以透過稱為 的工廠類別存取/AWS1/CL_TLA_L2_FACTORY。工廠類別包含為模組建立各種更高層級用戶端的方法，這些用戶端會與 API 文件一起記錄於 API 的其餘部分。

適用於 SAP ABAP 的 AWS SDK features

適用於 SAP ABAP 的 AWS SDK 提供下列功能。

主題

- 程式設計組態
- 等待程式

- 分頁程式
- 重試行為
- 預先簽章者
- 跨帳戶 IAM 角色鏈結

程式設計組態

使用適用於 SAP ABAP 的 AWS SDK 的 /n/AWS1/IMG IMG 轉換，以及適用於適用於 SAP ABAP 的 AWS SDK 的自訂商業組態應用程式 - 適用於程式設計組態的 BTP 版本。

若要開始程式設計組態，請先使用 get_config( )命令擷取組態物件。

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

每個組態物件都會實作/AWS1/IF_RT_CONFIG介面，其中包含對應至 的 GETters 和 SETtersIMG。例如，預設區域可以覆寫。請參閱下列範例命令。

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

有些組態物件沒有IMG表示法，只能以程式設計方式設定，例如重試次數上限。請參閱下列範例命令。

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

的組態物件 AWS 服務 也可以包含未在 中表示的服務特定方法/aws1/if_rt_config。
例如，Amazon S3 可以使用foobucket.s3.region.amazonaws.com虛擬端點
或s3.region.amazonaws.com/foobucket路徑樣式來定址名為 foobucket的儲存貯體。您可以使用下列範例命令強制使用路徑樣式。

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

如需服務組態的詳細資訊，請參閱 適用於 SAP ABAP 的 AWS SDK – API 參考指南。

等待程式

使用非同步 AWS APIs 時，您需要等待特定資源可用，才能採取進一步動作。例如， 的 CREATETABLE( ) API 會立即 Amazon DynamoDB 回應資料表狀態 CREATING。只有在資料表的狀態變更為 之後，您才能啟動讀取或寫入操作ACTIVE。等待程式可讓您先確認 AWS 資源處於特定狀態，再對其執行動作。

等待者使用服務操作輪詢 AWS 資源的狀態，直到資源達到預期狀態，或直到確定資源未達到預期狀態為止。編寫程式碼以持續輪詢 AWS 資源可能非常耗時且容易出錯。等待程式會代表您負責執行輪詢，協助簡化此複雜性。

請參閱下列使用等待程式的 Amazon S3 範例。

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

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

" Wait until the newly created bucket becomes available
lo_s3->get_waiter( )->bucketexists(
    iv_max_wait_time = 200
    iv_bucket = |amzn-s3-demo-bucket|
).
```

- 在此範例中，Amazon S3 用戶端用於建立儲存貯體。已實作 `get_waiter()`命令來指定 的時間`bucketexists`。
- 您必須為每個等待程式指定 `iv_max_wait_time` 參數。它代表等待程式在完成之前必須等待的總時間量。在上述範例中，等待程式可以執行 200 秒。
- 您可能需要為必要參數提供額外的輸入。在上述範例中，`iv_bucket`參數需要 Amazon S3 儲存貯體名稱。
- `/AWS1/CX_RT_WAITER_FAILURE` 例外狀況表示等待程式超過 `iv_max_wait_time` 參數中指定的時間上限。
- `/AWS1/CX_RT_WAITER_TIMEOUT` 例外狀況表示等待程式已因未達到所需狀態而停止。

分頁程式

有些 AWS 服務 操作提供分頁回應。它們會分頁，以傳回每個回應的固定資料量。您需要使用字符或標記提出後續請求，才能擷取整個結果集。例如，Amazon S3 `ListObjectsV2` 操作一次最多傳回 1，000 個物件。您必須使用適當的字符提出後續請求，以取得下一頁的結果。

分頁是傳送連續請求的程序，以從前一個請求停止的位置挑選。分頁程式是 SDK for SAP ABAP 所提供結果的迭代器。您可以輕鬆使用分頁 APIs，而無需了解使用分頁字符的 API 基礎機制。

使用分頁程式

您可以使用傳回分頁器物件的 get_paginator()方法建立分頁器。分頁器物件會呼叫正在分頁的操作。分頁器物件接受要提供給基礎 API 的必要參數。此程序會傳回迭代器物件，可用於使用 has_next()和 get_next()方法逐一查看分頁結果。

- has_next() 會傳回布林值，指出呼叫的操作是否有更多的回應或頁面可用。
- get_next() 會傳回操作回應。

下列範例列出使用分頁器擷取的 S3 儲存貯體中的所有物件。

```
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.
```

重試行為

適用於 SAP ABAP 的 SDK 可讓您為因限流或暫時性錯誤而 AWS 服務 失敗的 請求設定重試次數上限。服務用戶端層級允許的重試次數，也就是 SDK 在失敗和引發例外狀況之前重試操作的次數，由服務組態物件中的 AV_MAX_ATTEMPTS 屬性指定。建立服務用戶端物件時，開發套件會將 AV_MAX_ATTEMPTS 屬性設定為預設值 3。服務組態物件可用來以程式設計方式將重試次數上限設定為所需的值。如需詳細資訊，請參閱下列範例。

```
" 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**
>
> 雖然組態物件 ABAP SDK 允許使用 /AWS1/IF_RT_CONFIG~SET_RETRY_MODE()方法設定重試模式，但 SDK 僅支援standard重試模式。如需詳細資訊，請參閱重試行為 AWS SDKs和工具參考指南。

預先簽章者

您可以使用預先簽章URLs，將時間限制的存取權授予某些 URL AWS 服務。預先簽章的 URL 可以在瀏覽器中輸入，或由程式用來執行服務操作。您可以多次使用此預先簽章 URL，直到到期日期和時間為止。如需詳細資訊，請參閱針對支援預先簽章者的服務使用 SAP ABAP 用戶端的預先簽章 URLs 開發套件將有一個特殊方法，稱為 GET_PRESIGNER() 來建立該服務的預先簽章者。然後呼叫預先簽章者的方法，其對應於 API 用戶端的方法，但它們傳回預先簽章的 URL，而不是實際執行操作。

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

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

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

跨帳戶 IAM 角色鏈結

跨帳戶 IAM 角色鏈結支援可透過來源設定檔組態，跨多個 AWS 帳戶無縫存取資源。此功能可讓您設定多個角色假設，其中一個設定檔擔任的角色接著擔任另一個角色，進而啟用複雜的跨帳戶存取模式。

如需詳細資訊，請參閱使用來源設定檔進行跨帳戶存取。

使用 SDK 建置產品

使用 的產品或 ABAP 附加元件 AWS 服務 可以增強和擴展 SDK 的功能。您可以建置這類產品以搭配 SDK 使用。

主題

- 設定產品 ID

設定產品 ID

建議您在產品或附加元件內建立工作階段時設定產品 ID。如需詳細資訊，請參閱下列範例。

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

產品 ID 只能包含不含空格或特殊字元的字母、數字和底線。您可以將其與產品的技術名稱或任何其他識別符進行比對。如果您開發多個產品或附加元件，則每個產品的產品 ID 都必須是唯一的。例如，Invoice Analyzer、Tax Calculator 和 Pricing Engine 產品IDs 可以是 `INVOICE_ANALYZER`、`TAX_CALCULATOR`和 `PRICING_ENGINE`。

將產品 ID 新增至工作階段可增強 AWS 每次服務呼叫時傳送至 的遙測。進行呼叫之物件的產品 ID 和命名空間包含在遙測中。使用此遙測， 支援 可以識別在您客戶遇到 SDK 問題時撥打電話的產品。它有助於釐清產品實際上正在進行呼叫，而不是客戶的程式碼。

自訂對 的 HTTP 請求 AWS

適用於 SAP ABAP 的 AWS SDK 會處理建立 HTTP 請求、傳送承載和接收回應的程序。您可以自訂 HTTP 請求的行為或內容，以符合您自己的 IT 需求。SDK `/AWS1/RT_EHN_HTTP_CLIENT`將增強功能點定義為集中位置，以增強 HTTP 通訊。增強功能 spot 支援將 HTTP 標頭新增至向 提出的請求 AWS。

實作增強功能

SAP 提供下列指示來實作增強功能：

- 傳統 ABAP
- BTP ABAP

篩選增強功能

增強點支援多個可同時作用中的實作。如果您需要確保增強功能僅在呼叫特定 AWS 服務或 API 動作時執行，您可以根據下列屬性篩選 BAdi 的執行：

- TLA - 服務的三個字母縮寫，以大寫字母表示。
- OPERATION - API 動作名稱。例如，從 S3 儲存貯體取得物件的操作是 GetObject。動作名稱區分大小寫，可能不完全符合 ABAP 方法名稱。

編碼增強功能

增強功能提供下列方法。

MODIFY_REQ_HEADERS

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

您可以在CT_HEADERS內部資料表中附加和修改標頭。我們不建議修改標頭，因為這會改變 AWS 服務使用的資料。您新增的任何標頭會被 AWS 服務忽略，但可由您的 IT 基礎設施處理，例如代理伺服器或其他中介軟體。

在計算身分驗證和遙測標頭之前呼叫增強點，因此增強功能無法修改這些項目。

以下是實作範例。

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

限制

適用於 SAP ABAP 的 AWS SDK 包含所有 的 SDK 模組 AWS 服務。其中一些模組可能會有限制，如此處所述。

- 依賴MQTT通訊協定繫結的模組，例如 iotevents將無法運作。 MQTT 不是 HTTP 型通訊協定，目前不受 支援 適用於 SAP ABAP 的 AWS SDK。

- 支援傳回事件串流的操作，但會等到收到整個串流，再將控制權傳回 ABAP。例如 Amazon Bedrock Agents 執行期`InvokeAgent`和 AWS Lambda `InvokeWithResponseStream`
- 由於基礎 ABAP 平台的限制，不支援接收事件串流的操作。例如 Amazon Q Business `Chat`和 Amazon Lex `StartConversation`

適用於 SAP ABAP 的 AWS SDK 具有下列功能限制。

- 目前尚不支援下列 Amazon S3 功能。
  - 多區域存取點
  - Amazon S3 用戶端加密

AWS 適用於 SAP ABAP 的 SDK - BTP 版本在開發人員預覽期間具有下列限制。

- 有些模組可能無法使用。
- 無法解除安裝。
- 更新頻率較低。

適用於 SAP ABAP 的 SDK 程式碼範例

本主題中的程式碼範例示範如何使用適用於 SAP ABAP 的 AWS 開發套件 AWS。

基本概念是程式碼範例，這些範例說明如何在服務內執行基本操作。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

案例是向您展示如何呼叫服務中的多個函數或與其他 AWS 服務組合來完成特定任務的程式碼範例。

有些服務包含其他範例類別，示範如何利用特定於服務的程式庫或函數。

服務

- 使用適用於 SAP ABAP 的 SDK 的 ACM 範例
- 使用適用於 SAP ABAP 的 SDK 的 API Gateway 範例
- 使用 SDK for SAP ABAP 的應用程式復原控制器範例
- 使用適用於 SAP ABAP 的 SDK 的 Aurora 範例
- 使用適用於 SAP ABAP 的 SDK 的 Auto Scaling 範例
- 使用適用於 SAP ABAP 的 SDK 的 Amazon Bedrock 執行時期範例
- 使用適用於 SAP ABAP 的 SDK 的 Amazon Bedrock 代理程式執行時期範例
- 使用適用於 SAP ABAP 的 SDK 的 CloudFront 範例
- 使用適用於 SAP ABAP 的 SDK 的 CloudWatch 範例
- 使用適用於 SAP ABAP 的 SDK 的 CloudWatch Logs 範例
- 使用適用於 SAP ABAP 的 SDK 的 Amazon Cognito 身分提供者範例
- 使用適用於 SAP ABAP 的 SDK 的 Amazon Comprehend 範例
- AWS Config 使用適用於 SAP ABAP 的 SDK 的範例
- AWS Control Tower 使用適用於 SAP ABAP 的 SDK 的範例
- 使用適用於 SAP ABAP 的 SDK 的 Firehose 範例
- 使用適用於 SAP ABAP 的 SDK 的 DynamoDB 範例
- 使用適用於 SAP ABAP 的 SDK 的 Amazon EC2 範例
- 使用適用於 SAP ABAP 的 SDK 的 Amazon ECR 範例
- 使用適用於 SAP ABAP 的 SDK 的 Amazon EMR 範例

- 使用適用於 SAP ABAP 的 SDK 的 EventBridge 排程器範例
- AWS Glue 使用適用於 SAP ABAP 的 SDK 的範例
- HealthImaging 範例使用適用於 SAP ABAP 的 SDK
- 使用適用於 SAP ABAP 的 SDK 的 HealthLake 範例
- 使用 SDK for SAP ABAP 的 IAM 範例
- AWS IoT SiteWise 使用適用於 SAP ABAP 的 SDK 的範例
- 使用適用於 SAP ABAP 的 SDK 的 Amazon Keyspaces 範例
- 使用適用於 SAP ABAP 的 SDK 的 Kinesis 範例
- AWS KMS 使用適用於 SAP ABAP 的 SDK 的範例
- 使用適用於 SAP ABAP 的 SDK 的 Lambda 範例
- 使用適用於 SAP ABAP 的 SDK 的組織範例
- 使用適用於 SAP ABAP 的 SDK 的 Amazon Pinpoint 範例
- 使用適用於 SAP ABAP 的 SDK 的 Amazon Pinpoint SMS 和語音 API 範例
- 使用適用於 SAP ABAP 的 SDK 的 Amazon Polly 範例
- 使用適用於 SAP ABAP 的 SDK 的 Amazon RDS 範例
- 使用適用於 SAP ABAP 的 SDK 的 Amazon Redshift 範例
- 使用適用於 SAP ABAP 的 SDK 的 Amazon Rekognition 範例
- 使用適用於 SAP ABAP 的 SDK 的 Amazon S3 範例
- 使用適用於 SAP ABAP 的 SDK 的 SageMaker AI 範例
- 使用適用於 SAP ABAP 的 SDK 的 Secrets Manager 範例
- 使用適用於 SAP ABAP 的 SDK 的 Amazon SES 範例
- 使用適用於 SAP ABAP 的 SDK 的 Amazon SES API v2 範例
- 使用適用於 SAP ABAP 的 SDK 的 Amazon SNS 範例
- 使用適用於 SAP ABAP 的 SDK 的 Amazon SQS 範例
- 使用適用於 SAP ABAP 的 SDK 的 Step Functions 範例
- 使用適用於 SAP ABAP 的 SDK 的 Systems Manager 範例
- 使用適用於 SAP ABAP 的 SDK 的 Amazon Textract 範例
- 使用適用於 SAP ABAP 的 SDK 的 Amazon Transcribe 範例
- 使用適用於 SAP ABAP 的 SDK 的 Amazon Translate 範例

使用適用於 SAP ABAP 的 SDK 的 ACM 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 ACM 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

`AddTagsToCertificate`

以下程式碼範例顯示如何使用 `AddTagsToCertificate`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

```
      CATCH /aws1/cx_acmtoomanytagsex.
        MESSAGE 'Too many tags for certificate.' TYPE 'I'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 AddTagsToCertificate。

`DeleteCertificate`

以下程式碼範例顯示如何使用 `DeleteCertificate`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteCertificate。

`DescribeCertificate`

以下程式碼範例顯示如何使用 `DescribeCertificate`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeCertificate。

`GetCertificate`

以下程式碼範例顯示如何使用 `GetCertificate`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_certificate_arn = 'arn:aws:acm:region:123456789012:certificate/
certificate-id'
```

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetCertificate。

`ImportCertificate`

以下程式碼範例顯示如何使用 `ImportCertificate`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " Only pass certificate chain if it's provided (it's optional)
        IF iv_certificate_chain IS NOT INITIAL.
          DATA(lo_result) = lo_acm->importcertificate(
            iv_certificate = iv_certificate
            iv_privatekey = iv_private_key
            iv_certificatechain = iv_certificate_chain
          ).
        ELSE.
          lo_result = lo_acm->importcertificate(
            iv_certificate = iv_certificate
            iv_privatekey = iv_private_key
          ).
        ENDIF.
        ov_certificate_arn = lo_result->get_certificatearn( ).
```

```
      MESSAGE 'Certificate imported successfully.' TYPE 'I'.
    CATCH /aws1/cx_acminvalidparameterex.
      MESSAGE 'Invalid parameter provided.' TYPE 'I'.
    CATCH /aws1/cx_acmlimitexceededex.
      MESSAGE 'Certificate limit exceeded.' TYPE 'I'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ImportCertificate。

`ListCertificates`

以下程式碼範例顯示如何使用 `ListCertificates`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
      oo_result = lo_acm->listcertificates(
        iv_maxitems = iv_max_items
        it_certificatestatuses = it_statuses
        io_includes = io_includes
      ).
      MESSAGE 'Certificates listed successfully.' TYPE 'I'.
    CATCH /aws1/cx_acminvalidargsex.
      MESSAGE 'Invalid arguments provided.' TYPE 'I'.
    CATCH /aws1/cx_acmvalidationex.
      MESSAGE 'Validation error occurred.' TYPE 'I'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListCertificates。

`ListTagsForCertificate`

以下程式碼範例顯示如何使用 `ListTagsForCertificate`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListTagsForCertificate。

`RemoveTagsFromCertificate`

以下程式碼範例顯示如何使用 `RemoveTagsFromCertificate`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 RemoveTagsFromCertificate。

`RequestCertificate`

以下程式碼範例顯示如何使用 `RequestCertificate`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_domain_name = 'example.com'
        " iv_validation_method = 'DNS' or 'EMAIL'
        DATA(lo_result) = lo_acm->requestcertificate(
          iv_domainname = iv_domain_name
          it_subjectalternativenames = COND #( WHEN it_alternate_domains IS NOT
 INITIAL
                                               THEN it_alternate_domains )
          iv_validationmethod = iv_validation_method
```

```
        ).
        ov_certificate_arn = lo_result->get_certificatearn( ).
        MESSAGE 'Certificate requested successfully.' TYPE 'I'.
      CATCH /aws1/cx_acminvalidparameterex.
        MESSAGE 'Invalid parameter provided.' TYPE 'I'.
      CATCH /aws1/cx_acmlimitexceededex.
        MESSAGE 'Certificate limit exceeded.' TYPE 'I'.
      CATCH /aws1/cx_acminvdomvationoptsex.
        MESSAGE 'Invalid domain validation options.' TYPE 'I'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 RequestCertificate。

`ResendValidationEmail`

以下程式碼範例顯示如何使用 `ResendValidationEmail`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_certificate_arn = 'arn:aws:acm:region:123456789012:certificate/
certificate-id'
        " iv_domain = 'example.com'
        " iv_validation_domain = 'example.com'
        lo_acm->resendvalidationemail(
          iv_certificatearn = iv_certificate_arn
          iv_domain = iv_domain
          iv_validationdomain = iv_validation_domain
        ).
        MESSAGE 'Validation email resent successfully.' TYPE 'I'.
      CATCH /aws1/cx_acminvalidarnex.
        MESSAGE 'The certificate ARN is not valid.' TYPE 'I'.
      CATCH /aws1/cx_acmresourcenotfoundex.
```

```
      MESSAGE 'Certificate not found.' TYPE 'I'.
    CATCH /aws1/cx_acminvalidstateex.
      MESSAGE 'Certificate is not in a valid state.' TYPE 'I'.
    CATCH /aws1/cx_acminvdomvationoptsex.
      MESSAGE 'Invalid domain validation options.' TYPE 'I'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ResendValidationEmail。

使用適用於 SAP ABAP 的 SDK 的 API Gateway 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 API Gateway 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

`CreateDeployment`

以下程式碼範例顯示如何使用 `CreateDeployment`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
```

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateDeployment。

`CreateResource`

以下程式碼範例顯示如何使用 `CreateResource`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateResource。

`CreateRestApi`

以下程式碼範例顯示如何使用 `CreateRestApi`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

```
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateRestApi。

`DeleteRestApi`

以下程式碼範例顯示如何使用 `DeleteRestApi`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteRestApi。

`GetResources`

以下程式碼範例顯示如何使用 `GetResources`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetResources。

`GetRestApis`

以下程式碼範例顯示如何使用 `GetRestApis`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetRestApis。

`PutIntegration`

以下程式碼範例顯示如何使用 `PutIntegration`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 PutIntegration。

`PutIntegrationResponse`

以下程式碼範例顯示如何使用 `PutIntegrationResponse`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 PutIntegrationResponse。

PutMethod

以下程式碼範例顯示如何使用 PutMethod。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 PutMethod。

PutMethodResponse

以下程式碼範例顯示如何使用 PutMethodResponse。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 PutMethodResponse。

使用 SDK for SAP ABAP 的應用程式復原控制器範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配應用程式復原控制器來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

GetRoutingControlState

以下程式碼範例顯示如何使用 GetRoutingControlState。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

    " Parse the comma-separated cluster endpoints
    " Expected format: "https://endpoint1.com|us-west-2,https://endpoint2.com|us-
east-1"
    SPLIT iv_cluster_endpoints AT ',' INTO TABLE lt_endpoints.

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

    " Try each endpoint in order
    LOOP AT lt_endpoints INTO lv_endpoint.
      TRY.
          " Parse endpoint and region from the format "url|region"
          DATA(lv_pos) = find( val = lv_endpoint sub = '|' ).
          IF lv_pos > 0.
            DATA(lv_url) = substring( val = lv_endpoint len = lv_pos ).
            lv_region = substring( val = lv_endpoint off = lv_pos + 1 ).
```

```
        ELSE.
          " If no region specified, use default
          lv_url = lv_endpoint.
          lv_region = 'us-east-1'.
        ENDIF.

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

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

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

        " If successful, return the result
        RETURN.

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

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

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetRoutingControlState。

`UpdateRoutingControlState`

以下程式碼範例顯示如何使用 UpdateRoutingControlState。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

    " Parse the comma-separated cluster endpoints
    " Expected format: "https://endpoint1.com|us-west-2,https://endpoint2.com|us-
east-1"
    SPLIT iv_cluster_endpoints AT ',' INTO TABLE lt_endpoints.

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

    " Try each endpoint in order
    LOOP AT lt_endpoints INTO lv_endpoint.
      TRY.
          " Parse endpoint and region from the format "url|region"
          DATA(lv_pos) = find( val = lv_endpoint sub = '|' ).
          IF lv_pos > 0.
            DATA(lv_url) = substring( val = lv_endpoint len = lv_pos ).
            lv_region = substring( val = lv_endpoint off = lv_pos + 1 ).
          ELSE.
```

```
          " If no region specified, use default
          lv_url = lv_endpoint.
          lv_region = 'us-east-1'.
        ENDIF.

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

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

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

        " If successful, return the result
        RETURN.

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

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

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 UpdateRoutingControlState。

使用適用於 SAP ABAP 的 SDK 的 Aurora 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 Aurora 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

`CreateDBClusterParameterGroup`

以下程式碼範例顯示如何使用 `CreateDBClusterParameterGroup`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    DATA(lo_output) = lo_rds->createdbclusterparamgroup(
      iv_dbclusterparamgroupname = iv_param_group_name
      iv_dbparametergroupfamily = iv_param_group_family
      iv_description = iv_description
    ).
```

```
      oo_result = lo_output->get_dbclusterparametergroup( ).
    CATCH /aws1/cx_rdsdbparmgralrexfault.
      " Re-raise exception - parameter group already exists
      RAISE EXCEPTION TYPE /aws1/cx_rdsdbparmgralrexfault.
    CATCH /aws1/cx_rdsdbprmgrquotaexcd00.
      " Re-raise exception - quota exceeded
      RAISE EXCEPTION TYPE /aws1/cx_rdsdbprmgrquotaexcd00.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateDBClusterParameterGroup。

`DeleteDBClusterParameterGroup`

以下程式碼範例顯示如何使用 `DeleteDBClusterParameterGroup`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteDBClusterParameterGroup。

DescribeDBClusterParameterGroups

以下程式碼範例顯示如何使用 DescribeDBClusterParameterGroups。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeDBClusterParameterGroups。

DescribeDBClusterParameters

以下程式碼範例顯示如何使用 DescribeDBClusterParameters。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
```

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

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

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

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

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeDBClusterParameters。

DescribeDBEngineVersions

以下程式碼範例顯示如何使用 DescribeDBEngineVersions。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeDBEngineVersions。

`DescribeOrderableDBInstanceOptions`

以下程式碼範例顯示如何使用 `DescribeOrderableDBInstanceOptions`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    " iv_engine        = 'mysql'
    " iv_engineversion = '8.0.35'
    TRY.
        oo_result = lo_rds->descrorderabledbinstoptions(
```

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeOrderableDBInstanceOptions。

`ModifyDBClusterParameterGroup`

以下程式碼範例顯示如何使用 `ModifyDBClusterParameterGroup`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ModifyDBClusterParameterGroup。

使用適用於 SAP ABAP 的 SDK 的 Auto Scaling 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 Auto Scaling 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

`CreateAutoScalingGroup`

以下程式碼範例顯示如何使用 `CreateAutoScalingGroup`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

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

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

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

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

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

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateAutoScalingGroup。

DeleteAutoScalingGroup

以下程式碼範例顯示如何使用 DeleteAutoScalingGroup。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

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

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

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

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteAutoScalingGroup。

`DescribeAutoScalingGroups`

以下程式碼範例顯示如何使用 `DescribeAutoScalingGroups`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

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

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

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

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

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

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeAutoScalingGroups。

DescribeAutoScalingInstances

以下程式碼範例顯示如何使用 DescribeAutoScalingInstances。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

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

       ot_output = lo_output->get_autoscalinginstances( ).

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

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeAutoScalingInstances。

DescribeScalingActivities

以下程式碼範例顯示如何使用 DescribeScalingActivities。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

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

    ot_output = lo_output->get_activities( ).

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

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeScalingActivities。

`DisableMetricsCollection`

以下程式碼範例顯示如何使用 `DisableMetricsCollection`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

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

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

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DisableMetricsCollection。

`EnableMetricsCollection`

以下程式碼範例顯示如何使用 `EnableMetricsCollection`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

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

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

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 EnableMetricsCollection。

SetDesiredCapacity

以下程式碼範例顯示如何使用 SetDesiredCapacity。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

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

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

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 SetDesiredCapacity。

`TerminateInstanceInAutoScalingGroup`

以下程式碼範例顯示如何使用 `TerminateInstanceInAutoScalingGroup`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

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

    oo_output = lo_output->get_activity( ).

    MESSAGE 'Instance terminated successfully' TYPE 'I'.

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 TerminateInstanceInAutoScalingGroup。

UpdateAutoScalingGroup

以下程式碼範例顯示如何使用 UpdateAutoScalingGroup。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

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

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

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 UpdateAutoScalingGroup。

使用適用於 SAP ABAP 的 SDK 的 Amazon Bedrock 執行時期範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 Amazon Bedrock 執行期來執行動作和實作常見案例。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- Anthropic Claude
- Stable Diffusion

Anthropic Claude

InvokeModel

下列程式碼範例示範如何使用調用模型 API，將文字訊息傳送至 Anthropic Claude。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

調用 Anthropic Claude 2 基礎模型以產生文字。此範例使用 /US2/CL_JSON 的功能，這些功能在某些 NetWeaver 版本上可能無法使用。

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

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

    "Leave ls_input-stop_sequences empty.
```

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

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

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

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

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

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

    ENDTRY.
```

調用 Anthropic Claude 2 基礎模型，以使用 L2 高階用戶端產生文字。

```
    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.
```

調用 Anthropic Claude 3 基礎模型，以使用 L2 高階用戶端產生文字。

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

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

    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 InvokeModel。

Stable Diffusion

InvokeModel

下列程式碼範例示範如何在 Amazon Bedrock 上調用 Stability.ai Stable Diffusion XL 以產生映像。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

使用 Stable Diffusion 建立映像。

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

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

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

    DATA(lv_json) = /ui2/cl_json=>serialize(
```

```
      data = ls_input
                pretty_name   = /ui2/cl_json=>pretty_mode-low_case ).

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

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

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

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

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

    ENDTRY.
```

調用 Stability.ai Stable Diffusion XL 基礎模型，使用 L2 高階用戶端產生映像。

```
    TRY.
        DATA(lo_bdr_l2_sd) = /aws1/
cl_bdr_l2_factory=>create_stable_diffusion_xl_1( lo_bdr ).
        " iv_prompt contains a prompt like 'Show me a picture of a unicorn reading
 an enterprise financial report'.
        DATA(lv_image) = lo_bdr_l2_sd->text_to_image( iv_prompt ).
      CATCH /aws1/cx_bdraccessdeniedex INTO DATA(lo_ex).
        WRITE / lo_ex->get_text( ).
        WRITE / |Don't forget to enable model access at https://
console.aws.amazon.com/bedrock/home?#/modelaccess|.

    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 InvokeModel。

使用適用於 SAP ABAP 的 SDK 的 Amazon Bedrock 代理程式執行時期範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 Amazon Bedrock Agents 執行期來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

InvokeAgent

以下程式碼範例顯示如何使用 InvokeAgent。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

```
        " the stream of events can possibly contain an exception
        " which will be raised to break the loop
        " catch /AWS1/CX_BDZACCESSDENIEDEX.
        " catch /AWS1/CX_BDZINTERNALSERVEREX.
        " catch /AWS1/CX_BDZMODELNOTREADYEX.
        " catch /AWS1/CX_BDZVALIDATIONEX.
        " catch /AWS1/CX_BDZTHROTTLINGEX.
        " catch /AWS1/CX_BDZDEPENDENCYFAILEDEX.
        " catch /AWS1/CX_BDZBADGATEWAYEX.
        " catch /AWS1/CX_BDZRESOURCENOTFOUNDEX.
        " catch /AWS1/CX_BDZSERVICEQUOTAEXCDEX.
        " catch /AWS1/CX_BDZCONFLICTEXCEPTION.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 InvokeAgent。

使用適用於 SAP ABAP 的 SDK 的 CloudFront 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 CloudFront 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

`ListDistributions`

以下程式碼範例顯示如何使用 `ListDistributions`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListDistributions。

`UpdateDistribution`

以下程式碼範例顯示如何使用 `UpdateDistribution`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

```
      " Create a new distribution config with the updated comment "
      " Since the config object is immutable, we need to create a new one with all
existing values "
      DATA(lo_new_config) = NEW /aws1/cl_fntdistributionconfig(
        iv_callerreference = lo_old_config->get_callerreference( )
        io_aliases = lo_old_config->get_aliases( )
        iv_defaultrootobject = lo_old_config->get_defaultrootobject( )
        io_origins = lo_old_config->get_origins( )
        io_origingroups = lo_old_config->get_origingroups( )
        io_defaultcachebehavior = lo_old_config->get_defaultcachebehavior( )
        io_cachebehaviors = lo_old_config->get_cachebehaviors( )
        io_customerrorresponses = lo_old_config->get_customerrorresponses( )
        iv_comment = iv_comment
        io_logging = lo_old_config->get_logging( )
        iv_priceclass = lo_old_config->get_priceclass( )
        iv_enabled = lo_old_config->get_enabled( )
        io_viewercertificate = lo_old_config->get_viewercertificate( )
        io_restrictions = lo_old_config->get_restrictions( )
        iv_webaclid = lo_old_config->get_webaclid( )
        iv_httpversion = lo_old_config->get_httpversion( )
        iv_isipv6enabled = lo_old_config->get_isipv6enabled( ) ).

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 UpdateDistribution。

使用適用於 SAP ABAP 的 SDK 的 CloudWatch 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 CloudWatch 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

案例是向您展示如何呼叫服務中的多個函數或與其他 AWS 服務組合來完成特定任務的程式碼範例。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作
- 案例

動作

`DeleteAlarms`

以下程式碼範例顯示如何使用 `DeleteAlarms`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 DeleteAlarms。

`DescribeAlarms`

以下程式碼範例顯示如何使用 `DescribeAlarms`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 DescribeAlarms。

`DescribeAlarmsForMetric`

以下程式碼範例顯示如何使用 `DescribeAlarmsForMetric`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeAlarmsForMetric。

`DisableAlarmActions`

以下程式碼範例顯示如何使用 `DisableAlarmActions`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    "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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 DisableAlarmActions。

`EnableAlarmActions`

以下程式碼範例顯示如何使用 `EnableAlarmActions`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    "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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 EnableAlarmActions。

`GetMetricStatistics`

以下程式碼範例顯示如何使用 `GetMetricStatistics`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetMetricStatistics。

`ListMetrics`

以下程式碼範例顯示如何使用 `ListMetrics`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    "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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 ListMetrics。

`PutMetricAlarm`

以下程式碼範例顯示如何使用 `PutMetricAlarm`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 PutMetricAlarm。

`PutMetricData`

以下程式碼範例顯示如何使用 PutMetricData。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

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

INSERT lo_metricdatum INTO TABLE lt_metricdata.

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

將一組資料放入 CloudWatch 指標。

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

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

INSERT lo_metricdatum INTO TABLE lt_metricdata.

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 PutMetricData。

案例

開始使用警示

以下程式碼範例顯示做法：

- 建立警示。
- 停用警示動作。
- 描述警示。
- 刪除警示。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

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

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

"Disable alarm actions."
TRY.
    lo_cwt->disablealarmactions(
      it_alarmnames               = lt_alarmnames ).
    MESSAGE 'Alarm actions disabled' TYPE 'I'.
  CATCH /aws1/cx_rt_service_generic INTO DATA(lo_disablealarm_exception).
```

```
        DATA(lv_disablealarm_error) = |"{ lo_disablealarm_exception->av_err_code }"
 - { lo_disablealarm_exception->av_err_msg }|.
        MESSAGE lv_disablealarm_error TYPE 'E'.
    ENDTRY.

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

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

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的下列主題。
  - DeleteAlarms
  - DescribeAlarms
  - DisableAlarmActions
  - PutMetricAlarm

使用適用於 SAP ABAP 的 SDK 的 CloudWatch Logs 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 CloudWatch Logs 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

GetQueryResults

以下程式碼範例顯示如何使用 GetQueryResults。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

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

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetQueryResults。

StartQuery

以下程式碼範例顯示如何使用 StartQuery。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 StartQuery。

使用適用於 SAP ABAP 的 SDK 的 Amazon Cognito 身分提供者範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 Amazon Cognito 身分提供者來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

`AdminInitiateAuth`

以下程式碼範例顯示如何使用 `AdminInitiateAuth`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

```
        ).

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

        oo_result = lo_cgp->admininitiateauth(
          iv_userpoolid = iv_user_pool_id
          iv_clientid = iv_client_id
          iv_authflow = 'ADMIN_USER_PASSWORD_AUTH'
          it_authparameters = lt_auth_params
        ).

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

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

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

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 AdminInitiateAuth。

AdminRespondToAuthChallenge

以下程式碼範例顯示如何使用 AdminRespondToAuthChallenge。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

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

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

        oo_auth_result = lo_result->get_authenticationresult( ).

        IF oo_auth_result IS BOUND.
          MESSAGE 'MFA challenge completed successfully.' TYPE 'I'.
        ELSE.
          " Another challenge might be required
```

```
        DATA(lv_next_challenge) = lo_result->get_challengename( ).
        MESSAGE |Additional challenge required: { lv_next_challenge }.| TYPE 'I'.
      ENDIF.

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

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

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 AdminRespondToAuthChallenge。

`AssociateSoftwareToken`

以下程式碼範例顯示如何使用 `AssociateSoftwareToken`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

      ov_secret_code = lo_result->get_secretcode( ).

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

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

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 AssociateSoftwareToken。

`ListUsers`

以下程式碼範例顯示如何使用 `ListUsers`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

    ot_users = lo_result->get_users( ).

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

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

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListUsers。

`VerifySoftwareToken`

以下程式碼範例顯示如何使用 `VerifySoftwareToken`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

    ov_status = lo_result->get_status( ).

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

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

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 VerifySoftwareToken。

使用適用於 SAP ABAP 的 SDK 的 Amazon Comprehend 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 Amazon Comprehend 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

`CreateDocumentClassifier`

以下程式碼範例顯示如何使用 `CreateDocumentClassifier`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

```
  CATCH /aws1/cx_cpdinternalserverex.
    MESSAGE 'Internal server error occurred.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateDocumentClassifier。

`DeleteDocumentClassifier`

以下程式碼範例顯示如何使用 `DeleteDocumentClassifier`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteDocumentClassifier。

DescribeDocumentClassificationJob

以下程式碼範例顯示如何使用 DescribeDocumentClassificationJob。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeDocumentClassificationJob。

DescribeDocumentClassifier

以下程式碼範例顯示如何使用 DescribeDocumentClassifier。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeDocumentClassifier。

`DescribeTopicsDetectionJob`

以下程式碼範例顯示如何使用 `DescribeTopicsDetectionJob`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
```

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeTopicsDetectionJob。

`DetectDominantLanguage`

以下程式碼範例顯示如何使用 `DetectDominantLanguage`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DetectDominantLanguage。

`DetectEntities`

以下程式碼範例顯示如何使用 `DetectEntities`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DetectEntities。

`DetectKeyPhrases`

以下程式碼範例顯示如何使用 `DetectKeyPhrases`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DetectKeyPhrases。

`DetectPiiEntities`

以下程式碼範例顯示如何使用 `DetectPiiEntities`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DetectPiiEntities。

`DetectSentiment`

以下程式碼範例顯示如何使用 `DetectSentiment`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_cpd->detectsentiment(
      iv_text = iv_text
      iv_languagecode = iv_language_code
    ).
    MESSAGE 'Sentiment detected.' TYPE 'I'.
  CATCH /aws1/cx_cpdtextsizelmtexcdex.
    MESSAGE 'Text size exceeds limit.' TYPE 'E'.
  CATCH /aws1/cx_cpdunsuppedlanguageex.
```

```
      MESSAGE 'Unsupported language.' TYPE 'E'.
    CATCH /aws1/cx_cpdinternalserverex.
      MESSAGE 'Internal server error occurred.' TYPE 'E'.
    CATCH /aws1/cx_cpdinvalidrequestex.
      MESSAGE 'Invalid request.' TYPE 'E'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DetectSentiment。

`DetectSyntax`

以下程式碼範例顯示如何使用 `DetectSyntax`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DetectSyntax。

`ListDocumentClassificationJobs`

以下程式碼範例顯示如何使用 `ListDocumentClassificationJobs`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListDocumentClassificationJobs。

`ListDocumentClassifiers`

以下程式碼範例顯示如何使用 `ListDocumentClassifiers`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListDocumentClassifiers。

`ListTopicsDetectionJobs`

以下程式碼範例顯示如何使用 `ListTopicsDetectionJobs`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListTopicsDetectionJobs。

`StartDocumentClassificationJob`

以下程式碼範例顯示如何使用 `StartDocumentClassificationJob`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

```
  CATCH /aws1/cx_cpdtoomanytagsex.
    MESSAGE 'Too many tags.' TYPE 'E'.
  CATCH /aws1/cx_cpdresrclimitexcdex.
    MESSAGE 'Resource limit exceeded.' TYPE 'E'.
  CATCH /aws1/cx_cpdinternalserverex.
    MESSAGE 'Internal server error occurred.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 StartDocumentClassificationJob。

`StartTopicsDetectionJob`

以下程式碼範例顯示如何使用 `StartTopicsDetectionJob`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_cpd->starttopicsdetectionjob(
      iv_jobname = iv_job_name
      io_inputdataconfig = NEW /aws1/cl_cpdinputdataconfig(
        iv_s3uri = iv_input_s3_uri
        iv_inputformat = iv_input_format
      )
      io_outputdataconfig = NEW /aws1/cl_cpdoutputdataconfig(
        iv_s3uri = iv_output_s3_uri
      )
      iv_dataaccessrolearn = iv_data_access_role_arn
    ).
    MESSAGE 'Topics detection job started.' TYPE 'I'.
  CATCH /aws1/cx_cpdinvalidrequestex.
    MESSAGE 'Invalid request.' TYPE 'E'.
  CATCH /aws1/cx_cpdtoomanyrequestsex.
    MESSAGE 'Too many requests.' TYPE 'E'.
```

```
    CATCH /aws1/cx_cpdkmskeyvalidationex.
      MESSAGE 'KMS key validation error.' TYPE 'E'.
    CATCH /aws1/cx_cpdtoomanytagsex.
      MESSAGE 'Too many tags.' TYPE 'E'.
    CATCH /aws1/cx_cpdresrclimitexcdex.
      MESSAGE 'Resource limit exceeded.' TYPE 'E'.
    CATCH /aws1/cx_cpdinternalserverex.
      MESSAGE 'Internal server error occurred.' TYPE 'E'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 StartTopicsDetectionJob。

AWS Config 使用適用於 SAP ABAP 的 SDK 的範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS 開發套件來執行動作和實作常見案例 AWS Config。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

`DeleteConfigRule`

以下程式碼範例顯示如何使用 `DeleteConfigRule`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteConfigRule。

`DescribeConfigRules`

以下程式碼範例顯示如何使用 `DescribeConfigRules`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeConfigRules。

`PutConfigRule`

以下程式碼範例顯示如何使用 `PutConfigRule`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    " Create a config rule for S3 bucket public read prohibition
    lo_cfs->putconfigrule(
      io_configrule = NEW /aws1/cl_cfsconfigrule(
        iv_configrulename = iv_rule_name
        iv_description = |S3 Public Read Prohibited Bucket Rule|
        io_scope = NEW /aws1/cl_cfsscope(
          it_complianceresourcetypes = VALUE /aws1/
cl_cfscplncresrctypes_w=>tt_complianceresourcetypes(
            ( NEW /aws1/cl_cfscplncresrctypes_w( |AWS::S3::Bucket| ) )
          )
        )
        io_source = NEW /aws1/cl_cfssource(
          iv_owner = |AWS|
          iv_sourceidentifier = |S3_BUCKET_PUBLIC_READ_PROHIBITED|
        )
        iv_inputparameters = '{}'
        iv_configrulestate = |ACTIVE|
      )
    ).
    MESSAGE 'Created AWS Config rule.' TYPE 'I'.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 PutConfigRule。

AWS Control Tower 使用適用於 SAP ABAP 的 SDK 的範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS 開發套件來執行動作和實作常見案例 AWS Control Tower。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

DisableBaseline

以下程式碼範例顯示如何使用 DisableBaseline。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

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

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

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

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

      " Wait 30 seconds
```

```
          WAIT UP TO 30 SECONDS.
        ENDDO.

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DisableBaseline。

`DisableControl`

以下程式碼範例顯示如何使用 `DisableControl`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

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

    " Wait for operation to complete
    DATA lv_status TYPE /aws1/cttcontrolopstatus.
    DO 100 TIMES.
      lv_status = get_control_operation(
```

```
        io_ctt = io_ctt
        iv_operation_id = lv_operation_id
      ).

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

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

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

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DisableControl。

`EnableBaseline`

以下程式碼範例顯示如何使用 `EnableBaseline`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

    " Add Identity Center baseline parameter if provided
    IF iv_identity_center_baseline IS NOT INITIAL.
      " Create a JSON document with the baseline ARN value
```

```
      DATA(lv_json) = |\{ "IdentityCenterEnabledBaselineArn":
"{ iv_identity_center_baseline }" \}|.
      DATA(lo_param) = NEW /aws1/cl_cttenbdbaselineparam(
        iv_key = 'IdentityCenterEnabledBaselineArn'
        io_value = /aws1/cl_rt_document=>from_json_str( lv_json )
      ).
      APPEND lo_param TO lt_parameters.
    ENDIF.

    " Enable the baseline
    DATA(lo_output) = io_ctt->enablebaseline(
      iv_baselineidentifier = iv_baseline_identifier
      iv_baselineversion    = iv_baseline_version
      iv_targetidentifier   = iv_target_identifier
      it_parameters         = lt_parameters
    ).

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

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

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

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

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

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 EnableBaseline。

`EnableControl`

以下程式碼範例顯示如何使用 `EnableControl`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

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

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

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

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

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

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 EnableControl。

`GetBaselineOperation`

以下程式碼範例顯示如何使用 `GetBaselineOperation`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetBaselineOperation。

`GetControlOperation`

以下程式碼範例顯示如何使用 `GetControlOperation`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetControlOperation。

`ListBaselines`

以下程式碼範例顯示如何使用 `ListBaselines`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
DATA lt_baselines TYPE /aws1/cl_cttbaselinesummary=>tt_baselines.
DATA lv_nexttoken TYPE /aws1/cttstring.

" List all baselines using pagination
DO.
  DATA(lo_output) = io_ctt->listbaselines(
    iv_nexttoken = lv_nexttoken
  ).

  APPEND LINES OF lo_output->get_baselines( ) TO lt_baselines.

  lv_nexttoken = lo_output->get_nexttoken( ).
  IF lv_nexttoken IS INITIAL.
    EXIT.
  ENDIF.
ENDDO.

ot_baselines = lt_baselines.
MESSAGE 'Listed baselines successfully.' TYPE 'I'.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListBaselines。

`ListEnabledBaselines`

以下程式碼範例顯示如何使用 `ListEnabledBaselines`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    DATA lt_enabled_baselines TYPE /aws1/
cl_cttenbdbaselinesumm=>tt_enabledbaselines.
    DATA lv_nexttoken TYPE /aws1/cttlstenbdbaselinesnex00.

    " List all enabled baselines using pagination
    DO.
      DATA(lo_output) = io_ctt->listenabledbaselines(
        iv_nexttoken = lv_nexttoken
      ).

      APPEND LINES OF lo_output->get_enabledbaselines( ) TO lt_enabled_baselines.

      lv_nexttoken = lo_output->get_nexttoken( ).
      IF lv_nexttoken IS INITIAL.
        EXIT.
      ENDIF.
    ENDDO.

    ot_enabled_baselines = lt_enabled_baselines.
    MESSAGE 'Listed enabled baselines successfully.' TYPE 'I'.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListEnabledBaselines。

ListEnabledControls

以下程式碼範例顯示如何使用 ListEnabledControls。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    DATA lt_enabled_controls TYPE /aws1/
cl_cttenabledcontrolsumm=>tt_enabledcontrols.
    DATA lv_nexttoken TYPE /aws1/cttstring.

    " List all enabled controls using pagination
    DO.
      DATA(lo_output) = io_ctt->listenabledcontrols(
        iv_targetidentifier = iv_target_identifier
        iv_nexttoken        = lv_nexttoken
      ).

      APPEND LINES OF lo_output->get_enabledcontrols( ) TO lt_enabled_controls.

      lv_nexttoken = lo_output->get_nexttoken( ).
      IF lv_nexttoken IS INITIAL.
        EXIT.
      ENDIF.
    ENDDO.

    ot_enabled_controls = lt_enabled_controls.
    MESSAGE 'Listed enabled controls successfully.' TYPE 'I'.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListEnabledControls。

ListLandingZones

以下程式碼範例顯示如何使用 ListLandingZones。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    DATA lt_landing_zones TYPE /aws1/
cl_cttlandingzonesummary=>tt_landingzonesummaries.
    DATA lv_nexttoken TYPE /aws1/cttstring.

    " List all landing zones using pagination
    DO.
      DATA(lo_output) = io_ctt->listlandingzones(
        iv_nexttoken = lv_nexttoken
      ).

      APPEND LINES OF lo_output->get_landingzones( ) TO lt_landing_zones.

      lv_nexttoken = lo_output->get_nexttoken( ).
      IF lv_nexttoken IS INITIAL.
        EXIT.
      ENDIF.
    ENDDO.

    ot_landing_zones = lt_landing_zones.
    MESSAGE 'Listed landing zones successfully.' TYPE 'I'.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListLandingZones。

`ResetEnabledBaseline`

以下程式碼範例顯示如何使用 `ResetEnabledBaseline`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
" Reset the enabled baseline
DATA(lo_output) = io_ctt->resetenabledbaseline(
  iv_enabledbaselineidentifier = iv_enabled_baseline_identifier
).

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

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

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

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

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

ov_operation_id = lv_operation_id.
MESSAGE 'Baseline reset successfully.' TYPE 'I'.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ResetEnabledBaseline。

使用適用於 SAP ABAP 的 SDK 的 Firehose 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 Firehose 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

`PutRecord`

以下程式碼範例顯示如何使用 `PutRecord`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    DATA(lo_record) = NEW /aws1/cl_frhrecord( iv_data = iv_data ).

    DATA(lo_result) = lo_frh->putrecord(
      iv_deliverystreamname = iv_deliv_stream_name
      io_record             = lo_record ).

    MESSAGE 'Record sent to Firehose delivery stream.' TYPE 'I'.
  CATCH /aws1/cx_frhresourcenotfoundex.
    MESSAGE 'Delivery stream not found.' TYPE 'E'.
  CATCH /aws1/cx_frhinvalidargumentex.
    MESSAGE 'Invalid argument provided.' TYPE 'E'.
  CATCH /aws1/cx_frhserviceunavailex.
    MESSAGE 'Service temporarily unavailable.' TYPE 'E'.
```

```
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 PutRecord。

`PutRecordBatch`

以下程式碼範例顯示如何使用 PutRecordBatch。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   TRY.
       DATA(lo_result) = lo_frh->putrecordbatch(
         iv_deliverystreamname = iv_deliv_stream_name
         it_records            = it_records ).

       DATA(lv_failed_count) = lo_result->get_failedputcount( ).

       IF lv_failed_count > 0.
         MESSAGE |{ lv_failed_count } records failed to send.| TYPE 'I'.
       ELSE.
         MESSAGE 'All records sent successfully to Firehose delivery stream.' TYPE
'I'.
       ENDIF.
     CATCH /aws1/cx_frhresourcenotfoundex.
       MESSAGE 'Delivery stream not found.' TYPE 'E'.
     CATCH /aws1/cx_frhinvalidargumentex.
       MESSAGE 'Invalid argument provided.' TYPE 'E'.
     CATCH /aws1/cx_frhserviceunavailex.
       MESSAGE 'Service temporarily unavailable.' TYPE 'E'.
   ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 PutRecordBatch。

使用適用於 SAP ABAP 的 SDK 的 DynamoDB 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 DynamoDB 來執行動作和實作常見案例。

基本概念是程式碼範例，這些範例說明如何在服務內執行基本操作。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 基本概念
- 動作

基本概念

了解基本概念

以下程式碼範例顯示做法：

- 建立可存放電影資料的資料表。
- 放入、取得和更新資料表中的單個電影。
- 將影片資料從範例 JSON 檔案寫入資料表。
- 查詢特定年份發表的電影。
- 掃描某個年份範圍內發表的電影。
- 從資料表刪除電影，然後刪除資料表。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    " Create an Amazon Dynamo DB table.

    TRY.
        DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
        DATA(lo_dyn) = /aws1/cl_dyn_factory=>create( lo_session ).
        DATA(lt_keyschema) = VALUE /aws1/cl_dynkeyschemaelement=>tt_keyschema(
          ( NEW /aws1/cl_dynkeyschemaelement( iv_attributename = 'year'
                                              iv_keytype = 'HASH' ) )
          ( NEW /aws1/cl_dynkeyschemaelement( iv_attributename = 'title'
                                              iv_keytype = 'RANGE' ) ) ).
        DATA(lt_attributedefinitions) = VALUE /aws1/
cl_dynattributedefn=>tt_attributedefinitions(
          ( NEW /aws1/cl_dynattributedefn( iv_attributename = 'year'
                                           iv_attributetype = 'N' ) )
          ( NEW /aws1/cl_dynattributedefn( iv_attributename = 'title'
                                           iv_attributetype = 'S' ) ) ).

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

    " Describe table
    TRY.
        DATA(lo_table) = lo_dyn->describetable( iv_tablename = iv_table_name ).
        DATA(lv_tablename) = lo_table->get_table( )->ask_tablename( ).
```

```
        MESSAGE 'The table name is ' && lv_tablename TYPE 'I'.
      CATCH /aws1/cx_dynresourcenotfoundex.
        MESSAGE 'The table does not exist' TYPE 'E'.
    ENDTRY.

    " Put items into the table.
    TRY.
        DATA(lo_resp_putitem) = lo_dyn->putitem(
          iv_tablename = iv_table_name
          it_item      = VALUE /aws1/
cl_dynattributevalue=>tt_putiteminputattributemap(
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
              key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s =
 'Jaws' ) ) )
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
              key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '1975' }| ) ) )
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
              key = 'rating' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '7.5' }| ) ) )
          ) ).
        lo_resp_putitem = lo_dyn->putitem(
          iv_tablename = iv_table_name
          it_item      = VALUE /aws1/
cl_dynattributevalue=>tt_putiteminputattributemap(
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
              key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s = 'Star
 Wars' ) ) )
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
              key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '1978' }| ) ) )
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
              key = 'rating' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '8.1' }| ) ) )
          ) ).
        lo_resp_putitem = lo_dyn->putitem(
          iv_tablename = iv_table_name
          it_item      = VALUE /aws1/
cl_dynattributevalue=>tt_putiteminputattributemap(
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
              key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s =
 'Speed' ) ) )
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
```

```
              key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '1994' }| ) ) )
            ( VALUE /aws1/cl_dynattributevalue=>ts_putiteminputattrmap_maprow(
              key = 'rating' value = NEW /aws1/cl_dynattributevalue( iv_n = |
{ '7.9' }| ) ) )
          ) ).
        " TYPE REF TO /AWSEX/CL_AWS1_dyn_PUT_ITEM_OUTPUT
        MESSAGE '3 rows inserted into DynamoDB Table' && iv_table_name TYPE 'I'.
      CATCH /aws1/cx_dyncondalcheckfaile00.
        MESSAGE 'A condition specified in the operation could not be evaluated.'
 TYPE 'E'.
      CATCH /aws1/cx_dynresourcenotfoundex.
        MESSAGE 'The table or index does not exist' TYPE 'E'.
      CATCH /aws1/cx_dyntransactconflictex.
        MESSAGE 'Another transaction is using the item' TYPE 'E'.
    ENDTRY.

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

    " Query item from table.
    TRY.
        DATA(lt_attributelist) = VALUE /aws1/
cl_dynattributevalue=>tt_attributevaluelist(
```

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

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

    " Update items from table.
    TRY.
        DATA(lt_attributeupdates) = VALUE /aws1/
cl_dynattrvalueupdate=>tt_attributeupdates(
          ( VALUE /aws1/cl_dynattrvalueupdate=>ts_attributeupdates_maprow(
```

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

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

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的下列主題。
  - BatchWriteItem
  - CreateTable
  - DeleteItem
  - DeleteTable

- DescribeTable
- GetItem
- PutItem
- Query
- Scan
- UpdateItem

動作

`CreateTable`

以下程式碼範例顯示如何使用 `CreateTable`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 CreateTable。

`DeleteItem`

以下程式碼範例顯示如何使用 `DeleteItem`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 DeleteItem。

`DeleteTable`

以下程式碼範例顯示如何使用 `DeleteTable`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
      lo_dyn->deletetable( iv_tablename = iv_table_name ).
      " Wait till the table is actually deleted.
      lo_dyn->get_waiter( )->tablenotexists(
        iv_max_wait_time = 200
        iv_tablename     = iv_table_name ).
      MESSAGE 'Table ' && iv_table_name && ' deleted.' TYPE 'I'.
    CATCH /aws1/cx_dynresourcenotfoundex.
      MESSAGE 'The table ' && iv_table_name && ' does not exist' TYPE 'E'.
    CATCH /aws1/cx_dynresourceinuseex.
      MESSAGE 'The table cannot be deleted since it is in use' TYPE 'E'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 DeleteTable。

`DescribeTable`

以下程式碼範例顯示如何使用 `DescribeTable`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的 AWS SDK API 參考》中的 DescribeTable。

`GetItem`

以下程式碼範例顯示如何使用 `GetItem`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 GetItem。

`ListTables`

以下程式碼範例顯示如何使用 `ListTables`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

```
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 ListTables。

PutItem

以下程式碼範例顯示如何使用 PutItem。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 PutItem。

Query

以下程式碼範例顯示如何使用 Query。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 Query。

Scan

以下程式碼範例顯示如何使用 Scan。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " Scan movies for rating greater than or equal to the rating specified
        DATA(lt_attributelist) = VALUE /aws1/
cl_dynattributevalue=>tt_attributevaluelist(
            ( NEW /aws1/cl_dynattributevalue( iv_n = |{ iv_rating }| ) ) ).
        DATA(lt_filter_conditions) = VALUE /aws1/
cl_dyncondition=>tt_filterconditionmap(
          ( VALUE /aws1/cl_dyncondition=>ts_filterconditionmap_maprow(
          key = 'rating'
          value = NEW /aws1/cl_dyncondition(
          it_attributevaluelist = lt_attributelist
          iv_comparisonoperator = |GE|
          ) ) ) ).
        oo_scan_result = lo_dyn->scan( iv_tablename = iv_table_name
          it_scanfilter = lt_filter_conditions ).
        DATA(lt_items) = oo_scan_result->get_items( ).
        LOOP AT lt_items INTO DATA(lo_item).
          " You can loop over to get individual attributes.
          DATA(lo_title) = lo_item[ key = 'title' ]-value.
          DATA(lo_year) = lo_item[ key = 'year' ]-value.
        ENDLOOP.
        DATA(lv_count) = oo_scan_result->get_count( ).
        MESSAGE 'Found ' && lv_count && ' items' TYPE 'I'.
      CATCH /aws1/cx_dynresourcenotfoundex.
        MESSAGE 'The table or index does not exist' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 Scan。

`UpdateItem`

以下程式碼範例顯示如何使用 `UpdateItem`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 UpdateItem。

使用適用於 SAP ABAP 的 SDK 的 Amazon EC2 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 Amazon EC2 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

`AllocateAddress`

以下程式碼範例顯示如何使用 `AllocateAddress`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 AllocateAddress。

`AssociateAddress`

以下程式碼範例顯示如何使用 `AssociateAddress`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 AssociateAddress。

`AuthorizeSecurityGroupIngress`

以下程式碼範例顯示如何使用 `AuthorizeSecurityGroupIngress`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    " Create IP permissions for SSH access (port 22)
    " iv_cidr_ip = '192.0.2.0/24'
    DATA lt_ip_permissions TYPE /aws1/cl_ec2ippermission=>tt_ippermissionlist.
    DATA(lo_ip_permission) = NEW /aws1/cl_ec2ippermission(
```

```
      iv_ipprotocol = 'tcp'
      iv_fromport = 22
      iv_toport = 22
      it_ipranges = VALUE /aws1/cl_ec2iprange=>tt_iprangelist(
        ( NEW /aws1/cl_ec2iprange( iv_cidrip = iv_cidr_ip ) )
      )
    ).
    APPEND lo_ip_permission TO lt_ip_permissions.

    TRY.
        oo_result = lo_ec2->authsecuritygroupingress(             " oo_result is
 returned for testing purposes. "
          iv_groupid = iv_group_id
          it_ippermissions = lt_ip_permissions ).
        MESSAGE 'Authorized ingress rule for security group.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 AuthorizeSecurityGroupIngress。

`CreateKeyPair`

以下程式碼範例顯示如何使用 `CreateKeyPair`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        oo_result = lo_ec2->createkeypair( iv_keyname = iv_key_name ).
              " oo_result is returned for testing purposes. "
        MESSAGE 'Amazon EC2 key pair created.' TYPE 'I'.
```

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

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 CreateKeyPair。

`CreateSecurityGroup`

以下程式碼範例顯示如何使用 `CreateSecurityGroup`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 CreateSecurityGroup。

`CreateVpc`

以下程式碼範例顯示如何使用 `CreateVpc`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    " iv_cidr_block = '10.0.0.0/16'
    TRY.
        oo_result = lo_ec2->createvpc( iv_cidrblock = iv_cidr_block ).            "
 oo_result is returned for testing purposes. "
        DATA(lv_vpc_id) = oo_result->get_vpc( )->get_vpcid( ).
        MESSAGE 'Created VPC.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateVpc。

`CreateVpcEndpoint`

以下程式碼範例顯示如何使用 `CreateVpcEndpoint`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    " iv_vpc_id = 'vpc-abc123'
    " iv_service_name = 'com.amazonaws.region.service'
    TRY.
        oo_result = lo_ec2->createvpcendpoint(             " oo_result is returned
 for testing purposes. "
```

```
          iv_vpcid = iv_vpc_id
          iv_servicename = iv_service_name
          it_routetableids = it_route_table_ids ).
        DATA(lv_vpc_endpoint_id) = oo_result->get_vpcendpoint( )-
>get_vpcendpointid( ).
        MESSAGE 'Created VPC endpoint.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateVpcEndpoint。

`DeleteKeyPair`

以下程式碼範例顯示如何使用 `DeleteKeyPair`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的 AWS SDK API 參考》中的 DeleteKeyPair。

DeleteSecurityGroup

以下程式碼範例顯示如何使用 DeleteSecurityGroup。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的 AWS SDK API 參考》中的 DeleteSecurityGroup。

DeleteVpc

以下程式碼範例顯示如何使用 DeleteVpc。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        lo_ec2->deletevpc( iv_vpcid = iv_vpc_id ).
```

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteVpc。

`DeleteVpcEndpoints`

以下程式碼範例顯示如何使用 `DeleteVpcEndpoints`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        lo_ec2->deletevpcendpoints( it_vpcendpointids = it_vpc_endpoint_ids ).
        MESSAGE 'Deleted VPC endpoint(s).' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteVpcEndpoints。

`DescribeAddresses`

以下程式碼範例顯示如何使用 `DescribeAddresses`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的 AWS SDK API 參考》中的 DescribeAddresses。

`DescribeAvailabilityZones`

以下程式碼範例顯示如何使用 `DescribeAvailabilityZones`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的 AWS SDK API 參考》中的 DescribeAvailabilityZones。

`DescribeImages`

以下程式碼範例顯示如何使用 `DescribeImages`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        oo_result = lo_ec2->describeimages( it_imageids = it_image_ids ).
  " oo_result is returned for testing purposes. "
        DATA(lt_images) = oo_result->get_images( ).
        MESSAGE 'Retrieved information about Amazon Machine Images (AMIs).' TYPE
 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeImages。

DescribeInstanceTypes

以下程式碼範例顯示如何使用 DescribeInstanceTypes。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    " Create filters for architecture and instance type patterns
    " iv_architecture = 'x86_64'
    DATA lt_filters TYPE /aws1/cl_ec2filter=>tt_filterlist.
    APPEND NEW /aws1/cl_ec2filter(
      iv_name = 'processor-info.supported-architecture'
      it_values = VALUE /aws1/cl_ec2valuestringlist_w=>tt_valuestringlist(
        ( NEW /aws1/cl_ec2valuestringlist_w( iv_architecture ) )
      )
    ) TO lt_filters.
    " Filter for instance type patterns like '*.micro', '*.small'
    APPEND NEW /aws1/cl_ec2filter(
      iv_name = 'instance-type'
      it_values = VALUE /aws1/cl_ec2valuestringlist_w=>tt_valuestringlist(
        ( NEW /aws1/cl_ec2valuestringlist_w( '*.micro' ) )
        ( NEW /aws1/cl_ec2valuestringlist_w( '*.small' ) )
      )
    ) TO lt_filters.

    TRY.
        oo_result = lo_ec2->describeinstancetypes( it_filters = lt_filters ).
      " oo_result is returned for testing purposes. "
        DATA(lt_instance_types) = oo_result->get_instancetypes( ).
        MESSAGE 'Retrieved information about EC2 instance types.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeInstanceTypes。

`DescribeInstances`

以下程式碼範例顯示如何使用 `DescribeInstances`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

        " Retrieving details of EC2 instances. "
        DATA: lv_istance_id    TYPE /aws1/ec2string,
              lv_status        TYPE /aws1/ec2instancestatename,
              lv_instance_type TYPE /aws1/ec2instancetype,
              lv_image_id      TYPE /aws1/ec2string.
        LOOP AT oo_result->get_reservations( ) INTO DATA(lo_reservation).
          LOOP AT lo_reservation->get_instances( ) INTO DATA(lo_instance).
            lv_istance_id = lo_instance->get_instanceid( ).
            lv_status = lo_instance->get_state( )->get_name( ).
            lv_instance_type = lo_instance->get_instancetype( ).
            lv_image_id = lo_instance->get_imageid( ).
          ENDLOOP.
        ENDLOOP.
        MESSAGE 'Retrieved information about EC2 instances.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的 AWS SDK API 參考》中的 DescribeInstances。

DescribeKeyPairs

以下程式碼範例顯示如何使用 DescribeKeyPairs。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的 AWS SDK API 參考》中的 DescribeKeyPairs。

DescribeRegions

以下程式碼範例顯示如何使用 DescribeRegions。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的 AWS SDK API 參考》中的 DescribeRegions。

`DescribeRouteTables`

以下程式碼範例顯示如何使用 `DescribeRouteTables`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    " Create filter for VPC ID
    " iv_vpc_id = 'vpc-abc123'
    DATA lt_filters TYPE /aws1/cl_ec2filter=>tt_filterlist.
    APPEND NEW /aws1/cl_ec2filter(
```

```
      iv_name = 'vpc-id'
      it_values = VALUE /aws1/cl_ec2valuestringlist_w=>tt_valuestringlist(
        ( NEW /aws1/cl_ec2valuestringlist_w( iv_vpc_id ) )
      )
    ) TO lt_filters.

    TRY.
        oo_result = lo_ec2->describeroutetables( it_filters = lt_filters ).
    " oo_result is returned for testing purposes. "
        DATA(lt_route_tables) = oo_result->get_routetables( ).
        MESSAGE 'Retrieved information about route tables.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeRouteTables。

`DescribeSecurityGroups`

以下程式碼範例顯示如何使用 `DescribeSecurityGroups`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的 AWS SDK API 參考》中的 DescribeSecurityGroups。

`MonitorInstances`

以下程式碼範例顯示如何使用 `MonitorInstances`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的 AWS SDK API 參考》中的 MonitorInstances。

`RebootInstances`

以下程式碼範例顯示如何使用 `RebootInstances`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的 AWS SDK API 參考》中的 RebootInstances。

`ReleaseAddress`

以下程式碼範例顯示如何使用 `ReleaseAddress`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的 AWS SDK API 參考》中的 ReleaseAddress。

`RunInstances`

以下程式碼範例顯示如何使用 `RunInstances`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    " Create tags for resource created during instance launch. "
    DATA lt_tagspecifications TYPE /aws1/
cl_ec2tagspecification=>tt_tagspecificationlist.
    DATA ls_tagspecifications LIKE LINE OF lt_tagspecifications.
    ls_tagspecifications = NEW /aws1/cl_ec2tagspecification(
      iv_resourcetype = 'instance'
      it_tags = VALUE /aws1/cl_ec2tag=>tt_taglist(
        ( NEW /aws1/cl_ec2tag( iv_key = 'Name' iv_value = iv_tag_value ) )
      ) ).
    APPEND ls_tagspecifications TO lt_tagspecifications.

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

```
          iv_imageid = iv_ami_id
          iv_instancetype = 't3.micro'
          iv_maxcount = 1
          iv_mincount = 1
          it_tagspecifications = lt_tagspecifications
          iv_subnetid = iv_subnet_id ).
        MESSAGE 'EC2 instance created.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的 AWS SDK API 參考》中的 RunInstances。

`StartInstances`

以下程式碼範例顯示如何使用 `StartInstances`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的 AWS SDK API 參考》中的 StartInstances。

`StopInstances`

以下程式碼範例顯示如何使用 `StopInstances`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的 AWS SDK API 參考》中的 StopInstances。

使用適用於 SAP ABAP 的 SDK 的 Amazon ECR 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 Amazon ECR 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

`CreateRepository`

以下程式碼範例顯示如何使用 `CreateRepository`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_repository_name = 'my-repository'
        oo_result = lo_ecr->createrepository(
          iv_repositoryname = iv_repository_name ).
        DATA(lv_repository_uri) = oo_result->get_repository( )-
>get_repositoryuri( ).
        MESSAGE |Repository created with URI: { lv_repository_uri }| TYPE 'I'.
      CATCH /aws1/cx_ecrrepositoryalrexex.
        " If repository already exists, retrieve it
        DATA lt_repo_names TYPE /aws1/
cl_ecrrepositorynamels00=>tt_repositorynamelist.
        APPEND NEW /aws1/cl_ecrrepositorynamels00( iv_value = iv_repository_name )
 TO lt_repo_names.
        DATA(lo_describe_result) = lo_ecr->describerepositories( it_repositorynames
 = lt_repo_names ).
        DATA(lt_repos) = lo_describe_result->get_repositories( ).
        IF lines( lt_repos ) > 0.
          READ TABLE lt_repos INDEX 1 INTO DATA(lo_repo).
```

```
        oo_result = NEW /aws1/cl_ecrcrerepositoryrsp( io_repository = lo_repo ).
        MESSAGE |Repository { iv_repository_name } already exists.| TYPE 'I'.
      ENDIF.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateRepository。

`DeleteRepository`

以下程式碼範例顯示如何使用 `DeleteRepository`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_repository_name = 'my-repository'
        lo_ecr->deleterepository(
          iv_repositoryname = iv_repository_name
          iv_force = abap_true ).
        MESSAGE |Repository { iv_repository_name } deleted.| TYPE 'I'.
      CATCH /aws1/cx_ecrrepositorynotfndex.
        MESSAGE 'Repository not found.' TYPE 'I'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteRepository。

`DescribeImages`

以下程式碼範例顯示如何使用 `DescribeImages`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   TRY.
       " iv_repository_name = 'my-repository'
       " it_image_ids = VALUE #( ( NEW /aws1/cl_ecrimageidentifier( iv_imagetag =
'latest' ) ) )
       IF it_image_ids IS NOT INITIAL.
         oo_result = lo_ecr->describeimages(
           iv_repositoryname = iv_repository_name
           it_imageids = it_image_ids ).
       ELSE.
         oo_result = lo_ecr->describeimages(
           iv_repositoryname = iv_repository_name ).
       ENDIF.
       DATA(lt_image_details) = oo_result->get_imagedetails( ).
       MESSAGE |Found { lines( lt_image_details ) } images in repository.| TYPE
'I'.
     CATCH /aws1/cx_ecrrepositorynotfndex.
       MESSAGE 'Repository not found.' TYPE 'I'.
     CATCH /aws1/cx_ecrimagenotfoundex.
       MESSAGE 'Image not found.' TYPE 'I'.
     CATCH /aws1/cx_ecrinvalidparameterex.
       MESSAGE 'Invalid parameter provided.' TYPE 'I'.
   ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeImages。

`DescribeRepositories`

以下程式碼範例顯示如何使用 `DescribeRepositories`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " it_repository_names = VALUE #( ( NEW /aws1/
cl_ecrrepositorynamels00( iv_value = 'my-repository' ) ) )
        oo_result = lo_ecr->describerepositories(
          it_repositorynames = it_repository_names ).
        DATA(lt_repositories) = oo_result->get_repositories( ).
        MESSAGE |Found { lines( lt_repositories ) } repositories.| TYPE 'I'.
      CATCH /aws1/cx_ecrrepositorynotfndex.
        MESSAGE 'Repository not found.' TYPE 'I'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeRepositories。

`GetAuthorizationToken`

以下程式碼範例顯示如何使用 `GetAuthorizationToken`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        oo_result = lo_ecr->getauthorizationtoken( ).
        DATA(lt_auth_data) = oo_result->get_authorizationdata( ).
        IF lines( lt_auth_data ) > 0.
```

```
        READ TABLE lt_auth_data INDEX 1 INTO DATA(lo_auth_data).
        DATA(lv_token) = lo_auth_data->get_authorizationtoken( ).
        MESSAGE 'Authorization token retrieved.' TYPE 'I'.
      ENDIF.
    CATCH /aws1/cx_ecrserverexception.
      MESSAGE 'Server exception occurred.' TYPE 'I'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetAuthorizationToken。

`GetRepositoryPolicy`

以下程式碼範例顯示如何使用 `GetRepositoryPolicy`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
      " iv_repository_name = 'my-repository'
      oo_result = lo_ecr->getrepositorypolicy(
        iv_repositoryname = iv_repository_name ).
      DATA(lv_policy_text) = oo_result->get_policytext( ).
      MESSAGE 'Repository policy retrieved.' TYPE 'I'.
    CATCH /aws1/cx_ecrrepositorynotfndex.
      MESSAGE 'Repository not found.' TYPE 'I'.
    CATCH /aws1/cx_ecrrepositoryplynot00.
      MESSAGE 'Repository policy not found.' TYPE 'I'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetRepositoryPolicy。

PutLifeCyclePolicy

以下程式碼範例顯示如何使用 PutLifeCyclePolicy。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_repository_name = 'my-repository'
        " iv_lifecycle_policy_text = '{"rules":
[{"rulePriority":1,"description":"Expire images older than 14 days",...}]}'
        lo_ecr->putlifecyclepolicy(
          iv_repositoryname = iv_repository_name
          iv_lifecyclepolicytext = iv_lifecycle_policy_text ).
        MESSAGE |Lifecycle policy set for repository { iv_repository_name }.| TYPE
 'I'.
      CATCH /aws1/cx_ecrrepositorynotfndex.
        MESSAGE 'Repository not found.' TYPE 'I'.
      CATCH /aws1/cx_ecrvalidationex.
        MESSAGE 'Invalid lifecycle policy format.' TYPE 'I'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 PutLifeCyclePolicy。

SetRepositoryPolicy

以下程式碼範例顯示如何使用 SetRepositoryPolicy。

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    " iv_repository_name = 'my-repository'
    " iv_policy_text = '{"Version":"2012-10-17",       "Statement":[...]}'
    lo_ecr->setrepositorypolicy(
      iv_repositoryname = iv_repository_name
      iv_policytext = iv_policy_text ).
    MESSAGE |Policy set for repository { iv_repository_name }.| TYPE 'I'.
  CATCH /aws1/cx_ecrrepositorynotfndex.
    MESSAGE 'Repository not found.' TYPE 'I'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 SetRepositoryPolicy。

使用適用於 SAP ABAP 的 SDK 的 Amazon EMR 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 Amazon EMR 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

AddJobFlowSteps

以下程式碼範例顯示如何使用 AddJobFlowSteps。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    " Build args list for Spark submit
    DATA lt_args TYPE /aws1/cl_emrxmlstringlist_w=>tt_xmlstringlist.
    APPEND NEW /aws1/cl_emrxmlstringlist_w( 'spark-submit' ) TO lt_args.
    APPEND NEW /aws1/cl_emrxmlstringlist_w( '--deploy-mode' ) TO lt_args.
    APPEND NEW /aws1/cl_emrxmlstringlist_w( 'cluster' ) TO lt_args.
    APPEND NEW /aws1/cl_emrxmlstringlist_w( iv_script_uri ) TO lt_args.
    APPEND LINES OF it_script_args TO lt_args.

    " Create step configuration
    DATA(lo_hadoop_jar_step) = NEW /aws1/cl_emrhadoopjarstepcfg(
      iv_jar = 'command-runner.jar'
      it_args = lt_args
    ).

    DATA(lo_step_config) = NEW /aws1/cl_emrstepconfig(
      iv_name = iv_name
      iv_actiononfailure = 'CONTINUE'
      io_hadoopjarstep = lo_hadoop_jar_step
    ).

    DATA lt_steps TYPE /aws1/cl_emrstepconfig=>tt_stepconfiglist.
    APPEND lo_step_config TO lt_steps.

    DATA(lo_result) = lo_emr->addjobflowsteps(
      iv_jobflowid = iv_cluster_id
      it_steps = lt_steps
    ).
```

```
    " Get first step ID
    DATA(lt_step_ids) = lo_result->get_stepids( ).
    READ TABLE lt_step_ids INDEX 1 INTO DATA(lo_step_id_obj).
    IF sy-subrc = 0.
      ov_step_id = lo_step_id_obj->get_value( ).
      MESSAGE |Step added with ID { ov_step_id }| TYPE 'I'.
    ENDIF.
  CATCH /aws1/cx_emrinternalservererr INTO DATA(lo_internal_error).
    DATA(lv_error) = lo_internal_error->if_message~get_text( ).
    MESSAGE lv_error TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 AddJobFlowSteps。

`DescribeCluster`

以下程式碼範例顯示如何使用 `DescribeCluster`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_emr->describecluster(
      iv_clusterid = iv_cluster_id
    ).
    DATA(lo_cluster) = oo_result->get_cluster( ).
    DATA(lv_cluster_name) = lo_cluster->get_name( ).
    MESSAGE |Retrieved cluster information for { lv_cluster_name }| TYPE 'I'.
  CATCH /aws1/cx_emrinternalserverex INTO DATA(lo_internal_error).
    DATA(lv_error) = lo_internal_error->if_message~get_text( ).
    MESSAGE lv_error TYPE 'E'.
  CATCH /aws1/cx_emrinvalidrequestex INTO DATA(lo_invalid_error).
    lv_error = lo_invalid_error->if_message~get_text( ).
```

```
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeCluster。

`DescribeStep`

以下程式碼範例顯示如何使用 `DescribeStep`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        oo_result = lo_emr->describestep(
          iv_clusterid = iv_cluster_id
          iv_stepid = iv_step_id
        ).
        DATA(lo_step) = oo_result->get_step( ).
        DATA(lv_step_name) = lo_step->get_name( ).
        MESSAGE |Retrieved step information for { lv_step_name }| TYPE 'I'.
      CATCH /aws1/cx_emrinternalserverex INTO DATA(lo_internal_error).
        DATA(lv_error) = lo_internal_error->if_message~get_text( ).
        MESSAGE lv_error TYPE 'E'.
      CATCH /aws1/cx_emrinvalidrequestex INTO DATA(lo_invalid_error).
        lv_error = lo_invalid_error->if_message~get_text( ).
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeStep。

`ListSteps`

以下程式碼範例顯示如何使用 `ListSteps`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_emr->liststeps(
      iv_clusterid = iv_cluster_id
    ).
    DATA(lt_steps) = oo_result->get_steps( ).
    DATA(lv_step_count) = lines( lt_steps ).
    MESSAGE |Retrieved { lv_step_count } steps for cluster| TYPE 'I'.
  CATCH /aws1/cx_emrinternalserverex INTO DATA(lo_internal_error).
    DATA(lv_error) = lo_internal_error->if_message~get_text( ).
    MESSAGE lv_error TYPE 'E'.
  CATCH /aws1/cx_emrinvalidrequestex INTO DATA(lo_invalid_error).
    lv_error = lo_invalid_error->if_message~get_text( ).
    MESSAGE lv_error TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListSteps。

`RunJobFlow`

以下程式碼範例顯示如何使用 `RunJobFlow`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
```

```
      " Create instances configuration
      DATA(lo_instances) = NEW /aws1/cl_emrjobflowinstsconfig(
        iv_masterinstancetype = 'm5.xlarge'
        iv_slaveinstancetype = 'm5.xlarge'
        iv_instancecount = 3
        iv_keepjobflowalivewhennos00 = iv_keep_alive
        iv_emrmanagedmastersecgroup = iv_primary_sec_grp
        iv_emrmanagedslavesecgroup = iv_secondary_sec_grp
      ).

      DATA(lo_result) = lo_emr->runjobflow(
        iv_name = iv_name
        iv_loguri = iv_log_uri
        iv_releaselabel = 'emr-5.30.1'
        io_instances = lo_instances
        it_steps = it_steps
        it_applications = it_applications
        iv_jobflowrole = iv_job_flow_role
        iv_servicerole = iv_service_role
        iv_ebsrootvolumesize = 10
        iv_visibletoallusers = abap_true
      ).

      ov_cluster_id = lo_result->get_jobflowid( ).
      MESSAGE 'EMR cluster created successfully.' TYPE 'I'.
    CATCH /aws1/cx_emrinternalservererr INTO DATA(lo_internal_error).
      DATA(lv_error) = lo_internal_error->if_message~get_text( ).
      MESSAGE lv_error TYPE 'E'.
    CATCH /aws1/cx_emrclientexc INTO DATA(lo_client_error).
      lv_error = lo_client_error->if_message~get_text( ).
      MESSAGE lv_error TYPE 'E'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 RunJobFlow。

TerminateJobFlows

以下程式碼範例顯示如何使用 TerminateJobFlows。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    DATA lt_cluster_ids TYPE /aws1/cl_emrxmlstringlist_w=>tt_xmlstringlist.
    APPEND NEW /aws1/cl_emrxmlstringlist_w( iv_cluster_id ) TO lt_cluster_ids.

    lo_emr->terminatejobflows(
      it_jobflowids = lt_cluster_ids
    ).
    MESSAGE 'EMR cluster terminated successfully.' TYPE 'I'.
  CATCH /aws1/cx_emrinternalservererr INTO DATA(lo_internal_error).
    DATA(lv_error) = lo_internal_error->if_message~get_text( ).
    MESSAGE lv_error TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 TerminateJobFlows。

使用適用於 SAP ABAP 的 SDK 的 EventBridge 排程器範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 EventBridge 排程器來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

`CreateSchedule`

以下程式碼範例顯示如何使用 `CreateSchedule`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    " Constants for time calculations
    DATA lv_start_date TYPE /aws1/scdstartdate.
    DATA lv_end_date TYPE /aws1/scdenddate.
    DATA lv_start_timestamp TYPE timestamp.
    DATA lv_end_timestamp TYPE timestamp.
    DATA lv_hours_to_run TYPE i VALUE 1.

    " Get current timestamp
    GET TIME STAMP FIELD lv_start_timestamp.

    " Add 1 hour to the current timestamp using CL_ABAP_TSTMP
    lv_end_timestamp = cl_abap_tstmp=>add(
      tstmp = lv_start_timestamp
      secs = lv_hours_to_run * 3600 ).

    " Convert timestamps to decimal format for AWS API
    lv_start_date = lv_start_timestamp.
    lv_end_date = lv_end_timestamp.

    " Prepare flexible time window configuration
    DATA lo_flexible_time_window TYPE REF TO /aws1/cl_scdflexibletimewindow.
    IF iv_use_flexible_time_win = abap_true.
      " iv_use_flexible_time_win = ABAP_TRUE
      " Example: Set MaximumWindowInMinutes to 10 for flexible window
      lo_flexible_time_window = NEW /aws1/cl_scdflexibletimewindow(
        iv_mode = 'FLEXIBLE'
        iv_maximumwindowinminutes = 10 ).
```

```
        ELSE.
          lo_flexible_time_window = NEW /aws1/cl_scdflexibletimewindow(
            iv_mode = 'OFF' ).
        ENDIF.

        " Prepare target configuration
        " Example iv_target_arn = 'arn:aws:sqs:us-east-1:123456789012:my-queue'
        " Example iv_role_arn = 'arn:aws:iam::123456789012:role/SchedulerRole'
        " Example iv_input = '{"message": "Hello from EventBridge Scheduler"}'
        DATA(lo_target) = NEW /aws1/cl_scdtarget(
          iv_arn = iv_target_arn
          iv_rolearn = iv_role_arn
          iv_input = iv_input ).

        " Set action after completion if needed
        DATA lv_action_after_completion TYPE /aws1/scdactionaftercompletion.
        IF iv_delete_after_completion = abap_true.
          " iv_delete_after_completion = ABAP_TRUE
          lv_action_after_completion = 'DELETE'.
        ELSE.
          lv_action_after_completion = 'NONE'.
        ENDIF.

        " Create the schedule
        " Example iv_name = 'my-schedule'
        " Example iv_schedule_expression = 'rate(15 minutes)'
        " Example iv_schedule_group_name = 'my-schedule-group'
        DATA(lo_result) = lo_scd->createschedule(
          iv_name = iv_name
          iv_scheduleexpression = iv_schedule_expression
          iv_groupname = iv_schedule_group_name
          io_target = lo_target
          io_flexibletimewindow = lo_flexible_time_window
          iv_startdate = lv_start_date
          iv_enddate = lv_end_date
          iv_actionaftercompletion = lv_action_after_completion ).

        ov_schedule_arn = lo_result->get_schedulearn( ).
        MESSAGE 'Schedule created successfully.' TYPE 'I'.

      CATCH /aws1/cx_scdconflictexception INTO DATA(lo_conflict_ex).
        DATA(lv_error) = |Conflict creating schedule: { lo_conflict_ex-
>if_message~get_text( ) }|.
        MESSAGE lv_error TYPE 'I'.
```

```
      CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_ex).
        DATA(lv_generic_error) = |Error creating schedule: { lo_generic_ex-
>if_message~get_text( ) }|.
        MESSAGE lv_generic_error TYPE 'I'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateSchedule。

`CreateScheduleGroup`

以下程式碼範例顯示如何使用 `CreateScheduleGroup`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " Example iv_name = 'my-schedule-group'
        DATA(lo_result) = lo_scd->createschedulegroup(
          iv_name = iv_name ).

        ov_schedule_group_arn = lo_result->get_schedulegrouparn( ).
        MESSAGE 'Schedule group created successfully.' TYPE 'I'.

      CATCH /aws1/cx_scdconflictexception INTO DATA(lo_conflict_ex).
        DATA(lv_error) = |Conflict creating schedule group: { lo_conflict_ex-
>if_message~get_text( ) }|.
        MESSAGE lv_error TYPE 'I'.
      CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_ex).
        DATA(lv_generic_error) = |Error creating schedule group: { lo_generic_ex-
>if_message~get_text( ) }|.
        MESSAGE lv_generic_error TYPE 'I'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateScheduleGroup。

`DeleteSchedule`

以下程式碼範例顯示如何使用 `DeleteSchedule`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " Example iv_name = 'my-schedule'
        " Example iv_schedule_group_name = 'my-schedule-group'
        lo_scd->deleteschedule(
          iv_name = iv_name
          iv_groupname = iv_schedule_group_name ).
        MESSAGE 'Schedule deleted successfully.' TYPE 'I'.

      CATCH /aws1/cx_scdresourcenotfoundex INTO DATA(lo_not_found_ex).
        DATA(lv_error) = |Schedule not found: { lo_not_found_ex-
>if_message~get_text( ) }|.
        MESSAGE lv_error TYPE 'I'.
      CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_ex).
        DATA(lv_generic_error) = |Error deleting schedule: { lo_generic_ex-
>if_message~get_text( ) }|.
        MESSAGE lv_generic_error TYPE 'I'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteSchedule。

`DeleteScheduleGroup`

以下程式碼範例顯示如何使用 `DeleteScheduleGroup`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " Example iv_name = 'my-schedule-group'
        lo_scd->deleteschedulegroup(
          iv_name = iv_name ).
        MESSAGE 'Schedule group deleted successfully.' TYPE 'I'.

      CATCH /aws1/cx_scdresourcenotfoundex INTO DATA(lo_not_found_ex).
        DATA(lv_error) = |Schedule group not found: { lo_not_found_ex-
>if_message~get_text( ) }|.
        MESSAGE lv_error TYPE 'I'.
      CATCH /aws1/cx_rt_generic INTO DATA(lo_generic_ex).
        DATA(lv_generic_error) = |Error deleting schedule group: { lo_generic_ex-
>if_message~get_text( ) }|.
        MESSAGE lv_generic_error TYPE 'I'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteScheduleGroup。

AWS Glue 使用適用於 SAP ABAP 的 SDK 的範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS 開發套件來執行動作和實作常見案例 AWS Glue。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

`CreateCrawler`

以下程式碼範例顯示如何使用 `CreateCrawler`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    " iv_crawler_name = 'my-crawler'
    " iv_role_arn = 'arn:aws:iam::123456789012:role/AWSGlueServiceRole-Test'
    " iv_database_name = 'my-database'
    " iv_table_prefix = 'test_'
    " iv_s3_target = 's3://example-bucket/data/'

    DATA(lt_s3_targets) = VALUE /aws1/cl_glus3target=>tt_s3targetlist(
      ( NEW /aws1/cl_glus3target( iv_path = iv_s3_target ) ) ).

    DATA(lo_targets) = NEW /aws1/cl_glucrawlertargets(
      it_s3targets = lt_s3_targets ).

    lo_glu->createcrawler(
      iv_name = iv_crawler_name
      iv_role = iv_role_arn
      iv_databasename = iv_database_name
      iv_tableprefix = iv_table_prefix
      io_targets = lo_targets ).
    MESSAGE 'Crawler created successfully.' TYPE 'I'.
  CATCH /aws1/cx_glualreadyexistsex.
    MESSAGE 'Crawler already exists.' TYPE 'E'.
  CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
    DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
    MESSAGE lv_invalid_error TYPE 'E'.
  CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
    DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
    MESSAGE lv_timeout_error TYPE 'E'.
```

```
    CATCH /aws1/cx_gluresrcnumlmtexcdex INTO DATA(lo_limit_ex).
      DATA(lv_limit_error) = lo_limit_ex->if_message~get_longtext( ).
      MESSAGE lv_limit_error TYPE 'E'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateCrawler。

`CreateJob`

以下程式碼範例顯示如何使用 `CreateJob`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
      " iv_job_name = 'my-etl-job'
      " iv_description = 'ETL job for data transformation'
      " iv_role_arn = 'arn:aws:iam::123456789012:role/AWSGlueServiceRole-Test'
      " iv_script_location = 's3://example-bucket/scripts/my-script.py'

      DATA(lo_command) = NEW /aws1/cl_glujobcommand(
        iv_name = 'glueetl'
        iv_scriptlocation = iv_script_location
        iv_pythonversion = '3' ).

      lo_glu->createjob(
        iv_name = iv_job_name
        iv_description = iv_description
        iv_role = iv_role_arn
        io_command = lo_command
        iv_glueversion = '3.0' ).
      MESSAGE 'Job created successfully.' TYPE 'I'.
    CATCH /aws1/cx_glualreadyexistsex.
      MESSAGE 'Job already exists.' TYPE 'E'.
    CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
```

```
      DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
      MESSAGE lv_invalid_error TYPE 'E'.
    CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
      DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
      MESSAGE lv_internal_error TYPE 'E'.
    CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
      DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
      MESSAGE lv_timeout_error TYPE 'E'.
    CATCH /aws1/cx_gluresrcnumlmtexcdex INTO DATA(lo_limit_ex).
      DATA(lv_limit_error) = lo_limit_ex->if_message~get_longtext( ).
      MESSAGE lv_limit_error TYPE 'E'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateJob。

`DeleteCrawler`

以下程式碼範例顯示如何使用 `DeleteCrawler`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
      " iv_crawler_name = 'my-crawler'
      lo_glu->deletecrawler( iv_name = iv_crawler_name ).
      MESSAGE 'Crawler deleted successfully.' TYPE 'I'.
    CATCH /aws1/cx_glucrawlerrunningex.
      MESSAGE 'Crawler is currently running.' TYPE 'E'.
    CATCH /aws1/cx_gluentitynotfoundex.
      MESSAGE 'Crawler does not exist.' TYPE 'E'.
    CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
      DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
      MESSAGE lv_timeout_error TYPE 'E'.
    CATCH /aws1/cx_gluschdrtransingex.
      MESSAGE 'Scheduler is transitioning.' TYPE 'E'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteCrawler。

`DeleteDatabase`

以下程式碼範例顯示如何使用 `DeleteDatabase`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    " iv_database_name = 'my-database'
    lo_glu->deletedatabase( iv_name = iv_database_name ).
    MESSAGE 'Database deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_gluentitynotfoundex.
    MESSAGE 'Database does not exist.' TYPE 'E'.
  CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
    DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
    MESSAGE lv_invalid_error TYPE 'E'.
  CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
    DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
    MESSAGE lv_internal_error TYPE 'E'.
  CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
    DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
    MESSAGE lv_timeout_error TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteDatabase。

`DeleteJob`

以下程式碼範例顯示如何使用 `DeleteJob`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    " iv_job_name = 'my-etl-job'
    lo_glu->deletejob( iv_jobname = iv_job_name ).
    MESSAGE 'Job deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
    DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
    MESSAGE lv_invalid_error TYPE 'E'.
  CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
    DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
    MESSAGE lv_internal_error TYPE 'E'.
  CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
    DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
    MESSAGE lv_timeout_error TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteJob。

`DeleteTable`

以下程式碼範例顯示如何使用 `DeleteTable`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
```

```
      " iv_database_name = 'my-database'
      " iv_table_name = 'my-table'
      lo_glu->deletetable(
        iv_databasename = iv_database_name
        iv_name = iv_table_name ).
      MESSAGE 'Table deleted successfully.' TYPE 'I'.
    CATCH /aws1/cx_gluentitynotfoundex.
      MESSAGE 'Table or database does not exist.' TYPE 'E'.
    CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
      DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
      MESSAGE lv_invalid_error TYPE 'E'.
    CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
      DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
      MESSAGE lv_internal_error TYPE 'E'.
    CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
      DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
      MESSAGE lv_timeout_error TYPE 'E'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 DeleteTable。

GetCrawler

以下程式碼範例顯示如何使用 GetCrawler。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
      " iv_crawler_name = 'my-crawler'
      oo_result = lo_glu->getcrawler( iv_name = iv_crawler_name ).
      DATA(lo_crawler) = oo_result->get_crawler( ).
      MESSAGE 'Crawler information retrieved.' TYPE 'I'.
    CATCH /aws1/cx_gluentitynotfoundex.
      MESSAGE 'Crawler does not exist.' TYPE 'I'.
```

```
  CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
    DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
    MESSAGE lv_timeout_error TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetCrawler。

`GetDatabase`

以下程式碼範例顯示如何使用 `GetDatabase`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    " iv_database_name = 'my-database'
    oo_result = lo_glu->getdatabase( iv_name = iv_database_name ).
    DATA(lo_database) = oo_result->get_database( ).
    MESSAGE 'Database information retrieved.' TYPE 'I'.
  CATCH /aws1/cx_gluentitynotfoundex.
    MESSAGE 'Database does not exist.' TYPE 'E'.
  CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
    DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
    MESSAGE lv_invalid_error TYPE 'E'.
  CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
    DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
    MESSAGE lv_internal_error TYPE 'E'.
  CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
    DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
    MESSAGE lv_timeout_error TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetDatabase。

GetJobRun

以下程式碼範例顯示如何使用 GetJobRun。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    " iv_job_name = 'my-etl-job'
    " iv_run_id = 'jr_abcd1234567890abcdef1234567890abcdef12345678'
    oo_result = lo_glu->getjobrun(
      iv_jobname = iv_job_name
      iv_runid = iv_run_id ).
    DATA(lo_job_run) = oo_result->get_jobrun( ).
    MESSAGE 'Job run information retrieved.' TYPE 'I'.
  CATCH /aws1/cx_gluentitynotfoundex.
    MESSAGE 'Job or job run does not exist.' TYPE 'E'.
  CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
    DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
    MESSAGE lv_invalid_error TYPE 'E'.
  CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
    DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
    MESSAGE lv_internal_error TYPE 'E'.
  CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
    DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
    MESSAGE lv_timeout_error TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetJobRun。

GetJobRuns

以下程式碼範例顯示如何使用 GetJobRuns。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    " iv_job_name = 'my-etl-job'
    oo_result = lo_glu->getjobruns( iv_jobname = iv_job_name ).
    DATA(lt_job_runs) = oo_result->get_jobruns( ).
    MESSAGE 'Job runs retrieved successfully.' TYPE 'I'.
  CATCH /aws1/cx_gluentitynotfoundex.
    MESSAGE 'Job does not exist.' TYPE 'E'.
  CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
    DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
    MESSAGE lv_invalid_error TYPE 'E'.
  CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
    DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
    MESSAGE lv_internal_error TYPE 'E'.
  CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
    DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
    MESSAGE lv_timeout_error TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetJobRuns。

`GetTables`

以下程式碼範例顯示如何使用 `GetTables`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    " iv_database_name = 'my-database'
    oo_result = lo_glu->gettables( iv_databasename = iv_database_name ).
    DATA(lt_tables) = oo_result->get_tablelist( ).
    MESSAGE 'Tables retrieved successfully.' TYPE 'I'.
  CATCH /aws1/cx_gluentitynotfoundex.
    MESSAGE 'Database does not exist.' TYPE 'E'.
  CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
    DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
    MESSAGE lv_invalid_error TYPE 'E'.
  CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
    DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
    MESSAGE lv_internal_error TYPE 'E'.
  CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
    DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
    MESSAGE lv_timeout_error TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetTables。

`ListJobs`

以下程式碼範例顯示如何使用 `ListJobs`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_glu->listjobs( ).
    DATA(lt_job_names) = oo_result->get_jobnames( ).
    MESSAGE 'Job list retrieved successfully.' TYPE 'I'.
  CATCH /aws1/cx_gluentitynotfoundex.
    MESSAGE 'No jobs found.' TYPE 'I'.
  CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
    DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
```

```
      MESSAGE lv_invalid_error TYPE 'E'.
    CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
      DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
      MESSAGE lv_internal_error TYPE 'E'.
    CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
      DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
      MESSAGE lv_timeout_error TYPE 'E'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListJobs。

`StartCrawler`

以下程式碼範例顯示如何使用 `StartCrawler`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
      " iv_crawler_name = 'my-crawler'
      lo_glu->startcrawler( iv_name = iv_crawler_name ).
      MESSAGE 'Crawler started successfully.' TYPE 'I'.
    CATCH /aws1/cx_glucrawlerrunningex.
      MESSAGE 'Crawler is already running.' TYPE 'I'.
    CATCH /aws1/cx_gluentitynotfoundex.
      MESSAGE 'Crawler does not exist.' TYPE 'E'.
    CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
      DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
      MESSAGE lv_timeout_error TYPE 'E'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 StartCrawler。

`StartJobRun`

以下程式碼範例顯示如何使用 `StartJobRun`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_job_name = 'my-etl-job'
        " iv_input_database = 'my-database'
        " iv_input_table = 'my-table'
        " iv_output_bucket_url = 's3://example-output-bucket/'

        DATA lt_arguments TYPE /aws1/cl_glugenericmap_w=>tt_genericmap.
        lt_arguments = VALUE #(
          ( VALUE /aws1/cl_glugenericmap_w=>ts_genericmap_maprow(
            key = '--input_database'
            value = NEW /aws1/cl_glugenericmap_w( iv_value = iv_input_database ) ) )
          ( VALUE /aws1/cl_glugenericmap_w=>ts_genericmap_maprow(
            key = '--input_table'
            value = NEW /aws1/cl_glugenericmap_w( iv_value = iv_input_table ) ) )
          ( VALUE /aws1/cl_glugenericmap_w=>ts_genericmap_maprow(
            key = '--output_bucket_url'
            value = NEW /aws1/cl_glugenericmap_w( iv_value =
iv_output_bucket_url ) ) ) ).

        DATA(oo_result) = lo_glu->startjobrun(
          iv_jobname = iv_job_name
          it_arguments = lt_arguments ).
        ov_job_run_id = oo_result->get_jobrunid( ).
        MESSAGE 'Job run started successfully.' TYPE 'I'.
      CATCH /aws1/cx_gluconcurrentrunsex00.
        MESSAGE 'Maximum concurrent runs exceeded.' TYPE 'E'.
      CATCH /aws1/cx_gluentitynotfoundex.
        MESSAGE 'Job does not exist.' TYPE 'E'.
      CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
        DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
        MESSAGE lv_invalid_error TYPE 'E'.
```

```
      CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
        DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
        MESSAGE lv_internal_error TYPE 'E'.
      CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
        DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
        MESSAGE lv_timeout_error TYPE 'E'.
      CATCH /aws1/cx_gluresrcnumlmtexcdex INTO DATA(lo_limit_ex).
        DATA(lv_limit_error) = lo_limit_ex->if_message~get_longtext( ).
        MESSAGE lv_limit_error TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 StartJobRun。

HealthImaging 範例使用適用於 SAP ABAP 的 SDK

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 HealthImaging 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

`CopyImageSet`

以下程式碼範例顯示如何使用 `CopyImageSet`。

適用於 SAP ABAP 的開發套件

```
    TRY.
        " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
        " iv_source_image_set_id = '12345678901234567890123456789012345678901234567890'
        " iv_source_version_id = '1'
        " iv_destination_image_set_id =
'12345678901234567890123456789012345678901234567890' (optional)
```

```
      " iv_destination_version_id = '1' (optional)
      " iv_force = abap_false
      DATA(lo_source_info) = NEW /aws1/cl_migcpsrcimagesetinf00(
        iv_latestversionid = iv_source_version_id ).
      DATA(lo_copy_info) = NEW /aws1/cl_migcpimagesetinfmtion(
        io_sourceimageset = lo_source_info ).
      IF iv_destination_image_set_id IS NOT INITIAL AND
         iv_destination_version_id IS NOT INITIAL.
        DATA(lo_dest_info) = NEW /aws1/cl_migcopydstimageset(
          iv_imagesetid = iv_destination_image_set_id
          iv_latestversionid = iv_destination_version_id ).
        lo_copy_info = NEW /aws1/cl_migcpimagesetinfmtion(
          io_sourceimageset = lo_source_info
          io_destinationimageset = lo_dest_info ).
      ENDIF.
      oo_result = lo_mig->copyimageset(
        iv_datastoreid = iv_datastore_id
        iv_sourceimagesetid = iv_source_image_set_id
        io_copyimagesetinformation = lo_copy_info
        iv_force = iv_force ).
      DATA(lo_dest_props) = oo_result->get_dstimagesetproperties( ).
      DATA(lv_new_id) = lo_dest_props->get_imagesetid( ).
      MESSAGE |Image set copied with new ID: { lv_new_id }.| TYPE 'I'.
    CATCH /aws1/cx_migaccessdeniedex.
      MESSAGE 'Access denied.' TYPE 'I'.
    CATCH /aws1/cx_migconflictexception.
      MESSAGE 'Conflict error.' TYPE 'I'.
    CATCH /aws1/cx_miginternalserverex.
      MESSAGE 'Internal server error.' TYPE 'I'.
    CATCH /aws1/cx_migresourcenotfoundex.
      MESSAGE 'Image set not found.' TYPE 'I'.
    CATCH /aws1/cx_migservicequotaexcdex.
      MESSAGE 'Service quota exceeded.' TYPE 'I'.
    CATCH /aws1/cx_migthrottlingex.
      MESSAGE 'Request throttled.' TYPE 'I'.
    CATCH /aws1/cx_migvalidationex.
      MESSAGE 'Validation error.' TYPE 'I'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CopyImageSet。

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

`CreateDatastore`

以下程式碼範例顯示如何使用 `CreateDatastore`。

適用於 SAP ABAP 的開發套件

```
TRY.
    " iv_datastore_name = 'my-datastore-name'
    oo_result = lo_mig->createdatastore( iv_datastorename = iv_datastore_name ).
    DATA(lv_datastore_id) = oo_result->get_datastoreid( ).
    MESSAGE 'Data store created.' TYPE 'I'.
  CATCH /aws1/cx_migaccessdeniedex.
    MESSAGE 'Access denied.' TYPE 'I'.
  CATCH /aws1/cx_migconflictexception.
    MESSAGE 'Conflict. Data store may already exist.' TYPE 'I'.
  CATCH /aws1/cx_miginternalserverex.
    MESSAGE 'Internal server error.' TYPE 'I'.
  CATCH /aws1/cx_migservicequotaexcdex.
    MESSAGE 'Service quota exceeded.' TYPE 'I'.
  CATCH /aws1/cx_migthrottlingex.
    MESSAGE 'Request throttled.' TYPE 'I'.
  CATCH /aws1/cx_migvalidationex.
    MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateDatastore。

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

DeleteDatastore

以下程式碼範例顯示如何使用 DeleteDatastore。

適用於 SAP ABAP 的開發套件

```
TRY.
    " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
    oo_result = lo_mig->deletedatastore( iv_datastoreid = iv_datastore_id ).
    MESSAGE 'Data store deleted.' TYPE 'I'.
  CATCH /aws1/cx_migaccessdeniedex.
    MESSAGE 'Access denied.' TYPE 'I'.
  CATCH /aws1/cx_migconflictexception.
    MESSAGE 'Conflict. Data store may contain resources.' TYPE 'I'.
  CATCH /aws1/cx_miginternalserverex.
    MESSAGE 'Internal server error.' TYPE 'I'.
  CATCH /aws1/cx_migresourcenotfoundex.
    MESSAGE 'Data store not found.' TYPE 'I'.
  CATCH /aws1/cx_migthrottlingex.
    MESSAGE 'Request throttled.' TYPE 'I'.
  CATCH /aws1/cx_migvalidationex.
    MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteDatastore。

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

DeleteImageSet

以下程式碼範例顯示如何使用 DeleteImageSet。

適用於 SAP ABAP 的開發套件

```
TRY.
    " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
```

```
    " iv_image_set_id = '12345678901234567890123456789012345678901234567890'
    oo_result = lo_mig->deleteimageset(
      iv_datastoreid = iv_datastore_id
      iv_imagesetid = iv_image_set_id ).
    MESSAGE 'Image set deleted.' TYPE 'I'.
  CATCH /aws1/cx_migaccessdeniedex.
    MESSAGE 'Access denied.' TYPE 'I'.
  CATCH /aws1/cx_migconflictexception.
    MESSAGE 'Conflict error.' TYPE 'I'.
  CATCH /aws1/cx_miginternalserverex.
    MESSAGE 'Internal server error.' TYPE 'I'.
  CATCH /aws1/cx_migresourcenotfoundex.
    MESSAGE 'Image set not found.' TYPE 'I'.
  CATCH /aws1/cx_migthrottlingex.
    MESSAGE 'Request throttled.' TYPE 'I'.
  CATCH /aws1/cx_migvalidationex.
    MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteImageSet。

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

`GetDICOMImportJob`

以下程式碼範例顯示如何使用 `GetDICOMImportJob`。

適用於 SAP ABAP 的開發套件

```
TRY.
    " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
    " iv_job_id = '12345678901234567890123456789012'
    oo_result = lo_mig->getdicomimportjob(
      iv_datastoreid = iv_datastore_id
      iv_jobid = iv_job_id ).
    DATA(lo_job_props) = oo_result->get_jobproperties( ).
```

```
      DATA(lv_job_status) = lo_job_props->get_jobstatus( ).
      MESSAGE |Job status: { lv_job_status }.| TYPE 'I'.
    CATCH /aws1/cx_migaccessdeniedex.
      MESSAGE 'Access denied.' TYPE 'I'.
    CATCH /aws1/cx_migconflictexception.
      MESSAGE 'Conflict error.' TYPE 'I'.
    CATCH /aws1/cx_miginternalserverex.
      MESSAGE 'Internal server error.' TYPE 'I'.
    CATCH /aws1/cx_migresourcenotfoundex.
      MESSAGE 'Job not found.' TYPE 'I'.
    CATCH /aws1/cx_migthrottlingex.
      MESSAGE 'Request throttled.' TYPE 'I'.
    CATCH /aws1/cx_migvalidationex.
      MESSAGE 'Validation error.' TYPE 'I'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetDICOMImportJob。

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

`GetDatastore`

以下程式碼範例顯示如何使用 `GetDatastore`。

適用於 SAP ABAP 的開發套件

```
  TRY.
      " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
      oo_result = lo_mig->getdatastore( iv_datastoreid = iv_datastore_id ).
      DATA(lo_properties) = oo_result->get_datastoreproperties( ).
      DATA(lv_name) = lo_properties->get_datastorename( ).
      DATA(lv_status) = lo_properties->get_datastorestatus( ).
      MESSAGE 'Data store properties retrieved.' TYPE 'I'.
    CATCH /aws1/cx_migaccessdeniedex.
      MESSAGE 'Access denied.' TYPE 'I'.
    CATCH /aws1/cx_miginternalserverex.
```

```
      MESSAGE 'Internal server error.' TYPE 'I'.
    CATCH /aws1/cx_migresourcenotfoundex.
      MESSAGE 'Data store not found.' TYPE 'I'.
    CATCH /aws1/cx_migthrottlingex.
      MESSAGE 'Request throttled.' TYPE 'I'.
    CATCH /aws1/cx_migvalidationex.
      MESSAGE 'Validation error.' TYPE 'I'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetDatastore。

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

`GetImageFrame`

以下程式碼範例顯示如何使用 `GetImageFrame`。

適用於 SAP ABAP 的開發套件

```
  TRY.
      " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
      " iv_image_set_id = '12345678901234567890123456789012345678901234567890'
      " iv_image_frame_id = '12345678901234567890123456789012345678901234567890'
      oo_result = lo_mig->getimageframe(
        iv_datastoreid = iv_datastore_id
        iv_imagesetid = iv_image_set_id
        io_imageframeinformation = NEW /aws1/cl_migimageframeinfmtion(
          iv_imageframeid = iv_image_frame_id ) ).
      DATA(lv_frame_blob) = oo_result->get_imageframeblob( ).
      MESSAGE 'Image frame retrieved.' TYPE 'I'.
    CATCH /aws1/cx_migaccessdeniedex.
      MESSAGE 'Access denied.' TYPE 'I'.
    CATCH /aws1/cx_migconflictexception.
      MESSAGE 'Conflict error.' TYPE 'I'.
    CATCH /aws1/cx_miginternalserverex.
      MESSAGE 'Internal server error.' TYPE 'I'.
    CATCH /aws1/cx_migresourcenotfoundex.
```

```
      MESSAGE 'Image frame not found.' TYPE 'I'.
    CATCH /aws1/cx_migthrottlingex.
      MESSAGE 'Request throttled.' TYPE 'I'.
    CATCH /aws1/cx_migvalidationex.
      MESSAGE 'Validation error.' TYPE 'I'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetImageFrame。

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

`GetImageSet`

以下程式碼範例顯示如何使用 `GetImageSet`。

適用於 SAP ABAP 的開發套件

```
  TRY.
      " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
      " iv_image_set_id = '12345678901234567890123456789012345678901234567890'
      " iv_version_id = '1' (optional)
      IF iv_version_id IS NOT INITIAL.
        oo_result = lo_mig->getimageset(
          iv_datastoreid = iv_datastore_id
          iv_imagesetid = iv_image_set_id
          iv_versionid = iv_version_id ).
      ELSE.
        oo_result = lo_mig->getimageset(
          iv_datastoreid = iv_datastore_id
          iv_imagesetid = iv_image_set_id ).
      ENDIF.
      DATA(lv_state) = oo_result->get_imagesetstate( ).
      MESSAGE |Image set retrieved with state: { lv_state }.| TYPE 'I'.
    CATCH /aws1/cx_migaccessdeniedex.
      MESSAGE 'Access denied.' TYPE 'I'.
    CATCH /aws1/cx_migconflictexception.
```

```
      MESSAGE 'Conflict error.' TYPE 'I'.
    CATCH /aws1/cx_miginternalserverex.
      MESSAGE 'Internal server error.' TYPE 'I'.
    CATCH /aws1/cx_migresourcenotfoundex.
      MESSAGE 'Image set not found.' TYPE 'I'.
    CATCH /aws1/cx_migthrottlingex.
      MESSAGE 'Request throttled.' TYPE 'I'.
    CATCH /aws1/cx_migvalidationex.
      MESSAGE 'Validation error.' TYPE 'I'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetImageSet。

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

`GetImageSetMetadata`

以下程式碼範例顯示如何使用 `GetImageSetMetadata`。

適用於 SAP ABAP 的開發套件

```
  TRY.
      " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
      " iv_image_set_id = '12345678901234567890123456789012345678901234567890'
      " iv_version_id = '1' (optional)
      IF iv_version_id IS NOT INITIAL.
        oo_result = lo_mig->getimagesetmetadata(
          iv_datastoreid = iv_datastore_id
          iv_imagesetid = iv_image_set_id
          iv_versionid = iv_version_id ).
      ELSE.
        oo_result = lo_mig->getimagesetmetadata(
          iv_datastoreid = iv_datastore_id
          iv_imagesetid = iv_image_set_id ).
      ENDIF.
      DATA(lv_metadata_blob) = oo_result->get_imagesetmetadatablob( ).
      MESSAGE 'Image set metadata retrieved.' TYPE 'I'.
```

```
  CATCH /aws1/cx_migaccessdeniedex.
    MESSAGE 'Access denied.' TYPE 'I'.
  CATCH /aws1/cx_migconflictexception.
    MESSAGE 'Conflict error.' TYPE 'I'.
  CATCH /aws1/cx_miginternalserverex.
    MESSAGE 'Internal server error.' TYPE 'I'.
  CATCH /aws1/cx_migresourcenotfoundex.
    MESSAGE 'Image set not found.' TYPE 'I'.
  CATCH /aws1/cx_migthrottlingex.
    MESSAGE 'Request throttled.' TYPE 'I'.
  CATCH /aws1/cx_migvalidationex.
    MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetImageSetMetadata。

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

`ListDICOMImportJobs`

以下程式碼範例顯示如何使用 `ListDICOMImportJobs`。

適用於 SAP ABAP 的開發套件

```
TRY.
    " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
    oo_result = lo_mig->listdicomimportjobs( iv_datastoreid = iv_datastore_id ).
    DATA(lt_jobs) = oo_result->get_jobsummaries( ).
    DATA(lv_count) = lines( lt_jobs ).
    MESSAGE |Found { lv_count } DICOM import jobs.| TYPE 'I'.
  CATCH /aws1/cx_migaccessdeniedex.
    MESSAGE 'Access denied.' TYPE 'I'.
  CATCH /aws1/cx_migconflictexception.
    MESSAGE 'Conflict error.' TYPE 'I'.
  CATCH /aws1/cx_miginternalserverex.
    MESSAGE 'Internal server error.' TYPE 'I'.
```

```
  CATCH /aws1/cx_migresourcenotfoundex.
    MESSAGE 'Resource not found.' TYPE 'I'.
  CATCH /aws1/cx_migthrottlingex.
    MESSAGE 'Request throttled.' TYPE 'I'.
  CATCH /aws1/cx_migvalidationex.
    MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListDICOMImportJobs。

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

`ListDatastores`

以下程式碼範例顯示如何使用 `ListDatastores`。

適用於 SAP ABAP 的開發套件

```
TRY.
    oo_result = lo_mig->listdatastores( ).
    DATA(lt_datastores) = oo_result->get_datastoresummaries( ).
    DATA(lv_count) = lines( lt_datastores ).
    MESSAGE |Found { lv_count } data stores.| TYPE 'I'.
  CATCH /aws1/cx_migaccessdeniedex.
    MESSAGE 'Access denied.' TYPE 'I'.
  CATCH /aws1/cx_miginternalserverex.
    MESSAGE 'Internal server error.' TYPE 'I'.
  CATCH /aws1/cx_migthrottlingex.
    MESSAGE 'Request throttled.' TYPE 'I'.
  CATCH /aws1/cx_migvalidationex.
    MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListDatastores。

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

`ListImageSetVersions`

以下程式碼範例顯示如何使用 `ListImageSetVersions`。

適用於 SAP ABAP 的開發套件

```
TRY.
    " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
    " iv_image_set_id = '12345678901234567890123456789012345678901234567890'
    oo_result = lo_mig->listimagesetversions(
      iv_datastoreid = iv_datastore_id
      iv_imagesetid = iv_image_set_id ).
    DATA(lt_versions) = oo_result->get_imagesetpropertieslist( ).
    DATA(lv_count) = lines( lt_versions ).
    MESSAGE |Found { lv_count } image set versions.| TYPE 'I'.
  CATCH /aws1/cx_migaccessdeniedex.
    MESSAGE 'Access denied.' TYPE 'I'.
  CATCH /aws1/cx_migconflictexception.
    MESSAGE 'Conflict error.' TYPE 'I'.
  CATCH /aws1/cx_miginternalserverex.
    MESSAGE 'Internal server error.' TYPE 'I'.
  CATCH /aws1/cx_migresourcenotfoundex.
    MESSAGE 'Image set not found.' TYPE 'I'.
  CATCH /aws1/cx_migthrottlingex.
    MESSAGE 'Request throttled.' TYPE 'I'.
  CATCH /aws1/cx_migvalidationex.
    MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListImageSetVersions。

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

`ListTagsForResource`

以下程式碼範例顯示如何使用 `ListTagsForResource`。

適用於 SAP ABAP 的開發套件

```
    TRY.
        " iv_resource_arn = 'arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012'
        oo_result = lo_mig->listtagsforresource( iv_resourcearn = iv_resource_arn ).
        DATA(lt_tags) = oo_result->get_tags( ).
        DATA(lv_count) = lines( lt_tags ).
        MESSAGE |Found { lv_count } tags for resource.| TYPE 'I'.
      CATCH /aws1/cx_migaccessdeniedex.
        MESSAGE 'Access denied.' TYPE 'I'.
      CATCH /aws1/cx_miginternalserverex.
        MESSAGE 'Internal server error.' TYPE 'I'.
      CATCH /aws1/cx_migresourcenotfoundex.
        MESSAGE 'Resource not found.' TYPE 'I'.
      CATCH /aws1/cx_migthrottlingex.
        MESSAGE 'Request throttled.' TYPE 'I'.
      CATCH /aws1/cx_migvalidationex.
        MESSAGE 'Validation error.' TYPE 'I'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListTagsForResource。

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

SearchImageSets

以下程式碼範例顯示如何使用 SearchImageSets。

適用於 SAP ABAP 的開發套件

```
TRY.
    " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
    oo_result = lo_mig->searchimagesets(
      iv_datastoreid = iv_datastore_id
      io_searchcriteria = io_search_criteria ).
    DATA(lt_imagesets) = oo_result->get_imagesetsmetadatasums( ).
    DATA(lv_count) = lines( lt_imagesets ).
    MESSAGE |Found { lv_count } image sets.| TYPE 'I'.
  CATCH /aws1/cx_migaccessdeniedex.
    MESSAGE 'Access denied.' TYPE 'I'.
  CATCH /aws1/cx_migconflictexception.
    MESSAGE 'Conflict error.' TYPE 'I'.
  CATCH /aws1/cx_miginternalserverex.
    MESSAGE 'Internal server error.' TYPE 'I'.
  CATCH /aws1/cx_migresourcenotfoundex.
    MESSAGE 'Resource not found.' TYPE 'I'.
  CATCH /aws1/cx_migthrottlingex.
    MESSAGE 'Request throttled.' TYPE 'I'.
  CATCH /aws1/cx_migvalidationex.
    MESSAGE 'Validation error.' TYPE 'I'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 SearchImageSets。

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

StartDICOMImportJob

以下程式碼範例顯示如何使用 StartDICOMImportJob。

適用於 SAP ABAP 的開發套件

```
    TRY.
        " iv_job_name = 'import-job-1'
        " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
        " iv_role_arn = 'arn:aws:iam::123456789012:role/ImportJobRole'
        " iv_input_s3_uri = 's3://my-bucket/input/'
        " iv_output_s3_uri = 's3://my-bucket/output/'
        oo_result = lo_mig->startdicomimportjob(
          iv_jobname = iv_job_name
          iv_datastoreid = iv_datastore_id
          iv_dataaccessrolearn = iv_role_arn
          iv_inputs3uri = iv_input_s3_uri
          iv_outputs3uri = iv_output_s3_uri ).
        DATA(lv_job_id) = oo_result->get_jobid( ).
        MESSAGE |DICOM import job started with ID: { lv_job_id }.| TYPE 'I'.
      CATCH /aws1/cx_migaccessdeniedex.
        MESSAGE 'Access denied.' TYPE 'I'.
      CATCH /aws1/cx_migconflictexception.
        MESSAGE 'Conflict error.' TYPE 'I'.
      CATCH /aws1/cx_miginternalserverex.
        MESSAGE 'Internal server error.' TYPE 'I'.
      CATCH /aws1/cx_migresourcenotfoundex.
        MESSAGE 'Resource not found.' TYPE 'I'.
      CATCH /aws1/cx_migservicequotaexcdex.
        MESSAGE 'Service quota exceeded.' TYPE 'I'.
      CATCH /aws1/cx_migthrottlingex.
        MESSAGE 'Request throttled.' TYPE 'I'.
      CATCH /aws1/cx_migvalidationex.
        MESSAGE 'Validation error.' TYPE 'I'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 StartDICOMImportJob。

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

`TagResource`

以下程式碼範例顯示如何使用 `TagResource`。

適用於 SAP ABAP 的開發套件

```
    TRY.
        " iv_resource_arn = 'arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012'
        lo_mig->tagresource(
          iv_resourcearn = iv_resource_arn
          it_tags = it_tags ).
        MESSAGE 'Resource tagged successfully.' TYPE 'I'.
      CATCH /aws1/cx_migaccessdeniedex.
        MESSAGE 'Access denied.' TYPE 'I'.
      CATCH /aws1/cx_miginternalserverex.
        MESSAGE 'Internal server error.' TYPE 'I'.
      CATCH /aws1/cx_migresourcenotfoundex.
        MESSAGE 'Resource not found.' TYPE 'I'.
      CATCH /aws1/cx_migthrottlingex.
        MESSAGE 'Request throttled.' TYPE 'I'.
      CATCH /aws1/cx_migvalidationex.
        MESSAGE 'Validation error.' TYPE 'I'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 TagResource。

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

`UntagResource`

以下程式碼範例顯示如何使用 `UntagResource`。

適用於 SAP ABAP 的開發套件

```
    TRY.
```

```
        " iv_resource_arn = 'arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012'
        lo_mig->untagresource(
          iv_resourcearn = iv_resource_arn
          it_tagkeys = it_tag_keys ).
        MESSAGE 'Resource untagged successfully.' TYPE 'I'.
      CATCH /aws1/cx_migaccessdeniedex.
        MESSAGE 'Access denied.' TYPE 'I'.
      CATCH /aws1/cx_miginternalserverex.
        MESSAGE 'Internal server error.' TYPE 'I'.
      CATCH /aws1/cx_migresourcenotfoundex.
        MESSAGE 'Resource not found.' TYPE 'I'.
      CATCH /aws1/cx_migthrottlingex.
        MESSAGE 'Request throttled.' TYPE 'I'.
      CATCH /aws1/cx_migvalidationex.
        MESSAGE 'Validation error.' TYPE 'I'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 UntagResource。

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

`UpdateImageSetMetadata`

以下程式碼範例顯示如何使用 `UpdateImageSetMetadata`。

適用於 SAP ABAP 的開發套件

```
    TRY.
        " iv_datastore_id = '12345678901234567890123456789012345678901234567890'
        " iv_image_set_id = '12345678901234567890123456789012345678901234567890'
        " iv_latest_version_id = '1'
        " iv_force = abap_false
        oo_result = lo_mig->updateimagesetmetadata(
          iv_datastoreid = iv_datastore_id
          iv_imagesetid = iv_image_set_id
```

```
          iv_latestversionid = iv_latest_version_id
          io_updateimagesetmetupdates = io_metadata_updates
          iv_force = iv_force ).
        DATA(lv_new_version) = oo_result->get_latestversionid( ).
        MESSAGE |Image set metadata updated to version: { lv_new_version }.| TYPE
'I'.
      CATCH /aws1/cx_migaccessdeniedex.
        MESSAGE 'Access denied.' TYPE 'I'.
      CATCH /aws1/cx_migconflictexception.
        MESSAGE 'Conflict error.' TYPE 'I'.
      CATCH /aws1/cx_miginternalserverex.
        MESSAGE 'Internal server error.' TYPE 'I'.
      CATCH /aws1/cx_migresourcenotfoundex.
        MESSAGE 'Image set not found.' TYPE 'I'.
      CATCH /aws1/cx_migservicequotaexcdex.
        MESSAGE 'Service quota exceeded.' TYPE 'I'.
      CATCH /aws1/cx_migthrottlingex.
        MESSAGE 'Request throttled.' TYPE 'I'.
      CATCH /aws1/cx_migvalidationex.
        MESSAGE 'Validation error.' TYPE 'I'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 UpdateImageSetMetadata。

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

使用適用於 SAP ABAP 的 SDK 的 HealthLake 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 HealthLake 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

`CreateFHIRDatastore`

以下程式碼範例顯示如何使用 `CreateFHIRDatastore`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_datastore_name = 'MyHealthLakeDataStore'
        oo_result = lo_hll->createfhirdatastore(
          iv_datastorename = iv_datastore_name
          iv_datastoretypeversion = 'R4'
        ).
        MESSAGE 'Data store created successfully.' TYPE 'I'.
      CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
      CATCH /aws1/cx_hllinternalserverex INTO DATA(lo_internal_ex).
        lv_error = |Internal server error: { lo_internal_ex->av_err_code }-
{ lo_internal_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_internal_ex.
      CATCH /aws1/cx_hllthrottlingex INTO DATA(lo_throttling_ex).
        lv_error = |Throttling error: { lo_throttling_ex->av_err_code }-
{ lo_throttling_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_throttling_ex.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateFHIRDatastore。

`DeleteFHIRDatastore`

以下程式碼範例顯示如何使用 `DeleteFHIRDatastore`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
        oo_result = lo_hll->deletefhirdatastore(
          iv_datastoreid = iv_datastore_id
        ).
        MESSAGE 'Data store deleted successfully.' TYPE 'I'.
      CATCH /aws1/cx_hllaccessdeniedex INTO DATA(lo_access_ex).
        DATA(lv_error) = |Access denied: { lo_access_ex->av_err_code }-
{ lo_access_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_access_ex.
      CATCH /aws1/cx_hllconflictexception INTO DATA(lo_conflict_ex).
        lv_error = |Conflict error: { lo_conflict_ex->av_err_code }-
{ lo_conflict_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_conflict_ex.
      CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
        lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_notfound_ex.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteFHIRDatastore。

DescribeFHIRDatastore

以下程式碼範例顯示如何使用 DescribeFHIRDatastore。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
        oo_result = lo_hll->describefhirdatastore(
          iv_datastoreid = iv_datastore_id
        ).
        DATA(lo_datastore_properties) = oo_result->get_datastoreproperties( ).
        IF lo_datastore_properties IS BOUND.
          DATA(lv_datastore_name) = lo_datastore_properties->get_datastorename( ).
          DATA(lv_datastore_status) = lo_datastore_properties-
>get_datastorestatus( ).
          MESSAGE 'Data store described successfully.' TYPE 'I'.
        ENDIF.
      CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
        DATA(lv_error) = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_notfound_ex.
      CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        lv_error = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeFHIRDatastore。

DescribeFHIRExportJob

以下程式碼範例顯示如何使用 DescribeFHIRExportJob。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
        " iv_job_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
        oo_result = lo_hll->describefhirexportjob(
          iv_datastoreid = iv_datastore_id
          iv_jobid = iv_job_id
        ).
        DATA(lo_export_job_properties) = oo_result->get_exportjobproperties( ).
        IF lo_export_job_properties IS BOUND.
          DATA(lv_job_status) = lo_export_job_properties->get_jobstatus( ).
          MESSAGE |Export job status: { lv_job_status }.| TYPE 'I'.
        ENDIF.
      CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
        DATA(lv_error) = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_notfound_ex.
      CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        lv_error = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeFHIRExportJob。

DescribeFHIRImportJob

以下程式碼範例顯示如何使用 DescribeFHIRImportJob。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
        " iv_job_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
        oo_result = lo_hll->describefhirimportjob(
          iv_datastoreid = iv_datastore_id
          iv_jobid = iv_job_id
        ).
        DATA(lo_import_job_properties) = oo_result->get_importjobproperties( ).
        IF lo_import_job_properties IS BOUND.
          DATA(lv_job_status) = lo_import_job_properties->get_jobstatus( ).
          MESSAGE |Import job status: { lv_job_status }.| TYPE 'I'.
        ENDIF.
      CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
        DATA(lv_error) = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_notfound_ex.
      CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        lv_error = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeFHIRImportJob。

`ListFHIRDatastores`

以下程式碼範例顯示如何使用 `ListFHIRDatastores`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        oo_result = lo_hll->listfhirdatastores( ).
        DATA(lt_datastores) = oo_result->get_datastorepropertieslist( ).
        DATA(lv_datastore_count) = lines( lt_datastores ).
        MESSAGE |Found { lv_datastore_count } data store(s).| TYPE 'I'.
      CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
      CATCH /aws1/cx_hllthrottlingex INTO DATA(lo_throttling_ex).
        lv_error = |Throttling error: { lo_throttling_ex->av_err_code }-
{ lo_throttling_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_throttling_ex.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListFHIRDatastores。

`ListFHIRExportJobs`

以下程式碼範例顯示如何使用 `ListFHIRExportJobs`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
        IF iv_submitted_after IS NOT INITIAL.
          oo_result = lo_hll->listfhirexportjobs(
            iv_datastoreid = iv_datastore_id
            iv_submittedafter = iv_submitted_after
          ).
        ELSE.
          oo_result = lo_hll->listfhirexportjobs(
            iv_datastoreid = iv_datastore_id
          ).
        ENDIF.
        DATA(lt_export_jobs) = oo_result->get_exportjobpropertieslist( ).
        DATA(lv_job_count) = lines( lt_export_jobs ).
        MESSAGE |Found { lv_job_count } export job(s).| TYPE 'I'.
      CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
      CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
        lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_notfound_ex.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListFHIRExportJobs。

`ListFHIRImportJobs`

以下程式碼範例顯示如何使用 `ListFHIRImportJobs`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
        IF iv_submitted_after IS NOT INITIAL.
          oo_result = lo_hll->listfhirimportjobs(
            iv_datastoreid = iv_datastore_id
            iv_submittedafter = iv_submitted_after
          ).
        ELSE.
          oo_result = lo_hll->listfhirimportjobs(
            iv_datastoreid = iv_datastore_id
          ).
        ENDIF.
        DATA(lt_import_jobs) = oo_result->get_importjobpropertieslist( ).
        DATA(lv_job_count) = lines( lt_import_jobs ).
        MESSAGE |Found { lv_job_count } import job(s).| TYPE 'I'.
      CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
      CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
        lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_notfound_ex.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListFHIRImportJobs。

`ListTagsForResource`

以下程式碼範例顯示如何使用 `ListTagsForResource`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_resource_arn = 'arn:aws:healthlake:us-east-1:123456789012:datastore/
fhir/a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
        DATA(lo_result) = lo_hll->listtagsforresource(
          iv_resourcearn = iv_resource_arn
        ).
        ot_tags = lo_result->get_tags( ).
        DATA(lv_tag_count) = lines( ot_tags ).
        MESSAGE |Found { lv_tag_count } tag(s).| TYPE 'I'.
      CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
      CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
        lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_notfound_ex.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListTagsForResource。

`StartFHIRExportJob`

以下程式碼範例顯示如何使用 `StartFHIRExportJob`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_job_name = 'MyExportJob'
        " iv_output_s3_uri = 's3://my-bucket/export/output/'
        " iv_kms_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012'
        " iv_data_access_role_arn = 'arn:aws:iam::123456789012:role/
HealthLakeExportRole'
        oo_result = lo_hll->startfhirexportjob(
          iv_jobname = iv_job_name
          io_outputdataconfig = NEW /aws1/cl_hlloutputdataconfig(
            io_s3configuration = NEW /aws1/cl_hlls3configuration(
              iv_s3uri = iv_output_s3_uri
              iv_kmskeyid = iv_kms_key_id
            )
          )
          iv_dataaccessrolearn = iv_data_access_role_arn
          iv_datastoreid = iv_datastore_id
        ).
        DATA(lv_job_id) = oo_result->get_jobid( ).
        MESSAGE |Export job started with ID { lv_job_id }.| TYPE 'I'.
      CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
      CATCH /aws1/cx_hllthrottlingex INTO DATA(lo_throttling_ex).
        lv_error = |Throttling error: { lo_throttling_ex->av_err_code }-
{ lo_throttling_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_throttling_ex.
      CATCH /aws1/cx_hllaccessdeniedex INTO DATA(lo_access_ex).
        lv_error = |Access denied: { lo_access_ex->av_err_code }-{ lo_access_ex-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
```

```
        RAISE EXCEPTION lo_access_ex.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 StartFHIRExportJob。

`StartFHIRImportJob`

以下程式碼範例顯示如何使用 `StartFHIRImportJob`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_job_name = 'MyImportJob'
        " iv_input_s3_uri = 's3://my-bucket/import/data.ndjson'
        " iv_job_output_s3_uri = 's3://my-bucket/import/output/'
        " iv_kms_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012'
        " iv_data_access_role_arn = 'arn:aws:iam::123456789012:role/
HealthLakeImportRole'
        oo_result = lo_hll->startfhirimportjob(
          iv_jobname = iv_job_name
          io_inputdataconfig = NEW /aws1/cl_hllinputdataconfig( iv_s3uri =
 iv_input_s3_uri )
          io_joboutputdataconfig = NEW /aws1/cl_hlloutputdataconfig(
            io_s3configuration = NEW /aws1/cl_hlls3configuration(
              iv_s3uri = iv_job_output_s3_uri
              iv_kmskeyid = iv_kms_key_id
            )
          )
          iv_dataaccessrolearn = iv_data_access_role_arn
          iv_datastoreid = iv_datastore_id
        ).
        DATA(lv_job_id) = oo_result->get_jobid( ).
```

```
        MESSAGE |Import job started with ID { lv_job_id }.| TYPE 'I'.
      CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
      CATCH /aws1/cx_hllthrottlingex INTO DATA(lo_throttling_ex).
        lv_error = |Throttling error: { lo_throttling_ex->av_err_code }-
{ lo_throttling_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_throttling_ex.
      CATCH /aws1/cx_hllaccessdeniedex INTO DATA(lo_access_ex).
        lv_error = |Access denied: { lo_access_ex->av_err_code }-{ lo_access_ex-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_access_ex.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 StartFHIRImportJob。

`TagResource`

以下程式碼範例顯示如何使用 `TagResource`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_resource_arn = 'arn:aws:healthlake:us-east-1:123456789012:datastore/
fhir/a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
        lo_hll->tagresource(
          iv_resourcearn = iv_resource_arn
          it_tags = it_tags
        ).
```

```
        MESSAGE 'Resource tagged successfully.' TYPE 'I'.
      CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
      CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
        lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_notfound_ex.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 TagResource。

`UntagResource`

以下程式碼範例顯示如何使用 `UntagResource`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_resource_arn = 'arn:aws:healthlake:us-east-1:123456789012:datastore/
fhir/a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
        lo_hll->untagresource(
          iv_resourcearn = iv_resource_arn
          it_tagkeys = it_tag_keys
        ).
        MESSAGE 'Resource untagged successfully.' TYPE 'I'.
      CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
```

```
      CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
        lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_notfound_ex.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 UntagResource。

使用 SDK for SAP ABAP 的 IAM 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 IAM 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

`AttachRolePolicy`

以下程式碼範例顯示如何使用 `AttachRolePolicy`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
```

```
    lo_iam->attachrolepolicy(
      iv_rolename = iv_role_name
      iv_policyarn = iv_policy_arn ).
    MESSAGE 'Policy attached to role successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Role or policy does not exist.' TYPE 'E'.
  CATCH /aws1/cx_iamlimitexceededex.
    MESSAGE 'Policy attachment limit exceeded.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 AttachRolePolicy。

`AttachUserPolicy`

以下程式碼範例顯示如何使用 `AttachUserPolicy`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_iam->attachuserpolicy(
      iv_username = iv_user_name
      iv_policyarn = iv_policy_arn ).
    MESSAGE 'Policy attached to user successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'User or policy does not exist.' TYPE 'E'.
  CATCH /aws1/cx_iamlimitexceededex.
    MESSAGE 'Policy attachment limit exceeded.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 AttachUserPolicy。

`CreateAccessKey`

以下程式碼範例顯示如何使用 `CreateAccessKey`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_iam->createaccesskey(
      iv_username = iv_user_name ).
    MESSAGE 'Access key created successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'User does not exist.' TYPE 'E'.
  CATCH /aws1/cx_iamlimitexceededex.
    MESSAGE 'Maximum number of access keys reached.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateAccessKey。

`CreateAccountAlias`

以下程式碼範例顯示如何使用 `CreateAccountAlias`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
```

```
    lo_iam->createaccountalias(
      iv_accountalias = iv_account_alias ).
    MESSAGE 'Account alias created successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamentityalrdyexex.
    MESSAGE 'Account alias already exists.' TYPE 'E'.
  CATCH /aws1/cx_iamlimitexceededex.
    MESSAGE 'Account alias limit exceeded.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateAccountAlias。

`CreatePolicy`

以下程式碼範例顯示如何使用 `CreatePolicy`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_iam->createpolicy(
      iv_policyname = iv_policy_name
      iv_policydocument = iv_policy_document
      iv_description = iv_description ).
    MESSAGE 'Policy created successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamentityalrdyexex.
    MESSAGE 'Policy already exists.' TYPE 'E'.
  CATCH /aws1/cx_iammalformedplydocex.
    MESSAGE 'Policy document is malformed.' TYPE 'E'.
  CATCH /aws1/cx_iamlimitexceededex.
    MESSAGE 'Policy limit exceeded.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreatePolicy。

`CreatePolicyVersion`

以下程式碼範例顯示如何使用 `CreatePolicyVersion`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_iam->createpolicyversion(
      iv_policyarn = iv_policy_arn
      iv_policydocument = iv_policy_document
      iv_setasdefault = iv_set_as_default ).
    MESSAGE 'Policy version created successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Policy does not exist.' TYPE 'E'.
  CATCH /aws1/cx_iammalformedplydocex.
    MESSAGE 'Policy document is malformed.' TYPE 'E'.
  CATCH /aws1/cx_iamlimitexceededex.
    MESSAGE 'Policy version limit exceeded.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreatePolicyVersion。

`CreateRole`

以下程式碼範例顯示如何使用 `CreateRole`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_iam->createrole(
      iv_rolename = iv_role_name
      iv_assumerolepolicydocument = iv_assume_role_policy_document ).
    MESSAGE 'Role created successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamentityalrdyexex.
    MESSAGE 'Role already exists.' TYPE 'E'.
  CATCH /aws1/cx_iammalformedplydocex.
    MESSAGE 'Assume role policy document is malformed.' TYPE 'E'.
  CATCH /aws1/cx_iamlimitexceededex.
    MESSAGE 'Role limit exceeded.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateRole。

`CreateServiceLinkedRole`

以下程式碼範例顯示如何使用 `CreateServiceLinkedRole`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_iam->listpolicyversions(
      iv_policyarn = iv_policy_arn ).
    MESSAGE 'Retrieved policy versions list.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Policy does not exist.' TYPE 'E'.
  CATCH /aws1/cx_iamservicefailureex.
    MESSAGE 'Service failure when listing policy versions.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateServiceLinkedRole。

`CreateUser`

以下程式碼範例顯示如何使用 `CreateUser`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_iam->createuser(
      iv_username = iv_user_name ).
    MESSAGE 'User created successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamentityalrdyexex.
    MESSAGE 'User already exists.' TYPE 'E'.
  CATCH /aws1/cx_iamlimitexceededex.
    MESSAGE 'Limit exceeded for IAM users.' TYPE 'E'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Entity does not exist.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateUser。

`DeleteAccessKey`

以下程式碼範例顯示如何使用 `DeleteAccessKey`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
```

```
    lo_iam->deleteaccesskey(
      iv_accesskeyid = iv_access_key_id
      iv_username = iv_user_name ).
    MESSAGE 'Access key deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Access key or user does not exist.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteAccessKey。

`DeleteAccountAlias`

以下程式碼範例顯示如何使用 `DeleteAccountAlias`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_iam->deleteaccountalias(
      iv_accountalias = iv_account_alias ).
    MESSAGE 'Account alias deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Account alias does not exist.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteAccountAlias。

`DeletePolicy`

以下程式碼範例顯示如何使用 `DeletePolicy`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_iam->deletepolicy( iv_policyarn = iv_policy_arn ).
    MESSAGE 'Policy deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Policy does not exist.' TYPE 'E'.
  CATCH /aws1/cx_iamdeleteconflictex.
    MESSAGE 'Policy cannot be deleted due to attachments.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeletePolicy。

`DeletePolicyVersion`

以下程式碼範例顯示如何使用 `DeletePolicyVersion`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_iam->deletepolicyversion(
      iv_policyarn = iv_policy_arn
      iv_versionid = iv_version_id ).
    MESSAGE 'Policy version deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Policy or version does not exist.' TYPE 'E'.
```

```
  CATCH /aws1/cx_iamdeleteconflictex.
    MESSAGE 'Cannot delete default policy version.' TYPE 'E'.
  CATCH /aws1/cx_iamlimitexceededex.
    MESSAGE 'Limit exceeded.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeletePolicyVersion。

DeleteRole

以下程式碼範例顯示如何使用 DeleteRole。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_iam->deleterole( iv_rolename = iv_role_name ).
    MESSAGE 'Role deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Role does not exist.' TYPE 'E'.
  CATCH /aws1/cx_iamdeleteconflictex.
    MESSAGE 'Role cannot be deleted due to attached resources.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteRole。

DeleteUser

以下程式碼範例顯示如何使用 DeleteUser。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_iam->deleteuser( iv_username = iv_user_name ).
    MESSAGE 'User deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'User does not exist.' TYPE 'E'.
  CATCH /aws1/cx_iamdeleteconflictex.
    MESSAGE 'User cannot be deleted due to attached resources.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteUser。

`DetachRolePolicy`

以下程式碼範例顯示如何使用 `DetachRolePolicy`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_iam->detachrolepolicy(
      iv_rolename = iv_role_name
      iv_policyarn = iv_policy_arn ).
    MESSAGE 'Policy detached from role successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Role or policy does not exist.' TYPE 'E'.
```

```
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DetachRolePolicy。

`DetachUserPolicy`

以下程式碼範例顯示如何使用 `DetachUserPolicy`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_iam->detachuserpolicy(
      iv_username = iv_user_name
      iv_policyarn = iv_policy_arn ).
    MESSAGE 'Policy detached from user successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'User or policy does not exist.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DetachUserPolicy。

`GenerateCredentialReport`

以下程式碼範例顯示如何使用 `GenerateCredentialReport`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_iam->generatecredentialreport( ).
    MESSAGE 'Credential report generation started.' TYPE 'I'.
  CATCH /aws1/cx_iamlimitexceededex.
    MESSAGE 'Report generation limit exceeded.' TYPE 'E'.
  CATCH /aws1/cx_iamservicefailureex.
    MESSAGE 'Service failure when generating credential report.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GenerateCredentialReport。

`GetAccessKeyLastUsed`

以下程式碼範例顯示如何使用 `GetAccessKeyLastUsed`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_iam->getaccesskeylastused(
      iv_accesskeyid = iv_access_key_id ).
    MESSAGE 'Retrieved access key last used information.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Access key does not exist.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetAccessKeyLastUsed。

GetAccountAuthorizationDetails

以下程式碼範例顯示如何使用 GetAccountAuthorizationDetails。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   TRY.
       oo_result = lo_iam->getaccountauthdetails( ).
       MESSAGE 'Retrieved account authorization details.' TYPE 'I'.
     CATCH /aws1/cx_iamservicefailureex.
       MESSAGE 'Service failure when getting account authorization details.' TYPE
'E'.
   ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetAccountAuthorizationDetails。

GetAccountPasswordPolicy

以下程式碼範例顯示如何使用 GetAccountPasswordPolicy。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_iam->getaccountpasswordpolicy( ).
    MESSAGE 'Retrieved account password policy.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'No password policy exists.' TYPE 'I'.
  CATCH /aws1/cx_iamservicefailureex.
    MESSAGE 'Service failure when getting password policy.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetAccountPasswordPolicy。

`GetAccountSummary`

以下程式碼範例顯示如何使用 `GetAccountSummary`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_iam->getaccountsummary( ).
    MESSAGE 'Retrieved account summary.' TYPE 'I'.
  CATCH /aws1/cx_iamservicefailureex.
    MESSAGE 'Service failure when getting account summary.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetAccountSummary。

`GetCredentialReport`

以下程式碼範例顯示如何使用 `GetCredentialReport`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_iam->getcredentialreport( ).
    MESSAGE 'Retrieved credential report.' TYPE 'I'.
  CATCH /aws1/cx_iamcredrptnotpresen00.
    MESSAGE 'Credential report not present.' TYPE 'E'.
  CATCH /aws1/cx_iamcredrptexpiredex.
    MESSAGE 'Credential report expired.' TYPE 'E'.
  CATCH /aws1/cx_iamcredrptnotreadyex.
    MESSAGE 'Credential report not ready.' TYPE 'E'.
  CATCH /aws1/cx_iamservicefailureex.
    MESSAGE 'Service failure when getting credential report.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetCredentialReport。

`GetPolicy`

以下程式碼範例顯示如何使用 `GetPolicy`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_iam->getpolicy( iv_policyarn = iv_policy_arn ).
    MESSAGE 'Retrieved policy information.' TYPE 'I'.
```

```
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Policy does not exist.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetPolicy。

`GetRole`

以下程式碼範例顯示如何使用 `GetRole`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_iam->getrole( iv_rolename = iv_role_name ).
    MESSAGE 'Retrieved role information.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Role does not exist.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetRole。

`ListAccessKeys`

以下程式碼範例顯示如何使用 `ListAccessKeys`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_iam->listaccesskeys(
      iv_username = iv_user_name ).
    MESSAGE 'Retrieved access key list.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'User does not exist.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListAccessKeys。

`ListAccountAliases`

以下程式碼範例顯示如何使用 `ListAccountAliases`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_iam->listaccountaliases( ).
    MESSAGE 'Retrieved account alias list.' TYPE 'I'.
  CATCH /aws1/cx_iamservicefailureex.
    MESSAGE 'Service failure when listing account aliases.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListAccountAliases。

`ListAttachedRolePolicies`

以下程式碼範例顯示如何使用 `ListAttachedRolePolicies`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_iam->listattachedrolepolicies(
      iv_rolename = iv_role_name ).
    MESSAGE 'Retrieved attached policy list for role.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'Role does not exist.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListAttachedRolePolicies。

`ListGroups`

以下程式碼範例顯示如何使用 `ListGroups`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_iam->listgroups( ).
    MESSAGE 'Retrieved group list.' TYPE 'I'.
  CATCH /aws1/cx_iamservicefailureex.
    MESSAGE 'Service failure when listing groups.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListGroups。

`ListPolicies`

以下程式碼範例顯示如何使用 `ListPolicies`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_iam->listpolicies( iv_scope = iv_scope ).
    MESSAGE 'Retrieved policy list.' TYPE 'I'.
  CATCH /aws1/cx_iamservicefailureex.
    MESSAGE 'Service failure when listing policies.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListPolicies。

`ListPolicyVersions`

以下程式碼範例顯示如何使用 `ListPolicyVersions`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_iam->listpolicyversions(
```

```
        iv_policyarn = iv_policy_arn ).
      MESSAGE 'Retrieved policy versions list.' TYPE 'I'.
    CATCH /aws1/cx_iamnosuchentityex.
      MESSAGE 'Policy does not exist.' TYPE 'E'.
    CATCH /aws1/cx_iamservicefailureex.
      MESSAGE 'Service failure when listing policy versions.' TYPE 'E'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListPolicyVersions。

`ListRolePolicies`

以下程式碼範例顯示如何使用 `ListRolePolicies`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
      oo_result = lo_iam->listrolepolicies(
        iv_rolename = iv_role_name ).
      MESSAGE 'Retrieved inline policy list for role.' TYPE 'I'.
    CATCH /aws1/cx_iamnosuchentityex.
      MESSAGE 'Role does not exist.' TYPE 'E'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListRolePolicies。

`ListRoles`

以下程式碼範例顯示如何使用 `ListRoles`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_iam->listroles( ).
    MESSAGE 'Retrieved role list.' TYPE 'I'.
  CATCH /aws1/cx_iamservicefailureex.
    MESSAGE 'Service failure when listing roles.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListRoles。

`ListSAMLProviders`

以下程式碼範例顯示如何使用 `ListSAMLProviders`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_iam->listsamlproviders( ).
    MESSAGE 'Retrieved SAML provider list.' TYPE 'I'.
  CATCH /aws1/cx_iamservicefailureex.
    MESSAGE 'Service failure when listing SAML providers.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListSAMLProviders。

`ListUsers`

以下程式碼範例顯示如何使用 `ListUsers`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_iam->listusers( ).
    MESSAGE 'Retrieved user list.' TYPE 'I'.
  CATCH /aws1/cx_iamservicefailureex.
    MESSAGE 'Service failure when listing users.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListUsers。

`SetDefaultPolicyVersion`

以下程式碼範例顯示如何使用 `SetDefaultPolicyVersion`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_iam->setdefaultpolicyversion(
      iv_policyarn = iv_policy_arn
      iv_versionid = iv_version_id ).
    MESSAGE 'Default policy version set successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
```

```
      MESSAGE 'Policy or version does not exist.' TYPE 'E'.
    CATCH /aws1/cx_iaminvalidinputex.
      MESSAGE 'Invalid input provided.' TYPE 'E'.
    CATCH /aws1/cx_iamlimitexceededex.
      MESSAGE 'Limit exceeded.' TYPE 'E'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 SetDefaultPolicyVersion。

UpdateAccessKey

以下程式碼範例顯示如何使用 UpdateAccessKey。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
      lo_iam->updateaccesskey(
        iv_accesskeyid = iv_access_key_id
        iv_status = iv_status
        iv_username = iv_user_name ).
      MESSAGE 'Access key updated successfully.' TYPE 'I'.
    CATCH /aws1/cx_iamnosuchentityex.
      MESSAGE 'Access key or user does not exist.' TYPE 'E'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 UpdateAccessKey。

UpdateUser

以下程式碼範例顯示如何使用 UpdateUser。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_iam->updateuser(
      iv_username = iv_user_name
      iv_newusername = iv_new_user_name ).
    MESSAGE 'User updated successfully.' TYPE 'I'.
  CATCH /aws1/cx_iamnosuchentityex.
    MESSAGE 'User does not exist.' TYPE 'E'.
  CATCH /aws1/cx_iamentityalrdyexex.
    MESSAGE 'New user name already exists.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 UpdateUser。

AWS IoT SiteWise 使用適用於 SAP ABAP 的 SDK 的範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS 開發套件來執行動作和實作常見案例 AWS IoT SiteWise。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 開始使用
- 動作

開始使用

您好 AWS IoT SiteWise

下列程式碼範例示範如何開始使用 AWS IoT SiteWise。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        oo_result = lo_ios->listassetmodels( ). " oo_result is returned for testing
purposes. "
        DATA(lt_asset_models) = oo_result->get_assetmodelsummaries( ).
        MESSAGE 'Retrieved list of asset models.' TYPE 'I'.
      CATCH /aws1/cx_rt_generic.
        MESSAGE 'Unable to list asset models.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListAssetModels。

動作

`BatchPutAssetPropertyValue`

以下程式碼範例顯示如何使用 `BatchPutAssetPropertyValue`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_ios->batchputassetpropertyvalue(
      it_entries = it_entries
    ).
    MESSAGE 'Data sent to IoT SiteWise asset successfully.' TYPE 'I'.
  CATCH /aws1/cx_iosresourcenotfoundex.
    MESSAGE 'Asset does not exist.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 BatchPutAssetPropertyValue。

`CreateAsset`

以下程式碼範例顯示如何使用 `CreateAsset`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_ios->createasset(
      iv_assetname = iv_asset_name
      iv_assetmodelid = iv_asset_model_id
    ). " oo_result is returned for testing purposes. "
    MESSAGE 'IoT SiteWise asset created' TYPE 'I'.
  CATCH /aws1/cx_iosresourcenotfoundex.
    MESSAGE 'Asset model does not exist.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateAsset。

`CreateAssetModel`

以下程式碼範例顯示如何使用 `CreateAssetModel`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_ios->createassetmodel(
      iv_assetmodelname = iv_asset_model_name
      iv_assetmodeldescription = 'This is a sample asset model description.'
      it_assetmodelproperties = it_properties
    ). " oo_result is returned for testing purposes. "
    MESSAGE 'IoT SiteWise asset model created' TYPE 'I'.
  CATCH /aws1/cx_iosresrcalrdyexistsex.
    MESSAGE 'Asset model already exists.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateAssetModel。

`CreateGateway`

以下程式碼範例顯示如何使用 `CreateGateway`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
```

```
    oo_result = lo_ios->creategateway(
      iv_gatewayname = iv_gateway_name
      io_gatewayplatform = NEW /aws1/cl_iosgatewayplatform(
        io_greengrassv2 = NEW /aws1/cl_iosgreengrassv2(
          iv_coredevicethingname = iv_core_device_thing_name
        )
      )
      it_tags = VALUE /aws1/cl_iostagmap_w=>tt_tagmap(
        (
          VALUE /aws1/cl_iostagmap_w=>ts_tagmap_maprow(
            key = 'Environment'
            value = NEW /aws1/cl_iostagmap_w( 'Production' )
          )
        )
      )
    ). " oo_result is returned for testing purposes. "
    MESSAGE 'IoT SiteWise gateway created' TYPE 'I'.
  CATCH /aws1/cx_iosresrcalrdyexistsex.
    MESSAGE 'Gateway already exists.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateGateway。

`DeleteAsset`

以下程式碼範例顯示如何使用 `DeleteAsset`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_ios->deleteasset(
      iv_assetid = iv_asset_id
    ).
```

```
      MESSAGE 'IoT SiteWise asset deleted.' TYPE 'I'.
    CATCH /aws1/cx_rt_generic.
      MESSAGE 'Unable to delete asset.' TYPE 'E'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteAsset。

`DeleteAssetModel`

以下程式碼範例顯示如何使用 `DeleteAssetModel`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
      lo_ios->deleteassetmodel(
        iv_assetmodelid = iv_asset_model_id
      ).
      MESSAGE 'IoT SiteWise asset model deleted.' TYPE 'I'.
    CATCH /aws1/cx_rt_generic.
      MESSAGE 'Unable to delete asset model.' TYPE 'E'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteAssetModel。

`DeleteGateway`

以下程式碼範例顯示如何使用 `DeleteGateway`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_ios->deletegateway(
      iv_gatewayid = iv_gateway_id
    ).
    MESSAGE 'IoT SiteWise gateway deleted.' TYPE 'I'.
  CATCH /aws1/cx_iosresourcenotfoundex.
    MESSAGE 'Gateway does not exist.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteGateway。

`DescribeGateway`

以下程式碼範例顯示如何使用 `DescribeGateway`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_ios->describegateway(
      iv_gatewayid = iv_gateway_id
    ). " oo_result is returned for testing purposes. "
    MESSAGE 'Retrieved gateway description.' TYPE 'I'.
  CATCH /aws1/cx_iosresourcenotfoundex.
```

```
    MESSAGE 'Gateway does not exist.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeGateway。

`GetAssetPropertyValue`

以下程式碼範例顯示如何使用 `GetAssetPropertyValue`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_ios->getassetpropertyvalue(
      iv_assetid = iv_asset_id
      iv_propertyid = iv_property_id
    ). " oo_result is returned for testing purposes. "
    MESSAGE 'Retrieved asset property value.' TYPE 'I'.
  CATCH /aws1/cx_iosresourcenotfoundex.
    MESSAGE 'Asset or property does not exist.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetAssetPropertyValue。

`ListAssetModelProperties`

以下程式碼範例顯示如何使用 `ListAssetModelProperties`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        oo_result = lo_ios->listassetmodelproperties(
          iv_assetmodelid = iv_asset_model_id
        ). " oo_result is returned for testing purposes. "
        DATA(lt_properties) = oo_result->get_assetmodelpropertysums( ).
        MESSAGE 'Retrieved list of asset model properties.' TYPE 'I'.
      CATCH /aws1/cx_rt_generic.
        MESSAGE 'Unable to list asset model properties.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListAssetModelProperties。

`ListAssetModels`

以下程式碼範例顯示如何使用 `ListAssetModels`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        oo_result = lo_ios->listassetmodels( ). " oo_result is returned for testing
purposes. "
        DATA(lt_asset_models) = oo_result->get_assetmodelsummaries( ).
        MESSAGE 'Retrieved list of asset models.' TYPE 'I'.
```

```
    CATCH /aws1/cx_rt_generic.
      MESSAGE 'Unable to list asset models.' TYPE 'E'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListAssetModels。

使用適用於 SAP ABAP 的 SDK 的 Amazon Keyspaces 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 Amazon Keyspaces 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

`CreateKeyspace`

以下程式碼範例顯示如何使用 `CreateKeyspace`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
      oo_result = lo_kys->createkeyspace(
        iv_keyspacename = iv_keyspace_name ).
```

```
        MESSAGE 'Keyspace created successfully.' TYPE 'I'.
      CATCH /aws1/cx_kysconflictexception.
        MESSAGE 'Keyspace already exists.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateKeyspace。

`CreateTable`

以下程式碼範例顯示如何使用 `CreateTable`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " Define schema with columns
        DATA(lt_columns) = VALUE /aws1/
cl_kyscolumndefinition=>tt_columndefinitionlist(
          ( NEW /aws1/cl_kyscolumndefinition( iv_name = 'title' iv_type = 'text' ) )
          ( NEW /aws1/cl_kyscolumndefinition( iv_name = 'year' iv_type = 'int' ) )
          ( NEW /aws1/cl_kyscolumndefinition( iv_name = 'release_date' iv_type =
 'timestamp' ) )
          ( NEW /aws1/cl_kyscolumndefinition( iv_name = 'plot' iv_type = 'text' ) )
        ).

        " Define partition keys
        DATA(lt_partition_keys) = VALUE /aws1/
cl_kyspartitionkey=>tt_partitionkeylist(
          ( NEW /aws1/cl_kyspartitionkey( iv_name = 'year' ) )
          ( NEW /aws1/cl_kyspartitionkey( iv_name = 'title' ) )
```

```
        ).

        " Create schema definition
        DATA(lo_schema) = NEW /aws1/cl_kysschemadefinition(
          it_allcolumns = lt_columns
          it_partitionkeys = lt_partition_keys ).

        " Enable point-in-time recovery
        DATA(lo_pitr) = NEW /aws1/cl_kyspointintimerec(
          iv_status = 'ENABLED' ).

        oo_result = lo_kys->createtable(
          iv_keyspacename = iv_keyspace_name
          iv_tablename = iv_table_name
          io_schemadefinition = lo_schema
          io_pointintimerecovery = lo_pitr ).
        MESSAGE 'Table created successfully.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 CreateTable。

`DeleteKeyspace`

以下程式碼範例顯示如何使用 `DeleteKeyspace`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        lo_kys->deletekeyspace(
          iv_keyspacename = iv_keyspace_name ).
```

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteKeyspace。

`DeleteTable`

以下程式碼範例顯示如何使用 `DeleteTable`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        lo_kys->deletetable(
          iv_keyspacename = iv_keyspace_name
          iv_tablename = iv_table_name ).
        MESSAGE 'Table deleted successfully.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 DeleteTable。

`GetKeyspace`

以下程式碼範例顯示如何使用 `GetKeyspace`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        oo_result = lo_kys->getkeyspace(
          iv_keyspacename = iv_keyspace_name ).
        MESSAGE 'Keyspace retrieved successfully.' TYPE 'I'.
      CATCH /aws1/cx_kysresourcenotfoundex.
        MESSAGE 'Keyspace does not exist.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetKeyspace。

`GetTable`

以下程式碼範例顯示如何使用 `GetTable`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        oo_result = lo_kys->gettable(
          iv_keyspacename = iv_keyspace_name
          iv_tablename = iv_table_name ).
```

```
        MESSAGE 'Table information retrieved successfully.' TYPE 'I'.
      CATCH /aws1/cx_kysresourcenotfoundex.
        MESSAGE 'Table does not exist.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetTable。

`ListKeyspaces`

以下程式碼範例顯示如何使用 `ListKeyspaces`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        oo_result = lo_kys->listkeyspaces(
          iv_maxresults = iv_max_results ).
        MESSAGE 'Keyspaces listed successfully.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListKeyspaces。

`ListTables`

以下程式碼範例顯示如何使用 `ListTables`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        oo_result = lo_kys->listtables(
          iv_keyspacename = iv_keyspace_name ).
        MESSAGE 'Tables listed successfully.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 ListTables。

`RestoreTable`

以下程式碼範例顯示如何使用 `RestoreTable`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        oo_result = lo_kys->restoretable(
          iv_sourcekeyspacename = iv_source_keyspace_name
          iv_sourcetablename = iv_source_table_name
          iv_targetkeyspacename = iv_target_keyspace_name
          iv_targettablename = iv_target_table_name
          iv_restoretimestamp = iv_restore_timestamp ).
```

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 RestoreTable。

`UpdateTable`

以下程式碼範例顯示如何使用 `UpdateTable`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " Add a new column to track watched movies
        DATA(lt_add_columns) = VALUE /aws1/
cl_kyscolumndefinition=>tt_columndefinitionlist(
          ( NEW /aws1/cl_kyscolumndefinition( iv_name = 'watched' iv_type =
 'boolean' ) )
        ).

        oo_result = lo_kys->updatetable(
          iv_keyspacename = iv_keyspace_name
          iv_tablename = iv_table_name
          it_addcolumns = lt_add_columns ).
        MESSAGE 'Table updated successfully.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 UpdateTable。

使用適用於 SAP ABAP 的 SDK 的 Kinesis 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 Kinesis 來執行動作和實作常見案例。

基本概念是程式碼範例，這些範例說明如何在服務內執行基本操作。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 基本概念
- 動作

基本概念

了解基本概念

以下程式碼範例顯示做法：

- 建立串流，並將記錄放入其中。
- 建立碎片迭代器。
- 讀取記錄，然後清除資源。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
DATA lo_stream_describe_result TYPE REF TO /aws1/cl_knsdescrstreamoutput.
```

```
   DATA lo_stream_description TYPE REF TO /aws1/cl_knsstreamdescription.
   DATA lo_sharditerator TYPE REF TO /aws1/cl_knsgetsharditerator01.
   DATA lo_record_result TYPE REF TO /aws1/cl_knsputrecordoutput.

   "Create stream."
   TRY.
       lo_kns->createstream(
           iv_streamname = iv_stream_name
           iv_shardcount = iv_shard_count ).
       MESSAGE 'Stream created.' TYPE 'I'.
     CATCH /aws1/cx_knsinvalidargumentex.
       MESSAGE 'The specified argument was not valid.' TYPE 'E'.
     CATCH /aws1/cx_knslimitexceededex.
       MESSAGE 'The request processing has failed because of a limit exceeded
exception.' TYPE 'E'.
     CATCH /aws1/cx_knsresourceinuseex.
       MESSAGE 'The request processing has failed because the resource is in use.'
TYPE 'E'.
   ENDTRY.

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

   "Create record."
   TRY.
       lo_record_result = lo_kns->putrecord(
           iv_streamname = iv_stream_name
           iv_data       = iv_data
           iv_partitionkey = iv_partition_key ).
       MESSAGE 'Record created.' TYPE 'I'.
     CATCH /aws1/cx_knsinvalidargumentex.
       MESSAGE 'The specified argument was not valid.' TYPE 'E'.
     CATCH /aws1/cx_knskmsaccessdeniedex.
```

```
        MESSAGE 'You do not have permission to perform this AWS KMS action.' TYPE
'E'.
      CATCH /aws1/cx_knskmsdisabledex.
        MESSAGE 'KMS key used is disabled.' TYPE 'E'.
      CATCH /aws1/cx_knskmsinvalidstateex.
        MESSAGE 'KMS key used is in an invalid state. ' TYPE 'E'.
      CATCH /aws1/cx_knskmsnotfoundex.
        MESSAGE 'KMS key used is not found.' TYPE 'E'.
      CATCH /aws1/cx_knskmsoptinrequired.
        MESSAGE 'KMS key option is required.' TYPE 'E'.
      CATCH /aws1/cx_knskmsthrottlingex.
        MESSAGE 'The rate of requests to AWS KMS is exceeding the request quotas.'
TYPE 'E'.
      CATCH /aws1/cx_knsprovthruputexcdex.
        MESSAGE 'The request rate for the stream is too high, or the requested data
is too large for the available throughput.' TYPE 'E'.
      CATCH /aws1/cx_knsresourcenotfoundex.
        MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
    ENDTRY.

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

    "Read the record."
    TRY.
        oo_result = lo_kns->getrecords(                    " oo_result is returned
for testing purposes. "
            iv_sharditerator   = lo_sharditerator->get_sharditerator( ) ).
        MESSAGE 'Shard iterator created.' TYPE 'I'.
      CATCH /aws1/cx_knsexpirediteratorex.
        MESSAGE 'Iterator expired.' TYPE 'E'.
```

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

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

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的下列主題。
  - CreateStream
  - DeleteStream
  - GetRecords
  - GetShardIterator

- PutRecord

動作

`CreateStream`

以下程式碼範例顯示如何使用 `CreateStream`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   TRY.
       lo_kns->createstream(
           iv_streamname = iv_stream_name
           iv_shardcount = iv_shard_count ).
       MESSAGE 'Stream created.' TYPE 'I'.
     CATCH /aws1/cx_knsinvalidargumentex.
       MESSAGE 'The specified argument was not valid.' TYPE 'E'.
     CATCH /aws1/cx_knslimitexceededex.
       MESSAGE 'The request processing has failed because of a limit exceed
exception.' TYPE 'E'.
     CATCH /aws1/cx_knsresourceinuseex.
       MESSAGE 'The request processing has failed because the resource is in use.'
TYPE 'E'.
   ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 CreateStream。

`DeleteStream`

以下程式碼範例顯示如何使用 `DeleteStream`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 DeleteStream。

`DescribeStream`

以下程式碼範例顯示如何使用 `DescribeStream`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   TRY.
       oo_result = lo_kns->describestream(
           iv_streamname = iv_stream_name ).
       DATA(lt_stream_description) = oo_result->get_streamdescription( ).
```

```
        MESSAGE 'Streams retrieved.' TYPE 'I'.
      CATCH /aws1/cx_knslimitexceededex.
        MESSAGE 'The request processing has failed because of a limit exceed
exception.' TYPE 'E'.
      CATCH /aws1/cx_knsresourcenotfoundex.
        MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的 AWS SDK API 參考》中的 DescribeStream。

GetRecords

以下程式碼範例顯示如何使用 GetRecords。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        oo_result = lo_kns->getrecords(             " oo_result is returned for
testing purposes. "
            iv_sharditerator = iv_shard_iterator ).
        DATA(lt_records) = oo_result->get_records( ).
        MESSAGE 'Record retrieved.' TYPE 'I'.
      CATCH /aws1/cx_knsexpirediteratorex.
        MESSAGE 'Iterator expired.' TYPE 'E'.
      CATCH /aws1/cx_knsinvalidargumentex.
        MESSAGE 'The specified argument was not valid.' TYPE 'E'.
      CATCH /aws1/cx_knskmsaccessdeniedex.
        MESSAGE 'You do not have permission to perform this AWS KMS action.' TYPE
'E'.
      CATCH /aws1/cx_knskmsdisabledex.
        MESSAGE 'KMS key used is disabled.' TYPE 'E'.
      CATCH /aws1/cx_knskmsinvalidstateex.
        MESSAGE 'KMS key used is in an invalid state. ' TYPE 'E'.
```

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

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 GetRecords。

`ListStreams`

以下程式碼範例顯示如何使用 `ListStreams`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 ListStreams。

PutRecord

以下程式碼範例顯示如何使用 PutRecord。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

```
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 PutRecord。

`RegisterStreamConsumer`

以下程式碼範例顯示如何使用 `RegisterStreamConsumer`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   TRY.
       oo_result = lo_kns->registerstreamconsumer(       " oo_result is returned
for testing purposes. "
           iv_streamarn = iv_stream_arn
           iv_consumername = iv_consumer_name ).
       MESSAGE 'Stream consumer registered.' TYPE 'I'.
     CATCH /aws1/cx_knsinvalidargumentex.
       MESSAGE 'The specified argument was not valid.' TYPE 'E'.
     CATCH /aws1/cx_sgmresourcelimitexcd.
       MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
     CATCH /aws1/cx_sgmresourceinuse.
       MESSAGE 'Resource being accessed is in use.' TYPE 'E'.
     CATCH /aws1/cx_sgmresourcenotfound.
       MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
   ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的 AWS SDK API 參考》中的 RegisterStreamConsumer。

AWS KMS 使用適用於 SAP ABAP 的 SDK 的範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS 開發套件來執行動作和實作常見案例 AWS KMS。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 開始使用
- 動作

開始使用

您好 AWS KMS

下列程式碼範例示範如何開始使用 AWS Key Management Service。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_kms->listkeys( ).
    MESSAGE 'Retrieved KMS keys list.' TYPE 'I'.
  CATCH /aws1/cx_kmskmsinternalex.
    MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListKeys。

動作

`CreateAlias`

以下程式碼範例顯示如何使用 `CreateAlias`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_alias_name = 'alias/my-key-alias'
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        lo_kms->createalias(
          iv_aliasname = iv_alias_name
          iv_targetkeyid = iv_key_id
        ).
        MESSAGE 'Alias created successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsalreadyexistsex.
        MESSAGE 'Alias already exists.' TYPE 'E'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmsinvalidaliasnameex.
        MESSAGE 'Invalid alias name.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateAlias。

`CreateGrant`

以下程式碼範例顯示如何使用 `CreateGrant`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        " iv_grantee_principal = 'arn:aws:iam::123456789012:role/my-role'
        " it_operations contains 'Encrypt', 'Decrypt', 'GenerateDataKey'
        oo_result = lo_kms->creategrant(
          iv_keyid = iv_key_id
          iv_granteeprincipal = iv_grantee_principal
          it_operations = it_operations
        ).
        MESSAGE 'Grant created successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsdisabledexception.
        MESSAGE 'The key is disabled.' TYPE 'E'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateGrant。

`CreateKey`

以下程式碼範例顯示如何使用 `CreateKey`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_description = 'Created by the AWS SDK for SAP ABAP'
        oo_result = lo_kms->createkey( iv_description = iv_description ).
        MESSAGE 'KMS key created successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
      CATCH /aws1/cx_kmslimitexceededex.
        MESSAGE 'Limit exceeded for KMS resources.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateKey。

`Decrypt`

以下程式碼範例顯示如何使用 `Decrypt`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        " iv_ciphertext_blob contains the encrypted data
        oo_result = lo_kms->decrypt(
          iv_keyid = iv_key_id
          iv_ciphertextblob = iv_ciphertext_blob
        ).
        MESSAGE 'Text decrypted successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsdisabledexception.
        MESSAGE 'The key is disabled.' TYPE 'E'.
      CATCH /aws1/cx_kmsincorrectkeyex.
        MESSAGE 'Incorrect key for decryption.' TYPE 'E'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
```

```
    MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的解密。

DeleteAlias

以下程式碼範例顯示如何使用 DeleteAlias。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    " iv_alias_name = 'alias/my-key-alias'
    lo_kms->deletealias( iv_aliasname = iv_alias_name ).
    MESSAGE 'Alias deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_kmsnotfoundexception.
    MESSAGE 'Alias not found.' TYPE 'E'.
  CATCH /aws1/cx_kmskmsinternalex.
    MESSAGE 'An internal error occurred.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteAlias。

DescribeKey

以下程式碼範例顯示如何使用 DescribeKey。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        oo_result = lo_kms->describekey( iv_keyid = iv_key_id ).
        DATA(lo_key) = oo_result->get_keymetadata( ).
        MESSAGE 'Retrieved key information successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeKey。

`DisableKey`

以下程式碼範例顯示如何使用 `DisableKey`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        lo_kms->disablekey( iv_keyid = iv_key_id ).
```

```
        MESSAGE 'KMS key disabled successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DisableKey。

`EnableKey`

以下程式碼範例顯示如何使用 `EnableKey`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        lo_kms->enablekey( iv_keyid = iv_key_id ).
        MESSAGE 'KMS key enabled successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 EnableKey。

`EnableKeyRotation`

以下程式碼範例顯示如何使用 `EnableKeyRotation`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        lo_kms->enablekeyrotation( iv_keyid = iv_key_id ).
        MESSAGE 'Key rotation enabled successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsdisabledexception.
        MESSAGE 'The key is disabled.' TYPE 'E'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmsunsupportedopex.
        MESSAGE 'Operation not supported for this key.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 EnableKeyRotation。

`Encrypt`

以下程式碼範例顯示如何使用 `Encrypt`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
```

```
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        " iv_plaintext contains the data to encrypt
        oo_result = lo_kms->encrypt(
          iv_keyid = iv_key_id
          iv_plaintext = iv_plaintext
        ).
        MESSAGE 'Text encrypted successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsdisabledexception.
        MESSAGE 'The key is disabled.' TYPE 'E'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的加密。

`GenerateDataKey`

以下程式碼範例顯示如何使用 `GenerateDataKey`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        " iv_keyspec = 'AES_256'
        oo_result = lo_kms->generatedatakey(
          iv_keyid = iv_key_id
          iv_keyspec = 'AES_256'
        ).
        MESSAGE 'Data key generated successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsdisabledexception.
```

```
        MESSAGE 'The key is disabled.' TYPE 'E'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GenerateDataKey。

GetKeyPolicy

以下程式碼範例顯示如何使用 GetKeyPolicy。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        oo_result = lo_kms->getkeypolicy(
          iv_keyid = iv_key_id
          iv_policyname = 'default'
        ).
        MESSAGE 'Retrieved key policy successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetKeyPolicy。

`ListAliases`

以下程式碼範例顯示如何使用 `ListAliases`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        oo_result = lo_kms->listaliases( ).
        MESSAGE 'Retrieved KMS aliases list.' TYPE 'I'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListAliases。

`ListGrants`

以下程式碼範例顯示如何使用 `ListGrants`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        oo_result = lo_kms->listgrants( iv_keyid = iv_key_id ).
        MESSAGE 'Retrieved grants list.' TYPE 'I'.
```

```
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListGrants。

`ListKeyPolicies`

以下程式碼範例顯示如何使用 `ListKeyPolicies`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        oo_result = lo_kms->listkeypolicies( iv_keyid = iv_key_id ).
        MESSAGE 'Retrieved key policies list.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListKeyPolicies。

`ListKeys`

以下程式碼範例顯示如何使用 `ListKeys`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        oo_result = lo_kms->listkeys( ).
        MESSAGE 'Retrieved KMS keys list.' TYPE 'I'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListKeys。

`PutKeyPolicy`

以下程式碼範例顯示如何使用 `PutKeyPolicy`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        " iv_policy = '{"Version":"2012-10-17",        "Statement": [...]}'
        lo_kms->putkeypolicy(
          iv_keyid = iv_key_id
          iv_policyname = 'default'
          iv_policy = iv_policy
        ).
```

```
        MESSAGE 'Key policy updated successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmsmalformedplydocex.
        MESSAGE 'Malformed policy document.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 PutKeyPolicy。

`ReEncrypt`

以下程式碼範例顯示如何使用 `ReEncrypt`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_source_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        " iv_destination_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/5678dcba-56cd-78ef-90ab-5678901234cd'
        " iv_ciphertext_blob contains the encrypted data
        oo_result = lo_kms->reencrypt(
          iv_sourcekeyid = iv_source_key_id
          iv_destinationkeyid = iv_destination_key_id
          iv_ciphertextblob = iv_ciphertext_blob
        ).
        MESSAGE 'Ciphertext reencrypted successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsdisabledexception.
        MESSAGE 'The key is disabled.' TYPE 'E'.
      CATCH /aws1/cx_kmsincorrectkeyex.
        MESSAGE 'Incorrect source key for decryption.' TYPE 'E'.
      CATCH /aws1/cx_kmsnotfoundexception.
```

```
      MESSAGE 'Key not found.' TYPE 'E'.
    CATCH /aws1/cx_kmskmsinternalex.
      MESSAGE 'An internal error occurred.' TYPE 'E'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ReEncrypt。

`RetireGrant`

以下程式碼範例顯示如何使用 `RetireGrant`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
      " iv_grant_token = 'AQpAM2RhZ...'
      lo_kms->retiregrant( iv_granttoken = iv_grant_token ).
      MESSAGE 'Grant retired successfully.' TYPE 'I'.
    CATCH /aws1/cx_kmsnotfoundexception.
      MESSAGE 'Grant not found.' TYPE 'E'.
    CATCH /aws1/cx_kmsinvgranttokenex.
      MESSAGE 'Invalid grant token.' TYPE 'E'.
    CATCH /aws1/cx_kmskmsinternalex.
      MESSAGE 'An internal error occurred.' TYPE 'E'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 RetireGrant。

`RevokeGrant`

以下程式碼範例顯示如何使用 `RevokeGrant`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        " iv_grant_id = '1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p'
        lo_kms->revokegrant(
          iv_keyid = iv_key_id
          iv_grantid = iv_grant_id
        ).
        MESSAGE 'Grant revoked successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Grant or key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmsinvalidgrantidex.
        MESSAGE 'Invalid grant ID.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 RevokeGrant。

`ScheduleKeyDeletion`

以下程式碼範例顯示如何使用 `ScheduleKeyDeletion`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        " iv_pending_window_days = 7
        oo_result = lo_kms->schedulekeydeletion(
          iv_keyid = iv_key_id
          iv_pendingwindowindays = iv_pending_window_days
        ).
        MESSAGE 'Key scheduled for deletion.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ScheduleKeyDeletion。

`Sign`

以下程式碼範例顯示如何使用 `Sign`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab' (asymmetric key)
        " iv_message contains the message to sign
        " iv_signing_algorithm = 'RSASSA_PSS_SHA_256'
        oo_result = lo_kms->sign(
          iv_keyid = iv_key_id
          iv_message = iv_message
          iv_signingalgorithm = iv_signing_algorithm
        ).
```

```
        MESSAGE 'Message signed successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsdisabledexception.
        MESSAGE 'The key is disabled.' TYPE 'E'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmsinvalidkeyusageex.
        MESSAGE 'Key cannot be used for signing.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱登入適用於 AWS SAP ABAP 的 SDK API 參考。

`TagResource`

以下程式碼範例顯示如何使用 `TagResource`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    DATA lt_tags TYPE /aws1/cl_kmstag=>tt_taglist.

    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab'
        " iv_tag_key = 'Environment'
        " iv_tag_value = 'Production'
        APPEND NEW /aws1/cl_kmstag(
          iv_tagkey = iv_tag_key
          iv_tagvalue = iv_tag_value
        ) TO lt_tags.

        lo_kms->tagresource(
          iv_keyid = iv_key_id
          it_tags = lt_tags
        ).
```

```
        MESSAGE 'Tag added to KMS key successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmstagexception.
        MESSAGE 'Invalid tag format.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 TagResource。

`UpdateAlias`

以下程式碼範例顯示如何使用 `UpdateAlias`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_alias_name = 'alias/my-key-alias'
        " iv_target_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/5678dcba-56cd-78ef-90ab-5678901234cd'
        lo_kms->updatealias(
          iv_aliasname = iv_alias_name
          iv_targetkeyid = iv_target_key_id
        ).
        MESSAGE 'Alias updated successfully.' TYPE 'I'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Alias or key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 UpdateAlias。

`Verify`

以下程式碼範例顯示如何使用 `Verify`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab' (asymmetric key)
        " iv_message contains the original message
        " iv_signature contains the signature to verify
        " iv_signing_algorithm = 'RSASSA_PSS_SHA_256'
        oo_result = lo_kms->verify(
          iv_keyid = iv_key_id
          iv_message = iv_message
          iv_signature = iv_signature
          iv_signingalgorithm = iv_signing_algorithm
        ).
        DATA(lv_valid) = oo_result->get_signaturevalid( ).
        IF lv_valid = abap_true.
          MESSAGE 'Signature is valid.' TYPE 'I'.
        ELSE.
          MESSAGE 'Signature is invalid.' TYPE 'I'.
        ENDIF.
      CATCH /aws1/cx_kmsdisabledexception.
        MESSAGE 'The key is disabled.' TYPE 'E'.
      CATCH /aws1/cx_kmsnotfoundexception.
        MESSAGE 'Key not found.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinvalidsigex.
        MESSAGE 'Invalid signature.' TYPE 'E'.
      CATCH /aws1/cx_kmskmsinternalex.
        MESSAGE 'An internal error occurred.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的驗證。

使用適用於 SAP ABAP 的 SDK 的 Lambda 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 Lambda 來執行動作和實作常見案例。

基本概念是程式碼範例，這些範例說明如何在服務內執行基本操作。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 基本概念
- 動作

基本概念

了解基本概念

以下程式碼範例顯示做法：

- 建立 IAM 角色和 Lambda 函數，然後上傳處理常式程式碼。
- 調用具有單一參數的函數並取得結果。
- 更新函數程式碼並使用環境變數進行設定。
- 調用具有新參數的函數並取得結果。顯示傳回的執行日誌。
- 列出您帳戶的函數，然後清理相關資源。

如需詳細資訊，請參閱使用主控台建立 Lambda 函數。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   TRY.
       "Create an AWS Identity and Access Management (IAM) role that grants AWS
 Lambda permission to write to logs."
       DATA(lv_policy_document) = `{` &&
           `"Version":"2012-10-17",       ` &&
                 `"Statement": [` &&
                   `{` &&
                     `"Effect": "Allow",` &&
                     `"Action": [` &&
                       `"sts:AssumeRole"` &&
                     `],` &&
                     `"Principal": {` &&
                       `"Service": [` &&
                         `"lambda.amazonaws.com"` &&
                       `]` &&
                     `}` &&
                   `}` &&
                 `]` &&
               `}`.
       TRY.
           DATA(lo_create_role_output) = lo_iam->createrole(
                   iv_rolename = iv_role_name
                   iv_assumerolepolicydocument = lv_policy_document
                   iv_description = 'Grant lambda permission to write to logs' ).
           DATA(lv_role_arn) = lo_create_role_output->get_role( )->get_arn( ).
           MESSAGE 'IAM role created.' TYPE 'I'.
           WAIT UP TO 10 SECONDS.            " Make sure that the IAM role is ready
 for use. "
         CATCH /aws1/cx_iamentityalrdyexex.
           DATA(lo_role) = lo_iam->getrole( iv_rolename = iv_role_name ).
           lv_role_arn = lo_role->get_role( )->get_arn( ).
         CATCH /aws1/cx_iaminvalidinputex.
           MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
         CATCH /aws1/cx_iammalformedplydocex.
           MESSAGE 'Policy document in the request is malformed.' TYPE 'E'.
       ENDTRY.

       TRY.
           lo_iam->attachrolepolicy(
               iv_rolename  = iv_role_name
               iv_policyarn = 'arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole' ).
```

```
            MESSAGE 'Attached policy to the IAM role.' TYPE 'I'.
          CATCH /aws1/cx_iaminvalidinputex.
            MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
          CATCH /aws1/cx_iamnosuchentityex.
            MESSAGE 'The requested resource entity does not exist.' TYPE 'E'.
          CATCH /aws1/cx_iamplynotattachableex.
            MESSAGE 'Service role policies can only be attached to the service-
linked role for their service.' TYPE 'E'.
          CATCH /aws1/cx_iamunmodableentityex.
            MESSAGE 'Service that depends on the service-linked role is not
 modifiable.' TYPE 'E'.
        ENDTRY.

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

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

        "Invoke the function with a single parameter and get results."
        TRY.
            DATA(lv_json) = /aws1/cl_rt_util=>string_to_xstring(
```

```
              `{`  &&
                `"action": "increment",`  &&
                `"number": 10` &&
              `}` ).
            DATA(lo_initial_invoke_output) = lo_lmd->invoke(
                       iv_functionname = iv_function_name
                       iv_payload = lv_json ).
            ov_initial_invoke_payload = lo_initial_invoke_output->get_payload( ).
        " ov_initial_invoke_payload is returned for testing purposes. "
            DATA(lo_writer_json) = cl_sxml_string_writer=>create( type =
 if_sxml=>co_xt_json ).
            CALL TRANSFORMATION id SOURCE XML ov_initial_invoke_payload RESULT XML
 lo_writer_json.
            DATA(lv_result) = cl_abap_codepage=>convert_from( lo_writer_json-
>get_output( ) ).
            MESSAGE 'Lambda function invoked.' TYPE 'I'.
          CATCH /aws1/cx_lmdinvparamvalueex.
            MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
          CATCH /aws1/cx_lmdinvrequestcontex.
            MESSAGE 'Unable to parse request body as JSON.' TYPE 'E'.
          CATCH /aws1/cx_lmdresourcenotfoundex.
            MESSAGE 'The requested resource does not exist.' TYPE 'E'.
          CATCH /aws1/cx_lmdunsuppedmediatyp00.
            MESSAGE 'Invoke request body does not have JSON as its content type.'
 TYPE 'E'.
        ENDTRY.

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

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

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

        "Invoke the function with new parameters and get results. Display the
 execution log that's returned from the invocation."
        TRY.
            lv_json = /aws1/cl_rt_util=>string_to_xstring(
              `{`  &&
                `"action": "decrement",`  &&
                `"number": 10` &&
              `}` ).
            DATA(lo_updated_invoke_output) = lo_lmd->invoke(
                       iv_functionname = iv_function_name
                       iv_payload = lv_json ).
            ov_updated_invoke_payload = lo_updated_invoke_output->get_payload( ).
        " ov_updated_invoke_payload is returned for testing purposes. "
            lo_writer_json = cl_sxml_string_writer=>create( type =
 if_sxml=>co_xt_json ).
            CALL TRANSFORMATION id SOURCE XML ov_updated_invoke_payload RESULT XML
 lo_writer_json.
```

```
            lv_result = cl_abap_codepage=>convert_from( lo_writer_json-
>get_output( ) ).
            MESSAGE 'Lambda function invoked.' TYPE 'I'.
          CATCH /aws1/cx_lmdinvparamvalueex.
            MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
          CATCH /aws1/cx_lmdinvrequestcontex.
            MESSAGE 'Unable to parse request body as JSON.' TYPE 'E'.
          CATCH /aws1/cx_lmdresourcenotfoundex.
            MESSAGE 'The requested resource does not exist.' TYPE 'E'.
          CATCH /aws1/cx_lmdunsuppedmediatyp00.
            MESSAGE 'Invoke request body does not have JSON as its content type.'
 TYPE 'E'.
        ENDTRY.

        " List the functions for your account. "
        TRY.
            DATA(lo_list_output) = lo_lmd->listfunctions( ).
            DATA(lt_functions) = lo_list_output->get_functions( ).
            MESSAGE 'Retrieved list of Lambda functions.' TYPE 'I'.
          CATCH /aws1/cx_lmdinvparamvalueex.
            MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
        ENDTRY.

        " Delete the Lambda function. "
        TRY.
            lo_lmd->deletefunction( iv_functionname = iv_function_name ).
            MESSAGE 'Lambda function deleted.' TYPE 'I'.
          CATCH /aws1/cx_lmdinvparamvalueex.
            MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
          CATCH /aws1/cx_lmdresourcenotfoundex.
            MESSAGE 'The requested resource does not exist.' TYPE 'W'.
        ENDTRY.

        " Detach role policy. "
        TRY.
            lo_iam->detachrolepolicy(
                iv_rolename  = iv_role_name
                iv_policyarn = 'arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole' ).
            MESSAGE 'Detached policy from the IAM role.' TYPE 'I'.
          CATCH /aws1/cx_iaminvalidinputex.
            MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
          CATCH /aws1/cx_iamnosuchentityex.
            MESSAGE 'The requested resource entity does not exist.' TYPE 'W'.
```

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

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

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

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的下列主題。
  - CreateFunction
  - DeleteFunction
  - GetFunction
  - Invoke
  - ListFunctions
  - UpdateFunctionCode
  - UpdateFunctionConfiguration

動作

`CreateFunction`

以下程式碼範例顯示如何使用 `CreateFunction`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS 開發套件 API 參考》中的 CreateFunction。

`DeleteFunction`

以下程式碼範例顯示如何使用 `DeleteFunction`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS 開發套件API 參考》中的 DeleteFunction。

`GetFunction`

以下程式碼範例顯示如何使用 `GetFunction`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS 開發套件 API 參考》中的 GetFunction。

`Invoke`

以下程式碼範例顯示如何使用 `Invoke`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   TRY.
       DATA(lv_json) = /aws1/cl_rt_util=>string_to_xstring(
```

```
          `{`  &&
            `"action": "increment",`  &&
            `"number": 10` &&
          `}` ).
        oo_result = lo_lmd->invoke(                  " oo_result is returned for
testing purposes. "
                 iv_functionname = iv_function_name
                 iv_payload = lv_json ).
        MESSAGE 'Lambda function invoked.' TYPE 'I'.
      CATCH /aws1/cx_lmdinvparamvalueex.
        MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
      CATCH /aws1/cx_lmdinvrequestcontex.
        MESSAGE 'Unable to parse request body as JSON.' TYPE 'E'.
      CATCH /aws1/cx_lmdinvalidzipfileex.
        MESSAGE 'The deployment package could not be unzipped.' TYPE 'E'.
      CATCH /aws1/cx_lmdrequesttoolargeex.
        MESSAGE 'Invoke request body JSON input limit was exceeded by the request
payload.' TYPE 'E'.
      CATCH /aws1/cx_lmdresourceconflictex.
        MESSAGE 'Resource already exists or another operation is in progress.' TYPE
'E'.
      CATCH /aws1/cx_lmdresourcenotfoundex.
        MESSAGE 'The requested resource does not exist.' TYPE 'E'.
      CATCH /aws1/cx_lmdserviceexception.
        MESSAGE 'An internal problem was encountered by the AWS Lambda service.'
TYPE 'E'.
      CATCH /aws1/cx_lmdtoomanyrequestsex.
        MESSAGE 'The maximum request throughput was reached.' TYPE 'E'.
      CATCH /aws1/cx_lmdunsuppedmediatyp00.
        MESSAGE 'Invoke request body does not have JSON as its content type.' TYPE
'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 Invoke。

ListFunctions

以下程式碼範例顯示如何使用 ListFunctions。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 ListFunctions。

`UpdateFunctionCode`

以下程式碼範例顯示如何使用 `UpdateFunctionCode`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   TRY.
       oo_result = lo_lmd->updatefunctioncode(     " oo_result is returned for
testing purposes. "
```

```
              iv_functionname = iv_function_name
              iv_zipfile = io_zip_file ).

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

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 UpdateFunctionCode。

`UpdateFunctionConfiguration`

以下程式碼範例顯示如何使用 `UpdateFunctionConfiguration`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 UpdateFunctionConfiguration。

使用適用於 SAP ABAP 的 SDK 的組織範例

下列程式碼範例示範如何使用 AWS SDK for SAP ABAP with Organizations 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

AttachPolicy

以下程式碼範例顯示如何使用 AttachPolicy。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_org->attachpolicy(
      iv_policyid = iv_policy_id
      iv_targetid = iv_target_id ).
    MESSAGE 'Policy attached to target.' TYPE 'I'.
  CATCH /aws1/cx_orgaccessdeniedex.
    MESSAGE 'You do not have permission to attach the policy.' TYPE 'E'.
  CATCH /aws1/cx_orgpolicynotfoundex.
    MESSAGE 'The specified policy does not exist.' TYPE 'E'.
  CATCH /aws1/cx_orgtargetnotfoundex.
    MESSAGE 'The specified target does not exist.' TYPE 'E'.
  CATCH /aws1/cx_orgduplicateplyatta00.
    MESSAGE 'The policy is already attached to the target.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 AttachPolicy。

CreatePolicy

以下程式碼範例顯示如何使用 CreatePolicy。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        oo_result = lo_org->createpolicy(       " oo_result is returned for testing
purposes. "
          iv_name        = iv_policy_name
          iv_description = iv_policy_description
          iv_content     = iv_policy_content
          iv_type        = iv_policy_type ).
        MESSAGE 'Policy created.' TYPE 'I'.
      CATCH /aws1/cx_orgaccessdeniedex.
        MESSAGE 'You do not have permission to create a policy.' TYPE 'E'.
      CATCH /aws1/cx_orgduplicatepolicyex.
        MESSAGE 'A policy with this name already exists.' TYPE 'E'.
      CATCH /aws1/cx_orgmalformedplydocex.
        MESSAGE 'The policy content is malformed.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreatePolicy。

`DeletePolicy`

以下程式碼範例顯示如何使用 `DeletePolicy`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
```

```
        lo_org->deletepolicy(
          iv_policyid = iv_policy_id ).
        MESSAGE 'Policy deleted.' TYPE 'I'.
      CATCH /aws1/cx_orgaccessdeniedex.
        MESSAGE 'You do not have permission to delete the policy.' TYPE 'E'.
      CATCH /aws1/cx_orgpolicynotfoundex.
        MESSAGE 'The specified policy does not exist.' TYPE 'E'.
      CATCH /aws1/cx_orgpolicyinuseex.
        MESSAGE 'The policy is still attached to one or more targets.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeletePolicy。

`DescribePolicy`

以下程式碼範例顯示如何使用 `DescribePolicy`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        oo_result = lo_org->describepolicy(     " oo_result is returned for testing
purposes. "
          iv_policyid = iv_policy_id ).
        DATA(lo_policy) = oo_result->get_policy( ).
        MESSAGE 'Retrieved policy details.' TYPE 'I'.
      CATCH /aws1/cx_orgaccessdeniedex.
        MESSAGE 'You do not have permission to describe the policy.' TYPE 'E'.
      CATCH /aws1/cx_orgpolicynotfoundex.
        MESSAGE 'The specified policy does not exist.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribePolicy。

`DetachPolicy`

以下程式碼範例顯示如何使用 `DetachPolicy`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_org->detachpolicy(
      iv_policyid = iv_policy_id
      iv_targetid = iv_target_id ).
    MESSAGE 'Policy detached from target.' TYPE 'I'.
  CATCH /aws1/cx_orgaccessdeniedex.
    MESSAGE 'You do not have permission to detach the policy.' TYPE 'E'.
  CATCH /aws1/cx_orgpolicynotfoundex.
    MESSAGE 'The specified policy does not exist.' TYPE 'E'.
  CATCH /aws1/cx_orgtargetnotfoundex.
    MESSAGE 'The specified target does not exist.' TYPE 'E'.
  CATCH /aws1/cx_orgpolicynotattex.
    MESSAGE 'The policy is not attached to the target.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DetachPolicy。

`ListPolicies`

以下程式碼範例顯示如何使用 `ListPolicies`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   TRY.
       oo_result = lo_org->listpolicies(       " oo_result is returned for testing
purposes. "
         iv_filter = iv_filter ).
       DATA(lt_policies) = oo_result->get_policies( ).
       MESSAGE 'Retrieved list of policies.' TYPE 'I'.
     CATCH /aws1/cx_orgaccessdeniedex.
       MESSAGE 'You do not have permission to list policies.' TYPE 'E'.
     CATCH /aws1/cx_orgawsorgsnotinuseex.
       MESSAGE 'Your account is not a member of an organization.' TYPE 'E'.
   ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListPolicies。

使用適用於 SAP ABAP 的 SDK 的 Amazon Pinpoint 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 Amazon Pinpoint 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

`SendMessages`

以下程式碼範例顯示如何使用 `SendMessages`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

傳送電子郵件訊息。

```
" Build the addresses map from the list of to_addresses
DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
LOOP AT it_to_addresses INTO DATA(lo_address).
  INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
    key = lo_address->get_value( )
    value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'EMAIL' )
  ) INTO TABLE lt_addresses.
ENDLOOP.

" Send the email message
DATA(lo_result) = lo_ppt->sendmessages(
  iv_applicationid = iv_app_id
  io_messagerequest = NEW /aws1/cl_pptmessagerequest(
    it_addresses = lt_addresses
    io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
      io_emailmessage = NEW /aws1/cl_pptemailmessage(
        iv_fromaddress = iv_sender
        io_simpleemail = NEW /aws1/cl_pptsimpleemail(
          io_subject = NEW /aws1/cl_pptsimpleemailpart(
            iv_charset = iv_char_set
            iv_data = iv_subject
          )
          io_htmlpart = NEW /aws1/cl_pptsimpleemailpart(
            iv_charset = iv_char_set
            iv_data = iv_html_message
          )
          io_textpart = NEW /aws1/cl_pptsimpleemailpart(
            iv_charset = iv_char_set
            iv_data = iv_text_message
          )
        )
      )
    )
```

```
    )
  ).

  " Extract message IDs from response
  DATA(lo_message_response) = lo_result->get_messageresponse( ).
  ot_message_ids = lo_message_response->get_result( ).

  MESSAGE 'Email message sent successfully.' TYPE 'I'.
```

傳送一則 SMS 訊息。

```
  " Build the addresses map for the destination number
  DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
  INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
    key = iv_destination_number
    value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'SMS' )
  ) INTO TABLE lt_addresses.

  " Send the SMS message
  DATA(lo_result) = lo_ppt->sendmessages(
    iv_applicationid = iv_app_id
    io_messagerequest = NEW /aws1/cl_pptmessagerequest(
      it_addresses = lt_addresses
      io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
        io_smsmessage = NEW /aws1/cl_pptsmsmessage(
          iv_body = iv_message
          iv_messagetype = iv_message_type
          iv_originationnumber = iv_origination_number
        )
      )
    )
  ).

  " Extract message ID from response
  DATA(lo_message_response) = lo_result->get_messageresponse( ).
  DATA(lt_results) = lo_message_response->get_result( ).
  LOOP AT lt_results INTO DATA(ls_result).
    IF ls_result-key = iv_destination_number.
      ov_message_id = ls_result-value->get_messageid( ).
      EXIT.
    ENDIF.
  ENDLOOP.
```

```
    MESSAGE 'SMS message sent successfully.' TYPE 'I'.
```

使用現有的電子郵件範本，傳送電子郵件訊息。

```
    " Build the addresses map from the list of to_addresses
    DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
    LOOP AT it_to_addresses INTO DATA(lo_address).
      INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
        key = lo_address->get_value( )
        value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'EMAIL' )
      ) INTO TABLE lt_addresses.
    ENDLOOP.

    " Send the email message using a template
    DATA(lo_result) = lo_ppt->sendmessages(
      iv_applicationid = iv_app_id
      io_messagerequest = NEW /aws1/cl_pptmessagerequest(
        it_addresses = lt_addresses
        io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
          io_emailmessage = NEW /aws1/cl_pptemailmessage(
            iv_fromaddress = iv_sender
          )
        )
        io_templateconfiguration = NEW /aws1/cl_ppttemplateconf(
          io_emailtemplate = NEW /aws1/cl_ppttemplate(
            iv_name = iv_template_name
            iv_version = iv_template_version
          )
        )
      )
    ).

    " Extract message IDs from response
    DATA(lo_message_response) = lo_result->get_messageresponse( ).
    ot_message_ids = lo_message_response->get_result( ).

    MESSAGE 'Templated email message sent successfully.' TYPE 'I'.
```

使用現有的 SMS 範本傳送文字訊息。

```
" Build the addresses map for the destination number
DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
  key = iv_destination_number
  value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'SMS' )
) INTO TABLE lt_addresses.

" Send the SMS message using a template
DATA(lo_result) = lo_ppt->sendmessages(
  iv_applicationid = iv_app_id
  io_messagerequest = NEW /aws1/cl_pptmessagerequest(
    it_addresses = lt_addresses
    io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
      io_smsmessage = NEW /aws1/cl_pptsmsmessage(
        iv_messagetype = iv_message_type
        iv_originationnumber = iv_origination_number
      )
    )
    io_templateconfiguration = NEW /aws1/cl_ppttemplateconf(
      io_smstemplate = NEW /aws1/cl_ppttemplate(
        iv_name = iv_template_name
        iv_version = iv_template_version
      )
    )
  )
).

" Extract message ID from response
DATA(lo_message_response) = lo_result->get_messageresponse( ).
DATA(lt_results) = lo_message_response->get_result( ).
LOOP AT lt_results INTO DATA(ls_result).
  IF ls_result-key = iv_destination_number.
    ov_message_id = ls_result-value->get_messageid( ).
    EXIT.
  ENDIF.
ENDLOOP.

MESSAGE 'Templated SMS message sent successfully.' TYPE 'I'.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 SendMessages。

使用適用於 SAP ABAP 的 SDK 的 Amazon Pinpoint SMS 和語音 API 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 Amazon Pinpoint SMS 和語音 API 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

`CreateConfigurationSet`

以下程式碼範例顯示如何使用 `CreateConfigurationSet`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " Create a new configuration set
        lo_pps->createconfigurationset(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-config-
set'
        ).

        MESSAGE 'Configuration set created successfully.' TYPE 'I'.

      CATCH /aws1/cx_ppsalreadyexistsex INTO DATA(lo_already_exists_ex).
        MESSAGE lo_already_exists_ex->get_text( ) TYPE 'I'.
```

```
        RAISE EXCEPTION lo_already_exists_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppslimitexceededex INTO DATA(lo_limit_exceeded_ex).
        MESSAGE lo_limit_exceeded_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_limit_exceeded_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateConfigurationSet。

`CreateConfigurationSetEventDestination`

以下程式碼範例顯示如何使用 `CreateConfigurationSetEventDestination`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " Create event destination for the configuration set
        lo_pps->createconfseteventdst(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-config-
set'
          iv_eventdestinationname = iv_event_destination_name    " e.g., 'my-event-
dest'
          io_eventdestination = io_event_destination
        ).
```

```
      MESSAGE 'Event destination created successfully.' TYPE 'I'.

    CATCH /aws1/cx_ppsalreadyexistsex INTO DATA(lo_already_exists_ex).
      MESSAGE lo_already_exists_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_already_exists_ex.
    CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
      MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_not_found_ex.
    CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
      MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_bad_request_ex.
    CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
      MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_internal_error_ex.
    CATCH /aws1/cx_ppslimitexceededex INTO DATA(lo_limit_exceeded_ex).
      MESSAGE lo_limit_exceeded_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_limit_exceeded_ex.
    CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
      MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_too_many_requests_ex.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateConfigurationSetEventDestination。

`DeleteConfigurationSet`

以下程式碼範例顯示如何使用 `DeleteConfigurationSet`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
      " Delete the configuration set
      lo_pps->deleteconfigurationset(
```

```
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-config-
set'
        ).

        MESSAGE 'Configuration set deleted successfully.' TYPE 'I'.

      CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
        MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_not_found_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteConfigurationSet。

`DeleteConfigurationSetEventDestination`

以下程式碼範例顯示如何使用 `DeleteConfigurationSetEventDestination`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " Delete the event destination
        lo_pps->deleteconfseteventdst(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-config-
set'
```

```
          iv_eventdestinationname = iv_event_destination_name    " e.g., 'my-event-
dest'
        ).

        MESSAGE 'Event destination deleted successfully.' TYPE 'I'.

      CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
        MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_not_found_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteConfigurationSetEventDestination。

GetConfigurationSetEventDestinations

以下程式碼範例顯示如何使用 GetConfigurationSetEventDestinations。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " Get event destinations for the configuration set
        oo_result = lo_pps->getconfseteventdestinations(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-config-
set'
```

```
        ).

        " Process the event destinations
        LOOP AT oo_result->get_eventdestinations( ) INTO DATA(lo_event_dest).
          DATA(lv_dest_name) = lo_event_dest->get_name( ).
          DATA(lv_enabled) = lo_event_dest->get_enabled( ).

          MESSAGE |Event destination: { lv_dest_name }, Enabled: { lv_enabled }|
TYPE 'I'.

          " Check for CloudWatch Logs destination
          DATA(lo_cloudwatch_dest) = lo_event_dest->get_cloudwatchlogsdst( ).
          IF lo_cloudwatch_dest IS NOT INITIAL.
            DATA(lv_log_group_arn) = lo_cloudwatch_dest->get_loggrouparn( ).
            MESSAGE |  CloudWatch Logs destination: { lv_log_group_arn }| TYPE 'I'.
          ENDIF.

          " Check for Kinesis Firehose destination
          DATA(lo_firehose_dest) = lo_event_dest->get_kinesisfirehosedst( ).
          IF lo_firehose_dest IS NOT INITIAL.
            DATA(lv_delivery_stream) = lo_firehose_dest->get_deliverystreamarn( ).
            MESSAGE |  Kinesis Firehose destination: { lv_delivery_stream }| TYPE
'I'.
          ENDIF.

          " Check for SNS destination
          DATA(lo_sns_dest) = lo_event_dest->get_snsdestination( ).
          IF lo_sns_dest IS NOT INITIAL.
            DATA(lv_topic_arn) = lo_sns_dest->get_topicarn( ).
            MESSAGE |  SNS destination: { lv_topic_arn }| TYPE 'I'.
          ENDIF.
        ENDLOOP.

      CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
        MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_not_found_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
```

```
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetConfigurationSetEventDestinations。

`ListConfigurationSets`

以下程式碼範例顯示如何使用 `ListConfigurationSets`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " List all configuration sets
        oo_result = lo_pps->listconfigurationsets(
          iv_nexttoken = iv_next_token    " Optional: Token for pagination
          iv_pagesize = iv_page_size      " Optional: Number of results per page,
e.g., '10'
        ).

        " Process the configuration sets
        LOOP AT oo_result->get_configurationsets( ) INTO DATA(lo_config_set).
          DATA(lv_config_set_name) = lo_config_set->get_value( ).
          MESSAGE |Configuration set: { lv_config_set_name }| TYPE 'I'.
        ENDLOOP.

        " Check if there are more results
        DATA(lv_next_token) = oo_result->get_nexttoken( ).
        IF lv_next_token IS NOT INITIAL.
          MESSAGE |More results available. Next token: { lv_next_token }| TYPE 'I'.
        ENDIF.

      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
```

```
      RAISE EXCEPTION lo_bad_request_ex.
    CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
      MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_internal_error_ex.
    CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
      MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_too_many_requests_ex.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListConfigurationSets。

SendVoiceMessage

以下程式碼範例顯示如何使用 SendVoiceMessage。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
      " Create SSML message type object with voice parameters
      DATA(lo_ssml_message) = NEW /aws1/cl_ppsssmlmessagetype(
        iv_languagecode = iv_language_code    " e.g., 'en-US'
        iv_voiceid = iv_voice_id              " e.g., 'Matthew'
        iv_text = iv_ssml_message             " SSML formatted message text
      ).

      " Create voice message content with the SSML message
      DATA(lo_content) = NEW /aws1/cl_ppsvoicemessagecont(
        io_ssmlmessage = lo_ssml_message
      ).

      " Send the voice message
      DATA(lo_result) = lo_pps->sendvoicemessage(
        iv_originationphonenumber = iv_origination_number  " e.g., '+12065550110'
```

```
        iv_callerid = iv_caller_id                          " e.g., '+12065550199'
        iv_destinationphonenumber = iv_destination_number  " e.g., '+12065550142'
        io_content = lo_content
      ).

      " Retrieve the message ID from the response
      ov_message_id = lo_result->get_messageid( ).

      MESSAGE 'Voice message sent successfully.' TYPE 'I'.

    CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
      MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_bad_request_ex.
    CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
      MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_internal_error_ex.
    CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
      MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_too_many_requests_ex.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 SendVoiceMessage。

`UpdateConfigurationSetEventDestination`

以下程式碼範例顯示如何使用 `UpdateConfigurationSetEventDestination`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
      " Update the event destination
      lo_pps->updateconfseteventdst(
```

```
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-config-
set'
          iv_eventdestinationname = iv_event_destination_name    " e.g., 'my-event-
dest'
          io_eventdestination = io_event_destination
        ).

        MESSAGE 'Event destination updated successfully.' TYPE 'I'.

      CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
        MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_not_found_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 UpdateConfigurationSetEventDestination。

使用適用於 SAP ABAP 的 SDK 的 Amazon Polly 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 Amazon Polly 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

`DeleteLexicon`

以下程式碼範例顯示如何使用 `DeleteLexicon`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_ply->deletelexicon( iv_name ).
    MESSAGE 'Lexicon deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_plylexiconnotfoundex.
    MESSAGE 'Lexicon not found.' TYPE 'E'.
  CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteLexicon。

`DescribeVoices`

以下程式碼範例顯示如何使用 `DescribeVoices`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    " Only pass optional parameters if they have values
```

```
    IF iv_engine IS NOT INITIAL AND iv_language IS NOT INITIAL.
      oo_result = lo_ply->describevoices(
        iv_engine = iv_engine
        iv_languagecode = iv_language ).
    ELSEIF iv_engine IS NOT INITIAL.
      oo_result = lo_ply->describevoices(
        iv_engine = iv_engine ).
    ELSEIF iv_language IS NOT INITIAL.
      oo_result = lo_ply->describevoices(
        iv_languagecode = iv_language ).
    ELSE.
      oo_result = lo_ply->describevoices( ).
    ENDIF.
    MESSAGE 'Retrieved voice metadata.' TYPE 'I'.
  CATCH /aws1/cx_plyinvalidnexttokenex.
    MESSAGE 'The NextToken is invalid.' TYPE 'E'.
  CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeVoices。

`GetLexicon`

以下程式碼範例顯示如何使用 `GetLexicon`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_ply->getlexicon( iv_name ).
    DATA(lo_lexicon) = oo_result->get_lexicon( ).
    IF lo_lexicon IS BOUND.
      DATA(lv_lex_name) = lo_lexicon->get_name( ).
```

```
        MESSAGE |Retrieved lexicon: { lv_lex_name }| TYPE 'I'.
      ENDIF.
    CATCH /aws1/cx_plylexiconnotfoundex.
      MESSAGE 'Lexicon not found.' TYPE 'E'.
    CATCH /aws1/cx_plyservicefailureex.
      MESSAGE 'Service failure occurred.' TYPE 'E'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetLexicon。

`GetSpeechSynthesisTask`

以下程式碼範例顯示如何使用 `GetSpeechSynthesisTask`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
      oo_result = lo_ply->getspeechsynthesistask( iv_task_id ).
      DATA(lo_task) = oo_result->get_synthesistask( ).
      IF lo_task IS BOUND.
        DATA(lv_status) = lo_task->get_taskstatus( ).
        MESSAGE |Task status: { lv_status }| TYPE 'I'.
      ENDIF.
    CATCH /aws1/cx_plyinvalidtaskidex.
      MESSAGE 'Invalid task ID.' TYPE 'E'.
    CATCH /aws1/cx_plyservicefailureex.
      MESSAGE 'Service failure occurred.' TYPE 'E'.
    CATCH /aws1/cx_plysynthesistsknotf00.
      MESSAGE 'Synthesis task not found.' TYPE 'E'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetSpeechSynthesisTask。

`ListLexicons`

以下程式碼範例顯示如何使用 `ListLexicons`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_ply->listlexicons( ).
    DATA(lt_lexicons) = oo_result->get_lexicons( ).
    DATA(lv_count) = lines( lt_lexicons ).
    MESSAGE |Found { lv_count } lexicons| TYPE 'I'.
  CATCH /aws1/cx_plyinvalidnexttokenex.
    MESSAGE 'Invalid NextToken.' TYPE 'E'.
  CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListLexicons。

`ListSpeechSynthesisTasks`

以下程式碼範例顯示如何使用 `ListSpeechSynthesisTasks`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
```

```
      " Only pass optional parameters if they have values
      IF iv_max_results IS NOT INITIAL AND iv_status IS NOT INITIAL.
        oo_result = lo_ply->listspeechsynthesistasks(
          iv_maxresults = iv_max_results
          iv_status = iv_status ).
      ELSEIF iv_max_results IS NOT INITIAL.
        oo_result = lo_ply->listspeechsynthesistasks(
          iv_maxresults = iv_max_results ).
      ELSEIF iv_status IS NOT INITIAL.
        oo_result = lo_ply->listspeechsynthesistasks(
          iv_status = iv_status ).
      ELSE.
        oo_result = lo_ply->listspeechsynthesistasks( ).
      ENDIF.
      DATA(lt_tasks) = oo_result->get_synthesistasks( ).
      DATA(lv_count) = lines( lt_tasks ).
      MESSAGE |Found { lv_count } synthesis tasks| TYPE 'I'.
    CATCH /aws1/cx_plyinvalidnexttokenex.
      MESSAGE 'Invalid NextToken.' TYPE 'E'.
    CATCH /aws1/cx_plyservicefailureex.
      MESSAGE 'Service failure occurred.' TYPE 'E'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListSpeechSynthesisTasks。

PutLexicon

以下程式碼範例顯示如何使用 PutLexicon。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
      lo_ply->putlexicon(
```

```
        iv_name = iv_name
        iv_content = iv_content ).
      MESSAGE 'Lexicon created successfully.' TYPE 'I'.
    CATCH /aws1/cx_plyinvalidlexiconex.
      MESSAGE 'Invalid lexicon.' TYPE 'E'.
    CATCH /aws1/cx_plylexiconsizeexcdex.
      MESSAGE 'Lexicon size exceeded.' TYPE 'E'.
    CATCH /aws1/cx_plymaxlexemelengthe00.
      MESSAGE 'Maximum lexeme length exceeded.' TYPE 'E'.
    CATCH /aws1/cx_plymaxlexiconsnoexc00.
      MESSAGE 'Maximum number of lexicons exceeded.' TYPE 'E'.
    CATCH /aws1/cx_plyservicefailureex.
      MESSAGE 'Service failure occurred.' TYPE 'E'.
    CATCH /aws1/cx_plyunsuppedplsalpha00.
      MESSAGE 'Unsupported PLS alphabet.' TYPE 'E'.
    CATCH /aws1/cx_plyunsuppedplslangu00.
      MESSAGE 'Unsupported PLS language.' TYPE 'E'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 PutLexicon。

`StartSpeechSynthesisTask`

以下程式碼範例顯示如何使用 `StartSpeechSynthesisTask`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
      " Only pass optional parameters if they have values
      IF iv_lang_code IS NOT INITIAL AND iv_s3_key_prefix IS NOT INITIAL.
        oo_result = lo_ply->startspeechsynthesistask(
          iv_engine = iv_engine
          iv_outputformat = iv_audio_format
          iv_outputs3bucketname = iv_s3_bucket
          iv_outputs3keyprefix = iv_s3_key_prefix
```

```
            iv_text = iv_text
            iv_voiceid = iv_voice_id
            iv_languagecode = iv_lang_code ).
        ELSEIF iv_lang_code IS NOT INITIAL.
          oo_result = lo_ply->startspeechsynthesistask(
            iv_engine = iv_engine
            iv_outputformat = iv_audio_format
            iv_outputs3bucketname = iv_s3_bucket
            iv_text = iv_text
            iv_voiceid = iv_voice_id
            iv_languagecode = iv_lang_code ).
        ELSEIF iv_s3_key_prefix IS NOT INITIAL.
          oo_result = lo_ply->startspeechsynthesistask(
            iv_engine = iv_engine
            iv_outputformat = iv_audio_format
            iv_outputs3bucketname = iv_s3_bucket
            iv_outputs3keyprefix = iv_s3_key_prefix
            iv_text = iv_text
            iv_voiceid = iv_voice_id ).
        ELSE.
          oo_result = lo_ply->startspeechsynthesistask(
            iv_engine = iv_engine
            iv_outputformat = iv_audio_format
            iv_outputs3bucketname = iv_s3_bucket
            iv_text = iv_text
            iv_voiceid = iv_voice_id ).
        ENDIF.
        MESSAGE 'Speech synthesis task started.' TYPE 'I'.
      CATCH /aws1/cx_plyinvalids3bucketex.
        MESSAGE 'Invalid S3 bucket.' TYPE 'E'.
      CATCH /aws1/cx_plyinvalidssmlex.
        MESSAGE 'Invalid SSML.' TYPE 'E'.
      CATCH /aws1/cx_plylexiconnotfoundex.
        MESSAGE 'Lexicon not found.' TYPE 'E'.
      CATCH /aws1/cx_plyservicefailureex.
        MESSAGE 'Service failure occurred.' TYPE 'E'.
      CATCH /aws1/cx_plytextlengthexcdex.
        MESSAGE 'Text length exceeded maximum.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 StartSpeechSynthesisTask。

SynthesizeSpeech

以下程式碼範例顯示如何使用 SynthesizeSpeech。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    " Only pass optional language code if it has a value
    IF iv_lang_code IS NOT INITIAL.
      oo_result = lo_ply->synthesizespeech(
        iv_engine = iv_engine
        iv_outputformat = iv_output_fmt
        iv_text = iv_text
        iv_voiceid = iv_voice_id
        iv_languagecode = iv_lang_code ).
    ELSE.
      oo_result = lo_ply->synthesizespeech(
        iv_engine = iv_engine
        iv_outputformat = iv_output_fmt
        iv_text = iv_text
        iv_voiceid = iv_voice_id ).
    ENDIF.
    MESSAGE 'Speech synthesized successfully.' TYPE 'I'.
  CATCH /aws1/cx_plyinvalidssmlex.
    MESSAGE 'Invalid SSML.' TYPE 'E'.
  CATCH /aws1/cx_plylexiconnotfoundex.
    MESSAGE 'Lexicon not found.' TYPE 'E'.
  CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
  CATCH /aws1/cx_plytextlengthexcdex.
    MESSAGE 'Text length exceeded maximum.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 SynthesizeSpeech。

使用適用於 SAP ABAP 的 SDK 的 Amazon RDS 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 Amazon RDS 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

`CreateDBParameterGroup`

以下程式碼範例顯示如何使用 `CreateDBParameterGroup`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
" iv_dbparametergroupname   = 'mydbparametergroup'
" iv_dbparametergroupfamily = 'mysql8.0'
" iv_description            = 'My custom DB parameter group for MySQL 8.0'
TRY.
    oo_result = lo_rds->createdbparametergroup(
      iv_dbparametergroupname   = iv_dbparametergroupname
      iv_dbparametergroupfamily = iv_dbparametergroupfamily
      iv_description            = iv_description ).
    MESSAGE 'DB parameter group created.' TYPE 'I'.
  CATCH /aws1/cx_rdsdbparmgralrexfault.
    MESSAGE 'DB parameter group already exists.' TYPE 'I'.
  CATCH /aws1/cx_rdsdbprmgrquotaexcd00.
```

```
        MESSAGE 'DB parameter group quota exceeded.' TYPE 'I'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateDBParameterGroup。

`DeleteDBParameterGroup`

以下程式碼範例顯示如何使用 `DeleteDBParameterGroup`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
" iv_dbparametergroupname = 'mydbparametergroup'
TRY.
    lo_rds->deletedbparametergroup(
      iv_dbparametergroupname = iv_dbparametergroupname ).
    MESSAGE 'DB parameter group deleted.' TYPE 'I'.
  CATCH /aws1/cx_rdsdbprmgrnotfndfault.
    MESSAGE 'DB parameter group not found.' TYPE 'I'.
  CATCH /aws1/cx_rdsinvdbprmgrstatef00.
    MESSAGE 'DB parameter group is in an invalid state.' TYPE 'I'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteDBParameterGroup。

`DescribeDBEngineVersions`

以下程式碼範例顯示如何使用 `DescribeDBEngineVersions`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeDBEngineVersions。

`DescribeDBParameterGroups`

以下程式碼範例顯示如何使用 `DescribeDBParameterGroups`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    " iv_dbparametergroupname = 'mydbparametergroup'
    TRY.
        oo_result = lo_rds->describedbparametergroups(
          iv_dbparametergroupname = iv_dbparametergroupname ).
```

```
    MESSAGE 'DB parameter group retrieved.' TYPE 'I'.
  CATCH /aws1/cx_rdsdbprmgrnotfndfault.
    MESSAGE 'DB parameter group not found.' TYPE 'I'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeDBParameterGroups。

`DescribeDBParameters`

以下程式碼範例顯示如何使用 `DescribeDBParameters`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
" iv_dbparametergroupname = 'mydbparametergroup'
" iv_source               = 'user' (optional - filters by parameter source)
TRY.
    oo_result = lo_rds->describedbparameters(
      iv_dbparametergroupname = iv_dbparametergroupname
      iv_source               = iv_source ).
    DATA(lv_param_count) = lines( oo_result->get_parameters( ) ).
    MESSAGE |Retrieved { lv_param_count } parameters.| TYPE 'I'.
  CATCH /aws1/cx_rdsdbprmgrnotfndfault.
    MESSAGE 'DB parameter group not found.' TYPE 'I'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeDBParameters。

`DescribeOrderableDBInstanceOptions`

以下程式碼範例顯示如何使用 `DescribeOrderableDBInstanceOptions`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeOrderableDBInstanceOptions。

`ModifyDBParameterGroup`

以下程式碼範例顯示如何使用 `ModifyDBParameterGroup`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    " iv_dbparametergroupname = 'mydbparametergroup'
    " it_parameters - table containing parameter objects with:
    "   - parametername = 'max_connections'
    "   - parametervalue = '100'
```

```
"   - applymethod = 'immediate' or 'pending-reboot'
TRY.
    oo_result = lo_rds->modifydbparametergroup(
      iv_dbparametergroupname = iv_dbparametergroupname
      it_parameters           = it_parameters ).
    MESSAGE 'DB parameter group modified.' TYPE 'I'.
  CATCH /aws1/cx_rdsdbprmgrnotfndfault.
    MESSAGE 'DB parameter group not found.' TYPE 'I'.
  CATCH /aws1/cx_rdsinvdbprmgrstatef00.
    MESSAGE 'DB parameter group is in an invalid state.' TYPE 'I'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ModifyDBParameterGroup。

使用適用於 SAP ABAP 的 SDK 的 Amazon Redshift 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 Amazon Redshift 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

`CreateCluster`

以下程式碼範例顯示如何使用 `CreateCluster`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

建立 叢集

```
TRY.
    " Example values: iv_cluster_identifier = 'my-redshift-cluster'
    " Example values: iv_node_type = 'ra3.4xlarge'
    " Example values: iv_master_username = 'awsuser'
    " Example values: iv_master_password = 'AwsUser1000'
    " Example values: iv_publicly_accessible = abap_true
    " Example values: iv_number_of_nodes = 2
    oo_result = lo_rsh->createcluster(
      iv_clusteridentifier = iv_cluster_identifier
      iv_nodetype = iv_node_type
      iv_masterusername = iv_master_username
      iv_masteruserpassword = iv_master_password
      iv_publiclyaccessible = iv_publicly_accessible
      iv_numberofnodes = iv_number_of_nodes
    ).
    MESSAGE 'Redshift cluster created successfully.' TYPE 'I'.
  CATCH /aws1/cx_rshclustalrdyexfault.
    MESSAGE 'Cluster already exists.' TYPE 'I'.
  CATCH /aws1/cx_rshclstquotaexcdfault.
    MESSAGE 'Cluster quota exceeded.' TYPE 'I'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateCluster。

`DeleteCluster`

以下程式碼範例顯示如何使用 `DeleteCluster`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

刪除叢集。

```
TRY.
    " Example values: iv_cluster_identifier = 'my-redshift-cluster'
    lo_rsh->deletecluster(
      iv_clusteridentifier = iv_cluster_identifier
      iv_skipfinalclustersnapshot = abap_true
    ).
    MESSAGE 'Redshift cluster deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_rshclustnotfoundfault.
    MESSAGE 'Cluster not found.' TYPE 'I'.
  CATCH /aws1/cx_rshinvcluststatefault.
    MESSAGE 'Invalid cluster state for deletion.' TYPE 'I'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteCluster。

`DescribeClusters`

以下程式碼範例顯示如何使用 `DescribeClusters`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

描述叢集。

```
TRY.
```

```
    " Example values: iv_cluster_identifier = 'my-redshift-cluster' (optional)
    oo_result = lo_rsh->describeclusters(
      iv_clusteridentifier = iv_cluster_identifier
    ).
    lt_clusters = oo_result->get_clusters( ).
    lv_cluster_count = lines( lt_clusters ).
    MESSAGE |Retrieved { lv_cluster_count } cluster(s).| TYPE 'I'.
  CATCH /aws1/cx_rshclustnotfoundfault.
    MESSAGE 'Cluster not found.' TYPE 'I'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeClusters。

`DescribeStatement`

以下程式碼範例顯示如何使用 `DescribeStatement`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    " Example values: iv_statement_id = 'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx'
    oo_result = lo_rsd->describestatement(
      iv_id = iv_statement_id
    ).
    lv_status = oo_result->get_status( ).
    MESSAGE |Statement status: { lv_status }| TYPE 'I'.
  CATCH /aws1/cx_rsdresourcenotfoundex.
    MESSAGE 'Statement not found.' TYPE 'I'.
  CATCH /aws1/cx_rsdinternalserverex.
    MESSAGE 'Internal server error.' TYPE 'I'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeStatement。

`ExecuteStatement`

以下程式碼範例顯示如何使用 `ExecuteStatement`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " Example values: iv_cluster_identifier = 'redshift-cluster-movies'
        " Example values: iv_database_name = 'dev'
        " Example values: iv_user_name = 'awsuser'
        " Example values: iv_sql = 'SELECT * FROM movies WHERE year = :year'
        " Example values: it_parameter_list - SQL parameters for parameterized
queries

        " Only pass parameters if the list is not empty
        IF it_parameter_list IS NOT INITIAL.
          oo_result = lo_rsd->executestatement(
            iv_clusteridentifier = iv_cluster_identifier
            iv_database = iv_database_name
            iv_dbuser = iv_user_name
            iv_sql = iv_sql
            it_parameters = it_parameter_list
          ).
        ELSE.
          oo_result = lo_rsd->executestatement(
            iv_clusteridentifier = iv_cluster_identifier
            iv_database = iv_database_name
            iv_dbuser = iv_user_name
            iv_sql = iv_sql
          ).
        ENDIF.

        lv_statement_id = oo_result->get_id( ).
```

```
      MESSAGE |Statement executed. ID: { lv_statement_id }| TYPE 'I'.
    CATCH /aws1/cx_rsdexecutestatementex.
      MESSAGE 'Statement execution error.' TYPE 'I'.
    CATCH /aws1/cx_rsdresourcenotfoundex.
      MESSAGE 'Resource not found.' TYPE 'I'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ExecuteStatement。

`GetStatementResult`

以下程式碼範例顯示如何使用 `GetStatementResult`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

檢查陳述式結果。

```
  TRY.
      " Example values: iv_statement_id = 'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx'
      " Handle pagination for large result sets

      DO.
        lo_result_page = lo_rsd->getstatementresult(
          iv_id = iv_statement_id
          iv_nexttoken = lv_next_token
        ).

        " Collect records from this page
        lt_page_records = lo_result_page->get_records( ).
        APPEND LINES OF lt_page_records TO lt_all_records.

        " Check if there are more pages
        lv_next_token = lo_result_page->get_nexttoken( ).
```

```
        IF lv_next_token IS INITIAL.
          EXIT. " No more pages
        ENDIF.
      ENDDO.

      " For the last call, set oo_result for return value
      oo_result = lo_result_page.
      lv_record_count = lines( lt_all_records ).
      MESSAGE |Retrieved { lv_record_count } record(s).| TYPE 'I'.
    CATCH /aws1/cx_rsdresourcenotfoundex.
      MESSAGE 'Statement not found or results not available.' TYPE 'I'.
    CATCH /aws1/cx_rsdinternalserverex.
      MESSAGE 'Internal server error.' TYPE 'I'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetStatementResult。

`ListDatabases`

以下程式碼範例顯示如何使用 `ListDatabases`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
      " Example values: iv_cluster_identifier = 'redshift-cluster-movies'
      " Example values: iv_database_name = 'dev'
      " Example values: iv_database_user = 'awsuser'
      oo_result = lo_rsd->listdatabases(
        iv_clusteridentifier = iv_cluster_identifier
        iv_database = iv_database_name
        iv_dbuser = iv_database_user
      ).
      lt_databases = oo_result->get_databases( ).
```

```
      lv_db_count = lines( lt_databases ).
      MESSAGE |Retrieved { lv_db_count } database(s).| TYPE 'I'.
    CATCH /aws1/cx_rsddatabaseconnex.
      MESSAGE 'Database connection error.' TYPE 'I'.
    CATCH /aws1/cx_rsdresourcenotfoundex.
      MESSAGE 'Cluster not found.' TYPE 'I'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListDatabases。

`ModifyCluster`

以下程式碼範例顯示如何使用 `ModifyCluster`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

修改叢集。

```
  TRY.
      " Example values: iv_cluster_identifier = 'my-redshift-cluster'
      " Example values: iv_pref_maintenance_wn = 'wed:07:30-wed:08:00'
      lo_rsh->modifycluster(
        iv_clusteridentifier = iv_cluster_identifier
        iv_preferredmaintenancewin00 = iv_pref_maintenance_wn
      ).
      MESSAGE 'Redshift cluster modified successfully.' TYPE 'I'.
    CATCH /aws1/cx_rshclustnotfoundfault.
      MESSAGE 'Cluster not found.' TYPE 'I'.
    CATCH /aws1/cx_rshinvcluststatefault.
      MESSAGE 'Invalid cluster state for modification.' TYPE 'I'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ModifyCluster。

使用適用於 SAP ABAP 的 SDK 的 Amazon Rekognition 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 Amazon Rekognition 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

`CompareFaces`

以下程式碼範例顯示如何使用 `CompareFaces`。

如需詳細資訊，請參閱比較映像中的人臉。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    " Create S3 object reference for the source image
    DATA(lo_source_s3obj) = NEW /aws1/cl_reks3object(
      iv_bucket = iv_source_s3_bucket
      iv_name = iv_source_s3_key ).

    " Create source image object
    DATA(lo_source_image) = NEW /aws1/cl_rekimage(
      io_s3object = lo_source_s3obj ).
```

```
      " Create S3 object reference for the target image
      DATA(lo_target_s3obj) = NEW /aws1/cl_reks3object(
        iv_bucket = iv_target_s3_bucket
        iv_name = iv_target_s3_key ).

      " Create target image object
      DATA(lo_target_image) = NEW /aws1/cl_rekimage(
        io_s3object = lo_target_s3obj ).

      " Compare faces
      oo_result = lo_rek->comparefaces(
        io_sourceimage = lo_source_image
        io_targetimage = lo_target_image
        iv_similaritythreshold = iv_similarity ).

      DATA(lt_face_matches) = oo_result->get_facematches( ).
      DATA(lt_unmatched_faces) = oo_result->get_unmatchedfaces( ).

      " Get counts of matched and unmatched faces
      DATA(lv_matched_count) = lines( lt_face_matches ).
      DATA(lv_unmatched_count) = lines( lt_unmatched_faces ).

      " Output detailed comparison results
      DATA(lv_message) = |Face comparison completed: | &&
                         |{ lv_matched_count } matched face(s), | &&
                         |{ lv_unmatched_count } unmatched face(s).|.
      MESSAGE lv_message TYPE 'I'.
    CATCH /aws1/cx_rekinvalids3objectex.
      MESSAGE 'Invalid S3 object.' TYPE 'E'.
    CATCH /aws1/cx_rekinvalidparameterex.
      MESSAGE 'Invalid parameter value.' TYPE 'E'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CompareFaces。

CreateCollection

以下程式碼範例顯示如何使用 CreateCollection。

如需更多資訊，請參閱建立集合。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_rek->createcollection(
      iv_collectionid = iv_collection_id ).
    MESSAGE 'Collection created successfully.' TYPE 'I'.
  CATCH /aws1/cx_rekresrcalrdyexistsex.
    MESSAGE 'Collection already exists.' TYPE 'E'.
  CATCH /aws1/cx_rekinvalidparameterex.
    MESSAGE 'Invalid parameter value.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateCollection。

`DeleteCollection`

以下程式碼範例顯示如何使用 `DeleteCollection`。

如需更多資訊，請參閱刪除集合。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_rek->deletecollection(
      iv_collectionid = iv_collection_id ).
    MESSAGE 'Collection deleted successfully.' TYPE 'I'.
```

```
  CATCH /aws1/cx_rekresourcenotfoundex.
    MESSAGE 'Collection not found.' TYPE 'E'.
  CATCH /aws1/cx_rekinvalidparameterex.
    MESSAGE 'Invalid parameter value.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteCollection。

`DeleteFaces`

以下程式碼範例顯示如何使用 `DeleteFaces`。

如需詳細資訊，請參閱從集合中刪除人臉。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_rek->deletefaces(
      iv_collectionid = iv_collection_id
      it_faceids = it_face_ids ).

    DATA(lt_deleted_faces) = oo_result->get_deletedfaces( ).
    DATA(lv_deleted_count) = lines( lt_deleted_faces ).
    DATA(lv_msg6) = |{ lv_deleted_count } face(s) deleted successfully.|.
    MESSAGE lv_msg6 TYPE 'I'.
  CATCH /aws1/cx_rekresourcenotfoundex.
    MESSAGE 'Collection not found.' TYPE 'E'.
  CATCH /aws1/cx_rekinvalidparameterex.
    MESSAGE 'Invalid parameter value.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteFaces。

DescribeCollection

以下程式碼範例顯示如何使用 DescribeCollection。

如需詳細資訊，請參閱描述集合。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_rek->describecollection(
      iv_collectionid = iv_collection_id ).
    DATA(lv_face_count) = oo_result->get_facecount( ).
    DATA(lv_msg) = |Collection described: { lv_face_count } face(s) indexed.|.
    MESSAGE lv_msg TYPE 'I'.
  CATCH /aws1/cx_rekresourcenotfoundex.
    MESSAGE 'Collection not found.' TYPE 'E'.
  CATCH /aws1/cx_rekinvalidparameterex.
    MESSAGE 'Invalid parameter value.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeCollection。

DetectFaces

以下程式碼範例顯示如何使用 DetectFaces。

如需詳細資訊，請參閱偵測影像中的人臉。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    " Create S3 object reference for the image
    DATA(lo_s3object) = NEW /aws1/cl_reks3object(
      iv_bucket = iv_s3_bucket
      iv_name = iv_s3_key ).

    " Create image object
    DATA(lo_image) = NEW /aws1/cl_rekimage(
      io_s3object = lo_s3object ).

    " Detect faces in the image with all attributes
    DATA(lt_attributes) = VALUE /aws1/cl_rekattributes_w=>tt_attributes( ).
    DATA(lo_attr_wrapper) = NEW /aws1/cl_rekattributes_w( iv_value = 'ALL' ).
    INSERT lo_attr_wrapper INTO TABLE lt_attributes.

    oo_result = lo_rek->detectfaces(
      io_image = lo_image
      it_attributes = lt_attributes ).

    DATA(lt_face_details) = oo_result->get_facedetails( ).
    DATA(lv_detected_count) = lines( lt_face_details ).
    DATA(lv_msg8) = |{ lv_detected_count } face(s) detected in image.|.
    MESSAGE lv_msg8 TYPE 'I'.
  CATCH /aws1/cx_rekinvalids3objectex.
    MESSAGE 'Invalid S3 object.' TYPE 'E'.
  CATCH /aws1/cx_rekinvalidparameterex.
    MESSAGE 'Invalid parameter value.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DetectFaces。

DetectLabels

以下程式碼範例顯示如何使用 DetectLabels。

如需詳細資訊，請參閱偵測映像中的標籤。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    " Create S3 object reference for the image
    DATA(lo_s3object) = NEW /aws1/cl_reks3object(
      iv_bucket = iv_s3_bucket
      iv_name = iv_s3_key ).

    " Create image object
    DATA(lo_image) = NEW /aws1/cl_rekimage(
      io_s3object = lo_s3object ).

    " Detect labels in the image
    oo_result = lo_rek->detectlabels(
      io_image = lo_image
      iv_maxlabels = iv_max_labels ).

    DATA(lt_labels) = oo_result->get_labels( ).
    DATA(lv_label_count) = lines( lt_labels ).
    DATA(lv_msg9) = |{ lv_label_count } label(s) detected in image.|.
    MESSAGE lv_msg9 TYPE 'I'.
  CATCH /aws1/cx_rekinvalids3objectex.
    MESSAGE 'Invalid S3 object.' TYPE 'E'.
  CATCH /aws1/cx_rekinvalidparameterex.
    MESSAGE 'Invalid parameter value.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DetectLabels。

DetectModerationLabels

以下程式碼範例顯示如何使用 DetectModerationLabels。

如需詳細資訊，請參閱偵測不適合的映像。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    " Create S3 object reference for the image
    DATA(lo_s3object) = NEW /aws1/cl_reks3object(
      iv_bucket = iv_s3_bucket
      iv_name = iv_s3_key ).

    " Create image object
    DATA(lo_image) = NEW /aws1/cl_rekimage(
      io_s3object = lo_s3object ).

    " Detect moderation labels
    oo_result = lo_rek->detectmoderationlabels(
      io_image = lo_image ).

    DATA(lt_moderation_labels) = oo_result->get_moderationlabels( ).
    DATA(lv_mod_count) = lines( lt_moderation_labels ).
    DATA(lv_msg10) = |{ lv_mod_count } moderation label(s) detected.|.
    MESSAGE lv_msg10 TYPE 'I'.
  CATCH /aws1/cx_rekinvalids3objectex.
    MESSAGE 'Invalid S3 object.' TYPE 'E'.
  CATCH /aws1/cx_rekinvalidparameterex.
    MESSAGE 'Invalid parameter value.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DetectModerationLabels。

`DetectText`

以下程式碼範例顯示如何使用 `DetectText`。

如需更多資訊，請參閱偵測映像中的文字。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    " Create S3 object reference for the image
    DATA(lo_s3object) = NEW /aws1/cl_reks3object(
      iv_bucket = iv_s3_bucket
      iv_name = iv_s3_key ).

    " Create image object
    DATA(lo_image) = NEW /aws1/cl_rekimage(
      io_s3object = lo_s3object ).

    " Detect text in the image
    oo_result = lo_rek->detecttext(
      io_image = lo_image ).

    DATA(lt_text_detections) = oo_result->get_textdetections( ).
    DATA(lv_text_count) = lines( lt_text_detections ).
    DATA(lv_msg11) = |{ lv_text_count } text detection(s) found.|.
    MESSAGE lv_msg11 TYPE 'I'.
  CATCH /aws1/cx_rekinvalids3objectex.
    MESSAGE 'Invalid S3 object.' TYPE 'E'.
  CATCH /aws1/cx_rekinvalidparameterex.
    MESSAGE 'Invalid parameter value.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DetectText。

IndexFaces

以下程式碼範例顯示如何使用 IndexFaces。

如需詳細資訊，請參閱將人臉新增至集合。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    " Create S3 object reference for the image
    DATA(lo_s3object) = NEW /aws1/cl_reks3object(
      iv_bucket = iv_s3_bucket
      iv_name = iv_s3_key ).

    " Create image object
    DATA(lo_image) = NEW /aws1/cl_rekimage(
      io_s3object = lo_s3object ).

    " Index faces in the image
    oo_result = lo_rek->indexfaces(
      iv_collectionid = iv_collection_id
      io_image = lo_image
      iv_externalimageid = iv_external_id
      iv_maxfaces = iv_max_faces ).

    DATA(lt_face_records) = oo_result->get_facerecords( ).
    DATA(lv_indexed_count) = lines( lt_face_records ).
    DATA(lv_msg2) = |{ lv_indexed_count } face(s) indexed successfully.|.
    MESSAGE lv_msg2 TYPE 'I'.
  CATCH /aws1/cx_rekresourcenotfoundex.
    MESSAGE 'Collection not found.' TYPE 'E'.
  CATCH /aws1/cx_rekinvalids3objectex.
    MESSAGE 'Invalid S3 object.' TYPE 'E'.
  CATCH /aws1/cx_rekinvalidparameterex.
    MESSAGE 'Invalid parameter value.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 IndexFaces。

`ListCollections`

以下程式碼範例顯示如何使用 `ListCollections`。

如需詳細資訊，請參閱列出的集合。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_rek->listcollections(
      iv_maxresults = iv_max_results ).

    DATA(lt_collection_ids) = oo_result->get_collectionids( ).
    DATA(lv_coll_count) = lines( lt_collection_ids ).
    DATA(lv_msg7) = |{ lv_coll_count } collection(s) found.|.
    MESSAGE lv_msg7 TYPE 'I'.
  CATCH /aws1/cx_rekinvalidparameterex.
    MESSAGE 'Invalid parameter value.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListCollections。

`ListFaces`

以下程式碼範例顯示如何使用 `ListFaces`。

如需更多資訊，請參閱集合中列出的人臉。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_rek->listfaces(
      iv_collectionid = iv_collection_id
      iv_maxresults = iv_max_results ).

    DATA(lt_faces) = oo_result->get_faces( ).
    DATA(lv_face_count2) = lines( lt_faces ).
    DATA(lv_msg3) = |{ lv_face_count2 } face(s) found in collection.|.
    MESSAGE lv_msg3 TYPE 'I'.
  CATCH /aws1/cx_rekresourcenotfoundex.
    MESSAGE 'Collection not found.' TYPE 'E'.
  CATCH /aws1/cx_rekinvalidparameterex.
    MESSAGE 'Invalid parameter value.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListFaces。

`RecognizeCelebrities`

以下程式碼範例顯示如何使用 `RecognizeCelebrities`。

如需詳細資訊，請參閱在映像中辨識名人。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " Create S3 object reference for the image
        DATA(lo_s3object) = NEW /aws1/cl_reks3object(
          iv_bucket = iv_s3_bucket
          iv_name = iv_s3_key ).

        " Create image object
        DATA(lo_image) = NEW /aws1/cl_rekimage(
          io_s3object = lo_s3object ).

        " Recognize celebrities
        oo_result = lo_rek->recognizecelebrities(
          io_image = lo_image ).

        DATA(lt_celebrity_faces) = oo_result->get_celebrityfaces( ).
        DATA(lv_celeb_count) = lines( lt_celebrity_faces ).
        DATA(lv_msg12) = |{ lv_celeb_count } celebrity/celebrities recognized.|.
        MESSAGE lv_msg12 TYPE 'I'.
      CATCH /aws1/cx_rekinvalids3objectex.
        MESSAGE 'Invalid S3 object.' TYPE 'E'.
      CATCH /aws1/cx_rekinvalidparameterex.
        MESSAGE 'Invalid parameter value.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 RecognizeCelebrities。

SearchFaces

以下程式碼範例顯示如何使用 SearchFaces。

如需詳細資訊，請參閱搜尋人臉 (臉部 ID)。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        oo_result = lo_rek->searchfaces(
          iv_collectionid = iv_collection_id
          iv_faceid = iv_face_id
          iv_facematchthreshold = iv_threshold
          iv_maxfaces = iv_max_faces ).

        DATA(lt_face_matches) = oo_result->get_facematches( ).
        DATA(lv_match_count2) = lines( lt_face_matches ).
        DATA(lv_msg5) = |Face search completed: { lv_match_count2 } match(es)
found.|.
        MESSAGE lv_msg5 TYPE 'I'.
      CATCH /aws1/cx_rekresourcenotfoundex.
        MESSAGE 'Collection or face not found.' TYPE 'E'.
      CATCH /aws1/cx_rekinvalidparameterex.
        MESSAGE 'Invalid parameter value.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 SearchFaces。

SearchFacesByImage

以下程式碼範例顯示如何使用 SearchFacesByImage。

如需詳細資訊，請參閱搜尋人臉 (映像)。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " Create S3 object reference for the image
        DATA(lo_s3object) = NEW /aws1/cl_reks3object(
          iv_bucket = iv_s3_bucket
          iv_name = iv_s3_key ).
```

```
        " Create image object
        DATA(lo_image) = NEW /aws1/cl_rekimage(
          io_s3object = lo_s3object ).

        " Search for matching faces
        oo_result = lo_rek->searchfacesbyimage(
          iv_collectionid = iv_collection_id
          io_image = lo_image
          iv_facematchthreshold = iv_threshold
          iv_maxfaces = iv_max_faces ).

        DATA(lt_face_matches) = oo_result->get_facematches( ).
        DATA(lv_match_count) = lines( lt_face_matches ).
        DATA(lv_msg4) = |Face search completed: { lv_match_count } match(es)
found.|.
        MESSAGE lv_msg4 TYPE 'I'.
      CATCH /aws1/cx_rekresourcenotfoundex.
        MESSAGE 'Collection not found.' TYPE 'E'.
      CATCH /aws1/cx_rekinvalids3objectex.
        MESSAGE 'Invalid S3 object.' TYPE 'E'.
      CATCH /aws1/cx_rekinvalidparameterex.
        MESSAGE 'Invalid parameter value.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 SearchFacesByImage。

使用適用於 SAP ABAP 的 SDK 的 Amazon S3 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 Amazon S3 來執行動作和實作常見案例。

基本概念是程式碼範例，這些範例說明如何在服務內執行基本操作。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

案例是向您展示如何呼叫服務中的多個函數或與其他 AWS 服務組合來完成特定任務的程式碼範例。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 基本概念
- 動作
- 案例

基本概念

了解基本概念

以下程式碼範例顯示做法：

- 建立儲存貯體並上傳檔案到該儲存貯體。
- 從儲存貯體下載物件。
- 將物件複製至儲存貯體中的子文件夾。
- 列出儲存貯體中的物件。
- 刪除儲存貯體物件和該儲存貯體。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

    " Create an Amazon Simple Storage Service (Amazon S3) bucket. "
    TRY.
        " determine our region from our session
        DATA(lv_region) = CONV /aws1/s3_bucketlocationcnstrnt( lo_session-
>get_region( ) ).
        DATA lo_constraint TYPE REF TO /aws1/cl_s3_createbucketconf.
        " When in the us-east-1 region, you must not specify a constraint
        " In all other regions, specify the region as the constraint
        IF lv_region = 'us-east-1'.
          CLEAR lo_constraint.
```

```
    ELSE.
      lo_constraint = NEW /aws1/cl_s3_createbucketconf( lv_region ).
    ENDIF.

    lo_s3->createbucket(
        iv_bucket = iv_bucket_name
        io_createbucketconfiguration  = lo_constraint ).
    MESSAGE 'S3 bucket created.' TYPE 'I'.
  CATCH /aws1/cx_s3_bucketalrdyexists.
    MESSAGE 'Bucket name already exists.' TYPE 'E'.
  CATCH /aws1/cx_s3_bktalrdyownedbyyou.
    MESSAGE 'Bucket already exists and is owned by you.' TYPE 'E'.
ENDTRY.


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

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

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

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

" List objects in the bucket. "
TRY.
    DATA(lo_list) = lo_s3->listobjects(
       iv_bucket = iv_bucket_name ).
    MESSAGE 'Retrieved list of objects in S3 bucket.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.
DATA text TYPE string VALUE 'Object List - '.
DATA lv_object_key TYPE /aws1/s3_objectkey.
LOOP AT lo_list->get_contents( ) INTO DATA(lo_object).
  lv_object_key = lo_object->get_key( ).
  CONCATENATE lv_object_key ', ' INTO text.
ENDLOOP.
MESSAGE text TYPE'I'.

" Delete the objects in a bucket. "
TRY.
    lo_s3->deleteobject(
        iv_bucket = iv_bucket_name
        iv_key = iv_key ).
    lo_s3->deleteobject(
        iv_bucket = iv_bucket_name
        iv_key = |{ iv_copy_to_folder }/{ iv_key }| ).
    MESSAGE 'Objects deleted from S3 bucket.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.
```

```
" Delete the bucket. "
TRY.
    lo_s3->deletebucket(
        iv_bucket = iv_bucket_name ).
    MESSAGE 'Deleted S3 bucket.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的下列主題。
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

動作

`CopyObject`

以下程式碼範例顯示如何使用 `CopyObject`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 CopyObject。

`CreateBucket`

以下程式碼範例顯示如何使用 `CreateBucket`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " determine our region from our session
        DATA(lv_region) = CONV /aws1/s3_bucketlocationcnstrnt( lo_session->get_region( ) ).
        DATA lo_constraint TYPE REF TO /aws1/cl_s3_createbucketconf.
        " When in the us-east-1 region, you must not specify a constraint
        " In all other regions, specify the region as the constraint
        IF lv_region = 'us-east-1'.
          CLEAR lo_constraint.
        ELSE.
          lo_constraint = NEW /aws1/cl_s3_createbucketconf( lv_region ).
        ENDIF.

        lo_s3->createbucket(
            iv_bucket = iv_bucket_name
            io_createbucketconfiguration  = lo_constraint ).
        MESSAGE 'S3 bucket created.' TYPE 'I'.
      CATCH /aws1/cx_s3_bucketalrdyexists.
        MESSAGE 'Bucket name already exists.' TYPE 'E'.
      CATCH /aws1/cx_s3_bktalrdyownedbyyou.
```

```
        MESSAGE 'Bucket already exists and is owned by you.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 CreateBucket。

`DeleteBucket`

以下程式碼範例顯示如何使用 `DeleteBucket`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 DeleteBucket。

`DeleteBucketCors`

以下程式碼範例顯示如何使用 `DeleteBucketCors`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_s3->deletebucketcors(
      iv_bucket = iv_bucket_name ).
    MESSAGE 'Bucket CORS configuration deleted.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteBucketCors。

`DeleteBucketLifecycle`

以下程式碼範例顯示如何使用 `DeleteBucketLifecycle`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_s3->deletebucketlifecycle(
      iv_bucket = iv_bucket_name ).
    MESSAGE 'Bucket lifecycle configuration deleted.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteBucketLifecycle。

`DeleteBucketPolicy`

以下程式碼範例顯示如何使用 `DeleteBucketPolicy`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_s3->deletebucketpolicy(
      iv_bucket = iv_bucket_name ).
    MESSAGE 'Bucket policy deleted.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteBucketPolicy。

`DeleteObject`

以下程式碼範例顯示如何使用 `DeleteObject`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 DeleteObject。

`DeleteObjects`

以下程式碼範例顯示如何使用 `DeleteObjects`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        oo_result = lo_s3->deleteobjects(         " oo_result is returned for
testing purposes. "
          iv_bucket = iv_bucket_name
          io_delete = NEW /aws1/cl_s3_delete( it_objects = it_object_keys ) ).
        MESSAGE 'Objects deleted from S3 bucket.' TYPE 'I'.
      CATCH /aws1/cx_s3_nosuchbucket.
        MESSAGE 'Bucket does not exist.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteObjects。

`GetBucketAcl`

以下程式碼範例顯示如何使用 `GetBucketAcl`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        oo_result = lo_s3->getbucketacl(         " oo_result is returned for testing
purposes. "
          iv_bucket = iv_bucket_name ).
        MESSAGE 'Retrieved bucket ACL.' TYPE 'I'.
      CATCH /aws1/cx_s3_nosuchbucket.
        MESSAGE 'Bucket does not exist.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetBucketAcl。

GetBucketCors

以下程式碼範例顯示如何使用 GetBucketCors。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        oo_result = lo_s3->getbucketcors(         " oo_result is returned for
testing purposes. "
          iv_bucket = iv_bucket_name ).
        MESSAGE 'Retrieved bucket CORS configuration.' TYPE 'I'.
      CATCH /aws1/cx_s3_nosuchbucket.
        MESSAGE 'Bucket does not exist.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetBucketCors。

GetBucketLifecycleConfiguration

以下程式碼範例顯示如何使用 GetBucketLifecycleConfiguration。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   TRY.
       oo_result = lo_s3->getbucketlifecycleconf(         " oo_result is returned
for testing purposes. "
         iv_bucket = iv_bucket_name ).
       MESSAGE 'Retrieved bucket lifecycle configuration.' TYPE 'I'.
     CATCH /aws1/cx_s3_nosuchbucket.
       MESSAGE 'Bucket does not exist.' TYPE 'E'.
   ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetBucketLifecycleConfiguration。

`GetBucketPolicy`

以下程式碼範例顯示如何使用 `GetBucketPolicy`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   TRY.
       oo_result = lo_s3->getbucketpolicy(         " oo_result is returned for
testing purposes. "
         iv_bucket = iv_bucket_name ).
       DATA(lv_policy) = oo_result->get_policy( ).
       MESSAGE 'Retrieved bucket policy.' TYPE 'I'.
```

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

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetBucketPolicy。

`GetObject`

以下程式碼範例顯示如何使用 `GetObject`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 GetObject。

`GetObjectAcl`

以下程式碼範例顯示如何使用 `GetObjectAcl`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   TRY.
       oo_result = lo_s3->getobjectacl(         " oo_result is returned for testing
purposes. "
         iv_bucket = iv_bucket_name
         iv_key = iv_object_key ).
       MESSAGE 'Retrieved object ACL.' TYPE 'I'.
     CATCH /aws1/cx_s3_nosuchbucket.
       MESSAGE 'Bucket does not exist.' TYPE 'E'.
     CATCH /aws1/cx_s3_nosuchkey.
       MESSAGE 'Object key does not exist.' TYPE 'E'.
   ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetObjectAcl。

`GetObjectLegalHold`

以下程式碼範例顯示如何使用 `GetObjectLegalHold`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   TRY.
       oo_result = lo_s3->getobjectlegalhold(         " oo_result is returned for
testing purposes. "
         iv_bucket = iv_bucket_name
```

```
          iv_key = iv_object_key ).
        MESSAGE 'Retrieved object legal hold status.' TYPE 'I'.
      CATCH /aws1/cx_s3_nosuchbucket.
        MESSAGE 'Bucket does not exist.' TYPE 'E'.
      CATCH /aws1/cx_s3_nosuchkey.
        MESSAGE 'Object key does not exist.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetObjectLegalHold。

`GetObjectLockConfiguration`

以下程式碼範例顯示如何使用 `GetObjectLockConfiguration`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        oo_result = lo_s3->getobjectlockconfiguration(         " oo_result is
returned for testing purposes. "
          iv_bucket = iv_bucket_name ).
        MESSAGE 'Retrieved object lock configuration.' TYPE 'I'.
      CATCH /aws1/cx_s3_nosuchbucket.
        MESSAGE 'Bucket does not exist.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetObjectLockConfiguration。

`HeadBucket`

以下程式碼範例顯示如何使用 `HeadBucket`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   TRY.
       oo_result = lo_s3->headbucket(         " oo_result is returned for testing
purposes. "
         iv_bucket = iv_bucket_name ).
       MESSAGE 'Bucket exists and you have access to it.' TYPE 'I'.
     CATCH /aws1/cx_s3_nosuchbucket.
       MESSAGE 'Bucket does not exist.' TYPE 'E'.
   ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 HeadBucket。

`ListObjectVersions`

以下程式碼範例顯示如何使用 `ListObjectVersions`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   TRY.
       oo_result = lo_s3->listobjectversions(         " oo_result is returned for
testing purposes. "
         iv_bucket = iv_bucket_name
         iv_prefix = iv_prefix ).
       MESSAGE 'Retrieved object versions.' TYPE 'I'.
     CATCH /aws1/cx_s3_nosuchbucket.
```

```
        MESSAGE 'Bucket does not exist.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListObjectVersions。

`ListObjectsV2`

以下程式碼範例顯示如何使用 `ListObjectsV2`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 ListObjectsV2。

`PutBucketAcl`

以下程式碼範例顯示如何使用 `PutBucketAcl`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    " Example: Grant log delivery access to a bucket
    " iv_grantwrite = 'uri=http://acs.amazonaws.com/groups/s3/LogDelivery'
    lo_s3->putbucketacl(
      iv_bucket = iv_bucket_name
      iv_grantwrite = iv_grantwrite ).
    MESSAGE 'Bucket ACL updated.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 PutBucketAcl。

`PutBucketCors`

以下程式碼範例顯示如何使用 `PutBucketCors`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    " Example: Allow PUT, POST, DELETE methods from http://www.example.com
    lo_s3->putbucketcors(
      iv_bucket = iv_bucket_name
      io_corsconfiguration = NEW /aws1/cl_s3_corsconfiguration(
```

```
            it_corsrules = it_cors_rules ) ).
        MESSAGE 'Bucket CORS configuration set.' TYPE 'I'.
      CATCH /aws1/cx_s3_nosuchbucket.
        MESSAGE 'Bucket does not exist.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 PutBucketCors。

`PutBucketLifecycleConfiguration`

以下程式碼範例顯示如何使用 `PutBucketLifecycleConfiguration`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " Example: Expire objects with prefix 'logs/' after 30 days
        lo_s3->putbucketlifecycleconf(
          iv_bucket = iv_bucket_name
          io_lifecycleconfiguration = NEW /aws1/cl_s3_bucketlcconf(
            it_rules = it_lifecycle_rule ) ).
        MESSAGE 'Bucket lifecycle configuration set.' TYPE 'I'.
      CATCH /aws1/cx_s3_nosuchbucket.
        MESSAGE 'Bucket does not exist.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 PutBucketLifecycleConfiguration。

`PutBucketPolicy`

以下程式碼範例顯示如何使用 `PutBucketPolicy`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " Example policy JSON string
        " iv_policy = '{"Version":"2012-10-17",       "Statement":
[{"Effect":"Allow","Principal":{"AWS":"arn:aws:iam::123456789012:user/
user"},"Action":["s3:GetObject"],"Resource":["arn:aws:s3:::bucketname/*"]}]}'
        lo_s3->putbucketpolicy(
          iv_bucket = iv_bucket_name
          iv_policy = iv_policy ).
        MESSAGE 'Bucket policy set.' TYPE 'I'.
      CATCH /aws1/cx_s3_nosuchbucket.
        MESSAGE 'Bucket does not exist.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 PutBucketPolicy。

`PutBucketVersioning`

以下程式碼範例顯示如何使用 `PutBucketVersioning`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " Example: Enable versioning on a bucket
        " iv_status = 'Enabled'
```

```
    lo_s3->putbucketversioning(
      iv_bucket = iv_bucket_name
      io_versioningconfiguration = NEW /aws1/cl_s3_versioningconf(
        iv_status = iv_status ) ).
    MESSAGE 'Bucket versioning enabled.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 PutBucketVersioning。

`PutObject`

以下程式碼範例顯示如何使用 `PutObject`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
"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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 PutObject。

`PutObjectAcl`

以下程式碼範例顯示如何使用 `PutObjectAcl`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    " Example: Grant read access to an AWS user
    " iv_grantread = 'emailAddress=user@example.com'
    lo_s3->putobjectacl(
      iv_bucket = iv_bucket_name
      iv_key = iv_object_key
      iv_grantread = iv_grantread ).
    MESSAGE 'Object ACL updated.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
  CATCH /aws1/cx_s3_nosuchkey.
    MESSAGE 'Object key does not exist.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 PutObjectAcl。

`PutObjectLegalHold`

以下程式碼範例顯示如何使用 `PutObjectLegalHold`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    " Example: Set legal hold status to ON
    " iv_status = 'ON'
    lo_s3->putobjectlegalhold(
      iv_bucket = iv_bucket_name
      iv_key = iv_object_key
      io_legalhold = NEW /aws1/cl_s3_objlocklegalhold(
        iv_status = iv_status ) ).
    MESSAGE 'Object legal hold status set.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
  CATCH /aws1/cx_s3_nosuchkey.
    MESSAGE 'Object key does not exist.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 PutObjectLegalHold。

`PutObjectLockConfiguration`

以下程式碼範例顯示如何使用 `PutObjectLockConfiguration`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
```

```
    " Example: Enable object lock with default retention
    " iv_enabled = 'Enabled'
    lo_s3->putobjectlockconfiguration(
      iv_bucket = iv_bucket_name
      io_objectlockconfiguration = NEW /aws1/cl_s3_objectlockconf(
        iv_objectlockenabled = iv_enabled ) ).
    MESSAGE 'Object lock configuration set.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 PutObjectLockConfiguration。

`PutObjectRetention`

以下程式碼範例顯示如何使用 `PutObjectRetention`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    " Example: Set retention mode to GOVERNANCE for 30 days
    " iv_mode = 'GOVERNANCE'
    " iv_retain_date should be a timestamp in the future
    lo_s3->putobjectretention(
      iv_bucket = iv_bucket_name
      iv_key = iv_object_key
      io_retention = NEW /aws1/cl_s3_objectlockret(
        iv_mode = iv_mode
        iv_retainuntildate = iv_retain_date )
      iv_bypassgovernanceretention = abap_true ).
    MESSAGE 'Object retention set.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
```

```
  CATCH /aws1/cx_s3_nosuchkey.
    MESSAGE 'Object key does not exist.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 PutObjectRetention。

案例

建立預先簽章 URL

下列程式碼範例示範如何建立適用於 Amazon S3 預先簽署的 URL，並上傳物件。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

建立對 GET S3 物件的預先簽署請求。

```
" iv_bucket_name is the bucket name
" iv_key is the object name like "myfile.txt"

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

"Upload a nice Hello World file to an S3 bucket."
TRY.
    DATA(lv_contents) = cl_abap_codepage=>convert_to( 'Hello, World' ).
    lo_s3->putobject(
        iv_bucket = iv_bucket_name
        iv_key = iv_key
        iv_body = lv_contents
        iv_contenttype = 'text/plain' ).
    MESSAGE 'Object uploaded to S3 bucket.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
```

```
    ENDTRY.

    " now generate a presigned URL with a 600-second expiration
    DATA(lo_presigner) = lo_s3->get_presigner( iv_expires_sec = 600 ).
    " the presigner getobject() method has the same signature as
    " lo_s3->getobject(), but it doesn't actually make the call.
    " to the service.  It just prepares a presigned URL for a future call
    DATA(lo_presigned_req) = lo_presigner->getobject(
      iv_bucket = iv_bucket_name
      iv_key = iv_key ).

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

使用適用於 SAP ABAP 的 SDK 的 SageMaker AI 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 SageMaker AI 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

案例是向您展示如何呼叫服務中的多個函數或與其他 AWS 服務組合來完成特定任務的程式碼範例。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作
- 案例

動作

`CreateEndpoint`

以下程式碼範例顯示如何使用 `CreateEndpoint`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

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

    INSERT lo_production_variants INTO TABLE lt_production_variants.

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

    "Create an endpoint."
    TRY.
        oo_result = lo_sgm->createendpoint(     " oo_result is returned for testing
 purposes. "
            iv_endpointconfigname = iv_endpoint_config_name
            iv_endpointname = iv_endpoint_name ).
        MESSAGE 'Endpoint created.' TYPE 'I'.
```

```
      CATCH /aws1/cx_sgmresourcelimitexcd.
        MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 CreateEndpoint。

`CreateModel`

以下程式碼範例顯示如何使用 `CreateModel`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

    "Create an ABAP object for the container image based on input variables."
    lo_primarycontainer = NEW #( iv_image = iv_container_image
                                 iv_modeldataurl = iv_model_data_url ).

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

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 CreateModel。

`CreateTrainingJob`

以下程式碼範例顯示如何使用 `CreateTrainingJob`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   DATA lo_hyperparameters_w TYPE REF TO /aws1/cl_sgmhyperparameters_w.
   DATA lt_hyperparameters TYPE /aws1/cl_sgmhyperparameters_w=>tt_hyperparameters.
   DATA lt_input_data_config TYPE /aws1/cl_sgmchannel=>tt_inputdataconfig.
   DATA lo_trn_channel TYPE REF TO /aws1/cl_sgmchannel.
   DATA lo_trn_datasource TYPE REF TO /aws1/cl_sgmdatasource.
   DATA lo_trn_s3datasource TYPE REF TO /aws1/cl_sgms3datasource.
   DATA lo_val_channel TYPE REF TO /aws1/cl_sgmchannel.
   DATA lo_val_datasource TYPE REF TO /aws1/cl_sgmdatasource.
   DATA lo_val_s3datasource TYPE REF TO /aws1/cl_sgms3datasource.
   DATA lo_algorithm_specification TYPE REF TO /aws1/cl_sgmalgorithmspec.
   DATA lo_resource_config  TYPE REF TO /aws1/cl_sgmresourceconfig.
   DATA lo_output_data_config TYPE REF TO /aws1/cl_sgmoutputdataconfig.
   DATA lo_stopping_condition TYPE REF TO /aws1/cl_sgmstoppingcondition.

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

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

   lo_hyperparameters_w = NEW #( iv_value = iv_hp_eval_metric ).
```

```
   INSERT VALUE #( key = 'eval_metric' value = lo_hyperparameters_w )  INTO TABLE
lt_hyperparameters.

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

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

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

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

   "Create ABAP objects for training data sources."
   lo_trn_s3datasource = NEW #( iv_s3datatype = iv_trn_data_s3datatype
                                iv_s3datadistributiontype =
iv_trn_data_s3datadistribution
                                iv_s3uri = iv_trn_data_s3uri ).

   lo_trn_datasource = NEW #( io_s3datasource = lo_trn_s3datasource ).

   lo_trn_channel = NEW #( iv_channelname = 'train'
                           io_datasource = lo_trn_datasource
                           iv_compressiontype = iv_trn_data_compressiontype
                           iv_contenttype = iv_trn_data_contenttype ).

   INSERT lo_trn_channel INTO TABLE lt_input_data_config.

   "Create ABAP objects for validation data sources."
   lo_val_s3datasource = NEW #( iv_s3datatype = iv_val_data_s3datatype
                                iv_s3datadistributiontype =
iv_val_data_s3datadistribution
                                iv_s3uri = iv_val_data_s3uri ).

   lo_val_datasource = NEW #( io_s3datasource = lo_val_s3datasource ).

   lo_val_channel = NEW #( iv_channelname = 'validation'
                           io_datasource = lo_val_datasource
```

```
                            iv_compressiontype = iv_val_data_compressiontype
                            iv_contenttype = iv_val_data_contenttype ).

   INSERT lo_val_channel INTO TABLE lt_input_data_config.

   "Create an ABAP object for algorithm specification."
   lo_algorithm_specification = NEW #( iv_trainingimage = iv_training_image
                                       iv_traininginputmode =
iv_training_input_mode ).

   "Create an ABAP object for resource configuration."
   lo_resource_config = NEW #( iv_instancecount = iv_instance_count
                               iv_instancetype = iv_instance_type
                               iv_volumesizeingb = iv_volume_sizeingb ).

   "Create an ABAP object for output data configuration."
   lo_output_data_config = NEW #( iv_s3outputpath = iv_s3_output_path ).

   "Create an ABAP object for stopping condition."
   lo_stopping_condition = NEW #( iv_maxruntimeinseconds =
iv_max_runtime_in_seconds ).

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

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 CreateTrainingJob。

`CreateTransformJob`

以下程式碼範例顯示如何使用 `CreateTransformJob`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

   "Create an ABAP object for an Amazon Simple Storage Service (Amazon S3) data
source."
   lo_s3datasource = NEW #( iv_s3uri = iv_tf_data_s3uri
                            iv_s3datatype = iv_tf_data_s3datatype ).

   "Create an ABAP object for data source."
   lo_datasource = NEW #( io_s3datasource = lo_s3datasource ).

   "Create an ABAP object for transform data source."
   lo_transforminput = NEW #( io_datasource = lo_datasource
                              iv_contenttype = iv_tf_data_contenttype
                              iv_compressiontype = iv_tf_data_compressiontype ).

   "Create an ABAP object for resource configuration."
   lo_transformresources = NEW #( iv_instancecount = iv_instance_count
                                  iv_instancetype = iv_instance_type ).

   "Create an ABAP object for output data configuration."
   lo_transformoutput = NEW #( iv_s3outputpath = iv_s3_output_path ).
```

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

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 CreateTransformJob。

`DeleteEndpoint`

以下程式碼範例顯示如何使用 `DeleteEndpoint`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    "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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的 AWS SDK API 參考》中的 DeleteEndpoint。

`DeleteModel`

以下程式碼範例顯示如何使用 `DeleteModel`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 DeleteModel。

`DescribeTrainingJob`

以下程式碼範例顯示如何使用 `DescribeTrainingJob`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的 AWS SDK API 參考》中的 DescribeTrainingJob。

`ListAlgorithms`

以下程式碼範例顯示如何使用 `ListAlgorithms`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        oo_result = lo_sgm->listalgorithms(         " oo_result is returned for
 testing purposes. "
          iv_namecontains = iv_name_contains ).
        MESSAGE 'Retrieved list of algorithms.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 ListAlgorithms。

`ListModels`

以下程式碼範例顯示如何使用 `ListModels`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 ListModels。

`ListNotebookInstances`

以下程式碼範例顯示如何使用 `ListNotebookInstances`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 ListNotebookInstances。

`ListTrainingJobs`

以下程式碼範例顯示如何使用 `ListTrainingJobs`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 ListTrainingJobs。

案例

開始使用模型和端點

以下程式碼範例顯示做法：

- 開始訓練任務，並建立 SageMaker AI 模型。
- 建立端點組態。
- 建立端點，然後清除資源。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    DATA lo_hyperparameters_w TYPE REF TO /aws1/cl_sgmhyperparameters_w.
    DATA lo_trn_channel TYPE REF TO /aws1/cl_sgmchannel.
    DATA lo_trn_datasource TYPE REF TO /aws1/cl_sgmdatasource.
    DATA lo_trn_s3datasource TYPE REF TO /aws1/cl_sgms3datasource.
    DATA lo_val_channel TYPE REF TO /aws1/cl_sgmchannel.
    DATA lo_val_datasource TYPE REF TO /aws1/cl_sgmdatasource.
    DATA lo_val_s3datasource TYPE REF TO /aws1/cl_sgms3datasource.
    DATA lo_algorithm_specification TYPE REF TO /aws1/cl_sgmalgorithmspec.
    DATA lo_resource_config  TYPE REF TO /aws1/cl_sgmresourceconfig.
    DATA lo_output_data_config TYPE REF TO /aws1/cl_sgmoutputdataconfig.
    DATA lo_stopping_condition TYPE REF TO /aws1/cl_sgmstoppingcondition.
    DATA lo_primarycontainer TYPE REF TO /aws1/cl_sgmcontainerdefn.
    DATA lo_production_variants TYPE REF TO /aws1/cl_sgmproductionvariant.
    DATA lo_ep_config_result TYPE REF TO /aws1/cl_sgmcreateendptcfgout.
    DATA lo_training_result TYPE REF TO /aws1/cl_sgmdescrtrnjobrsp.
    DATA lt_production_variants TYPE /aws1/
cl_sgmproductionvariant=>tt_productionvariantlist.
    DATA lt_input_data_config TYPE /aws1/cl_sgmchannel=>tt_inputdataconfig.
    DATA lt_hyperparameters TYPE /aws1/cl_sgmhyperparameters_w=>tt_hyperparameters.
    DATA lv_model_data_url TYPE /aws1/sgmurl.

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

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

    lo_hyperparameters_w = NEW #( iv_value = iv_hp_eta ).
```

```
   INSERT VALUE #( key = 'eta' value = lo_hyperparameters_w )  INTO TABLE
lt_hyperparameters.

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

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

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

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

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

   "Create ABAP internal table for data based on input variables."
   "Training data."
   lo_trn_s3datasource = NEW #( iv_s3datatype = iv_trn_data_s3datatype
                                iv_s3datadistributiontype =
iv_trn_data_s3datadistribution
                                iv_s3uri = iv_trn_data_s3uri ).

   lo_trn_datasource = NEW #( io_s3datasource = lo_trn_s3datasource ).

   lo_trn_channel = NEW #( iv_channelname = 'train'
                           io_datasource = lo_trn_datasource
                           iv_compressiontype = iv_trn_data_compressiontype
                           iv_contenttype = iv_trn_data_contenttype ).
   INSERT lo_trn_channel INTO TABLE lt_input_data_config.

   "Validation data."
   lo_val_s3datasource = NEW #( iv_s3datatype = iv_val_data_s3datatype
                                iv_s3datadistributiontype =
iv_val_data_s3datadistribution
                                iv_s3uri = iv_val_data_s3uri ).
```

```
    lo_val_datasource = NEW #( io_s3datasource = lo_val_s3datasource ).

    lo_val_channel = NEW #( iv_channelname = 'validation'
                            io_datasource = lo_val_datasource
                            iv_compressiontype = iv_val_data_compressiontype
                            iv_contenttype = iv_val_data_contenttype ).
    INSERT lo_val_channel INTO TABLE lt_input_data_config.

    "Create an ABAP object for algorithm specification based on input variables."
    lo_algorithm_specification = NEW #( iv_trainingimage = iv_training_image
                                        iv_traininginputmode =
iv_training_input_mode ).

    "Create an ABAP object for resource configuration."
    lo_resource_config = NEW #( iv_instancecount = iv_instance_count
                                iv_instancetype = iv_instance_type
                                iv_volumesizeingb = iv_volume_sizeingb ).

    "Create an ABAP object for output data configuration."
    lo_output_data_config = NEW #( iv_s3outputpath = iv_s3_output_path ).

    "Create an ABAP object for stopping condition."
    lo_stopping_condition = NEW #( iv_maxruntimeinseconds =
iv_max_runtime_in_seconds ).

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

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

   "Create ABAP object for the container image based on input variables."
   lo_primarycontainer = NEW #( iv_image = iv_training_image
                                iv_modeldataurl = lv_model_data_url ).

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

   "Create an endpoint production variant."
   lo_production_variants = NEW #( iv_variantname = iv_ep_variant_name
                                   iv_modelname = iv_model_name
                                   iv_initialinstancecount =
iv_ep_initial_instance_count
                                   iv_instancetype = iv_ep_instance_type ).
   INSERT lo_production_variants INTO TABLE lt_production_variants.

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

       "Create an endpoint."
```

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

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

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

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

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

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的下列主題。

- CreateEndpoint
- CreateEndpointConfig
- CreateModel
- CreateTrainingJob
- DeleteEndpoint
- DeleteEndpointConfig
- DeleteModel
- DescribeEndpoint
- DescribeTrainingJob

使用適用於 SAP ABAP 的 SDK 的 Secrets Manager 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 Secrets Manager 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

`BatchGetSecretValue`

以下程式碼範例顯示如何使用 `BatchGetSecretValue`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " iv_filter_name = 'mySecret'
        DATA(lo_result) = lo_smr->batchgetsecretvalue(
          it_filters = VALUE /aws1/cl_smrfilter=>tt_filterslisttype(
            (
              NEW /aws1/cl_smrfilter(
                iv_key = 'name'
                it_values = VALUE /aws1/
cl_smrfiltvalsstrlist_w=>tt_filtervaluesstringlist(
                  ( NEW /aws1/cl_smrfiltvalsstrlist_w( iv_value = iv_filter_name ) )
                )
              )
            )
          )
        ).
        ot_secret_values = lo_result->get_secretvalues( ).
        MESSAGE 'Secrets retrieved successfully.' TYPE 'I'.
      CATCH /aws1/cx_smrresourcenotfoundex.
        MESSAGE 'One or more requested secrets were not found.' TYPE 'E'.
      CATCH /aws1/cx_smrdecryptionfailure.
        MESSAGE 'Failed to decrypt one or more secrets.' TYPE 'E'.
      CATCH /aws1/cx_smrinvalidparameterex.
        MESSAGE 'Invalid parameter provided.' TYPE 'E'.
      CATCH /aws1/cx_smrinvalidrequestex.
        MESSAGE 'Invalid request.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 BatchGetSecretValue。

`GetSecretValue`

以下程式碼範例顯示如何使用 `GetSecretValue`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    " iv_secret_name = 'MySecretName'
    DATA(lo_result) = lo_smr->getsecretvalue( iv_secretid = iv_secret_name ).
    ov_secret_value = lo_result->get_secretstring( ).
    MESSAGE 'Secret value retrieved successfully.' TYPE 'I'.
  CATCH /aws1/cx_smrresourcenotfoundex.
    MESSAGE 'The requested secret was not found.' TYPE 'E'.
  CATCH /aws1/cx_smrdecryptionfailure.
    MESSAGE 'Failed to decrypt the secret.' TYPE 'E'.
  CATCH /aws1/cx_smrinvalidparameterex.
    MESSAGE 'Invalid parameter provided.' TYPE 'E'.
  CATCH /aws1/cx_smrinvalidrequestex.
    MESSAGE 'Invalid request.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetSecretValue。

使用適用於 SAP ABAP 的 SDK 的 Amazon SES 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 Amazon SES 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

`CreateReceiptFilter`

以下程式碼範例顯示如何使用 `CreateReceiptFilter`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
" iv_allow = abap_true means 'Allow', abap_false means 'Block'
DATA(lv_policy) = COND /aws1/sesreceiptfilterpolicy(
  WHEN iv_allow = abap_true THEN 'Allow'
  ELSE 'Block'
).

DATA(lo_ip_filter) = NEW /aws1/cl_sesreceiptipfilter(
  iv_policy = lv_policy
  iv_cidr = iv_ip_address_or_range
).

DATA(lo_filter) = NEW /aws1/cl_sesreceiptfilter(
  iv_name = iv_filter_name
  io_ipfilter = lo_ip_filter
).

TRY.
    lo_ses->createreceiptfilter( io_filter = lo_filter ).
    MESSAGE 'Receipt filter created successfully' TYPE 'I'.
  CATCH /aws1/cx_sesalreadyexistsex INTO DATA(lo_ex1).
    DATA(lv_error) = |Filter already exists: { lo_ex1->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex1.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
    lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex_generic.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateReceiptFilter。

CreateReceiptRule

以下程式碼範例顯示如何使用 CreateReceiptRule。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
" Create S3 action for copying emails to S3
DATA(lo_s3_action) = NEW /aws1/cl_sess3action(
  iv_bucketname = iv_bucket_name
  iv_objectkeyprefix = iv_prefix
).

" Create receipt action with S3 action
DATA(lo_action) = NEW /aws1/cl_sesreceiptaction(
  io_s3action = lo_s3_action
).

" Create list of actions
DATA lt_actions TYPE /aws1/cl_sesreceiptaction=>tt_receiptactionslist.
APPEND lo_action TO lt_actions.

" Create receipt rule
DATA(lo_rule) = NEW /aws1/cl_sesreceiptrule(
  iv_name = iv_rule_name
  iv_enabled = abap_true
  it_recipients = it_recipients
  it_actions = lt_actions
).

TRY.
    lo_ses->createreceiptrule(
      iv_rulesetname = iv_rule_set_name
      io_rule = lo_rule
    ).
    MESSAGE 'Receipt rule created successfully' TYPE 'I'.
  CATCH /aws1/cx_sesinvalids3confex INTO DATA(lo_ex1).
    DATA(lv_error) = |Invalid S3 configuration: { lo_ex1->get_text( ) }|.
```

```
      MESSAGE lv_error TYPE 'I'.
      RAISE EXCEPTION lo_ex1.
    CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
      lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
      MESSAGE lv_error TYPE 'I'.
      RAISE EXCEPTION lo_ex_generic.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateReceiptRule。

`CreateReceiptRuleSet`

以下程式碼範例顯示如何使用 `CreateReceiptRuleSet`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
      lo_ses->createreceiptruleset( iv_rulesetname = iv_rule_set_name ).
      MESSAGE 'Receipt rule set created successfully' TYPE 'I'.
    CATCH /aws1/cx_sesalreadyexistsex INTO DATA(lo_ex1).
      DATA(lv_error) = |Rule set already exists: { lo_ex1->get_text( ) }|.
      MESSAGE lv_error TYPE 'I'.
      RAISE EXCEPTION lo_ex1.
    CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
      lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
      MESSAGE lv_error TYPE 'I'.
      RAISE EXCEPTION lo_ex_generic.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateReceiptRuleSet。

`CreateTemplate`

以下程式碼範例顯示如何使用 `CreateTemplate`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
DATA(lo_template) = NEW /aws1/cl_sestemplate(
  iv_templatename = iv_name
  iv_subjectpart = iv_subject
  iv_textpart = iv_text
  iv_htmlpart = iv_html
).

TRY.
    lo_ses->createtemplate( io_template = lo_template ).
    MESSAGE 'Template created successfully' TYPE 'I'.
  CATCH /aws1/cx_sesalreadyexistsex INTO DATA(lo_ex1).
    DATA(lv_error) = |Template already exists: { lo_ex1->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex1.
  CATCH /aws1/cx_sesinvalidtemplateex INTO DATA(lo_ex2).
    lv_error = |Invalid template: { lo_ex2->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex2.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
    lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex_generic.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateTemplate。

DeleteIdentity

以下程式碼範例顯示如何使用 DeleteIdentity。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_ses->deleteidentity( iv_identity = iv_identity ).
    MESSAGE 'Identity deleted successfully' TYPE 'I'.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).
    DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteIdentity。

DeleteReceiptFilter

以下程式碼範例顯示如何使用 DeleteReceiptFilter。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_ses->deletereceiptfilter( iv_filtername = iv_filter_name ).
    MESSAGE 'Receipt filter deleted successfully' TYPE 'I'.
```

```
  CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).
    DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteReceiptFilter。

`DeleteReceiptRule`

以下程式碼範例顯示如何使用 `DeleteReceiptRule`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_ses->deletereceiptrule(
      iv_rulesetname = iv_rule_set_name
      iv_rulename = iv_rule_name
    ).
    MESSAGE 'Receipt rule deleted successfully' TYPE 'I'.
  CATCH /aws1/cx_sesrulesetdoesnotexex INTO DATA(lo_ex1).
    DATA(lv_error) = |Rule set does not exist: { lo_ex1->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex1.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
    lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex_generic.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteReceiptRule。

`DeleteReceiptRuleSet`

以下程式碼範例顯示如何使用 `DeleteReceiptRuleSet`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_ses->deletereceiptruleset( iv_rulesetname = iv_rule_set_name ).
    MESSAGE 'Receipt rule set deleted successfully' TYPE 'I'.
  CATCH /aws1/cx_sescannotdeleteex INTO DATA(lo_ex1).
    DATA(lv_error) = |Cannot delete rule set: { lo_ex1->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex1.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
    lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex_generic.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteReceiptRuleSet。

`DeleteTemplate`

以下程式碼範例顯示如何使用 `DeleteTemplate`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_ses->deletetemplate( iv_templatename = iv_template_name ).
    MESSAGE 'Template deleted successfully' TYPE 'I'.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).
    DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteTemplate。

`DescribeReceiptRuleSet`

以下程式碼範例顯示如何使用 `DescribeReceiptRuleSet`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_ses->describereceiptruleset(
      iv_rulesetname = iv_rule_set_name
    ).
    MESSAGE 'Receipt rule set described successfully' TYPE 'I'.
  CATCH /aws1/cx_sesrulesetdoesnotexex INTO DATA(lo_ex1).
    DATA(lv_error) = |Rule set does not exist: { lo_ex1->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex1.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
    lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex_generic.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeReceiptRuleSet。

`GetIdentityVerificationAttributes`

以下程式碼範例顯示如何使用 `GetIdentityVerificationAttributes`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   DATA lt_identities TYPE /aws1/cl_sesidentitylist_w=>tt_identitylist.
   APPEND NEW /aws1/cl_sesidentitylist_w( iv_value = iv_identity ) TO
lt_identities.

   TRY.
       DATA(lo_result) = lo_ses->getidentityverificationattrs(
         it_identities = lt_identities
       ).

       DATA(lt_attrs) = lo_result->get_verificationattributes( ).
       IF lt_attrs IS NOT INITIAL.
         LOOP AT lt_attrs ASSIGNING FIELD-SYMBOL(<ls_attr>).
           ov_status = <ls_attr>-value->get_verificationstatus( ).
           EXIT.
         ENDLOOP.
       ELSE.
         ov_status = 'NotFound'.
       ENDIF.
     CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).
       DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.
       MESSAGE lv_error TYPE 'I'.
       RAISE EXCEPTION lo_ex.
   ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetIdentityVerificationAttributes。

GetTemplate

以下程式碼範例顯示如何使用 GetTemplate。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    DATA(lo_result) = lo_ses->gettemplate( iv_templatename = iv_template_name ).
    oo_template = lo_result->get_template( ).
    MESSAGE 'Template retrieved successfully' TYPE 'I'.
  CATCH /aws1/cx_sestmpldoesnotexistex INTO DATA(lo_ex1).
    DATA(lv_error) = |Template does not exist: { lo_ex1->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex1.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
    lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex_generic.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetTemplate。

ListIdentities

以下程式碼範例顯示如何使用 ListIdentities。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    DATA(lo_result) = lo_ses->listidentities(
      iv_identitytype = iv_identity_type
      iv_maxitems = iv_max_items
    ).
    ot_identities = lo_result->get_identities( ).
    MESSAGE 'Identities retrieved successfully' TYPE 'I'.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).
    DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListIdentities。

`ListReceiptFilters`

以下程式碼範例顯示如何使用 `ListReceiptFilters`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    DATA(lo_result) = lo_ses->listreceiptfilters( ).
    ot_filters = lo_result->get_filters( ).
```

```
      MESSAGE 'Receipt filters retrieved successfully' TYPE 'I'.
    CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).
      DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.
      MESSAGE lv_error TYPE 'I'.
      RAISE EXCEPTION lo_ex.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListReceiptFilters。

`ListTemplates`

以下程式碼範例顯示如何使用 `ListTemplates`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
      DATA(lo_result) = lo_ses->listtemplates( iv_maxitems = iv_max_items ).
      ot_templates = lo_result->get_templatesmetadata( ).
      MESSAGE 'Templates retrieved successfully' TYPE 'I'.
    CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).
      DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.
      MESSAGE lv_error TYPE 'I'.
      RAISE EXCEPTION lo_ex.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListTemplates。

`SendEmail`

以下程式碼範例顯示如何使用 `SendEmail`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
" Create message object
DATA(lo_subject) = NEW /aws1/cl_sescontent( iv_data = iv_subject ).
DATA(lo_text_body) = NEW /aws1/cl_sescontent( iv_data = iv_text ).
DATA(lo_html_body) = NEW /aws1/cl_sescontent( iv_data = iv_html ).
DATA(lo_body) = NEW /aws1/cl_sesbody(
  io_text = lo_text_body
  io_html = lo_html_body
).
DATA(lo_message) = NEW /aws1/cl_sesmessage(
  io_subject = lo_subject
  io_body = lo_body
).

TRY.
    " Send email
    DATA(lo_result) = lo_ses->sendemail(
      iv_source = iv_source
      io_destination = io_destination
      io_message = lo_message
      it_replytoaddresses = it_reply_tos
    ).
    ov_msg_id = lo_result->get_messageid( ).
    MESSAGE 'Email sent successfully' TYPE 'I'.
  CATCH /aws1/cx_sesacctsendingpause00 INTO DATA(lo_ex1).
    DATA(lv_error) = |Account sending paused: { lo_ex1->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex1.
  CATCH /aws1/cx_sesmessagerejected INTO DATA(lo_ex2).
    lv_error = |Message rejected: { lo_ex2->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex2.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
    lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
```

```
        RAISE EXCEPTION lo_ex_generic.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 SendEmail。

`SendTemplatedEmail`

以下程式碼範例顯示如何使用 `SendTemplatedEmail`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    " Send templated email
    DATA(lo_result) = lo_ses->sendtemplatedemail(
      iv_source = iv_source
      io_destination = io_destination
      iv_template = iv_template_name
      iv_templatedata = iv_template_data
      it_replytoaddresses = it_reply_tos
    ).
    ov_msg_id = lo_result->get_messageid( ).
    MESSAGE 'Templated email sent successfully' TYPE 'I'.
  CATCH /aws1/cx_sestmpldoesnotexistex INTO DATA(lo_ex1).
    DATA(lv_error) = |Template does not exist: { lo_ex1->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex1.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
    lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex_generic.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 SendTemplatedEmail。

`UpdateTemplate`

以下程式碼範例顯示如何使用 `UpdateTemplate`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
DATA(lo_template) = NEW /aws1/cl_sestemplate(
  iv_templatename = iv_name
  iv_subjectpart = iv_subject
  iv_textpart = iv_text
  iv_htmlpart = iv_html
).

TRY.
    lo_ses->updatetemplate( io_template = lo_template ).
    MESSAGE 'Template updated successfully' TYPE 'I'.
  CATCH /aws1/cx_sestmpldoesnotexistex INTO DATA(lo_ex1).
    DATA(lv_error) = |Template does not exist: { lo_ex1->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex1.
  CATCH /aws1/cx_sesinvalidtemplateex INTO DATA(lo_ex2).
    lv_error = |Invalid template: { lo_ex2->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex2.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_ex_generic).
    lv_error = |An error occurred: { lo_ex_generic->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex_generic.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 UpdateTemplate。

`VerifyDomainIdentity`

以下程式碼範例顯示如何使用 `VerifyDomainIdentity`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        DATA(lo_result) = lo_ses->verifydomainidentity( iv_domain =
iv_domain_name ).
        ov_token = lo_result->get_verificationtoken( ).
        MESSAGE 'Domain verification initiated' TYPE 'I'.
      CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).
        DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_ex.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 VerifyDomainIdentity。

`VerifyEmailIdentity`

以下程式碼範例顯示如何使用 `VerifyEmailIdentity`。

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_ses->verifyemailidentity( iv_emailaddress = iv_email_address ).
    MESSAGE 'Email verification initiated' TYPE 'I'.
  CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).
    DATA(lv_error) = |An error occurred: { lo_ex->get_text( ) }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_ex.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 VerifyEmailIdentity。

使用適用於 SAP ABAP 的 SDK 的 Amazon SES API v2 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 Amazon SES API v2 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

`CreateContact`

以下程式碼範例顯示如何使用 `CreateContact`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_se2->createcontact(
      iv_contactlistname = iv_contact_list_name
      iv_emailaddress = iv_email_address ).
    MESSAGE 'Contact created successfully.' TYPE 'I'.
  CATCH /aws1/cx_se2alreadyexistsex.
    MESSAGE 'Contact already exists.' TYPE 'I'.
  CATCH /aws1/cx_se2badrequestex.
    MESSAGE 'Bad request.' TYPE 'E'.
  CATCH /aws1/cx_se2notfoundexception.
    MESSAGE 'Contact list not found.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateContact。

`CreateContactList`

以下程式碼範例顯示如何使用 `CreateContactList`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_se2->createcontactlist(
      iv_contactlistname = iv_contact_list_name ).
```

```
      MESSAGE 'Contact list created successfully.' TYPE 'I'.
    CATCH /aws1/cx_se2alreadyexistsex.
      MESSAGE 'Contact list already exists.' TYPE 'I'.
    CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).
      MESSAGE 'Bad request - contact list limit may be reached.' TYPE 'I'.
      " Re-raise the exception so the caller can handle it
      RAISE EXCEPTION lo_bad_request.
    CATCH /aws1/cx_se2limitexceededex INTO DATA(lo_limit_exceeded).
      MESSAGE 'Limit exceeded - contact list limit reached.' TYPE 'I'.
      " Re-raise the exception so the caller can handle it
      RAISE EXCEPTION lo_limit_exceeded.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateContactList。

`CreateEmailIdentity`

以下程式碼範例顯示如何使用 `CreateEmailIdentity`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
      lo_se2->createemailidentity(
        iv_emailidentity = iv_email_identity ).
      MESSAGE 'Email identity created successfully.' TYPE 'I'.
    CATCH /aws1/cx_se2alreadyexistsex.
      MESSAGE 'Email identity already exists.' TYPE 'I'.
    CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).
      MESSAGE lo_bad_request TYPE 'I' DISPLAY LIKE 'E'.
    CATCH /aws1/cx_se2limitexceededex INTO DATA(lo_limit_exceeded).
      MESSAGE lo_limit_exceeded TYPE 'I' DISPLAY LIKE 'E'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateEmailIdentity。

`CreateEmailTemplate`

以下程式碼範例顯示如何使用 `CreateEmailTemplate`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    DATA(lo_template_content) = NEW /aws1/cl_se2emailtmplcontent(
      iv_subject = iv_subject
      iv_html = iv_html
      iv_text = iv_text ).

    lo_se2->createemailtemplate(
      iv_templatename = iv_template_name
      io_templatecontent = lo_template_content ).
    MESSAGE 'Email template created successfully.' TYPE 'I'.
  CATCH /aws1/cx_se2alreadyexistsex.
    MESSAGE 'Email template already exists.' TYPE 'I'.
  CATCH /aws1/cx_se2badrequestex.
    MESSAGE 'Bad request.' TYPE 'E'.
  CATCH /aws1/cx_se2limitexceededex.
    MESSAGE 'Limit exceeded.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateEmailTemplate。

`DeleteContactList`

以下程式碼範例顯示如何使用 `DeleteContactList`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_se2->deletecontactlist(
      iv_contactlistname = iv_contact_list_name ).
    MESSAGE 'Contact list deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_se2notfoundexception.
    MESSAGE 'Contact list not found.' TYPE 'I'.
  CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).
    MESSAGE 'Bad request.' TYPE 'I'.
    RAISE EXCEPTION lo_bad_request.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteContactList。

`DeleteEmailIdentity`

以下程式碼範例顯示如何使用 `DeleteEmailIdentity`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_se2->deleteemailidentity(
      iv_emailidentity = iv_email_identity ).
    MESSAGE 'Email identity deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_se2notfoundexception.
```

```
      MESSAGE 'Email identity not found.' TYPE 'I'.
    CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).
      MESSAGE 'Bad request.' TYPE 'I'.
      RAISE EXCEPTION lo_bad_request.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteEmailIdentity。

`DeleteEmailTemplate`

以下程式碼範例顯示如何使用 `DeleteEmailTemplate`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
      lo_se2->deleteemailtemplate(
        iv_templatename = iv_template_name ).
      MESSAGE 'Email template deleted successfully.' TYPE 'I'.
    CATCH /aws1/cx_se2notfoundexception.
      MESSAGE 'Email template not found.' TYPE 'I'.
    CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).
      MESSAGE 'Bad request.' TYPE 'I'.
      RAISE EXCEPTION lo_bad_request.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteEmailTemplate。

`ListContacts`

以下程式碼範例顯示如何使用 `ListContacts`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_se2->listcontacts(
      iv_contactlistname = iv_contact_list_name ).
    DATA(lv_count) = lines( oo_result->get_contacts( ) ).
    MESSAGE |Retrieved { lv_count } contacts from list.| TYPE 'I'.
  CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).
    MESSAGE 'Bad request.' TYPE 'I'.
    RAISE EXCEPTION lo_bad_request.
  CATCH /aws1/cx_se2notfoundexception INTO DATA(lo_not_found).
    MESSAGE 'Contact list not found.' TYPE 'I'.
    RAISE EXCEPTION lo_not_found.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListContacts。

`SendEmail`

以下程式碼範例顯示如何使用 `SendEmail`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

傳送訊息。

```
TRY.
    " Create destination with recipient address
```

```
        DATA lt_to_addresses TYPE /aws1/
cl_se2emailaddresslist_w=>tt_emailaddresslist.
        APPEND NEW /aws1/cl_se2emailaddresslist_w( iv_value = iv_to_email_address )
 TO lt_to_addresses.
        DATA(lo_destination) = NEW /aws1/cl_se2destination(
          it_toaddresses = lt_to_addresses ).

        " Create message content
        DATA(lo_subject) = NEW /aws1/cl_se2content( iv_data = iv_subject ).
        DATA(lo_text_body) = NEW /aws1/cl_se2content( iv_data = iv_text_body ).
        DATA(lo_html_body) = NEW /aws1/cl_se2content( iv_data = iv_html_body ).
        DATA(lo_body) = NEW /aws1/cl_se2body(
          io_text = lo_text_body
          io_html = lo_html_body ).
        DATA(lo_message) = NEW /aws1/cl_se2message(
          io_subject = lo_subject
          io_body = lo_body ).

        DATA(lo_content) = NEW /aws1/cl_se2emailcontent(
          io_simple = lo_message ).

        " Send the email
        lo_se2->sendemail(
          iv_fromemailaddress = iv_from_email_address
          io_destination = lo_destination
          io_content = lo_content ).
        MESSAGE 'Email sent successfully.' TYPE 'I'.
      CATCH /aws1/cx_se2accountsuspendedex INTO DATA(lo_account_suspended).
        MESSAGE 'Account suspended.' TYPE 'I'.
        RAISE EXCEPTION lo_account_suspended.
      CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).
        MESSAGE 'Bad request.' TYPE 'I'.
        RAISE EXCEPTION lo_bad_request.
      CATCH /aws1/cx_se2messagerejected INTO DATA(lo_message_rejected).
        MESSAGE 'Message rejected - check email verification.' TYPE 'I'.
        RAISE EXCEPTION lo_message_rejected.
    ENDTRY.
```

使用範本傳送訊息。

```
    TRY.
        " Create destination with recipient address
```

```
        DATA lt_to_addresses TYPE /aws1/
cl_se2emailaddresslist_w=>tt_emailaddresslist.
        APPEND NEW /aws1/cl_se2emailaddresslist_w( iv_value = iv_to_email_address )
 TO lt_to_addresses.
        DATA(lo_destination) = NEW /aws1/cl_se2destination(
          it_toaddresses = lt_to_addresses ).

        " Create template reference
        DATA(lo_template) = NEW /aws1/cl_se2template(
          iv_templatename = iv_template_name
          iv_templatedata = iv_template_data ).

        DATA(lo_content) = NEW /aws1/cl_se2emailcontent(
          io_template = lo_template ).

        " Create list management options
        DATA(lo_list_mgmt) = NEW /aws1/cl_se2listmanagementopts(
          iv_contactlistname = iv_contact_list_name ).

        " Send the email using template
        lo_se2->sendemail(
          iv_fromemailaddress = iv_from_email_address
          io_destination = lo_destination
          io_content = lo_content
          io_listmanagementoptions = lo_list_mgmt ).
        MESSAGE 'Email sent using template successfully.' TYPE 'I'.
      CATCH /aws1/cx_se2accountsuspendedex INTO DATA(lo_account_suspended).
        MESSAGE 'Account suspended.' TYPE 'I'.
        RAISE EXCEPTION lo_account_suspended.
      CATCH /aws1/cx_se2badrequestex INTO DATA(lo_bad_request).
        MESSAGE 'Bad request.' TYPE 'I'.
        RAISE EXCEPTION lo_bad_request.
      CATCH /aws1/cx_se2messagerejected INTO DATA(lo_message_rejected).
        MESSAGE 'Message rejected - check email verification.' TYPE 'I'.
        RAISE EXCEPTION lo_message_rejected.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 SendEmail。

使用適用於 SAP ABAP 的 SDK 的 Amazon SNS 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 Amazon SNS 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

案例是向您展示如何呼叫服務中的多個函數或與其他 AWS 服務組合來完成特定任務的程式碼範例。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作
- 案例

動作

`CreateTopic`

以下程式碼範例顯示如何使用 `CreateTopic`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP API 的AWS SDK 參考》中的 CreateTopic。

`DeleteTopic`

以下程式碼範例顯示如何使用 `DeleteTopic`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
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.
```

- 如需 API 詳細資訊，請參閱《適用於 Kotlin 的AWS SDK API 參考》中的 DeleteTopic。

`GetTopicAttributes`

以下程式碼範例顯示如何使用 `GetTopicAttributes`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 GetTopicAttributes。

`ListSubscriptions`

以下程式碼範例顯示如何使用 `ListSubscriptions`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的ListSubscriptions。

`ListTopics`

以下程式碼範例顯示如何使用 `ListTopics`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 ListTopics。

`Publish`

以下程式碼範例顯示如何使用 `Publish`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   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.
```

將具有屬性的訊息發佈至主題。

```
    TRY.
        oo_result = lo_sns->publish(              " oo_result is returned for
testing purposes. "
          iv_topicarn = iv_topic_arn
          iv_message = iv_message
          it_messageattributes = it_msg_attrs ).
        MESSAGE 'Message with attributes published to SNS topic.' TYPE 'I'.
      CATCH /aws1/cx_snsnotfoundexception.
        MESSAGE 'Topic does not exist.' TYPE 'E'.
    ENDTRY.
```

將多格式訊息發佈至主題。

```
    " Build JSON message structure for multi-format message
    DATA(lv_json_message) = |\{ "default": "{ iv_default_message }", "sms":
"{ iv_sms_message }", "email": "{ iv_email_message }" \}|.

    TRY.
        oo_result = lo_sns->publish(              " oo_result is returned for
testing purposes. "
          iv_topicarn = iv_topic_arn
          iv_message = lv_json_message
          iv_subject = iv_subject
          iv_messagestructure = 'json' ).
        MESSAGE 'Multi-format message published to SNS topic.' TYPE 'I'.
      CATCH /aws1/cx_snsnotfoundexception.
        MESSAGE 'Topic does not exist.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的發佈。

SetSubscriptionAttributes

以下程式碼範例顯示如何使用 SetSubscriptionAttributes。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_sns->setsubscriptionattributes(
        iv_subscriptionarn = iv_subscription_arn
        iv_attributename  = 'FilterPolicy'
        iv_attributevalue = iv_filter_policy ).
    MESSAGE 'Added filter policy to subscription.' TYPE 'I'.
  CATCH /aws1/cx_snsnotfoundexception.
    MESSAGE 'Subscription does not exist.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 SetSubscriptionAttributes。

`SetTopicAttributes`

以下程式碼範例顯示如何使用 `SetTopicAttributes`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 SetTopicAttributes。

Subscribe

以下程式碼範例顯示如何使用 Subscribe。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

透過電子郵件地址訂閱某個主題。

```
  TRY.
      oo_result = lo_sns->subscribe(                    "oo_result is returned
for testing purposes."
              iv_topicarn = iv_topic_arn
              iv_protocol = 'email'
              iv_endpoint = iv_email_address
              iv_returnsubscriptionarn = abap_true ).
      MESSAGE 'Email address subscribed to SNS topic.' TYPE 'I'.
    CATCH /aws1/cx_snsnotfoundexception.
      MESSAGE 'Topic does not exist.' TYPE 'E'.
    CATCH /aws1/cx_snssubscriptionlmte00.
      MESSAGE 'Unable to create subscriptions. You have reached the maximum number
of subscriptions allowed.' TYPE 'E'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的訂閱。

`Unsubscribe`

以下程式碼範例顯示如何使用 `Unsubscribe`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的取消訂閱。

案例

建立並發布到 FIFO 主題

下列程式碼範例示範如何建立並發布到 FIFO Amazon SNS 主題。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

建立 FIFO 主題、將 Amazon SQS FIFO 佇列訂閱至主題，然後將訊息發佈至 Amazon SNS 主題。

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

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

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

```
    ENDTRY.

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

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

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的下列主題。
  - CreateTopic
  - 發布
  - Subscribe

發布簡訊

下列程式碼範例示範如何使用 Amazon SNS 發佈訊息。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    " iv_phone_number = '+12065550101' - Phone number in E.164 format
    TRY.
        oo_result = lo_sns->publish(              " oo_result is returned for
testing purposes. "
          iv_phonenumber = iv_phone_number
          iv_message = iv_message ).
        MESSAGE 'Message published to phone number.' TYPE 'I'.
      CATCH /aws1/cx_snsnotfoundexception.
        MESSAGE 'Phone number does not exist.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的發佈。

使用適用於 SAP ABAP 的 SDK 的 Amazon SQS 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 Amazon SQS 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

案例是向您展示如何呼叫服務中的多個函數或與其他 AWS 服務組合來完成特定任務的程式碼範例。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作
- 案例

動作

`CreateQueue`

以下程式碼範例顯示如何使用 `CreateQueue`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

建立 Amazon SQS 標準佇列。

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

建立等待訊息送達的 Amazon SQS 佇列。

```
    TRY.
        DATA lt_attributes TYPE /aws1/cl_sqsqueueattrmap_w=>tt_queueattributemap.
        DATA ls_attribute TYPE /aws1/
cl_sqsqueueattrmap_w=>ts_queueattributemap_maprow.
        ls_attribute-key = 'ReceiveMessageWaitTimeSeconds'.               " Time in
 seconds for long polling, such as how long the call waits for a message to arrive
 in the queue before returning. "
        ls_attribute-value = NEW /aws1/cl_sqsqueueattrmap_w( iv_value =
 iv_wait_time ).
        INSERT ls_attribute INTO TABLE lt_attributes.
        oo_result = lo_sqs->createqueue(                  " oo_result is returned
 for testing purposes. "
```

```
                iv_queuename = iv_queue_name
                it_attributes = lt_attributes ).
        MESSAGE 'SQS queue created.' TYPE 'I'.
      CATCH /aws1/cx_sqsqueuedeldrecently.
        MESSAGE 'After deleting a queue, wait 60 seconds before creating another
queue with the same name.' TYPE 'E'.
      CATCH /aws1/cx_sqsqueuenameexists.
        MESSAGE 'A queue with this name already exists.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 CreateQueue。

`DeleteMessage`

以下程式碼範例顯示如何使用 `DeleteMessage`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        lo_sqs->deletemessage(
           iv_queueurl = iv_queue_url
           iv_receipthandle = iv_receipt_handle ).
        MESSAGE 'Message deleted from SQS queue.' TYPE 'I'.
      CATCH /aws1/cx_sqsinvalididformat.
        MESSAGE 'The specified receipt handle is not valid.' TYPE 'E'.
      CATCH /aws1/cx_sqsreceipthandleisinv.
        MESSAGE 'The specified receipt handle is not valid for the current version.'
TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteMessage。

DeleteMessageBatch

以下程式碼範例顯示如何使用 DeleteMessageBatch。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   TRY.
       oo_result = lo_sqs->deletemessagebatch(       " oo_result is returned for
testing purposes. "
          iv_queueurl = iv_queue_url
          it_entries = it_entries ).
       MESSAGE 'Messages deleted from SQS queue.' TYPE 'I'.
     CATCH /aws1/cx_sqsbtcentidsnotdist00.
       MESSAGE 'Two or more batch entries in the request have the same ID.' TYPE
'E'.
     CATCH /aws1/cx_sqsemptybatchrequest.
       MESSAGE 'The batch request does not contain any entries.' TYPE 'E'.
     CATCH /aws1/cx_sqsinvbatchentryid.
       MESSAGE 'The ID of a batch entry in a batch request is not valid.' TYPE 'E'.
     CATCH /aws1/cx_sqstoomanyentriesin00.
       MESSAGE 'The batch request contains more entries than allowed.' TYPE 'E'.
   ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteMessageBatch。

DeleteQueue

以下程式碼範例顯示如何使用 DeleteQueue。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 DeleteQueue。

`GetQueueUrl`

以下程式碼範例顯示如何使用 `GetQueueUrl`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 GetQueueUrl。

`ListQueues`

以下程式碼範例顯示如何使用 `ListQueues`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 ListQueues。

`ReceiveMessage`

以下程式碼範例顯示如何使用 `ReceiveMessage`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

接收 Amazon SQS 佇列的訊息。

```
    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.
```

使用長輪詢支援從 Amazon SQS 佇列接收訊息。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 ReceiveMessage。

SendMessage

以下程式碼範例顯示如何使用 SendMessage。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 SendMessage。

SendMessageBatch

以下程式碼範例顯示如何使用 SendMessageBatch。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
      oo_result = lo_sqs->sendmessagebatch(         " oo_result is returned for
testing purposes. "
        iv_queueurl = iv_queue_url
        it_entries = it_messages ).
      MESSAGE 'Messages sent to SQS queue.' TYPE 'I'.
    CATCH /aws1/cx_sqsbtcentidsnotdist00.
      MESSAGE 'Two or more batch entries in the request have the same ID.' TYPE
'E'.
    CATCH /aws1/cx_sqsbatchreqtoolong.
      MESSAGE 'The length of all the messages put together is more than the
limit.' TYPE 'E'.
    CATCH /aws1/cx_sqsemptybatchrequest.
      MESSAGE 'The batch request does not contain any entries.' TYPE 'E'.
    CATCH /aws1/cx_sqsinvbatchentryid.
      MESSAGE 'The ID of a batch entry in a batch request is not valid.' TYPE 'E'.
    CATCH /aws1/cx_sqstoomanyentriesin00.
      MESSAGE 'The batch request contains more entries than allowed.' TYPE 'E'.
    CATCH /aws1/cx_sqsunsupportedop.
      MESSAGE 'Operation not supported.' TYPE 'E'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 SendMessageBatch。

案例

建立並發布到 FIFO 主題

下列程式碼範例示範如何建立並發布到 FIFO Amazon SNS 主題。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

建立 FIFO 主題、將 Amazon SQS FIFO 佇列訂閱至主題，然後將訊息發佈至 Amazon SNS 主題。

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

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

```
        MESSAGE 'Unable to create more topics. You have reached the maximum number
 of topics allowed.' TYPE 'E'.
    ENDTRY.

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

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

        DATA(lo_result) = lo_sns->publish(
             iv_topicarn = lv_topic_arn
             iv_message = 'The price of your mobile plan has been increased from $19
 to $23'
             iv_subject = 'Changes to mobile plan'
             iv_messagegroupid = 'Update-2'
             iv_messagededuplicationid = 'Update-2.1'
```

```
            it_messageattributes = lt_msg_attributes ).
        ov_message_id = lo_result->get_messageid( ).                    "
ov_message_id is returned for testing purposes. "
        MESSAGE 'Message was published to SNS topic.' TYPE 'I'.
      CATCH /aws1/cx_snsnotfoundexception.
        MESSAGE 'Topic does not exist.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的下列主題。
  - CreateTopic
  - 發布
  - Subscribe

使用適用於 SAP ABAP 的 SDK 的 Step Functions 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 Step Functions 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

`CreateActivity`

以下程式碼範例顯示如何使用 `CreateActivity`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    DATA(lo_result) = lo_sfn->createactivity(
      iv_name = iv_name
    ).
    ov_activity_arn = lo_result->get_activityarn( ).
    MESSAGE 'Activity created successfully.' TYPE 'I'.
  CATCH /aws1/cx_sfnactivityalrdyex.
    MESSAGE 'Activity already exists.' TYPE 'E'.
  CATCH /aws1/cx_sfninvalidname.
    MESSAGE 'Invalid activity name.' TYPE 'E'.
  CATCH /aws1/cx_sfnactivitylimitexcd.
    MESSAGE 'Activity limit exceeded.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateActivity。

`CreateStateMachine`

以下程式碼範例顯示如何使用 `CreateStateMachine`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    DATA(lo_result) = lo_sfn->createstatemachine(
```

```
        iv_name = iv_name
        iv_definition = iv_definition
        iv_rolearn = iv_role_arn
      ).
      ov_state_machine_arn = lo_result->get_statemachinearn( ).
      MESSAGE 'State machine created successfully.' TYPE 'I'.
    CATCH /aws1/cx_sfnstatemachinealrex.
      MESSAGE 'State machine already exists.' TYPE 'E'.
    CATCH /aws1/cx_sfninvaliddefinition.
      MESSAGE 'Invalid state machine definition.' TYPE 'E'.
    CATCH /aws1/cx_sfninvalidname.
      MESSAGE 'Invalid state machine name.' TYPE 'E'.
    CATCH /aws1/cx_sfninvalidarn.
      MESSAGE 'Invalid role ARN.' TYPE 'E'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateStateMachine。

`DeleteActivity`

以下程式碼範例顯示如何使用 `DeleteActivity`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
      lo_sfn->deleteactivity(
        iv_activityarn = iv_activity_arn
      ).
      MESSAGE 'Activity deleted successfully.' TYPE 'I'.
    CATCH /aws1/cx_sfninvalidarn.
      MESSAGE 'Invalid activity ARN.' TYPE 'E'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteActivity。

`DeleteStateMachine`

以下程式碼範例顯示如何使用 `DeleteStateMachine`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_sfn->deletestatemachine(
      iv_statemachinearn = iv_state_machine_arn
    ).
    MESSAGE 'State machine deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_sfninvalidarn.
    MESSAGE 'Invalid state machine ARN.' TYPE 'E'.
  CATCH /aws1/cx_sfnvalidationex.
    MESSAGE 'Validation error occurred.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteStateMachine。

`DescribeExecution`

以下程式碼範例顯示如何使用 `DescribeExecution`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_sfn->describeexecution(
      iv_executionarn = iv_execution_arn
    ).
    MESSAGE 'Execution described successfully.' TYPE 'I'.
  CATCH /aws1/cx_sfnexecdoesnotexist.
    MESSAGE 'Execution does not exist.' TYPE 'E'.
  CATCH /aws1/cx_sfninvalidarn.
    MESSAGE 'Invalid execution ARN.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeExecution。

`DescribeStateMachine`

以下程式碼範例顯示如何使用 `DescribeStateMachine`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_sfn->describestatemachine(
      iv_statemachinearn = iv_state_machine_arn
    ).
    MESSAGE 'State machine described successfully.' TYPE 'I'.
  CATCH /aws1/cx_sfnstatemachinedoes00.
    MESSAGE 'State machine does not exist.' TYPE 'E'.
  CATCH /aws1/cx_sfninvalidarn.
    MESSAGE 'Invalid state machine ARN.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeStateMachine。

GetActivityTask

以下程式碼範例顯示如何使用 GetActivityTask。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_sfn->getactivitytask(
      iv_activityarn = iv_activity_arn
    ).
    MESSAGE 'Activity task retrieved successfully.' TYPE 'I'.
  CATCH /aws1/cx_sfnactivitydoesnotex.
    MESSAGE 'Activity does not exist.' TYPE 'E'.
  CATCH /aws1/cx_sfninvalidarn.
    MESSAGE 'Invalid activity ARN.' TYPE 'E'.
  CATCH /aws1/cx_sfnactivityworkerlm00.
    MESSAGE 'Activity worker limit exceeded.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetActivityTask。

ListActivities

以下程式碼範例顯示如何使用 ListActivities。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    DATA(lo_result) = lo_sfn->listactivities( ).
    DATA(lt_activities) = lo_result->get_activities( ).
    LOOP AT lt_activities INTO DATA(lo_activity).
      IF lo_activity->get_name( ) = iv_name.
        ov_activity_arn = lo_activity->get_activityarn( ).
        EXIT.
      ENDIF.
    ENDLOOP.
    MESSAGE 'Activities listed successfully.' TYPE 'I'.
  CATCH /aws1/cx_sfninvalidtoken.
    MESSAGE 'Invalid pagination token.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListActivities。

`ListStateMachines`

以下程式碼範例顯示如何使用 `ListStateMachines`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    DATA(lo_result) = lo_sfn->liststatemachines( ).
    DATA(lt_state_machines) = lo_result->get_statemachines( ).
```

```
    LOOP AT lt_state_machines INTO DATA(lo_state_machine).
      IF lo_state_machine->get_name( ) = iv_name.
        ov_state_machine_arn = lo_state_machine->get_statemachinearn( ).
        EXIT.
      ENDIF.
    ENDLOOP.
    MESSAGE 'State machines listed successfully.' TYPE 'I'.
  CATCH /aws1/cx_sfninvalidtoken.
    MESSAGE 'Invalid pagination token.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListStateMachines。

SendTaskSuccess

以下程式碼範例顯示如何使用 SendTaskSuccess。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_sfn->sendtasksuccess(
      iv_tasktoken = iv_task_token
      iv_output = iv_task_response
    ).
    MESSAGE 'Task success sent successfully.' TYPE 'I'.
  CATCH /aws1/cx_sfninvalidtoken.
    MESSAGE 'Invalid task token.' TYPE 'E'.
  CATCH /aws1/cx_sfntaskdoesnotexist.
    MESSAGE 'Task does not exist.' TYPE 'E'.
  CATCH /aws1/cx_sfninvalidoutput.
    MESSAGE 'Invalid task output.' TYPE 'E'.
  CATCH /aws1/cx_sfntasktimedout.
    MESSAGE 'Task timed out.' TYPE 'E'.
```

```
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 SendTaskSuccess。

`StartExecution`

以下程式碼範例顯示如何使用 `StartExecution`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    DATA(lo_result) = lo_sfn->startexecution(
      iv_statemachinearn = iv_state_machine_arn
      iv_input = iv_input
    ).
    ov_execution_arn = lo_result->get_executionarn( ).
    MESSAGE 'Execution started successfully.' TYPE 'I'.
  CATCH /aws1/cx_sfnstatemachinedoes00.
    MESSAGE 'State machine does not exist.' TYPE 'E'.
  CATCH /aws1/cx_sfninvalidarn.
    MESSAGE 'Invalid state machine ARN.' TYPE 'E'.
  CATCH /aws1/cx_sfninvalidexecinput.
    MESSAGE 'Invalid execution input.' TYPE 'E'.
  CATCH /aws1/cx_sfnexeclimitexceeded.
    MESSAGE 'Execution limit exceeded.' TYPE 'E'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 StartExecution。

使用適用於 SAP ABAP 的 SDK 的 Systems Manager 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 Systems Manager 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

`CreateDocument`

以下程式碼範例顯示如何使用 `CreateDocument`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_ssm->createdocument(
        iv_name = iv_name
        iv_content = iv_content
        iv_documenttype = 'Command' ).
    MESSAGE 'Document created.' TYPE 'I'.
  CATCH /aws1/cx_ssmdocalreadyexists.
    MESSAGE 'Document already exists.' TYPE 'I'.
  CATCH /aws1/cx_ssminvaliddoccontent.
    MESSAGE 'Invalid document content.' TYPE 'I'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateDocument。

`CreateMaintenanceWindow`

以下程式碼範例顯示如何使用 `CreateMaintenanceWindow`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_ssm->createmaintenancewindow(
        iv_name = iv_name
        iv_schedule = iv_schedule
        iv_duration = iv_duration
        iv_cutoff = iv_cutoff
        iv_allowunassociatedtargets = iv_allow_unassociated_targets ).
    MESSAGE 'Maintenance window created.' TYPE 'I'.
  CATCH /aws1/cx_ssmresrclimitexcdex.
    MESSAGE 'Resource limit exceeded.' TYPE 'I'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateMaintenanceWindow。

`CreateOpsItem`

以下程式碼範例顯示如何使用 `CreateOpsItem`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_ssm->createopsitem(
        iv_title = iv_title
        iv_source = iv_source
        iv_category = iv_category
        iv_severity = iv_severity
        iv_description = iv_description ).
    MESSAGE 'OpsItem created.' TYPE 'I'.
  CATCH /aws1/cx_ssmopsitemlimitexcdex.
    MESSAGE 'You have exceeded your open OpsItem limit.' TYPE 'I'.
  CATCH /aws1/cx_ssmopsitemalrdyexex.
    MESSAGE 'OpsItem already exists.' TYPE 'I'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateOpsItem。

`DeleteDocument`

以下程式碼範例顯示如何使用 `DeleteDocument`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
```

```
      lo_ssm->deletedocument( iv_name = iv_name ).
      MESSAGE 'Document deleted.' TYPE 'I'.
    CATCH /aws1/cx_ssminvaliddocument.
      MESSAGE 'Invalid document.' TYPE 'I'.
    CATCH /aws1/cx_ssmassocdinstances.
      MESSAGE 'Document has associated instances.' TYPE 'I'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteDocument。

`DeleteMaintenanceWindow`

以下程式碼範例顯示如何使用 `DeleteMaintenanceWindow`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
      lo_ssm->deletemaintenancewindow( iv_windowid = iv_window_id ).
      MESSAGE 'Maintenance window deleted.' TYPE 'I'.
    CATCH /aws1/cx_ssminternalservererr.
      MESSAGE 'Internal server error occurred.' TYPE 'I'.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteMaintenanceWindow。

`DeleteOpsItem`

以下程式碼範例顯示如何使用 `DeleteOpsItem`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_ssm->deleteopsitem( iv_opsitemid = iv_ops_item_id ).
    MESSAGE 'OpsItem deleted.' TYPE 'I'.
  CATCH /aws1/cx_ssmopsiteminvparamex.
    MESSAGE 'Invalid OpsItem parameter.' TYPE 'I'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteOpsItem。

`DescribeDocument`

以下程式碼範例顯示如何使用 `DescribeDocument`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    DATA(lo_result) = lo_ssm->describedocument( iv_name = iv_name ).
    DATA(lo_document) = lo_result->get_document( ).
    IF lo_document IS BOUND.
      rv_status = lo_document->get_status( ).
      MESSAGE |Document status: { rv_status }| TYPE 'I'.
    ENDIF.
  CATCH /aws1/cx_ssminvaliddocument.
    MESSAGE 'Invalid document.' TYPE 'I'.
```

```
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeDocument。

`DescribeOpsItems`

以下程式碼範例顯示如何使用 `DescribeOpsItems`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " Create filter for OpsItem ID
        DATA(lt_filters) = VALUE /aws1/cl_ssmopsitemfilter=>tt_opsitemfilters(
          ( NEW /aws1/cl_ssmopsitemfilter(
              iv_key = 'OpsItemId'
              it_values = VALUE /aws1/
cl_ssmopsitemfiltvals_w=>tt_opsitemfiltervalues(
                ( NEW /aws1/cl_ssmopsitemfiltvals_w( iv_value = iv_ops_item_id ) )
              )
              iv_operator = 'Equal'
            ) )
        ).

        " Use paginator to get all results
        DATA(lo_paginator) = lo_ssm->get_paginator( ).
        DATA(lo_iterator) = lo_paginator->describeopsitems(
          it_opsitemfilters = lt_filters ).

        rv_found = abap_false.

        WHILE lo_iterator->has_next( ).
          DATA(lo_result) = CAST /aws1/cl_ssmdescropsitemsrsp( lo_iterator-
>get_next( ) ).
```

```
          LOOP AT lo_result->get_opsitemsummaries( ) INTO DATA(lo_item).
            DATA(lv_title) = lo_item->get_title( ).
            DATA(lv_status) = lo_item->get_status( ).
            MESSAGE |The OpsItem title is { lv_title } and the status is
{ lv_status }| TYPE 'I'.
            rv_found = abap_true.
          ENDLOOP.
        ENDWHILE.
      CATCH /aws1/cx_ssminternalservererr.
        MESSAGE 'Internal server error occurred.' TYPE 'I'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeOpsItems。

`ListCommandInvocations`

以下程式碼範例顯示如何使用 `ListCommandInvocations`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " Use paginator to get all results
        DATA(lo_paginator) = lo_ssm->get_paginator( ).
        DATA(lo_iterator) = lo_paginator->listcommandinvocations(
          iv_instanceid = iv_instance_id ).

        DATA lv_count TYPE i VALUE 0.

        WHILE lo_iterator->has_next( ).
          DATA(lo_result) = CAST /aws1/cl_ssmlistcmdinvcsresult( lo_iterator-
>get_next( ) ).
          LOOP AT lo_result->get_commandinvocations( ) INTO DATA(lo_invocation).
            lv_count = lv_count + 1.
```

```
            DATA(lv_requested_datetime) = lo_invocation->get_requesteddatetime( ).
            MESSAGE |Command invocation requested at: { lv_requested_datetime }|
TYPE 'I'.
          ENDLOOP.
        ENDWHILE.

        MESSAGE |{ lv_count } command invocation(s) found for instance
{ iv_instance_id }.| TYPE 'I'.
      CATCH /aws1/cx_ssminvalidinstanceid.
        MESSAGE 'Invalid instance ID.' TYPE 'I'.
      CATCH /aws1/cx_ssminvalidcommandid.
        MESSAGE 'Invalid command ID.' TYPE 'I'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListCommandInvocations。

SendCommand

以下程式碼範例顯示如何使用 SendCommand。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        DATA(lo_result) = lo_ssm->sendcommand(
            it_instanceids = it_instance_ids
            iv_documentname = iv_document_name
            iv_timeoutseconds = 3600 ).
        DATA(lo_command) = lo_result->get_command( ).
        IF lo_command IS BOUND.
          rv_command_id = lo_command->get_commandid( ).
          MESSAGE 'Command sent successfully.' TYPE 'I'.
        ENDIF.
      CATCH /aws1/cx_ssminvaliddocument.
```

```
      MESSAGE 'Invalid document.' TYPE 'I'.
    CATCH /aws1/cx_ssminvalidinstanceid.
      MESSAGE 'Invalid instance ID.' TYPE 'I'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 SendCommand。

`UpdateMaintenanceWindow`

以下程式碼範例顯示如何使用 `UpdateMaintenanceWindow`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_ssm->updatemaintenancewindow(
        iv_windowid = iv_window_id
        iv_name = iv_name
        iv_enabled = iv_enabled
        iv_schedule = iv_schedule
        iv_duration = iv_duration
        iv_cutoff = iv_cutoff
        iv_allowunassociatedtargets = iv_allow_unassociated_targets ).
    MESSAGE 'Maintenance window updated.' TYPE 'I'.
  CATCH /aws1/cx_ssmdoesnotexistex.
    MESSAGE 'Maintenance window does not exist.' TYPE 'I'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 UpdateMaintenanceWindow。

`UpdateOpsItem`

以下程式碼範例顯示如何使用 `UpdateOpsItem`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    lo_ssm->updateopsitem(
        iv_opsitemid = iv_ops_item_id
        iv_title = iv_title
        iv_description = iv_description
        iv_status = iv_status ).
    MESSAGE 'OpsItem updated.' TYPE 'I'.
  CATCH /aws1/cx_ssmopsitemnotfoundex.
    MESSAGE 'OpsItem not found.' TYPE 'I'.
  CATCH /aws1/cx_ssmopsiteminvparamex.
    MESSAGE 'Invalid OpsItem parameter.' TYPE 'I'.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 UpdateOpsItem。

使用適用於 SAP ABAP 的 SDK 的 Amazon Textract 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 Amazon Textract 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

案例是向您展示如何呼叫服務中的多個函數或與其他 AWS 服務組合來完成特定任務的程式碼範例。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作
- 案例

動作

`AnalyzeDocument`

以下程式碼範例顯示如何使用 `AnalyzeDocument`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   "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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的 AWS SDK API 參考》中的 AnalyzeDocument。

DetectDocumentText

以下程式碼範例顯示如何使用 DetectDocumentText。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   "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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 DetectDocumentText。

`GetDocumentAnalysis`

以下程式碼範例顯示如何使用 `GetDocumentAnalysis`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

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

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 GetDocumentAnalysis。

`StartDocumentAnalysis`

以下程式碼範例顯示如何使用 `StartDocumentAnalysis`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   "Starts the asynchronous analysis of an input document for relationships"
   "between detected items such as key-value pairs, tables, and selection
elements."

   "Create ABAP objects for feature type."
   "Add TABLES to return information about the tables."
   "Add FORMS to return detected form data."
   "To perform both types of analysis, add TABLES and FORMS to FeatureTypes."

   DATA(lt_featuretypes) = VALUE /aws1/cl_texfeaturetypes_w=>tt_featuretypes(
     ( NEW /aws1/cl_texfeaturetypes_w( iv_value = 'FORMS' ) )
     ( NEW /aws1/cl_texfeaturetypes_w( iv_value = 'TABLES' ) ) ).
   "Create an ABAP object for the Amazon S3 object."
   DATA(lo_s3object) = NEW /aws1/cl_texs3object( iv_bucket = iv_s3bucket
     iv_name   = iv_s3object ).
   "Create an ABAP object for the document."
   DATA(lo_documentlocation) = NEW /aws1/cl_texdocumentlocation( io_s3object =
lo_s3object ).

   "Start async document analysis."
   TRY.
       oo_result = lo_tex->startdocumentanalysis(      "oo_result is returned for
testing purposes."
         io_documentlocation     = lo_documentlocation
         it_featuretypes         = lt_featuretypes ).
       DATA(lv_jobid) = oo_result->get_jobid( ).

       MESSAGE 'Document analysis started.' TYPE 'I'.
     CATCH /aws1/cx_texaccessdeniedex.
       MESSAGE 'You do not have permission to perform this action.' TYPE 'E'.
     CATCH /aws1/cx_texbaddocumentex.
       MESSAGE 'Amazon Textract is not able to read the document.' TYPE 'E'.
     CATCH /aws1/cx_texdocumenttoolargeex.
       MESSAGE 'The document is too large.' TYPE 'E'.
     CATCH /aws1/cx_texidempotentprmmis00.
       MESSAGE 'Idempotent parameter mismatch exception.' TYPE 'E'.
     CATCH /aws1/cx_texinternalservererr.
       MESSAGE 'Internal server error.' TYPE 'E'.
     CATCH /aws1/cx_texinvalidkmskeyex.
       MESSAGE 'AWS KMS key is not valid.' TYPE 'E'.
     CATCH /aws1/cx_texinvalidparameterex.
       MESSAGE 'Request has non-valid parameters.' TYPE 'E'.
```

```
      CATCH /aws1/cx_texinvalids3objectex.
        MESSAGE 'Amazon S3 object is not valid.' TYPE 'E'.
      CATCH /aws1/cx_texlimitexceededex.
        MESSAGE 'An Amazon Textract service limit was exceeded.' TYPE 'E'.
      CATCH /aws1/cx_texprovthruputexcdex.
        MESSAGE 'Provisioned throughput exceeded limit.' TYPE 'E'.
      CATCH /aws1/cx_texthrottlingex.
        MESSAGE 'The request processing exceeded the limit.' TYPE 'E'.
      CATCH /aws1/cx_texunsupporteddocex.
        MESSAGE 'The document is not supported.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的 AWS SDK API 參考》中的 StartDocumentAnalysis。

`StartDocumentTextDetection`

以下程式碼範例顯示如何使用 `StartDocumentTextDetection`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

    "Create an ABAP object for the Amazon S3 object."
    DATA(lo_s3object) = NEW /aws1/cl_texs3object( iv_bucket = iv_s3bucket
      iv_name   = iv_s3object ).
    "Create an ABAP object for the document."
    DATA(lo_documentlocation) = NEW /aws1/cl_texdocumentlocation( io_s3object =
lo_s3object ).
    "Start document analysis."
    TRY.
```

```
        oo_result = lo_tex->startdocumenttextdetection( io_documentlocation =
lo_documentlocation ).
        DATA(lv_jobid) = oo_result->get_jobid( ).            "oo_result is returned
for testing purposes."
        MESSAGE 'Document analysis started.' TYPE 'I'.
      CATCH /aws1/cx_texaccessdeniedex.
        MESSAGE 'You do not have permission to perform this action.' TYPE 'E'.
      CATCH /aws1/cx_texbaddocumentex.
        MESSAGE 'Amazon Textract is not able to read the document.' TYPE 'E'.
      CATCH /aws1/cx_texdocumenttoolargeex.
        MESSAGE 'The document is too large.' TYPE 'E'.
      CATCH /aws1/cx_texidempotentprmmis00.
        MESSAGE 'Idempotent parameter mismatch exception.' TYPE 'E'.
      CATCH /aws1/cx_texinternalservererr.
        MESSAGE 'Internal server error.' TYPE 'E'.
      CATCH /aws1/cx_texinvalidkmskeyex.
        MESSAGE 'AWS KMS key is not valid.' TYPE 'E'.
      CATCH /aws1/cx_texinvalidparameterex.
        MESSAGE 'Request has non-valid parameters.' TYPE 'E'.
      CATCH /aws1/cx_texinvalids3objectex.
        MESSAGE 'Amazon S3 object is not valid.' TYPE 'E'.
      CATCH /aws1/cx_texlimitexceededex.
        MESSAGE 'An Amazon Textract service limit was exceeded.' TYPE 'E'.
      CATCH /aws1/cx_texprovthruputexcdex.
        MESSAGE 'Provisioned throughput exceeded limit.' TYPE 'E'.
      CATCH /aws1/cx_texthrottlingex.
        MESSAGE 'The request processing exceeded the limit.' TYPE 'E'.
      CATCH /aws1/cx_texunsupporteddocex.
        MESSAGE 'The document is not supported.' TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的 AWS SDK API 參考》中的 StartDocumentTextDetection。

案例

開始使用文件分析

以下程式碼範例顯示做法：

- 開始非同步分析。

- 取得文件分析。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   "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.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的下列主題。
  - GetDocumentAnalysis

- StartDocumentAnalysis

使用適用於 SAP ABAP 的 SDK 的 Amazon Transcribe 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 Amazon Transcribe 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作

動作

`CreateVocabulary`

以下程式碼範例顯示如何使用 `CreateVocabulary`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    IF it_phrases IS NOT INITIAL.
      oo_result = lo_tnb->createvocabulary(
        iv_vocabularyname = iv_vocabulary_name
        iv_languagecode = iv_language_code
        it_phrases = it_phrases ).
    ELSEIF iv_vocab_file_uri IS NOT INITIAL.
      oo_result = lo_tnb->createvocabulary(
        iv_vocabularyname = iv_vocabulary_name
        iv_languagecode = iv_language_code
```

```
            iv_vocabularyfileuri = iv_vocab_file_uri ).
      ENDIF.
      MESSAGE 'Custom vocabulary created.' TYPE 'I'.
    CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
      MESSAGE lo_bad_request_ex TYPE 'I'.
      RAISE EXCEPTION lo_bad_request_ex.
    CATCH /aws1/cx_tnblimitexceededex INTO DATA(lo_limit_ex).
      MESSAGE lo_limit_ex TYPE 'I'.
      RAISE EXCEPTION lo_limit_ex.
    CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
      MESSAGE lo_internal_ex TYPE 'I'.
      RAISE EXCEPTION lo_internal_ex.
    CATCH /aws1/cx_tnbconflictexception INTO DATA(lo_conflict_ex).
      MESSAGE lo_conflict_ex TYPE 'I'.
      RAISE EXCEPTION lo_conflict_ex.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateVocabulary。

`DeleteTranscriptionJob`

以下程式碼範例顯示如何使用 `DeleteTranscriptionJob`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
      lo_tnb->deletetranscriptionjob( iv_job_name ).
      MESSAGE 'Transcription job deleted.' TYPE 'I'.
    CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
      MESSAGE lo_bad_request_ex TYPE 'I'.
      RAISE EXCEPTION lo_bad_request_ex.
    CATCH /aws1/cx_tnblimitexceededex INTO DATA(lo_limit_ex).
      MESSAGE lo_limit_ex TYPE 'I'.
```

```
      RAISE EXCEPTION lo_limit_ex.
    CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
      MESSAGE lo_internal_ex TYPE 'I'.
      RAISE EXCEPTION lo_internal_ex.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteTranscriptionJob。

`DeleteVocabulary`

以下程式碼範例顯示如何使用 `DeleteVocabulary`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
      lo_tnb->deletevocabulary( iv_vocabulary_name ).
      MESSAGE 'Vocabulary deleted.' TYPE 'I'.
    CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
      MESSAGE lo_bad_request_ex TYPE 'I'.
    CATCH /aws1/cx_tnblimitexceededex INTO DATA(lo_limit_ex).
      MESSAGE lo_limit_ex TYPE 'I'.
      RAISE EXCEPTION lo_limit_ex.
    CATCH /aws1/cx_tnbnotfoundexception INTO DATA(lo_not_found_ex).
      MESSAGE lo_not_found_ex TYPE 'I'.
    CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
      MESSAGE lo_internal_ex TYPE 'I'.
      RAISE EXCEPTION lo_internal_ex.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteVocabulary。

GetTranscriptionJob

以下程式碼範例顯示如何使用 GetTranscriptionJob。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_tnb->gettranscriptionjob( iv_job_name ).
    DATA(lo_job) = oo_result->get_transcriptionjob( ).
    MESSAGE 'Retrieved transcription job details.' TYPE 'I'.
  CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
    MESSAGE lo_bad_request_ex TYPE 'I'.
    RAISE EXCEPTION lo_bad_request_ex.
  CATCH /aws1/cx_tnbnotfoundexception INTO DATA(lo_not_found_ex).
    MESSAGE lo_not_found_ex TYPE 'I'.
    RAISE EXCEPTION lo_not_found_ex.
  CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
    MESSAGE lo_internal_ex TYPE 'I'.
    RAISE EXCEPTION lo_internal_ex.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetTranscriptionJob。

GetVocabulary

以下程式碼範例顯示如何使用 GetVocabulary。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    oo_result = lo_tnb->getvocabulary( iv_vocabulary_name ).
    MESSAGE 'Retrieved vocabulary details.' TYPE 'I'.
  CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
    MESSAGE lo_bad_request_ex TYPE 'I'.
    RAISE EXCEPTION lo_bad_request_ex.
  CATCH /aws1/cx_tnbnotfoundexception INTO DATA(lo_not_found_ex).
    MESSAGE lo_not_found_ex TYPE 'I'.
    RAISE EXCEPTION lo_not_found_ex.
  CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
    MESSAGE lo_internal_ex TYPE 'I'.
    RAISE EXCEPTION lo_internal_ex.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetVocabulary。

`ListTranscriptionJobs`

以下程式碼範例顯示如何使用 `ListTranscriptionJobs`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
TRY.
    IF iv_job_filter IS NOT INITIAL.
```

```
          oo_result = lo_tnb->listtranscriptionjobs( iv_jobnamecontains =
iv_job_filter ).
        ELSE.
          oo_result = lo_tnb->listtranscriptionjobs( ).
        ENDIF.
        MESSAGE 'Retrieved transcription jobs list.' TYPE 'I'.
      CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
        MESSAGE lo_internal_ex TYPE 'I'.
        RAISE EXCEPTION lo_internal_ex.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListTranscriptionJobs。

`ListVocabularies`

以下程式碼範例顯示如何使用 `ListVocabularies`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        IF iv_vocab_filter IS NOT INITIAL.
          oo_result = lo_tnb->listvocabularies( iv_namecontains = iv_vocab_filter ).
        ELSE.
          oo_result = lo_tnb->listvocabularies( ).
        ENDIF.
        MESSAGE 'Retrieved vocabularies list.' TYPE 'I'.
      CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
```

```
        MESSAGE lo_internal_ex TYPE 'I'.
        RAISE EXCEPTION lo_internal_ex.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListVocabularies。

`StartTranscriptionJob`

以下程式碼範例顯示如何使用 `StartTranscriptionJob`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        DATA(lo_media) = NEW /aws1/cl_tnbmedia( iv_mediafileuri = iv_media_uri ).
        DATA(lo_settings) = NEW /aws1/cl_tnbsettings( ).
        IF iv_vocabulary_name IS NOT INITIAL.
          lo_settings = NEW /aws1/cl_tnbsettings( iv_vocabularyname =
iv_vocabulary_name ).
        ENDIF.

        oo_result = lo_tnb->starttranscriptionjob(
          iv_transcriptionjobname = iv_job_name
          io_media = lo_media
          iv_mediaformat = iv_media_format
          iv_languagecode = iv_language_code
          io_settings = lo_settings ).

        MESSAGE 'Transcription job started.' TYPE 'I'.
      CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_tnblimitexceededex INTO DATA(lo_limit_ex).
        MESSAGE lo_limit_ex TYPE 'I'.
```

```
      RAISE EXCEPTION lo_limit_ex.
    CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
      MESSAGE lo_internal_ex TYPE 'I'.
      RAISE EXCEPTION lo_internal_ex.
    CATCH /aws1/cx_tnbconflictexception INTO DATA(lo_conflict_ex).
      MESSAGE lo_conflict_ex TYPE 'I'.
      RAISE EXCEPTION lo_conflict_ex.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 StartTranscriptionJob。

`UpdateVocabulary`

以下程式碼範例顯示如何使用 `UpdateVocabulary`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
      IF it_phrases IS NOT INITIAL.
        oo_result = lo_tnb->updatevocabulary(
          iv_vocabularyname = iv_vocabulary_name
          iv_languagecode = iv_language_code
          it_phrases = it_phrases ).
      ELSEIF iv_vocab_file_uri IS NOT INITIAL.
        oo_result = lo_tnb->updatevocabulary(
          iv_vocabularyname = iv_vocabulary_name
          iv_languagecode = iv_language_code
          iv_vocabularyfileuri = iv_vocab_file_uri ).
      ENDIF.
      MESSAGE 'Vocabulary updated.' TYPE 'I'.
    CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
      MESSAGE lo_bad_request_ex TYPE 'I'.
    CATCH /aws1/cx_tnblimitexceededex INTO DATA(lo_limit_ex).
```

```
      MESSAGE lo_limit_ex TYPE 'I'.
      RAISE EXCEPTION lo_limit_ex.
    CATCH /aws1/cx_tnbnotfoundexception INTO DATA(lo_not_found_ex).
      MESSAGE lo_not_found_ex TYPE 'I'.
    CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
      MESSAGE lo_internal_ex TYPE 'I'.
      RAISE EXCEPTION lo_internal_ex.
    CATCH /aws1/cx_tnbconflictexception INTO DATA(lo_conflict_ex).
      MESSAGE lo_conflict_ex TYPE 'I'.
      RAISE EXCEPTION lo_conflict_ex.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 UpdateVocabulary。

使用適用於 SAP ABAP 的 SDK 的 Amazon Translate 範例

下列程式碼範例示範如何使用適用於 SAP ABAP 的 AWS SDK 搭配 Amazon Translate 來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

案例是向您展示如何呼叫服務中的多個函數或與其他 AWS 服務組合來完成特定任務的程式碼範例。

每個範例均包含完整原始碼的連結，您可在連結中找到如何設定和執行內容中程式碼的相關指示。

主題

- 動作
- 案例

動作

`DescribeTextTranslationJob`

以下程式碼範例顯示如何使用 `DescribeTextTranslationJob`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   "Gets the properties associated with an asynchronous batch translation job."
   "Includes properties such as name, ID, status, source and target languages, and
input/output Amazon Simple Storage Service (Amazon S3) buckets."
   TRY.
       oo_result = lo_xl8->describetexttranslationjob(      "oo_result is returned
for testing purposes."
         iv_jobid        = iv_jobid ).
       MESSAGE 'Job description retrieved.' TYPE 'I'.
     CATCH /aws1/cx_xl8internalserverex.
       MESSAGE 'An internal server error occurred. Retry your request.' TYPE 'E'.
     CATCH /aws1/cx_xl8resourcenotfoundex.
       MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
     CATCH /aws1/cx_xl8toomanyrequestsex.
       MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
   ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的 AWS SDK API 參考》中的 DescribeTextTranslationJob。

`ListTextTranslationJobs`

以下程式碼範例顯示如何使用 `ListTextTranslationJobs`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

   DATA lo_filter TYPE REF TO /aws1/cl_xl8textxlationjobfilt.

   "Create an ABAP object for filtering using jobname."
   lo_filter = NEW #( iv_jobname = iv_jobname ).

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

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 ListTextTranslationJobs。

StartTextTranslationJob

以下程式碼範例顯示如何使用 StartTextTranslationJob。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

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

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

    "Create an ABAP object for the input data config."
    lo_inputdataconfig = NEW #( iv_s3uri = iv_input_data_s3uri
                                iv_contenttype = iv_input_data_contenttype ).

    "Create an ABAP object for the output data config."
    lo_outputdataconfig = NEW #( iv_s3uri = iv_output_data_s3uri ).

    "Create an internal table for target languages."
    lo_targetlanguagecodes = NEW #( iv_value = iv_targetlanguagecode ).
    INSERT lo_targetlanguagecodes  INTO TABLE lt_targetlanguagecodes.

    TRY.
        oo_result = lo_xl8->starttexttranslationjob(      "oo_result is returned for
 testing purposes."
          io_inputdataconfig = lo_inputdataconfig
            io_outputdataconfig = lo_outputdataconfig
            it_targetlanguagecodes = lt_targetlanguagecodes
            iv_dataaccessrolearn = iv_dataaccessrolearn
            iv_jobname = iv_jobname
            iv_sourcelanguagecode = iv_sourcelanguagecode ).
        MESSAGE 'Translation job started.' TYPE 'I'.
      CATCH /aws1/cx_xl8internalserverex.
        MESSAGE 'An internal server error occurred. Retry your request.' TYPE 'E'.
      CATCH /aws1/cx_xl8invparamvalueex.
        MESSAGE 'The value of the parameter is not valid.' TYPE 'E'.
      CATCH /aws1/cx_xl8invalidrequestex.
        MESSAGE 'The request that you made is not valid.' TYPE 'E'.
      CATCH /aws1/cx_xl8resourcenotfoundex.
        MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
      CATCH /aws1/cx_xl8toomanyrequestsex.
        MESSAGE 'You have made too many requests within a short period of time.'
 TYPE 'E'.
      CATCH /aws1/cx_xl8unsuppedlanguage00.
        MESSAGE 'Amazon Translate does not support translation from the language of
 the source text into the requested target language.' TYPE 'E'.
```

```
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的 AWS SDK API 參考》中的 StartTextTranslationJob。

StopTextTranslationJob

以下程式碼範例顯示如何使用 StopTextTranslationJob。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

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

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的 AWS SDK API 參考》中的 StopTextTranslationJob。

`TranslateText`

以下程式碼範例顯示如何使用 `TranslateText`。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   "Translates input text from the source language to the target language."
   TRY.
       oo_result = lo_xl8->translatetext(      "oo_result is returned for testing
purposes."
         iv_text        = iv_text
           iv_sourcelanguagecode = iv_sourcelanguagecode
           iv_targetlanguagecode = iv_targetlanguagecode ).
       MESSAGE 'Translation completed.' TYPE 'I'.
     CATCH /aws1/cx_xl8detectedlanguage00.
       MESSAGE 'The confidence that Amazon Comprehend accurately detected the
source language is low.' TYPE 'E'.
     CATCH /aws1/cx_xl8internalserverex.
       MESSAGE 'An internal server error occurred.' TYPE 'E'.
     CATCH /aws1/cx_xl8invalidrequestex.
       MESSAGE 'The request that you made is not valid.' TYPE 'E'.
     CATCH /aws1/cx_xl8resourcenotfoundex.
       MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
     CATCH /aws1/cx_xl8serviceunavailex.
       MESSAGE 'The Amazon Translate service is temporarily unavailable.' TYPE 'E'.
     CATCH /aws1/cx_xl8textsizelmtexcdex.
       MESSAGE 'The size of the text you submitted exceeds the size limit. ' TYPE
'E'.
     CATCH /aws1/cx_xl8toomanyrequestsex.
       MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
     CATCH /aws1/cx_xl8unsuppedlanguage00.
       MESSAGE 'Amazon Translate does not support translation from the language of
the source text into the requested target language. ' TYPE 'E'.
   ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的 TranslateText。

案例

開始執行翻譯任務

以下程式碼範例顯示做法：

- 啟動非同步批次翻譯任務。
- 等候非同步任務完成。
- 描述非同步任務。

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

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

    "Create an ABAP object for the input data config."
    lo_inputdataconfig = NEW #( iv_s3uri = iv_input_data_s3uri
                                iv_contenttype = iv_input_data_contenttype ).

    "Create an ABAP object for the output data config."
    lo_outputdataconfig = NEW #( iv_s3uri = iv_output_data_s3uri ).

    "Create an internal table for target languages."
    lo_targetlanguagecodes = NEW #( iv_value = iv_targetlanguagecode ).
    INSERT lo_targetlanguagecodes  INTO TABLE lt_targetlanguagecodes.

    TRY.
        DATA(lo_translationjob_result) = lo_xl8->starttexttranslationjob(
```

```
          io_inputdataconfig = lo_inputdataconfig
            io_outputdataconfig = lo_outputdataconfig
            it_targetlanguagecodes = lt_targetlanguagecodes
            iv_dataaccessrolearn = iv_dataaccessrolearn
            iv_jobname = iv_jobname
            iv_sourcelanguagecode = iv_sourcelanguagecode ).
        MESSAGE 'Translation job started.' TYPE 'I'.
      CATCH /aws1/cx_xl8internalserverex.
        MESSAGE 'An internal server error occurred. Retry your request.' TYPE 'E'.
      CATCH /aws1/cx_xl8invparamvalueex.
        MESSAGE 'The value of the parameter is not valid.' TYPE 'E'.
      CATCH /aws1/cx_xl8invalidrequestex.
        MESSAGE 'The request that you made is not valid.' TYPE 'E'.
      CATCH /aws1/cx_xl8resourcenotfoundex.
        MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
      CATCH /aws1/cx_xl8toomanyrequestsex.
        MESSAGE 'You have made too many requests within a short period of time. '
 TYPE 'E'.
      CATCH /aws1/cx_xl8unsuppedlanguage00.
        MESSAGE 'Amazon Translate does not support translation from the language of
 the source text into the requested target language.' TYPE 'E'.
    ENDTRY.

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

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

    TRY.
        oo_result = lo_xl8->describetexttranslationjob(      "oo_result is returned
 for testing purposes."
          iv_jobid        = lv_jobid ).
        MESSAGE 'Job description retrieved.' TYPE 'I'.
```

```
      CATCH /aws1/cx_xl8internalserverex.
        MESSAGE 'An internal server error occurred. Retry your request.' TYPE 'E'.
      CATCH /aws1/cx_xl8resourcenotfoundex.
        MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
      CATCH /aws1/cx_xl8toomanyrequestsex.
        MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的AWS SDK API 參考》中的下列主題。
  - DescribeTextTranslationJob
  - StartTextTranslationJob

中的安全性 適用於 SAP ABAP 的 AWS SDK

的雲端安全性 AWS 是最高優先順序。身為 AWS 客戶，您可以受益於資料中心和網路架構，這些架構是為了符合最安全敏感組織的需求而建置。

安全性是 AWS 與您之間的共同責任。共同責任模式將其描述為雲端的安全性，和雲端中的安全性：

- 雲端的安全性 – AWS 負責保護在 AWS 服務 中執行的基礎設施 AWS 雲端。 AWS 也為您提供可安全使用的服務。作為AWS 合規計畫的一部分，第三方稽核人員會定期測試和驗證我們安全的有效性。若要了解適用的合規計劃 適用於 SAP ABAP 的 AWS SDK，請參閱AWS 服務 合規計劃範圍。
- 雲端的安全性 – 您的責任取決於您使用 AWS 服務 的 。您也必須對其他因素負責，包括資料的機密性、您公司的要求和適用法律和法規。

本節涵蓋下列主題。

主題

- 上的 SAP 系統身分驗證 AWS
- IAM 安全性的最佳實務
- SAP 授權
- 安全操作
- 搭配 SSF 加密使用私密存取金鑰身分驗證
- 搭配 IAM Roles Anywhere 使用憑證
- 使用來源描述檔進行跨帳戶存取
- 使用 SAP 登入資料存放區

上的 SAP 系統身分驗證 AWS

在 SAP 系統可以 AWS 代表 SAP 使用者呼叫 之前，SAP 系統必須向 進行自我驗證 AWS。 適用於 SAP ABAP 的 AWS SDK 支援以下在 的 SDK 設定檔設定中選取的三種基本身分驗證方法IMG。

AWS 適用於 SAP ABAP 的 SDK - BTP 版本只能使用 SAP Credential Store 的 the section called “私密存取金鑰身分驗證”方法進行身分驗證。

對於跨帳戶存取案例，適用於 SAP ABAP 的 SDK 也支援來源描述檔，可使用任何基本身分驗證方法跨帳戶鏈結多個 IAM 角色假設。如需詳細資訊，請參閱the section called “跨帳戶存取的來源描述檔身分驗證”。

主題

- Amazon EC2 執行個體中繼資料身分驗證
- 私密存取金鑰身分驗證
- 使用 IAM Roles Anywhere 的憑證型身分驗證
- 跨帳戶存取的來源描述檔身分驗證
- 下一步驟

Amazon EC2 執行個體中繼資料身分驗證

在 Amazon EC2 上執行的 SAP 系統可以從 Amazon EC2 執行個體中繼資料取得短期、自動輪換的登入資料。如需詳細資訊，請參閱使用 Amazon EC2 執行個體中繼資料的登入資料。

我們強烈建議在使用適用於 SAP ABAP 的 SDK 時使用此身分驗證方法。若要啟用，Basis 管理員必須啟用傳出 HTTP 通訊。不需要進一步的基礎組態。

> **Note**
>
> 此身分驗證方法僅適用於 SAP 系統在 Amazon EC2 上執行的情況。內部部署或其他雲端環境中託管的 SAP 系統無法使用此方法進行身分驗證。

私密存取金鑰身分驗證

透過此方法，您可以使用存取金鑰 ID 和私密存取金鑰來驗證 SAP 系統 AWS。SAP 系統 AWS 使用 IAM 使用者登入 。如需詳細資訊，請參閱管理 IAM 使用者的存取金鑰。

基礎管理員會從 AWS IAM 管理員收到存取金鑰 ID 和私密存取金鑰。您的 SAP 系統必須設定為存放存取金鑰 ID 和私密存取金鑰。

- 安全、存放和轉送 (SSF)
  - 使用 SSF 功能來驗證適用於 SAP ABAP 的 AWS SDK。如需詳細資訊，請參閱數位簽章和加密。
  - 您也可以使用SSF02 報告測試 SSF 的 envelope和 develope功能。如需詳細資訊，請參閱測試 SSF 安裝。

- `/AWS1/IMG` 交易中說明設定 SSF for SDK for SAP ABAP 的步驟。前往技術先決條件，然後選取現場部署系統的其他設定。如需詳細組態步驟，請參閱搭配 SSF 加密使用私密存取金鑰身分驗證。
- SAP 登入資料存放區
  - 使用 SAP Credential Store 驗證適用於 SAP ABAP 的 AWS SDK - BTP 版本。如需詳細資訊，請參閱什麼是 SAP 登入資料存放區？
  - 如需組態步驟，請參閱使用 SAP 登入資料存放區。

使用 IAM Roles Anywhere 的憑證型身分驗證

憑證授權機構 (CA) 發行的 X.509 憑證可用於向 AWS Identity and Access Management Roles Anywhere 進行身分驗證。憑證必須在 中設定STRUST。CA 必須向 IAM Roles Anywhere 註冊為信任錨點，而且必須建立設定檔來指定 IAM Roles Anywhere 將擔任的角色和政策。如需詳細資訊，請參閱在 AWS Identity and Access Management Roles Anywhere 中建立信任錨點和設定檔。

如需如何搭配適用於 SAP ABAP 的 SDK 使用 IAM Roles Anywhere 的詳細步驟，請參閱搭配 IAM Roles Anywhere 使用憑證。

> **Note**
>
> 憑證撤銷僅透過使用匯入的憑證撤銷清單來支援。如需詳細資訊，請參閱撤銷。

跨帳戶存取的來源描述檔身分驗證

來源描述檔是一項進階功能，可讓您跨 AWS 帳戶鏈結多個 IAM 角色假設。使用此方法，一個設定檔會擔任角色，然後擔任另一個角色，以此類推，類似於 AWS CLI 中的 `source_profile` 參數。

來源描述檔適用於三種基本身分驗證方法（執行個體中繼資料、私密存取金鑰或憑證類型）中的任一種。鏈結中的第一個設定檔必須使用其中一個基本方法，而鏈結中的後續設定檔必須使用上一個設定檔的登入資料來擔任下一個角色。

這適用於您需要周遊多個 AWS 帳戶以到達目標資源的跨帳戶存取案例。如需詳細組態步驟，請參閱使用來源設定檔進行跨帳戶存取。

下一步驟

在 中驗證 SAP 系統之後 AWS，適用於 SAP ABAP 的 SDK 會自動執行`sts:assumeRole` ，以擔任 SAP 使用者業務職能的適當 IAM 角色。

IAM 安全性的最佳實務

IAM 管理員將負責以下三個關鍵領域。

- 確保 SAP 系統可以使用 Amazon EC2 中繼資料或私密金鑰憑證進行自我驗證。
- 確保 SAP 系統具有使用 提升自我所需的許可`sts:assumeRole`。
- 對於每個邏輯 IAM 角色，為具有執行業務功能所需許可的 SAP 使用者建立 IAM 角色（例如，Amazon S3、DynamoDB 或其他 服務的必要許可）。這些是 SAP 使用者將擔任的角色。

如需詳細資訊，請參閱 SAP Lens： AWS Well-Architected Framework 中的安全章節。

主題

- Amazon EC2 執行個體描述檔的最佳實務
- SAP 使用者的 IAM 角色
- 來源描述檔安全性考量

Amazon EC2 執行個體描述檔的最佳實務

SAP 系統執行所在的 Amazon EC2 執行個體，會根據其執行個體描述檔擁有一組授權。一般而言，執行個體描述檔只需要具有呼叫 的許可`sts:assumeRole`，以允許 SAP 系統視需要擔任業務特定的 IAM 角色。這種提升到其他角色可確保 ABAP 程式可以擔任角色，為使用者提供執行其任務所需的最低權限。例如，執行個體描述檔可能包含下列陳述式。

JSON

```
{
 "Version":"2012-10-17",
 "Statement": [
   {
     "Sid": "VisualEditor0",
```

```
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource":[
         "arn:aws:iam::012345678912:role/finance-cfo",
         "arn:aws:iam::012345678912:role/finance-auditor",
         "arn:aws:iam::012345678912:role/finance-reporting"
       ]
    }
  ]
}
```

上述範例允許 SAP 系統擔任 CFO、AUDITOR 或 REPORTING user 的 IAM 角色。 AWS SDK 會根據使用者在 SAP 中的 PFCG 角色，為使用者選擇正確的 IAM 角色。

Amazon EC2 執行個體描述檔也可以用於其他 函數。

- AWS SAP HANA 的後端代理程式
- SAP on AWS 高可用性與浮水印 IP 地址路由

這些解決方案也可能需要備份或容錯移轉特定角色的`sts:assumeRole`許可，或者可能需要直接指派給執行個體描述檔的許可。

SAP 使用者的 IAM 角色

ABAP 程式需要許可才能執行使用者的工作：讀取 DynamoDB 資料表、在 Amazon S3 中的 PDF 物件上叫用 Amazon Textract、執行 AWS Lambda 函數。相同的安全模型用於 AWS SDKs。您可以使用用於另一個 AWS SDK 的現有 IAM 角色。

SAP 業務分析師將針對所需的每個邏輯角色，向 IAM 管理員詢問 arn：aws： 的 IAM 角色。例如，在金融案例中，業務分析師可以定義下列邏輯 IAM 角色。

- `CFO`
- `AUDITOR`
- `REPORTING`

IAM 管理員會為每個邏輯 IAM 角色定義 IAM 角色。

`CFO`

- arn:aws:iam::*0123456789*:role/finance-cfo
- 讀取和寫入許可至 Amazon S3 儲存貯體
- 讀取和寫入許可至 DynamoDB 資料庫

AUDITOR

- arn:aws:iam::*0123456789*:role/finance-auditor
- Amazon S3 儲存貯體的讀取許可
- DynamoDB 資料庫的讀取許可

REPORTING

- arn:aws:iam::*0123456789*:role/finance-reporting
- DynamoDB 資料庫的讀取許可
- 沒有 Amazon S3 儲存貯體的許可

業務分析師會將 IAM 角色輸入映射表中，以將邏輯 IAM 角色映射至實體 IAM 角色。

SAP 使用者的 IAM 角色需要允許信任委託人sts:assumeRole的動作。信任的委託人可能會根據 SAP 系統在 上的身分驗證方式而有所不同 AWS。如需詳細資訊，請參閱指定委託人。

以下是一些最常見的 SAP 案例範例。

- 在已指派執行個體描述檔的 Amazon EC2 上執行的 SAP 系統 – 在此，Amazon EC2 執行個體描述檔會連接至 IAM 角色。

  JSON

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

```
        }
    ]
}
```

- 在沒有執行個體描述檔的 Amazon EC2 上執行的 SAP 系統 – 在此，Amazon EC2 會擔任 SAP 使用者的角色。

  JSON

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

- 內部部署執行的 SAP 系統 – 內部部署執行的 SAP 系統只能使用私密存取金鑰進行驗證。如需詳細資訊，請參閱 上的 SAP 系統身分驗證 AWS。

  在此，SAP 使用者擔任的任何 IAM 角色都必須具有信任 SAP 使用者的信任關係。

  JSON

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

```
}
```

來源描述檔安全性考量

使用來源描述檔時：

IAM 角色管理

關鍵：必須嚴格管理來源設定檔鏈中的 IAM 角色，以防止未經授權的存取和權限提升：

- 套用最低權限原則 - 僅授予每個角色特定用途所需的最低許可
- 定期稽核角色許可 - 每季或在需求變更時檢閱和更新角色政策
- 監控角色用量 - 用於追蹤 AssumeRole API 呼叫並識別異常模式
- 限制信任關係 - 將哪些委託人可以擔任每個角色僅限於絕對需要存取的委託人
- 在信任政策中使用條件 - 在適當情況下新增條件，例如來源 IP、MFA 要求或時間型限制
- 文件角色用途 - 維護每個角色預期使用案例和必要許可的明確文件

授權和存取控制

- 確保鏈結中的所有中繼設定檔都已設定適當的信任政策
- 使用者必須擁有鏈結中所有設定檔/AWS1/SESS的授權，包括中繼設定檔
- 每個 IAM 角色都必須明確信任鏈結中的先前角色

技術保護措施

- SDK 會強制執行 5 個設定檔的最大鏈結深度，以防止 STS API 呼叫過多
- 系統會自動偵測並防止循環參考
- 基本設定檔身分驗證方法經過驗證，以確保其使用標準方法 (INST、SSF 或 RLA)

如需設定來源描述檔的詳細資訊，請參閱使用來源描述檔進行跨帳戶存取。

SAP 授權

設定 SDK 所需的授權取決於 SDK 版本。

主題

- 組態的授權
- 最終使用者的 SAP 授權

組態的授權

如需詳細資訊，請參閱下列索引標籤。

SDK for SAP ABAP

設定適用於 SAP ABAP 的 SDK 需要下列授權。

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

    從下列授權群組中選擇。

    - /AWS1/CFG- 適用於 SAP ABAP 的 AWS SDK v1 - 組態
    - /AWS1/MOD- 適用於 SAP ABAP 的 AWS SDK v1 - 執行時間
    - /AWS1/PFL- 適用於 SAP ABAP 的 AWS SDK v1 - SDK 設定檔
    - /AWS1/RES- 適用於 SAP ABAP 的 AWS SDK v1 - 邏輯資源
    - /AWS1/TRC- 適用於 SAP ABAP 的 AWS SDK v1 - 追蹤

SDK for SAP ABAP - BTP edition

使用下列步驟，以允許適用於 SAP ABAP 的 SDK - BTP 版本存取組態。

1. 使用業務角色範本建立新的SAP_BR_BPC_EXPERT業務角色。此範本可讓您存取 Cutsom Business Configuration 應用程式。
2. 在一般角色詳細資訊下，前往存取類別，然後選擇不受限制的讀取、寫入、值說明。
3. 前往商業目錄索引標籤，並指派/AWS1/RTBTP_BCAT商業目錄以提供 SDK 組態的存取權。
4. 前往商業使用者索引標籤，並指派商業使用者來授予 SDK 組態的存取權。

最終使用者的 SAP 授權

先決條件：定義 SDK 設定檔

在 SAP 安全管理員可以定義其角色之前，商業分析師將在交易中定義適用於 SAP ABAP /AWS1/IMG的 AWS SDK 或適用於適用於 SAP ABAP 的 SDK 的自訂商業組態應用程式 - BTP 版本。一般而言，軟體開發套件設定檔將根據其業務職能命名：ZFINANCE、ZBILLING、ZMFG、ZPAYROLL 等。對於每個 SDK 描述檔，商業分析師將定義簡短名稱的邏輯 IAM 角色，例如 CFO、AUDITOR、REPORTING。這些將由 IAM 安全管理員映射至實際的 IAM 角色。

定義 PFCG 或業務角色

> **Note**
>
> PFCG 角色在 SAP BTP、ABAP 環境中稱為業務角色。

SAP 安全管理員接著會新增授權物件/AWS1/SESS，以授予對 SDK 設定檔的存取權。

驗證物件 /AWS1/SESS

- 欄位 /AWS1/PROF = ZFINANCE

使用者也應對應到每個 SDK 設定檔的邏輯 IAM 角色，視其任務職能而定。例如，具有報告存取權的財務稽核人員可能獲得授權，擔任名為 的邏輯 IAM 角色AUDITOR。

驗證物件 /AWS1/LROL

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

同時，具有讀取/寫入授權的 CFO 可能具有授權其 邏輯角色的 PFCG 角色CFO。

驗證物件 /AWS1/LROL

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

一般而言，每個 SDK 描述檔只能授權使用者一個邏輯 IAM 角色。如果使用者獲得多個 IAM 角色的授權（例如，如果 CFO 同時獲得 CFO和邏輯 AUDITOR IAM 角色的授權），則 AWS SDK 會透過確保較高優先順序（較低的序號）角色生效來中斷關聯。

如同所有安全案例，使用者應獲得執行其任務職能的最低權限。管理 PFCG 角色的簡單策略是根據其授權的 SDK 描述檔和邏輯角色命名單一 PFCG 角色。例如，角色會Z_AWS_PFL_ZFINANCE_CFO授予對設定檔 ZFINANCE和邏輯 IAM 角色 的存取權CFO。然後，這些單一角色可以指派給定義任務函數的複合角色。每個公司都有自己的角色管理策略，建議您定義適合您的 PFCG 策略。

安全操作

靜態資料加密

AWS 私密存取金鑰用於驗證 SDK。它們由 SAP 使用 SSF 或登入資料存放區功能加密。

傳輸中資料的加密

對 的所有呼叫 AWS 服務 都會使用 HTTPS 加密。SAP ICM 管理 HTTPS connection. AWS certificates 必須在 STRUST 中受信任。

API 用量

當 ABAP 使用者使用 擔任角色時sts:assumeRole，工作階段名稱會命名為 USERID-SID-MANDT，其中：

- USERID 是來自 SY-UNAME變數的 ABAP 使用者。
- SID 是來自 SY-SYSID變數的 ABAP 系統 ID。
- MANDT 是來自 SY-MANDT變數的 ABAP 用戶端。

工作階段名稱會在 CloudTrail 中顯示為使用者名稱。這可確保來自 ABAP 使用者的 API 呼叫可以追蹤回啟動呼叫的系統、用戶端和使用者。如需詳細資訊，請參閱什麼是 AWS CloudTrail？

搭配 SSF 加密使用私密存取金鑰身分驗證

內部部署 SAP 系統（或其他雲端中執行的系統）可以透過使用私密存取金鑰身分驗證 AWS 在 上進行身分驗證 AWS Identity and Access Management。SAP 的安全存放區和轉送機制 (SSF) 用於加

密和安全存放 IAM 使用者的 AWS 登入資料（存取金鑰 ID 和私密存取金鑰）。SAP 系統 AWS 使用 IAM 使用者登入 ，請參閱管理 IAM 使用者的存取金鑰以取得資訊。

先決條件

開始組態之前，必須符合下列先決條件：

- SAP 使用者的 IAM 角色必須由 IAM 管理員建立。角色必須具有呼叫所需 的許可 AWS 服務。如需詳細資訊，請參閱 IAM 安全的最佳實務。
- 建立執行/AWS1/IMG交易的授權。如需詳細資訊，請參閱組態的授權。

程序

請遵循下列指示來設定 SSF 加密憑證儲存：

步驟

- 步驟 1 – 定義登入資料儲存的 SSF 應用程式
- 步驟 2 – 設定 SSF 應用程式的加密參數
- 步驟 3 – 建立 SSF 應用程式的 PSE
- 步驟 4 – 將 SSF 應用程式指派給 適用於 SAP ABAP 的 AWS SDK
- 步驟 5 – 設定 SDK 描述檔以使用 SSF 加密的登入資料

步驟 1 – 定義登入資料儲存的 SSF 應用程式

1. 執行交易程式碼SE16以定義 SSF 應用程式。
2. 輸入SSFAPPLIC資料表名稱，然後選取新增項目。
3. 輸入下列詳細資訊：
   - APPLIC：ZAWS1(SSF 應用程式的名稱）。
   - DESCRIPT：SSF Encryption for the ### SAP ABAP # AWS SDK（描述）。
   - 選擇其餘欄位Selected(X)的選項。
4. 選取儲存。

步驟 2 – 設定 SSF 應用程式的加密參數

1. 執行交易程式碼/n/AWS1/IMG以啟動 的實作指南 (IMG) 適用於 SAP ABAP 的 AWS SDK。

2. 展開現場部署系統的 IMG 節點適用於 SAP ABAP 的 AWS SDK 設定 > 技術先決條件 > 其他設定。
3. 執行設定 SSF 參數 IMG 活動。
4. 選取新增項目，然後選擇在上一個步驟中建立的 SSF 應用程式。選取儲存。
5. 將雜湊演算法修改為 SHA256（或更新版本），並將加密演算法修改為 AES256-CBC。將其他設定保留為預設值，然後選取儲存。

這些加密設定將用於安全加密 AWS 登入資料。

步驟 3 – 建立 SSF 應用程式的 PSE

1. 執行/n/AWS1/IMG交易，然後選取適用於 SAP ABAP 的 AWS SDK 設定 > 技術先決條件 > 現場部署系統的其他設定。
2. 執行 IMG 活動 Create PSE for SSF Application，這會引導您前往 STRUST transaction。選擇 Edit (編輯)。
3. 以滑鼠右鍵選取在 中建立的 SSF 應用程式the section called “步驟 1”，然後選擇建立。保留所有其他預設設定，然後選取繼續。請確定您選擇的演算法RSA不是 DSA。

步驟 4 – 將 SSF 應用程式指派給 適用於 SAP ABAP 的 AWS SDK

1. 執行/n/AWS1/IMG交易，然後選取適用於 SAP ABAP 的 AWS SDK 設定 > 技術先決條件 > 現場部署系統的其他設定。
2. 執行 IMG 活動 將 SSF 應用程式指派給 適用於 SAP ABAP 的 AWS SDK。
3. 選取新增項目，然後輸入在 中建立的 SSF 應用程式the section called “步驟 1”。選取儲存。

步驟 5 – 設定 SDK 描述檔以使用 SSF 加密的登入資料

1. 執行/n/AWS1/IMG交易，然後選取適用於 SAP ABAP 的 AWS SDK 設定 > 應用程式組態。
2. 執行 IMG 活動 SDK 設定檔。
3. 選取新增項目。輸入設定檔名稱和描述。選取儲存。
4. 反白顯示您建立的項目，然後按一下身分驗證和設定樹狀目錄分支。
5. 選取新增項目，然後輸入下列詳細資訊：
   - SID：SAP 系統的系統 ID。
   - 用戶端：SAP 系統的用戶端。
   - 案例 ID：選取您的 Basis 管理員建立的DEFAULT案例。

- 您要呼叫的區域AWS ： AWS 區域。
- 身分驗證方法：Credentials from SSF Storage從下拉式清單中選取 ，然後選取儲存。選取設定登入資料，然後輸入 IAM 使用者的存取金鑰 ID 和私密存取金鑰。
- 停用 IAM 角色：將此保留為預設，即未勾選。
- 選取儲存。

6. 按一下 IAM 角色映射樹狀目錄分支。選取新增項目。輸入序號、邏輯 IAM 角色的名稱，以及 IAM 管理員提供的 AWS IAM 角色 ARN。選取儲存。

如需詳細資訊，請參閱應用程式組態。

搭配 IAM Roles Anywhere 使用憑證

SAP 系統可以在 上使用以憑證為基礎的身分驗證搭配 AWS Identity and Access Management Roles Anywhere AWS 進行身分驗證。您必須在 中設定憑證STRUST，並在 中設定 SDK 設定檔/AWS1/IMG。

先決條件

在開始設定認證之前，必須符合下列先決條件。

- 您的憑證授權單位 (CA) 發行的 X.509 憑證必須符合下列要求。
  - 簽署憑證必須是 v3 憑證。
  - 鏈不能超過 5 個憑證。
  - 憑證必須支援 RSA 或 ECDSA 演算法。
- 向 IAM Roles Anywhere 註冊您的 CA 作為信任錨點，並建立設定檔以指定 IAM Roles Anywhere 的角色/政策。如需詳細資訊，請參閱在 AWS Identity and Access Management Roles Anywhere 中建立信任錨點和設定檔。
- SAP 使用者的 IAM 角色必須由 IAM 管理員建立。角色必須具有呼叫所需 的許可 AWS 服務。如需詳細資訊，請參閱 IAM 安全的最佳實務。
- 建立執行/AWS1/IMG交易的授權。如需詳細資訊，請參閱組態的授權。

程序

請遵循這些指示來設定憑證型身分驗證。

步驟

- 步驟 1 – 使用 SAP 的安全存放區和轉送 (SSF) 定義 SSF 應用程式
- 步驟 2 – 設定 SSF 參數
- 步驟 3 – 建立 PSE 和憑證請求
- 步驟 4 – 將憑證回應匯入相關 PSE
- 步驟 5 – 設定 SDK 描述檔以使用 IAM Roles Anywhere

步驟 1 – 使用 SAP 的安全存放區和轉送 (SSF) 定義 SSF 應用程式

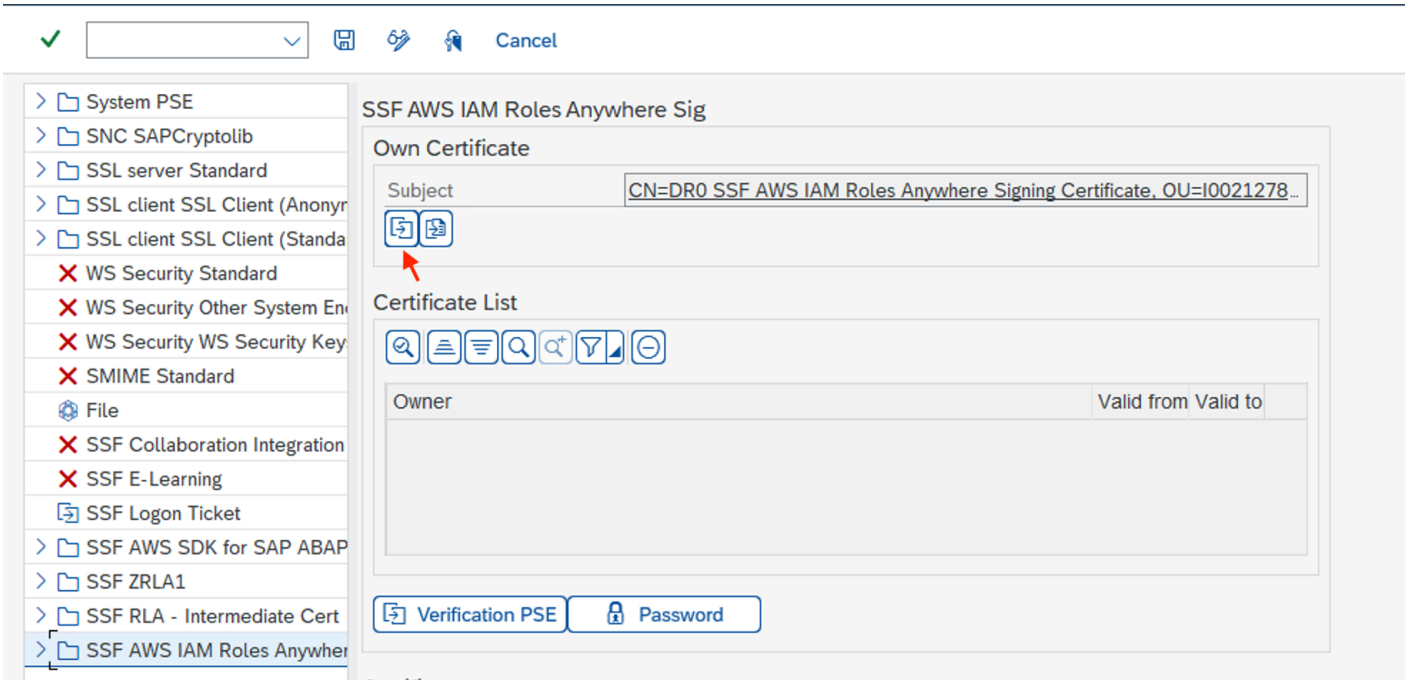
1. 執行交易程式碼SE16來定義 SSF 應用程式。
2. 輸入SSFAPPLIC資料表名稱，然後選取新增項目。
3. 在 APPLIC 檔案中輸入 SSF 應用程式的名稱、在 DESCRIPT 檔案中輸入描述，並為其餘欄位選取Selected (X)選項。

步驟 2 – 設定 SSF 參數

1. 執行 /n/AWS1/IMG以啟動 適用於 SAP ABAP 的 AWS SDK 實作指南 (IMG)。
2. 選取適用於 SAP ABAP 的 AWS SDK 設定 > 技術先決條件 > 現場部署系統的其他設定。
3. 執行設定 SSF 參數 IMG 活動。
4. 選取新增項目，然後選擇在上一個步驟中建立的 SSF 應用程式。選取儲存。
5. 將雜湊演算法修改為 SHA256，並將加密演算法修改為 AES256-CBC。將其他設定保留為預設值，然後選取儲存。

步驟 3 – 建立 PSE 和憑證請求

1. 執行/n/AWS1/IMG交易，然後選取適用於 SAP ABAP 的 AWS SDK 設定 > 技術先決條件 > 現場部署系統的其他設定。
2. 執行 Create PSE for SSF Application IMG 活動。
3. 選取STRUST交易的編輯。
4. 以滑鼠右鍵選取在 中建立的 SSF 應用程式the section called “步驟 1”，然後選擇建立。保留所有其他預設設定，然後選取繼續。
5. 選取建立憑證請求。請參閱下圖。保留預設選項，然後選取繼續。複製或匯出產生的憑證請求，並將其提供給您的 CA。您的 CA 驗證請求，並以簽章的公有金鑰憑證回應。



簽署程序會根據您的 CA 及其使用的技術而有所不同。如需範例，請參閱使用私有憑證授權機構發行私有終端實體憑證。 AWS

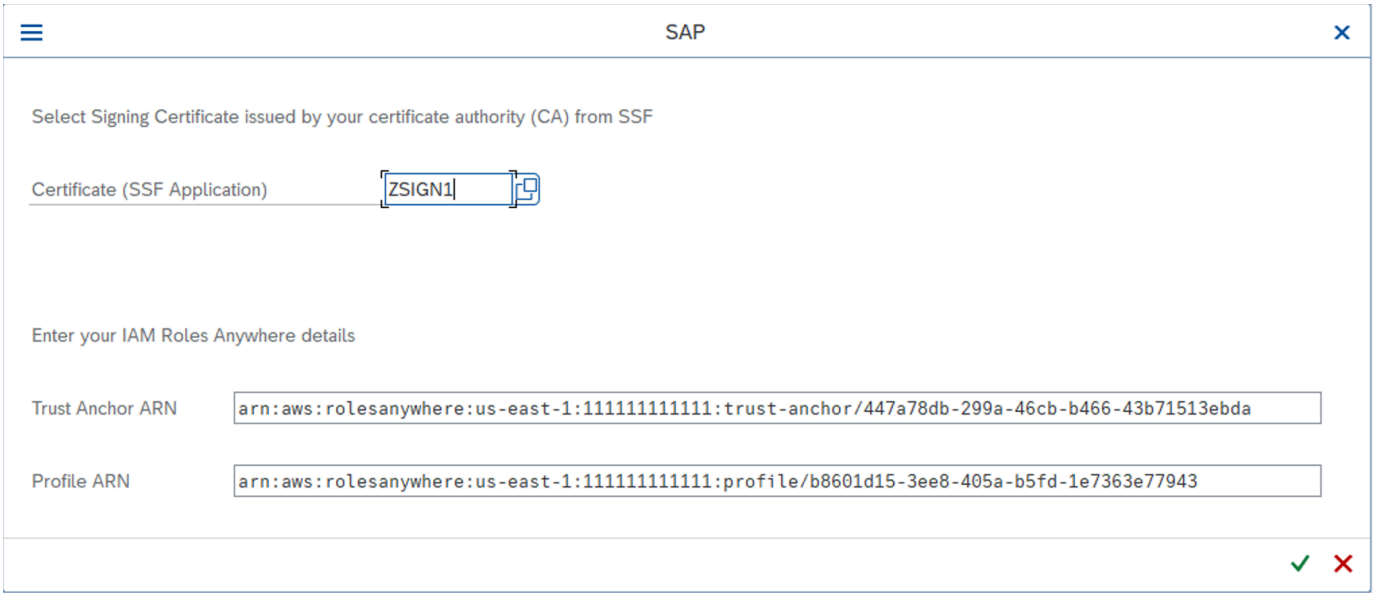
步驟 4 – 將憑證回應匯入相關 PSE

1. 執行/n/AWS1/IMG交易，然後選取適用於 SAP ABAP 的 AWS SDK 設定 > 技術先決條件 > 現場部署系統的其他設定。
2. 執行 `Create PSE for SSF Application` IMG 活動。
3. 選取STRUST交易的編輯。
4. 選擇 SSF 應用程式，然後選取位於主旨下方 PSE 區段中的匯入憑證回應。將憑證回應複製並貼到文字方塊中，或從檔案系統匯入檔案。選取繼續 > 儲存。
5. 選取主旨兩次，即可檢視憑證詳細資訊。資訊會顯示在憑證區段中。

步驟 5 – 設定 SDK 描述檔以使用 IAM Roles Anywhere

1. 執行/n/AWS1/IMG交易，然後選取適用於 SAP ABAP 的 AWS SDK 設定 > 應用程式組態。
2. 建立新的 SDK 設定檔，並將其命名為 。
3. 選擇 IAM Roles Anywhere 作為身分驗證方法。
   - 在左側窗格中，選取身分驗證和設定。

- 建立新的項目，並輸入 SAP 系統的資訊，以及 AWS 區域。
- 針對身分驗證方法選取 IAM Roles Anywhere，然後選取儲存。
- 選取輸入詳細資訊，然後在快顯視窗中選擇在 中建立的 SSF 應用程式the section called “步驟 1”。輸入在 中建立的信任錨點 ARN 和設定檔 ARNthe section called “先決條件”。請參閱下圖。選取繼續。



4. 在左側窗格中，選取 IAM 角色映射。輸入名稱，並提供 IAM 管理員提供的 IAM 角色 ARN。

如需詳細資訊，請參閱應用程式組態。

使用來源描述檔進行跨帳戶存取

來源描述檔可讓 SAP 系統透過鏈結 IAM 角色假設，跨多個帳戶存取 AWS 資源。一個設定檔會擔任角色，然後擔任另一個角色，以此類推，類似於 CLI AWS 中的 `source_profile` 參數。這適用於您需要周遊多個 AWS 帳戶以到達目標資源的跨帳戶存取案例。

範例：您的 SAP 系統在帳戶 A ((111111111111) 中執行，且需要存取帳戶 C ((333333333333) 中的 Amazon S3 儲存貯體。您可以設定三個設定檔：

1. DEV_BASE 從 Amazon EC2 執行個體中繼資料取得基本憑證，並擔任帳戶 A 中的角色 P
2. SHARED_SERVICES 使用DEV_BASE登入資料擔任帳戶 B 中的角色 Q (222222222222)
3. PROD_S3_ACCESS 使用SHARED_SERVICES登入資料來擔任帳戶 C 中的角色 R

當您的應用程式使用 時PROD_S3_ACCESS，開發套件會自動執行鏈結：從執行個體中繼資料取得登入資料 → 擔任角色 P → 擔任角色 Q → 擔任角色 R。

先決條件

在設定來源描述檔之前，必須符合下列先決條件：

- 鏈中每個步驟的 IAM 角色必須由 IAM 管理員建立。每個角色必須具有：
  - 呼叫所需 的許可 AWS 服務
  - 將信任關係設定為允許鏈中的上一個角色擔任它

  如需詳細資訊，請參閱 IAM 安全的最佳實務。
- 建立執行/AWS1/IMG交易的授權。如需詳細資訊，請參閱組態的授權。
- 使用者必須擁有鏈結中所有設定檔/AWS1/SESS的授權，包括中繼設定檔。

程序

請依照這些指示來設定來源描述檔。

步驟

- 步驟 1 – 設定基本設定檔
- 步驟 2 – 設定鏈結的設定檔

步驟 1 – 設定基本設定檔

基本設定檔是鏈中的第一個設定檔，必須使用標準身分驗證方法。

1. 執行/n/AWS1/IMG交易以啟動 適用於 SAP ABAP 的 AWS SDK 實作指南 (IMG)。
2. 選取AWS 適用於 SAP ABAP 的 SDK 設定 > 應用程式組態 > SDK 設定檔。
3. 選取新增項目並輸入設定檔名稱和描述，以建立新的設定檔做為基本設定檔。選取儲存。

> **Note**
>
> 如果您使用的是已使用標準身分驗證方法 (INST、SSF 或 RLA) 設定的現有設定檔，您可以略過本節中的其餘步驟，直接前往 the section called “步驟 2”。

4. 選取您建立的設定檔，然後選取身分驗證和設定 > 新項目，然後輸入下列詳細資訊：

- SID：SAP 系統的系統 ID
- 用戶端：SAP 系統的用戶端
- 案例 ID：選取您的 Basis 管理員建立的DEFAULT案例
- 您要呼叫的區域AWS ： AWS 區域
- 身分驗證方法：選取下列其中一項：
  - 透過在 Amazon EC2 上執行之 SAP 系統的中繼資料的執行個體角色
  - 內部部署或其他雲端系統的 SSF Storage 登入資料
  - 用於憑證型身分驗證的 IAM Roles Anywhere

選取儲存。

5. 選取 IAM 角色映射 > 新項目，然後輸入：
   - 序號：1
   - 邏輯 IAM 角色：描述性名稱（例如 DEV_BASE_ROLE)
   - IAM 角色 ARN：第一個帳戶中 IAM 角色的 ARN（例如 arn:aws:iam::111111111111:role/DevBaseRole)

   選取儲存。

步驟 2 – 設定鏈結的設定檔

設定鏈結中的每個中繼和最終設定檔。

對於**SHARED_SERVICES**設定檔（來自 的鏈**DEV_BASE**)：

1. 執行/n/AWS1/IMG交易。
2. 選取AWS 適用於 SAP ABAP 的 SDK 設定 > 應用程式組態 > SDK 設定檔。
3. 選取新項目。輸入設定檔名稱（例如 SHARED_SERVICES) 和描述。選取儲存。
4. 選取您建立的設定檔，然後選取身分驗證和設定 > 新項目，然後輸入下列詳細資訊：
   - SID：SAP 系統的系統 ID
   - 用戶端：SAP 系統的用戶端
   - 案例 ID：選取您的 Basis 管理員建立的DEFAULT案例
   - 您要呼叫的區域AWS ： AWS 區域
   - 身分驗證方法：從下拉式清單中選取來源設定檔
   - 來源設定檔 ID：輸入基本設定檔的設定檔 ID（例如 DEV_BASE)

選取儲存。

5. 選取 IAM 角色映射 > 新項目，然後輸入：
   - 序號：1
   - 邏輯 IAM 角色：描述性名稱 （例如 SHARED_ROLE)
   - IAM 角色 ARN： `arn:aws:iam::222222222222:role/SharedServicesRole`

   選取儲存。

對於**PROD_S3_ACCESS**設定檔 （來自 的鏈**SHARED_SERVICES**)：

重複與 相同的步驟SHARED_SERVICES，但：

- 使用 PROD_S3_ACCESS做為名稱
- 將來源設定檔 ID 設定為 SHARED_SERVICES
- 在 IAM 角色映射`arn:aws:iam::333333333333:role/ProdS3AccessRole`中使用 PROD_S3_ROLE和

如需 IAM 角色管理、信任政策組態和授權需求等安全最佳實務，請參閱 IAM 安全的最佳實務。

使用 SAP 登入資料存放區

SAP 登入資料存放區用於 SAP Business Technology Platform，以安全地存放登入資料，以進行私密存取金鑰身分驗證 AWS。您必須擁有訂閱才能使用 服務。

下列指示假設您已設定 SDK 設定檔。如需詳細資訊，請參閱設定 適用於 SAP ABAP 的 AWS SDK。

開始組態之前，請確定您符合先決條件。如需詳細資訊，請參閱 SAP Credential Store。

主題

- 組態步驟
- 搭配 SDK 使用 SAP 登入資料存放區

組態步驟

步驟

- 步驟 1：設定身分驗證的設定
- 步驟 2：建立服務金鑰
- 步驟 3：將服務金鑰轉換為.p12格式
- 步驟 4：連線至 SAP BTP、ABAP 環境

步驟 1：設定身分驗證的設定

使用下列步驟來設定用於身分驗證的 Credential Store 設定。

1. 導覽至 SAP Credential Store 執行個體的設定索引標籤。
2. 選取編輯組態：
   - 選擇相互 TLS 作為預設身分驗證類型。
   - 針對承載加密狀態選取停用。承載會使用 HTTPS 在傳輸中加密。不過，承載目前無法進行雙重加密。
3. 選取儲存。

步驟 2：建立服務金鑰

使用下列步驟來建立 Credential Store 的服務金鑰。

1. 在 SAP Credential Store 應用程式的左側窗格中，導覽至服務金鑰。
2. 選取建立服務金鑰。
3. 輸入服務金鑰的名稱，然後選取建立。

服務金鑰會根據選擇的身分驗證類型建立。下載服務金鑰，並保持安全以供日後使用。

步驟 3：將服務金鑰轉換為.p12格式

需要 .p12 格式的用戶端憑證，才能建立通訊系統的傳出使用者。使用下列步驟，.p12從 Credential Store Service 金鑰中提供的憑證詳細資訊產生憑證。

1. 從 SAP Trust Center Services 下載 SAP Cloud 根 CA 憑證 (SAP 要求）。 https://support.sap.com/en/offerings-programs/support-services/trust-center-services.html
2. 以任何文字檔案格式開啟 SAP Cloud 根 CA 憑證。在檔案結尾，按下 Enter，然後從服務金鑰的憑證欄位中複製貼上憑證。將新行字元取代\n為實際的新行 (Enter)，並以.cer檔案格式儲存整個憑證。

3. 從服務金鑰的金鑰欄位複製金鑰。此私有金鑰必須視為敏感資料。將它貼到文字檔案中，並以實際的新行 \n (Enter) 取代新行字元。將私有金鑰儲存在文字檔案中。
4. 使用先前步驟中產生的憑證和私有金鑰，執行下列命令來產生.p12憑證。

```
openssl pkcs12 -export -out <.p12_filename> -inkey <private_key.key> -in
 <certificate.cer>
```

命令需要驗證匯出密碼。保留密碼以供進一步使用。

刪除儲存在私有金鑰中的.key文字檔案。

步驟 4：連線至 SAP BTP、ABAP 環境

設定 SAP BTP、ABAP 環境以與 SAP Credential Store 連線。

主題

- 通訊系統
- 通訊安排

通訊系統

使用下列步驟建立通訊系統，以啟用從 SAP BTP、ABAP 環境到 SAP Credential Store 的通訊。

1. 開啟 ABAP 環境系統的 Fiori 啟動台。
2. 選取通訊系統圖磚以開啟應用程式。
3. 選取新的。
4. 輸入通訊系統的名稱和 ID，然後選取建立。例如，您可以命名系統 ZSAP_CREDSTORE。
5. 輸入其他必要資訊：
   - 主機名稱：從服務金鑰 URL 複製主機名稱。例如，如果 URL 為 https://credstore.mesh.cf.us10.hana.ondemand.com/api/v1/credentials，則主機名稱為 credstore.mesh.cf.us10.hana.ondemand.com。
   - 外撥通訊的使用者：選取 + 以新增使用者。
     a. 選取 SSL 用戶端憑證做為身分驗證機制。
     b. 選取上傳新憑證：
        - 瀏覽上一個步驟中產生的.p12憑證。

- 輸入描述。
- 輸入用來產生 .p12 憑證的匯出密碼。
- 選取上傳。

c. 選取建立以建立傳出使用者。

6. 選取儲存。
7. 刪除上一個步驟中下載的服務金鑰。

通訊安排

使用下列步驟建立通訊安排，以提供傳出通訊的通訊案例。

1. 開啟 ABAP 環境系統的 Fiori 啟動台。
2. 選取通訊安排圖磚以開啟應用程式。
3. 選取新的。
4. 選取通訊案例 /AWS1/CRED_COMM_SCENARIO，然後輸入通訊安排的名稱。例如 Z_AWS_SDK_TO_SAP_CREDSTORE。
5. 選取建立。
6. 在通訊系統欄位中，瀏覽上一個步驟中建立的通訊系統。其他資訊會在系統選取後自動填入。
7. 選取儲存。
8. 選取檢查連線以測試您的連線。

一旦完成此設定，ABAP 環境可以使用通訊安排，透過傳出服務 (HTTP) 使用 SAP Credential Store 服務。

搭配 SDK 使用 SAP 登入資料存放區

步驟

- 步驟 1：建立命名空間和登入資料（們）
- 步驟 2：設定自訂業務組態應用程式

步驟 1：建立命名空間和登入資料（們）

使用 SAP 說明在 SAP 登入資料存放區中建立命名空間和登入資料 – 建立、編輯和刪除登入資料。

輸入下列詳細資訊以建立 Key 類型的登入資料。

- 命名空間 – 輸入命名空間的名稱，並將相關的登入資料分組在一起。
- 名稱 – 輸入金鑰的名稱。我們建議 aws-0123456789012-username，其中：
  - 0123456789012 是憑證授予存取權的 AWS 帳戶 ID
  - username 是登入資料所屬的 IAM 使用者名稱
- 值 – 輸入 base-64 編碼的私密存取金鑰。使用以下命令來對秘密存取金鑰進行 base-64 編碼。

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

  命令會從標準輸入讀取秘密存取金鑰，並將其傳遞至 base64，而不會有結尾換行。它會將 base-64 編碼的私密存取金鑰輸出至畫面。將值複製到 SAP Credential Store 之後，請清除或關閉終端機。
- 使用者名稱 – 輸入您的存取金鑰 ID。

- 選取建立。

系統會建立具有一個登入資料的新命名空間，並在此命名空間中新增、刪除或修改登入資料。

遵循最低權限原則來管理對存放在 命名空間中登入資料的存取。

步驟 2：設定自訂業務組態應用程式

使用下列步驟來設定自訂商業組態應用程式，以定義要用於 SDK 身分驗證的憑證。

1. 開啟 ABAP 環境系統的 Fiori 啟動台。
2. 瀏覽自訂業務組態圖磚以開啟應用程式。
3. 開啟 SDK Profile Business Configuration。
4. 選取必須為 SAP Credential Store 設定身分驗證設定的 SDK 設定檔。
5. 在所選設定檔的身分驗證和設定索引標籤中，選取編輯，然後輸入下列詳細資訊：
   - 身分驗證方法 – 從 SAP Credential Store 選取登入資料。
   - 命名空間 – 輸入在 SAP 登入資料存放區中建立的命名空間。如需詳細資訊，請參閱the section called “步驟 1：建立命名空間和登入資料（們）”。
   - 金鑰名稱 – 輸入所建立金鑰憑證的名稱。如需詳細資訊，請參閱the section called “步驟 1：建立命名空間和登入資料（們）”。

- 通訊安排 – 輸入所建立通訊安排的名稱。如需詳細資訊，請參閱the section called “通訊安排”。

6. 選取套用以前往 AWS SDK 設定檔畫面。
7. 選取選取傳輸以使用值說明選取傳輸。
8. 選取儲存。

故障診斷 適用於 SAP ABAP 的 AWS SDK

本節提供可能錯誤案例的疑難排解步驟。

主題

- 匯入失敗
- 未指定的位置限制條件
- SSL 錯誤
- 設定檔組態
- IAM 授權
- 執行必要動作的授權
- 作用中案例
- 程式碼中的特殊字元
- 連線能力

匯入失敗

問題 – 類別 'CL_SYSTEM_UUID' 不包含界面 'IF_SYSTEM_UUID_RFC4122_STATIC

原因 – 您的系統缺少 SAP Note 0002619546。

解決方法 – 確定 SAP Note 0002619546 已套用至您的系統。

未指定的位置限制條件

問題 – 未指定的位置限制條件與傳送此請求`region`的特定端點不相容

原因 – 您的 Amazon S3 儲存貯體缺少 `io_createbucketconfiguration` 參數中的 AWS 區域。

解決方法 – 在 以外的任何區域中建立儲存貯體時`us-east-1`，請使用 中的 `io_createbucketconfiguration` 參數指定 Amazon S3 儲存貯體的區域`createbucket()`。您不需要為 指定限制條件`us-east-1`。

下列範例顯示正確設定的`io_createbucketconfiguration`參數。

```
createbucket(
```

```
    iv_bucket = 'amzn-s3-demo-bucket'
    io_createbucketconfiguration = NEW /aws1/cl_s3_createbucketconf( 'us-west-1' )
).
```

SSL 錯誤

問題 – SSL 伺服器憑證主機名稱不相符或 SSL 交握與 docs.aws.amazon.com：//443 失敗：SSSLERR_NO_SSL_RESPONSE

原因 – icm/HTTPS/client_sni_enabled 參數未在 DEFAULT 設定檔TRUE中設定為 。

解決方法 – 使用下列步驟來疑難排解指定問題或任何其他 SSL 相關問題。

1. 開啟 SAPGUI 並前往命令列。
2. 執行交易 RZ10。
3. 前往設定檔，然後選擇DEFAULT設定檔。版本會自動填入。
4. 在編輯設定檔區段中，選取延長維護，然後選取變更。
5. 搜尋 icm/HTTPS/client_sni_enabled 參數。
   - 如果參數存在，請編輯參數值並將其設定為 TRUE。
   - 如果 參數不存在，請使用下列步驟建立參數。
     1. 選取 參數。

        > **Note**
        > 請確定您選取要建立的參數，而不是編輯 （鉛筆圖示）。

     2. icm/HTTPS/client_sni_enabled 在參數名稱欄位中輸入 。
     3. TRUE 在參數值欄位中輸入 。
     4. 選取 Save (儲存)。
6. 在DEFAULT設定檔中儲存這些變更，然後結束。

設定檔組態

問題 – 找不到描述檔 <profile_name> 下的組態，案例為 DEFAULT for <sid>：<client>

原因 – <profile_name> 不正確或尚未設定。

解決方案 – 使用下列步驟來設定設定檔。

1. 開啟 SAPGUI 並執行交易 /n/AWS1/IMG。
2. 移至應用程式組態 > SDK 設定檔。
   - 如果您的設定檔已設定，請確認設定檔名稱正確。
   - 如果您的設定檔未設定，請依照下列步驟設定設定檔。
3. 選取新增項目。
   a. 輸入設定檔的名稱和描述。
   b. 選取 Save (儲存)。
4. 選擇您在上一個步驟中建立的項目，然後選取身分驗證和設定。
5. 選取新增項目，輸入下列詳細資訊，然後選取儲存。
   - SID
   - 用戶端
   - 案例 ID
   - AWS 區域
   - 身分驗證方法
     - 針對執行於 的 SAP 系統，透過中繼資料選取執行個體角色 AWS。
     - 從 SSF Storage 中選取登入資料，以供內部部署或其他雲端執行的 SAP 系統使用。
6. 選取 IAM 角色映射 > 新項目，輸入下列詳細資訊，然後選取儲存。
   - 序號
   - 邏輯 IAM 角色
   - IAM 角色 ARN

IAM 授權

問題 – 無法擔任角色 <iam_role_arn> 或使用者：<user_arn> 未獲授權執行：sts：AssumeRole on resource：<iam_role_arn>

原因 – 下列可能是此錯誤的潛在原因。

- 指定的 IAM 角色 ARN 不正確
- IAM 使用者缺少存取 IAM 角色的許可
- 擔任的 IAM 角色與擔任的 IAM 角色或 IAM 使用者之間缺乏信任關係

解決方案 – 使用下列步驟來確保 IAM 角色 ARN 正確。

1. 開啟 SAPGUI 並執行交易 `/n/AWS1/IMG`。
2. 移至應用程式組態 > SDK 設定檔，然後選擇已使用 IAM 角色設定的設定檔。
3. 選取 IAM 角色映射，並驗證或更正您的 IAM 角色 ARN。
    - 如果您的 IAM 角色 ARN 正確，請確定您的 IAM 角色已正確設定。如需詳細資訊，請參閱疑難排解 IAM 角色。

執行必要動作的授權

問題 – 使用者 <user_arn> 未獲授權，無法對資源執行：<action>：<resource_arn>

原因 – 使用者沒有執行動作的許可。

解決方案 – `user_arn` 必須在 上設定必要的許可`resource_arn`，才能執行指定的 `action`。如需詳細資訊，請參閱存取 IAM 資源所需的許可。

作用中案例

問題 – 未設定作用中案例

原因 – 遺漏了作用中案例的設定。

解決方法 – 請參閱執行期設定以設定作用中案例。

程式碼中的特殊字元

警告 – 字元 0x00A0 不能是 ABAP 文字的一部分

> **Note**
>
> 此警告前面可能會顯示不同的錯誤訊息。

原因 – 從不同來源複製和貼上程式碼可以在程式碼中插入特殊字元。

解決方法 – 當您在 ABAP 原始程式碼編輯器中貼上任何程式碼時，您會看到下列快顯視窗。

偵測到不中斷的空間字元。轉換為空格？

選擇是以回答此問題。此外，我們建議您選取程式碼來複製，而不是使用程式碼方塊中的複製按鈕。

連線能力

問題 – SCLNT_HTTP(411)：直接連線至 tla.region.amazonaws.com：//443 失敗：NIECONN_REFUSED(-10)

原因 – SAP 系統沒有網際網路連線，而且無法建立與 tla.region.amazonaws.com 連接埠 443 的 TCP/IP 連線。

解決方案 – SAP 系統必須能夠直接或透過代理伺服器建立與 HTTPS 連接埠 443 AWS 端點的連線。您可以使用下列其中一個選項建立/驗證網際網路連線。

- 透過 NAT 或網際網路閘道直接對外連線至網際網路
- 透過代理伺服器的連線

  如需詳細資訊，請參閱透過代理伺服器的連線。

其他主題

本節涵蓋下列主題。

主題

- 適用於 SAP ABAP 的 AWS SDK 版本
- SAP 授權

適用於 SAP ABAP 的 AWS SDK 版本

AWS 適用於 SAP ABAP 的 SDK 以傳輸形式交付，適用於 SAP ABAP 的 AWS SDK - BTP 版本則以附加元件形式交付。匯入傳輸和附加元件的機制不同，但技術功能相同。如需詳細資訊，請參閱設定。

主題

- 發行策略
- 最佳實務
- 修補適用於 SAP ABAP 的 SDK
- 安裝其他模組
- 解除安裝適用於 SAP ABAP 的 SDK

發行策略

第 1 版 適用於 SAP ABAP 的 AWS SDK 會經常更新。根據 的版本和更新，每週或每天都會發行新修補程式 AWS 服務。的修補程式 AWS 服務 可以包含更新 SDK 修補程式層級的錯誤修正和其他變更。如需詳細資訊，請參閱 AWS SDKs和工具維護政策。

最佳實務

我們建議為所有 SAP 系統 （開發、QA 和生產） 保留相同的適用於 SAP ABAP 的 SDK 修補程式層級。

修補 SDK 時，請在沙盒中匯入最新版本。然後，您可以依照正常的變更控制程序，將其匯入到開發、QA 和生產系統。

修補適用於 SAP ABAP 的 SDK

每個適用於 SAP ABAP 的 SDK 版本都會以一組累積傳輸形式交付，包括所有錯誤修正、功能和更新。修補程式和安裝傳輸之間沒有差異。您必須將最新的傳輸匯入修補程式 SDK for SAP ABAP。

由於 `core` Runtime 和 API 模組的相依性，個別 API 模組無法單獨修補。修補時，您必須更新已安裝的`core`模組和所有其他 API 模組，即使某些模組不再使用。

下列案例說明修補程序：

- 案例 1：定期 SDK 版本更新

  修補至最新的 SDK 版本時：

  - 同時匯入所有傳輸（核心和已安裝的 API 模組）
  - 不需要特定的匯入序列
  - 例如，如果您在安裝 SDK 時匯入 `core`、 `ec2`和 `lmd`傳輸，則必須在修補`lmd`時匯入 `core`、`ec2`和 的最新傳輸。
- 案例 2：新增 API 模組

  從與目前安裝模組不同的 SDK 版本匯入新的 API 模組（例如 Amazon Textract `tex`) 時：

  - 必須修補所有現有的模組，以符合新模組的 SDK 發行版本
  - 同時匯入（無特定序列）：
    - 所有現有模組 (`core` 和已安裝的 API 模組） 的最新傳輸
    - 新 API 模組的傳輸 `tex`
- 案例 3：更新現有的 API 模組

  將現有 API 模組（例如 Amazon Translate `xl8`) 更新至最新的 SDK 版本時，由於相互依存性，因此不支援個別模組更新。您必須依照 中概述的程序，將`core`模組和所有已安裝的 API 模組一起更新，以修補整個 SDKRegular SDK Version Update (p. 559)。

安裝其他模組

在與現有 `core`和 模組相同的修補程式層級匯入新模組的傳輸，以在 SAP 系統中安裝額外的 API 模組。the section called “修補適用於 SAP ABAP 的 SDK” 如果您想要匯入更新版本的模組，請遵循 中的準則。這可確保修補程式層級在所有 SDK 模組之間相容。

解除安裝適用於 SAP ABAP 的 SDK

若要解除安裝適用於 SAP ABAP 的 SDK，您必須從 https：//https://sdk-for-sapabap.aws.amazon.com/awsSdkSapabapV1/release/uninstall-abapsdk-LATEST.zip 下載刪除傳輸套件。

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

您可以從 https：//https://sdk-for-sapabap.aws.amazon.com/awsSdkSapabapV1/release/uninstall-abapsdk-LATEST.sig 下載簽章檔案。若要驗證檔案，請參閱驗證適用於 SAP ABAP 的 SDK。

對於 SAP 系統上安裝的每個 SDK 模組，必須從先前的 ZIP 檔案匯入對應的刪除傳輸。您可以移除單一模組，而無需解除安裝整個 SDK。您可以僅匯入要移除之模組的刪除傳輸來執行此操作。如果您要解除安裝整個開發套件及其所有模組，則必須最後匯入核心刪除傳輸。

我們建議您在嘗試開發、QA 或生產系統之前，先在沙盒中測試解除安裝。

考量

解除安裝 SDK 之前，請參閱下列考量事項。

- 來自 的 SDK 組態設定將會遺失。安裝時IMG必須重新設定 。
- 如果您有依賴軟體開發套件的 Z 程式，它們會在移除軟體開發套件後產生語法錯誤。
- 包含 SDK 授權參考的 PFCG 或商業角色在移除 SDK 後會有無效的授權。解除安裝 SDK 之前，請從 PFCG 角色移除 SDK 授權參考。

> **Note**
>
> AWS 適用於 SAP ABAP 的 SDK - 開發人員預覽期間無法解除安裝 BTP 版本。

SAP 授權

SAP 軟體的使用受 SAP 條款的約束。您有責任遵守 SAP 授權條款，包括軟體分發和間接授權條件。提供的任何資訊都不是法律建議，不應基於授權合規目的而依賴。如果您對 SAP 軟體的授權或權利有任何疑問，請洽詢您的法務團隊、SAP 和/或 SAP 經銷商。

問題 ：適用於 SAP ABAP 的 SDK 用量會影響我的 SAP 授權嗎？

答案： 適用於 SAP ABAP 的 AWS SDK 可讓您 AWS 服務 使用自己的 ABAP 程式碼。它用於 SAP 系統和 之間的整合案例 AWS 服務。來自 SAP 系統的資料傳送至第三方 （非 SAP) 系統或由該系統建立的任何案例，都可能會對間接授權造成影響。SAP 有多種方法來定義間接存取，例如以使用者為基礎的計算和以結果為基礎的計算。定義間接存取的方法取決於您與 SAP 的合約。您必須了解與 SAP 簽訂的合約中提供的指引，而且您可以進一步與 SAP 或其經銷商討論。

在 2018 年，SAP 發佈了兩份文件：SAP 已安裝基本客戶的間接存取指南，以及適用於數位年齡的 SAP ERP 定價 - 解決間接/數位存取。您可以在 SAP 網站上找到這些文件，並且是間接授權方法的範例。不過，這些文件不會反映您與 SAP 的特定協議。

適用於 SAP ABAP 的 AWS SDK 開發人員指南的文件歷史記錄

下表說明 文件的發行版本 適用於 SAP ABAP 的 AWS SDK。

變更	描述	日期
新內容	新增使用來源描述檔進行跨帳戶存取。	2026 年 3 月 4 日
新內容	新增搭配 SSF 加密使用私密存取金鑰身分驗證。	2025 年 1 月 12 日
新內容	適用於 SAP ABAP 的 SDK 開發人員預覽 - BTP 版本。	2024 年 5 月 31 日
新內容	新增使用具有 IAM Roles Anywhere 的憑證。	2023 年 12 月 1 日
新內容	新增使用 SDK 建置產品。	2023 年 12 月 1 日
新內容	新增重試行為。	2023 年 12 月 1 日
新內容	新增 SAP 授權。	2023 年 9 月 22 日
公開發行	適用於 SAP ABAP 的 AWS SDK 開發人員指南的初始重新啟動。	2023 年 6 月 30 日
新內容	新增 適用於 SAP ABAP 的 AWS SDK 功能。	2023 年 5 月 30 日
新內容	新增故障診斷 適用於 SAP ABAP 的 AWS SDK。	2023 年 2 月 17 日
開發人員預覽版	開發人員指南的 適用於 SAP ABAP 的 AWS SDK 開發人員預覽。	2022 年 11 月 17 日

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。