



開發人員指南

AWS SDK 適用於 SAP ABAP



AWS SDK 適用於 SAP ABAP: 開發人員指南

Copyright © 2024 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
維護	1
API 參考	1
定價	2
資源	2
開始使用	3
步驟 1：準備您的 AWS 帳戶	3
IAMSAP使用者的角色	3
身分驗證	4
步驟 2：安裝 SDK	5
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：運行應用程序	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 的開發套件-BTP 版	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 開發套件傳輸	21
安裝 SDK-BTP 版	25
安SDK裝 SAP ABAP-BTP 版本	25
模組	25
修SDK補 SAP ABAP-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	29
SDK 設定檔	30

邏輯資源解析器	31
範例	31
執行期設定	32
記錄和追蹤	32
選擇加入：增強型遙測	32
作用中案例	32
進階連線案例	32
透過代理伺服器連線	33
通過數據包檢查防火牆進行連接	33
閘道端點	34
自訂介面端點	34
存取多個區域中的端點	35
服務提供者設定	36
重新整理、追蹤和遙測	36
SAP 系統重新整理	36
追蹤	37
遙測	38
使用開發套件	39
數據表示	39
資料類型	39
AWS 資料類型	41
示例程序	42
先決條件	42
代碼	43
程式碼區段	43
概念	45
API 類別	46
其他物件	46
結構類	47
陣列	48
地圖	49
更高級別的函數	49
功能	1
程式設計組態	50
等待程式	51
分頁程式	52

重試行為	53
建築產品	53
設定產品 ID	53
限制	54
程式碼範例	55
Amazon Bedrock 執行期	55
Anthropic Claude	56
Stable Diffusion	58
CloudWatch	61
動作	61
案例	65
DynamoDB	68
基本概念	68
動作	61
Amazon EC2	82
動作	61
Kinesis	97
基本概念	68
動作	61
Lambda	107
基本概念	68
動作	61
Amazon S3	120
基本概念	68
動作	61
SageMaker	129
動作	61
案例	65
Amazon SNS	147
動作	61
案例	65
Amazon SQS	155
動作	61
案例	65
Amazon Textract	163
動作	61

案例	65
Amazon Translate	173
動作	61
案例	65
安全	182
系統認證	182
元數據驗證	183
密鑰訪問密鑰認證	183
使用任何角色進行憑證型驗證 IAM	184
下一步驟	184
IAM 安全的最佳做法	184
亞馬遜 EC2 實例配置文件的最佳實踐	185
適用於 SAP 使用者的 IAM 角色	185
SAP 授權	188
組態的授權	188
SAP 最終使用者的授權	189
安全作業	190
靜態資料加密	190
傳輸中的資料加密	190
API 使用情況	1
使用憑證	191
必要條件	191
程序	191
認證存放區	20
設定步驟	195
搭配 SDK 使用 SAP 認證存放區	196
疑難排解	200
匯入失敗	200
未指定的位置限制	200
SSL 錯誤	201
設定檔組態	201
IAM 授權	202
動作的授權	203
作用中案例	32
特殊字元	203
連線能力	204

其他主題	205
推出	205
發行策略	205
最佳實務	184
修補軟體開發套件	206
安裝附加模組	206
解除安裝軟體開發套件	206
SAP 授權	207
文件歷史紀錄	208
.....	ccix

什麼是 適用於 SAP ABAP 的 AWS SDK ？

適用於 SAP ABAP 的 AWS SDK 提供以該ABAP語言提供的服務 AWS 的接口。使用SDK，您可以在上實作ABAPBADIs、報告、交易、OData服務和其他ABAP成品 AWS 服務，例如亞馬遜簡單儲存服務 (Amazon S3) Amazon DynamoDB Amazon Translate、等等。您也可以為ABAP基礎的系統進行開發，從 SAP NetWeaver 7.4 開始，並在SAP商業技術平台環境中。如需詳細資訊，請參閱[AWS SDK 安裝 SAP ABAP-BTP 版本](#)。

主題

- [的功能 適用於 SAP ABAP 的 AWS SDK](#)
- [SDK主要版本的維護和支持](#)
- [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 已被設計為開SAP發人員感到熟悉和自然。例如，儘管所有 AWS 服務 使用true和false字符串來表示布爾數據XML和JSON結構，但用SDK於SAPABAP將這些數據轉換為 ABAP-native 'X' 和 ' ' 單字符值。SDKin 盡可能SAPABAP使用本地ABAP構造，包括數據類型和時間戳格式。因此，程式設計人ABAP員不需要關心基礎JSON和XML序列化或API通訊協定的線格式。

SDK主要版本的維護和支持

如需有關SDK主要版本及其基礎相依性的維護和支援的資訊，請參閱《[AWS SDKs和工具參考指南](#)》中的下列內容：

- [AWS SDKs和工具維護政策](#)
- [AWS SDKs和工具和版本支援對照表](#)

API 參考

要查看的完整列表 適用於 SAP ABAP 的 AWS SDK APIs，請參閱 [適用於 SAP ABAP 的 AWS SDK - API 參考指南](#)。

若要查看的完整模組清單 適用於 SAP ABAP 的 AWS SDK TLAs，請參閱 [適用於 SAP ABAP 的 AWS SDK - 模組清單](#)。

要查看完整的模塊列表 SAP ABAP-BTP 版本開發人員預覽TLAs，請參閱 [AWS SDK 閱 SAP ABAP-BTP 版本-模塊列表](#)。SDK

定價

適用於 SAP ABAP 的 AWS SDK 可供您使用，無需額外費用。您只需為使用的 AWS 資源和服務付費 SDK。

其他資源

除了本指南之外，還有下列線上資源可供使SDK用SAPABAP。

- [SAP關於 AWS 文件](#)
- [AWS 開發者博客](#)
- [AWS 開發者論壇](#)
- [AWS SDK程式碼範例程式](#)
- [@awsdevelopers](#) (推特)

開始使用 適用於 SAP ABAP 的 AWS SDK

本節說明如何開始使用SDK. 其中包含有關安裝SDK、執行基本設定，以及建立 Hello World 程式碼範例的相關資訊，將片語從一種語言翻譯成另一種語言。如果您是新手 AWS SDK，建議您在沙箱環境中執行這些步驟。

步驟

- [步驟 1：準備您的 AWS 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：運行應用程序](#)

步驟 1：準備您的 AWS 帳戶

若要開始使用SDK的 SAPABAP，您必須有一個作用中的 AWS 帳戶。AWS 帳戶即使您的SAP系統位於內部部署、SAP商務技術平台 (BTP) 或其他雲端供應商託管，您也需要一個。

如果您的SAP系統在 AWS Cloud 上運行，那麼您將在 AWS 帳戶. AWS

主題

- [IAMSAP使用者的角色](#)
- [身分驗證](#)

IAMSAP使用者的角色

- 按照《AWS Identity and Access Management 用戶指南》中提供的說明創建IAM角色。如需詳細資訊，請參閱[建立角色以將權限委派給 AWS 服務](#)。請記下IAM角色的 Amazon 資源名稱 (ARN) 以供日後使用。
- 選擇 Amazon EC2 作為用例。
- 作SapDemoTranslate為角色的名稱使用。
- 將TranslateReadOnly設定檔附加至角色。

- 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。

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

此範例顯示來自的任何主參與者都 AWS 帳戶 **"111122223333"** 可以擔任該角色。這是一個廣泛的權限，適用於 proof-of-concept. 您可以使用較窄的主體來進行生產，例如下列範例。

- 特定使用者 — 當系SAP系統正在使用下列其中一項時：
 - SSF來自內部部署SAP系統的加密認證
 - 來自SAP認證存放區服務的認證 SAPBTP, ABAP 環境
- 特定角色 — SAP 系統在 Amazon 上EC2且有執行個體設定檔時。
- Amazon EC2-當SAP系統在 Amazon 上EC2並且沒有實例配置文件時。

如需詳細資訊，請參閱[IAM安全性的最佳做法](#)。

身分驗證

驗證取決於您的SAP系統託管位置。

位置

- [在 AWS 雲端](#)
- [內部部署SAPBTP或其他雲端](#)

在 AWS 雲端

請確定EC2執行SAP系統的執行個體擁有具有下列權限的執行個體設定檔。

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

新增您ARN在上一個步驟中記下的。

此權限可讓您的SAP系統代表ABAP使用者擔任該SapDemoTranslate角色。

內部部署SAPBTP或其他雲端

如果您的SAP系統位於內部部署、雲端上SAPBTP或其他雲端上，請使用下列步驟建立使用秘密存取金鑰驗證的連線。

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. SapDemoSID作為使用者的名稱使IAM用。SID是您系統的SAP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系統中設定這些認證。

Note

如果您的SAP系統位於內部部署、在其他雲端上SAPBTP或其他雲端上，您可以使用下列其中一個選項進行驗證。

- 使用SSF或SAP認證存放區的[秘密存取金鑰驗證](#)
- [使用憑證搭配任何IAM角色](#)

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

匯入SDK以便在您的SAP系統中進行SAPABAP傳輸。您可以將傳輸匯入任何用戶端。如需詳細資訊，請參閱[安裝SDK的 SAP ABAP](#)。

SDK for SAP ABAP - BTP edition

使SDK用SAPABAP部署產品應用程式安裝BTP版本。如需詳細資訊，請參閱[SDK安裝 SAP ABAP-BTP 版本](#)。

步驟 3：設定 SDK

在配置之前SDK，請確定您具有必要的授權。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交易以開啟的「實行SDK指南」SAP ABAP。若要執行此交易，請/n/AWS1/IMG在SAP系統的命令列中輸入，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連線能力](#)。
- 轉到全局設置 → 配置方案。
 - 根據「[全域](#)」設定中的建議變更設定。
- 轉到全局設置 → 技術設置。
 - 根據「[全域](#)」設定中的建議變更設定。
- 轉到運行時設置 → 日誌和跟踪。
 - 選取「新項目」。
 - 跟踪級別：無跟踪。
 - 最大傾印行數：100。
 - OPT-在：enh 遙測：保持這個空白。
 - 選取 Save (儲存)。
- 轉到運行時設置 → 活動方案。
 - 在新案例下，選取DEFAULT。

- 選取確認案例變更。
- 接受提示。

內部部署系統的前提

如果您的SAP系統在內部部署或其他雲端中執行，則認證必須儲存在資SAP料庫中。認證會使用加密，SAPSSF且需要設定SAP的 CommonCryptoLib密碼編譯程式庫，例如。

的配置SSF步驟SAPABAP在/AWS1/IMG交易中說明。SDK

Note

如果您的SAP系統在 Amazon 上執行，則上述先決條件不適用EC2。SAP在 Amazon 上執行的系統會從 Amazon 執行個體中繼資料EC2擷取短暫且自動輪換的登入EC2資料。

SDK for SAP ABAP - BTP edition

在網頁瀏覽器中開啟您的ABAP環境，然後瀏覽至「自訂商務組態」應用程式。

完成下列組態。

- 移至設定案例。
 - 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。

SDK for SAP ABAP

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

- 轉到應用程式配置 → 配置SDK文件。
 - 選取「新項目」。

- 配置文件：DEMO。
- 說明:Demo profile.
- 選取 Save (儲存)。
- 反白您建立的項目，然後按一下 [驗證和設定] 樹狀目錄分支。
 - 選取「新項目」。
 - SID：您目前所在SAP系統的系統 ID。
 - 用戶端：您目前所在SAP系統的用戶端。
 - 案例 ID：下拉式清單，您可以在其中找到您的 Basis 管理員所建立的DEFAULT案例。
 - AWS 區域：輸入您要撥打電話的「AWS 區域」。如果您的SAP系統正在執行中 AWS，請輸入該系統執行的 AWS 地區。
 - 驗證方法：
 - 如果SAP系統在 Amazon 上執行，請透過中繼資料選取執行個體角色EC2。
 - 如果您的SAP系統在內部部署或其他雲端中執行，請選取來自SSF儲存的認證。
 - 選取「設定認證」。
 - 輸入您在上一個步驟中建立的「存取金鑰 ID」和「秘密存取金鑰」。
 - 將停用IAM角色保持空白。
 - 選取 Save (儲存)。
 - 按一下「IAM角色對應」樹狀結構分支。
 - 選取「新項目」。
 - 輸入「序號」：010。
 - 輸入邏輯IAM角色：TESTUSER。
 - 輸入IAM角色 ARN：輸入角色的 `arn: aws:`，該IAM角色包含在上一個步驟中建立的TranslateReadOnly策略。

SDK for SAP ABAP - BTP edition

使用SAP認證存放區設定驗證。如需詳細資訊，請參閱[使用SAP認證存放區](#)。

在網頁瀏覽器中開啟您的ABAP環境，然後瀏覽至「自訂商務組態」應用程式。

- 轉到「SDK配置文件」。
 - 選取「編輯」以建立新的設定檔。
 - 配置文件：DEMO。

- 說明:Demo profile.
- 選取已建立項目旁邊的向右箭頭鍵，以導覽至 [驗證和設定] 索引標籤。

選取「新項目」。

- SID：您目前所在SAP系統的系統 ID。
- 用戶端：您目前所在SAP系統的用戶端。
- 案例 ID：下拉式清單，您可以在其中找到您的 Basis 管理員所建立的DEFAULT案例。
- AWS 區域：輸入您要撥打電話的「AWS 區域」。如果您的SAP系統正在執行中 AWS，請輸入該系統執行的 AWS 地區。
- 驗證方法：從認證存放區中選取SAP認證。
- 輸入「認證存放區」中SAP所儲存之認證的命名空間和金鑰名稱。
- 輸入建立的「通訊安排」名稱，以建立「BTP版本」與「SAP認證存放區」之SDK間的通訊。SAP ABAP
- 將停用IAM角色保持空白。
- 在建立的項目旁邊的右箭頭鍵上按一下滑鼠右鍵，以導覽至「IAM角色對應」索引標籤。

選取「新項目」。

- 輸入「序號」：010。
- 輸入邏輯IAM角色：TESTUSER。
- 輸入IAM角色 ARN：輸入角色的 `arn: aws:`，該IAM角色包含在上一個步驟中建立的TranslateReadOnly策略。

步驟 5：授權SAP使用者

SAP根據預設，使用者未獲授權使用 AWS 功能。必須使用授權明確SAP授權使用者。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。
- 說明:Role for demo AWS SDK functionality.

- 移至「授權」標籤。
- 選取「變更授權資料」並接受資訊快顯視窗。
- 在「選擇範本」彈出式視窗中，選取「不要選取範本」。
- 從工具列中選取「手動新增」。
- 新增下列授權物件：
 - /AWS1/LROL
 - /AWS1/SESS
- 在授權樹狀結構中，輸入：
 - 訪問配置文件 AWSAPIs : DEMO
 - 邏輯IAM角色 : TESTUSER
- 選取 Save (儲存)。
- 選取「產生」。
- 選取 [返回]。
- 選取 [儲存] 以儲存角色。

將PFCG角色指派給使SAP用者

任何具有指定ZAWS_SDK_DEMO_TESTUSER角色的使用者都將獲得授權，可以使用配置DEMOSDK文件中配置的設置的 AWS SDK功能。授權使用者也會擔任對IAM應至該設定檔中TESTUSER邏輯IAM角色的角色。

- 執行交易SU01。
 - 輸入要測試 AWS SDK功能之SAP使用者的使用者 ID。
 - 選取 [變更]。
 - 移至 [角色] 索引標籤，並將ZAWS_SDK_DEMO_TESTUSER角色指派給使用者。
 - 選取 Save (儲存)。

SDK for SAP ABAP - BTP edition

建立商務角色

- 在 Web 瀏覽器中開啟您的ABAP環境，然後導覽至維護商務角色應用程式。

步驟 5: 授權SAP使用者

- 選取「從範本建立」，然後輸入下列詳細資訊。

- 模板-選擇/AWS1/RT_BTP_ENDUSER。
- 新商務角色 ID — 輸入 ID。
- 新企業角色描述 — 輸入說明。
- 選取確定以查看商務角色的頁面。
- 在「一般角色詳細資料」標籤下，移至「存取類別」，並將「寫入」、「讀取」、「值說明」欄位設定為「受限」。
- 選取維護限制，然後展開左側導覽窗格中的「指派的限制類型」。更新「限制與值」區段中的下列欄位。
 - 在「選擇SDK工作階段」下，選取「SDK設定檔」旁邊的鉛筆圖示，然後導覽至「範圍」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」旁邊的鉛筆圖示，然後瀏覽至「範圍」索引標籤。輸入**TESTUSER**，然後選取「新增」。

選擇旁邊的鉛筆圖標SDK簡介，然後導航到範圍選項卡。輸入**DEMO**，然後選取「新增」。

- 導覽回「商務角色」範本，然後開啟「商務使用者」標籤。選取新增以將新建立的商務角色指派給將測試SDK功能的SAP商業使用者。選取 Save (儲存)。

指派給已建立商務角色的任何商務使用者都將獲得授權，可以使用在設定DEMOSDK檔中設定的設定的 AWS SDK功能。授權使用者也會擔任對IAM應至該設定檔中TESTUSER邏輯IAM角色的角色。

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. 開啟的交易SE38。
 - 輸ZDEMO_TRANSLATE_HELLO_WORLD入程式名稱。
 - 選取 Create。
 - 輸入AWS SDK Hello World In Any Language作為標題。
 - 類型：選擇可執行程序。
 - 狀態：選擇「測試程式」。
 - 選取 Save (儲存)。

- 將程式儲存為「本機物件」。

添加以下代碼。

```

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

```

SDK for SAP ABAP - BTP edition

1. 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. 輸入ZCL_DEMO_XL8_SIMPLE類別名稱，然後加入類別說明。選取下一步。
3. 建立或選擇傳輸要求。選取 [完成]。

添加以下代碼。

```
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_xl8unsuppdedlanguage00 INTO DATA(lo_lang).
      out->write( |ERROR - Cannot translate from { lo_lang->sourcelanguagecode }
        to { lo_lang->targetlanguagecode }| ).
```

```
CATCH cx_root INTO DATA(lo_root).
    out->write( |ERROR - { lo_root->get_text( ) }| ).
ENDTRY.
ENDMETHOD.
ENDCLASS.
```

如需如何撰寫使用的ABAP程式碼的詳細資訊SDK，請參閱[使用 適用於 SAP ABAP 的 AWS SDK](#)。

步驟 7：運行應用程序

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、組態或基礎必要條件，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角色授權您使用設定SDK檔，並將其對應到邏輯IAM角色，而您的IAM權限未設定給SAP系統擔任該IAM角色，則以下內容將是您的輸出。

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

在此情況下，請檢閱中定義之IAM角色、使用者或兩者的IAM權限和信任組態[the section called “步驟 1：準備您的 AWS 帳戶”](#)。

SDK for SAP ABAP - BTP edition

在 Eclipse > 運行方式 > 應用ABAP程序 (控制台) 上運行應用程序。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、組態或基礎必要條件，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角色授權您使用設定SDK檔，並將其對應到邏輯IAM角色，而您的IAM權限未設定給SAP系統擔任該IAM角色，則以下內容將是您的輸出。

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

在此情況下，請檢閱中定義之IAM角色、使用者或兩者的IAM權限和信任組態[the section called “步驟 1：準備您的 AWS 帳戶”](#)。

設定

本節提供有關如何設定要使用的開發環境的資訊 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 AWS SDK 裝 SAP ABAP-BTP 版本](#)

SAP 先決條件

在裝載 SAP 系統時，適用於安裝 SDK 的下列先決條件 AWS。

主題

- [AWS 軟體開發套件的先決條件](#)
- [SAP ABAP AWS 軟體開發套件的先決條件-BTP 版](#)

AWS 軟體開發套件的先決條件

以下是適用於 SAP ABAP 的 AWS SDK 的先決條件。

主題


- [基礎版本](#)
- [核心版本](#)
- [參數](#)
- [備註](#)
- [出站連線](#)
- [HTTPS 連線能力](#)
- [存取 Amazon EC2 執行個體中繼資料](#)

基礎版本

適用於 SAP NetWeaver 7.4 及更高版本的開發套件相容。適用於 SAP ABAP 的 SDK 不會觸及任何 SAP 應用程式資料表。它是完全不可知的應用程序，如 SAP 企業資源規劃和 SAP 景觀轉換複製服務器。

支援的最低 SP 層級 SAP_BASIS 740 為 SP 0008。如需詳細資訊，請參閱 [CL_HTTP_ENTITY](#) 中相同名稱的支援表單欄位 (需要存取 SAP 入口網站)。根據您的業務需求，您可以選擇較高的 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
PL_BASIS	740	0008	SAPK-74008INSAPBASIS	Basic Plug-In

SAP_BASIS 750 和更高版本沒有 SP 層級的最低要求。

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，仰賴 SAP 核心的加密、HTTP、XML 和 JSON 功能。我們建議您使用與 SAP NetWeaver 平台相容的最新核心版本。最低要求是內核版本 741。如需詳細資訊，請參閱 [SAP 注意事項 2083594-SAP 核心版本和 SAP 核心修補程式層級](#) (需要存取 SAP 入口網站)。

如果您使用的是核心版本 741 或 742，則需要下列修補程式等級：

- 修補程式
- 修補程式

參數

您的 SAP 系統必須支援伺服器名稱指示 (SNI)，如下列 SAP 備註所述 (需要存取 SAP 入口網站)。

- [SAP 附註 2124480-IC/ 網頁調度程式：TLS 延伸伺服器名稱指示 \(SNI\) 做為用戶端](#)
- [SAP 注意事項-用戶端傳送 TLS 延伸 SNI 的 SAPSSL 更新，透過 SAPHTTP、分析程式、網路安全傳輸 \(SAPSSL\)](#)

在檔案中設定下列參數DEFAULT.PFL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>

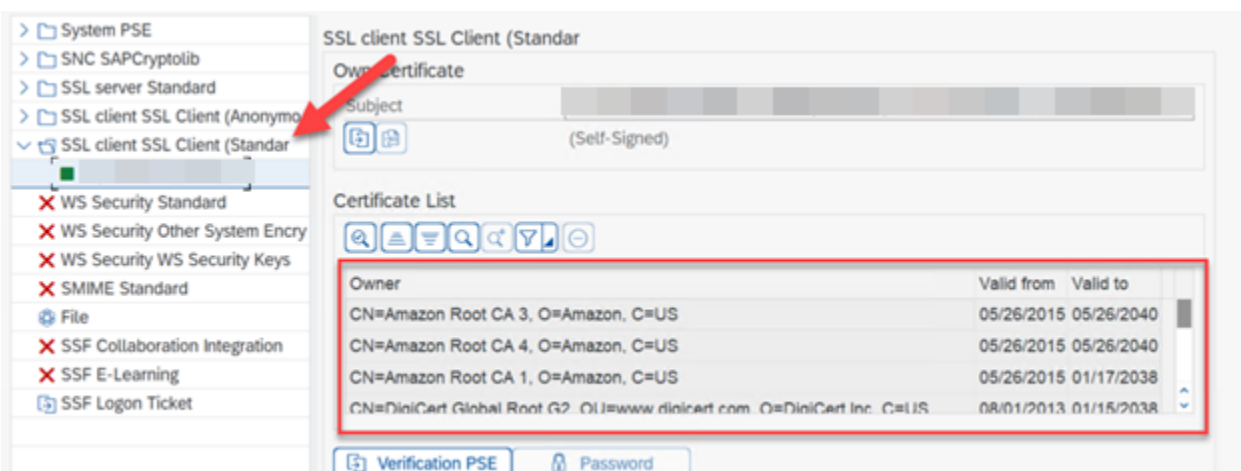
出站連線

開發套件為 SAP ABAP 是一個 HTTPS 用戶端。SAP 系統會將 HTTPS 郵件傳送出埠。不需要輸入連線。

HTTPS 連線能力

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，才能建立輸出 HTTPS 連線。

1. 轉到 <https://www.amazontrust.com/repository/>。
2. 在根 CA 下，使用 PEM 連結下載所有憑證。
3. 將這些憑證匯入您SSL Client (Standard) PSE每個 SAP 系統的中，如下圖所示。STRUST



存取 Amazon EC2 執行個體中繼資料

ABAP 系統與本地主機 (<http://169.254.169.254>) 進行未加密的 HTTP 連接，以啟用 Amazon EC2 實例元數據。HTTP 通道僅用於從本機伺服器擷取 AWS 認證。HTTP 流量會保留在主機內。

中繼資料可讓 SAP 系統安全地驗證本身，而無需在 SAP 安全存放區中儲存秘密金鑰。AWS 此功能僅適用於託管在 Amazon EC2 上的 SAP 系統。

使用下列參數設定DEFAULT.PFL檔案，讓 SAP 系統建立未加密的輸出 HTTP 連線。

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

使用下列參數來啟用輸出 HTTP 連線，而不啟用輸入連線。

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

使用下列步驟確認 SAP 系統已針對輸出 HTTP 連線設定：

1. 執行中芯交易。
2. 移至 [使用中的服務]。
3. 確認您在 HTTP 列的「作用中」欄下看到綠色的核取記號，如下圖所示。

Active Services						
No.	Protocol	Service Name/Port	Host Name	Keep Alive	Proc.Timeo	Actv E:
<input type="checkbox"/>	1	HTTPS	50001	60	600	✓
<input type="checkbox"/>	2	HTTP	0	60	600	✓

SAP ABAP AWS 軟體開發套件的先決條件-BTP 版

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

主題

- [SAP 景觀入口網站 — BTP 版](#)
- [SAP 認證存放區 — BTP 版本](#)

SAP 景觀入口網站 — BTP 版

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

SAP 景觀入口網站是唯一支援在 SAP BTP 環境中安裝附加元件的機制。確保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 認證存放區 — BTP 版本

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

在開發人員預覽中，秘密存取金鑰驗證是針對 SAP ABAP-BTP 版本驗證 AWS SDK 的唯一支援機制。SDK 會從認證存放區讀取認證，並安全地儲存秘密存取金鑰。

您必須符合下列先決條件。

- 認證存放區的訂閱。
- 指派為 BTP 子帳戶權利的認證存放區。如需詳細資訊，請參閱[初始設定](#)。
- 具有認證存放區標準計劃的服務執行個體。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 認證存放區](#)。

SAP 憑證存儲服務在 ABAP BTP 系統之外的 SAP BTP 中運行。如需詳細資訊，請參閱[SAP 認證存放區](#)。

安裝 適用於 SAP ABAP 的 AWS SDK

主題

- [下載適用於 SAP](#)
- [驗證軟體開發套件適用於 SAP ABAP 檔案 — 選用](#)
- [AWS 開發套件傳輸](#)

下載適用於 SAP

V1/release/abapsdk-LATEST.zip V1/release/abapsdk-LATEST.zip

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

下載完成後，我們建議您將下載的檔案解壓縮到目錄中，例如/tmp/awssdk。

驗證軟體開發套件適用於 SAP ABAP 檔案 — 選用

這個驗證 SDK 檔案簽章的選用步驟可協助您確認 SDK 未遭到竄改。請使用下列步驟來驗證您的 SDK 檔案。

1. 使用以下命令下載 SDK 簽名文件。

```
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-----
MIICIjANBgkqhkiG9w0BAQEFAAACAg8AMIICCgKCAgEAmS3oN3wKBh4HJ0Ga0tye
15RR5909nuw0Jx0vEDCT709wUrXS3mjgEw6b6hvr2dLdoFr+eH4ewT5bV16U3gDv
051sTdEJJpfLEWJJZZNK3v9fGWKyXgYe+ifmsPmf4lhNd2auzpvIy2UrlSYijCRB
BWZFW+Ux00kILz+8vCFSXMZ6Z0qtLI1ZFbGrn6A5adbwwzf0qkg9BUEZK0wB6TAi
ZTnkMdBZGCBM9K2MRKKMxtrixUn+TFcAYyh5pM9tUAb2q4XE5m7092UnZG7ur/QY1
1FSZwAhQmk8hUPgUaq00QRC6z3TRzIGK0A/DI0cUPJMzFR4LCxEJkgh4rkRaU9V2
07DthUpj8b7QcQai0pnMpBf3zWlgbjNmX0hB0Eprg8/nVRHspf3zuiscJ21MPkz0
cHOR31MNsMLzm+d/gVklT31R/JwAcFCkXTWvR8/V0WNGZZXdVUbefrfI/k7fP60B
bzUrI1N4poq16rc4Tk5Derg+wQ7r0WjXkXop2kiCMjbYo0o10kS/At64PLj pz8dH
Zg25o79U9EJ1n+lpqZ297Ks+Hoct0v2GPbeeh0s7+N0fRTy0r81EZIUURLPKLVQUw
otVRzNDgLOA7eA667N1megZFhCmqEwK9tXakZUHAcMzRPyhALc/HtmovxdStN9h1
JC4ex0GqstAv1fX5QaTbMSECAwEAAQ==
-----END PUBLIC KEY-----
```

3. 使用以下命令驗證下載的 SDK ZIP 文件。該命令要求openssl這是許多 Linux 發行版的一部分。

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

4. 驗證上述命令的輸出是否為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，請不要安裝 SDK 並聯繫 AWS Support。

AWS 開發套件傳輸

主題

- [目錄](#)
- [匯入中](#)
- [命名空間](#)

目錄

SAP ABAP 的 SDK 安裝是透過 ABAP 傳輸完成的。您必須將這些傳輸匯入您的開發或沙箱環境。

每個適用於 SAP ABAP 發行版本的 SDK 都會完全取代前一個版本。不需要套用增量傳輸。傳輸被捆綁在一個 ZIP 文件中。以下是 ZIP 檔案的結構。

```
transports/  
transports/core/  
transports/core/Knnnnnnn.AWS  
transports/core/Rnnnnnnn.AWS  
transports/tla1/  
transports/tla1/Knnnnnnn.AWS  
transports/tla1/Rnnnnnnn.AWS  
transports/tla2/  
transports/tla2/Knnnnnnn.AWS  
transports/tla2/Rnnnnnnn.AWS  
.  
.  
.
```

資料 transports 料夾包含 core 子資料夾。core 子資料夾包含核心執行階段傳輸和每個模組的子資料夾，以模組的三個字母縮寫命名。如需 TLA 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。根據 TMS 路由的組態，SDK 可能不會在匯入先前的系統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。

當您的項目準備好進入下一個階段時，AWS SDK 可以與單獨的傳輸一起導入，其中包含您自己的 Z 代碼和業務功能。如果您正在使用變更控制系統，例如 SAP 變更請求管理 (Charm)，請洽詢 Charm 管理員，以正確處理協力廠商傳輸。

匯入中

主題

- [關鍵指針](#)
- [匯入時間](#)

AWS SDK 傳輸與用戶端無關。核心傳輸是強制性的，其中包含 SDK 執行階段程式碼 AWS Security Token Service、的 API 以及適用於 Amazon 簡單儲存服務的 API。其餘的 SDK 模組會以個別的傳輸方式傳送。為了使系統中 SDK 的大小保持較小，每個 SDK 模塊都是可選的。如果您的業務邏輯需要，您可以稍後安裝其他模組。

例如，如果您想要針對 Amazon S3 使用 API Amazon Translate，並匯入 core 傳輸 (包含核心執行階段、Amazon S3 和 AWS STS 模組) 和 x18 傳輸 (包含用於傳輸的模組 Amazon Translate)。

若要查看適用於 SAP ABAP API 的完整開發套件清單，請參閱適用於 [SAP ABAP 的開發套件-API 參考指南](#)。

以下是匯入 AWS SDK 傳輸時的關鍵指標。

- 每項運輸均以 Knnnnnn.AWS 及 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 [匯入傳輸要求] > [選項] > [匯入選項] 中找到的 [略過無效元件版本]。
- 所有需要的運輸都可以同時導入。
- 如果單獨導入 core 運輸，則必須先導入運輸。
- 所有傳輸的發行級別必須相同。

匯入時間

AWS SDK 傳輸可能需要很多分鐘才能匯入。如果 STMS 顯示綠色 (RC=0) 或黃色 () 指示燈，則傳輸成功。RC=4

- 紅燈 (RC=8) 表示匯入發生語法錯誤。
 - 選取「請求 → 顯示 → 記錄」以檢查匯入錯誤。
 - 在匯入期間，如果因為缺少介面而擲回錯誤 IF_SYSTEM_UUID_RFC4122_STATIC，請確定 SAP Note 2619546 已套用至系統。如需詳細資訊，請參閱 [附註](#)。
 - 如果錯誤的原因不明，請聯繫 AWS Support。
- 紅色閃電 (RC=12) 表示傳輸檔案尚未正確載入 /usr/sap/trans 或沒有必要的權限。

关键指针

以下是匯入 AWS SDK 傳輸時的關鍵指標。

- 每項運輸均以 Knnnnnn.AWS 及 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 [匯入傳輸要求] > [選項] > [匯入選項] 中找到的 [略過無效元件版本]。
- 所有需要的運輸都可以同時導入。
- 如果單獨導入運輸，則必須先導入運輸。
- 所有傳輸的發行級別必須相同。

匯入時間

AWS SDK 傳輸可能需要很多分鐘才能匯入。如果 STMS 顯示綠色 (RC=0) 或黃色 () 指示燈，則傳輸成功。RC=4

- 紅燈 (RC=8) 表示匯入有語法錯誤。
 - 選取「請求 → 顯示 → 記錄」以檢查匯入錯誤。
 - 在匯入過程中，如果由於缺少介面而引發錯誤 IF_SYSTEM_UUID_RFC4122_STATIC，請確定 SAP Note 2619546 已套用至系統。如需詳細資訊，請參閱[附註](#)。
 - 如果錯誤的原因不明，請聯繫 AWS Support。
- 紅色閃電 (RC=12) 表示傳輸檔案尚未正確載入 /usr/sap/trans 或沒有必要的權限。

命名空間

適用於 SAP ABAP 的 SDK 會使用 /AWS1/命名空間，並且不會修改 SAP 物件或系統中的任何其他物件，但下列情況例外。

- AWS auth 對象位於 Auth 對象類中。Auth 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 使用身份驗證對象類是 YAW1。如果您 YAW1 在事務中已經有一個 auth 對象類 SU21，請在安裝 AWS Support 之前聯繫。

安 AWS SDK裝 SAP ABAP-BTP 版本

該BTP版本處於開發人員預覽中，可以通過加入預覽來安裝。要安裝SDK，填寫參與表格 [SAPABAP-BTP 版開發者預覽](#)。AWS SDK

在安裝 SAP ABAP-BTP 版本之前，請確保您符合必要的必要條件。如需詳細資訊，請參閱[SAP橫向入口網站](#)和[SAP認證存放區](#)。

主題

- [安SDK裝 SAP ABAP-BTP 版本](#)
- [模組](#)
- [修SDK補 SAP ABAP-BTP 版本](#)

安SDK裝 SAP ABAP-BTP 版本

1. 轉到您的SAP景觀門戶實例，並啟動部署產品視窗應用程序。
2. 在產品中，選取合作夥伴產品/**AWS1/SDK_OMNI**下方的。

如 AWS Support 果您在開發人員預覽中被接受/**AWS1/SDK_OMNI**後看不到，請與我們聯繫。

3. 在「目標版本」中，選擇您要在系統上安裝的BTP版本 SAP ABAP-版本。SDK
4. 在可用系統中，勾選您要安裝SIDs的所有項目的核取方塊SDK。
5. 選取部署，輸入排程詳細資料，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，包括系統停機時間。如需詳細資訊，請參閱[部署產品](#)。

模組

以下模塊包含在 AWS SDK為 SAP ABAP-BTP 版本的開發人員預覽中。

- [Amazon API 網關 \[agw\]](#)
- [Amazon Athena \[ath\]](#)
- [Amazon 基岩運行時 \[\] bdr](#)
- [Amazon Comprehend \[\] cpd](#)
- [Amazon EventBridge \[evb\]](#)
- [Amazon Forecast \[fcs\]](#)

- [Amazon Kinesis \[\] kns](#)
- [Amazon 數據 Firehose \[\] frh](#)
- [Amazon SageMaker \[sgm\]](#)
- [Amazon 簡單通知服務 \[sns\]](#)
- [Amazon 簡單隊列服務 \[sqs\]](#)
- [Amazon 簡單存儲服務 \[s3\]](#)
- [AWS Systems Manager \[ssm\]](#)
- [Amazon Textract 品 \[\] tex](#)
- [Amazon Transcribe \[\] tnb](#)
- [Amazon Translate \[x18\]](#)
- [AWS CloudTrail \[tr1\]](#)
- [AWS IoT \[iot\]](#)
- [AWS KMS \[kms\]](#)
- [AWS Lambda \[lmd\]](#)
- [AWS Secrets Manager \[smr\]](#)
- [AWS Security Token Service \[sts\]](#)
- [AWS Transfer Family \[trn\]](#)
- [IAM任何地方的角色 \[rla\]](#)
- [Amazon Redshift 數據 API \[\] rsd](#)

修SDK補 SAP ABAP-BTP 版本

對於 SAP ABAP-BTP 版本SDK的修補程序與安裝程序類似。如果您SDK在已安裝舊版本的系統上安裝，則會將其修補至您選擇的新版本。SDK

配置 適用於 SAP ABAP 的 AWS SDK

在使用之前 適用於 SAP ABAP 的 AWS SDK，您必須使 SDK 用操作所需的技術和功能設置進 SDK 行配置。某些設置是可傳輸的，有些是運行時設置。許多設定與在其他 .INI 檔案中定義的設定直接類似。SDKs

除了執行階段設定外，SDK 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，將組態傳輸至 QA 和生產環境。不建議在生產環境中使用可傳輸的組態。

如果您沒有設定權限 AWS SDK，請參閱[SAP 授權](#)。

配置 適用於 SAP ABAP 的 AWS SDK

若要執行組態交易，請 /n/AWS1/IMG 在 SAPGUI 命令列中輸入。

配置 AWS SDK SAP ABAP-BTP 版本

使用以下步驟配置 SDK SAP ABAP-BTP 版本。

1. 在網頁瀏覽器中開啟您的 ABAP 環境。
2. 導航到自定義業務配置應用程序。

若要使用「匯出自訂傳輸」應用程式建立自訂請求，請參閱[在匯出自訂傳輸應用程式-建立請求中工作](#)。

在「自訂業務組態」應用程式中，您可以根據 SDK 設定類型來分組組態。使用下列步驟將組態分組。

1. 在網頁瀏覽器中開啟您的 ABAP 環境，然後瀏覽至「自訂商務組態」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. 組態現在可在階層式結構中使用，如影像中所示。欲儲存檢視，請參閱[檢視 \(變體管理\)-元件](#)。

Custom Business Configurations (4)

Name	Description	
Application Configuration		
SDK Profile	Maintain AWS SDK Profile	>
Logical Resource Resolver	Maintain Logical Resource Resolution	>
Global Settings		
Technical Settings	Maintain Technical Settings	>
Configure Scenarios	Configure Scenarios	>

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

針對 SAP ABAP 的 AWS SDK 使用 /n/AWS1/IMG IMG 事務，以及適用於 SAP ABAP 的 AWS SDK 的自訂業務組態應用程式-BTP 版本來配置全域設定。本主題可互換使用 IMG 和自訂業務組態。

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 管理員進行設定。您可以將這些值設定為下列建議的設定。

- 選取「新項目」。
 - S3 區域化：[使用存取 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：將此欄位保持空白。此欄位在 BTP 版本中是唯讀的，預設設定為「是」。
 - 中繼資料結束模式：使用 IPv4 中繼資料端點。此欄位在 BTP 版本中是唯讀的，且會自動更新。
 - 中繼資料結束 URL：將此欄位保留空白。此欄位在 BTP 版本中是唯讀的。
- 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「新項目」。
 - 案例識別碼：DEFAULT
 - 案例摘要：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獨特案例，則可以設定多個案例。

- DEFAULT-標準操作。
- DR-如果發生災難需要將整個系統移動到另一個區域，則需要特殊配置。
- DR_TEST-模擬災難的特殊配置，例如，在生產的臨時克隆中。

應用程式組態

為 SAP ABAP 配置 SDK 類似於配置其他基於 ABAP 的應用程序。它被組織成不同的配置文件，以分組各種場景的設置。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 ZVA01ZVA02、和已 ZVA03 在發票相關交易 (例如 Amazon S3 AWS 服務、和) 上執行 AWS

Lambda，則ZINVOICE可以建立名為的 SDK 設定檔。Amazon SageMaker此設定檔可將發票相關功能的技術設定、SAP 授權和 IAM 角色對應分組。

使用適用於 SAP ABAP 的 AWS SDK /n/AWS1/IMG 交易，以及適用於 SAP ABAP 的自訂業務組態應用 AWS 程式-BTP 版本的自訂業務組態應用程式來設定全域設定。本主題可互換使用 IMG 和自訂業務組態。

主題

- [SDK 設定檔](#)
- [邏輯資源解析器](#)
- [範例](#)

SDK 設定檔

ABAP SDK 設定檔會針對每個 SID 和用戶端定義下列項目。

Note

客戶端始終是 100 在 SAP BTP，ABAP 環境。

- 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 資源很可能也位於相同的「區域」中，而且這應該是您的預設「區域」。您的 ABAP 程式碼可以覆寫預設的區域。
- 身分驗證方法
 - 對於在 Amazon EC2 上執行的 SAP 系統，我們強烈建議您選擇執行個體角色中繼資料，以便從短期的自動輪換登入資料中受益。
 - 對於在內部部署或其他雲端中執行的 SAP 系統，您必須從 SSF 儲存區選擇認證。
 - 對於在 SAP BTP 上執行的 ABAP 系統，您必須從 SAP 認證存放區中選擇認證。如需詳細資訊，請參閱[使用 SAP 認證存放區進行驗證](#)。
- 邏輯 IAM 角色與 IAM 角色的對應。
 - 此對映會依遞減優先順序排序。
 - 系統會自動為使用者選取具有最高優先順序的 IAM 角色，而該使用者在 PFCG 角色中獲得授權。

Note

PFGC 角色被稱為在 SAP BTP，ABAP 環境中的業務角色。

當一個 ABAP 程序想要連接到一個 AWS 服務，它會指定一個 ABAP SDK 配置文件，提取必要的設置。AUTHORIZATION-CHECK將執行以確認使用者具有存取 SDK 設定檔的權限。您的 SAP 安全管理員可以定義 PFGC 角色，以授與您存取適當的使用者。

邏輯資源解析器

邏輯資源解析器使您可以使用標準位置來存儲資源名稱。它附帶了 SDK 的 SAP ABAP。其動作類似於 FILE 交易將邏輯檔名對應至實體檔案名稱的方式。

邏輯資源定義了資源的概念，例如保存我們發票的 Amazon S3 儲存貯體。例如，可以命名此邏輯資源，ZINVOICES_OUTBOUND並且可以對應至不同的實體值區名稱，具體取決於 SAP 系統是開發、QA 還是生產環境。

SAP ABAP SDK 的設定可讓 QA 系統將邏輯資源解析為 QA 實體資源，即使從生產環境中重新整理系統也是如此。所有系統的資源對應是在您的開發 SAP 系統中定義並向前傳輸。這種方法與 SAP 系統中的通常設置不同，其中映射作為主數據處理並在每個系統中進行設置。SDK 針對 SAP ABAP 提供的邏輯資源解析程式的優點在於，系統重新整理後發生錯誤傳輸的可能性幾乎沒有。

範例

有四個獨立的 Amazon S3 儲存貯體-每個儲存貯體用於開發、生產和品質保證，以及第二個用於回歸測試的 QA 儲存貯體。

當 SDK 將邏輯資源解析 ZINVOICE_OUTBOUND 為實體資源時，它會檢查 SY-SYSID 並 SY-MANDT 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。

如果生產環境中的資源對應需要變更，您必須變更開發系統中 IMG 的對應，並將其傳送至下一步。如此可確保將 AWS 資源重新指派給 SAP 系統時，會受到變更控制，就像其他任何傳輸一樣。

Note

由於 SDK 組態與用戶端相關，因此重新指派資源會在自訂要求中傳輸，而且必須將傳輸匯入每個用戶端。

執行期設定

本節涵蓋下列主題。

Note

這些設定無法傳輸，而且是每個 SAP 系統的本機設定。

主題

- [記錄和追蹤](#)
- [選擇加入：增強型遙測](#)
- [作用中案例](#)

記錄和追蹤

您可以啟動追蹤以進行偵錯。除非診斷技術問題，否則建議將跟踪級別保持在「無痕」。如需詳細資訊，請參閱安全作業。

這些設定不適用於適用於 BTP 版本的 SDK。

選擇加入：增強型遙測

所有 SDK 都會將遙測資訊傳送至 AWS 支援目的。您可以選擇加入增強型遙測。當您聯繫 AWS Support 以識別特定 API 調用的來源時，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 BTP 版本的 SDK。

作用中案例

在此交易中激活您的DEFAULT方案。每個系統只需要啟動一次，除非系統正在進行多區域災難復原，否則不應變更。在多區域設定中，您可以使用此設定將 SAP 系統切換至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AWS 服務過對AWS端點進行 HTTPS 呼叫而消耗。一般而言，AWS端點可透過網際網路存取。SAP 系統必須能夠與網際網路連線，才能建立這些輸出連線。適用於 SAP ABAP 的 SDK 永遠不需要從互聯網到 SAP 系統的入站連接。

下列案例提供建立輸出連線的不同方法。

案例

- [透過代理伺服器連線](#)
- [通過數據包檢查防火牆進行連接](#)
- [閘道端點](#)
- [自訂介面端點](#)
- [存取多個區域中的端點](#)

透過代理伺服器連線

若要透過 Proxy 伺服器建立連線，請使用下列步驟。

1. 在 SDK 中，移至 [交易] **SICF**。
2. 選擇 Execute (執行)。
3. 在功能表中，選擇「用戶端 > 代理伺服器」。
4. 將代理伺服器設定為作用中。
5. 在「下列位址的無 Proxy」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，並且不需要任何設定即可使用 SAP 系統的 Proxy 伺服器組態。

Note

如果您使用 [Amazon EC2 執行個體中繼資料身份驗證](#)，則 SAP 系統無法在使用代理伺服器存取本機執行個體中繼資料 `http://169.254.169.254`。您必須在 `169.254.169.254` 在下列位址的「無 Proxy」欄位中包含。

通過數據包檢查防火牆進行連接

您可以為輸出連線設定封包檢查防火牆。這些防火牆會解密 SSL 流量，然後再將其重新加密，然後再將其傳遞到端點。此配置通常需要防火牆將自己的證書發行給正在使用 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

有些提供 AWS 服務供閘道端點，可在沒有網際網路的情況下提供高性能存取 VPC。這些端點對於 SAP ABAP 的 SDK 來說是透明的，而且不需要任何設定。

如需詳細資訊，請參閱[閘道端點](#)。

自訂介面端點

如果您需要使用自訂端點覆寫預設端點解析度，可以使用介面端點為您的 VPC 提供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，這些端點會有自己的 DNS 位址，而 ABAP 程式必須明確覆寫通常的端點解析邏輯。如需詳細資訊，請參閱 AWS re:Post — [為什麼無法解析介面 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 系統未使用私有 DNS，並使用自訂端點呼叫服務。中定義的邏輯資源 /AWS1/IMG 代表實體介面端點位址，例如 `vpce-0123456789abcdef-hd52vxz.translate.us-west-2.vpce.amazonaws.com`。這樣可以避免在代碼中對 DNS 進行硬編碼。

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。然後將它們提供給 AWS 會話類的工廠方法（用 AWS STS 於承擔 IAM 角色）並翻譯 API 類。

```
" 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，任何方法都go_xl8會呼叫端點https://vpce-0123456789abcdefg-12345567.translate.us-west-2.vpce.amazonaws.com。

存取多個區域中的端點

AWS端點是根據 SDK 配置文件中定義的默認值AWS 區域自動確定。您也可以透過程式設計方式指定區域，覆寫預設區域。這可以在工廠CREATE()方法中覆蓋，或者稍後使用 SDK 的配置對象覆蓋。如需詳細資訊，請參閱[程式設定](#)。

在下列範例中，工廠CREATE()方法用於設定區域，並在和us-east-1區us-west-2域中列出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

Base 系統管理員有時需要從用戶端控制 SDK 整個系統中的某些功能。對於代表客戶在自己的 AWS 帳戶中操作系統的託管和服務提供商來說，這是一種常見的情況。AWS SDK 為 SAP ABAP 支援「服務提供者」設定。這些設定是在用戶端中設定的，並會影響所有用戶 SDK 端的設定。BTP 版本中 SDK 不支援服務提供者設定。SAP ABAP

服務提供者設定是在交易中設定的 /AWS1/IMG，且必須在用戶端中進行設定。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上生效，並在發生衝突時取代其他 IMG 設定。

請使用下列步驟來設定用戶端中的「服務提供者」設定。

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 SDK 中 EC2 繼資料 — 即使設定 SDK 檔設定為使用 EC2 執行個體中繼資料進行驗證，也可防止存取所有用戶端中繼資料中的 EC2 執行個體。如果 ABAP 程式嘗試使用存取執行個體中繼資料，則 SDK 會引發例外狀況 SDK。
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

系統重新整理之後，Basis 管理員的主要挑戰是確保不同的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 為這項挑戰提供了一種安全意識的邏輯資源方法。業務分析師可以採取以下步驟。

1. 定義邏輯資源，例如ZINVOICE_OUTBOUND。
2. 對應開發系統中的所有系統和用戶端。
3. 向前傳輸所有系統的配置，直到生產環境。

重新整理後的基礎步驟

1. 檢查驗證

- 如果系統使用秘密存取金鑰驗證，則 SSF 加密的認證將無效，因為它們儲存在主資料中。[必須重新輸入認證，這可能需要在 <https://console.aws.amazon.com/iam/> 中重新產生新的秘密存取金鑰。](#)
- 如果系統正在使用 EC2 執行個體中繼資料進行驗證，則不需要執行任何步驟。

檢查追蹤設定

- 在中/AWS1/IMG，確定追蹤設定是您想要的。這些設定無法傳輸。

追蹤

追蹤輸出在 IMG 執行階段設定中進行控制。

您可以使用的跟踪級別是：

- 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。

如果啟動了 API 追蹤，則會將追蹤寫入aws1_trace-YYYY-MM-DD.log檔案DIR_WORK中。

如果另外激活有效負載跟踪，則會為每個呼叫和有效負載組件創建帶有標題aws1_payload_*的其他文件。有效負載追蹤長度可以限制套用至每個個別有效負載追蹤失敗的長度限制。

有效負載追蹤主要用來收集發生序列化錯誤時要 AWS Support 提供的資訊。除非您嘗試診斷 SDK 錯誤，否則建議您選擇「無追蹤」。

Note

有效負載跟踪可以包含未加密的業務信息。我們建議您僅針對 Support 援要求開啟這些追蹤，以協 AWS 助您進行疑難排解。您可以在解析度後關閉這些軌跡。追蹤不會自動刪除，而且需要在不再需要時由系統管理員移除。

這些設定不適用於適用於 BTP 版本的 SDK。

遙測

SDK 會將遙測資訊傳送至 AWS Support. 適用於 SAP ABAP 的 SDK 會收集下列資訊：

- OS 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

您可以選擇將下列資訊傳送至 AWS Support。

- 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)

其他信息可 AWS Support 以幫助您更好。AWS Support 可以檢測為什麼進行某個 API 調用，並可以在 SAP 系統中進一步找到相關交易。

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

使用 適用於 SAP ABAP 的 AWS SDK

SDK 對於 SAP ABAP 有兩個主要組成部分。

- SDK 運行時 (包/AWS1/RT) -一組支持安全性，身份驗證，跟踪，配置，數據轉換和其他跨 API 功能的對象。適用於 Amazon S3 的 API 模組 AWS STS、IAM 角色任何地方和 Secrets Manager 都是強制性的。
- API (包/AWS1/API及其子包) -每個 API 的子包，其中每個 API 的對象彼此完全獨立，確保一個 API 中的更改不會破壞另一個 API。要查看 API 的完整列 適用於 SAP ABAP 的 AWS SDK 表，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

- [在 ABAP 數據的表示](#)
- [亞馬遜 S3 示例程序](#)
- [適用於 SAP ABAP 概念的開發套件](#)
- [適用於 SAP ABAP 的 AWS SDK features](#)
- [使用 SDK 建置產品](#)
- [限制](#)

在 ABAP 數據的表示

本節涵蓋下列主題。

主題

- [資料類型](#)
- [AWS 資料類型](#)

資料類型

AWS 服務有一組標準的數據類型，必須映射到 ABAP 數據類型。如需詳細資訊，請參閱下表。

AWS 資料類型	數據類型	說明
布林值	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 服務將截斷溢出，但這種行為沒有正式定義。
Short	INT2	
整數	INT4	
Long	十二月十九	INT8 在阿巴巴 750 之前不可用。DEC19 用於在所有支持的 ABAP 平台上實現兼容性和一致性。
Blob	XSTRING	代表二進制數據
Float	STRING	雖然 ABAP 支持十字浮點數，它不能表示值，如 NaN，無窮大和-無窮大。AWSSDK 將這些內部表示為字符串，並在運行時將它們轉換為 DECFLOAT16。如果 NaN，無限或 + 無窮大表示，開發人員可以處理這些響應一組特殊的異常或映射。
Double	STRING	
大整數	STRING	這些值表示不能在 ABAP 中表示的無限長度數字，並使用字符串代替 BigInteger。
大十進制	STRING	
時間戳記	TZNTSTMP5	TZNTSTMP5 使用本機 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	數據類型	說明
結構	類別	
UNION	類別	聯集與結構相同，不同之處在於聯集永遠不會有多個欄位集。所有其他字段將被設置為沒有值。
Array (陣列)	標準表	
雜湊	雜湊表	散列表將只有兩列：一個 KEY (字符串) 和一個 VALUE (類)。

AWS 資料類型

已整合下列方法以支援AWS 服務在阿巴巴。

- 某些AWS數據類型不能在 ABAP 來表示。例如，floatABAP 中的資料類型不支援NaN,Infinity，或-Infinity價值觀。因此，float數據類型表示為STRING並被翻譯為DECFLOAT16在運行時。
- AWS數據在線路上表示為 JSON 或 XML，並且值是可選的。例如，請參閱以下由AWS 服務在JSON 中。

```
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 不區分長度為 0 的字符串和沒有值的字符串。其他語言可能會為字符串分配 NULL 值，或者將字符串包裝在構造中（例如 `JavaOptional<>` 包裝）。這些在 ABAP 中不受支援。因此，適用於 SAP ABAP 的 SDK 透過提供的變體來促進值的區別吸氣方法。

亞馬遜 S3 示例程序

本節將引導您完成一個簡單的示例程序，以列出 Amazon S3 存儲桶的內容 `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。

- 你有一個亞馬遜 S3 存儲桶。在本教學課程中，值區會命名為 `demo-invoices.customer.com`。
- 交易/AWS1/IMG:
 - 具有名為的已定義 SDK 設定檔 `DEMO_S3`。
 - 在 SDK 設定檔中，邏輯 IAM 角色 `TESTUSER` 必須對應至 IAM 角色，例如 `arn:aws:iam::111122223333:role/SapDemoFinance` 其中授予 `s3:ListBucket` 允許列出您的亞馬遜 S3 存儲桶的內容。
 - 具有名為的邏輯資源 `DEMO_BUCKET` 它使用 SAP 系統的 SID 和客戶端映射到您的亞馬遜 S3 存儲桶。
- 您的使用者具有下列項目的 PFCG 角色：
 - 授權使用者存取 `DEMO_S3` 通過驗證對象的 SDK 配置文件 `-/AWS1/SESS`。
 - 授權使用者使用邏輯 IAM 角色 `TESTUSER` 通過身份驗證對象訪問 `-/AWS1/LROL`。
- 您的 SAP 系統可以自行驗證 AWS 使用 SDK 配置文件中定義的方法。

- 您的亞馬遜 EC2 執行個體設定檔授予您的 SAP 系統的權利sts:assumeRole在 IAM 角色中arn:aws:iam::111122223333:role/SapDemoFinance映射在 SDK 配置文件中。

代碼

下面的代碼塊演示了你的代碼將是什麼樣子。

```
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( ).
    ENDLLOOP.
  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

以下是對部分代碼的審查。

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

使用者無法指定實體值區名稱。他們指定了一個邏輯存儲桶和系統管理員（特別是企業分析師）與 AWS 管理員將邏輯值區對應至實體值區/AWS1/IMG。在大多數商務案例中，使用者沒有機會選擇邏輯值區 — 邏輯資源 ID 會在程式碼中硬式編碼，或在自訂組態表中進行配置。

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

這一行建立了一個安全會話，並聲明這個 ABAP 程序希望使用 DEMO_S3 SDK 設定檔。此呼叫是 SDK 設定的連線，並提取預設區域、驗證設定和所需的 IAM 角色。一個電話 AUTHORIZATION-CHECK 自動進行，以確保授權對象/AWS1/SESS 很滿意。此外，AUTHORIZATION-CHECK 將根據授權對象進行調用以確定用戶獲得授權的最強大（序列號較低的序列號）邏輯 IAM 角色/AWS1/LROL。SDK 會假設 IAM 角色已對應至 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動追蹤 IMG。

如果使用者未獲得要求的 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 ).
```

這一行將邏輯資源解析為實體值區名稱。如果因為組態沒有此 SID/用戶端組合的對應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 ).
```

這一行使用亞馬遜 S3 創建一個 API 對象 create() 的方法 /aws1/cl_s3_factory。返回的對象是類型 /aws1/if_s3 這是一個亞馬遜 S3 API 的接口。必須為每個服務建立個別的 API 物件。例如，如果一個 ABAP 程序正在使用亞馬遜 S3，AWS Lambda，然後它會從中建立 API 物件 /aws1/cl_s3_factory, /aws1/cl_lmd_factory，以及 /aws1/cl_dyn_factory。

構造函數有一些可選參數，如果要覆蓋配置的默認區域，則可以指定 RegionIMG。通過這種方式，可以有兩個實例 /aws1/if_s3，一個用於 us-east-1 和一個用於 us-west-2，如果您要將物件從一個「區域」中的值區複製到另一個「區域」中的值區。同樣地，您可以建立兩個不同的安全性工作階段物件，並使用它們建立兩個獨立的執行個體 /aws1/cl_s3，如果您需要從財務相關值區讀取報告，並將物件寫入物件至物流相關值區。

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

這條線是一個電話ListObjectsV2。它需要簡單的輸入參數並返回一個對象。這些對象可以代表深層JSON和XML數據，反序列化為ABAP面向對象的構造。在某些情況下，它可能非常複雜。現在，您只需要處理輸出即可列出存儲桶的內容。

```
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.
```

數據是使用GET...()樣式方法，隱藏數據的內部表示。GET_CONTENTS()返回一個ABAP表，每一行本身包含一個代表單個Amazon S3條目的對象。在大多數情況下，AWS SDK採用這種面向對象的方法和所有數據表示為對象和表。該LastModified字段被表示為可以轉換為具有ABAP原生日期的時間戳CONVERT TIME STAMP指令。GET_SIZE()返回一個INT4用於簡單的數學和格式化操作。

```
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 — 連接，4xx 客戶端，5xx 服務器或任何 ABAP 錯誤，如授權或配置錯誤，都表示為異常。您可以單獨處理每個異常。您可以選擇是否應將例外狀況當做資訊性錯誤、重試、警告、短暫傾印或任何其他類型的處理來處理。

適用於 SAP ABAP 概念的開發套件

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](#)
- [陣列](#)
- [地圖](#)
- [更高級別的函數](#)

API 類別

每個 AWS 服務 被分配一個三個字母的縮寫或 TLA。該服務由 /AWS1/IF_<TLA> 格式的接口表示。我們將其稱為服務接口。API 類別位於 /AWS1/API_<TLA> 封裝中。服務接口由每個 AWS 操作一個方法 (我們將調用這些方法操作)。若要查看 適用於 SAP ABAP 的 AWS SDK TLA 的完整模組清單，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 參數和最多一個 RETURNING 參數。通常，這些參數將是具有複雜構造函數和一長串 GET...() 方法的對象。在許多情況下，物件會包含巢狀物件、遞迴參考、物件表格、資料表等。這 AWS 服務 是因為傳遞了深度的 XML 和 JSON 結構，這些結構不能由一組平面的參數來表示。

該 RETURNING 參數始終是一個類，即使該類只包含一個屬性。

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。
- 出現在表中的任何原始數據類型的類。例如，如果服務傳回字串資料表，ABAP API 會將它表示為物件表格，其中每個物件都是封裝字串的包裝類別。這是為了使包裝類可以隱藏表示不能在 ABAP 中本地表示的空字串的詳細信息。
- 服務所定義之任何特定錯誤的例外狀況類別。
- 每個基本數據類型的數據元素。每個數據類型都有自己的數據元素，以便是自我記錄。
- 用於內部處理的其他物件，例如用於序列化和反序列化 XML 和 JSON 承載的 XSLT 轉換。

結構類

由服務發送和接收的大多 AWS 數據由 AWS SDK 表示為類。這些類表示數據的結構和隱藏存儲的內部細節。特別是，類隱藏 SDK 表示此字段沒有值的方式。

對於結構類別中的每個欄位，有三種方法。

GET_field()

該 GET_field() 方法

- 返回字段的值，或
- 如果該字段沒有值，則返回一個默認值，您可以將其設置為可選參數。

例如，請考慮下列列印值區位置限制的程式碼。

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

如果存儲桶根本沒有位置限制 (如 us-east-1)，則 GET_LOCATIONCONSTRAINT() 將返回空字符串。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' ).
```

現在程序將寫，Bucket Location: assuming us-east-1 如果 getbucketlocation() 的結果沒有返回一個位置。

如果要求的值完全遺失，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' ).
```

在這種情況下，如果沒有位置約束，`GET_LOCATIONCONSTRAINT()`將返回`NopeNopeNope`。

HAS_field()

`HAS_field()`方法是找出該字段是否具有值的方法。請參閱以下範例。

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

如果某個字段已知始終具有值，則不會有任何`HAS_field()`方法。

ASK_field()

該`ASK_field()`方法返回字段的值或引發異常，如果它沒有值。這是處理許多字段的便捷方法，如果任何字段沒有值，則可以從邏輯中保釋並採取不同的方法。

```
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，這/`AWS1/CX_RT_VALUE_MISSING`是一個靜態異常，如果您選擇不 `catch` 獲它，則會收到警告。

最佳實務

通常，您可以使用該`GET_field()`方法，因為它將空字符串視為空字符串，並且是三個選項中最類似 ABAP 的方法。但是，它不能讓您輕鬆區分字段具有空值的情況以及字段沒有值的情況。如果您的業務邏輯依賴於區分缺少的數據與空白數據，那麼`HAS`或`ASK`方法可以讓您處理這些情況。

陣列

數組被表示為對象的 ABAP 標準表。

JSON 陣列可以包含空值，例如下列陣列：`['cat', 'dog', null, 'horse']`。這稱為稀疏陣列。它在 ABAP 中表示為對象引用的內部表，並且該`null`值在表中表示為一個真正的 ABAP `null` 值。重複執行稀疏資料表時，您必須檢查`null`值，以避免存取`null`物件並取得例外狀況。`CX_SY_REF_IS_INITIAL`實際上，稀疏陣列在 AWS 服務中很少見。

要初始化對象的數組，它是方便使用新的 ABAP 7.40 構造。請考慮此次推出的 Amazon EC2 執行個體，其中指派了多個安全群組：

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

地圖

JSON 映射在 ABAP 中表示為 Hashed Tables 其中每個表行只有兩個組件。

- KEY—一個字符串，它是表 UNIQUE KEY 的。
- VALUE—包含值的物件。

地圖是 AWS SDK 使用真實結構而不是類的極少數情況之一。這是必要的，因為 ABAP 哈希表不能將對象引用作為鍵字段，並且 AWS 映射鍵始終是非空字符串。

更高級別的函數

上一節中 [API 類別](#) 所描述的精確鏡像 AWS 服務 API，並將這些 API 表示為熟悉的 ABAP 類別。在某些情況下，SDK 還包括建立在 API 類別之上的更高層級函數，以簡化某些操作。包含更高級別的函數，以方便程序員使用，並且不會替換較低級別的 API 類。

如果 SDK 包含模塊的更高級別的功能，它們將包含在相同的傳輸中，並且可以通過名為的工廠類進行訪問 /AWS1/CL_TLA_L2_FACTORY。工廠類包括用於為模塊創建各種更高級別客戶端的方法，這些客戶端與 [API 文檔](#) 一起記錄了其餘部分。

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](#)

程式設計組態

使用適用於 AWS SDK SAP 的 `/n/AWS1/IMG IMG` 傳輸 ABAP，以及適用於 AWS SDK 適用於 - SAP ABAP BTP 版本的自訂業務組態應用程式進程式設計組態。

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` 路徑樣式來解決名為 foobucket 的儲存貯體。您可以使用下列範例命令強制執行路徑樣式的使用。

```
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

使用非同步時 AWS APIs，您需要等待特定資源可用，才能採取進一步動作。例如，`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 `ACTIVE` 之後，您才能啟動讀取或寫入操作。等待者可讓您在對 AWS 資源執行動作之前，確認資源處於特定狀態。

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 資源可能耗時且容易出錯。等候者會負責代表您執行輪詢，協助簡化此複雜性。

請參閱下列使用服務生的 Amazon S3 範例。

```
DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pf1 ).
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。執行 `get_waiter()` 命令以指定的時間 `bucketexists`。
- 您必須為每個服務生指定 `iv_max_wait_time` 參數。它代表服務生在完成之前必須等待的總時間。在上述範例中，服務生可以執行 200 秒。
- 您可能需要為必要的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` 例外狀況表示服務生超過 `iv_max_wait_time` 參數中指定的時間上限。
- `/AWS1/CX_RT_WAITER_TIMEOUT` 例外狀況表示服務生因為未達到所需的狀態而停止。

分頁程式

有些 AWS 服務 操作提供分頁回應。它們會分頁，在每個回應中傳回固定量的資料。您需要使用權杖或標記提出後續請求，才能擷取整個結果集。例如，Amazon S3 ListObjectsV2 操作一次最多可傳回 1,000 個物件。您必須使用適當的權杖提出後續請求，以取得下一頁的結果。

Pagination 是傳送連續請求的程序，以從先前請求停止的位置提取。Paginator 是 SDK 為 SAP 提供結果的迭代器 ABAP。您可以 APIs 輕鬆使用分頁，而無需了解 API 使用分頁權杖的基礎機制。

使用分頁器

您可以使用傳回分頁器物件 `get_paginator()` 的方法建立分頁器。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的必要參數 API。此程序會傳回迭代器物件，可用於使用 `has_next()` 和 `get_next()` 方法，對分頁結果進行迭代。

- `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 paginator 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

SDK 的 SAPABAP 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，也就是服務組態物件中的 AV_MAX_ATTEMPTS 屬性在失敗和提出例外狀況之前 SDK 重試操作的次數。建立服務用戶端物件時，會將 AV_MAX_ATTEMPTS 屬性 SDK 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，請參閱中的 [AWS SDKs 重試行為](#) 和工具參考指南。

使用 SDK 建置產品

消耗的產品或 ABAP 附加元件 AWS 服務可以增強和擴充 SDK 的功能。您可以建置此類產品以搭配 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 只能包含字母、數字和底線，不得包含空格或特殊字元。您可以將其與產品的技術名稱或任何其他識別碼進行比對。如果您開發多個產品或附加元件，每項產品的產品 ID 必須是唯一的。例如，「商業發票分析器」、「稅金計算器」及「訂價引擎」產品的產品 ID 可以是 INVOICE_ANALYZERTAX_CALCULATOR、和 PRICING_ENGINE。

將產品 ID 新增至工作階段，可增強每次服務呼叫傳送至 AWS 的遙測功能。發出呼叫之物件的產品識別碼和命名空間包含在遙測中。使用此遙測，AWS Support 可以識別在客戶遇到 SDK 問題時進行呼叫的產品。它可以幫助澄清調用實際上是由產品進行的，而不是客戶的代碼。

限制

適用於 SAP ABAP 的 AWS SDK 包括適用於所有人的 SDK 模塊 AWS 服務。其中一些模組可能有限制，如此處所述。

- 依賴 MQTT 通訊協定繫結 (例如 iotevents) 的模組將無法運作。MQTT 不是以 HTTP 為基礎的通訊協定，目前不受支援。適用於 SAP ABAP 的 AWS SDK
- 依賴 HTTP/2 串流功能的模組尚不受支援。尚不支援與事件串流搭配使用的服務的某些操作，而且服務的媒體串流作業 (例如 Amazon Kinesis Video Streams) 將無法運作。

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有以下限制。

- 某些模組可能無法使用。
- 無法解除安裝。
- 它更新的頻率較低。

SDK 以取得SAPABAP程式碼範例

本主題中的程式碼範例會示範如何將 AWS SDK的 SAP ABAP 與 搭配使用 AWS。

基本知識是程式碼範例，示範如何在服務中執行基本操作。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會示範如何呼叫個別服務函數，但您可以在相關案例中查看內容中的動作。

案例是程式碼範例，示範如何透過呼叫服務內的多個函數或與其他 結合來完成特定任務 AWS 服務。

服務

- [使用 SDK 的 Amazon Bedrock Runtime 範例 SAP ABAP](#)
- [CloudWatch 使用 SDK 的範例 SAP ABAP](#)
- [使用 SDK 的 DynamoDB 範例 SAP ABAP](#)
- [使用 SDK 的 Amazon EC2範例 SAP ABAP](#)
- [使用 SDK 的 Kinesis 範例 SAP ABAP](#)
- [使用 SDK的 Lambda 範例 SAP ABAP](#)
- [使用 SDK 的 Amazon S3 範例 SAP ABAP](#)
- [SageMaker 使用 SDK 的 範例 SAP ABAP](#)
- [使用 SDK的 Amazon SNS範例 SAP ABAP](#)
- [使用 SDK 的 Amazon SQS範例 SAP ABAP](#)
- [使用 SDK 的 Amazon Textract 範例 SAP ABAP](#)
- [使用 SDK 的 Amazon Translate 範例 SAP ABAP](#)

使用 SDK 的 Amazon Bedrock Runtime 範例 SAP ABAP

下列程式碼範例示範如何使用 SAPABAP搭配 Amazon Bedrock Runtime 使用 AWS SDK來執行動作和實作常見案例。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執程式碼的指示。

主題

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

Anthropic Claude

InvokeModel

下列程式碼範例示範如何使用叫用模型 將文字訊息傳送至 Anthropic ClaudeAPI。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

叫用 Anthropic Claude 2 基礎模型來產生文字。此範例使用 功能 of /US2/CL_JSON，但在某些 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.

```

叫用 Anthropic Claude 2 基礎模型，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.

```

叫用 Anthropic Claude 3 基礎模型，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[InvokeModel](#)中的 AWS SDK 以取得SAPABAPAPI參考。

Stable Diffusion

InvokeModel

下列程式碼範例示範如何在 Amazon Bedrock 上叫用 Stability.ai Stable Diffusion XL 來產生映像。

SDK 適用於 SAP ABAP

 Note

還有更多。GitHub 尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

使用穩定擴散建立映像。

```
"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.

```

叫用 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[InvokeModel](#)中的 AWS SDK 以取得SAPABAPAPI參考。

CloudWatch 使用 SDK 的範例 SAP ABAP

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

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會示範如何呼叫個別服務函數，但您可以在相關案例中查看內容中的動作。

案例是程式碼範例，示範如何透過呼叫服務內的多個函數或與其他 結合來完成特定任務 AWS 服務。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執程式碼的指示。

主題

- [動作](#)
- [案例](#)

動作

DeleteAlarms

下列程式碼範例示範如何使用 DeleteAlarms。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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[DeleteAlarms](#)中的 AWS SDK 以取得SAPABAPAPI參考。

DescribeAlarms

下列程式碼範例示範如何使用 DescribeAlarms。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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[DescribeAlarms](#)中的 AWS SDK 以取得SAPABAPAPI參考。

DisableAlarmActions

下列程式碼範例示範如何使用 DisableAlarmActions。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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[DisableAlarmActions](#)中的 AWS SDK 以取得SAPABAPAPI參考。

EnableAlarmActions

下列程式碼範例示範如何使用 EnableAlarmActions。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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[EnableAlarmActions](#)中的 AWS SDK 以取得SAPABAPAPI參考。

ListMetrics

下列程式碼範例示範如何使用 ListMetrics。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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[ListMetrics](#)中的 AWS SDK 以取得SAPABAPAPI參考。

PutMetricAlarm

下列程式碼範例示範如何使用 PutMetricAlarm。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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[PutMetricAlarm](#)中的 AWS SDK 以取得SAPABAPAPI參考。

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。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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."
CREATE OBJECT lo_alarmname EXPORTING iv_value = iv_alarm_name.
INSERT lo_alarmname INTO TABLE lt_alarmnames.

"Disable alarm actions."

```

```

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

"Describe alarm using the same ABAP internal table."
TRY.
  oo_result = lo_cwt->describealarms(
    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 中的下列主題AWS SDK，以取得SAPABAPAPI參考。

- [DeleteAlarms](#)
- [DescribeAlarms](#)
- [DisableAlarmActions](#)
- [PutMetricAlarm](#)

使用 SDK 的 DynamoDB 範例 SAP ABAP

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

基本知識是程式碼範例，示範如何在 服務中執行基本操作。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會示範如何呼叫個別服務函數，但您可以在相關案例中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- [基本概念](#)
- [動作](#)

基本概念

了解基本知識

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。
- 從範例JSON檔案將影片資料寫入資料表。
- 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。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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 ZCL_AWS1_dyn_PUT_ITEM_OUTPUT
    MESSAGE '3 rows inserted into DynamoDB Table' && iv_table_name TYPE 'I'.
    CATCH /aws1/cx_dyncondalcheckfaile00.
    MESSAGE 'A condition specified in the operation could not be evaluated.'
TYPE 'E'.
    CATCH /aws1/cx_dynresourcenotfoundex.
    MESSAGE 'The table or index does not exist' TYPE 'E'.
    CATCH /aws1/cx_dyntransactconflictex.
    MESSAGE 'Another transaction is using the item' TYPE 'E'.
    ENDTRY.

" Get item from table.
TRY.
    DATA(lo_resp_getitem) = lo_dyn->getitem(
        iv_tablename          = iv_table_name
        it_key                 = VALUE /aws1/cl_dynattributevalue=>tt_key(
            ( VALUE /aws1/cl_dynattributevalue=>ts_key_maprow(
                key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s =
'Jaws' ) ) ) )
            ( VALUE /aws1/cl_dynattributevalue=>ts_key_maprow(
                key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n =
'1975' ) ) ) )
        ) ).
    DATA(lt_attr) = lo_resp_getitem->get_item( ).
    DATA(lo_title) = lt_attr[ key = 'title' ]-value.
    DATA(lo_year) = lt_attr[ key = 'year' ]-value.
    DATA(lo_rating) = lt_attr[ key = 'year' ]-value.
    MESSAGE 'Movie name is: ' && lo_title->get_s( ) TYPE 'I'.
    MESSAGE 'Movie year is: ' && lo_year->get_n( ) TYPE 'I'.
    MESSAGE 'Movie rating is: ' && lo_rating->get_n( ) TYPE 'I'.
    CATCH /aws1/cx_dynresourcenotfoundex.
    MESSAGE 'The table or index does not exist' TYPE 'E'.
    ENDTRY.

" Query item from table.
TRY.
    DATA(lt_attributelist) = VALUE /aws1/
cl_dynattributevalue=>tt_attributelist(
        ( NEW /aws1/cl_dynattributevalue( iv_n = '1975' ) ) ).
    DATA(lt_keyconditions) = VALUE /aws1/cl_dyncondition=>tt_keyconditions(
        ( VALUE /aws1/cl_dyncondition=>ts_keyconditions_maprow(

```

```

        key = 'year'
        value = NEW /aws1/cl_dyncondition(
            it_attributevalue = 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 中的下列主題AWS SDK，以取得SAPABAPAPI參考。
 - [BatchWriteItem](#)
 - [CreateTable](#)
 - [DeleteItem](#)
 - [DeleteTable](#)
 - [DescribeTable](#)
 - [GetItem](#)

- [PutItem](#)
- [查詢](#)
- [掃描](#)
- [UpdateItem](#)

動作

CreateTable

下列程式碼範例示範如何使用 CreateTable。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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[CreateTable](#)中的 AWS SDK 以取得SAPABAPAPI參考。

DeleteItem

下列程式碼範例示範如何使用 DeleteItem。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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[DeleteItem](#)中的 AWS SDK 以取得SAPABAPAPI參考。

DeleteTable

下列程式碼範例示範如何使用 DeleteTable。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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[DeleteTable](#)中的 AWS SDK 以取得SAPABAPAPI參考。

DescribeTable

下列程式碼範例示範如何使用 DescribeTable。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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[DescribeTable](#)中的 AWS SDK 以取得SAPABAPAPI參考。

GetItem

下列程式碼範例示範如何使用 GetItem。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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[GetItem](#)中的 AWS SDK 以取得SAPABAPAPI參考。

ListTables

下列程式碼範例示範如何使用 ListTables。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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( ).  
    ENDLLOOP.  
    DATA(lv_tablecount) = lines( oo_result->get_tablenames( ) ).  
    MESSAGE 'Found ' && lv_tablecount && ' tables' TYPE 'I'.  
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).  
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception->av_err_msg }|.  
        MESSAGE lv_error TYPE 'E'.  
ENDTRY.
```

- 如需API詳細資訊，請參閱[ListTables](#)中的 AWS SDK 以取得SAPABAPAPI參考。

PutItem

下列程式碼範例示範如何使用 PutItem。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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[PutItem](#)中的 AWS SDK 以取得SAPABAPAPI參考。

Query

下列程式碼範例示範如何使用 Query。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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 中的[查詢](#) AWS SDK以取得SAPABAPAPI參考。

Scan

下列程式碼範例示範如何使用 Scan。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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，請參閱在 [中掃描](#) AWS SDK以取得SAPABAPAPI參考。

UpdateItem

下列程式碼範例示範如何使用 UpdateItem。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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[UpdateItem](#)中的 AWS SDK 以取得SAPABAPAPI參考。

使用 SDK 的 Amazon EC2範例 SAP ABAP

下列程式碼範例示範如何使用 SAPABAP搭配 Amazon 使用 AWS SDK來執行動作和實作常見案例 EC2。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會示範如何呼叫個別服務函數，但您可以在相關案例中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執程式碼的指示。

主題

- [動作](#)

動作

AllocateAddress

下列程式碼範例示範如何使用 AllocateAddress。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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[AllocateAddress](#)中的 AWS SDK 以取得SAPABAPAPI參考。

AssociateAddress

下列程式碼範例示範如何使用 AssociateAddress。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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[AssociateAddress](#)中的 AWS SDK 以取得SAPABAPAPI參考。

CreateKeyPair

下列程式碼範例示範如何使用 CreateKeyPair。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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[CreateKeyPair](#)中的 AWS SDK 以取得SAPABAPAPI參考。

CreateSecurityGroup

下列程式碼範例示範如何使用 CreateSecurityGroup。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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( iv_groupname = iv_security_group_name " 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[CreateSecurityGroup](#)中的 AWS SDK 以取得SAPABAPAPI參考。

DeleteKeyPair

下列程式碼範例示範如何使用 DeleteKeyPair。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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[DeleteKeyPair](#)中的 AWS SDK 以取得SAPABAPAPI參考。

DeleteSecurityGroup

下列程式碼範例示範如何使用 DeleteSecurityGroup。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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 [DeleteSecurityGroup](#) 中的 AWS SDK 以取得 SAP ABAP API 參考。

DescribeAddresses

下列程式碼範例示範如何使用 DescribeAddresses。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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[DescribeAddresses](#)中的 AWS SDK 以取得SAPABAPAPI參考。

DescribeAvailabilityZones

下列程式碼範例示範如何使用 DescribeAvailabilityZones。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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[DescribeAvailabilityZones](#)中的 AWS SDK 以取得SAPABAPAPI參考。

DescribeInstances

下列程式碼範例示範如何使用 DescribeInstances。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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_instance_id    TYPE /aws1/ec2string,
      lv_status         TYPE /aws1/ec2instancename,
      lv_instance_type  TYPE /aws1/ec2instancetype,
      lv_image_id       TYPE /aws1/ec2string.
LOOP AT oo_result->get_reservations( ) INTO DATA(lo_reservation).
    LOOP AT lo_reservation->get_instances( ) INTO DATA(lo_instance).
        lv_instance_id = lo_instance->get_instanceid( ).
        lv_status = lo_instance->get_state( )->get_name( ).
        lv_instance_type = lo_instance->get_instancetype( ).
        lv_image_id = lo_instance->get_imageid( ).
    ENDLLOOP.
ENDLOOP.
MESSAGE 'Retrieved information about EC2 instances.' TYPE 'I'.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
    MESSAGE lv_error TYPE 'E'.
ENDTRY.

```

- 如需 API 詳細資訊，請參閱 [DescribeInstances](#) 中的 AWS SDK 以取得 SAP ABAP API 參考。

DescribeKeyPairs

下列程式碼範例示範如何使用 DescribeKeyPairs。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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[DescribeKeyPairs](#)中的 AWS SDK 以取得SAPABAPAPI參考。

DescribeRegions

下列程式碼範例示範如何使用 DescribeRegions。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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[DescribeRegions](#)中的 AWS SDK 以取得SAPABAPAPI參考。

DescribeSecurityGroups

下列程式碼範例示範如何使用 DescribeSecurityGroups。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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[DescribeSecurityGroups](#)中的 AWS SDK 以取得SAPABAPAPI參考。

MonitorInstances

下列程式碼範例示範如何使用 MonitorInstances。

SDK 適用於 SAP ABAP

 Note

還有更多。GitHub 尋找完整範例，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[MonitorInstances](#)中的 AWS SDK 以取得SAPABAPAPI參考。

RebootInstances

下列程式碼範例示範如何使用 RebootInstances。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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[RebootInstances](#)中的 AWS SDK 以取得SAPABAPAPI參考。

ReleaseAddress

下列程式碼範例示範如何使用 ReleaseAddress。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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[ReleaseAddress](#)中的 AWS SDK 以取得SAPABAPAPI參考。

RunInstances

下列程式碼範例示範如何使用 RunInstances。

SDK 適用於 SAP ABAP

 Note

還有更多。GitHub 尋找完整範例，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_tag specifications TYPE /aws1/
cl_ec2tag specifications=>tt_tag specifications list.
DATA ls_tag specifications LIKE LINE OF lt_tag specifications.
ls_tag specifications = NEW /aws1/cl_ec2tag specifications(
  iv_resource type = 'instance'
  it_tags = VALUE /aws1/cl_ec2tag=>tt_tag list(
    ( NEW /aws1/cl_ec2tag( iv_key = 'Name' iv_value = iv_tag_value ) )
  )
).
APPEND ls_tag specifications TO lt_tag specifications.

TRY.
  " Create/launch Amazon Elastic Compute Cloud (Amazon EC2) instance. "
  oo_result = lo_ec2->run_instances( " oo_result is
returned for testing purposes. "
  iv_image_id = iv_ami_id
  iv_instance type = 't2.micro'
  iv_max_count = 1
  iv_min_count = 1
  it_tag specifications = lt_tag specifications
  iv_subnet_id = 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 [RunInstances](#) 中的 AWS SDK 以取得 SAP ABAP API 參考。

StartInstances

下列程式碼範例示範如何使用 StartInstances。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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[StartInstances](#)中的 AWS SDK 以取得SAPABAPAPI參考。

StopInstances

下列程式碼範例示範如何使用 StopInstances。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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[StopInstances](#)中的 AWS SDK 以取得SAPABAPAPI參考。

使用 SDK 的 Kinesis 範例 SAP ABAP

下列程式碼範例示範如何使用 SAPABAP搭配 Kinesis 使用 AWS SDK來執行動作和實作常見案例。

基本知識是程式碼範例，示範如何在 服務中執行基本操作。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會示範如何呼叫個別服務函數，但您可以在相關案例中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- [基本概念](#)
- [動作](#)

基本概念

了解基本知識

以下程式碼範例顯示做法：

- 建立串流並將其放入記錄。
- 建立碎片迭代器。
- 讀取記錄，然後清除資源。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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_knslimitexceededx .
    MESSAGE 'The request processing has failed because of a limit exceeded
exception.' TYPE 'E'.
CATCH /aws1/cx_knsresourceinuseex .
    MESSAGE 'The request processing has failed because the resource is in use.'
TYPE 'E'.
ENDTRY.

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

```
ENDIF.
WAIT UP TO 10 SECONDS.
lo_stream_describe_result = lo_kns->describestream( iv_streamname =
iv_stream_name ).
lo_stream_description = lo_stream_describe_result->get_streamdescription( ).
ENDWHILE.

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

"Create a shard iterator in order to read the record."
TRY.
    lo_sharditerator = lo_kns->getsharditerator(
        iv_shardid = lo_record_result->get_shardid( )
        iv_sharditeratortype = iv_sharditeratortype
        iv_streamname = iv_stream_name
```

```

    ).
    MESSAGE 'Shard iterator created.' TYPE 'I'.
    CATCH /aws1/cx_knsinvalidargumentex.
    MESSAGE 'The specified argument was not valid.' TYPE 'E'.
    CATCH /aws1/cx_knsprovthruputexcdex .
    MESSAGE 'The request rate for the stream is too high, or the requested data
is too large for the available throughput.' TYPE 'E'.
    CATCH /aws1/cx_sgmresourcenotfound.
    MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
    ENDRY.

"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_knskmshtrottingex .
    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'.
    ENDRY.

"Delete stream."

```

```
TRY.
  lo_kns->deletestream(
    iv_streamname = iv_stream_name
  ).
  MESSAGE 'Stream deleted.' TYPE 'I'.
CATCH /aws1/cx_knslimitexceededException .
  MESSAGE 'The request processing has failed because of a limit exceeded
exception.' TYPE 'E'.
CATCH /aws1/cx_knsresourceinuseException .
  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 中的下列主題AWS SDK，以取得SAPABAPAPI參考。
 - [CreateStream](#)
 - [DeleteStream](#)
 - [GetRecords](#)
 - [GetShardIterator](#)
 - [PutRecord](#)

動作

CreateStream

下列程式碼範例示範如何使用 CreateStream。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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_knslimitexceeddex .
MESSAGE 'The request processing has failed because of a limit exceed
exception.' TYPE 'E'.
CATCH /aws1/cx_knsresourceinuseex .
MESSAGE 'The request processing has failed because the resource is in use.'
TYPE 'E'.
ENDTRY.
```

- 如需API詳細資訊，請參閱[CreateStream](#)中的 AWS SDK 以取得SAPABAPAPI參考。

DeleteStream

下列程式碼範例示範如何使用 DeleteStream。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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_knslimitexceeddex .
MESSAGE 'The request processing has failed because of a limit exceed
exception.' TYPE 'E'.
CATCH /aws1/cx_knsresourceinuseex .
MESSAGE 'The request processing has failed because the resource is in use.'
TYPE 'E'.
ENDTRY.
```

- 如需API詳細資訊，請參閱[DeleteStream](#)中的 AWS SDK 以取得SAPABAPAPI參考。

DescribeStream

下列程式碼範例示範如何使用 DescribeStream。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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_knslimitexceededException .  
        MESSAGE 'The request processing has failed because of a limit exceed  
exception.' TYPE 'E'.  
    CATCH /aws1/cx_knsresourcenotfoundException .  
        MESSAGE 'Resource being accessed is not found.' TYPE 'E'.  
ENDTRY.
```

- 如需 API 詳細資訊，請參閱 [DescribeStream](#) 中的 AWS SDK 以取得 SAP ABAP API 參考。

GetRecords

下列程式碼範例示範如何使用 GetRecords。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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_knskmsstrottingex .
        MESSAGE 'The rate of requests to AWS KMS is exceeding the request quotas.'
TYPE 'E'.
    CATCH /aws1/cx_knsprovthruputexcdex .
        MESSAGE 'The request rate for the stream is too high, or the requested data
is too large for the available throughput.' TYPE 'E'.
    CATCH /aws1/cx_knsresourcenotfoundex .
        MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
    ENDTRY.

```

- 如需API詳細資訊，請參閱[GetRecords](#)中的 AWS SDK 以取得SAPABAPAPI參考。

ListStreams

下列程式碼範例示範如何使用 ListStreams。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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_knslimitexceeddex .
        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[ListStreams](#)中的 AWS SDK 以取得SAPABAPAPI參考。

PutRecord

下列程式碼範例示範如何使用 PutRecord。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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_knskmssthrrottlingex .
    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'.
  ENDRTRY.

```

- 如需API詳細資訊，請參閱[PutRecord](#)中的 AWS SDK 以取得SAPABAPAPI參考。

RegisterStreamConsumer

下列程式碼範例示範如何使用 RegisterStreamConsumer。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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_sgmresourcecelimitexcd.
    MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.

```

```
CATCH /aws1/cx_sgmresourceinuse.  
    MESSAGE 'Resource being accessed is in use.' TYPE 'E'.  
CATCH /aws1/cx_sgmresourcenotfound.  
    MESSAGE 'Resource being accessed is not found.' TYPE 'E'.  
ENDTRY.
```

- 如需API詳細資訊，請參閱[RegisterStreamConsumer](#)中的 AWS SDK 以取得SAPABAPAPI參考。

使用 SDK的 Lambda 範例 SAP ABAP

下列程式碼範例示範如何使用 SAPABAP搭配 Lambda 使用 AWS SDK來執行動作和實作常見案例。

基本知識是程式碼範例，示範如何在 服務中執行基本操作。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會示範如何呼叫個別服務函數，但您可以在相關案例中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執程式碼的指示。

主題

- [基本概念](#)
- [動作](#)

基本概念

了解基本知識

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](#)。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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'
    ).
    MESSAGE 'IAM role created.' TYPE 'I'.
    WAIT UP TO 10 SECONDS.          " Make sure that the IAM role is ready
for use. "
    CATCH /aws1/cx_iamentityalrddyexex.
        MESSAGE 'IAM role already exists.' TYPE 'E'.
    CATCH /aws1/cx_iaminvalidinputex.
        MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
    CATCH /aws1/cx_iammalformedplydocex.
        MESSAGE 'Policy document in the request is malformed.' TYPE 'E'.
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_iamplynnotattachableex.
      MESSAGE 'Service role policies can only be attached to the service-
linked role for their service.' TYPE 'E'.
    CATCH /aws1/cx_iamunmodableentityex.
      MESSAGE 'Service that depends on the service-linked role is not
modifiable.' TYPE 'E'.
    ENDTRY.

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

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

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

    "Invoke the function with new parameters and get results. Display the
execution log that's returned from the invocation."
    TRY.
        lv_json = /aws1/cl_rt_util=>string_to_xstring(
            `{` ` &&
            ` "action": "decrement", ` ` &&

```

```

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

" List the functions for your account. "
TRY.
DATA(lo_list_output) = lo_lmd->listfunctions( ).
DATA(lt_functions) = lo_list_output->get_functions( ).
MESSAGE 'Retrieved list of Lambda functions.' TYPE 'I'.
CATCH /aws1/cx_lmdinvparamvalueex.
MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
ENDTRY.

" Delete the Lambda function. "
TRY.
lo_lmd->deletefunction( iv_functionname = iv_function_name ).
MESSAGE 'Lambda function deleted.' TYPE 'I'.
CATCH /aws1/cx_lmdinvparamvalueex.
MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
CATCH /aws1/cx_lmdresourcenotfoundex.
MESSAGE 'The requested resource does not exist.' TYPE 'E'.

```

```
ENDTRY.

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

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

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

- 如需API詳細資訊，請參閱 中的下列主題AWS SDK，以取得SAPABAPAPI參考。
 - [CreateFunction](#)
 - [DeleteFunction](#)

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

動作

CreateFunction

下列程式碼範例示範如何使用 CreateFunction。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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_lmdcodestorageexclex.  
    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[CreateFunction](#)中的 AWS SDK 以取得SAPABAPAPI參考。

DeleteFunction

下列程式碼範例示範如何使用 DeleteFunction。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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[DeleteFunction](#)中的 AWS SDK 以取得SAPABAPAPI參考。

GetFunction

下列程式碼範例示範如何使用 GetFunction。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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[GetFunction](#)中的 AWS SDK 以取得SAPABAPAPI參考。

Invoke

下列程式碼範例示範如何使用 Invoke。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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_lmdunsuppmediatyp00.
    MESSAGE 'Invoke request body does not have JSON as its content type.' TYPE
'E'.
  ENDTRY.

```

- 如需API詳細資訊，請參閱在 [中叫用](#) AWS SDK 以取得SAPABAPAPI參考。

ListFunctions

下列程式碼範例示範如何使用 ListFunctions。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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 [ListFunctions](#) 中的 AWS SDK 以取得 SAP ABAP API 參考。

UpdateFunctionCode

下列程式碼範例示範如何使用 UpdateFunctionCode。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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_lmdcodestorageexcidx.
    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[UpdateFunctionCode](#)中的 AWS SDK 以取得SAPABAPAPI參考。

UpdateFunctionConfiguration

下列程式碼範例示範如何使用 UpdateFunctionConfiguration。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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[UpdateFunctionConfiguration](#)中的 AWS SDK 以取得SAPABAPAPI參考。

使用 SDK 的 Amazon S3 範例 SAP ABAP

下列程式碼範例示範如何使用 SAPABAP搭配 Amazon S3 使用 AWS SDK來執行動作和實作常見案例。

基本知識是程式碼範例，示範如何在 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。雖然動作會示範如何呼叫個別服務函數，但您可以在相關案例中查看內容中的動作。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執程式碼的指示。

主題

- [基本概念](#)
- [動作](#)

基本概念

了解基本知識

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。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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.
    lo_s3->createbucket(
        iv_bucket = iv_bucket_name
    ).
    MESSAGE 'S3 bucket created.' TYPE 'I'.
CATCH /aws1/cx_s3_bucketalrddyexists.
```



```
    MESSAGE 'Bucket name already exists.' TYPE 'E'.
  CATCH /aws1/cx_s3_bkta1rdyownedbyyou.
    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 }|
  ).
ENDTRY.
```

```
    ).
    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 中的下列主題AWS SDK，以取得SAPABAPAPI參考。
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)
 - [ListObjectsV2](#)
 - [PutObject](#)

動作

CopyObject

下列程式碼範例示範如何使用 CopyObject。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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[CopyObject](#)中的 AWS SDK 以取得SAPABAPAPI參考。

CreateBucket

下列程式碼範例示範如何使用 CreateBucket。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#)中設定和執行。

```
TRY.
  lo_s3->createbucket(
    iv_bucket = iv_bucket_name
  ).
  MESSAGE 'S3 bucket created.' TYPE 'I'.
CATCH /aws1/cx_s3_bucketalrddyexists.
  MESSAGE 'Bucket name already exists.' TYPE 'E'.
CATCH /aws1/cx_s3_bktalrddyownedbyyou.
  MESSAGE 'Bucket already exists and is owned by you.' TYPE 'E'.
ENDTRY.
```

- 如需API詳細資訊，請參閱[CreateBucket](#)中的 AWS SDK 以取得SAPABAPAPI參考。

DeleteBucket

下列程式碼範例示範如何使用 DeleteBucket。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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 [DeleteBucket](#) 中的 AWS SDK 以取得SAPABAPAPI參考。

DeleteObject

下列程式碼範例示範如何使用 DeleteObject。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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[DeleteObject](#)中的 AWS SDK 以取得SAPABAPAPI參考。

GetObject

下列程式碼範例示範如何使用 GetObject。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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[GetObject](#)中的 AWS SDK 以取得SAPABAPAPI參考。

ListObjectsV2

下列程式碼範例示範如何使用 ListObjectsV2。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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 中的 [ListObjectsV2](#) AWS SDK 以取得SAPABAPAPI參考。

PutObject

下列程式碼範例示範如何使用 PutObject。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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[PutObject](#)中的 AWS SDK 以取得SAPABAPAPI參考。

SageMaker 使用 SDK 的 範例 SAP ABAP

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

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會示範如何呼叫個別服務函數，但您可以在其相關案例中查看內容中的動作。

案例是程式碼範例，示範如何透過呼叫服務內的多個函數或與其他 結合，來完成特定任務 AWS 服務。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執程式碼的指示。

主題

- [動作](#)
- [案例](#)

動作

CreateEndpoint

下列程式碼範例示範如何使用 CreateEndpoint。

SDK 適用於 SAP ABAP

 Note

還有更多。GitHub 尋找完整範例，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."
CREATE OBJECT lo_production_variants
  EXPORTING
    iv_variantname      = iv_variant_name
    iv_modelname        = iv_model_name
    iv_initialinstancecount = iv_initial_instance_count
    iv_instancetype     = iv_instance_type.

INSERT lo_production_variants INTO TABLE lt_production_variants.

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

"Create an endpoint."
TRY.
  oo_result = lo_sgm->createendpoint( " oo_result is returned for testing
purposes. "
    iv_endpointconfigname = iv_endpoint_config_name
    iv_endpointname       = iv_endpoint_name
  ).
```

```

MESSAGE 'Endpoint created.' TYPE 'I'.
CATCH /aws1/cx_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[CreateEndpoint](#)中的 AWS SDK 以取得SAPABAPI參考。

CreateModel

下列程式碼範例示範如何使用 CreateModel。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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."
CREATE OBJECT lo_primarycontainer
  EXPORTING
    iv_image          = iv_container_image
    iv_modeldataurl  = iv_model_data_url.

"Create an Amazon SageMaker model."
TRY.
  oo_result = lo_sgm->createmodel(
    " oo_result is returned for testing
    purposes. "
    iv_executionrolearn = iv_execution_role_arn
    iv_modelname       = iv_model_name
    io_primarycontainer = lo_primarycontainer
  ).
  MESSAGE 'Model created.' TYPE 'I'.
CATCH /aws1/cx_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[CreateModel](#)中的 AWS SDK 以取得SAPABAPAPI參考。

CreateTrainingJob

下列程式碼範例示範如何使用 CreateTrainingJob。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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_sgmalgorithmsspec.
DATA lo_resource_config TYPE REF TO /aws1/cl_sgmresourceconfig.
DATA lo_output_data_config TYPE REF TO /aws1/cl_sgmoutputdataconfig.
DATA lo_stopping_condition TYPE REF TO /aws1/cl_sgmstoppingcondition.
```

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

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

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

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

```
CREATE OBJECT lo_hyperparameters_w EXPORTING iv_value = iv_hp_eval_metric.
```

```
INSERT VALUE #( key = 'eval_metric' value = lo_hyperparameters_w ) INTO TABLE
lt_hyperparameters.

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

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

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

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

"Create ABAP objects for training data sources."
CREATE OBJECT lo_trn_s3datasource
EXPORTING
  iv_s3datatype           = iv_trn_data_s3datatype
  iv_s3datadistributiontype = iv_trn_data_s3datadistribution
  iv_s3uri                = iv_trn_data_s3uri.

CREATE OBJECT lo_trn_datasource
EXPORTING
  io_s3datasource = lo_trn_s3datasource.

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

INSERT lo_trn_channel INTO TABLE lt_input_data_config.

"Create ABAP objects for validation data sources."
CREATE OBJECT lo_val_s3datasource
EXPORTING
  iv_s3datatype           = iv_val_data_s3datatype
  iv_s3datadistributiontype = iv_val_data_s3datadistribution
```

```
        iv_s3uri                = iv_val_data_s3uri.

CREATE OBJECT lo_val_datasource
EXPORTING
    io_s3datasource = lo_val_s3datasource.

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

INSERT lo_val_channel INTO TABLE lt_input_data_config.

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

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

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

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

"Create a training job."
TRY.
    oo_result = lo_sgm->createtrainingjob( " oo_result is returned for
testing purposes. "
        iv_trainingjobname      = iv_training_job_name
        iv_rolearn              = iv_role_arn
```

```

        it_hyperparameters      = lt_hyperparameters
        it_inputdataconfig     = lt_input_data_config
        io_algorithmspecification = lo_algorithm_specification
        io_outputdataconfig    = lo_output_data_config
        io_resourceconfig      = lo_resource_config
        io_stoppingcondition   = lo_stopping_condition
    ).
    MESSAGE 'Training job created.' TYPE 'I'.
    CATCH /aws1/cx_sgmresourceinuse.
        MESSAGE 'Resource being accessed is in use.' TYPE 'E'.
    CATCH /aws1/cx_sgmresourceNotFound.
        MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
    CATCH /aws1/cx_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[CreateTrainingJob](#)中的 AWS SDK 以取得SAPABAPAPI參考。

CreateTransformJob

下列程式碼範例示範如何使用 CreateTransformJob。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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."
CREATE OBJECT lo_s3datasource
    EXPORTING
        iv_s3uri      = iv_tf_data_s3uri

```

```
        iv_s3datatype = iv_tf_data_s3datatype.

"Create an ABAP object for data source."
CREATE OBJECT lo_datasource
    EXPORTING
        io_s3datasource = lo_s3datasource.

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

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

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

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

- 如需API詳細資訊，請參閱[CreateTransformJob](#)中的 AWS SDK 以取得SAPABAPAPI參考。

DeleteEndpoint

下列程式碼範例示範如何使用 DeleteEndpoint。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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[DeleteEndpoint](#)中的 AWS SDK 以取得SAPABAPAPI參考。

DeleteModel

下列程式碼範例示範如何使用 DeleteModel。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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[DeleteModel](#)中的 AWS SDK 以取得SAPABAPAPI參考。

DescribeTrainingJob

下列程式碼範例示範如何使用 DescribeTrainingJob。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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[DescribeTrainingJob](#)中的 AWS SDK 以取得SAPABAPAPI參考。

ListAlgorithms

下列程式碼範例示範如何使用 ListAlgorithms。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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[ListAlgorithms](#)中的 AWS SDK 以取得SAPABAPAPI參考。

ListModels

下列程式碼範例示範如何使用 ListModels。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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[ListModels](#)中的 AWS SDK 以取得SAPABAPAPI參考。

ListNotebookInstances

下列程式碼範例示範如何使用 ListNotebookInstances。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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[ListNotebookInstances](#)中的 AWS SDK 以取得SAPABAPAPI參考。

ListTrainingJobs

下列程式碼範例示範如何使用 ListTrainingJobs。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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[ListTrainingJobs](#)中的 AWS SDK 以取得SAPABAPAPI參考。

案例

模型和端點入門

以下程式碼範例顯示做法：

- 啟動訓練任務並建立 SageMaker 模型。
- 建立端點組態。
- 建立端點，然後清除資源。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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_sgmalgorithm_spec.
DATA lo_resource_config TYPE REF TO /aws1/cl_sgmresourceconfig.
DATA lo_output_data_config TYPE REF TO /aws1/cl_sgmoutputdataconfig.
DATA lo_stopping_condition TYPE REF TO /aws1/cl_sgmstoppingcondition.
DATA lo_primarycontainer TYPE REF TO /aws1/cl_sgmcontainerdefn.
DATA lo_production_variants TYPE REF TO /aws1/cl_sgmproductionvariant.
DATA lo_ep_config_result TYPE REF TO /aws1/cl_sgmcreateendptcfgout.
DATA lo_training_result TYPE REF TO /aws1/cl_sgmdescrtrnjobrsp.
DATA lt_production_variants TYPE /aws1/
cl_sgmproductionvariant=>tt_productionvariantlist.
DATA lt_input_data_config TYPE /aws1/cl_sgmchannel=>tt_inputdataconfig.
DATA lt_hyperparameters TYPE /aws1/cl_sgmhyperparameters_w=>tt_hyperparameters.
DATA lv_model_data_url TYPE /aws1/sgmurl.

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

"Create ABAP internal table for hyperparameters based on input variables."
"These hyperparameters are based on Amazon SageMaker built-in algorithm -
XGBoost"
CREATE OBJECT lo_hyperparameters_w EXPORTING iv_value = iv_hp_max_depth.

```

```
INSERT VALUE #( key = 'max_depth' value = lo_hyperparameters_w ) INTO TABLE
lt_hyperparameters.

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

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

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

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

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

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

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

CREATE OBJECT lo_trn_datasource EXPORTING io_s3datasource = lo_trn_s3datasource.

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

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

CREATE OBJECT lo_val_datasource EXPORTING io_s3datasource = lo_val_s3datasource.

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

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

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

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

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

TRY.
  lo_sgm->createtrainingjob(
    iv_trainingjobname      = iv_training_job_name
    iv_rolearn              = iv_role_arn
    it_hyperparameters      = lt_hyperparameters
```

```
        it_inputdataconfig      = lt_input_data_config
        io_algorithmspecification = lo_algorithm_specification
        io_outputdataconfig     = lo_output_data_config
        io_resourceconfig       = lo_resource_config
        io_stoppingcondition    = lo_stopping_condition
    ).
    MESSAGE 'Training job created.' TYPE 'I'.
    CATCH /aws1/cx_sgmresourceinuse.
        MESSAGE 'Resource being accessed is in use.' TYPE 'E'.
    CATCH /aws1/cx_sgmresourceNotFound.
        MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
    CATCH /aws1/cx_sgmresourceLimitExcd.
        MESSAGE 'You have reached the limit on the number of resources.' TYPE 'E'.
    ENDTRY.

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

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

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



```

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

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

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

"Wait for endpoint creation to be completed."
DATA(lo_endpoint_result) = lo_sgm->describeendpoint( iv_endpointname =
iv_ep_name ).
WHILE lo_endpoint_result->get_endpointstatus( ) <> 'InService'.
  IF sy-index = 30.
    EXIT.          "Maximum 900 seconds."
  ENDIF.
  WAIT UP TO 30 SECONDS.
  lo_endpoint_result = lo_sgm->describeendpoint( iv_endpointname = iv_ep_name ).
ENDWHILE.

TRY.
  "Delete an endpoint."
  lo_sgm->deleteendpoint(
    iv_endpointname = iv_ep_name

```

```
    ).  
    MESSAGE 'Endpoint deleted' TYPE 'I'.  
  
    "Delete an endpoint configuration."  
    lo_sgm->deleteendpointconfig(  
        iv_endpointconfigname = iv_ep_cfg_name  
    ).  
    MESSAGE 'Endpoint configuration deleted.' TYPE 'I'.  
  
    "Delete model."  
    lo_sgm->deletemodel(  
        iv_modelname = iv_model_name  
    ).  
    MESSAGE 'Model deleted.' TYPE 'I'.  
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_endpointconfig_exception).  
    DATA(lv_endpointconfig_error) = |"{ lo_endpointconfig_exception->av_err_code }" - { lo_endpointconfig_exception->av_err_msg }|.  
    MESSAGE lv_endpointconfig_error TYPE 'E'.  
ENDTRY.
```

- 如需API詳細資訊，請參閱 中的下列主題AWS SDK，以取得SAPABAPAPI參考。
 - [CreateEndpoint](#)
 - [CreateEndpointConfig](#)
 - [CreateModel](#)
 - [CreateTrainingJob](#)
 - [DeleteEndpoint](#)
 - [DeleteEndpointConfig](#)
 - [DeleteModel](#)
 - [DescribeEndpoint](#)
 - [DescribeTrainingJob](#)

使用 SDK的 Amazon SNS範例 SAP ABAP

下列程式碼範例示範如何使用 SAPABAP搭配 Amazon 使用 AWS SDK來執行動作和實作常見案例 SNS。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會示範如何呼叫個別服務函數，但您可以在其相關案例中查看內容中的動作。

案例是程式碼範例，示範如何透過呼叫服務內的多個函數或與其他 結合來完成特定任務 AWS 服務。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- [動作](#)
- [案例](#)

動作

CreateTopic

下列程式碼範例示範如何使用 CreateTopic。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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_snstopiclimitexclex.  
        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 [CreateTopic](#) 中的 AWS SDK 以取得 SAP ABAP API 參考。

DeleteTopic

下列程式碼範例示範如何使用 DeleteTopic。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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[DeleteTopic](#)中的 AWS SDK 以取得SAPABAPAPI參考。

GetTopicAttributes

下列程式碼範例示範如何使用 GetTopicAttributes。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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[GetTopicAttributes](#)中的 AWS SDK 以取得SAPABAPAPI參考。

ListSubscriptions

下列程式碼範例示範如何使用 ListSubscriptions。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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 [ListSubscriptions](#) 中的 AWS SDK 以取得 SAP ABAP API 參考。

ListTopics

下列程式碼範例示範如何使用 ListTopics。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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[ListTopics](#)中的 AWS SDK 以取得SAPABAPAPI參考。

Publish

下列程式碼範例示範如何使用 Publish。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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.
```

- 如需API詳細資訊，請參閱在 [發佈](#) AWS SDK以供SAPABAPAPI參考。

SetTopicAttributes

下列程式碼範例示範如何使用 SetTopicAttributes。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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[SetTopicAttributes](#)中的 AWS SDK 以取得SAPABAPAPI參考。

Subscribe

下列程式碼範例示範如何使用 Subscribe。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

訂閱主題的電子郵件地址。

```

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_snssubscriptionlmt00.
  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 中的[訂閱](#) AWS SDK以取得SAPABAPAPI參考。

Unsubscribe

下列程式碼範例示範如何使用 Unsubscribe。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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 中的[取消訂閱](#) AWS SDK以取得SAPABAPAPI參考。

案例

建立和發佈至FIFO主題

下列程式碼範例示範如何建立和發佈至 FIFO Amazon SNS主題。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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、訂閱主題的 Amazon SQSFIFO佇列，以及發佈訊息至 Amazon SNS主題。

```

" Creates a FIFO topic. "
DATA lt_tpc_attributes TYPE /aws1/cl_snstopicattrsm_w=>tt_topicattributesmap.
DATA ls_tpc_attributes TYPE /aws1/
cl_snstopicattrsm_w=>ts_topicattributesmap_maprow.
ls_tpc_attributes-key = 'FifoTopic'.
ls_tpc_attributes-value = NEW /aws1/cl_snstopicattrsm_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 中的下列主題AWS SDK，以取得SAPABAPAPI參考。
 - [CreateTopic](#)
 - [發布](#)
 - [Subscribe](#)

使用 SDK 的 Amazon SQS範例 SAP ABAP

下列程式碼範例示範如何使用 SAPABAP搭配 Amazon 使用 AWS SDK來執行動作和實作常見案例 SQS。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會示範如何呼叫個別服務函數，但您可以在其相關案例中查看內容中的動作。

案例是程式碼範例，示範如何透過呼叫服務內的多個函數或與其他 結合來完成特定任務 AWS 服務。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執程式碼的指示。

主題

- [動作](#)
- [案例](#)

動作

CreateQueue

下列程式碼範例示範如何使用 CreateQueue。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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_sqsqueueexists.  
        MESSAGE 'A queue with this name already exists.' TYPE 'E'.  
ENDTRY.
```

建立等待訊息到達的 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_queue_name = iv_queue_name
    it_attributes = lt_attributes
  ).
  MESSAGE 'SQS queue created.' TYPE 'I'.
  CATCH /aws1/cx_sqsqueuedeletedrecently.
  MESSAGE 'After deleting a queue, wait 60 seconds before creating another
queue with the same name.' TYPE 'E'.
  CATCH /aws1/cx_sqsqueuealreadyexists.
  MESSAGE 'A queue with this name already exists.' TYPE 'E'.
ENDTRY.

```

- 如需API詳細資訊，請參閱[CreateQueue](#)中的 AWS SDK 以取得SAPABAPAPI參考。

DeleteQueue

下列程式碼範例示範如何使用 DeleteQueue。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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[DeleteQueue](#)中的 AWS SDK 以取得SAPABAPAPI參考。

GetQueueUrl

下列程式碼範例示範如何使用 GetQueueUrl。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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_queue_name = iv_queue_name ).      "  
oo_result is returned for testing purposes. "  
    MESSAGE 'Queue URL retrieved.' TYPE 'I'.  
    CATCH /aws1/cx_sqsqueue_does_not_exist.  
        MESSAGE 'The requested queue does not exist.' TYPE 'E'.  
ENDTRY.
```

- 如需API詳細資訊，請參閱[GetQueueUrl](#)中的 AWS SDK 以取得SAPABAPAPI參考。

ListQueues

下列程式碼範例示範如何使用 ListQueues。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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'.
    ENDRY.

```

- 如需API詳細資訊，請參閱[ListQueues](#)中的 AWS SDK 以取得SAPABAPAPI參考。

ReceiveMessage

下列程式碼範例示範如何使用 ReceiveMessage。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#)中設定和執行。

從 Amazon SQS佇列接收訊息。

```

    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'.
    ENDRY.

```

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[ReceiveMessage](#)中的 AWS SDK 以取得SAPABAPAPI參考。

SendMessage

下列程式碼範例示範如何使用 SendMessage。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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[SendMessage](#)中的 AWS SDK 以取得SAPABAPAPI參考。

案例

建立和發佈至FIFO主題

下列程式碼範例示範如何建立和發佈至 FIFO Amazon SNS主題。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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、訂閱主題的 Amazon SQS FIFO 佇列，以及發佈訊息至 Amazon SNS 主題。

```

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

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

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

```



```

        ov_subscription_arn = lv_subscription_arn.
ov_subscription_arn is returned for testing purposes. "
        MESSAGE 'SQS queue was subscribed to SNS topic.' TYPE 'I'.
        CATCH /aws1/cx_snsnotfoundexception.
        MESSAGE 'Topic does not exist.' TYPE 'E'.
        CATCH /aws1/cx_snssubscriptionlmt00.
        MESSAGE 'Unable to create subscriptions. You have reached the maximum number
of subscriptions allowed.' TYPE 'E'.
        ENDRY.

" 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'.
        ENDRY.

```

- 如需API詳細資訊，請參閱 中的下列主題AWS SDK，以取得SAPABAPAPI參考。
 - [CreateTopic](#)
 - [發布](#)

- [Subscribe](#)

使用 SDK 的 Amazon Textract 範例 SAP ABAP

下列程式碼範例示範如何使用 SAP ABAP 搭配 Amazon Textract 使用 AWS SDK 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。雖然動作會示範如何呼叫個別服務函數，但您可以在其相關案例中查看內容中的動作。

案例是程式碼範例，示範如何透過呼叫服務內的多個函數或與其他 結合來完成特定任務 AWS 服務。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- [動作](#)
- [案例](#)

動作

AnalyzeDocument

下列程式碼範例示範如何使用 AnalyzeDocument。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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.
  ENDLLOOP.
  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 Textextract 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_textthrottlingex.
  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[AnalyzeDocument](#)中的 AWS SDK 以取得SAPABAPAPI參考。

DetectDocumentText

下列程式碼範例示範如何使用 DetectDocumentText。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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[DetectDocumentText](#)中的 AWS SDK 以取得SAPABAPAPI參考。

GetDocumentAnalysis

下列程式碼範例示範如何使用 GetDocumentAnalysis。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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[GetDocumentAnalysis](#)中的 AWS SDK 以取得SAPABAPAPI參考。

StartDocumentAnalysis

下列程式碼範例示範如何使用 StartDocumentAnalysis。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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_texlimitexceeddex.
    MESSAGE 'An Amazon Textract service limit was exceeded.' TYPE 'E'.
  CATCH /aws1/cx_texprovthruputexcdex.
    MESSAGE 'Provisioned throughput exceeded limit.' TYPE 'E'.
  CATCH /aws1/cx_texthrottlingex.
    MESSAGE 'The request processing exceeded the limit.' TYPE 'E'.
  CATCH /aws1/cx_texunsupporteddocex.
    MESSAGE 'The document is not supported.' TYPE 'E'.
  ENDTRY.

```

- 如需API詳細資訊，請參閱[StartDocumentAnalysis](#)中的 AWS SDK 以取得SAPABAPAPI參考。

StartDocumentTextDetection

下列程式碼範例示範如何使用 StartDocumentTextDetection。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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_texlimitexceeddex.
    MESSAGE 'An Amazon Textract service limit was exceeded.' TYPE 'E'.
CATCH /aws1/cx_texprovthruputexcdex.
    MESSAGE 'Provisioned throughput exceeded limit.' TYPE 'E'.
CATCH /aws1/cx_texthrottlingex.
    MESSAGE 'The request processing exceeded the limit.' TYPE 'E'.
CATCH /aws1/cx_texunsupporteddocex.
    MESSAGE 'The document is not supported.' TYPE 'E'.
ENDTRY.
```

- 如需API詳細資訊，請參閱[StartDocumentTextDetection](#)中的 AWS SDK 以取得SAPABAPAPI參考。

案例

文件分析入門

以下程式碼範例顯示做法：

- 開始非同步分析。
- 取得文件分析。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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_texlimitexceeddex.
  MESSAGE 'An Amazon Textract service limit was exceeded.' TYPE 'E'.
CATCH /aws1/cx_texprovthruputexcdex.
  MESSAGE 'Provisioned throughput exceeded limit.' TYPE 'E'.
CATCH /aws1/cx_texthrottlingex.
  MESSAGE 'The request processing exceeded the limit.' TYPE 'E'.
CATCH /aws1/cx_texunsupporteddocex.
  MESSAGE 'The document is not supported.' TYPE 'E'.
ENDTRY.

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

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

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

```
ENDIF.  
ENDLOOP.
```

- 如需API詳細資訊，請參閱 中的下列主題AWS SDK，以取得SAPABAPAPI參考。
 - [GetDocumentAnalysis](#)
 - [StartDocumentAnalysis](#)

使用 SDK 的 Amazon Translate 範例 SAP ABAP

下列程式碼範例示範如何使用 AWS SDKSAPABAP搭配 Amazon Translate 的 來執行動作並實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會示範如何呼叫個別服務函數，但您可以在其相關案例中查看內容中的動作。

案例是程式碼範例，示範如何透過呼叫服務內的多個函數或與其他 結合來完成特定任務 AWS 服務。

每個範例都包含完整原始程式碼的連結，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- [動作](#)
- [案例](#)

動作

DescribeTextTranslationJob

下列程式碼範例示範如何使用 DescribeTextTranslationJob。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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."
        EXPORTING
            iv_jobid      = iv_jobid
        ).
        MESSAGE 'Job description retrieved.' TYPE 'I'.
    CATCH /aws1/cx_xl8internalserverex .
        MESSAGE 'An internal server error occurred. Retry your request.' TYPE 'E'.
    CATCH /aws1/cx_xl8resourcenotfoundex .
        MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
    CATCH /aws1/cx_xl8toomanyrequestsex.
        MESSAGE 'You have made too many requests within a short period of time.'
    TYPE 'E'.
    ENDTRY.

```

- 如需API詳細資訊，請參閱[DescribeTextTranslationJob](#)中的 AWS SDK 以取得SAPABAPAPI參考。

ListTextTranslationJobs

下列程式碼範例示範如何使用 ListTextTranslationJobs。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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."

```

```

CREATE OBJECT lo_filter
    EXPORTING
        iv_jobname = iv_jobname.

```

```

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

```

- 如需API詳細資訊，請參閱[ListTextTranslationJobs](#)中的 AWS SDK 以取得SAPABAPAPI參考。

StartTextTranslationJob

下列程式碼範例示範如何使用 StartTextTranslationJob。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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."
CREATE OBJECT lo_inputdataconfig
  EXPORTING
    iv_s3uri      = iv_input_data_s3uri
    iv_contenttype = iv_input_data_contenttype.

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

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

TRY.
  oo_result = lo_xl8->starttexttranslationjob(      "oo_result is returned for
testing purposes."
  EXPORTING
    io_inputdataconfig = lo_inputdataconfig
    io_outputdataconfig = lo_outputdataconfig
    it_targetlanguagecodes = lt_targetlanguagecodes
    iv_dataaccessrolelearn = iv_dataaccessrolelearn
    iv_jobname = iv_jobname
    iv_sourcelanguagecode = iv_sourcelanguagecode
  ).
  MESSAGE 'Translation job started.' TYPE 'I'.
  CATCH /aws1/cx_xl8internalserverex .
  MESSAGE 'An internal server error occurred. Retry your request.' TYPE 'E'.
  CATCH /aws1/cx_xl8invparamvalueex .
  MESSAGE 'The value of the parameter is not valid.' TYPE 'E'.
  CATCH /aws1/cx_xl8invalidrequestex.
  MESSAGE 'The request that you made is not valid.' TYPE 'E'.
  CATCH /aws1/cx_xl8resourcenotfoundex .
  MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
  CATCH /aws1/cx_xl8toomanyrequestsex.
  MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
  CATCH /aws1/cx_xl8unsuppdedlanguage00 .
  MESSAGE 'Amazon Translate does not support translation from the language of
the source text into the requested target language.' TYPE 'E'.
```

```
ENDTRY.
```

- 如需API詳細資訊，請參閱[StartTextTranslationJob](#)中的 AWS SDK 以取得SAPABAPAPI參考。

StopTextTranslationJob

下列程式碼範例示範如何使用 StopTextTranslationJob。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub尋找完整範例，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."
  EXPORTING
    iv_jobid      = iv_jobid
  ).
  MESSAGE 'Translation job stopped.' TYPE 'I'.
  CATCH /aws1/cx_xl8internalserverex .
  MESSAGE 'An internal server error occurred.' TYPE 'E'.
  CATCH /aws1/cx_xl8resourcenotfoundex .
  MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
  CATCH /aws1/cx_xl8toomanyrequestsex.
  MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
ENDTRY.
```

- 如需API詳細資訊，請參閱[StopTextTranslationJob](#)中的 AWS SDK 以取得SAPABAPAPI參考。

TranslateText

下列程式碼範例示範如何使用 TranslateText。

SDK 適用於 SAP ABAP

 Note

還有更多。GitHub 尋找完整範例，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."
    EXPORTING
        iv_text      = iv_text
        iv_sourcelanguagecode = iv_sourcelanguagecode
        iv_targetlanguagecode = iv_targetlanguagecode
    ).
    MESSAGE 'Translation completed.' TYPE 'I'.
    CATCH /aws1/cx_xl8detectedlanguage00 .
        MESSAGE 'The confidence that Amazon Comprehend accurately detected the
source language is low.' TYPE 'E'.
    CATCH /aws1/cx_xl8internalserverex .
        MESSAGE 'An internal server error occurred.' TYPE 'E'.
    CATCH /aws1/cx_xl8invalidrequestex .
        MESSAGE 'The request that you made is not valid.' TYPE 'E'.
    CATCH /aws1/cx_xl8resourcenotfoundex .
        MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
    CATCH /aws1/cx_xl8serviceunavailex .
        MESSAGE 'The Amazon Translate service is temporarily unavailable.' TYPE 'E'.
    CATCH /aws1/cx_xl8textsizefmtexcdex .
        MESSAGE 'The size of the text you submitted exceeds the size limit. ' TYPE
'E'.
    CATCH /aws1/cx_xl8toomanyrequestsex .
        MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
    CATCH /aws1/cx_xl8unsuppdedlanguage00 .
        MESSAGE 'Amazon Translate does not support translation from the language of
the source text into the requested target language. ' TYPE 'E'.
ENDTRY.
```

- 如需API詳細資訊，請參閱[TranslateText](#)中的 AWS SDK 以取得SAPABAPAPI參考。

案例

開始使用翻譯任務

以下程式碼範例顯示做法：

- 啟動非同步批次翻譯任務。
- 等待非同步任務完成。
- 描述非同步任務。

SDK 適用於 SAP ABAP

Note

還有更多。GitHub 尋找完整範例，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."
CREATE OBJECT lo_inputdataconfig
  EXPORTING
    iv_s3uri      = iv_input_data_s3uri
    iv_contenttype = iv_input_data_contenttype.

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

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

```

TRY.
  DATA(lo_translationjob_result) = lo_xl8->starttexttranslationjob(
    EXPORTING
      io_inputdataconfig = lo_inputdataconfig
      io_outputdataconfig = lo_outputdataconfig
      it_targetlanguagecodes = lt_targetlanguagecodes
      iv_dataaccessrolelearn = iv_dataaccessrolelearn
      iv_jobname = iv_jobname
      iv_sourcelanguagecode = iv_sourcelanguagecode
    ).
  MESSAGE 'Translation job started.' TYPE 'I'.
  CATCH /aws1/cx_xl8internalserverex .
    MESSAGE 'An internal server error occurred. Retry your request.' TYPE 'E'.
  CATCH /aws1/cx_xl8invparamvalueex .
    MESSAGE 'The value of the parameter is not valid.' TYPE 'E'.
  CATCH /aws1/cx_xl8invalidrequestex .
    MESSAGE 'The request that you made is not valid.' TYPE 'E'.
  CATCH /aws1/cx_xl8resourcenotfoundex .
    MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
  CATCH /aws1/cx_xl8toomanyrequestsex .
    MESSAGE 'You have made too many requests within a short period of time. '
TYPE 'E'.
  CATCH /aws1/cx_xl8unsuppdedlanguage00 .
    MESSAGE 'Amazon Translate does not support translation from the language of
the source text into the requested target language.' TYPE 'E'.
  ENDTRY.

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

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

  TRY.

```

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

- 如需API詳細資訊，請參閱 中的下列主題AWS SDK，以取得SAPABAPAPI參考。
 - [DescribeTextTranslationJob](#)
 - [StartTextTranslationJob](#)

中的安全性 適用於 SAP ABAP 的 AWS SDK

雲安全 AWS 是最高的優先級。身為 AWS 客戶，您可以從資料中心和網路架構中獲益，這些架構是為了滿足對安全性最敏感的組織的需求而建置的。

安全是 AWS 與您之間共同承擔的責任。[共同責任模型](#)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 AWS 服務 中執行的基礎架構 AWS 雲端。AWS 還為您提供可以安全使用的服務。若要瞭解適用於的規範遵循方案 適用於 SAP ABAP 的 AWS SDK，請參閱[AWS 服務 遵循規範計劃](#)。
- 雲端中的安全性 — 您的責任取決於您使用的資料。AWS 服務 您也必須對其他因素負責，包括資料的機密性、您的公司的要求和適用法律和法規。

本節涵蓋下列主題。

主題

- [SAP系統驗證開啟 AWS](#)
- [IAM 安全的最佳做法](#)
- [SAP 授權](#)
- [安全作業](#)
- [將憑證與 IAM Roles Anywhere 搭配使用](#)
- [使用 SAP 認證存放區](#)

SAP系統驗證開啟 AWS

SAP系統可以代表SAP使用者進行呼叫之前，SAP系統必須先 AWS 對其進行驗證 AWS。適用於 SAP ABAP 的 AWS SDK 支援在中的設定SDK檔設定中選取的下列三種驗證方法IMG。

AWS SDK為 SAP ABAP-BTP 版本只能使用SAP憑據存儲的[the section called “密鑰訪問密鑰認證”](#)方法進行身份驗證。

主題

- [Amazon EC2 執行個體中繼資料](#)
- [密鑰訪問密鑰認證](#)

- [使用任何角色進行憑證型驗證 IAM](#)
- [下一步驟](#)

Amazon EC2 執行個體中繼資料

SAP在 Amazon 上執行的系統EC2可以從 Amazon 執行個體中繼資料取得短暫、自動旋轉的登入資料。EC2有關詳情，請參閱[使用 Amazon EC2 執行個體中繼資料的登入資料](#)。

我們強烈建議您在使SDK用 SAPABAP. 若要啟用，Basis 管理員必須啟用輸出HTTP通訊。不需要進一步的基礎配置。

Note

只有在您的SAP系統在 Amazon 上執行時，才適用此身份驗證方法EC2。SAP託管在內部部署或其他雲端環境的系統無法使用此方法進行驗證。

密鑰訪問密鑰認證

使用此方法，您可以使用存取金鑰 ID 和秘密存取金鑰來驗證您的SAP系統 AWS。系SAP統會 AWS 使用使用IAM者登入。如需詳細資訊，請參閱[管理IAM使用者的存取金鑰](#)。

Basis 管理員會從管理員收到存取金鑰 ID 和秘密存取 AWS IAM金鑰。您的SAP系統必須設定為儲存存取金鑰 ID 和秘密存取金鑰。

- 安全、儲存和轉寄 (SSF)
 - 使用SSF功能進行驗 AWS SDK證SAPABAP。如需詳細資訊，請參閱[數位簽章與加密](#)。
 - 您還可以使用SSF02報告測試SSF的envelope和develop功能。如需詳細資訊，請參閱[測試SSF安裝](#)。
 - 的配置SSF步驟SAPABAP在/AWS1/IMG交易中說明。SDK移至 [技術先決條件]，然後選取 [內部部署系統的其他設定]。
- SAP認證存放區
 - 使用SAP認證存放區進 AWS SDK行驗證 SAP ABAP-BTP 版本。如需詳細資訊，請參閱[什麼是SAP認證存放區？](#)
 - 如需設定步驟，請參閱[使用SAP認證存放區](#)

使用任何角色進行憑證型驗證 IAM

憑證授權單位 (CA) 核發的 X.509 憑證可用於透過任何 AWS Identity and Access Management 角色進行驗證。憑證必須在中設定 STRUST。CA 必須在「任何 IAM 角色」中註冊為信任錨點，而且必須建立設定檔，以指定「任何角色」所採用的 IAM 角色和原則。如需詳細資訊，請參閱在任 [何 AWS Identity and Access Management 角色中建立信任錨點和設定檔](#)。

如需如何搭 SDK 配使用的隨處 IAM 角色的詳細步驟 SAPABAP，請參閱在任何地方 [使用憑證與 IAM 角色](#) 一起使用。

Note

只有透過使用匯入的憑證撤銷清單才支援憑證撤銷。如需詳細資訊，請參閱 [撤銷](#)。

下一步驟

在中驗證您的 SAP 系統後 AWS，SDK SAPABAP 自動執行 `sts:assumeRole` 以承擔用 SAP 戶業務功能的適當 IAM 角色。

IAM 安全的最佳做法

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 中繼資料或秘密金鑰登入資料進行身份驗證。
- 確保 SAP 系統具有提升自身所需的權限 `sts:assumeRole`。
- 針對每個邏輯 IAM 角色，為 SAP 使用者建立 IAM 角色，並具有執行業務功能所需的許可 (例如，Amazon S3、DynamoDB 或其他服務的必要許可)。這些是 SAP 使用者將扮演的角色。

如需詳細資訊，請參閱 [安全 SAP 鏡頭](#) 中的章節：AWS 架構良好的框架。

主題

- [亞馬遜 EC2 實例配置文件的最佳實踐](#)
- [適用於 SAP 使用者的 IAM 角色](#)

亞馬遜 EC2 實例配置文件的最佳實踐

您的 SAP 系統執行所在的 Amazon EC2 執行個體會根據其執行個體設定檔提供一組授權。一般而言，執行個體設定檔只需要具備呼叫權限 `sts:assumeRole`，以允許 SAP 系統視需要擔任特定業務的 IAM 角色。此提升至其他角色可確保 ABAP 程式可以扮演一個角色，該角色可為使用者提供完成工作所需的最低權限。例如，執行個體設定檔可能包含下列陳述式。

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

上述範例可讓 SAP 系統擔任 CFO、稽核員或報告使用者的 IAM 角色。AWSSDK 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 執行個體設定檔也可用於其他功能。

- [AWS對於 SAP 哈納的巴肯特代理](#)
- [關於 SAPAWS覆蓋 IP 位址路由的高可用性](#)

這些解決方案也可能需要 `sts:assumeRole` 備份或容錯移轉特定角色的權限，或者他們可能需要將權限直接指派給執行個體設定檔。

適用於 SAP 使用者的 IAM 角色

ABAP 程式需要許可才能執行使用者的任務：讀取 DynamoDB 表格、在 Amazon S3 中的 PDF 物件上叫用亞馬遜文字區塊、執行 AWS Lambda 功能。所有使用相同的安全性模型 AWS 軟體開發套件。您可以使用用於其他人的現有 IAM 角色 AWSSDK。

SAP 業務分析師會向 IAM 管理員詢問每個所需邏輯角色的 IAM 角色的 `arn:aws:`。例如，在財務案例中，業務分析師可以定義下列邏輯 IAM 角色。

- CFO
- AUDITOR
- REPORTING

IAM 管理員將為每個邏輯 IAM 角色定義 IAM 角色。

CFO

- `arn:aws:iam::0123456789:role/finance-cfo`
- 亞馬遜 S3 儲存貯體的讀取和寫入許可
- 對 DynamoDB 資料庫的讀取和寫入權限

AUDITOR

- `arn:aws:iam::0123456789:role/finance-auditor`
- 讀取亞馬遜 S3 存儲桶的許可
- 讀取權限

REPORTING

- `arn:aws:iam::0123456789:role/finance-reporting`
- 讀取權限
- 亞馬遜 S3 存儲桶沒有許可

業務分析師將在對應表中輸入 IAM 角色，以將邏輯 IAM 角色與實體 IAM 角色對應。

適用於 SAP 使用者的 IAM 角色必須允許 `sts:assumeRole` 受信任主體的動作。受信任的主體可能會根據 SAP 系統的驗證方式而有所不同 AWS。如需詳細資訊，請參閱 [指定主參與者](#)。

以下是一些最常見 SAP 案例的範例。

- 在亞馬遜 EC2 上執行的 SAP 系統，並指派執行個體設定檔— 在這裡，一個亞馬遜 EC2 實例配置文件附加到 IAM 角色。

```
{
```

```

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

```

- 在亞馬遜 EC2 上執行的 SAP 系統，無需執行個體設定— 在這裡，亞馬遜 EC2 為 SAP 用戶擔任角色。

```

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

```

- 內部部署執行 SAP 系統— 內部部署執行的 SAP 系統只能使用秘密存取金鑰進行驗證。如需詳細資訊，請參閱[開啟 SAP 系統驗證AWS](#)。

在此，SAP 使用者承擔的任何 IAM 角色都必須具有信任 SAP 使用者的信任關係。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [

```

```
        "sts:AssumeRole"
      ],
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/SAP_SYSTEM_S4H"
      }
    }
  ]
}
```

SAP 授權

設定 所需的授權SDK取決於 SDK版本。

主題

- [組態的授權](#)
- [SAP 最終使用者的授權](#)

組態的授權

如需詳細資訊，請參閱下列索引標籤。

SDK for SAP ABAP

SDK 為 SAP 設定 需要下列授權ABAP。

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

從下列授權群組中進行選擇。

- /AWS1/CFG- 適用於 SAP ABAP 的 AWS SDK v1 - 組態
- /AWS1/MOD- 適用於 SAP ABAP 的 AWS SDK v1 - 執行期
- /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SDK允許 - SAP ABAP BTP版本的組態存取權。

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，然後選擇無限制的讀取、寫入、值說明。
3. 前往 Business Catalog 索引標籤，並指派 /AWS1/RTBTP_BCAT Business Catalog 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。

SAP 最終使用者的授權

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 /AWS1/IMG AWS SDK的交易中定義SDK設定檔，SAPABAP或為 SDK SAP ABAP BTP版本定義自訂商業組態應用程式。一般而言，SDK設定檔會根據其業務函數命名：ZFINANCE、ZBILLING、ZMFG、ZPAYROLL等。對於每個SDK設定檔，商業分析師將定義名稱較短的邏輯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。

然後，SAP安全管理員將新增授權物件/AWS1/SESS，以授予對SDK設定檔的存取。

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，具體取決於其任務函數。例如，具有報告存取權的財務稽核員可能獲得授權，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，CFO具有讀取/寫入授權的 可能具有授權其邏輯角色PFCG的角色CFO。

驗證物件 /AWS1/LROL

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

一般而言，每個SDK設定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AUDITOR邏輯IAM角色的授權），則 AWS SDK透過確保較高優先順序（較低的序號）角色生效來中斷綁定。

與所有安全案例一樣，應給予使用者執行其任務功能的最小權限。管理PFCG角色的簡單策略是PFCG根據其授權的SDK描述檔和邏輯角色命名單一角色。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，可以將這些單一角色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，我們鼓勵您定義適合您的PFCG策略。

安全作業

靜態資料加密

AWS 密碼存取金鑰用於驗證 SDK。它們會使用 SSF 或認證存放區功能由 SAP 加密。

傳輸中的資料加密

所有撥打的通話 AWS 服務 都會使用 HTTPS 加密。SAP ICM 會管理 HTTPS 連線。AWS 證書必須在 STRUST 中受到信任。

API 使用情況

當 ABAP 使用者擔任使用的角色時sts:assumeRole，工作階段名稱的標題為USERID-SID-MANDT，其中：

- USERID是來自SY-UNAME變量的 ABAP 用戶。

- SID是來自SY-SYSID變數的 ABAP 系統識別碼。
- MANDT是來自SY-MANDT變量的 ABAP 客戶端。

工作階段名稱會 CloudTrail 以使用者名稱的形式出現。如此可確保 ABAP 使用者的 API 呼叫可追溯回啟動呼叫的系統、用戶端和使用者。如需詳細資訊，請參閱[什麼是 AWS CloudTrail ?](#)

將憑證與 IAM Roles Anywhere 搭配使用

SAP 系統可以透過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。

必要條件

在開始設定認證之前，必須符合下列先決條件。

- 您的憑證授權機構（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](#)。
- IAM SAP使用者的角色必須由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安全性的最佳實務](#)。
- 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

請依照這些指示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的 PSE](#)
- [步驟 5 – 設定 SDK 設定檔以使用 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

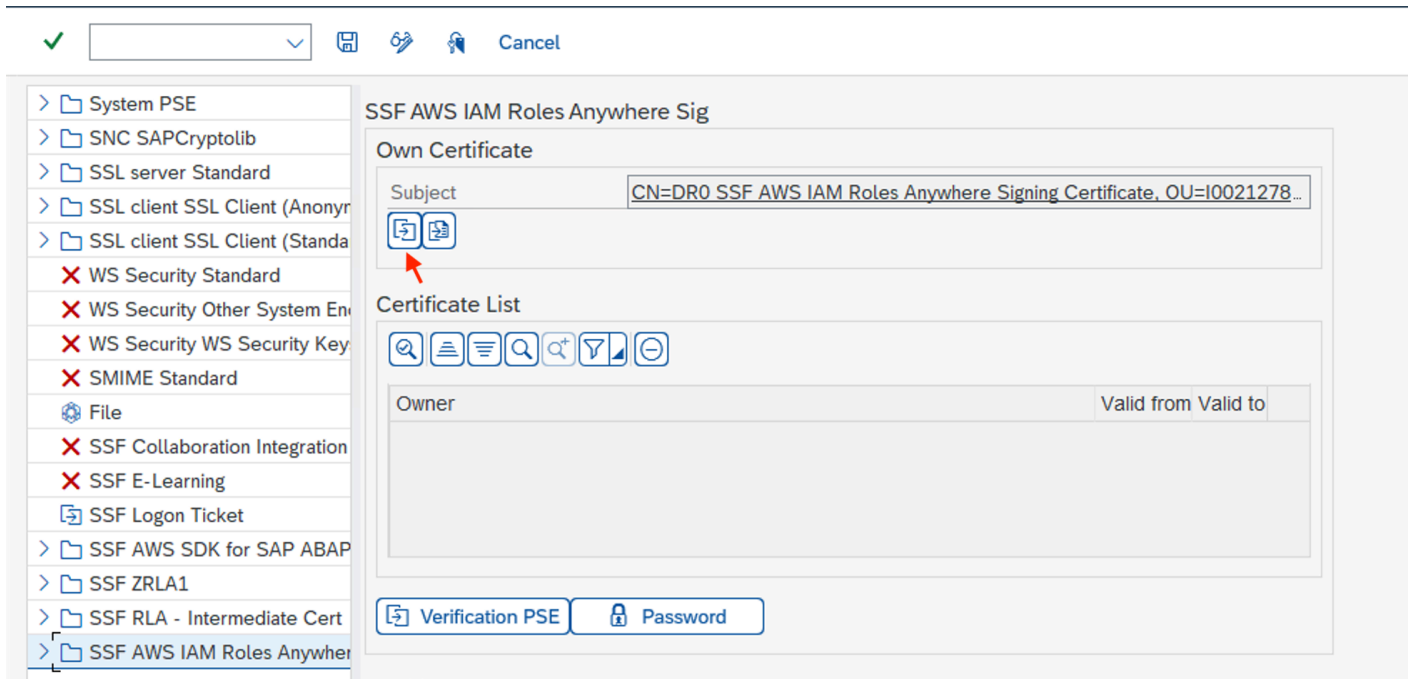
1. 執行交易程式碼 SE16 以定義 SSF 應用程式。
2. 輸入 SSFAPPLIC 資料表名稱，然後選取新增項目。
3. 在 APPLIC 檔案中輸入 SSF 應用程式的名稱、在 DESCRIPT 檔案中輸入描述，並為其餘欄位選取 Selected (X) 選項。

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 > 內部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 SSF 的應用程式。選取 Save (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 > 內部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. 用滑鼠右鍵選取在 中建立 SSF 的應用程式 [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 CA。您的 CA 會驗證請求，並以已簽署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[私有憑證授權單位發行私有終端實體憑證](#)。AWS

步驟 4 – 將憑證回應匯入相關的 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 > 內部部署系統的其他設定。
2. 執行Create PSE for SSF ApplicationIMG活動。
3. 選取STRUST交易的編輯。
4. 選擇SSF應用程式，然後選擇位於主旨下方的 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. 選取主題兩次，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設定檔以使用 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。
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您系統的資訊，以及 AWS 區域。
- 為身分驗證方法選取 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SSF的應用程式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 “[必要條件](#)”。請參閱下圖。選取繼續。

Select Signing Certificate issued by your certificate authority (CA) from SSF

Certificate (SSF Application)

Enter your IAM Roles Anywhere details

Trust Anchor ARN

Profile ARN

4. 在左側窗格中，選取IAM角色映射。輸入名稱，並提供IAM管理員ARN提供IAM的角色。

如需詳細資訊，請參閱[應用程式組態](#)。

使用 SAP 認證存放區

SAP 業務技術平台中使用 SAP 認證存放區，以安全地將秘密存取金鑰驗證的認證儲存至 AWS。您必須訂閱才能使用該服務。

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 認證存放區](#)。

主題

- [設定步驟](#)
- [搭配 SDK 使用 SAP 認證存放區](#)

設定步驟

步驟

- [步驟 1：設定驗證設定](#)
- [步驟 2：建立服務金鑰](#)
- [步驟 3：將服務密鑰轉換為.p12格式](#)
- [步驟 4：Connect 至 SAP BTP、ABAP 環境](#)

步驟 1：設定驗證設定

使用下列步驟設定認證存放區設定以進行驗證。

1. 導覽至 SAP 認證存放區執行個體的「設定」標籤。
2. 選擇編輯模型組態：
 - 選擇相互 TLS 作為預設驗證類型。
 - 為承載加密狀態選取 [停用]。承載在傳輸過程中使用 HTTPS 進行加密。但是，承載目前無法雙重加密。
3. 選取 Save (儲存)。

步驟 2：建立服務金鑰

使用下列步驟建立認證存放區的服務金鑰。

1. 在 SAP 認證存放區應用程式的左窗格中，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。

服務金鑰是根據選擇的驗證類型建立的。下載服務密鑰，並保持其安全以備以後使用。

步驟 3：將服務密鑰轉換為.p12格式

為通訊系統建立輸出使用者需要.p12格式的用戶端憑證。使用下列步驟，從認.p12證存放區服務金鑰中提供的憑證詳細資料產生憑證。

1. 從 SAP 信任中心服務下載 SAP 雲端根 CA [憑證 \(SAP 需要\)](#)。

2. 以任何文字檔案格式開啟 SAP 雲端根 CA 憑證。在檔案結尾按 Enter，然後從服務金鑰的憑證欄位複製並貼上憑證。以實際\n的新行取代新行字元 (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。此私密金鑰必須被視為敏感資料。將其粘貼到文本文件中，並\n用實際的新行替換新行字符 (Enter)。將私密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密金鑰，執行下列命令以產生 .p12憑證。

```
openssl pkcs12 -export -out <.p12_filename> -inkey <private_key.key> -in  
<certificate.cer>
```

該命令需要驗證導出密碼。保留密碼以供進一步使用。

刪除保存在私鑰中的 .key 文本文件。

步驟 4：Connect 至 SAP BTP、ABAP 環境

配置 SAP BTP，ABAP 環境與 SAP 憑證存儲連接。

主題

- [通訊系統](#)
- [溝通安排](#)

通訊系統

使用下列步驟建立通訊系統，以啟用 SAP BTP、ABAP 環境與 SAP 認證存放區的通訊。

1. 打開 ABAP 環境系統的菲奧里啟動板。
2. 選取「通訊系統」圖標以開啟應用程式。
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 是 `https://credstore.mesh.cf.us10.hana.ondemand.com/api/v1/credentials`，則主機名稱為 `credstore.mesh.cf.us10.hana.ondemand.com`。
 - 輸出通訊的使用者：選+取此選項可新增使用者。
 - a. 選取「SSL 用戶端憑證」作為驗證機制。

- b. 選取 [上傳新憑證] :
 - 瀏覽上述步驟中產生的 .p12憑證。
 - 輸入描述。
 - 輸入用於產生 .p12憑證的匯出密碼。
 - 選取 [上傳]。
 - c. 選取建立以建立輸出使用者。
6. 選取 Save (儲存)。
 7. 刪除在上一個步驟中下載的服務金鑰。

溝通安排

使用下列步驟建立通訊安排，以提供輸出通訊的通訊案例。

1. 打開 ABAP 環境系統的菲奧里啟動板。
2. 選取「通訊安排」動態磚以開啟應用程式。
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，瀏覽至上一個步驟中建立的「通訊系統」。其他信息是系統的選擇後自動填充。
7. 選取 Save (儲存)。
8. 選取「檢查連線」以測試您的連線。

完成此設定之後，ABAP 環境可以使用通訊安排，透過輸出服務 (HTTP) 使用 SAP 認證存放區服務。

搭配 SDK 使用 SAP 認證存放區

步驟

- [步驟 1：建立命名空間和認證](#)
- [步驟 2：配置自定義業務配置應用程序](#)

步驟 1：建立命名空間和認證

使用 SAP 說明在 SAP 認證存放區中建立命名空間和認證 — [建立、編輯及刪除認證](#)。

輸入下列詳細資料以建立 Key 類型的認證。

- 命名空間 — 輸入命名空間的名稱，並將相關認證分組在一起。
- 名稱 — 輸入金鑰的名稱。我們建議 `aws-0123456789012-username`，其中：
 - `0123456789012` 是認 AWS 帳戶 證授與存取權的 ID
 - `username` 是認證所屬的 IAM 使用者名稱
- 值 — 輸入以 64 為基礎編碼的秘密存取金鑰。使用以下命令對您的秘密訪問密鑰進行 base-64 編碼。

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

該命令從標準輸入讀取秘密訪問密鑰，並將其傳遞給 base64 而不帶尾隨換行符。它將 base-64 編碼的秘密訪問密鑰輸出到屏幕上。將值複製到 SAP 認證存放區後，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。

會建立具有一個認證的新命名空間，並且可以在此命名空間內新增、刪除或修改認證。

遵循最低權限原則來管理儲存在命名空間中之認證的存取權。

步驟 2：配置自定義業務配置應用程式

使用下列步驟來設定「自訂業務組態」應用程式，以定義 SDK 用於驗證的認證。

1. 打開 ABAP 環境系統的菲奧里啟動板。
2. 瀏覽 [自訂業務組態] 動態磚以開啟應用程式。
3. 打開 SDK 配置文件業務配置。
4. 選取必須針對 SAP 認證存放區設定驗證設定的 SDK 設定檔。
5. 在所選設定檔的 [驗證與設定] 索引標籤中，選取 [編輯]，然後輸入下列詳細資訊：

- 驗證方法 — 從 SAP 認證存放區中選取認證。
 - 命名空間 — 輸入在 SAP 認證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：建立命名空間和認證”](#)。
 - 金鑰名稱 — 輸入已建立之服務金鑰的名稱。如需詳細資訊，請參閱 [the section called “步驟 2：建立服務金鑰”](#)。
 - 通訊安排 — 輸入已建立之通訊安排的名稱。如需詳細資訊，請參閱 [the section called “溝通安排”](#)。
6. 選擇「應用」以轉到「AWS SDK 配置文件」屏幕。
 7. 選取 [選取傳輸] 以使用值說明選取傳輸。
 8. 選取 Save (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](#)
- [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' 不包含介面 'IF_SYSTEM_UUID_RFC4122__STATIC'

原因 – 您的系統上缺少SAP備註 0002619546。

解決方案 – 確保[SAP備註 0002619546](#) 已套用至您的系統。

未指定的位置限制

問題 – 未指定的位置限制與傳送此請求region的特定端點不相容

原因 – Amazon S3 儲存貯體缺少io_createbucketconfiguration參數中的 AWS 區域。

解決方案 – 在除 us-east-1 以外的任何區域中建立儲存貯體時，請使用 `io_createbucketconfiguration` 參數指定 Amazon S3 儲存貯體的區域 `createbucket()`。您不需要為 `createbucket()` 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 docs.aws.amazon.com : 443 失敗：
SSSLERR_NO_SSL_RESPONSE

原因 – DEFAULT設定檔TRUE中的icm/HTTPS/client_sni_enabled參數未設定為。

解決方案 – 使用下列步驟對指定問題或任何其他 SSL相關問題進行故障診斷。

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，選取延伸維護，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

請確定您正在選取要建立的參數，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 – 在 <profile_name> 設定檔下找不到 <sid> : <client> 案例DEFAULT的組態

原因 – <profile_name> 不正確或尚未設定。

解析度 – 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. 前往應用程式組態 > SDK設定檔。
 - 如果您的設定檔已設定，請確認設定檔名稱是否正確。
 - 如果您的設定檔未設定，請依照步驟設定設定檔。
3. 選取新項目。
 - 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，然後選擇身分驗證和設定。
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
 - 針對在中執行SAP的系統，透過中繼資料選取執行個體角色 AWS。
 - 針對執行內部部署或其他雲端SAP的系統，從 SSF Storage 選取憑證。
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。

- ARN 已指定不正確IAM的角色
- IAM 使用者缺少存取IAM角色的許可
- 擔任IAM的角色與擔任IAM的角色或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. 前往應用程式組態 > 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，然後驗證或更正您的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[對IAM角色 進行疑難排解](#)。

執行必要動作的授權

問題 – 使用者 <user_arn> 未獲授權在資源上執行：<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 - 未設定作用中案例。

解決方案 – 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單字的一部分

Note

此警告前面可能會出現不同的錯誤訊息。

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。

解決方案 – 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。

偵測到非中斷空間字元。轉換為空格？

選擇是回答此問題。此外，我們建議您選擇程式碼來複製，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，且無法建立 TCP/IP 連線至 tla.region.amazonaws.com 的連接埠 443。

解析 – 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 上 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
- 透過代理伺服器連線

如需詳細資訊，請參閱[透過代理伺服器 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 以傳輸方式提供，而適用於 SAP ABAP 的 AWS SDK-BTP 版本以附加元件形式提供。導入傳輸和附加組件的機制不同，但技術功能是相同的。如需詳細資訊，請參閱[設定](#)。

主題

- [發行策略](#)
- [最佳實務](#)
- [修補軟體開發套件](#)
- [安裝附加模組](#)
- [解除安裝軟體開發套件](#)

發行策略

的版本 1 適用於 SAP ABAP 的 AWS SDK 會經常更新。新的修補程式會根據的發行版本和更新，每週或每天發行 AWS 服務。的修補程式 AWS 服務 可以包含錯誤修正和其他更新 SDK 修補程式層級的變更。如需詳細資訊，請參閱 [AWS SDK 和工具維護](#) 政策。

最佳實務

我們建議針對所有 SAP 系統 (開發、品質保證和生產) 保留相同的 SDK 修補程式等級。

修補 SDK 時，請在沙箱中匯入最新版本。然後，您可以依照正常的變更控制程序，將其匯入開發、QA 和生產系統。

修補軟體開發套件

每個 SAP ABAP 版本的 SDK 都會以一組累積傳輸方式提供，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區別。您必須匯入最新的傳輸，以修補 SAP ABAP 的 SDK。

由於 R core runtime 和 API 模組的相依性，即使您不再使用這些 core 模組，您也必須修補該模組和已安裝的所有其他模組。例如，如果您在安裝 SDK 時匯入 coreec2、和 lmd 傳輸，則必須匯入、以及修補 lmd 時的 core 最新傳輸。ec2

安裝附加模組

在 core 與現有模組相同的修補程式層級匯入新模組的傳輸，以便在 SAP 系統中安裝額外的 API 模組。[the section called “修補軟體開發套件”](#) 如果您要匯入較新版本的模組，請遵循中的指導方針。這可確保修補程式層級在所有 SDK 模組之間相容。

解除安裝軟體開發套件

若要解除安裝軟體開發套件，您必須從下載刪除傳輸套 [件 sdk-for-sapabap. awsSdkSapabap V1/ release/uninstall-abapsdk-LATEST.zip](#)

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

您可以從以下位置下載一個簽名文件：[//sdk-for-sapabap.aws. awsSdkSapabap amazon.com/ V1 /發布/卸載-有關 apSDK 的最新版本](#)。若要驗證檔案，請參閱[驗證 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。您可以在不卸載整個 SDK 的情況下刪除單個模塊。您可以透過僅匯入要移除之模組的刪除傳輸來執行此操作。如果您要解除安裝包含所有模組的整個 SDK，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，先在沙箱中測試解除安裝。

考量

解除安裝 SDK 之前，請參閱下列考量事項。

- 將會遺失的 SDK 組態設定。IMG 必須在安裝時重新設定。
- 如果您有依賴 SDK 的 Z 程序，它們將在刪除 SDK 後生成語法錯誤。
- 在移除 SDK 之後，PFCG 或包含 SDK 授權參照的商務角色將具有無效的授權。在解除安裝 SDK 之前，請先從 PFCG 角色中移除 SDK 授權參照。

Note

AWS 適用於 SAP ABAP 的 SDK-在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 條款約束。您有責任遵守 SAP 授權條款，包括軟體發佈和間接授權條件。所提供的任何信息都不是法律建議，也不應該被依賴於許可合規目的。如果您對 SAP 軟體的授權或權利有任何疑問，請諮詢您的法律團隊、SAP 和/或 SAP 經銷商。

問題：適用於 SAP ABAP 的 SDK 使用是否會影響我的 SAP 授權？

答：適用於 SAP ABAP 的 AWS SDK使您可以使用自己AWS 服務的 ABAP 代碼來消費。它是在 SAP 系統和之間的集成方案中使用AWS 服務。任何來自 SAP 系統的資料傳送至協力廠商 (非 SAP) 系統，或由該系統建立，都可能影響間接授權。SAP 有多種定義間接存取的方法，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 簽訂合約中提供的指引，並且可以與 SAP 或其經銷商進一步討論此問題。

2018 年，SAP 發布了兩份文件-SAP 安裝基礎客戶的間接訪問指南和 SAP ERP 數字時代定價-解決間接/數字訪問。這些文件可在 SAP 網站上找到，也是間接授權方法的範例。但是，這些文件並不反映您與 SAP 的特定協議。

適用於 SAP ABAP 的 AWS SDK 開發人員指南的文件記錄

下表說明的文件版本 適用於 SAP ABAP 的 AWS SDK。

變更	描述	日期
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	新增在 任何地方使用憑證搭配 IAM 角色 。	2023 年 12 月 1 日
新內容	新增 使用 SDK 建構產品 。	2023 年 12 月 1 日
新內容	新增 重試行為 。	2023 年 12 月 1 日
新內容	已新增 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 疑難排解 適用於 SAP ABAP 的 AWS SDK 。	2023 年 2 月 17 日
開發者預覽	開發人員指南的 適用於 SAP ABAP 的 AWS SDK 開發者預覽。	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。