



开发人员指南

AWS 适用于 SAP 的 SDK ABA



AWS 适用于 SAP 的 SDK ABA: 开发人员指南

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆、贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其他商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

Table of Contents

什么是 适用于 SAP ABAP 的 AWS SDK ?	1
的特点 适用于 SAP ABAP 的 AWS SDK	1
维护	1
API 参考	1
定价	2
资源	2
入门	3
第 1 步：准备 AWS 账户	3
适用于 SAP 用户的 IAM 角色	3
身份验证	4
第 2 步：安装软件开发工具包	5
步骤 3：配置 SDK	6
第 4 步：功能设置	7
步骤 5：授权 SAP 用户	9
第 6 步：编写代码	11
步骤 7：运行应用程序	14
设置	16
SAP 先决条件	16
适用于 SAP ABAP 的 SDK	16
适用于 SAP ABAP 的 SDK-BTP 版	20
正在安装 适用于 SAP ABAP 的 AWS SDK	20
下载 SDK	21
验证文件	21
AWS SDK 传输	22
正在安装 SDK-BTP 版	25
安装适用于 SAP ABAP 的 SDK-BTP 版	25
模块	26
适用于 SAP ABAP 的补丁 SDK-BTP 版	27
配置	28
全局设置	29
技术设置	30
配置场景	30
应用程序配置	31
SDK 配置文件	31

逻辑资源解析程序	32
示例	32
运行时设置	33
记录和跟踪	33
选择加入：增强型遥测功能	33
活动场景	33
高级连接场景	33
通过代理服务器连接	34
通过数据包检查防火墙连接	34
网关端点	35
自定义接口端点	35
多区域端点访问	36
服务提供商设置	37
刷新、跟踪和遥测	37
SAP 系统刷新	37
跟踪	38
遥测	39
使用 SDK	40
数据表示形式	40
数据类型	40
AWS 数据类型	42
示例程序	43
先决条件	43
代码	44
代码部分	44
概念	46
API 类	47
其他对象	47
结构类	47
数组	49
映射	50
更高级别的函数	50
特征	1
编程配置	51
Waiter	51
分页器	52

重试行为	53
构建产品	54
设置产品 ID	54
将 HTTP 请求自定义为 AWS	55
实施增强	55
筛选增强功能	55
对增强功能进行编码	55
限制	56
代码示例	57
Amazon Bedrock 运行时系统	57
Anthropic Claude	58
Stable Diffusion	61
Amazon Bedrock 代理运行时	63
操作	64
CloudWatch	65
操作	64
场景	70
DynamoDB	72
基本功能	72
操作	64
Amazon EC2	86
操作	64
Kinesis	102
基本功能	72
操作	64
Lambda	111
基本功能	72
操作	64
Amazon S3	125
基本功能	72
操作	64
SageMaker AI	134
操作	64
场景	70
Amazon SNS	152
操作	64

场景	70
Amazon SQS	160
操作	64
场景	70
Amazon Textract	168
操作	64
场景	70
Amazon Translate	178
操作	64
场景	70
安全性	187
系统身份验证	187
元数据身份验证	188
使用秘密访问密钥进行身份验证	188
使用 IAM Roles Anywhere 进行基于证书的身份验证	189
后续步骤	189
IAM 安全最佳实践	189
Amazon EC2 实例配置文件的最佳实践	190
适用于 SAP 用户的 IAM 角色	190
SAP 授权	193
配置授权	193
最终用户的 SAP 授权	194
安全运营	195
静态数据加密	195
传输中数据加密	195
API 用法	1
使用凭证	196
先决条件	196
过程	196
凭据存储	20
配置步骤	200
将 SAP 凭据存储与 SDK 配合使用	201
故障排除	205
导入失败	205
未指定的位置限制	205
SSL 错误	206

配置文件配置	206
IAM 授权	207
操作授权	208
活动场景	33
特殊字符	208
连接	209
其他主题	210
发行版	210
版本策略	210
最佳实践	189
修补适用于 SAP ABAP 的 SDK	211
安装其他模块	211
卸载适用于 SAP ABAP 的 SDK	211
SAP 许可	212
文档历史记录	213
.....	CCxiv

什么是 适用于 SAP ABAP 的 AWS SDK ?

适用于 SAP ABAP 的 AWS SDK 为 ABAP 语言提供的 AWS 服务提供了接口。使用软件开发工具包，您可以在上实现 ABAP BADIs、报告、交易、 OData 服务以及其他 ABAP 项目 AWS 服务，例如亚马逊简单存储服务 (Amazon S3) Simple Service、等 Amazon DynamoDB。 Amazon Translate您还可以在 SAP 业务技术平台环境中为基于 ABAP 的系统进行开发，从 SAP NetWeaver 7.4 开始。有关更多信息，请参阅[安装 AWS 适用于 SAP ABAP 的 SDK-BTP 版](#)。

主题

- [的特点 适用于 SAP ABAP 的 AWS SDK](#)
- [SDK 主要版本的维护和支持](#)
- [API 参考](#)
- [定价](#)
- [其他资源](#)

的特点 适用于 SAP ABAP 的 AWS SDK

适用于 SAP ABAP 的 AWS SDK 旨在让 SAP 开发人员感到熟悉和自然。例如，虽然所有字符串都 AWS 服务 使用true和false字符串来表示 XML 和 JSON 结构中的布尔数据，但 SDK for SAP ABAP 会将其转换为 ABAP 原生 'X' 值和单字符值。' ' 适用于 SAP ABAP 的 SDK 会尽量使用 ABAP 原生构造，其中包括数据类型和时间戳格式。因此，ABAP 程序员无需担心底层 JSON 和 XML 序列化，也不必担心 API 协议的线路格式。

SDK 主要版本的维护和支持

有关 SDK 主要版本及其底层依赖项的维护和支持的信息，请参阅《[AWS SDKs 和工具参考指南](#)》中的以下内容：

- [AWS SDKs 和工具维护政策](#)
- [AWS SDKs 以及工具和版本支持矩阵](#)

API 参考

要查看完整列表 适用于 SAP ABAP 的 AWS SDK APIs，请参阅 [适用于 SAP ABAP 的 AWS SDK -API 参考指南](#)。

要查看的完整模块列表 适用于 SAP ABAP 的 AWS SDK TLAs，请参阅 [适用于 SAP ABAP 的 AWS SDK -模块列表](#)。

要查看适用于 SAP ABAP 的 SDK-BTP 版开发者预览版的完整模块列表 TLAs，请参阅[AWS 适用于 SAP ABAP 的 SDK-BTP 版本](#)——模块列表。

定价

适用于 SAP ABAP 的 AWS SDK 无需支付额外费用即可使用。您只需为使用 SDK 使用的 AWS 资源和服务付费。

其他资源

除本指南外，适用于 SAP ABAP 的 SDK 还可使用以下在线资源。

- [SAP 正在谈 AWS 文档](#)
- [AWS 开发人员博客](#)
- [AWS 开发者论坛](#)
- [AWS SDK 代码示例库](#)
- [@awsdevelopers](#) (Twitter)

入门 适用于 SAP ABAP 的 AWS SDK

本节介绍如何开始使用 SDK。其中包括有关安装 SDK、执行基本配置和创建使用不同语言翻译短语的 Hello World 代码示例的信息。如果您不熟悉 AWS SDK，我们建议您在沙盒环境中执行这些步骤。

步骤

- [第 1 步：准备 AWS 账户](#)
- [第 2 步：安装软件开发工具包](#)
- [步骤 3：配置 SDK](#)
- [第 4 步：功能设置](#)
- [步骤 5：授权 SAP 用户](#)
- [第 6 步：编写代码](#)
- [步骤 7：运行应用程序](#)

第 1 步：准备 AWS 账户

必须准备一个活动 AWS 账户，才能开始使用适用于 SAP ABAP 的 SDK。AWS 账户 即使你的 SAP 系统托管在本地、SAP 业务技术平台 (BTP) 上或其他云提供商，你也需要一个。

如果您的 SAP 系统在 AWS Cloud 上运行，那么您将调用自己的 AWS 服务 AWS 账户。

主题

- [适用于 SAP 用户的 IAM 角色](#)
- [身份验证](#)

适用于 SAP 用户的 IAM 角色

- 根据 AWS Identity and Access Management 用户指南说明，创建 IAM 角色。有关更多信息，请参阅[创建角色以向 AWS 服务委派权限](#)。记下 IAM 角色的 Amazon 资源名称 (ARN) 以备后用。
- 选择 Amazon EC2 作为用例。
- 使用 SapDemoTranslate 作为角色名称。
- 为角色附加 TranslateReadOnly 配置文件。

- 角色必须具有以下实体，才能支持 SAP 系统代入此角色。将 **"111122223333"** 替换为您的 AWS 账号。

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

此示例表明，来自的任何委托人 AWS 账户 **"111122223333"** 都可以担任该角色。这是一个广泛的权限，适用于 proof-of-concept。在生产环境中可以使用权限更小的主体，如以下示例。

- 特定用户 — 当 SAP 系统使用以下任一用户时：
 - 来自本地 SAP 系统的 SSF 加密证书
 - 来自 SAP BTP、ABAP 环境上的 SAP 凭证存储服务的证书
- 特定角色 — 当 SAP 系统在 Amazon 上 EC2 并且有实例配置文件时。
- 亚马逊 EC2 — 当 SAP 系统位于亚马逊上 EC2 且没有实例配置文件时。

有关更多信息，请参阅 [IAM 安全最佳实践](#)。

身份验证

身份验证取决于 SAP 系统的托管位置。

位置

- [在 AWS 云上](#)
- [本地、SAP BTP 或其他云端](#)

在 AWS 云上

确保运行您的 SAP 系统的 EC2 实例具有具有以下权限的实例配置文件。

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

添加之前步骤中记录的 ARN。

此权限能让 SAP 系统代表 ABAP 用户代入 SapDemoTranslate 角色。

本地、SAP BTP 或其他云端

如果您的 SAP 系统位于本地、SAP BTP 或其他云上，请使用以下步骤使用私有访问密钥建立身份验证连接。

1. 创建 IAM 用户。有关更多信息，请参阅[创建 IAM 用户（控制台）](#)。
2. 使用 SapDemoSID 作为 IAM 用户名以及 SID 作为 SAP 系统的系统 ID。
3. 为用户分配 SapDemoTranslate 角色。

保留access_key和secret_access_key。您必须在 SAP 系统中配置这些凭证。

Note

如果您的 SAP 系统位于本地、SAP BTP 或其他云上，则可以使用以下选项之一进行身份验证。

- 使用 SSF 或 SAP 凭据存储进行@@ [秘密访问密钥身份验证](#)
- [随处使用带有 IAM 角色的证书](#)

第 2 步：安装软件开发工具包

有关安装说明，请参阅以下选项卡。

SDK for SAP ABAP

向 SAP 系统传输适用于 SAP ABAP 的 SDK。此传输内容可导入任意客户端。有关更多信息，请参阅[安装适用于 SAP ABAP 的开发工具包](#)。

SDK for SAP ABAP - BTP edition

使用 Deploy Product 应用程序安装适用于 SAP ABAP 的 SDK-BTP 版。有关更多信息，请参阅[安装适用于 SAP ABAP 的 SDK-BTP 版](#)。

步骤 3：配置 SDK

在配置 SDK 之前，请确保您拥有所需的授权。有关更多信息，请参阅[SAP 授权](#)。

有关配置说明，请参阅以下选项卡。

SDK for SAP ABAP

运行 /AWS1/IMG 事务以打开适用于 SAP ABAP 的 SDK 实施指南。如需运行此事务，请在 SAP 系统的命令栏中输入 /n/AWS1/IMG，然后点击 Enter。

完成以下配置。

- 转到技术先决条件。
 - 查看推荐的[参数](#)和[HTTPS 连接](#)。
- 转到全局设置 → 配置场景。
 - 根据[全局设置中的建议更改设置](#)。
- 转到全局设置 → 技术设置。
 - 根据[全局设置中的建议更改设置](#)。
- 转到运行时设置 → 记录和跟踪。
 - 选择新建条目。
 - 跟踪等级：无跟踪。
 - 最大转储行数：100。
 - 选择加入：增强遥测功能：留空。
 - 选择保存。
- 转到运行时设置 → 活动场景。
 - 在新建场景中选择 DEFAULT。

- 选择提交场景更改。
- 接受提示。

本地系统的先决条件

如果 SAP 系统运行在本地或其他云中，则必须将凭证存储在 SAP 数据库中。凭证使用 SAP SSF 进行加密，需要配置加密库，例如 SAP 的加密库。CommonCryptoLib

/AWS1/IMG 事务描述了为适用于 SAP ABAP 的 SDK 配置 SSF 的步骤。

Note

如果您的 SAP 系统在 Amazon 上运行，则上述先决条件不适用 EC2。在 Amazon 上运行的 SAP 系统会从亚马逊 EC2 实例元数据中 EC2 检索短暂的、自动轮换的证书。

SDK for SAP ABAP - BTP edition

在 Web 浏览器中打开您的 ABAP 环境，然后导航到“自定义业务配置”应用程序。

完成以下配置。

- 转至配置方案。
 - 根据[全局设置中的建议更改设置](#)。
- 前往技术设置。
 - 根据[全局设置中的建议更改设置](#)。

第 4 步：功能设置

有关设置说明，请参阅以下选项卡。

SDK for SAP ABAP

运行事务 /AWS1/IMG (/n/AWS1/IMG 在命令栏中输入，然后选择 Enter) 以打开 AWS SDK 的实现指南。

- 转到应用程序配置 → SDK 配置文件。
 - 选择新建条目。

- 配置文件：DEMO。
- 描述：Demo profile
- 选择保存。
- 突出显示创建的条目，然后单击身份验证和设置树分支。
 - 选择新建条目。
 - SID：当前 SAP 系统的系统 ID。
 - 客户端：当前 SAP 系统的客户端。
 - 场景 ID：下拉列表，其中列出了由基础管理员创建的默认场景。
 - AWS 区域：输入您要拨打电话的 AWS 区域。如果您的 SAP 系统正在运行 AWS，请输入其运行所在的 AWS 区域。
 - 身份验证方法：
 - 如果您的 SAP 系统在 Amazon 上运行，请通过元数据选择实例角色 EC2。
 - 如果 SAP 系统运行在本地或其他云中，则选择 SSF 存储中的凭证。
 - 选择设置凭证。
 - 输入之前步骤中创建的访问密钥 ID 和秘密访问密钥。
 - 将禁用 IAM 角色留空。
 - 选择保存。
 - 单击 IAM 角色映射树分支。
 - 选择新建条目。
 - 输入序列号：010。
 - 输入 IAM 逻辑角色：TESTUSER。
 - 输入 IAM 角色 ARN：输入 IAM 角色的 arn:aws:，其中包含之前步骤中创建的 TranslateReadOnly 策略。

SDK for SAP ABAP - BTP edition

使用 SAP 凭据存储设置身份验证。有关更多信息，请参阅[使用 SAP 凭据存储](#)。

在 Web 浏览器中打开您的 ABAP 环境，然后导航到“自定义业务配置”应用程序。

- 前往 SDK 配置文件。
 - 选择“编辑”以创建新的配置文件。
 - 配置文件：DEMO。

- 描述：Demo profile
- 选择创建的条目旁边的右箭头键导航到“身份验证和设置”选项卡。

选择新建条目。

- SID：当前 SAP 系统的系统 ID。
- 客户端：当前 SAP 系统的客户端。
- 场景 ID：下拉列表，其中列出了由基础管理员创建的默认场景。
- AWS 区域：输入您要拨打电话的 AWS 区域。如果您的 SAP 系统正在运行 AWS，请输入其运行所在的 AWS 区域。
- 身份验证方法：从 SAP 凭据存储中选择凭据。
- 输入存储在 SAP 凭证存储中的凭据的命名空间和密钥名称。
- 输入为在适用于 SAP ABAP 的 SDK-BTP 版本和 SAP 凭证存储区之间建立通信而创建的通信安排的名称。
- 将禁用 IAM 角色留空。
- 右键单击创建的条目旁边的右箭头键以导航到 IAM 角色映射选项卡。

选择新建条目。

- 输入序列号：010。
- 输入 IAM 逻辑角色：TESTUSER。
- 输入 IAM 角色 ARN：输入 IAM 角色的 arn:aws:，其中包含之前步骤中创建的 TranslateReadOnly 策略。

步骤 5：授权 SAP 用户

默认情况下，SAP 用户无权使用 AWS 功能。必须使用 SAP 权限对用户进行明确授权。有关更多详细信息，请参阅以下选项卡。

SDK for SAP ABAP

创建 PFCG 角色

- 转到 PFCG 事务
- 输入角色名称 ZAWS_SDK_DEMO_TESTUSER 并选择创建单个角色。
- 描述：Role for demo AWS SDK functionality.

- 转到授权选项卡。
- 选择更改授权数据并接受信息弹出窗口。
- 在选择模板弹出窗口中选择不选择模板。
- 在工具栏中选择手动添加。
- 添加以下授权对象：
 - /AWS1/LROL
 - /AWS1/SESS
- 在授权树中输入：
 - 访问 AWS 的个人资料 APIs : DEMO
 - IAM 逻辑角色 : TESTUSER
- 选择保存。
- 选择生成。
- 选择返回。
- 选择保存，保存角色。

向 SAP 用户分配 PFCG 角色

任何分配了ZAWS_SDK_DEMO_TESTUSER角色的用户都将被授权使用 AWS SDK 配置文件中配置的设置的 DEMO SDK 功能。授权用户还将代入 IAM 角色，其映射在配置文件的 TESTUSER IAM 逻辑角色中。

- 运行 SU01 事务。
 - 输入将要测试 S AWS DK 功能的 SAP 用户的用户 ID。
 - 选择更改。
 - 转到角色选项卡，然后向用户分配 ZAWS_SDK_DEMO_TESTUSER 角色。
 - 选择保存。

SDK for SAP ABAP - BTP edition

创建业务角色

- 在 Web 浏览器中打开 ABAP 环境，然后导航到“维护业务角色”应用程序。
- 选择“从模板创建”，然后输入以下详细信息。

- 模板 —选择 **/AWS1/RT_BTP_ENDUSER**。
- 新的业务角色 ID-输入 ID。
- 新业务角色描述-输入描述。
- 选择“确定”以查看业务角色的页面。
- 在“常规角色详细信息”选项卡下，转到“访问类别”，然后将“写入、读取、值帮助”字段设置为“受限”。
- 选择“维护限制”，然后从左侧导航窗格中展开“分配的限制类型”。更新“限制和值”部分中的以下字段。
 - 在“选择 SDK 会话”下，选择 SDK 配置文件旁边的铅笔图标，然后导航到“范围”选项卡。输入 **DEMO**，然后选择添加。
 - 在“选择逻辑 IAM 角色”下，选择“逻辑 IAM 角色”旁边的铅笔图标，然后导航到“范围”选项卡。输入 **TESTUSER**，然后选择添加。

选择 SDK 配置文件旁边的铅笔图标，然后导航到“范围”选项卡。输入 **DEMO**，然后选择添加

- 返回到“业务角色”模板，然后打开“业务用户”选项卡。选择“添加”，将新创建的业务角色分配给将测试 SDK 功能的 SAP 业务用户。选择保存。

分配给所创建业务角色的任何业务用户都将被授权使用 AWS SDK 配置文件中配置的设置的 DEMO SDK 功能。授权用户还将代入 IAM 角色，其映射在配置文件的 TESTUSER IAM 逻辑角色中。

第 6 步：编写代码

有关更多详细信息，请参阅以下选项卡。

SDK for SAP ABAP

1. 打开 SE38 事务。
 - 输入 ZDEMO_TRANSLATE_HELLO_WORLD 作为程序名称。
 - 选择 Create。
 - 输入 AWS SDK Hello World In Any Language 作为标题。
 - 类型：选择可执行程序。
 - 状态：选择测试程序。
 - 选择保存。

- 将程序另存为本地对象。

添加以下代码。

```
*&-----*
*& 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( 'DEM0' ).
    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. 右键单击需要创建 ABAP 类的软件包，然后选择“新建”>“ABAP 类”。
2. 输入 **ZCL_DEMO_XL8_SIMPLE** 类名，然后添加班级描述。选择下一步。
3. 创建或选择传输请求。选择“完成”。

添加以下代码。

```

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.
```

有关如何编写使用 SDK 的 ABAP 代码的详细信息，请参阅[使用 适用于 SAP ABAP 的 AWS SDK](#)。

步骤 7：运行应用程序

有关更多详细信息，请参阅以下选项卡。

SDK for SAP ABAP

在 SE38 中运行应用程序。成功运行后将输出以下内容。

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

如果缺失授权、配置或基础先决条件，您会收到一条错误消息。请参阅以下示例。

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

如果您尚未向 SAP 系统配置可代入 IAM 角色的 IAM 权限，就已获得 SAP 角色授予的 SDK 配置文件使用权限，并能够将其映射到 IAM 逻辑角色，则您会收到以下输出内容。

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

此时，请查看 [the section called “第 1 步：准备 AWS 账户”](#) 中定义的 IAM 角色和/或用户上的 IAM 权限和信任配置。

SDK for SAP ABAP - BTP edition

在 Eclipse > 运行方式 > ABAP 应用程序（控制台）上运行该应用程序。成功运行后将输出以下内容。

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

如果缺失授权、配置或基础先决条件，您会收到一条错误消息。请参阅以下示例。

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

如果您尚未向 SAP 系统配置可代入 IAM 角色的 IAM 权限，就已获得 SAP 角色授予的 SDK 配置文件使用权限，并能够将其映射到 IAM 逻辑角色，则您会收到以下输出内容。

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

此时，请查看 [the section called “第 1 步：准备 AWS 账户”](#) 中定义的 IAM 角色和/或用户上的 IAM 权限和信任配置。

设置

本节介绍如何设置使用 适用于 SAP ABAP 的 AWS SDK 的开发环境。

主题

- [SAP 先决条件](#)
- [正在安装 适用于 SAP ABAP 的 AWS SDK](#)
- [安装 AWS 适用于 SAP ABAP 的 SDK-BTP 版](#)

SAP 先决条件

在托管您的 SAP 系统时，安装软件开发工具包的以下先决条件适用 AWS。

主题

- [AWS 适用于 SAP 的软件开发工具包的先决条件](#)
- [适用于 SAP ABAP 的 S AWS DK 的先决条件-BTP 版](#)

AWS 适用于 SAP 的软件开发工具包的先决条件

以下是适用于 SAP ABAP 的 AWS SDK 的先决条件。

主题

- [基础发布](#)
- [内核发布](#)
- [参数](#)
- [注意事项](#)
- [出站连接](#)
- [HTTPS 连接](#)
- [访问 Amazon EC2 实例元数据](#)

基础发布

适用于 SAP ABAP 的 SDK 与 SAP NetWeaver 7.4 及更高版本兼容。适用于 SAP ABAP 的 SDK 不含任何 SAP 应用程序表。其与应用程序完全无关，如 SAP 企业资源规划和 SAP Landscape Transformation Replication Server。

对于 SAP_BASIS 740，最低支持 SP 级别为 SP 0008。有关更多信息，请参阅 [SAP Note 1856171：支持 CL_HTTP_ENTITY 中的同名表单字段](#)（需要 SAP 门户访问权限）。您可以根据业务需求选择更高的 SP 级别，如下图所示。

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-74008INPLBASIS	Basic Plug-In

SAP_BASIS 750 及更高版本未设置最低 SP 级别要求。

内核发布

对于适用于 SAP ABAP 的 SDK 和使用 Internet Communication Manager (ICM) 进行 HTTP 连接的工具，其具备的加密、HTTP、XML 和 JSON 功能全部依赖于 SAP 内核。我们建议使用与您的 SAP NetWeaver 平台兼容的最新内核版本。最低内核版本为 741。有关更多信息，请参阅 [SAP Note 2083594：SAP 内核版本及其补丁级别](#)（需要 SAP 门户访问权限）。

如果使用 741 或 742 版本的内核，则需搭配以下补丁级别：

- 741 patchno 212
- 742 patchno 111

参数

您的 SAP 系统必须支持服务器名称指示 (SNI)，如以下 SAP 注意事项所述（需要 SAP 门户访问权限）。

- [SAP Note 2124480：ICM/Web 调度程序：作为客户端的 TLS 扩展服务器名称指示 \(SNI\)](#)

- [SAP Note 2582368 : 通过 saphttp、sapkprotp 和 sldreg 在客户端发送 TLS 扩展 SNI 的 SapSSL 更新](#)

在 DEFAULT.PFL 文件中配置以下参数。

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

注意事项

为系统应用以下 SAP 注意事项。

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

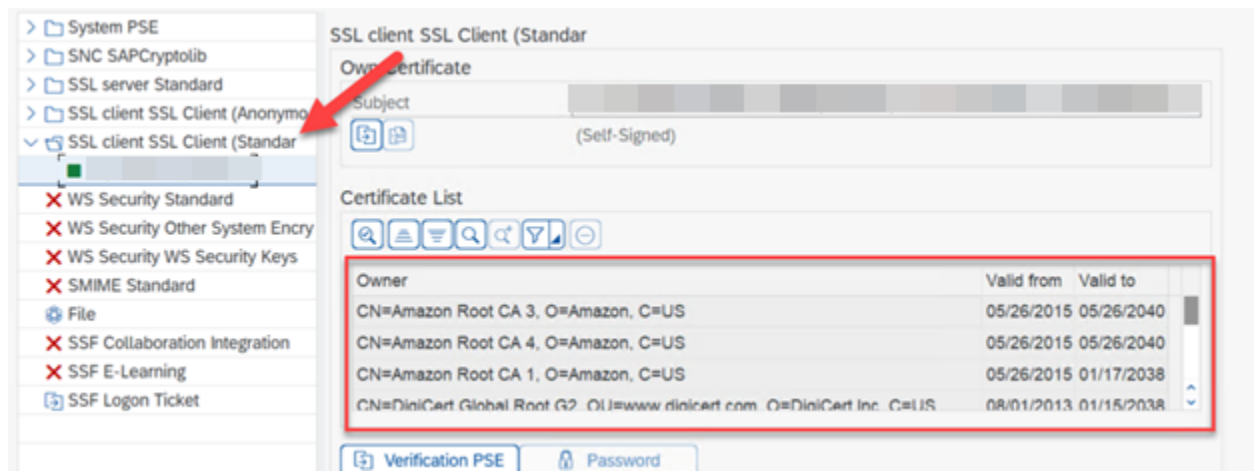
出站连接

适用于 SAP ABAP 的 SDK 是 HTTPS 客户端。SAP 系统会发送 HTTPS 出站消息。此操作无需入站连接。

HTTPS 连接

所有 AWS API 调用均使用加密的 HTTPS 通道进行。必须将 SAP 系统设置为信任 AWS 证书才能建立出站 HTTPS 连接。

1. 转到 <https://www.amazontrust.com/repository/>。
2. 在根目录下 CAs，使用 PEM 链接下载所有证书。
3. 将这些证书导入每个 SAP 系统的 SSL Client (Standard) PSE 的 STRUST 中，如下图所示。



访问 Amazon EC2 实例元数据

ABAP 系统与本地主机 (http://169.254.169.254) 建立未加密的 HTTP 连接，以启用亚马逊 EC2 实例元数据。HTTP 通道仅用于从本地服务器检索 AWS 凭证。HTTP 流量留存在主机内。

元数据允许 SAP 系统安全地 AWS 进行身份验证，而无需在 SAP 安全存储区中存储密钥。此功能仅适用于亚马逊上托管的 SAP 系统 EC2。

使用以下参数配置 DEFAULT.PFL 文件，为 SAP 系统建立未加密的 HTTP 出站连接。

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

在不启用入站连接的情况下，使用以下参数启用 HTTP 出站连接。

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

请按照以下步骤，验证经过配置的 SAP 系统能够启用 HTTP 出站连接：

- 1. 运行 SMICM 事务。
- 2. 转到活动服务。
- 3. 验证您是否在活动列下的 HTTP 行中看到了绿色复选标记。

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

适用于 SAP ABAP 的 S AWS DK 的先决条件-BTP 版

以下是适用于 SAP ABAP 的 AWS SDK-BTP 版本的唯一先决条件。

主题

- [SAP 景观门户 — BTP 版](#)
- [SAP 凭证存储 — BTP 版](#)

SAP 景观门户 — BTP 版

此先决条件仅适用于适用于 SAP ABAP 的 S AWS DK-BTP 版本。

SAP 景观门户是唯一支持在 SAP BTP 环境中安装插件的机制。确保您已订阅使用此服务。有关更多信息，请参阅[景观门户](#)。

SAP 凭证存储 — BTP 版

此先决条件仅适用于适用于 SAP ABAP 的 S AWS DK-BTP 版本。

在开发者预览版中，秘密访问密钥身份验证是唯一支持对 SAP ABAP 的 AWS SDK 进行身份验证-BTP 版本的机制。SDK 从凭证存储区读取凭证，并安全地存储私有访问密钥。

您必须满足以下先决条件。

- 订阅凭证存储。
- 凭据存储作为权限分配给您的 BTP 子账户。有关更多详细信息，请参阅[初始设置](#)。
- 具有凭据存储标准套餐的服务实例。有关更多详细信息，[请参阅创建服务实例](#)。

有关更多信息，请参阅[使用 SAP 凭据存储](#)。

SAP 凭证存储服务在 ABAP BTP 系统之外的 SAP BTP 中运行。有关更多详细信息，请参阅[SAP 凭据存储](#)。

正在安装 适用于 SAP ABAP 的 AWS SDK

主题

- [下载适用于 SAP ABAP 的 SDK](#)

- [验证适用于 SAP ABAP 的 SDK 文件 – 可选](#)
- [AWS SDK 传输](#)

下载适用于 SAP ABAP 的 SDK

从 <https://sdk-for-sapabap.aws.amazon.com/awsSdkSapabapV1/release/abapsdk-LATEST.zip> 下载 SDK。

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

下载完成后，建议将下载的文件解压到目录，如 /tmp/awssdk。

验证适用于 SAP ABAP 的 SDK 文件 – 可选

此可选步骤可验证 SDK 文件签名，以确认 SDK 未被篡改。使用以下步骤验证 SDK 文件。

1. 按照以下命令下载 SDK 签名文件。

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

2. 复制以下公钥并将其保存到 abapsdk-signing-key.pem 文件。

```
-----BEGIN PUBLIC KEY-----
MIICIjANBgkqhkiG9w0BAQEFAAACAg8AMIICCGKCAgEAmS3oN3wKBh4HJ0Ga0tye
15RR5909nuw0Jx0vEDCT709wUrXS3mjgEw6b6hvr2dLdoFr+eH4ewT5bV16U3gDv
051sTdEJJpfLEWJJZZNk3v9fGWkyXgYe+ifmsPmf4lhNd2auzpvIy2UrlSYijCRB
BWZFW+Ux00kILz+8vCFSXMZ6Z0qtLI1ZFbGrn6A5adbwwzf0qkg9BUEZK0wB6TAi
ZTnkMdBZGCBM9K2MRKKMxtrxUn+TFcAYyh5pM9tUAb2q4XE5m7092UnZG7ur/QY1
1FSZwAhQmk8hUPgUaq00QRC6z3TRzIGK0A/DI0cUPJMzFR4LCxEJkgh4rkRaU9V2
07DthUpj8b7QcQai0pnMpBf3zWLgbjNmX0hB0Eprg8/nVRHspf3zuiscJ21MPkz0
cH0R31MNsMLzm+d/gVklT31R/JwAcFCkXTWvR8/V0WNGZZXdVubefrI/k7fP60B
bzUrI1N4poq16rc4Tk5Derg+wQ7r0WjXkXop2kiCMjbYo0o10kS/At64PLj pz8dH
Zg25o79U9EJ1n+lpqZ297Ks+Hoct0v2GPbeeh0s7+N0fRTy0r81EZIURLPKLVQUw
otVRzNDgLOA7eA667NrmegZfHCmqEwK9tXakZUHAcMzRPyhALc/HtmovxdStN9h1
JC4ex0GqstAv1fX5QaTbMSECAwEAAQ==
-----END PUBLIC KEY-----
```

3. 运行以下命令以验证下载的 SDK ZIP 文件。此命令需要用到许多 Linux 分发版中的 openssl。

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

4. 验证上述命令的输出内容是否为 Verified OK。
5. 如果是 Verification Failure，则重复上述步骤。如果持续收到输出失败的消息，则取消安装 SDK 并联系支持。

AWS SDK 传输

主题

- [内容](#)
- [导入](#)
- [命名空间](#)

内容

通过 ABAP 传输来安装适用于 SAP ABAP 的 SDK。请务必将这些传输内容导入开发环境或沙盒环境。

每个适用于 SAP ABAP 的 SDK 版本都完全取代上一版本。无需应用增量传输。所有传输内容均捆绑在 ZIP 文件中，ZIP 文件结构如下。

```
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  
.  
.  
.
```

transports 文件夹中含有 core 子文件夹。core 子文件夹中含有运行时核心传输内容与各模块的子文件夹，以模块的三字母缩写命名。有关完整的模块列表 TLAs，请参阅 [适用于 SAP ABAP 的 AWS SDK - 模块列表](#)。

AWS SDK 传输是工作台请求。根据 TMS 路由配置，在将 SDK 导入之前的系统后，SDK 可能无法自动转发到质量保证和生产队列。必须手动添加到每个系统的队列中。

当您的项目为下一阶段做好准备时，可以导入 AWS SDK 以及包含您自己的具有业务功能的 Z 代码的单独传输。如果使用 SAP 更改请求管理 (ChaRM) 等更改控制系统，请咨询 ChaRM 管理员如何正确处理第三方传输内容。

导入

主题

- [关键提示](#)
- [开始导入](#)

AWS SDK 传输独立于客户端。核心传输是必需的，它包含 SDK 运行时代码、适用于 Amazon 简单存储服务的 API 和适用于 Amazon 简单存储服务的 API。AWS Security Token Service 其余的 SDK 模块均以单独传输内容进行传输。为保持系统内的小型 SDK，每个 SDK 模块都是可选的。您也可以按照自己的业务逻辑，稍后安装其他模块。

例如，如果您想将用 APIs 于 Amazon S3 和 Amazon Translate，请导入 core 传输（包含核心运行时、Amazon S3 和 AWS STS 模块）和 x18 传输（包含用于的模块 Amazon Translate）传输。

要查看适用于 SAP ABAP 的 SDK 的完整列表 APIs，请参阅 [适用于 SAP ABAP 的 SDK — API 参考指南](#)。

以下是导入 AWS SDK 传输时的关键提示。

- 每次以 Knnnnnnn.AWS 和 Rnnnnnnn.AWS 进行传输
 - 必须复制 Knnnnnnn.AWS 到 /usr/sap/trans/cofiles
 - 必须复制 Rnnnnnnn.AWS 到 /usr/sap/trans/data
- 导入传输内容时，必须在导入传输请求 > 选项 > 导入选项中选择忽略无效组件版本。
- 可以同时导入所需的传输内容。
- 单独导入传输内容时，必须先导入 core 传输内容。
- 必须采用相同的传输版本级别。

开始导入

AWS SDK 传输可能需要很多分钟才能导入。如果 STMS 显示绿灯 (RC=0) 或黄灯 (RC=4)，即为传输成功。

- 红灯 (RC=8) 表示导入时出现语法错误。
 - 选择请求 → 显示 → 日志，检查导入错误。
 - 如果在导入时因缺失接口 IF_SYSTEM_UUID_RFC4122_STATIC 而发生错误，请确保对系统应用 SAP Note 2619546。有关更多信息，请参阅[注意事项](#)。
 - 如果错误原因不明，请与联系 支持。
- 红色闪电 (RC=12) 表示传输文件尚未正确加载到 /usr/sap/trans，或缺少必要权限。

关键提示

以下是导入 AWS SDK 传输时的关键提示。

- 每次以 Knnnnnn.AWS 和 Rnnnnnn.AWS 进行传输
 - 必须复制 Knnnnnn.AWS 到 /usr/sap/trans/cofiles
 - 必须复制 Rnnnnnn.AWS 到 /usr/sap/trans/data
- 导入传输内容时，必须在导入传输请求 > 选项 > 导入选项中选择忽略无效组件版本。
- 可以同时导入所需的传输内容。
- 单独导入传输内容时，必须先导入 core 传输内容。
- 必须采用相同的传输版本级别。

开始导入

AWS SDK 传输可能需要很多分钟才能导入。如果 STMS 显示绿灯 (RC=0) 或黄灯 (RC=4)，即为传输成功。

- 红灯 (RC=8) 表示导入时出现语法错误。
 - 选择请求 → 显示 → 日志，检查导入错误。
 - 如果在导入时因缺失接口 IF_SYSTEM_UUID_RFC4122_STATIC 而发生错误，请确保对系统应用 SAP Note 2619546。有关更多信息，请参阅[注意事项](#)。
 - 如果错误原因不明，请与联系 支持。
- 红色闪电 (RC=12) 表示传输文件尚未正确加载到 /usr/sap/trans，或缺少必要权限。

命名空间

适用于 SAP ABAP 的 SDK 会使用 /AWS1/ 命名空间，但不会修改系统中的 SAP 对象或其他任何对象，以下情况除外。

- AWS auth对象位于身份验证对象类中。身份验证对象类仅限四字符，并且不支持命名空间。适用于 SAP ABAP 的 SDK 使用 YAW1 身份验证对象类。如果您 YAW1 在事务中已经有身份验证对象类 SU21，支持 请在安装前联系。

安装 AWS 适用于 SAP ABAP 的 SDK-BTP 版

BTP 版本为开发者预览版，可通过加入预览版进行安装。要安装 SDK，请填写适用于 [SAP ABAP 的 AWS SDK-BTP 版开发者预览版](#) 中的参与表格。

在安装适用于 SAP ABAP 的 SDK-BTP 版本之前，请确保满足所需的先决条件。有关更多信息，请参阅 [SAP 景观门户](#) 和 [SAP 凭据存储](#)。

主题

- [安装适用于 SAP ABAP 的 SDK-BTP 版](#)
- [模块](#)
- [适用于 SAP ABAP 的补丁 SDK-BTP 版](#)

安装适用于 SAP ABAP 的 SDK-BTP 版

1. 转到你的 SAP 景观门户实例，然后启动 Deploy Product fiori 应用程序。
2. 在“产品”中，在“合作伙伴产品” /AWS1/SDK_OMNI 下选择。

支持 如果您在开发者预览版中被接受 /AWS1/SDK_OMNI 后没有看到，请联系我们。

3. 在目标版本中，选择要在系统上安装的适用于 SAP ABAP 的 SDK 版本-BTP 版本。
4. 在“可用系统”中，选中所有要安装软件开发工具包的复选框。SIDs
5. 选择部署，输入计划详细信息，然后选择计划。您可以在“产品版本部署状态”中监控进度。

安装可能需要 30-45 分钟，包括系统停机时间。有关更多详细信息，请参阅[部署产品](#)。

模块

适用于 SAP ABAP 的 AWS SDK-BTP 版本的开发者预览版中包含以下模块。

- [亚马逊 API Gateway \[agw\]](#)
- [亚马逊 Athena \[\] ath](#)
- [亚马逊 Bedrock Runtime \[\] bdr](#)
- [亚马逊 Comprehend \[\] cpd](#)
- [亚马逊 EventBridge \[evb\]](#)
- [Amazon Forecast \[fcs\]](#)
- [亚马逊 Kinesis \[\] kns](#)
- [亚马逊 Data Firehose \[\] frh](#)
- [亚马逊 SageMaker AI \[sgm\]](#)
- [Amazon 简单通知服务 \[sns\]](#)
- [Amazon 简单队列服务 \[sqs\]](#)
- [亚马逊简单存储服务 \[s3\]](#)
- [AWS Systems Manager \[ssm\]](#)
- [亚马逊 Textract \[\] tex](#)
- [Amazon Transcribe \[\] tnb](#)
- [Amazon Translate \[x18\]](#)
- [AWS CloudTrail \[trl\]](#)
- [AWS IoT \[iot\]](#)
- [AWS KMS \[kms\]](#)
- [AWS Lambda \[lmd\]](#)
- [AWS Secrets Manager \[smr\]](#)
- [AWS Security Token Service \[sts\]](#)
- [AWS Transfer Family \[trn\]](#)
- [任何地方的 IAM 角色 \[rla\]](#)
- [亚马逊 Redshift 数据 API \[\] rsd](#)

适用于 SAP ABAP 的补丁 SDK-BTP 版

适用于 SAP ABAP 的 SDK-BTP 版本的修补过程与安装过程类似。如果您在已经安装了旧版本的系统上安装了 SDK，则该软件开发工具包会被修补到您选择的新版本。

正在配置 适用于 SAP ABAP 的 AWS SDK

在使用之前 适用于 SAP ABAP 的 AWS SDK，必须使用 SDK 操作所需的技术和功能设置来配置 SDK。设置分为可传输型和运行时型。许多设置直接类似于 .INI 文件中为其他设置定义的设置。SDKs

SDK 配置（运行时设置除外）必须在您的开发环境中完成。您可以根据传输和变更控制的常用规则，将配置传输到 QA 和生产环境。不建议在生产环境中使用便携式配置。

如果您没有配置 AWS SDK 的权限，请参阅 [S AP 授权](#)。

正在配置 适用于 SAP ABAP 的 AWS SDK

如需运行配置事务，请在 SAPGUI 命令栏中输入 /n/AWS1/IMG。

为 SAP ABAP 配置 S AWS DK-BTP 版

使用以下步骤为 SAP ABAP-BTP 版本配置 SDK。

1. 在网络浏览器中打开您的 ABAP 环境。
2. 导航到“自定义业务配置”应用程序。

要使用“导出自定义传输”应用程序创建自定义请求，请参阅[使用导出自定义传输](#)应用程序-创建请求。

在“自定义业务配置”应用程序中，您可以根据 SDK 设置的类型对配置进行分组。使用以下步骤对配置进行分组。

1. 在 Web 浏览器中打开 ABAP 环境，然后导航到“自定义业务配置”应用程序。
2. 选择设置 > 群组，然后从下拉列表中选择配置组。选择确定。
3. 如图所示，这些配置现在以分层结构提供。要保存视图，请参阅[图（变体管理）-组件](#)。

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	>

本节涵盖以下主题。

主题

- [全局设置](#)
- [应用程序配置](#)
- [运行时设置](#)
- [高级连接场景](#)
- [服务提供商设置](#)
- [刷新、追踪和遥测主题 适用于 SAP ABAP 的 AWS SDK](#)

全局设置

使用适用于 SAP ABAP 的 AWS SDK 的 /n/AWS1/IMG IMG 交易和适用于 SAP ABAP 的 AWS SDK 的自定义业务配置应用程序——BTP 版本来配置全局设置。本主题可以互换使用 IMG 和自定义业务配置。

本节涵盖以下主题。

主题

- [技术设置](#)

- [配置场景](#)

技术设置

/AWS1/IMG 事务的全局设置会影响 SDK 的整体行为。这些设置通常由基础管理员负责配置。您可以根据以下建议设定这些值。

- 选择新建条目。
 - S3 区域化：通过 s3.amazonaws.com 访问 us-east-1 存储桶。
 - STS 区域化：使用全局端点访问 STS。
 - 禁用 EC2 元数据：将此字段留空。在 BTP 版本中，此字段为只读字段，默认设置为“是”。
 - 元数据端点模式：使用 IPv4 元数据端点。此字段在 BTP 版本中为只读字段，并且会自动更新。
 - 元数据端点 URL：留空此字段。在 BTP 版本中，此字段为只读字段。
- 选择保存。

配置场景

场景可启用 AWS SDK，以在多区域灾难测试或灾难恢复测试场景中更高效地切换设置。您可能无需使用此功能，只要配置以下默认场景即可。

- 选择新建条目。
 - 场景 ID：DEFAULT
 - 场景描述：默认场景
- 选择保存。

如果存在多区域灾难恢复设置，或出现需要快速更改设置的其他特殊情况，则可以配置多个场景。

- DEFAULT：标准操作。
- DR：因为灾难而需要将整个系统移动到另一个区域时采用的特殊配置。
- DR_TEST：用于模拟灾难的特殊配置，如产品临时克隆。

应用程序配置

适用于 SAP ABAP 的 SDK 的配置流程与其他 ABAP 应用程序相似。可整理为不同配置文件，对各种场景设置进行分组。ABAP SDK 配置文件定义了特定应用场景所需的设置。例如，如果交易、ZVA01ZVA02、和ZVA03增强了与发票相关的交易并在上运行 AWS 服务，例如 Amazon S3 AWS Lambda Amazon SageMaker AI、和，则ZINVOICE可以创建名为的 SDK 配置文件。发票相关功能的配置文件可对技术设置、SAP 授权和 IAM 角色映射进行分组。

使用适用于 SAP ABAP 的 S AWS DK 的 /n/AWS1/IMG交易和适用于 SAP ABAP 的 AWS SDK 的自定义业务配置应用程序——BTP 版本来配置全局设置。本主题可以互换使用 IMG 和自定义业务配置。

主题

- [SDK 配置文件](#)
- [逻辑资源解析程序](#)
- [示例](#)

SDK 配置文件

ABAP SDK 配置文件为每个 SID 和客户端定义以下内容。

Note

在 SAP BTP、ABAP 环境中，客户端始终为 100。

- 所有 API 调用的默认 AWS 区域。例如，如果您的 SAP 系统在该us-east-1区域运行，则您的其他 AWS 资源很可能也位于同一区域，而这应该是您的默认区域。可以用 ABAP 代码覆盖默认区域。
- 身份验证方法
 - 对于在 Amazon 上运行的 SAP 系统 EC2，我们强烈建议您选择实例角色元数据，以便从短暂的、自动轮换的证书中受益。
 - 对于运行在本地或其他云中的 SAP 系统，请务必选择 SSF 存储中的凭证。
 - 对于在 SAP BTP 上运行的 ABAP 系统，您必须从 SAP 凭据存储中选择凭据。有关更多信息，请参阅[使用 SAP 凭据存储进行身份验证](#)。
- 向 IAM 角色映射 IAM 逻辑角色。
 - 此映射操作按优先级降序排序。
 - 系统将自动为用户选择其有权代入的 PFCG 角色中优先级最高的 IAM 角色。

Note

在 SAP BTP、ABAP 环境中，PFCG 角色被称为业务角色。

当 ABAP 程序想要连接时 AWS 服务，它将指定一个提取必要设置的 ABAP SDK 配置文件。通过执行 AUTHORIZATION-CHECK 确认用户有权访问 SDK 配置文件。SAP 安全管理员可定义 PFCG 角色，授予您访问相应用户的权限。

逻辑资源解析程序

逻辑资源解析程序可提供用于存储资源名称的标准位置。其使用适用于 SAP ABAP 的 SDK 进行传输。这种操作方式类似于 FILE 事务映射逻辑文件名称到物理文件名称的方式。

逻辑资源定义了资源的概 AWS 念，例如存放我们发票的 Amazon S3 存储桶。以此逻辑资源为例，可以将其命名为 ZINVOICES_OUTBOUND，也可以将其映射到不同的物理存储桶名称，这取决于 SAP 系统到底是开发系统、QA 系统还是生产系统。

适用于 SAP ABAP 的 SDK 经过设置后，QA 系统可将逻辑资源解析为 QA 物理资源，即使刷新生产系统也能操作。SAP 开发系统可定义所有系统的资源映射，并将其传输至前端。这种方法不同于 SAP 系统中将映射作为主数据处理并设置于各个系统的常规设置。适用于 SAP ABAP 的 SDK 具备的逻辑资源解析程序的优势在于，刷新系统后几乎不可能发生传输错误。

示例

共有四个单独的 Amazon S3 存储桶，分别是开发存储桶、生产存储桶、QA 存储桶和用于回归测试的第二个 QA 存储桶。

SDK 在将 ZINVOICE_OUTBOUND 等逻辑资源解析为物理资源时，会检查 SY-SYSID 和 SY-MANDT 并提问现在运行在哪个 SID 和客户端中？，然后自动选择正确的物理资源。

如需更改生产资源映射，必须更改开发系统中的 IMG 映射，然后将其传输至前端。这样可以确保将 AWS 资源重新分配给 SAP 系统时与任何其他传输一样受到变更控制。

Note

由于 SDK 配置取决于客户端，因此必须在自定义请求中传输重新分配的资源，并将此传输操作导入到各客户端。

运行时设置

本节涵盖以下主题。

Note

这些设置不可传输，均存储在各 SAP 系统的本地设置中。

主题

- [记录和跟踪](#)
- [选择加入：增强型遥测功能](#)
- [活动场景](#)

记录和跟踪

您可以出于调试目的激活跟踪。除非诊断出技术问题，否则建议您始终设置跟踪级别为无跟踪。有关更多信息，请参阅《安全操作》。

这些设置不适用于适用于 SAP ABAP 的 SDK-BTP 版本。

选择加入：增强型遥测功能

所有人都会 SDKs 将遥测信息发送给 AWS 以提供支持。您可以选择加入增强型遥测功能。当您联系支持 以确定特定 API 调用的来源时，这特别有用。有关更多信息，请参阅[跟踪](#)和[遥测](#)。

这些设置不适用于适用于 SAP ABAP 的 SDK-BTP 版本。

活动场景

激活此事务中的 DEFAULT 场景。每个系统激活一次场景后就无需再更改，除非系统进行多区域灾难恢复。在多区域设置中，您可以使用此设置将 SAP 系统切换到灾难恢复环境或灾难恢复测试场景。

高级连接场景

适用于 SAP ABAP 的 AWS SDK AWS 服务 通过对 AWS 端点进行 HTTPS 调用来消耗。通常可通过互联网访问 AWS 端点。SAP 系统必须联网才能建立出站连接。适用于 SAP ABAP 的 SDK 从不需要在互联网和 SAP 系统之间建立入站连接。

以下场景提供了不同的出站连接建立方法。

场景

- [通过代理服务器连接](#)
- [通过数据包检查防火墙连接](#)
- [网关端点](#)
- [自定义接口端点](#)
- [多区域端点访问](#)

通过代理服务器连接

如需通过代理服务器建立连接，请遵循以下步骤。

1. 进入 SDK，转到 **SICF** 事务。
2. 选择执行。
3. 进入菜单，选择客户端 > 代理服务器。
4. 将代理设置设为活动状态。
5. 在以下地址无代理字段中，用分号列出所有异常。
6. 在 HTTP 协议和HTTPs 协议字段中，指定代理服务器的连接详细信息。

SDK 虽然不知道代理服务器的存在，但无需任何设置就能使用 SAP 系统的代理服务器配置。

Note

如果您使用 [Amazon EC2 实例元数据身份验证](#)，则 SAP 系统无法使用代理服务器访问位于的本地实例元数据 `http://169.254.169.254`。请务必在以下地址无代理字段中加入 `169.254.169.254`。

通过数据包检查防火墙连接

您可以配置数据包检查防火墙，用于建立出站连接。这些防火墙可解密并重新加密 SSL 流量，然后将其传递到端点。此配置通常要求防火墙向使用 AWS 服务的 SAP 系统签发自己的证书。必须将防火墙的 CA 证书安装在 STRUST 中。有关更多信息，请参阅 [HTTPS 连接](#)。

网关端点

有些 AWS 服务 提供网关终端节点，以便在没有互联网的情况下为 VPC 提供高性能访问。这些端点对适用于 SAP ABAP 的 SDK 透明，无需进行任何配置。

有关更多信息，请参阅[网关端点](#)。

自定义接口端点

如需使用自定义端点覆盖默认端点解析，则可以使用接口端点为 VPC 提供无需联网的高性能访问权限。有关更多信息，请参阅[配置接口端点](#)。

这些端点不使用私有 DNS 时，会使用专有的 DNS 地址，同时 ABAP 程序必须显式覆盖常用的端点解析逻辑。有关更多信息，请参阅 AWS re:Post — [为什么我无法解析接口 VPC 终端节点的服务域名？](#)

在以下示例中，为 AWS STS 和创建了一个接口终端节点 Amazon Translate。SAP 系统没有使用私有 DNS，而是使用自定义端点调用服务。/AWS1/IMG 中定义的逻辑资源表示的是物理接口端点地址，如 `vpce-0123456789abcdef-hd52vxz.translate.us-west-2.vpce.amazonaws.com`。这样可以避免在代码中对 DNS 进行硬编码。

在以下代码中，首先将 /AWS1/IMG 中的逻辑资源解析为物理端点名称。然后将它们提供给 AWS 会话类（用于 AWS STS 担任 IAM 角色）和翻译 API 类的工厂方法。

```
" 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...

```

如示例所示，所有调用 go_xl8 的方法都会转到端点 `https://vpce-0123456789abcdefg-12345567.translate.us-west-2.vpce.amazonaws.com`。

多区域端点访问

AWS 终端节点是根据您在 SDK 配置文件中定义的默认值 AWS 区域自动确定的。您也能以编程方式指定区域以覆盖默认区域。可使用 CREATE() 工厂方法或者之后使用 SDK 配置对象来覆盖此默认区域。有关更多信息，请参阅[编程配置](#)。

以下示例使用 CREATE() 工厂方法设置区域，并在 us-east-1 和 us-west-2 区域中列出 Amazon SQS 队列。

```

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.

```

服务提供商设置

Basis 管理员有时需要从客户端控制整个系统中 SDK 的某些功能。对于代表客户使用自己的 AWS 账户操作系统的托管和服务提供商来说，这是一种常见的情况。AWS 适用于 SAP 的 SDK ABAP 支持服务提供商设置。这些设置是在客户端中配置的，会影响所有客户端上的 SDK。适用于 SAP 的 SDK ABAP-BTP 版本不支持服务提供商设置。

服务提供商设置是在事务中配置的 /AWS1/IMG，并且必须在客户端中配置。其他客户端中的服务提供商设置将被忽略。客户端中的设置在所有客户端上生效，如果发生冲突，则会取代其他 IMG 设置。

使用以下步骤在客户端中配置服务提供商设置。

1. 在事务中展开服务提供商设置分支 /AWS1/IMG。
2. 选择服务提供商护栏
3. 选择“新建条目”，然后根据您的业务需求调整设置。
 - 禁用 EC2 元数据-阻止 SDK 访问所有客户端中的 EC2 实例元数据，即使 SDK 配置文件配置为使用 EC2 实例元数据进行身份验证。如果 ABAP 程序尝试使用 SDK 访问实例元数据，则 SDK 会引发异常。
4. 选择保存。

刷新、追踪和遥测主题 适用于 SAP ABAP 的 AWS SDK

本节涵盖以下主题。

主题

- [SAP 系统刷新](#)
- [跟踪](#)
- [遥测](#)

SAP 系统刷新

系统刷新后，如何保证不同系统不会访问彼此的资源是基础管理员面临的主要挑战。例如，您可能想阻止 QA SAP 系统访问生产场景中的资源，如 S3 存储桶。

为应对这一挑战，适用于 SAP ABAP 的 SDK 提出了一种具有安全意识的逻辑资源法。业务分析师可以采取以下步骤。

1. 定义 ZINVOICE_OUTBOUND 等逻辑资源。
2. 映射开发系统中的所有系统和客户端。
3. 将所有系统配置向前传输到生产场景。

刷新后的基础步骤

1. 检查身份验证信息

- 如果系统正在使用秘密访问密钥进行身份验证，则存储在主数据中的 SSF 加密凭证将会失效。必须重新输入证书，这可能需要在中重新生成新的私有访问密钥。<https://console.aws.amazon.com/iam/>
- 如果系统正在使用 EC2 实例元数据进行身份验证，则无需执行任何步骤。

检查跟踪设置

- 确保 /AWS1/IMG 中的跟踪设置满足需求。此类设置不可传输。

跟踪

IMG 运行时设置会控制跟踪输出。

可供使用的跟踪级别有：

- 无跟踪
- 跟踪 API 调用
- 跟踪 API 调用和有效负载

此选项包含未加密的有效负载信息。

- 跟踪 API 调用、有效负载和 XML 内部转换

此选项包含未加密的有效负载信息。

如果激活 API 跟踪，则会将跟踪写入 `aws1_trace-YYYY-MM-DD.log` 文件中的 DIR_WORK。

如果再激活有效负载跟踪，则需要为每个调用和有效负载组件创建以 `aws1_payload_*` 为标题的附加文件。可以使用适用于每个失败的单独有效负载跟踪的长度限制，对有效负载的跟踪长度进行限制。

Payload trace 主要用于收集 支持 在发生序列化错误时要提供的信息。除非您正在尝试诊断 SDK 错误，否则亚马逊建议您选择无跟踪。

Note

有效负载跟踪包含未加密的业务信息。我们建议仅在 Support 请求时开启这些跟踪以帮助您进行故障排除。AWS 您可以在解决问题后关闭跟踪。由于跟踪不会自动删除，当不再需要此类跟踪时，由系统管理员手动删除。

这些设置不适用于适用于 SAP ABAP 的 SDK-BTP 版本。

遥测

SDKs 将遥测信息发送到。支持适用于 SAP ABAP 的 SDK 负责收集以下信息：

- 操作系统版本和补丁级别
- SAP_BASIS 版本和补丁级别
- SAP 内核版本和补丁级别

您可以选择向 支持 发送以下信息。

- SAP SID 和实例名称 (host_sid_nn)
- SAP 客户端 (SY-MANDT)
- 事务代码 (SY-TCODE) 和报告 (SY-REPID)

附加信息可以 支持 更好地帮助您。支持 可以检测某个 API 调用的原因，并可以进一步在 SAP 系统中找到相关事务。

遥测仅限于适用于 SAP ABAP 的 SDK 和 API 版本——BTP 版本。

使用 适用于 SAP ABAP 的 AWS SDK

适用于 SAP ABAP 的 SDK 包含两个主要组件。

- SDK 运行时系统 (/AWS1/RT 包) : 一组支持安全、身份验证、跟踪、配置、数据转换和其他跨 API 功能的对象。Amazon S3 AWS STS、IAM Roles Anywhere 和 Secrets Manager 的 API 模块是必需的。
- APIs (软件包/AWS1/API及其子包) — 每个 API 的子包，其中每个 API 的对象彼此完全独立，确保一个 API 的更改不会破坏另一个 API。要查看完整列表 适用于 SAP ABAP 的 AWS SDK APIs，请参阅 [适用于 SAP ABAP 的 AWS SDK -API 参考指南](#)。

本节涵盖以下主题。

主题

- [ABAP 中的数据表示](#)
- [Amazon S3 示例程序](#)
- [适用于 SAP ABAP 的 SDK 概念](#)
- [适用于 SAP ABAP 的 AWS SDK features](#)
- [使用 SDK 构建产品](#)
- [将 HTTP 请求自定义为 AWS](#)
- [限制](#)

ABAP 中的数据表示

本节涵盖以下主题。

主题

- [数据类型](#)
- [AWS 数据类型](#)

数据类型

AWS 服务 有一组必须映射到 ABAP 数据类型的标准数据类型。有关更多信息，请参阅下表。

AWS 数据类型	ABAP 数据类型	评论
布尔值	C	单字符 "X" 和 " "
字符串	string	
字节	INT2	INT2 的范围大于 0-255。大多数 AWS 服务 会截断溢出，但这种行为尚未正式定义。
短型	INT2	
整数	INT4	
长整型	DEC19	INT8 直到 ABAP 750 才可用。DEC19 用于在所有支持的 ABAP 平台之间实现兼容性和一致性。
Blob	XSTRING	表示二进制数据
浮点型	STRING	虽然 ABAP 支持 DECFLOATs，但它不能表示诸如 NaN、Infinity 和 -Infinity 之类的值。AWS SDK 在内部将它们表示为 STRINGs，并在运行 DECFLOAT16 时将其转换为。如果 NaN、Infinity 或 +Infinity 得以表示，则开发人员可以根据一组特殊异常或映射来处理这些值。
双精度	string	
bigInteger	string	这些值表示无法在 ABAP 中表示的无限长度数字，用于代替 BigInteger STRINGs。
bigDecimal	string	
Timestamp	TZNTSTMPs	TZNTSTMPs 允许使用 ABAP 原生时间戳函数进行处理。

AWS 服务 还会返回以下聚合数据类型。

AWS 数据类型	ABAP 数据类型	评论
结构	类	
Union	类	并集的字集数量始终不超过一个，除此以外，它与结构一模一样。其他所有字段均设为无值。
数组	标准表	
哈希	哈希表	经过哈希处理的表只有两列：键列（字符串）和值列（类）。

AWS 数据类型

已在 ABAP AWS 服务 中集成了以下方法来提供支持。

- 某些 AWS 数据类型无法在 ABAP 中表示。例如，以 ABAP 语言表示的 float 数据类型不支持 NaN、Infinity 或 -Infinity 值。因此将 float 数据类型表示为 STRING，并在运行时将其翻译为 DECFLOAT16。
- AWS 数据在线上以 JSON 或 XML 的形式表示，并且这些值是可选的。例如，请参阅以下由 JSON 格式返回 AWS 服务 的示例。

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

如果 Ana 没有中间名，则此服务会返回以下输出内容。

```
Fullname: {
```

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

ABAP 并不区分长度为 0 的字符串和无值的字符串。其他语言可能会向字符串分配 NULL 值，或者对构造函数中的字符串进行包装（如 Java 的 `Optional<>` 包装程序）。ABAP 不支持上述操作。因此，适用于 SAP ABAP 的 SDK 可以通过提供 getter 方法的变体，有效区分这些值。

Amazon S3 示例程序

本节将使用简单的示例程序引导您调用 `ListObjectsV2`，以列出 Amazon S3 存储桶中的内容。

主题

- [先决条件](#)
- [代码](#)
- [代码部分](#)

先决条件

您必须满足以下先决条件才能运行此示例程序。

- 您拥有 Amazon S3 存储桶。将本教程中的存储桶命名为 `demo-invoices.customer.com`。
- `/AWS1/IMG` 事务：
 - 将已定义的 SDK 配置文件命名为 `DEMO_S3`。
 - 必须将 SDK 配置文件中的 IAM 逻辑角色 `TESTUSER` 映射到 IAM 角色，如 `arn:aws:iam::111122223333:role/SapDemoFinance`，其授予该角色列出 Amazon S3 存储桶内容的 `s3:ListBucket` 权限。
 - 将逻辑资源命名为 `DEMO_BUCKET`，其通过 SAP 系统的 SID 和客户端映射到 Amazon S3 存储桶。
- 您的用户拥有 `PFCG` 角色，其：
 - 授予用户通过身份验证对象 `/AWS1/SESS` 访问 `DEMO_S3` SDK 配置文件的权限。
 - 授予用户通过身份验证对象 `/AWS1/LR0L` 访问 IAM 逻辑角色 `TESTUSER` 的权限。
- 您的 SAP 系统可以 AWS 使用 SDK 配置文件中定义的方法进行身份验证。

- 您的 Amazon EC2 实例配置文件授予您的 SAP 系统使用 `sts:assumeRole` 软件开发工具包配置文件中 `arn:aws:iam::111122223333:role/SapDemoFinance` 映射的 IAM 角色的权利。

代码

以下代码块展示了您的代码情况。

```
REPORT  zdemo_s3_listbuckets.

START-OF-SELECTION.
  PARAMETERS pv_lres TYPE  /aws1/rt_resource_logical
                DEFAULT 'DEMO_BUCKET' OBLIGATORY.

  DATA(go_session) = /aws1/cl_rt_session_aws=>create( 'DEMO_S3' ).
  DATA(gv_bucket)  = go_session->resolve_lresource( pv_lres ).

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

  TRY.
    DATA(lo_output) = go_s3->listobjectsv2(
      iv_bucket = CONV string( gv_bucket )
      iv_maxkeys = 100
    ).
    LOOP AT lo_output->get_contents( ) INTO DATA(lo_object).
      DATA lv_mdate TYPE datum.
      CONVERT TIME STAMP lo_object->get_lastmodified( )
        TIME ZONE 'UTC'
        INTO DATE lv_mdate.
      WRITE: / CONV text30( lo_object->get_key( ) ),
        lv_mdate, lo_object->get_size( ).
    ENDLOOP.
    CATCH /aws1/cx_rt_generic INTO DATA(lo_ex).
      DATA(lv_msg) = lo_ex->if_message~get_text( ).
      MESSAGE lv_msg TYPE 'I'.
    ENDTRY.
```

代码部分

各部分的代码情况审查如下。

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

用户无法指定物理存储桶名称，但可以指定逻辑存储桶与配合 AWS 管理员工作的系统管理员（尤其是业务分析师），从而向 /AWS1/IMG 中的物理存储桶映射逻辑存储桶。在大多数业务场景中，逻辑资源 ID 会在代码中进行硬编码或在自定义配置表中进行配置，因此用户没有机会选择逻辑存储桶。

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

使用此行建立安全会话并声明 ABAP 程序期望使用 DEMO_S3 SDK 配置文件。执行调用操作，以建立 SDK 配置连接并提取默认区域、身份验证设置和所需的 IAM 角色。自动调用 AUTHORIZATION-CHECK，以确保满足授权对象 /AWS1/SESS 的要求。此外，基于授权对象 /AWS1/LROL 执行 AUTHORIZATION-CHECK 调用操作，以确定授予用户的功能最强大（序列号较小）的 IAM 逻辑角色。SDK 会假设 IAM 角色已映射到适用于 SID 和客户端的 IAM 逻辑角色。随后，会话对象将根据 IMG 跟踪设置激活跟踪。

如果用户无权使用请求的 SDK 配置文件或任何可用的 IAM 逻辑角色，则会发生异常。

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

此行将逻辑资源解析为物理存储桶名称。如果因为配置缺乏适用于 SID/客户端组合的映射而无法解析逻辑资源，则会发生异常。

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

此行使用 /aws1/cl_s3_factory 方法的 create()，创建适用于 Amazon S3 的 API 对象。返回 /aws1/if_s3 类型的对象作为 Amazon S3 的 API 接口。必须为每项服务创建单独的 API 对象。例如，如果 ABAP 程序正在使用 Amazon S3 AWS Lambda、和 DynamoDB，则它会从、和创建 API 对象 /aws1/cl_s3_factory。 /aws1/cl_lmd_factory /aws1/cl_dyn_factory

如需覆盖经过默认区域配置的 IMG，则可以在支持区域指定的构造函数中选择部分可选参数。这样，您就能拥有两个 /aws1/if_s3 实例，分别用于 us-east-1 和 us-west-2，以便在不同区域的存储桶之间复制对象。同样，如果要通过报告读取财务相关存储桶中的数据并向物流相关存储桶写入对象，您可以创建两个不同的安全会话对象，用于创建两个单独的 /aws1/cl_s3 实例。

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

此行表示 ListObjectsV2 调用。输入简单参数后，此行会返回单一对象。这些对象可能表示 JSON 和 XML 深层数据，将其反序列化为面向 ABAP 对象的构造。某些情况下可能非常复杂。您现在只需处理输出内容，列出存储桶内容。

```
LOOP AT lo_output->get_contents( ) INTO DATA(lo_object).
    DATA lv_mdate TYPE datum.
    CONVERT TIME STAMP lo_object->get_lastmodified( )
        TIME ZONE 'UTC'
        INTO DATE lv_mdate.
    WRITE: / CONV text30( lo_object->get_key( ) ),
        lv_mdate, lo_object->get_size( ).
ENDLOOP.
```

使用隐藏数据内部表示方式的 GET...() 样式法，即可访问数据。GET_CONTENTS() 会返回 ABAP 表，其中每一行都含有表示 Amazon S3 单一条目的对象。在大多数情况下，AWS SDK 采用这种面向对象的方法，所有数据都以对象和表格的形式表示。该 LastModified 字段以时间戳表示，可以使用 abap-native CONVERT TIME STAMP 命令将其转换为日期。为了便于数学运算和格式化操作，会 GET_SIZE() 返回一个 INT4。

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

所有错误：连接错误、4xx 客户端错误、5xx 服务器错误或任何 ABAP 错误（如授权或配置错误）均表示为异常。这些异常可以单独解决。您可以决定是将异常作为信息性错误、重试、警告、Short Dump 还是作为其他任意类型的错误进行处理。

适用于 SAP ABAP 的 SDK 概念

本节涵盖的基本概念 适用于 SAP ABAP 的 AWS SDK。

主题

- [API 类](#)

- [其他对象](#)
- [结构类](#)
- [数组](#)
- [映射](#)
- [更高级别的函数](#)

API 类

每个都分配了一个 AWS 服务 由三个字母组成的首字母缩略词或 TLA。此服务由 /AWS1/IF_<TLA> 格式的接口表示。我们称之为服务接口。API 类在 /AWS1/API_<TLA> 软件包中。服务接口由每个 AWS 操作的一个方法组成 (我们将这些方法称为操作方法)。要查看的完整模块列表 适用于 SAP ABAP 的 AWS SDK TLAs，请参阅 [适用于 SAP ABAP 的 AWS SDK -模块列表](#)。

每种操作方法有多个 IMPORTING 参数，但最多只有一个 RETURNING 参数。这些参数通常会作为对象，具有复杂的构造函数和一套 GET...() 方法。在许多情况下，对象将包含嵌套对象、递归引用、对象表和表格等。这是因为传递 AWS 服务 的是深层的 XML 和 JSON 结构，这些结构不能用一组扁平的参数来表示。

RETURNING参数始终是一个类，即使该类只包含一个属性。

其他对象

每个 API 包中不仅有 API 主类，还有各种相关的存储库和数据字典对象。

- 每个结构类型对象的类。
- 表中出现的任意原始数据类型的类。例如，当服务返回字符串表时，ABAP API 会将其作为对象表，其中每个对象都是用来封装字符串的包装程序类。这样，包装程序类就能隐藏关于表示空字符串 (无法在 ABAP 中进行本地表示) 的详细信息。
- 适用于服务定义的特定错误的异常类
- 适用于每种原始数据类型的数据元素。每种数据类型都有数据元素，可自行记录。
- 适用于内部处理的其他对象，例如对 XML 和 JSON 负载进行序列化和反序列化操作的 XSLT 转换。

结构类

服务发送和接收的大部分 AWS 数据都由 AWS SDK 表示为类。这些类代表了数据结构，同时也隐藏了存储内部的详细信息。这些类尤其会隐藏表示此字段没有值的 SDK 方式。

一共有三种方法可应对结构类中的每个字段。

GET_field()

GET_field() 法

- 返回字段值，或者
- 如果字段中没有值，则返回默认值，可设为可选参数。

例如，可以考虑以下打印存储桶位置限制的代码。

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

如果存储桶完全没有位置限制（如 us-east-1），则 GET_LOCATIONCONSTRAINT() 将返回空字符串。如果字段中没有值，则可以覆盖此行为并指定所需值。

```
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' ).
```

如果 getbucketlocation() 未在结果中返回位置，则此程序将写入 Bucket Location: assuming us-east-1。

如果缺失请求值，请参阅以下代码示例，采用 GET() 法返回特定结果。

```
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' ).
```

此时，如果没有位置限制，GET_LOCATIONCONSTRAINT() 将返回 NopeNopeNope。

HAS_field()

HAS_field() 法用于确定字段中是否存在值。请参阅以下示例。

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

如果已知某个字段始终有值，则无需使用 HAS_field() 法。

ASK_field()

如果字段中没有值，则 ASK_field() 法会返回值或引发异常。这种方法能够轻松处理大量字段，并在字段不含值时摆脱逻辑，采用其他方法。

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

请注意，/AWS1/CX_RT_VALUE_MISSING 是静态异常，如果不捕获此异常，就会收到警告。

最佳实践

通常可以采用 GET_field() 法，将空字符串作为空白字符串，这也是三个选项中最接近 ABAP 的方法。但这种方法难以分辨字段到底是空白值还是没有值。如需通过区分缺失数据和空白数据来建立业务逻辑，则采用 HAS 法或 ASK 法。

数组

数组用 ABAP 标准对象表进行表示。

JSON 数组中可包含 NULL 值，如以下数组：['cat', 'dog', null, 'horse']。这是稀疏数组。稀疏数组在 ABAP 中表示为对象引用的内部表，而 null 值在表中表示为 ABAP null 真实值。迭代稀疏表时，必须检查 null 值，以免接触 null 对象或产生 CX_SY_REF_IS_INITIAL 异常。AWS 服务中其实很少出现稀疏数组。

使用 ABAP 7.40 新构造可以轻松初始化对象数组。以此次启动一个分配了多个安全组的 Amazon EC2 实例为例：

```
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_tagsspecifications = make_tag_spec( 'instance' )
)

```

映射

JSON 映射在 ABAP 中表示为 Hashed Tables，其中每个表行都只有两个组件。

- KEY：表中的 UNIQUE KEY 字符串。
- VALUE：含值对象。

映射是 AWS SDK 使用真实结构而不是类的极少数案例之一。这是必要的，因为 ABAP 哈希表不能将对象引用作为键字段，而且 AWS 映射键始终是非空字符串。

更高级别的函数

上一节中[API 类](#)描述的完全反映了该 AWS 服务，APIs 并将它们表示 APIs 为熟悉的 ABAP 类。在某些情况下，SDK 还包括更高级别的函数，这些函数在 API 类之上构建，以简化某些操作。为了方便程序员而包含更高级别的函数，并且不会取代较低级别的 API 类。

如果 SDK 包含模块的更高级别的函数，则它们包含在同一个传输中，并且可以通过名为的工厂类进行访问/AWS1/CL_TLA_L2_FACTORY。工厂类包括为模块创建各种更高级别的客户端的方法，这些客户端与 API [文档](#)中的 API 的其余部分一起记录在案。

适用于 SAP ABAP 的 AWS SDK features

适用于 SAP ABAP 的 AWS SDK 提供以下功能。

主题

- [编程配置](#)
- [Waiter](#)
- [分页器](#)
- [重试行为](#)

编程配置

使用适用于 SAP ABAP 的 AWS SDK 的 `/n/AWS1/IMG IMG` 交易，使用适用于 SAP ABAP 的 AWS SDK 的自定义业务配置应用程序——BTP 版本进行编程配置。

如需开始进行编程配置，请先使用 `get_config( )` 命令检索配置对象。

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

所有配置对象均使用 `/AWS1/IF_RT_CONFIG` 接口，其中包括与 IMG 对应的 GETter 和 SETter。例如，您可以覆盖默认区域。请参阅以下示例命令。

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

有些配置对象（如最大重试次数）没有 IMG 表示，只能以编程方式进行设置。请参阅以下示例命令。

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

的配置对象 AWS 服务 还可以包括中未表示的特定于服务的方法 `/aws1/if_rt_config`。

例如，Amazon S3 可以使用 `foobucket.s3.region.amazonaws.com` 虚拟端点或 `s3.region.amazonaws.com/foobucket` 路径样式，对 foobucket 存储桶进行寻址。您可以使用以下示例命令，强制使用路径样式。

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

有关服务配置的更多信息，请参阅 [适用于 SAP ABAP 的 AWS SDK — API 参考指南](#)。

Waiter

使用异步操作时 AWS APIs，您需要等待特定资源可用后再采取进一步的操作。例如，的 `CREATETABLE( )` API Amazon DynamoDB 会立即使用表格状态进行响应 `CREATING`。只有当表状态

更改为 ACTIVE 后，才能启动读取或写入操作。服务员使您能够在对 AWS 资源执行操作之前确认资源是否处于特定状态。

服务员使用服务操作来轮询 AWS 资源的状态，直到资源达到预期状态或确定资源未达到所需状态。通过编写代码来持续轮询 AWS 资源的操作可能既耗时又容易出错。Waiter 负责代表您执行轮询，有助于简化流程。

查看以下使用 Waiter 的 Amazon S3 示例。

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

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

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

- 本示例使用 Amazon S3 客户端创建存储桶。执行 `get_waiter()` 命令以指定 `bucketexists` 的时机。
- 请务必为每个 Waiter 指定 `iv_max_wait_time` 参数。此参数表示 Waiter 在完成指定操作前的等待总时长。在上述示例中，Waiter 可运行 200 秒。
- 必填参数中需要输入其他信息。在上述示例中，必须为 `iv_bucket` 参数输入 Amazon S3 存储桶名称。
- `/AWS1/CX_RT_WAITER_FAILURE` 异常表示 Waiter 的等待时长已超出 `iv_max_wait_time` 参数指定的最大时长。
- `/AWS1/CX_RT_WAITER_TIMEOUT` 异常表示 Waiter 因未达到所需状态而停止运行。

分页器

有些 AWS 服务 操作会提供分页响应。分页后，这些操作会在每次响应时返回固定数量的数据。您需要使用令牌或标记发送后续请求，才能检索整组结果。例如，Amazon S3 `ListObjectsV2` 操作每次最多只能返回 1000 个对象。您必须使用相应令牌发送后续请求，才能获得下一页的结果。

分页流程会连续发送请求，以从上次请求中断的地方继续操作。分页器是一种结果型迭代器，由适用于 SAP ABAP 的 SDK 提供。你可以轻松使用分页，而不必了解 APIs 使用分页令牌的 API 的底层机制。

分页器应用

您可以使用会返回分页器对象的 `get_paginator()` 方法创建分页器。分页器对象会调用正在进行的分页操作。分页器对象会接受需要提供给底层 API 的参数。分页流程会返回迭代器对象，可使用 `has_next()` 和 `get_next()` 方法迭代分页结果。

- `has_next()` 会返回布尔值，指示调用操作是否还有更多可用响应或页面。
- `get_next()` 会返回操作响应。

以下示例列出了使用分页器检索到的 S3 存储桶中的所有对象。

```
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.
```

重试行为

利用适用于 SAP ABAP 的 SDK，您可以为针对 AWS 服务 发出的由于节流或暂时错误而失败的请求配置最大重试次数。服务客户端级别允许的重试次数（即 SDK 在失败并引发异常之前重试操作的次数）由服务配置对象中的 `AV_MAX_ATTEMPTS` 属性指定。创建服务客户端对象时，SDK 会将 `AV_MAX_ATTEMPTS` 属性配置为默认值 3。服务配置对象可用于以编程方式将最大重试尝试次数设置为所需值。有关更多详细信息，请参阅以下示例。

```
" 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

尽管配置对象 ABAP SDK 允许使用 `/AWS1/IF_RT_CONFIG~SET_RETRY_MODE()` 方法设置重试模式，但 SDK 仅支持 standard 重试模式。有关更多信息，请参阅《工具参考指南》中的“[AWS SDKs 重试行为](#)”。

使用 SDK 构建产品

消费的产品或 ABAP 附加组件 AWS 服务 可以增强和扩展 SDK 的功能。您可以构建此类产品以与 SDK 结合使用。

主题

- [设置产品 ID](#)

设置产品 ID

建议您在产品或附加组件内建立会话时设置产品 ID。有关更多详细信息，请参阅以下示例。

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

产品 ID 只能包含字母、数字和下划线，不能包含空格或特殊字符。您可以将其与产品的技术名称或任何其它标识符进行匹配。如果您开发多个产品或附加组件，则每个产品的产品 ID 必须是唯一的。例如，发票分析器、税务计算器和定价引擎产品的产品可以是 INVOICE_ANALYZER、TAX_CALCULATOR 和 PRICING_ENGINE。IDs

向会话中添加产品 ID 可增强每次服务呼叫发送到 AWS 的遥测功能。遥测中包含产品 ID 和进行调用的对象的命名空间。借助这种遥测技术，支持可以识别正在拨打电话的产品，以防您的客户在使用 SDK 时遇到问题。它可以帮助澄清调用实际上是由产品而不是客户的代码发出的。

将 HTTP 请求自定义为 AWS

适用于 SAP ABAP 的 AWS SDK 处理创建 HTTP 请求、发送有效负载和接收响应的过程。

您可以自定义 HTTP 请求的行为或内容以满足您自己的 IT 要求。SDK 将增强点定义/AWS1/RT_EHN_HTTP_CLIENT为增强 HTTP 通信的中心位置。增强版支持向发出的请求中添加 HTTP 标头 AWS。

实施增强

SAP 为实施增强点提供了以下说明：

- [经典 ABAP](#)
- [BTP ABAP](#)

筛选增强功能

增强点支持多个可以同时激活的实现。如果您需要确保增强功能仅在调用特定 AWS 服务或 API 操作时运行，则可以 BAdi 根据以下属性筛选执行情况：

- TLA-服务的三个字母缩写，以大写字母表示。
- OPERATION-API 操作名称。例如，从 S3 存储桶获取对象的操作是[GetObject](#)。操作名称区分大小写，可能与 ABAP 方法名称不完全匹配。

对增强功能进行编码

此增强功能提供了以下方法。

修改请求标头

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

可以在CT_HEADERS内部表中追加和修改标题。我们不建议修改标头，因为这会改变 AWS 服务使用的数据。AWS 服务会忽略您添加的任何标头，但可以由您的 IT 基础架构（例如代理服务器或其他中间件）进行处理。

增强点是在计算身份验证和遥测标头之前调用的，因此增强功能无法对其进行修改。

以下是一个实现示例。

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

限制

适用于 SAP ABAP 的 AWS SDK 包括适用于所有人的 SDK 模块 AWS 服务。其中部分模块会受到如下限制。

- 依赖MQTT协议绑定的模块（例如iotevents）将无法运行。MQTT不是基于 HTTP 的协议，目前不支持。适用于 SAP ABAP 的 AWS SDK
- 目前尚不支持依赖 HTTP/2 流式功能的模块。目前尚不支持与事件流结合使用的特定服务操作，且无法运行 Amazon Kinesis Video Streams 等媒体流式服务操作。

适用于 SAP ABAP 的 AWS SDK 具有以下功能限制。

- 目前尚不支持以下 Amazon S3 功能。
 - 多区域接入点
 - Amazon S3 客户端加密

AWS 适用于 SAP ABAP 的 SDK-BTP 版本在开发者预览版期间存在以下限制。

- 某些模块可能不可用。
- 它无法卸载。
- 它的更新频率较低。

适用于 SAP ABAP 的 SDK 的代码示例

本主题中的代码示例向您展示了如何使用适用于 SAP ABAP 的 S AWS DK。AWS

基础知识是向您展示如何在服务中执行基本操作的代码示例。

操作是大型程序的代码摘录，必须在上下文中运行。您可以通过操作了解如何调用单个服务函数，还可以通过函数相关场景的上下文查看操作。

场景是向您展示如何通过在一个服务中调用多个函数或与其他 AWS 服务结合来完成特定任务的代码示例。

某些服务包含其他示例类别，这些类别说明如何利用特定于服务的库或函数。

服务

- [使用 SDK for SAP ABAP 的 Amazon Bedrock 运行时系统示例](#)
- [使用适用于 SAP 的软件开发工具包的 Amazon 基岩代理运行时示例 ABAP](#)
- [CloudWatch 使用适用于 SAP 的 SDK 的示例](#)
- [使用 SDK for SAP ABAP 的 DynamoDB 示例](#)
- [使用适用于 SAP 的软件开发工具包的亚马逊 EC2 示例 ABAP](#)
- [使用 SDK for SAP ABAP 的 Kinesis 示例](#)
- [使用 SDK for SAP ABAP 的 Lambda 示例](#)
- [使用 SDK for SAP ABAP 的 Amazon S3 示例](#)
- [SageMaker 使用适用于 SAP 的 SDK 的人工智能示例](#)
- [使用 SDK for SAP ABAP 的 Amazon SNS 示例](#)
- [使用 SDK for SAP ABAP 的 Amazon SQS 示例](#)
- [使用 SDK for SAP ABAP 的 Amazon Textract 示例](#)
- [使用 SDK for SAP ABAP 的 Amazon Translate 示例](#)

使用 SDK for SAP ABAP 的 Amazon Bedrock 运行时系统示例

以下代码示例向您展示了如何使用适用于 SAP ABAP 的 AWS 软件开发工具包和 Amazon Bedrock Runtime 来执行操作和实现常见场景。

每个示例都包含一个指向完整源代码的链接，您可以从中找到有关如何在上下文中设置和运行代码的说明。

主题

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

Anthropic Claude

InvokeModel

以下代码示例展示了如何使用 Invoke Model API 向 Anthropic Claude 发送短信。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

调用 Anthropic Claude 2 基础模型以生成文本。此示例使用了 of /US2/CL_JSON 功能，这些功能在某些 NetWeaver 版本上可能不可用。

```
"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.

```

使用 L2 高级别客户端调用 Anthropic Claude 2 基础模型来生成文本。

```

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.

```

使用 L2 高级别客户端调用 Anthropic Claude 3 基础模型来生成文本。

```

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.

```

- 有关 API 的详细信息，请参阅适用[InvokeModel](#)于 SAP 的 AWS SDK ABAP API 参考。

Stable Diffusion

InvokeModel

以下代码示例展示了如何在 Amazon Bedrock 上调用 Stability.ai Stable Diffusion XL 来生成图像。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

使用 Stable Diffusion 创建图像。

```
"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.
```

使用 L2 高级别客户端调用 Stability.ai Stable Diffusion XL 基础模型来生成图像。

```
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.
```

- 有关 API 的详细信息，请参阅适用[InvokeModel](#)于 SAP 的 AWS SDK ABAP API 参考。

使用适用于 SAP 的软件开发工具包的 Amazon 基岩代理运行时示例 ABAP

以下代码示例向您展示了如何使用适用于 SAP ABAP 的 AWS 软件开发工具包和 Amazon Bedrock Agents Runtime 来执行操作和实现常见场景。

操作是大型程序的代码摘录，必须在上下文中运行。您可以通过操作了解如何调用单个服务函数，还可以通过函数相关场景的上下文查看操作。

每个示例都包含一个指向完整源代码的链接，您可以从中找到有关如何在上下文中设置和运行代码的说明。

主题

- [操作](#)

操作

InvokeAgent

以下代码示例演示了如何使用 InvokeAgent。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

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

```

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

```

- 有关 API 的详细信息，请参阅适用[InvokeAgent](#)于 S AP 的AWS SDK ABAP API 参考。

CloudWatch 使用适用于 SAP 的 SDK 的示例

以下代码示例向您展示了如何使用适用于 SAP ABAP 的 AWS SDK 来执行操作和实现常见场景。

CloudWatch

操作是大型程序的代码摘录，必须在上下文中运行。您可以通过操作了解如何调用单个服务函数，还可以通过函数相关场景的上下文查看操作。

场景是向您演示如何通过在一个服务中调用多个函数或与其他 AWS 服务结合来完成特定任务的代码示例。

每个示例都包含一个指向完整源代码的链接，您可以从中找到有关如何在上下文中设置和运行代码的说明。

主题

- [操作](#)
- [场景](#)

操作

DeleteAlarms

以下代码示例演示了如何使用 DeleteAlarms。

适用于 SAP ABAP 的 SDK

 Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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.
```

- 有关 API 的详细信息，请参阅适用[DeleteAlarms](#)于 SAP 的 AWS SDK ABAP API 参考。

DescribeAlarms

以下代码示例演示了如何使用 DescribeAlarms。

适用于 SAP ABAP 的 SDK

 Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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.
```

- 有关 API 的详细信息，请参阅适用[DescribeAlarms](#)于 S AP 的AWS SDK ABAP API 参考。

DisableAlarmActions

以下代码示例演示了如何使用 DisableAlarmActions。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
"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.
```

- 有关 API 的详细信息，请参阅适用[DisableAlarmActions](#)于 S AP 的AWS SDK ABAP API 参考。

EnableAlarmActions

以下代码示例演示了如何使用 EnableAlarmActions。

适用于 SAP ABAP 的 SDK

 Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
"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.
```

- 有关 API 的详细信息，请参阅适用[EnableAlarmActions](#)于 SAP 的 AWS SDK ABAP API 参考。

ListMetrics

以下代码示例演示了如何使用 ListMetrics。

适用于 SAP ABAP 的 SDK

 Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
"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.

```

- 有关 API 的详细信息，请参阅适用[ListMetrics](#)于 SAP 的 AWS SDK ABAP API 参考。

PutMetricAlarm

以下代码示例演示了如何使用 PutMetricAlarm。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

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.

```

- 有关 API 的详细信息，请参阅适用[PutMetricAlarm](#)于 SAP 的 AWS SDK ABAP API 参考。

场景

开始使用警报

以下代码示例显示了操作流程：

- 创建警报。
- 禁用警报操作。
- 描述警报。
- 删除警报。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

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

"Create an alarm"
TRY.
    lo_cwt->putmetricalarm(
        iv_alarmname           = iv_alarm_name
        iv_comparisonoperator   = iv_comparison_operator
        iv_evaluationperiods    = iv_evaluation_periods
        iv_metricname           = iv_metric_name
        iv_namespace            = iv_namespace
        iv_statistic             = iv_statistic
        iv_threshold             = iv_threshold
        iv_actionsenabled        = iv_actions_enabled
        iv_alarmdescription     = iv_alarm_description
        iv_unit                  = iv_unit
        iv_period                = iv_period
```

```

        it_dimensions          = it_dimensions ).
    MESSAGE 'Alarm created' TYPE 'I'.
    CATCH /aws1/cx_cwtlimitexceededfault.
        MESSAGE 'The request processing has exceeded the limit' TYPE 'E'.
    ENDTRY.

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

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

    "Describe alarm using the same ABAP internal table."
    TRY.
        oo_result = lo_cwt->describealarms(
            it_alarmnames          = lt_alarmnames ).
        " oo_result is
returned for testing purpose "
        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.

```

- 有关 API 详细信息，请参阅《适用于 SAP ABAP 的 AWS SDK 的 API 参考》中的以下主题。

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

使用 SDK for SAP ABAP 的 DynamoDB 示例

以下代码示例向您展示了如何使用适用于 SAP ABAP 的软件开发工具包和 Dynamo AWS DB 来执行操作和实现常见场景。

基础知识是向您展示如何在服务中执行基本操作的代码示例。

操作是大型程序的代码摘录，必须在上下文中运行。您可以通过操作了解如何调用单个服务函数，还可以通过函数相关场景的上下文查看操作。

每个示例都包含一个指向完整源代码的链接，您可以从中找到有关如何在上下文中设置和运行代码的说明。

主题

- [基本功能](#)
- [操作](#)

基本功能

了解基础知识

以下代码示例展示了如何：

- 创建可保存电影数据的表。
- 在表中加入单一电影，获取并更新此电影。
- 向 JSON 示例文件的表中写入电影数据。
- 查询在给定年份发行的电影。
- 扫描在年份范围内发行的电影。
- 删除表中的电影后再删除表。

适用于 SAP ABAP 的 SDK

 Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
" 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' }| ) ) )
            ) ).
    TRY.

```

```

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

" Scan items from table.
TRY.
  DATA(lo_scan_result) = lo_dyn->scan( iv_tablename = iv_table_name ).
  lt_items = lo_scan_result->get_items( ).
  " Read the first item and display the attributes.
  READ TABLE lo_query_result->get_items( ) INTO lt_item INDEX 1.
  lo_title = lt_item[ key = 'title' ]-value.
  lo_year = lt_item[ key = 'year' ]-value.
  lo_rating = lt_item[ key = 'rating' ]-value.
  MESSAGE 'Movie name is: ' && lo_title->get_s( ) TYPE 'I'.
  MESSAGE 'Movie year is: ' && lo_year->get_n( ) TYPE 'I'.
  MESSAGE 'Movie rating is: ' && lo_rating->get_n( ) TYPE 'I'.
CATCH /aws1/cx_dynresourcenotfoundex.

```

```

        MESSAGE 'The table or index does not exist' TYPE 'E'.
    ENDTRY.

" Update items from table.
TRY.
    DATA(lt_attributeupdates) = VALUE /aws1/
cl_dynattrvalueupdate=>tt_attributeupdates(
    ( VALUE /aws1/cl_dynattrvalueupdate=>ts_attributeupdates_maprow(
        key = 'rating' value = NEW /aws1/cl_dynattrvalueupdate(
            io_value = NEW /aws1/cl_dynattributevalue( iv_n = '7.6' )
            iv_action = |PUT| ) ) ) ).
    DATA(lt_key) = VALUE /aws1/cl_dynattributevalue=>tt_key(
        ( VALUE /aws1/cl_dynattributevalue=>ts_key_maprow(
            key = 'year' value = NEW /aws1/cl_dynattributevalue( iv_n = '1975' ) ) )
        ( VALUE /aws1/cl_dynattributevalue=>ts_key_maprow(
            key = 'title' value = NEW /aws1/cl_dynattributevalue( iv_s =
'1980' ) ) ) ).
    DATA(lo_resp) = lo_dyn->updateitem(
        iv_tablename      = iv_table_name
        it_key             = lt_key
        it_attributeupdates = lt_attributeupdates ).
    MESSAGE '1 item updated in DynamoDB Table' && iv_table_name TYPE 'I'.
    CATCH /aws1/cx_dyncondalcheckfaile00.
        MESSAGE 'A condition specified in the operation could not be evaluated.'
TYPE 'E'.
    CATCH /aws1/cx_dynresourcenotfoundex.
        MESSAGE 'The table or index does not exist' TYPE 'E'.
    CATCH /aws1/cx_dyntransactconflictex.
        MESSAGE 'Another transaction is using the item' TYPE 'E'.
    ENDTRY.

" Delete table.
TRY.
    lo_dyn->deletetable( iv_tablename = iv_table_name ).
    lo_dyn->get_waiter( )->tablenotexists(
        iv_max_wait_time = 200
        iv_tablename      = iv_table_name ).
    MESSAGE 'DynamoDB Table deleted.' TYPE 'I'.
    CATCH /aws1/cx_dynresourcenotfoundex.
        MESSAGE 'The table or index does not exist' TYPE 'E'.
    CATCH /aws1/cx_dynresourceinuseex.
        MESSAGE 'The table cannot be deleted as it is in use' TYPE 'E'.
    ENDTRY.

```

- 有关 API 详细信息，请参阅《适用于 SAP ABAP 的 AWS SDK 的 API 参考》中的以下主题。

- [BatchWriteItem](#)
- [CreateTable](#)
- [DeleteItem](#)
- [DeleteTable](#)
- [DescribeTable](#)
- [GetItem](#)
- [PutItem](#)
- [Query](#)
- [Scan](#)
- [UpdateItem](#)

操作

CreateTable

以下代码示例演示了如何使用 CreateTable。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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.

```

- 有关 API 的详细信息，请参阅适用[CreateTable](#)于 S AP 的AWS SDK ABAP API 参考。

DeleteItem

以下代码示例演示了如何使用 DeleteItem。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

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.

```

- 有关 API 的详细信息，请参阅适用[DeleteItem](#)于 SAP 的 AWS SDK ABAP API 参考。

DeleteTable

以下代码示例演示了如何使用 DeleteTable。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

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.

```

- 有关 API 的详细信息，请参阅适用[DeleteTable](#)于 S AP 的AWS SDK ABAP API 参考。

DescribeTable

以下代码示例演示了如何使用 DescribeTable。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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.
```

- 有关 API 的详细信息，请参阅适用[DescribeTable](#)于 S AP 的AWS SDK ABAP API 参考。

GetItem

以下代码示例演示了如何使用 GetItem。

适用于 SAP ABAP 的 SDK

 Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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.
```

- 有关 API 的详细信息，请参阅适用[GetItem](#)于 S AP 的AWS SDK ABAP API 参考。

ListTables

以下代码示例演示了如何使用 ListTables。

适用于 SAP ABAP 的 SDK

 Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
TRY.
```

```

oo_result = lo_dyn->listtables( ).
" You can loop over the oo_result to get table properties like this.
LOOP AT oo_result->get_tablenames( ) INTO DATA(lo_table_name).
  DATA(lv_tablename) = lo_table_name->get_value( ).
ENDLOOP.
DATA(lv_tablecount) = lines( oo_result->get_tablenames( ) ).
MESSAGE 'Found ' && lv_tablecount && ' tables' TYPE 'I'.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
  DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
  MESSAGE lv_error TYPE 'E'.
ENDTRY.

```

- 有关 API 的详细信息，请参阅适用[ListTables](#)于 S AP 的AWS SDK ABAP API 参考。

PutItem

以下代码示例演示了如何使用 PutItem。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

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.

```

- 有关 API 的详细信息，请参阅适用[PutItem](#)于 SAP 的 AWS SDK ABAP API 参考。

Query

以下代码示例演示了如何使用 Query。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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.
  ENDLLOOP.
  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.
```

- 有关 API 详细信息，请参阅《AWS SDK for SAP ABAP API 参考》中的 [Query](#)。

Scan

以下代码示例演示了如何使用 Scan。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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.
    ENDLLOOP.
    DATA(lv_count) = oo_scan_result->get_count( ).
    MESSAGE 'Found ' && lv_count && ' items' TYPE 'I'.
    CATCH /aws1/cx_dynresourcenotfoundex.
        MESSAGE 'The table or index does not exist' TYPE 'E'.
```

```
ENDTRY.
```

- 有关 API 详细信息，请参阅《AWS SDK for SAP ABAP API 参考》中的 [Scan](#)。

UpdateItem

以下代码示例演示了如何使用 UpdateItem。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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.
```

- 有关 API 的详细信息，请参阅适用[UpdateItem](#)于 S AP 的AWS SDK ABAP API 参考。

使用适用于 SAP 的软件开发工具包的亚马逊 EC2 示例 ABAP

以下代码示例向您展示了如何使用适用于 SAP ABAP 的 AWS 软件开发工具包和 Amazon EC2 来执行操作和实现常见场景。

操作是大型程序的代码摘录，必须在上下文中运行。您可以通过操作了解如何调用单个服务函数，还可以通过函数相关场景的上下文查看操作。

每个示例都包含一个指向完整源代码的链接，您可以从中找到有关如何在上下文中设置和运行代码的说明。

主题

- [操作](#)

操作

AllocateAddress

以下代码示例演示了如何使用 AllocateAddress。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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.
```

- 有关 API 的详细信息，请参阅适用 [AllocateAddress](#) 于 SAP 的 AWS SDK ABAP API 参考。

AssociateAddress

以下代码示例演示了如何使用 AssociateAddress。

适用于 SAP ABAP 的 SDK

 Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

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

- 有关 API 的详细信息，请参阅适用 [AssociateAddress](#) 于 SAP 的 AWS SDK ABAP API 参考。

CreateKeyPair

以下代码示例演示了如何使用 CreateKeyPair。

适用于 SAP ABAP 的 SDK

 Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
TRY.
    oo_result = lo_ec2->createkeypair( iv_keyname = iv_key_name ).
    MESSAGE 'Amazon EC2 key pair created.' TYPE 'I'.
```

```

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

```

- 有关 API 的详细信息，请参阅适用[CreateKeyPair](#)于 S AP 的AWS SDK ABAP API 参考。

CreateSecurityGroup

以下代码示例演示了如何使用 CreateSecurityGroup。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

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

```

- 有关 API 的详细信息，请参阅适用[CreateSecurityGroup](#)于 S AP 的AWS SDK ABAP API 参考。

DeleteKeyPair

以下代码示例演示了如何使用 DeleteKeyPair。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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.
```

- 有关 API 的详细信息，请参阅适用[DeleteKeyPair](#)于 SAP 的 AWS SDK ABAP API 参考。

DeleteSecurityGroup

以下代码示例演示了如何使用 DeleteSecurityGroup。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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.
```

- 有关 API 的详细信息，请参阅适用[DeleteSecurityGroup](#)于 S AP 的AWS SDK ABAP API 参考。

DescribeAddresses

以下代码示例演示了如何使用 DescribeAddresses。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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.
```

- 有关 API 的详细信息，请参阅适用[DescribeAddresses](#)于 S AP 的AWS SDK ABAP API 参考。

DescribeAvailabilityZones

以下代码示例演示了如何使用 DescribeAvailabilityZones。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

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.

```

- 有关 API 的详细信息，请参阅适用[DescribeAvailabilityZones](#)于 SAP 的 AWS SDK ABAP API 参考。

DescribeInstances

以下代码示例演示了如何使用 DescribeInstances。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

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/ec2instancetypename,
           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( ).
    ENDLOOP.
ENDLOOP.
MESSAGE 'Retrieved information about EC2 instances.' TYPE 'I'.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
    MESSAGE lv_error TYPE 'E'.
ENDTRY.

```

- 有关 API 的详细信息，请参阅适用[DescribeInstances](#)于 S AP 的AWS SDK ABAP API 参考。

DescribeKeyPairs

以下代码示例演示了如何使用 DescribeKeyPairs。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

    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.

```

- 有关 API 的详细信息，请参阅适用[DescribeKeyPairs](#)于 S AP 的AWS SDK ABAP API 参考。

DescribeRegions

以下代码示例演示了如何使用 DescribeRegions。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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.
```

- 有关 API 的详细信息，请参阅适用[DescribeRegions](#)于 SAP 的 AWS SDK ABAP API 参考。

DescribeSecurityGroups

以下代码示例演示了如何使用 DescribeSecurityGroups。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
TRY.
```

```

DATA lt_group_ids TYPE /aws1/cl_ec2groupidstringlist_w=>tt_groupidstringlist.
APPEND NEW /aws1/cl_ec2groupidstringlist_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.

```

- 有关 API 的详细信息，请参阅适用[DescribeSecurityGroups](#)于 S AP 的AWS SDK ABAP API 参考。

MonitorInstances

以下代码示例演示了如何使用 MonitorInstances。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

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.

```

- 有关 API 的详细信息，请参阅适用[MonitorInstances](#)于 S AP 的AWS SDK ABAP API 参考。

RebootInstances

以下代码示例演示了如何使用 RebootInstances。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

    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.

```

- 有关 API 的详细信息，请参阅适用[RebootInstances](#)于 SAP 的 AWS SDK ABAP API 参考。

ReleaseAddress

以下代码示例演示了如何使用 ReleaseAddress。

适用于 SAP ABAP 的 SDK

 Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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.
```

- 有关 API 的详细信息，请参阅适用[ReleaseAddress](#)于 SAP 的 AWS SDK ABAP API 参考。

RunInstances

以下代码示例演示了如何使用 RunInstances。

适用于 SAP ABAP 的 SDK

 Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
" Create tags for resource created during instance launch. "
DATA lt_tagsspecifications TYPE /aws1/
cl_ec2tagsspecification=>tt_tagsspecificationlist.
DATA ls_tagsspecifications LIKE LINE OF lt_tagsspecifications.
ls_tagsspecifications = NEW /aws1/cl_ec2tagsspecification(
    iv_resourcetype = 'instance'
```

```

        it_tags = VALUE /aws1/cl_ec2tag=>tt_taglist(
            ( NEW /aws1/cl_ec2tag( iv_key = 'Name' iv_value = iv_tag_value ) )
        ) ).
    APPEND ls_tagspecifications TO lt_tagspecifications.

TRY.
    " Create/launch Amazon Elastic Compute Cloud (Amazon EC2) instance. "
    oo_result = lo_ec2->runinstances(                                " oo_result is
returned for testing purposes. "
        iv_imageid = iv_ami_id
        iv_instancetype = 't3.micro'
        iv_maxcount = 1
        iv_mincount = 1
        it_tagspecifications = lt_tagspecifications
        iv_subnetid = iv_subnet_id ).
    MESSAGE 'EC2 instance created.' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
    MESSAGE lv_error TYPE 'E'.
ENDTRY.

```

- 有关 API 的详细信息，请参阅适用[RunInstances](#)于 S AP 的AWS SDK ABAP API 参考。

StartInstances

以下代码示例演示了如何使用 StartInstances。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

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.

```

- 有关 API 的详细信息，请参阅适用[StartInstances](#)于 SAP 的 AWS SDK ABAP API 参考。

StopInstances

以下代码示例演示了如何使用 StopInstances。

适用于 SAP ABAP 的 SDK

 Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

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.
```

- 有关 API 的详细信息，请参阅适用[StopInstances](#)于 S AP 的AWS SDK ABAP API 参考。

使用 SDK for SAP ABAP 的 Kinesis 示例

以下代码示例向您展示了如何使用适用于 SAP ABAP 的软件开发工具包和 AWS Kinesis 来执行操作和实现常见场景。

基础知识是向您展示如何在服务中执行基本操作的代码示例。

操作是大型程序的代码摘录，必须在上下文中运行。您可以通过操作了解如何调用单个服务函数，还可以通过函数相关场景的上下文查看操作。

每个示例都包含一个指向完整源代码的链接，您可以从中找到有关如何在上下文中设置和运行代码的说明。

主题

- [基本功能](#)
- [操作](#)

基本功能

了解基础知识

以下代码示例展示了如何：

- 创建流并在其中放置一条记录。
- 创建分片迭代器。
- 读取记录，然后清理函数。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

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_knslimitexceededException.
    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_knskmsoptionrequired.
        MESSAGE 'KMS key option is required.' TYPE 'E'.
    CATCH /aws1/cx_knskms throttlingex.
        MESSAGE 'The rate of requests to AWS KMS is exceeding the request quotas.'
TYPE 'E'.
    CATCH /aws1/cx_knsprovthruputexcdex.
        MESSAGE 'The request rate for the stream is too high, or the requested data
is too large for the available throughput.' TYPE 'E'.
    CATCH /aws1/cx_knsresourcenotfoundex.
        MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
ENDTRY.

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

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

```

```

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

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

```

- 有关 API 详细信息，请参阅《适用于 SAP ABAP 的 AWS SDK 的 API 参考》中的以下主题。
 - [CreateStream](#)
 - [DeleteStream](#)

- [GetRecords](#)
- [GetShardIterator](#)
- [PutRecord](#)

操作

CreateStream

以下代码示例演示了如何使用 CreateStream。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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_knslimitexceededException.
    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.
```

- 有关 API 的详细信息，请参阅适用[CreateStream](#)于 S AP 的AWS SDK ABAP API 参考。

DeleteStream

以下代码示例演示了如何使用 DeleteStream。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
TRY.
    lo_kns->deletestream(
        iv_streamname = iv_stream_name ).
    MESSAGE 'Stream deleted.' TYPE 'I'.
CATCH /aws1/cx_knslimitexceedex.
    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.
```

- 有关 API 的详细信息，请参阅适用[DeleteStream](#)于 SAP 的 AWS SDK ABAP API 参考。

DescribeStream

以下代码示例演示了如何使用 DescribeStream。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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_knslimitexceedex.
    MESSAGE 'The request processing has failed because of a limit exceed
exception.' TYPE 'E'.
    CATCH /aws1/cx_knsresourcenotfoundex.
    MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
    ENDTRY.

```

- 有关 API 的详细信息，请参阅适用[DescribeStream](#)于 S AP 的AWS SDK ABAP API 参考。

GetRecords

以下代码示例演示了如何使用 GetRecords。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

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_knskms throttlingex.
  MESSAGE 'The rate of requests to AWS KMS is exceeding the request quotas.'
TYPE 'E'.
CATCH /aws1/cx_knsprovthruputexcdex.
  MESSAGE 'The request rate for the stream is too high, or the requested data
is too large for the available throughput.' TYPE 'E'.
CATCH /aws1/cx_knsresourcenotfoundex.
  MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
ENDTRY.

```

- 有关 API 的详细信息，请参阅适用[GetRecords](#)于 S AP 的AWS SDK ABAP API 参考。

ListStreams

以下代码示例演示了如何使用 ListStreams。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

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_knslimitexceedex.
  MESSAGE 'The request processing has failed because of a limit exceed
exception.' TYPE 'E'.
ENDTRY.

```

- 有关 API 的详细信息，请参阅适用[ListStreams](#)于 S AP 的AWS SDK ABAP API 参考。

PutRecord

以下代码示例演示了如何使用 PutRecord。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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_knskms throttlingex.
    MESSAGE 'The rate of requests to AWS KMS is exceeding the request quotas.'
TYPE 'E'.
CATCH /aws1/cx_knsprovthruputexcdex.
    MESSAGE 'The request rate for the stream is too high, or the requested data
is too large for the available throughput.' TYPE 'E'.
CATCH /aws1/cx_knsresourcenotfoundex.
    MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
ENDTRY.
```

- 有关 API 的详细信息，请参阅适用[PutRecord](#)于 S AP 的AWS SDK ABAP API 参考。

RegisterStreamConsumer

以下代码示例演示了如何使用 RegisterStreamConsumer。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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_sgmresourceimitexcd.
    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.
```

- 有关 API 的详细信息，请参阅适用[RegisterStreamConsumer](#)于 S AP 的AWS SDK ABAP API 参考。

使用 SDK for SAP ABAP 的 Lambda 示例

以下代码示例向您展示了如何使用适用于 SAP ABAP 的 AWS 软件开发工具包和 Lambda 来执行操作和实现常见场景。

基础知识是向您展示如何在服务中执行基本操作的代码示例。

操作是大型程序的代码摘录，必须在上下文中运行。您可以通过操作了解如何调用单个服务函数，还可以通过函数相关场景的上下文查看操作。

每个示例都包含一个指向完整源代码的链接，您可以从中找到有关如何在上下文中设置和运行代码的说明。

主题

- [基本功能](#)
- [操作](#)

基本功能

了解基础知识

以下代码示例展示了如何：

- 创建 IAM 角色和 Lambda 函数，然后上传处理程序代码。
- 使用单个参数来调用函数并获取结果。
- 更新函数代码并使用环境变量进行配置。
- 使用新参数来调用函数并获取结果。显示返回的执行日志。
- 列出账户函数，然后清除函数。

有关更多信息，请参阅[使用控制台创建 Lambda 函数](#)。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
TRY.  
  "Create an AWS Identity and Access Management (IAM) role that grants AWS  
  Lambda permission to write to logs."  
  DATA(lv_policy_document) = `{` &&  
    `"Version": "2012-10-17",` &&
```

```

        `Statement": [ ` &&
        `{ ` &&
        ` "Effect": "Allow", ` &&
        ` "Action": [ ` &&
        `   ` "sts:AssumeRole" ` &&
        ` ], ` &&
        ` "Principal": { ` &&
        `   ` "Service": [ ` &&
        `     ` "lambda.amazonaws.com" ` &&
        `   ] ` &&
        ` } ` &&
        ` } ` &&
        ` ] ` &&
        ` } ` &&
    TRY.
        DATA(lo_create_role_output) = lo_iam->createrole(
            iv_rolename = iv_role_name
            iv_assumerolepolicydocument = lv_policy_document
            iv_description = 'Grant lambda permission to write to logs' ).
        DATA(lv_role_arn) = lo_create_role_output->get_role( )->get_arn( ).
        MESSAGE 'IAM role created.' TYPE 'I'.
        WAIT UP TO 10 SECONDS.           " Make sure that the IAM role is ready
for use. "
        CATCH /aws1/cx_iamentityalrddyexex.
            DATA(lo_role) = lo_iam->getrole( iv_rolename = iv_role_name ).
            lv_role_arn = lo_role->get_role( )->get_arn( ).
        CATCH /aws1/cx_iaminvalidinputex.
            MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
        CATCH /aws1/cx_iammalformedplydocex.
            MESSAGE 'Policy document in the request is malformed.' TYPE 'E'.
        ENDTRY.

    TRY.
        lo_iam->attachrolepolicy(
            iv_rolename = iv_role_name
            iv_policyarn = 'arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole' ).
        MESSAGE 'Attached policy to the IAM role.' TYPE 'I'.
        CATCH /aws1/cx_iaminvalidinputex.
            MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
        CATCH /aws1/cx_iamnosuchentityex.
            MESSAGE 'The requested resource entity does not exist.' TYPE 'E'.
        CATCH /aws1/cx_iamplynotattachableex.

```



```

    MESSAGE 'Service role policies can only be attached to the service-
linked role for their service.' TYPE 'E'.
    CATCH /aws1/cx_iamunmodableentityex.
    MESSAGE 'Service that depends on the service-linked role is not
modifiable.' TYPE 'E'.
    ENDTRY.

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

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

"Invoke the function with a single parameter and get results."
TRY.
    DATA(lv_json) = /aws1/cl_rt_util=>string_to_xstring(
        `{` &&
        `"action": "increment",` &&
        `"number": 10` &&
        `}` ).
    DATA(lo_initial_invoke_output) = lo_lmd->invoke(
        iv_functionname = iv_function_name

```

```

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

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

        TRY.
            DATA lt_variables TYPE /aws1/
cl_lmdenvironmentvaria00=>tt_environmentvariables.
            DATA ls_variable LIKE LINE OF lt_variables.
            ls_variable-key = 'LOG_LEVEL'.

```

```

ls_variable-value = NEW /aws1/cl_lmdenvironmentvaria00( iv_value =
'info' ).
INSERT ls_variable INTO TABLE lt_variables.

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

"Invoke the function with new parameters and get results. Display the
execution log that's returned from the invocation."
TRY.
    lv_json = /aws1/cl_rt_util=>string_to_xstring(
        `{` &&
        `"action": "decrement",` &&
        `"number": 10` &&
        `}` ).
    DATA(lo_updated_invoke_output) = lo_lmd->invoke(
        iv_functionname = iv_function_name
        iv_payload = lv_json ).
    ov_updated_invoke_payload = lo_updated_invoke_output->get_payload( ).
    " ov_updated_invoke_payload is returned for testing purposes. "
    lo_writer_json = cl_sxml_string_writer=>create( type =
if_sxml=>co_xt_json ).
    CALL TRANSFORMATION id SOURCE XML ov_updated_invoke_payload RESULT XML
lo_writer_json.
    lv_result = cl_abap_codepage=>convert_from( lo_writer_json-
>get_output( ) ).
    MESSAGE 'Lambda function invoked.' TYPE 'I'.
    CATCH /aws1/cx_lmdinvparamvalueex.
    MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
    CATCH /aws1/cx_lmdinvrequestcontext.
    MESSAGE 'Unable to parse request body as JSON.' TYPE 'E'.

```

```

        CATCH /aws1/cx_lmdresourcenotfoundex.
            MESSAGE 'The requested resource does not exist.' TYPE 'E'.
        CATCH /aws1/cx_lmdunsuppedmediatyp00.
            MESSAGE 'Invoke request body does not have JSON as its content type.'
TYPE 'E'.
        ENDTRY.

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

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

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

```

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

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

- 有关 API 详细信息，请参阅《适用于 SAP ABAP 的 AWS SDK 的 API 参考》中的以下主题。

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

操作

CreateFunction

以下代码示例演示了如何使用 CreateFunction。

适用于 SAP ABAP 的 SDK

 Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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_lmdcodesigningcfn00.
    MESSAGE 'Code signing configuration does not exist.' TYPE 'E'.
CATCH /aws1/cx_lmdcodestorageexcdex.
    MESSAGE 'Maximum total code size per account exceeded.' TYPE 'E'.
CATCH /aws1/cx_lmdcodeverification00.
    MESSAGE 'Code signature failed one or more validation checks for signature
mismatch or expiration.' TYPE 'E'.
CATCH /aws1/cx_lmdinvalidcodesigex.
    MESSAGE 'Code signature failed the integrity check.' TYPE 'E'.
CATCH /aws1/cx_lmdinvparamvalueex.
    MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
CATCH /aws1/cx_lmdresourceconflictex.
    MESSAGE 'Resource already exists or another operation is in progress.' TYPE
'E'.
CATCH /aws1/cx_lmdresourcenotfoundex.
    MESSAGE 'The requested resource does not exist.' TYPE 'E'.
CATCH /aws1/cx_lmdserviceexception.
    MESSAGE 'An internal problem was encountered by the AWS Lambda service.'
TYPE 'E'.
CATCH /aws1/cx_lmdtoomanyrequestsex.
    MESSAGE 'The maximum request throughput was reached.' TYPE 'E'.
ENDTRY.
```

- 有关 API 的详细信息，请参阅适用[CreateFunction](#)于 SAP 的 AWS SDK ABAP API 参考。

DeleteFunction

以下代码示例演示了如何使用 DeleteFunction。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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.
```

- 有关 API 的详细信息，请参阅适用[DeleteFunction](#)于 S AP 的AWS SDK ABAP API 参考。

GetFunction

以下代码示例演示了如何使用 GetFunction。

适用于 SAP ABAP 的 SDK

 Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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.
```

- 有关 API 的详细信息，请参阅适用[GetFunction](#)于 SAP 的 AWS SDK ABAP API 参考。

Invoke

以下代码示例演示了如何使用 Invoke。

适用于 SAP ABAP 的 SDK

 Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
TRY.
    DATA(lv_json) = /aws1/cl_rt_util=>string_to_xstring(
        `{` &&
```



```

        `"action": "increment",` &&
        `"number": 10` &&
    `}` ).
    oo_result = lo_lmd->invoke(
testing purposes. "
        iv_functionname = iv_function_name
        iv_payload = lv_json ).
    MESSAGE 'Lambda function invoked.' TYPE 'I'.
    CATCH /aws1/cx_lmdinvparamvalueex.
        MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
    CATCH /aws1/cx_lmdinvrequestctx.
        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.

```

- 有关 API 详细信息，请参阅适用于 SAP ABAP 的 AWS SDK 的 API 参考中的[调用](#)。

ListFunctions

以下代码示例演示了如何使用 ListFunctions。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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.
```

- 有关 API 的详细信息，请参阅适用[ListFunctions](#)于 S AP 的 AWS SDK ABAP API 参考。

UpdateFunctionCode

以下代码示例演示了如何使用 UpdateFunctionCode。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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_lmdcodesigningcfn00.
        MESSAGE 'Code signing configuration does not exist.' TYPE 'E'.
    CATCH /aws1/cx_lmdcodestorageexcdex.
        MESSAGE 'Maximum total code size per account exceeded.' TYPE 'E'.
    CATCH /aws1/cx_lmdcodeverification00.
        MESSAGE 'Code signature failed one or more validation checks for signature
mismatch or expiration.' TYPE 'E'.
    CATCH /aws1/cx_lmdinvalidcodesigex.
        MESSAGE 'Code signature failed the integrity check.' TYPE 'E'.
    CATCH /aws1/cx_lmdinvparamvalueex.
        MESSAGE 'The request contains a non-valid parameter.' TYPE 'E'.
    CATCH /aws1/cx_lmdresourceconflictex.
        MESSAGE 'Resource already exists or another operation is in progress.' TYPE
'E'.
    CATCH /aws1/cx_lmdresourcenotfoundex.
        MESSAGE 'The requested resource does not exist.' TYPE 'E'.
    CATCH /aws1/cx_lmdserviceexception.
        MESSAGE 'An internal problem was encountered by the AWS Lambda service.'
TYPE 'E'.
    CATCH /aws1/cx_lmdtoomanyrequestsex.
        MESSAGE 'The maximum request throughput was reached.' TYPE 'E'.
    ENDTRY.

```

- 有关 API 的详细信息，请参阅适用[UpdateFunctionCode](#)于 SAP 的 AWS SDK ABAP API 参考。

UpdateFunctionConfiguration

以下代码示例演示了如何使用 UpdateFunctionConfiguration。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

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_lmdcodesigningcfn00.
        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.

```

- 有关 API 的详细信息，请参阅适用[UpdateFunctionConfiguration](#)于 S AP 的 AWS SDK ABAP API 参考。

使用 SDK for SAP ABAP 的 Amazon S3 示例

以下代码示例向您展示了如何使用适用于 SAP ABAP 的软件开发工具包和 Amazon AWS S3 来执行操作和实现常见场景。

基础知识是向您展示如何在服务中执行基本操作的代码示例。

操作是大型程序的代码摘录，必须在上下文中运行。您可以通过操作了解如何调用单个服务函数，还可以通过函数相关场景的上下文查看操作。

每个示例都包含一个指向完整源代码的链接，您可以从中找到有关如何在上下文中设置和运行代码的说明。

主题

- [基本功能](#)
- [操作](#)

基本功能

了解基础知识

以下代码示例展示了如何：

- 创建桶并将文件上载到其中。
- 从桶中下载对象。
- 将对象复制到存储桶中的子文件夹。
- 列出存储桶中的对象。
- 删除存储桶及其对象。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

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

" Create an Amazon Simple Storage Service (Amazon S3) bucket. "
TRY.
    " determine our region from our session
    DATA(lv_region) = CONV /aws1/s3_bucketlocationcnstrnt( lo_session-
>get_region( ) ).
```

```

DATA lo_constraint TYPE REF TO /aws1/cl_s3_createbucketconf.
" When in the us-east-1 region, you must not specify a constraint
" In all other regions, specify the region as the constraint
IF lv_region = 'us-east-1'.
    CLEAR lo_constraint.
ELSE.
    lo_constraint = NEW /aws1/cl_s3_createbucketconf( lv_region ).
ENDIF.

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

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

    lo_s3->putobject(
        iv_bucket = iv_bucket_name
        iv_key = iv_key
        iv_body = lv_file_content ).
    MESSAGE 'Object uploaded to S3 bucket.' TYPE 'I'.
    CATCH /aws1/cx_s3_nosuchbucket.
        MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.

" Get an object from a bucket. "
TRY.
    DATA(lo_result) = lo_s3->getobject(
        iv_bucket = iv_bucket_name
        iv_key = iv_key ).
    DATA(lv_object_data) = lo_result->get_body( ).
    MESSAGE 'Object retrieved from S3 bucket.' TYPE 'I'.

```

```

CATCH /aws1/cx_s3_nosuchbucket.
  MESSAGE 'Bucket does not exist.' TYPE 'E'.
CATCH /aws1/cx_s3_nosuchkey.
  MESSAGE 'Object key does not exist.' TYPE 'E'.
ENDTRY.

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

" List objects in the bucket. "
TRY.
  DATA(lo_list) = lo_s3->listobjects(
    iv_bucket = iv_bucket_name ).
  MESSAGE 'Retrieved list of objects in S3 bucket.' TYPE 'I'.
CATCH /aws1/cx_s3_nosuchbucket.
  MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.
DATA text TYPE string VALUE 'Object List - '.
DATA lv_object_key TYPE /aws1/s3_objectkey.
LOOP AT lo_list->get_contents( ) INTO DATA(lo_object).
  lv_object_key = lo_object->get_key( ).
  CONCATENATE lv_object_key ', ' INTO text.
ENDLOOP.
MESSAGE text TYPE 'I'.

" Delete the objects in a bucket. "
TRY.
  lo_s3->deleteobject(
    iv_bucket = iv_bucket_name
    iv_key = iv_key ).
  lo_s3->deleteobject(
    iv_bucket = iv_bucket_name
    iv_key = |{ iv_copy_to_folder }/{ iv_key }| ).
  MESSAGE 'Objects deleted from S3 bucket.' TYPE 'I'.

```

```
CATCH /aws1/cx_s3_nosuchbucket.  
  MESSAGE 'Bucket does not exist.' TYPE 'E'.  
ENDTRY.  
  
" Delete the bucket. "  
TRY.  
  lo_s3->deletebucket(  
    iv_bucket = iv_bucket_name ).  
  MESSAGE 'Deleted S3 bucket.' TYPE 'I'.  
CATCH /aws1/cx_s3_nosuchbucket.  
  MESSAGE 'Bucket does not exist.' TYPE 'E'.  
ENDTRY.
```

• 有关 API 详细信息，请参阅《适用于 SAP ABAP 的 AWS SDK 的 API 参考》中的以下主题。

- [CopyObject](#)
- [CreateBucket](#)
- [DeleteBucket](#)
- [DeleteObjects](#)
- [GetObject](#)
- [ListObjectsV2](#)
- [PutObject](#)

操作

CopyObject

以下代码示例演示了如何使用 CopyObject。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。


```

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.

```

- 有关 API 的详细信息，请参阅适用[CopyObject](#)于 SAP 的 AWS SDK ABAP API 参考。

CreateBucket

以下代码示例演示了如何使用 CreateBucket。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

TRY.
  " determine our region from our session
  DATA(lv_region) = CONV /aws1/s3_bucketlocationcnstrnt( lo_session-
>get_region( ) ).
  DATA lo_constraint TYPE REF TO /aws1/cl_s3_createbucketconf.
  " When in the us-east-1 region, you must not specify a constraint
  " In all other regions, specify the region as the constraint
  IF lv_region = 'us-east-1'.
    CLEAR lo_constraint.
  ELSE.
    lo_constraint = NEW /aws1/cl_s3_createbucketconf( lv_region ).
  ENDIF.

  lo_s3->createbucket(

```

```

        iv_bucket = iv_bucket_name
        io_createbucketconfiguration = io_constraint ).
    MESSAGE 'S3 bucket created.' TYPE 'I'.
CATCH /aws1/cx_s3_bucketalrdyexists.
    MESSAGE 'Bucket name already exists.' TYPE 'E'.
CATCH /aws1/cx_s3_bktalrdyownedbyyou.
    MESSAGE 'Bucket already exists and is owned by you.' TYPE 'E'.
ENDTRY.

```

- 有关 API 的详细信息，请参阅适用[CreateBucket](#)于 S AP 的AWS SDK ABAP API 参考。

DeleteBucket

以下代码示例演示了如何使用 DeleteBucket。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

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.

```

- 有关 API 的详细信息，请参阅适用[DeleteBucket](#)于 S AP 的AWS SDK ABAP API 参考。

DeleteObject

以下代码示例演示了如何使用 DeleteObject。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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.
```

- 有关 API 的详细信息，请参阅适用 [DeleteObject](#) 于 SAP 的 AWS SDK ABAP API 参考。

GetObject

以下代码示例演示了如何使用 GetObject。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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.

```

- 有关 API 的详细信息，请参阅适用[GetObject](#)于 SAP 的 AWS SDK ABAP API 参考。

ListObjectsV2

以下代码示例演示了如何使用 ListObjectsV2。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

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.

```

- 有关 API 的详细信息，请参阅适用于 SAP 的 AWS SDK ABAP API 参考中的 [ListObjectsV2](#)。

PutObject

以下代码示例演示了如何使用 PutObject。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
"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.
```

- 有关 API 的详细信息，请参阅适用 [PutObject](#) 于 SAP 的 AWS SDK ABAP API 参考。

SageMaker 使用适用于 SAP 的 SDK 的人工智能示例

以下代码示例向您展示了如何使用带有 A SageMaker I 的 SAP ABAP AWS SDK 来执行操作和实现常见场景。

操作是大型程序的代码摘录，必须在上下文中运行。您可以通过操作了解如何调用单个服务函数，还可以通过函数相关场景的上下文查看操作。

场景是向您演示如何通过在一个服务中调用多个函数或与其他 AWS 服务结合来完成特定任务的代码示例。

每个示例都包含一个指向完整源代码的链接，您可以从中找到有关如何在上下文中设置和运行代码的说明。

主题

- [操作](#)
- [场景](#)

操作

CreateEndpoint

以下代码示例演示了如何使用 CreateEndpoint。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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_sgmcreeendptcfgout.

"Create a production variant as an ABAP object."
"Identifies a model that you want to host and the resources chosen to deploy for
hosting it."
lo_production_variants = NEW #( iv_variantname = iv_variant_name
                                iv_modelname = iv_model_name
                                iv_initialinstancecount =
iv_initial_instance_count
                                iv_instancetype = iv_instance_type ).

INSERT lo_production_variants INTO TABLE lt_production_variants.

"Create an endpoint configuration."
TRY.
```

```

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

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

```

- 有关 API 的详细信息，请参阅适用[CreateEndpoint](#)于 S AP 的AWS SDK ABAP API 参考。

CreateModel

以下代码示例演示了如何使用 CreateModel。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

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

```
"Create an ABAP object for the container image based on input variables."
```

```
lo_primarycontainer = NEW #( iv_image = iv_container_image
                             iv_modeldataurl = iv_model_data_url ).
```

```

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

```

- 有关 API 的详细信息，请参阅适用[CreateModel](#)于 SAP 的 AWS SDK ABAP API 参考。

CreateTrainingJob

以下代码示例演示了如何使用 CreateTrainingJob。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

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."
lo_hyperparameters_w = NEW #( iv_value = iv_hp_max_depth ).
INSERT VALUE #( key = 'max_depth' value = lo_hyperparameters_w ) INTO TABLE
lt_hyperparameters.

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

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

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

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

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

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

"Create ABAP objects for training data sources."
lo_trn_s3datasource = NEW #( iv_s3datatype = iv_trn_data_s3datatype
                             iv_s3datadistributiontype =
iv_trn_data_s3datadistribution
                             iv_s3uri = iv_trn_data_s3uri ).

lo_trn_datasource = NEW #( io_s3datasource = lo_trn_s3datasource ).

lo_trn_channel = NEW #( iv_channelname = 'train'
                        io_datasource = lo_trn_datasource
                        iv_compressiontype = iv_trn_data_compressiontype
                        iv_contenttype = iv_trn_data_contenttype ).

```

```

INSERT lo_trn_channel INTO TABLE lt_input_data_config.

"Create ABAP objects for validation data sources."
lo_val_s3datasource = NEW #( iv_s3datatype = iv_val_data_s3datatype
                             iv_s3datadistributiontype =
iv_val_data_s3datadistribution
                             iv_s3uri = iv_val_data_s3uri ).

lo_val_datasource = NEW #( io_s3datasource = lo_val_s3datasource ).

lo_val_channel = NEW #( iv_channelname = 'validation'
                        io_datasource = lo_val_datasource
                        iv_compressiontype = iv_val_data_compressiontype
                        iv_contenttype = iv_val_data_contenttype ).

INSERT lo_val_channel INTO TABLE lt_input_data_config.

"Create an ABAP object for algorithm specification."
lo_algorithm_specification = NEW #( iv_trainingimage = iv_training_image
                                    iv_traininginputmode =
iv_training_input_mode ).

"Create an ABAP object for resource configuration."
lo_resource_config = NEW #( iv_instancecount = iv_instance_count
                            iv_instancetype = iv_instance_type
                            iv_volumesizeingb = iv_volume_sizeingb ).

"Create an ABAP object for output data configuration."
lo_output_data_config = NEW #( iv_s3outputpath = iv_s3_output_path ).

"Create an ABAP object for stopping condition."
lo_stopping_condition = NEW #( iv_maxruntimeinseconds =
iv_max_runtime_in_seconds ).

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

```

```

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

```

- 有关 API 的详细信息，请参阅适用[CreateTrainingJob](#)于 SAP 的 AWS SDK ABAP API 参考。

CreateTransformJob

以下代码示例演示了如何使用 CreateTransformJob。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

DATA lo_transforminput TYPE REF TO /aws1/cl_sgmtransforminput.
DATA lo_transformoutput TYPE REF TO /aws1/cl_sgmtransformoutput.
DATA lo_transformresources TYPE REF TO /aws1/cl_sgmtransformresources.
DATA lo_datasource TYPE REF TO /aws1/cl_sgmtransformdatasrc.
DATA lo_s3datasource TYPE REF TO /aws1/cl_sgmtransforms3datasrc.

"Create an ABAP object for an Amazon Simple Storage Service (Amazon S3) data
source."
lo_s3datasource = NEW #( iv_s3uri = iv_tf_data_s3uri
                        iv_s3datatype = iv_tf_data_s3datatype ).

"Create an ABAP object for data source."
lo_datasource = NEW #( io_s3datasource = lo_s3datasource ).

"Create an ABAP object for transform data source."

```

```

lo_transforminput = NEW #( io_datasource = lo_datasource
                           iv_contenttype = iv_tf_data_contenttype
                           iv_compressiontype = iv_tf_data_compressiontype ).

"Create an ABAP object for resource configuration."
lo_transformresources = NEW #( iv_instancecount = iv_instance_count
                               iv_instancetype = iv_instance_type ).

"Create an ABAP object for output data configuration."
lo_transformoutput = NEW #( iv_s3outputpath = iv_s3_output_path ).

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

```

- 有关 API 的详细信息，请参阅适用[CreateTransformJob](#)于 S AP 的AWS SDK ABAP API 参考。

DeleteEndpoint

以下代码示例演示了如何使用 DeleteEndpoint。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

"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.

```

- 有关 API 的详细信息，请参阅适用[DeleteEndpoint](#)于 S AP 的AWS SDK ABAP API 参考。

DeleteModel

以下代码示例演示了如何使用 DeleteModel。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

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.

```

- 有关 API 的详细信息，请参阅适用[DeleteModel](#)于 S AP 的AWS SDK ABAP API 参考。

DescribeTrainingJob

以下代码示例演示了如何使用 DescribeTrainingJob。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

    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.

```

- 有关 API 的详细信息，请参阅适用[DescribeTrainingJob](#)于 S AP 的AWS SDK ABAP API 参考。

ListAlgorithms

以下代码示例演示了如何使用 ListAlgorithms。

适用于 SAP ABAP 的 SDK

 Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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.
```

- 有关 API 的详细信息，请参阅适用[ListAlgorithms](#)于 SAP 的 AWS SDK ABAP API 参考。

ListModels

以下代码示例演示了如何使用 ListModels。

适用于 SAP ABAP 的 SDK

 Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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.

```

- 有关 API 的详细信息，请参阅适用[ListModels](#)于 S AP 的AWS SDK ABAP API 参考。

ListNotebookInstances

以下代码示例演示了如何使用 ListNotebookInstances。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

    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.

```

- 有关 API 的详细信息，请参阅适用[ListNotebookInstances](#)于 S AP 的AWS SDK ABAP API 参考。

ListTrainingJobs

以下代码示例演示了如何使用 ListTrainingJobs。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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.
```

- 有关 API 的详细信息，请参阅适用[ListTrainingJobs](#)于 SAP 的 AWS SDK ABAP API 参考。

场景

开始使用模型和端点

以下代码示例展示了如何：

- 开始训练任务并创建 A SageMaker I 模型。
- 创建端点配置。
- 创建端点，然后清理资源。

适用于 SAP ABAP 的 SDK

 Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

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_sgmdesctrnjobrsp.
DATA lt_production_variants TYPE /aws1/
cl_sgmproductionvariant=>tt_productionvariantlist.
DATA lt_input_data_config TYPE /aws1/cl_sgmchannel=>tt_inputdataconfig.
DATA lt_hyperparameters TYPE /aws1/cl_sgmhyperparameters_w=>tt_hyperparameters.
DATA lv_model_data_url TYPE /aws1/sgmurl.

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

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

lo_hyperparameters_w = NEW #( iv_value = iv_hp_eta ).

```

```

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

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

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

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

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

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

    "Create ABAP internal table for data based on input variables."
    "Training data."
    lo_trn_s3datasource = NEW #( iv_s3datatype = iv_trn_data_s3datatype
                                iv_s3datadistributiontype =
iv_trn_data_s3datadistribution
                                iv_s3uri = iv_trn_data_s3uri ).

    lo_trn_datasource = NEW #( io_s3datasource = lo_trn_s3datasource ).

    lo_trn_channel = NEW #( iv_channelname = 'train'
                            io_datasource = lo_trn_datasource
                            iv_compressiontype = iv_trn_data_compressiontype
                            iv_contenttype = iv_trn_data_contenttype ).
    INSERT lo_trn_channel INTO TABLE lt_input_data_config.

    "Validation data."
    lo_val_s3datasource = NEW #( iv_s3datatype = iv_val_data_s3datatype
                                iv_s3datadistributiontype =
iv_val_data_s3datadistribution
                                iv_s3uri = iv_val_data_s3uri ).

```

```

lo_val_datasource = NEW #( io_s3datasource = lo_val_s3datasource ).

lo_val_channel = NEW #( iv_channelname = 'validation'
                        io_datasource = lo_val_datasource
                        iv_compressiontype = iv_val_data_compressiontype
                        iv_contenttype = iv_val_data_contenttype ).
INSERT lo_val_channel INTO TABLE lt_input_data_config.

"Create an ABAP object for algorithm specification based on input variables."
lo_algorithm_specification = NEW #( iv_trainingimage = iv_training_image
                                    iv_traininginputmode =
iv_training_input_mode ).

"Create an ABAP object for resource configuration."
lo_resource_config = NEW #( iv_instancecount = iv_instance_count
                            iv_instancetype = iv_instance_type
                            iv_volumesizeingb = iv_volume_sizeingb ).

"Create an ABAP object for output data configuration."
lo_output_data_config = NEW #( iv_s3outputpath = iv_s3_output_path ).

"Create an ABAP object for stopping condition."
lo_stopping_condition = NEW #( iv_maxruntimeinseconds =
iv_max_runtime_in_seconds ).

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

```

```

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

    "Create ABAP object for the container image based on input variables."
    lo_primarycontainer = NEW #( iv_image = iv_training_image
                                iv_modeldataurl = lv_model_data_url ).

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

    "Create an endpoint production variant."
    lo_production_variants = NEW #( iv_variantname = iv_ep_variant_name
                                    iv_modelname = iv_model_name
                                    iv_initialinstancecount =
iv_ep_initial_instance_count
                                    iv_instancetype = iv_ep_instance_type ).
    INSERT lo_production_variants INTO TABLE lt_production_variants.

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

        "Create an endpoint."

```

```

        oo_ep_output = lo_sgm->createendpoint(          " oo_ep_output is returned for
testing purposes. "
        iv_endpointconfigname = iv_ep_cfg_name
        iv_endpointname = iv_ep_name ).
    MESSAGE 'Endpoint created.' TYPE 'I'.
    CATCH /aws1/cx_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.

```

- 有关 API 详细信息，请参阅《适用于 SAP ABAP 的 AWS SDK 的 API 参考》中的以下主题。

- [CreateEndpoint](#)
- [CreateEndpointConfig](#)
- [CreateModel](#)
- [CreateTrainingJob](#)
- [DeleteEndpoint](#)
- [DeleteEndpointConfig](#)
- [DeleteModel](#)
- [DescribeEndpoint](#)
- [DescribeTrainingJob](#)

使用 SDK for SAP ABAP 的 Amazon SNS 示例

以下代码示例向您展示了如何使用适用于 SAP ABAP 的软件开发工具包和 Amazon AWS SNS 来执行操作和实现常见场景。

操作是大型程序的代码摘录，必须在上下文中运行。您可以通过操作了解如何调用单个服务函数，还可以通过函数相关场景的上下文查看操作。

场景是向您演示如何通过在一个服务中调用多个函数或与其他 AWS 服务结合来完成特定任务的代码示例。

每个示例都包含一个指向完整源代码的链接，您可以从中找到有关如何在上下文中设置和运行代码的说明。

主题

- [操作](#)
- [场景](#)

操作

CreateTopic

以下代码示例演示了如何使用 CreateTopic。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
TRY.  
    oo_result = lo_sns->createtopic( iv_name = iv_topic_name ). " oo_result is  
returned for testing purposes. "  
    MESSAGE 'SNS topic created' TYPE 'I'.  
    CATCH /aws1/cx_snstopiclimitexcdex.  
        MESSAGE 'Unable to create more topics. You have reached the maximum number  
of topics allowed.' TYPE 'E'.  
ENDTRY.
```

- 有关 API 的详细信息，请参阅适用[CreateTopic](#)于 SAP 的 AWS SDK ABAP API 参考。

DeleteTopic

以下代码示例演示了如何使用 DeleteTopic。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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.
```


- 有关 API 的详细信息，请参阅适用[DeleteTopic](#)于 S AP 的AWS SDK ABAP API 参考。

GetTopicAttributes

以下代码示例演示了如何使用 GetTopicAttributes。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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.
```

- 有关 API 的详细信息，请参阅适用[GetTopicAttributes](#)于 S AP 的AWS SDK ABAP API 参考。

ListSubscriptions

以下代码示例演示了如何使用 ListSubscriptions。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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.

```

- 有关 API 的详细信息，请参阅适用[ListSubscriptions](#)于 S AP 的AWS SDK ABAP API 参考。

ListTopics

以下代码示例演示了如何使用 ListTopics。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

    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.

```

- 有关 API 的详细信息，请参阅适用[ListTopics](#)于 S AP 的AWS SDK ABAP API 参考。

Publish

以下代码示例演示了如何使用 Publish。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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.
```

- 有关 API 详细信息，请参阅适用于 SAP ABAP 的 AWS SDK 的 API 参考中的 [发布](#)。

SetTopicAttributes

以下代码示例演示了如何使用 SetTopicAttributes。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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.
```

- 有关 API 的详细信息，请参阅适用[SetTopicAttributes](#)于 S AP 的AWS SDK ABAP API 参考。

Subscribe

以下代码示例演示了如何使用 Subscribe。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

将电子邮件地址订阅到主题。

```
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.
```

- 有关 API 详细信息，请参阅适用于 SAP ABAP 的AWS SDK 的 API 参考中的[订阅](#)。

Unsubscribe

以下代码示例演示了如何使用 Unsubscribe。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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.
```

- 有关 API 详细信息，请参阅适用于 SAP ABAP 的 AWS SDK 的 API 参考中的 [Unsubscribe](#)。

场景

创建并发布到 FIFO 主题

以下代码示例显示了如何创建并发布到 Amazon SNS FIFO 主题。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

创建 FIFO 主题并为此订阅 Amazon SQS FIFO 队列，然后向 Amazon SNS 主题发布消息。

```
" Creates a FIFO topic. "
```

```

DATA lt_tpc_attributes TYPE /aws1/cl_snstopicattrmap_w=>tt_topicattributesmap.
DATA ls_tpc_attributes TYPE /aws1/
cl_snstopicattrmap_w=>ts_topicattributesmap_maprow.
ls_tpc_attributes-key = 'FifoTopic'.
ls_tpc_attributes-value = NEW /aws1/cl_snstopicattrmap_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_snssubscriptionlmt00.
      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.

```

- 有关 API 详细信息，请参阅适用于 SAP ABAP 的 AWS SDK 的 API 参考中的以下主题。
 - [CreateTopic](#)
 - [发布](#)
 - [订阅](#)

使用 SDK for SAP ABAP 的 Amazon SQS 示例

以下代码示例向您展示了如何使用适用于 SAP ABAP 的软件开发工具包和 Amazon AWS SQS 来执行操作和实现常见场景。

操作是大型程序的代码摘录，必须在上下文中运行。您可以通过操作了解如何调用单个服务函数，还可以通过函数相关场景的上下文查看操作。

场景是向您演示如何通过在一个服务中调用多个函数或与其他 AWS 服务结合来完成特定任务的代码示例。

每个示例都包含一个指向完整源代码的链接，您可以从中找到有关如何在上下文中设置和运行代码的说明。

主题

- [操作](#)
- [场景](#)

操作

CreateQueue

以下代码示例演示了如何使用 CreateQueue。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

创建 Amazon SQS 标准队列。

```
TRY.
    oo_result = lo_sqs->createqueue( iv_queue_name = iv_queue_name ).
    oo_result is returned for testing purposes. "
    MESSAGE 'SQS queue created.' TYPE 'I'.
CATCH /aws1/cx_sqsqueuecreatedrecently.
    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.
```

创建等待接收消息的 Amazon SQS 队列。

```
TRY.
    DATA lt_attributes TYPE /aws1/cl_sqsqueueattribmap_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_sqsqueuecreatedrecently.
  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.

```

- 有关 API 的详细信息，请参阅适用[CreateQueue](#)于 S AP 的AWS SDK ABAP API 参考。

DeleteQueue

以下代码示例演示了如何使用 DeleteQueue。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

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

```

- 有关 API 的详细信息，请参阅适用[DeleteQueue](#)于 S AP 的AWS SDK ABAP API 参考。

GetQueueUrl

以下代码示例演示了如何使用 GetQueueUrl。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
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.
```

- 有关 API 的详细信息，请参阅适用[GetQueueUrl](#)于 S AP 的AWS SDK ABAP API 参考。

ListQueues

以下代码示例演示了如何使用 ListQueues。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
TRY.  
    oo_result = lo_sqs->listqueues( ).      " oo_result is returned for  
testing purposes. "
```

```
MESSAGE 'Retrieved list of queues.' TYPE 'I'.
ENDTRY.
```

- 有关 API 的详细信息，请参阅适用[ListQueues](#)于 SAP 的 AWS SDK ABAP API 参考。

ReceiveMessage

以下代码示例演示了如何使用 ReceiveMessage。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

接收来自 Amazon SQS 队列的消息。

```
TRY.
    oo_result = lo_sqs->receivemessage( iv_queueurl = iv_queue_url ).      "
    oo_result is returned for testing purposes. "
    DATA(lt_messages) = oo_result->get_messages( ).
    MESSAGE 'Message received from SQS queue.' TYPE 'I'.
    CATCH /aws1/cx_sqsoverlimit.
        MESSAGE 'Maximum number of in-flight messages reached.' TYPE 'E'.
ENDTRY.
```

使用长轮询支持接收来自 Amazon SQS 队列的消息。

```
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.

```

- 有关 API 的详细信息，请参阅适用[ReceiveMessage](#)于 S AP 的AWS SDK ABAP API 参考。

SendMessage

以下代码示例演示了如何使用 SendMessage。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

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_sqsinvalidmsgcnts.
  MESSAGE 'Message contains non-valid characters.' TYPE 'E'.
CATCH /aws1/cx_sqsunsupportedop.
  MESSAGE 'Operation not supported.' TYPE 'E'.
ENDTRY.

```

- 有关 API 的详细信息，请参阅适用[SendMessage](#)于 S AP 的AWS SDK ABAP API 参考。

场景

创建并发布到 FIFO 主题

以下代码示例显示了如何创建并发布到 Amazon SNS FIFO 主题。

适用于 SAP ABAP 的 SDK

 Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

创建 FIFO 主题并为此订阅 Amazon SQS FIFO 队列，然后向 Amazon SNS 主题发布消息。

```
" 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_snstopicclimitexcdex.
    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'.
    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.

```

- 有关 API 详细信息，请参阅适用于 SAP ABAP 的 AWS SDK 的 API 参考中的以下主题。
 - [CreateTopic](#)
 - [发布](#)

- [订阅](#)

使用 SDK for SAP ABAP 的 Amazon Textract 示例

以下代码示例向您展示了如何使用适用于 SAP ABAP 的 AWS 软件开发工具包和 Amazon Textract 来执行操作和实现常见场景。

操作是大型程序的代码摘录，必须在上下文中运行。您可以通过操作了解如何调用单个服务函数，还可以通过函数相关场景的上下文查看操作。

场景是向您演示如何通过在一个服务中调用多个函数或与其他 AWS 服务结合来完成特定任务的代码示例。

每个示例都包含一个指向完整源代码的链接，您可以从中找到有关如何在上下文中设置和运行代码的说明。

主题

- [操作](#)
- [场景](#)

操作

AnalyzeDocument

以下代码示例演示了如何使用 AnalyzeDocument。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
"Detects text and additional elements, such as forms or tables,"  
"in a local image file or from in-memory byte data."  
"The image must be in PNG or JPG format."
```

```

"Create ABAP objects for feature type."
"Add TABLES to return information about the tables."
"Add FORMS to return detected form data."
"To perform both types of analysis, add TABLES and FORMS to FeatureTypes."

DATA(lt_featuretypes) = VALUE /aws1/cl_texfeaturetypes_w=>tt_featuretypes(
  ( NEW /aws1/cl_texfeaturetypes_w( iv_value = 'FORMS' ) )
  ( NEW /aws1/cl_texfeaturetypes_w( iv_value = 'TABLES' ) ) ).

"Create an ABAP object for the Amazon Simple Storage Service (Amazon S3)
object."
DATA(lo_s3object) = NEW /aws1/cl_texs3object( iv_bucket = iv_s3bucket
  iv_name    = iv_s3object ).

"Create an ABAP object for the document."
DATA(lo_document) = NEW /aws1/cl_texdocument( io_s3object = lo_s3object ).

"Analyze document stored in Amazon S3."
TRY.
  oo_result = lo_tex->analyzedocument(      "oo_result is returned for testing
purposes."
    io_document      = lo_document
    it_featuretypes  = lt_featuretypes ).
  LOOP AT oo_result->get_blocks( ) INTO DATA(lo_block).
    IF lo_block->get_text( ) = 'INGREDIENTS: POWDERED SUGAR* (CANE SUGAR,'.
      MESSAGE 'Found text in the doc: ' && lo_block->get_text( ) TYPE 'I'.
    ENDIF.
  ENDLOOP.
  MESSAGE 'Analyze document completed.' TYPE 'I'.
CATCH /aws1/cx_texaccesssdeniedex.
  MESSAGE 'You do not have permission to perform this action.' TYPE 'E'.
CATCH /aws1/cx_texbaddocumentex.
  MESSAGE 'Amazon Textract is not able to read the document.' TYPE 'E'.
CATCH /aws1/cx_texdocumenttoolargeex.
  MESSAGE 'The document is too large.' TYPE 'E'.
CATCH /aws1/cx_texhlquotaexceededex.
  MESSAGE 'Human loop quota exceeded.' TYPE 'E'.
CATCH /aws1/cx_texinternalservererr.
  MESSAGE 'Internal server error.' TYPE 'E'.
CATCH /aws1/cx_texinvalidparameterex.
  MESSAGE 'Request has non-valid parameters.' TYPE 'E'.

CATCH /aws1/cx_texinvalids3objectex.

```



```

    MESSAGE 'Amazon S3 object is not valid.' TYPE 'E'.
    CATCH /aws1/cx_texpvthruputexcdex.
    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.

```

- 有关 API 的详细信息，请参阅适用[AnalyzeDocument](#)于 S AP 的AWS SDK ABAP API 参考。

DetectDocumentText

以下代码示例演示了如何使用 DetectDocumentText。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

"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.
    ENDLLOOP.
    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.

```

- 有关 API 的详细信息，请参阅适用[DetectDocumentText](#)于 SAP 的 AWS SDK ABAP API 参考。

GetDocumentAnalysis

以下代码示例演示了如何使用 GetDocumentAnalysis。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

"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.

```

- 有关 API 的详细信息，请参阅适用[GetDocumentAnalysis](#)于 SAP 的 AWS SDK ABAP API 参考。

StartDocumentAnalysis

以下代码示例演示了如何使用 StartDocumentAnalysis。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
"Starts the asynchronous analysis of an input document for relationships"
"between detected items such as key-value pairs, tables, and selection
elements."

"Create ABAP objects for feature type."
"Add TABLES to return information about the tables."
"Add FORMS to return detected form data."
"To perform both types of analysis, add TABLES and FORMS to FeatureTypes."

DATA(lt_featuretypes) = VALUE /aws1/cl_texfeaturetypes_w=>tt_featuretypes(
  ( NEW /aws1/cl_texfeaturetypes_w( iv_value = 'FORMS' ) )
  ( NEW /aws1/cl_texfeaturetypes_w( iv_value = 'TABLES' ) ) ).
"Create an ABAP object for the Amazon S3 object."
DATA(lo_s3object) = NEW /aws1/cl_texs3object( iv_bucket = iv_s3bucket
  iv_name   = iv_s3object ).
"Create an ABAP object for the document."
DATA(lo_documentlocation) = NEW /aws1/cl_texdocumentlocation( io_s3object =
lo_s3object ).

"Start async document analysis."
TRY.
  oo_result = lo_tex->startdocumentanalysis(      "oo_result is returned for
testing purposes."
    io_documentlocation      = lo_documentlocation
    it_featuretypes          = lt_featuretypes ).
  DATA(lv_jobid) = oo_result->get_jobid( ).

  MESSAGE 'Document analysis started.' TYPE 'I'.
CATCH /aws1/cx_texaccessdeniedex.
  MESSAGE 'You do not have permission to perform this action.' TYPE 'E'.
```

```

CATCH /aws1/cx_texbaddocumentex.
  MESSAGE 'Amazon Textract is not able to read the document.' TYPE 'E'.
CATCH /aws1/cx_texdocumenttoolargeex.
  MESSAGE 'The document is too large.' TYPE 'E'.
CATCH /aws1/cx_texidempotentprmmis00.
  MESSAGE 'Idempotent parameter mismatch exception.' TYPE 'E'.
CATCH /aws1/cx_texinternalservererr.
  MESSAGE 'Internal server error.' TYPE 'E'.
CATCH /aws1/cx_texinvalidkmskeyex.
  MESSAGE 'AWS KMS key is not valid.' TYPE 'E'.
CATCH /aws1/cx_texinvalidparameterex.
  MESSAGE 'Request has non-valid parameters.' TYPE 'E'.
CATCH /aws1/cx_texinvalids3objectex.
  MESSAGE 'Amazon S3 object is not valid.' TYPE 'E'.
CATCH /aws1/cx_texlimitexceededex.
  MESSAGE 'An Amazon Textract service limit was exceeded.' TYPE 'E'.
CATCH /aws1/cx_texprovthruputexcdex.
  MESSAGE 'Provisioned throughput exceeded limit.' TYPE 'E'.
CATCH /aws1/cx_texthrottlingex.
  MESSAGE 'The request processing exceeded the limit.' TYPE 'E'.
CATCH /aws1/cx_texunsupporteddocex.
  MESSAGE 'The document is not supported.' TYPE 'E'.
ENDTRY.

```

- 有关 API 的详细信息，请参阅适用[StartDocumentAnalysis](#)于 SAP 的 AWS SDK ABAP API 参考。

StartDocumentTextDetection

以下代码示例演示了如何使用 StartDocumentTextDetection。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

"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_texlimitexceedex.
  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.

```

- 有关 API 的详细信息，请参阅适用[StartDocumentTextDetection](#)于 S AP 的AWS SDK ABAP API 参考。

场景

开始使用文档分析

以下代码示例显示了操作流程：

- 开始异步分析。
- 获取文档分析。

SDK for SAP ABAP

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
"Create ABAP objects for feature type."
"Add TABLES to return information about the tables."
"Add FORMS to return detected form data."
"To perform both types of analysis, add TABLES and FORMS to FeatureTypes."

DATA(lt_featuretypes) = VALUE /aws1/cl_texfeaturetypes_w=>tt_featuretypes(
  ( NEW /aws1/cl_texfeaturetypes_w( iv_value = 'FORMS' ) )
  ( NEW /aws1/cl_texfeaturetypes_w( iv_value = 'TABLES' ) ) ).

"Create an ABAP object for the Amazon Simple Storage Service (Amazon S3)
object."
DATA(lo_s3object) = NEW /aws1/cl_texs3object( iv_bucket = iv_s3bucket
  iv_name    = iv_s3object ).

"Create an ABAP object for the document."
DATA(lo_documentlocation) = NEW /aws1/cl_texdocumentlocation( io_s3object =
lo_s3object ).

"Start document analysis."
```

```

TRY.
  DATA(lo_start_result) = lo_tex->startdocumentanalysis(
    io_documentlocation    = lo_documentlocation
    it_featuretypes        = lt_featuretypes ).
  MESSAGE 'Document analysis started.' TYPE 'I'.
CATCH /aws1/cx_texaccessdeniedex.
  MESSAGE 'You do not have permission to perform this action.' TYPE 'E'.
CATCH /aws1/cx_texbaddocumentex.
  MESSAGE 'Amazon Textract is not able to read the document.' TYPE 'E'.
CATCH /aws1/cx_texdocumenttoolargeex.
  MESSAGE 'The document is too large.' TYPE 'E'.
CATCH /aws1/cx_texidempotentprmmis00.
  MESSAGE 'Idempotent parameter mismatch exception.' TYPE 'E'.
CATCH /aws1/cx_texinternalservererr.
  MESSAGE 'Internal server error.' TYPE 'E'.
CATCH /aws1/cx_texinvalidkmskeyex.
  MESSAGE 'AWS KMS key is not valid.' TYPE 'E'.
CATCH /aws1/cx_texinvalidparameterex.
  MESSAGE 'Request has non-valid parameters.' TYPE 'E'.
CATCH /aws1/cx_texinvalids3objectex.
  MESSAGE 'Amazon S3 object is not valid.' TYPE 'E'.
CATCH /aws1/cx_texlimitexceededex.
  MESSAGE 'An Amazon Textract service limit was exceeded.' TYPE 'E'.
CATCH /aws1/cx_texprovthruputexcdex.
  MESSAGE 'Provisioned throughput exceeded limit.' TYPE 'E'.
CATCH /aws1/cx_texthrottlingex.
  MESSAGE 'The request processing exceeded the limit.' TYPE 'E'.
CATCH /aws1/cx_texunsupporteddocex.
  MESSAGE 'The document is not supported.' TYPE 'E'.
ENDTRY.

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

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

```



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

- 有关 API 详细信息，请参阅《适用于 SAP ABAP 的 AWS SDK 的 API 参考》中的以下主题。
- [GetDocumentAnalysis](#)
- [StartDocumentAnalysis](#)

使用 SDK for SAP ABAP 的 Amazon Translate 示例

以下代码示例向您展示了如何使用适用于 SAP ABAP 的 AWS 软件开发工具包和 Amazon Translate 来执行操作和实现常见场景。

操作是大型程序的代码摘录，必须在上下文中运行。您可以通过操作了解如何调用单个服务函数，还可以通过函数相关场景的上下文查看操作。

场景是向您演示如何通过在一个服务中调用多个函数或与其他 AWS 服务结合来完成特定任务的代码示例。

每个示例都包含一个指向完整源代码的链接，您可以从中找到有关如何在上下文中设置和运行代码的说明。

主题

- [操作](#)
- [场景](#)

操作

DescribeTextTranslationJob

以下代码示例演示了如何使用 DescribeTextTranslationJob。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
"Gets the properties associated with an asynchronous batch translation job."
"Includes properties such as name, ID, status, source and target languages, and
input/output Amazon Simple Storage Service (Amazon S3) buckets."
TRY.
    oo_result = lo_xl8->describetexttranslationjob(      "oo_result is returned
for testing purposes."
    iv_jobid      = iv_jobid ).
    MESSAGE 'Job description retrieved.' TYPE 'I'.
CATCH /aws1/cx_xl8internalserverex.
    MESSAGE 'An internal server error occurred. Retry your request.' TYPE 'E'.
CATCH /aws1/cx_xl8resourcenotfoundex.
    MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
CATCH /aws1/cx_xl8toomanyrequestsex.
    MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
ENDTRY.
```

- 有关 API 的详细信息，请参阅适用[DescribeTextTranslationJob](#)于 S AP 的AWS SDK ABAP API 参考。

ListTextTranslationJobs

以下代码示例演示了如何使用 ListTextTranslationJobs。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```

"Gets a list of the batch translation jobs that you have submitted."

DATA lo_filter TYPE REF TO /aws1/cl_xl8textxl8translationjobfilt.

"Create an ABAP object for filtering using jobname."
lo_filter = NEW #( iv_jobname = iv_jobname ).

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

```

- 有关 API 的详细信息，请参阅适用[ListTextTranslationJobs](#)于 S AP 的AWS SDK ABAP API 参考。

StartTextTranslationJob

以下代码示例演示了如何使用 StartTextTranslationJob。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
"Starts an asynchronous batch translation job."
```

"Use batch translation jobs to translate large volumes of text across multiple documents at once."

```

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

"Create an ABAP object for the input data config."
lo_inputdataconfig = NEW #( iv_s3uri = iv_input_data_s3uri
                           iv_contenttype = iv_input_data_contenttype ).

"Create an ABAP object for the output data config."
lo_outputdataconfig = NEW #( iv_s3uri = iv_output_data_s3uri ).

"Create an internal table for target languages."
lo_targetlanguagecodes = NEW #( iv_value = iv_targetlanguagecode ).
INSERT lo_targetlanguagecodes INTO TABLE lt_targetlanguagecodes.

TRY.
    oo_result = lo_xl8->starttexttranslationjob(      "oo_result is returned for
testing purposes."
    io_inputdataconfig = lo_inputdataconfig
    io_outputdataconfig = lo_outputdataconfig
    it_targetlanguagecodes = lt_targetlanguagecodes
    iv_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_xl8unsuppedlanguage00.
        MESSAGE 'Amazon Translate does not support translation from the language of
the source text into the requested target language.' TYPE 'E'.

```

```
ENDTRY.
```

- 有关 API 的详细信息，请参阅适用[StartTextTranslationJob](#)于 S AP 的AWS SDK ABAP API 参考。

StopTextTranslationJob

以下代码示例演示了如何使用 StopTextTranslationJob。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

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

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

- 有关 API 的详细信息，请参阅适用[StopTextTranslationJob](#)于 S AP 的AWS SDK ABAP API 参考。

TranslateText

以下代码示例演示了如何使用 TranslateText。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
"Translates input text from the source language to the target language."
TRY.
    oo_result = lo_xl8->translatetext(      "oo_result is returned for testing
purposes."
        iv_text      = iv_text
        iv_sourcelanguagecode = iv_sourcelanguagecode
        iv_targetlanguagecode = iv_targetlanguagecode ).
    MESSAGE 'Translation completed.' TYPE 'I'.
CATCH /aws1/cx_xl8detectedlanguage00.
    MESSAGE 'The confidence that Amazon Comprehend accurately detected the
source language is low.' TYPE 'E'.
CATCH /aws1/cx_xl8internalserverex.
    MESSAGE 'An internal server error occurred.' TYPE 'E'.
CATCH /aws1/cx_xl8invalidrequestex.
    MESSAGE 'The request that you made is not valid.' TYPE 'E'.
CATCH /aws1/cx_xl8resourcenotfoundex.
    MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
CATCH /aws1/cx_xl8serviceunavailex.
    MESSAGE 'The Amazon Translate service is temporarily unavailable.' TYPE 'E'.
CATCH /aws1/cx_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_xl8unsuppedlanguage00.
    MESSAGE 'Amazon Translate does not support translation from the language of
the source text into the requested target language. ' TYPE 'E'.
ENDTRY.
```

- 有关 API 的详细信息，请参阅适用[TranslateText](#)于 S AP 的AWS SDK ABAP API 参考。

场景

开始使用翻译任务

以下代码示例显示了操作流程：

- 启动异步批处理翻译任务。
- 等待异步任务完成。
- 描述异步任务。

适用于 SAP ABAP 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

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

"Create an ABAP object for the input data config."
lo_inputdataconfig = NEW #( iv_s3uri = iv_input_data_s3uri
                           iv_contenttype = iv_input_data_contenttype ).

"Create an ABAP object for the output data config."
lo_outputdataconfig = NEW #( iv_s3uri = iv_output_data_s3uri ).

"Create an internal table for target languages."
lo_targetlanguagecodes = NEW #( iv_value = iv_targetlanguagecode ).
INSERT lo_targetlanguagecodes INTO TABLE lt_targetlanguagecodes.

TRY.
    DATA(lo_translationjob_result) = lo_xl8->starttexttranslationjob(
```

```

        io_inputdataconfig = lo_inputdataconfig
        io_outputdataconfig = lo_outputdataconfig
        it_targetlanguagecodes = lt_targetlanguagecodes
        iv_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_texttranslationjobproperties( )-
>get_jobstatus( ) <> 'COMPLETED'.
    IF sy-index = 30.
        EXIT.                "Maximum 900 seconds."
    ENDIF.
    WAIT UP TO 30 SECONDS.
    lo_des_translation_result = lo_xl8->describetexttranslationjob( iv_jobid =
lv_jobid ).
ENDWHILE.

TRY.
    oo_result = lo_xl8->describetexttranslationjob(      "oo_result is returned
for testing purposes."
        iv_jobid      = lv_jobid ).
    MESSAGE 'Job description retrieved.' TYPE 'I'.

```



```
CATCH /aws1/cx_xl8internalserverex.  
    MESSAGE 'An internal server error occurred. Retry your request.' TYPE 'E'.  
CATCH /aws1/cx_xl8resourcenotfoundex.  
    MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.  
CATCH /aws1/cx_xl8toomanyrequestsex.  
    MESSAGE 'You have made too many requests within a short period of time.'  
TYPE 'E'.  
ENDTRY.
```

- 有关 API 详细信息，请参阅《适用于 SAP ABAP 的 AWS SDK 的 API 参考》中的以下主题。
 - [DescribeTextTranslationJob](#)
 - [StartTextTranslationJob](#)

安全性 适用于 SAP ABAP 的 AWS SDK

云安全 AWS 是重中之重。作为 AWS 客户，您可以受益于专为满足大多数安全敏感型组织的要求而构建的数据中心和网络架构。

安全是双方共同承担 AWS 的责任。[责任共担模式](#)将其描述为云的安全性和云中的安全性：

- 云安全 — AWS 负责保护在云 AWS 服务 中运行的基础架构 AWS Cloud。AWS 还为您提供可以安全使用的服务。作为[AWS 合规计划](#)的一部分，第三方审计师定期测试和验证我们安全的有效性。要了解适用于的合规计划 适用于 SAP ABAP 的 AWS SDK，请参阅“按合规计划划分[AWS 服务的范围](#)”中的“[按合规计划AWS 服务](#)”。
- 云端安全 — 您的责任由您 AWS 服务 使用的内容决定。您还需要对其他因素负责，包括您的数据的敏感性、您的公司的要求以及适用的法律法规。

本节涵盖以下主题。

主题

- [SAP 系统身份验证已开启 AWS](#)
- [IAM 安全最佳实践](#)
- [SAP 授权](#)
- [安全运营](#)
- [将证书与 IAM Roles Anywhere 结合使用](#)
- [使用 SAP 凭据存储](#)

SAP 系统身份验证已开启 AWS

AWS 在 SAP 系统能够代表 SAP 用户向发出呼叫之前，SAP 系统必须对其进行身份验证 AWS。适用于 SAP ABAP 的 AWS SDK 支持在中的 SDK 配置文件设置中选择的以下三种身份验证方法

AWS 适用于 SAP ABAP 的 SDK-BTP 版本只能使用 SAP 凭据存储使用[the section called “使用秘密访问密钥进行身份验证”](#)方法进行身份验证。

主题

- [Amazon EC2 实例元数据身份验证](#)

- [使用秘密访问密钥进行身份验证](#)
- [使用 IAM Roles Anywhere 进行基于证书的身份验证](#)
- [后续步骤](#)

Amazon EC2 实例元数据身份验证

在亚马逊上运行的 SAP 系统 EC2 可以从亚马逊实例元数据中获取短期、自动轮换的证书。EC2 有关更多信息，请参阅[使用 Amazon EC2 实例元数据的凭证](#)。

亚马逊强烈建议您在使用适用于 SAP ABAP 的 SDK 时选择这种身份验证方法。如需采用此方法，基础管理员必须启用 HTTP 出站通信。无需进一步基础配置。

Note

这种身份验证方法仅适用于您的 SAP 系统在 Amazon 上运行的情况 EC2。托管在本地或其他云环境中的 SAP 系统无法使用此方法进行身份验证。

使用秘密访问密钥进行身份验证

此方法使用访问密钥 ID 和秘密访问密钥，对 AWS SAP 系统进行身份验证。SAP 系统 AWS 使用 IAM 用户登录。有关更多信息，请参阅[管理适用于 IAM 用户的访问密钥](#)。

基础管理员会收到 AWS IAM 管理员发送的访问密钥 ID 和秘密访问密钥。您的 SAP 系统必须配置为存储访问密钥 ID 和私有访问密钥。

- 安全、存储和转发 (SSF)
 - 使用 SSF 功能对 SAP ABAP AWS 的 SDK 进行身份验证。有关更多信息，请参阅[数字签名和加密](#)。
 - 您还可以使用 SSF02 报告测试 SSF envelope 和 develope 功能。有关更多信息，请参阅[测试 SSF 安装流程](#)。
 - /AWS1/IMG 事务描述了为适用于 SAP ABAP 的 SDK 配置 SSF 的步骤。转到技术先决条件，然后选择适用于本地系统的其他设置。
- SAP 凭据存储
 - 使用 SAP 凭据存储对适用于 SAP ABAP 的 S AWS DK 进行身份验证——BTP 版本。有关更多信息，请参阅[什么是 SAP 凭据存储](#)？

- 有关配置步骤，请参阅[使用 SAP 凭据存储](#)。

使用 IAM Roles Anywhere 进行基于证书的身份验证

您的证书颁发机构 (CA) 颁发的 X.509 证书可用于对 Roles Anywhere 进行 AWS Identity and Access Management 身份验证。必须在 STRUST 中配置此证书。CA 必须在 IAM Roles Anywhere 中注册为信任锚点，并且必须创建配置文件以指定 IAM Roles Anywhere 将代入的角色和策略。有关更多信息，请参阅 [Roles Anywhere 中的 AWS Identity and Access Management 创建信任锚和个人资料](#)。

有关如何通过适用于 SAP ABAP 的 SDK 使用 IAM Roles Anywhere 的详细步骤，请参阅[将证书与 IAM Roles Anywhere 结合使用](#)。

Note

只有通过使用导入的证书吊销列表才能支持证书吊销。有关更多信息，请参阅[吊销](#)。

后续步骤

在中对您的 SAP 系统进行身份验证后 AWS，适用于 SAP ABAP 的 SDK 会自动执行，为该 SAP 用户的业务职能承担相应的 IAM 角色。sts:assumeRole

IAM 安全最佳实践

IAM 管理员将负责以下三个关键领域的安全。

- 确保 SAP 系统可以使用 Amazon EC2 元数据或密钥凭证进行身份验证。
- 确保 SAP 系统具有提升 sts:assumeRole 所需的权限。
- 为 SAP 用户创建 IAM 逻辑角色，其具有执行业务职能所需的权限（如执行 Amazon S3、DynamoDB 或其他服务所需的权限）。这些都是 SAP 用户将会代入的角色。

有关更多信息，请参阅 SAP Lens：AWS Well-Architected Framework 中的[安全](#)一章。

主题

- [Amazon EC2 实例配置文件的最佳实践](#)
- [适用于 SAP 用户的 IAM 角色](#)

Amazon EC2 实例配置文件的最佳实践

运行 SAP 系统的 Amazon EC2 实例具有一组基于其实例配置文件的授权。这类实例配置文件通常只需要具有调用 `sts:assumeRole` 的权限，即可允许 SAP 系统按需代入业务特定的 IAM 角色。提升其他角色可确保 ABAP 程序代入角色，向用户授予履职所需的最低权限。例如，实例配置文件可能包含以下声明。

```
{
  "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"
      ]
    }
  ]
}
```

前面的示例允许 SAP 系统担任 CFO、审计员或报告用户的 IAM 角色。AWS SDK 将根据用户在 SAP 中的 PFCG 角色为用户选择正确的 IAM 角色。

Amazon EC2 实例配置文件也可以用于其他功能。

- [AWS 适用于 SAP HANA 的 Backint Agent](#)
- [具有覆盖 IP 地址路由的 AWS 高可用性配置上的 SAP](#)

这些解决方案可能还需要特定于备份或失效转移的 `sts:assumeRole` 角色的权限，或者需要直接分配给实例配置文件的权限。

适用于 SAP 用户的 IAM 角色

ABAP 程序需要权限才能执行用户的工作：读取 DynamoDB 表、在 Amazon S3 中的 PDF 对象上调用 Amazon Textract、运行函数。AWS Lambda 所有产品都使用相同的安全模型 AWS SDKs。您可以使用另一个 AWS SDK 所用的现有 IAM 角色。

SAP 业务分析师将针对每个所需的逻辑角色，向 IAM 管理员要求 IAM 角色的 `arn:aws:`。例如，业务分析师会在财务场景中定义以下 IAM 逻辑角色。

- CFO
- AUDITOR
- REPORTING

IAM 管理员将为每个 IAM 逻辑角色定义 IAM 角色。

CFO

- `arn:aws:iam::0123456789:role/finance-cfo`
- Amazon S3 存储桶的读写权限
- DynamoDB 数据库的读写权限

AUDITOR

- `arn:aws:iam::0123456789:role/finance-auditor`
- Amazon S3 存储桶的读取权限
- DynamoDB 数据库的读取权限

REPORTING

- `arn:aws:iam::0123456789:role/finance-reporting`
- DynamoDB 数据库的读取权限
- 取消 Amazon S3 存储桶的权限

业务分析师将 IAM 角色输入映射表，实现 IAM 逻辑角色与 IAM 物理角色的映射。

SAP 用户的 IAM 角色需允许可信主体执行 `sts:assumeRole` 操作。根据 SAP 系统在 AWS 上的身份验证方式，可信主体有所不同。有关更多信息，请参阅[指定主体](#)。

以下列举了部分最常见的 SAP 场景示例：

- 在亚马逊 EC2 上运行的 SAP 系统分配了实例配置文件 — 在这里，Amazon EC2 实例配置文件附加到 IAM 角色。

```
{
```

```

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

```

- 在没有@@ 实例配置文件 EC2 的情况下在亚马逊上运行的 SAP 系统——在这里，亚马逊 EC2 担任 SAP 用户的角色。

```

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

```

- 本地运行的 SAP 系统：本地运行的 SAP 系统只能使用秘密访问密钥验证身份。有关更多信息，请参阅[在 AWS 上验证 SAP 系统](#)。

SAP 用户必须与其代入的所有 IAM 角色建立信任关系。

```

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

```

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

SAP 授权

配置 SDK 所需的授权取决于 SDK 的版本。

主题

- [配置授权](#)
- [最终用户的 SAP 授权](#)

配置授权

有关更多详细信息，请参阅以下选项卡。

SDK for SAP ABAP

必须具有以下权限才能配置适用于 SAP ABAP 的 SDK。

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

从以下授权组中选择：

- /AWS1/CFG- 适用于 SAP ABAP 的 AWS SDK v1-Config
- /AWS1/MOD- 适用于 SAP ABAP 的 AWS SDK v1-运行时间
- /AWS1/PFL- 适用于 SAP ABAP 的 AWS SDK v1-软件开发工具包配置文件
- /AWS1/RES- 适用于 SAP ABAP 的 AWS SDK v1-逻辑资源

- /AWS1/TRC- 适用于 SAP ABAP 的 AWS SDK v1-Trace

SDK for SAP ABAP - BTP edition

使用以下步骤允许适用于 SAP ABAP 的 SDK-BTP 版本访问配置。

1. 使用业务角色模板创建新的SAP_BR_BPC_EXPERT业务角色。此模板提供对“自定义业务配置”应用程序的访问权限。
2. 在“一般角色详细信息”下，转到“访问类别”，然后在“读取、写入、值帮助”中选择“不受限制”。
3. 前往业务目录选项卡，然后分配/AWS1/RTBTP_BCAT业务目录以提供对 SDK 配置的访问权限。
4. 前往企业用户选项卡，然后分配业务用户以授予对 SDK 配置的访问权限。

最终用户的 SAP 授权

先决条件：定义 SDK 配置文件

在 SAP 安全管理员定义其角色之前，业务分析师将在交易/AWS1/IMG中为 SAP ABAP 的 SD AWS K 或适用于 SAP ABAP 的 SDK 的自定义业务配置应用程序——BTP 版本定义 SDK 配置文件。SDK 配置文件通常根据业务职能命名：ZFINANCE、ZBILLING、ZMFG、ZPAYROLL 等。业务分析师会使用短名称（如 CFO、AUDITOR 和 REPORTING），为每个 SDK 配置文件定义 IAM 逻辑角色。IAM 安全管理员会将这些角色映射到 IAM 真实角色。

定义 PFCG 或业务角色

Note

在 SAP BTP、ABAP 环境中，PFCG 角色被称为业务角色。

随后，SAP 安全管理员会添加授权对象 /AWS1/SESS，授予对 SDK 配置文件的访问权限。

身份验证对象 /AWS1/SESS

- 字段 /AWS1/PROF = ZFINANCE

还应根据用户的工作职能，将其映射到各 SDK 配置文件的 IAM 逻辑角色。例如，财务审计员具有报告访问权限，可能会映射到 AUDITOR IAM 逻辑角色。

身份验证对象 /AWS1/LROL

- 字段 /AWS1/PROF = ZFINANCE
- 字段 /AWS1/LROL = AUDITOR

同时，CFO 具有读/写权限，可能会获得 PFCG 角色中的 CFO 逻辑角色。

身份验证对象 /AWS1/LROL

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

在每个 SDK 配置文件中，每位用户通常只能获得一个 IAM 逻辑角色。如果用户获得多个 IAM 角色的授权（例如，如果首席财务官同时获得两个角色 CFO 和 AUDITOR 逻辑 IAM 角色的授权），那么 AWS SDK 会确保优先级更高（序列号较低）的角色生效，从而打破局面。

与所有安全场景一样，应授予用户履行其工作职能的最低权限。根据 SDK 配置文件及其授权的逻辑角色来命名单一 PFCG 角色，这是管理 PFCG 角色的一种简单策略。例如，Z_AWS_PFL_ZFINANCE_CFO 角色可授予对 ZFINANCE 配置文件和 CFO IAM 逻辑角色的访问权限。这些单一角色可分配给负责定义工作职能的复合角色。每家公司都有独一无二的角色管理策略，亚马逊鼓励您制定适合贵公司的 PFCG 策略。

安全运营

静态数据加密

AWS 秘密访问密钥用于对 SDK 进行身份验证。它们由 SAP 使用 SSF 或凭据存储功能进行加密。

传输中数据加密

所有对的 AWS 服务 呼叫都使用 HTTPS 加密。SAP ICM 管理 HTTPS 连接。AWS 在 STRUST 中必须信任证书。

API 用法

当 ABAP 用户使用 `sts:assumeRole` 代入角色时，会话名称为 `USERID-SID-MANDT`，其中：

- USERID 是 SY-UNAME 变量中的 ABAP 用户。
- SID 是 SY-SYSID 变量中的 ABAP 系统 ID。
- MANDT 是 SY-MANDT 变量中的 ABAP 客户端。

会话名称 CloudTrail 作为用户名出现在中。这样就能确保从启动 API 调用的系统、客户端和用户源头跟踪 ABAP 用户。有关更多信息，请参阅[什么是 AWS CloudTrail？](#)

将证书与 IAM Roles Anywhere 结合使用

AWS 通过使用基于证书的身份验证和 Roles Anywhere，可以在 SAP 系统上进行身份验证 AWS Identity and Access Management。您必须在 STRUST 中设置证书，并在 /AWS1/IMG 中配置 SDK 配置文件。

先决条件

在开始设置证书之前，必须满足以下先决条件。

- 您的证书颁发机构 (CA) 颁发的 X.509 证书必须满足以下要求。
 - 签名证书必须是 v3 证书。
 - 链不得超过 5 个证书。
 - 证书必须支持 RSA 或 ECDSA 算法。
- 在 IAM Roles Anywhere 中将您的 CA 注册为信任锚点，然后创建一个配置文件来指定 IAM Roles Anywhere 的角色/策略。有关更多信息，请参阅 [Roles Anywhere 中的 AWS Identity and Access Management 创建信任锚和个人资料](#)。
- SAP 用户的 IAM 角色必须由 IAM 管理员创建。角色必须具有调用所需角色的权限 AWS 服务。有关更多信息，请参阅 [IAM 安全最佳实践](#)。
- 创建授权以运行 /AWS1/IMG 事务。有关更多信息，请参阅[针对配置的授权](#)。

过程

按照以下说明设置基于证书的身份验证。

步骤

- [步骤 1 – 使用 SAP 的安全存储和转发 \(SSF\) 定义 SSF 应用程序](#)
- [步骤 2 – 设置 SSF 参数](#)

- [步骤 3 - 创建 PSE 和证书请求](#)
- [步骤 4 - 将证书响应导入到相关的 PSE](#)
- [步骤 5 - 配置 SDK 配置文件以使用 IAM Roles Anywhere](#)

步骤 1 – 使用 SAP 的安全存储和转发 (SSF) 定义 SSF 应用程序

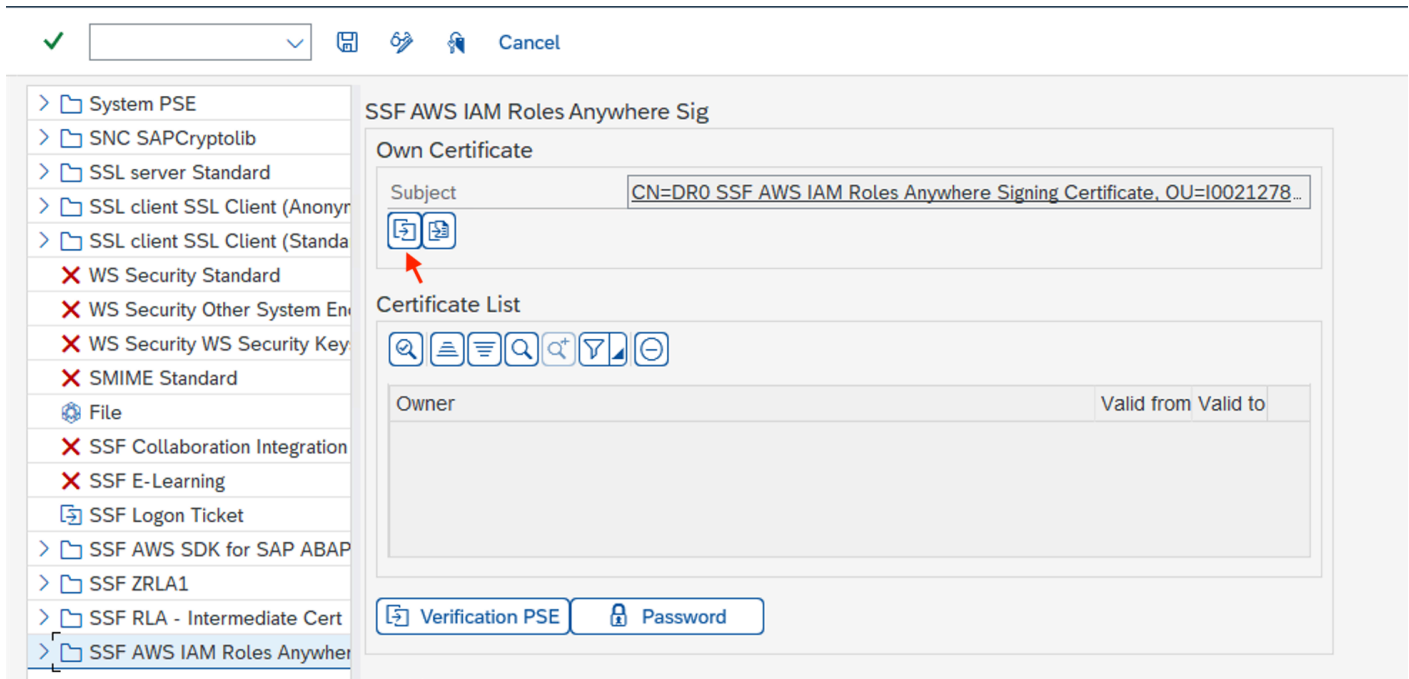
1. 运行事务代码 SE16 来定义 SSF 应用程序。
2. 输入 SSFAPPLIC 表名称，然后选择新建条目。
3. 在 APPLIC 字段中输入 SSF 应用程序的名称，在 DESCRIPT 字段中输入描述，并为其余字段选择 Selected (X) 选项。

步骤 2 – 设置 SSF 参数

1. 运行启动 /n/AWS1/IMG 适用于 SAP ABAP 的 AWS SDK 实施指南 (IMG)。
2. 选择 适用于 SAP ABAP 的 AWS SDK 设置 > 技术先决条件 > 适用于本地系统的其它设置。
3. 运行设置 SSF 参数 IMG 活动。
4. 选择新建条目，然后选择在上一步中创建的 SSF 应用程序。选择保存。
5. 将哈希算法修改为 SHA256，将加密算法修改为 AES256-CBC。将其它设置保留为默认设置，然后选择保存。

步骤 3 - 创建 PSE 和证书请求

1. 运行 /n/AWS1/IMG 事务，然后选择 适用于 SAP ABAP 的 AWS SDK 设置 > 技术先决条件 > 适用于本地系统的其它设置。
2. 运行 Create PSE for SSF Application IMG 活动。
3. 对于 STRUST 事务选择编辑。
4. 右键选择在 [the section called “步骤 1”](#) 中创建的 SSF 应用程序，然后选择创建。保留所有其它默认设置，然后选择继续。
5. 选择创建证书请求。参阅下图。保留默认选项，然后选择继续。复制或导出生成的证书请求，并将其提供给您的 CA。您的 CA 会验证请求，并使用签名的公有密钥证书进行响应。



签名过程因您的 CA 及其使用的技术而异。有关示例，请参阅使用[私有证书颁发机构颁发 AWS 私有终端实体证书](#)。

步骤 4 - 将证书响应导入到相关的 PSE

1. 运行 /n/AWS1/IMG 事务，然后选择 适用于 SAP ABAP 的 AWS SDK 设置 > 技术先决条件 > 适用于本地系统的其它设置。
2. 运行 Create PSE for SSF Application IMG 活动。
3. 对于 STRUST 事务选择编辑。
4. 选择 SSF 应用程序，然后选择位于主题下方 PSE 部分中的导入证书响应。将证书响应复制并粘贴到文本框中，或者从文件系统导入文件。选择继续 > 保存。
5. 通过两次选择主题即可查看证书详细信息。信息显示在证书部分中。

步骤 5 – 配置 SDK 配置文件以使用 IAM Roles Anywhere

1. 运行事/n/AWS1/IMG务，然后选择适用于 SAP ABAP 的 AWS SDK 设置 > 应用程序配置。
2. 创建新的 SDK 配置文件并对其命名。
3. 选择 IAM Roles Anywhere 作为身份验证方法。
 - 在左侧窗格中，选择身份验证和设置。

- 创建一个新条目，然后输入 SAP 系统的信息以及 AWS 区域。
- 对于身份验证方法选择 IAM Roles Anywhere，然后选择保存。
- 选择输入详细信息，然后在弹出窗口中，选择在[the section called “步骤 1”](#)中创建的 SSF 应用程序。输入在[the section called “先决条件”](#)中创建的信任锚点 ARN 和配置文件 ARN。参阅下图。选择继续。

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. 在左侧窗格中，选择 IAM 角色映射。输入名称，并提供 IAM 管理员提供的 IAM 角色的 ARN。

有关更多信息，请参阅[应用程序配置](#)。

使用 SAP 凭据存储

SAP 凭据存储在 SAP 业务技术平台中用于安全地存储用于秘密访问密钥身份验证的凭据。AWS 您必须订阅才能使用该服务。

以下说明假设您已经配置了 SDK 配置文件。有关更多信息，请参阅[配置 适用于 SAP ABAP 的 AWS SDK](#)。

在开始配置之前，请确保满足先决条件。有关更多信息，请参阅[SAP 凭据存储](#)。

主题

- [配置步骤](#)
- [将 SAP 凭据存储与 SDK 配合使用](#)

配置步骤

步骤

- [步骤 1：配置身份验证设置](#)
- [步骤 2：创建服务密钥](#)
- [步骤 3：将服务密钥转换为.p12格式](#)
- [第 4 步：连接到 SAP BTP、ABAP 环境](#)

步骤 1：配置身份验证设置

使用以下步骤配置用于身份验证的凭据存储设置。

1. 导航到 SAP 凭据存储实例的设置选项卡。
2. 选择编辑配置：
 - 选择双向 TLS 作为默认身份验证类型。
 - 在“有效负载加密状态”中选择“禁用”。有效载荷在传输过程中使用 HTTPS 加密。但是，目前无法对有效载荷进行双重加密。
3. 选择保存。

步骤 2：创建服务密钥

使用以下步骤为凭据存储创建服务密钥。

1. 在 SAP 凭据存储应用程序的左侧窗格中，导航到服务密钥。
2. 选择“创建服务密钥”。
3. 输入服务密钥的名称，然后选择创建。

服务密钥是根据所选的身份验证类型创建的。下载服务密钥，并妥善保管，以备日后使用。

步骤 3：将服务密钥转换为 .p12 格式

为通信系统创建出站用户需要.p12格式为的客户端证书。使用以下步骤根据凭据存储服务密钥中提供的证书详细信息生成证书。 .p12

1. 从 SAP 信任中心服务下载 SAP 云根 CA [证书 \(SAP 需要 \)](#)。

2. 以任何文本文件格式打开 SAP 云根 CA 证书。在文件末尾，按 Enter，然后从服务密钥的证书字段中复制粘贴证书。将换行符替换为实际的换行符\n (Enter)，然后以 .cer 文件格式保存整个证书。
3. 从服务密钥的密钥字段中复制密钥。必须将此私钥视为敏感数据。将其粘贴到文本文件中，然后将换行符替换为实际\n的换行符 (Enter)。将私钥保存在文本文件中。
4. 使用前面步骤中生成的证书和私钥，运行以下命令来生成 .p12 证书。

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

该命令需要验证导出密码。保留密码以备将来使用。

删除保存在私钥中的 .key 文本文件。

第 4 步：连接到 SAP BTP、ABAP 环境

将 SAP BTP、ABAP 环境配置为连接 SAP 凭据存储。

主题

- [通信系统](#)
- [沟通安排](#)

通信系统

使用以下步骤创建通信系统，该系统允许从 SAP BTP、ABAP 环境到 SAP 凭据存储区进行通信。

1. 打开 ABAP 环境系统的 Fiori 启动板。
2. 选择“通信系统”图块以打开应用程序。
3. 选择新建。
4. 输入通信系统的名称和 ID，然后选择创建。例如，您可以为系统命名 ZSAP_CREDSTORE。
5. 输入其他必填信息：
 - 主机名：从服务密钥 URL 中复制主机名。例如，如果 URL 是 `https://credstore.mesh.cf.us10.hana.ondemand.com/api/v1/credentials`，则主机名是 `credstore.mesh.cf.us10.hana.ondemand.com`。
 - 出站通信用户：选择+此选项可添加新用户。
 - a. 选择 SSL 客户端证书作为身份验证机制。

b. 选择上传新证书：

- 浏览在上一步中生成的 .p12 证书。
- 输入描述。
- 输入用于生成 .p12 证书的导出密码。
- 选择上传。

c. 选择“创建”以创建出站用户。

6. 选择保存。

7. 删除在上一步中下载的服务密钥。

沟通安排

使用以下步骤创建通信安排，为出站通信提供通信场景。

1. 打开 ABAP 环境系统的 Fiori 启动板。
2. 选择“通信安排”图块以打开应用程序。
3. 选择新建。
4. 选择通信场景/AWS1/CRED_COMM_SCENARIO，然后输入通信安排的名称。例如，Z_AWS_SDK_TO_SAP_CREDSTORE。
5. 选择创建。
6. 在“通信系统”字段中，浏览到在上一步中创建的通信系统。其他信息是自动填充系统的帖子选择。
7. 选择保存。
8. 选择“检查连接”以测试您的连接。

设置完成后，ABAP 环境可以使用通信安排，通过出站服务 (HTTP) 使用 SAP 凭据存储服务。

将 SAP 凭据存储与 SDK 配合使用

步骤

- [步骤 1：创建命名空间和凭证](#)
- [步骤 2：配置自定义业务配置应用程序](#)

步骤 1：创建命名空间和凭证

使用 SAP 帮助 — [创建、编辑和删除凭据，在 SAP 凭据存储中创建命名空间和凭据](#)。

输入以下详细信息以创建密钥类型的凭证。

- 命名空间-输入命名空间的名称，并将相关凭据组合在一起。
- 名称-输入密钥的名称。我们建议aws-0123456789012-username，在哪里：
 - 0123456789012是凭 AWS 账户 证向其授予访问权限的 ID
 - username是证书所属的 IAM 用户名
- 值-输入 base-64 编码的私有访问密钥。使用以下命令对您的私有访问密钥进行 base-64 编码。

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

该命令从标准输入中读取私有访问密钥，并将其传递给 base64，不带尾随换行符。它将以 base-64 编码的秘密访问密钥输出到屏幕。将值复制到 SAP 凭据存储区后，请清除或关闭终端。

- 用户名-输入您的访问密钥 ID。
- 选择创建。

创建了一个具有一个凭据的新命名空间，并且可以在此命名空间中添加、删除或修改证书。

遵循最低权限原则来管理对存储在命名空间中的凭据的访问权限。

步骤 2：配置自定义业务配置应用程序

使用以下步骤配置自定义业务配置应用程序，以定义用于由 SDK 进行身份验证的凭据。

1. 打开 ABAP 环境系统的 Fiori 启动板。
2. 浏览自定义业务配置图块以打开应用程序。
3. 打开 SDK 配置文件业务配置。
4. 选择必须为 SAP 凭据存储配置身份验证设置的 SDK 配置文件。
5. 在所选配置文件的“身份验证和设置”选项卡中，选择“编辑”，然后输入以下详细信息：
 - 身份验证方法-从 SAP 凭据存储中选择凭据。
 - 命名空间-输入在 SAP 凭据存储中创建的命名空间。有关更多信息，请参阅 [the section called “步骤 1：创建命名空间和凭证”](#)。
 - 密钥名称-输入已创建的密钥凭据的名称。有关更多信息，请参阅 [the section called “步骤 1：创建命名空间和凭证”](#)。

- 通信安排-输入已创建的通信安排的名称。有关更多信息，请参阅 [the section called “沟通安排”](#)。
6. 选择“应用”以转至“AWS SDK 配置文件”屏幕。
 7. 选择“选择传输”，使用值 help 选择传输。
 8. 选择保存。

故障排除 适用于 SAP ABAP 的 AWS SDK

本节针对可能出现的错误场景介绍了问题排查步骤。

主题

- [导入失败](#)
- [未指定的位置限制](#)
- [SSL 错误](#)
- [配置文件配置](#)
- [IAM 授权](#)
- [必要操作执行授权](#)
- [活动场景](#)
- [代码中的特殊字符](#)
- [连接](#)

导入失败

问题 — “CL_SYSTEM_UUID” 类不包含接口 'IF_SYSTEM_UUID_' _STATIC RFC4122

原因：系统缺失 SAP Note 0002619546

解决方案：确保对系统应用 [SAP Note 0002619546](#)

未指定的位置限制

问题：未指定的位置限制无法兼容用于接收请求的 region 特定端点

原因-您的 Amazon S3 存储桶中缺少 AWS 区域io_createbucketconfiguration参数。

解决方案：在任意区域 (us-east-1 除外) 创建存储桶，同时使用 createbucket() 中的 io_createbucketconfiguration 参数为 Amazon S3 存储桶指定区域 并不一定要为 us-east-1 指定限制。

以下示例显示了正确配置的 io_createbucketconfiguration 参数。

```
createbucket(
```

```
iv_bucket = 'amzn-s3-demo-bucket'  
io_createbucketconfiguration = NEW /aws1/cl_s3_createbucketconf( 'us-west-1' )  
).
```

SSL 错误

问题：SSL 服务器证书与主机名不匹配或 SSL 与 docs.aws.amazon.com:443 握手失败：
SSSLERR_NO_SSL_RESPONSE

原因：没有为 DEFAULT 配置文件的 TRUE 设置 icm/HTTPS/client_sni_enabled 参数

解决方案：根据以下问题排查步骤，解决给定问题或其他与 SSL 有关的问题

1. 打开 SAPGUI 并转到命令栏。
2. 运行 RZ10 事务。
3. 转到配置文件并选择 DEFAULT 配置文件。系统会自动填充版本。
4. 在编辑配置文件部分，选择扩展维护，然后选择更改。
5. 搜索 icm/HTTPS/client_sni_enabled 参数。
 - 如果存在此参数，则编辑参数值并将其设为 TRUE。
 - 如果不存在此参数，则使用以下步骤创建参数。
 1. 选择参数。

Note

请确保您选择的是创建型参数，而不是编辑型参数（铅笔图标）。

2. 在参数名称字段中输入 icm/HTTPS/client_sni_enabled。
 3. 在参数值字段中输入 TRUE。
 4. 选择保存。
6. 将上述更改保存在 DEFAULT 配置文件中，然后退出。

配置文件配置

问题：无法使用 <sid>:<client> 默认场景查找 <profile_name> 配置文件下的配置

原因：<profile_name> 不正确或尚未配置

解决方案：根据以下步骤设定配置文件

1. 打开 SAPGUI 并运行 /n/AWS1/IMG 事务。
2. 转到应用程序配置 > SDK 配置文件。
 - 若已设定配置文件，则验证配置文件名称是否正确。
 - 若未设定配置文件，则按步骤操作。
3. 选择新建条目。
 - a. 输入配置文件的名称和描述。
 - b. 选择保存。
4. 选择上一步创建的条目，然后选择身份验证和设置。
5. 选择新建条目，输入以下详细信息，然后选择保存。
 - SID
 - 客户端
 - 场景 ID
 - AWS 区域
 - 身份验证方法
 - 通过元数据为中运行的 SAP 系统选择实例角色 AWS。
 - 为运行在本地或其他云中的 SAP 系统选择 SSF 存储中的凭证。
6. 选择 IAM 角色映射 > 新建条目，输入以下详细信息，然后选择保存。
 - 序列号
 - IAM 逻辑角色
 - IAM 角色 ARN

IAM 授权

问题-无法担任角色<iam_role_arn>或用户:<user_arn>无权AssumeRole 在资源上执行:sts:<iam_role_arn>

原因：以下可能是造成此错误的原因

- 指定的 IAM 角色 ARN 不正确
- IAM 用户无权访问 IAM 角色
- 已代入的 IAM 角色未与正在代入的 IAM 角色或 IAM 用户建立信任关系

解决方案：采用以下步骤，确保 IAM 角色 ARN 正确无误

1. 打开 SAPGUI 并运行 /n/AWS1/IMG 事务。
2. 转到应用程序配置 > SDK 配置文件，然后选择经过 IAM 角色配置的配置文件。
3. 选择 IAM 角色映射，然后验证或更正 IAM 角色 ARN。
 - 如果 IAM 角色 ARN 正确，则确保已妥善配置 IAM 角色。有关更多信息，请参阅 [IAM 角色问题排查](#)。

必要操作执行授权

问题：<user_arn> 用户无权在 <resource_arn> 资源上执行 <action> 操作

原因：用户无权执行操作

解决方案：必须使用 resource_arn 的所需权限设置 user_arn，以执行指定的 action 操作 有关更多信息，请参阅[访问 IAM 资源所需的权限](#)。

活动场景

问题：未配置活动场景

原因：缺失活动场景设置

解决方案：请参阅[运行时设置](#)，以配置活动场景

代码中的特殊字符

警告：ABAP 单词中不得含有 0x00A0 字符

Note

在收到这条警告之前，您可能会先收到各种错误消息。

原因：复制并粘贴不同来源的代码，会在您的代码中插入特殊字符

解决方案：在粘贴 ABAP 源代码编辑器中的代码时会看到以下弹出窗口

检测到不间断空格字符。是否转换为空格？

选择是作为答案。此外，亚马逊建议您选定要复制的代码，而不是使用代码框中的复制按钮。

连接

问题：SCLNT_HTTP(411)：未能直接连接到 tla.region.amazonaws.com:443：
NIECONN_REFUSED(-10)

原因：SAP 系统未联网，无法与 tla.region.amazonaws.com 的端口 443 建立 TCP/IP 连接

解决方案 — SAP 系统必须能够直接或通过代理服务器与 HTTPS AWS 端口 443 上的端点建立连接。
您可以使用以下选项之一建立/验证互联网连接。

- 通过 NAT 或互联网网关，直接与互联网建立出站连接
- 通过代理服务器连接

有关更多信息，请参阅[通过代理服务器连接](#)。

其他主题

本节涵盖以下主题。

主题

- [适用于 SAP ABAP 的 AWS SDK 发布](#)
- [SAP 许可](#)

适用于 SAP ABAP 的 AWS SDK 发布

AWS 适用于 SAP ABAP 的 SDK 以传输形式交付，适用于 SAP ABAP 的 S AWS DK-BTP 版本作为附加组件提供。导入传输和附加组件的机制不同，但技术功能相同。有关更多信息，请参阅[设置](#)。

主题

- [版本策略](#)
- [最佳实践](#)
- [修补适用于 SAP ABAP 的 SDK](#)
- [安装其他模块](#)
- [卸载适用于 SAP ABAP 的 SDK](#)

版本策略

的 适用于 SAP ABAP 的 AWS SDK 版本 1 经常更新。根据的发布和更新，每周或每天都会发布新补丁 AWS 服务。的补丁 AWS 服务 可能包括错误修复和其他更新 SDK 补丁级别的更改。有关更多信息，请参阅[AWS SDKs 和工具维护政策](#)。

最佳实践

亚马逊建议为所有 SAP 系统（开发、QA 和生产系统）保留适用于 SAP ABAP 的 SDK 的相同补丁级别。

在修补 SDK 时，向沙盒导入最新版本。然后按照常规更改控制程序，向开发、QA 和生产系统导入此版本。

修补适用于 SAP ABAP 的 SDK

适用于 SAP ABAP 的 SDK 的每个版本都是以一套累积传输的形式交付的，其中包括所有错误修复、功能和更新内容。补丁和安装传输是一样的。为修补适用于 SAP ABAP 的 SDK，请务必导入最新传输版本。

鉴于 core 运行时系统与 API 模块的依赖关系，即使您不再使用这些模块，也必须修补 core 模块和其他所有已安装的模块。例如，如果您在安装 SDK 时导入了 core、ec2 和 lmd 传输，则必须在修补时导入 core、ec2 和 lmd 的最新传输版本。

安装其他模块

将传输版本导入与现有 core 和模块具有相同补丁级别的新模块，从而在 SAP 系统中安装其他 API 模块。如需导入更新版本的模块，请遵循 [the section called “修补适用于 SAP ABAP 的 SDK”](#) 指南。以此确保补丁级别可兼容所有 SDK 模块。

卸载适用于 SAP ABAP 的 SDK

要卸载适用于 SAP ABAP 的 SDK，你必须从 <https://sdk-for-sapabap.aws.amazon.com/awsSdkSapabapV1/release/uninstall-abapsdk-LATEST.zip> 下载删除传输套件。

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

你可以从 <https://sdk-for-sapabap.aws.amazon.com/awsSdkSapabapV1/release/uninstall-abapsdk-latest.sig> 下载签名文件。如需验证此文件，请参阅[验证适用于 SAP ABAP 的 SDK](#)。

对于安装在 SAP 系统上的所有 SDK 模块，请务必从上述 ZIP 文件中导入相应的删除传输套件。无需卸载整个 SDK 即可移除单个模块。只要向需移除的模块导入删除传输套件即可。如需卸载整个 SDK 及其所有模块，则必须在最后导入核心删除传输。

亚马逊建议您先在沙盒中测试卸载流程，再将其应用于开发、QA 或生产系统。

注意事项

请先阅读以下注意事项再卸载 SDK。

- 来自的 SDK 配置设置将丢失。必须重新配置 IMG 才能完成安装。
- SDK 移除后，基于 SDK 运行的 Z 程序会生成语法错误。

- 移除 SDK 后，包含 SDK 授权参考的 PFCG 或企业角色的授权将无效。请先删除 PFCG 角色中的 SDK 授权参考，再卸载 SDK。

Note

AWS 开发者预览版期间无法卸载适用于 SAP ABAP 的 SDK-BTP 版本。

SAP 许可

请根据 SAP 条款使用 SAP 软件。您有责任遵守 SAP 许可条款，包括软件发行和间接许可条件。此条款所述信息并非法律意见，不应用于许可合规目的。如果您对 SAP 软件的许可或权利有任何疑问，请咨询您的法律团队、SAP 或 SAP 分销商。

问题：使用适用于 SAP ABAP 的 SDK 会影响我的 SAP 许可证吗？

答案：适用于 SAP ABAP 的 AWS SDK 允许您使用自己的 ABAP 代码 AWS 服务 进行消费。可用于集成 SAP 系统和 AWS 服务。向第三方（非 SAP）系统发送 SAP 系统数据，或使用此系统创建数据的所有场景都会对间接许可产生影响。SAP 可通过多种方法定义间接访问，例如基于用户和结果进行计算。您与 SAP 签订的合同决定了间接访问的定义方法。您必须掌握此合同中提供的指南，可与 SAP 或其经销商展开进一步讨论。

SAP 在 2018 年发布了两份文件：适用于已安装 SAP 的基础用户的间接访问指南和数字时代的 SAP ERP 定价策略：解决间接/数字访问风险。这两份文件都是间接许可方式的示例，可在 SAP 网站上查看。但这些文件并不能反映您与 SAP 签订的特定协议。

《适用于 SAP ABAP 的 AWS SDK 开发者指南》的文档历史记录

下表描述了文档版本 适用于 SAP ABAP 的 AWS SDK。

变更	说明	日期
新增内容	适用于 SAP ABAP 的 SDK 的开发者预览版——BTP 版。	2024 年 5 月 31 日
新增内容	添加了 将证书与 IAM Roles Anywhere 结合使用 。	2023 年 12 月 1 日
新增内容	添加了 使用 SDK 构建产品 。	2023 年 12 月 1 日
新增内容	添加了 重试行为 。	2023 年 12 月 1 日
新增内容	添加了 SAP 许可 。	2023 年 9 月 22 日
公开发行	首次重新发布《适用于 SAP ABAP 的 AWS SDK 开发者指南》。	2023 年 6 月 30 日
新增内容	新增 适用于 SAP ABAP 的 AWS SDK 功能 。	2023 年 5 月 30 日
新增内容	新增 适用于 SAP ABAP 的 AWS SDK问题排查 。	2023 年 2 月 17 日
开发人员预览	开发者指南的 适用于 SAP ABAP 的 AWS SDK 开发者预览版。	2022 年 11 月 17 日

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。