



參考指南

AWS SDKs 和 工具



AWS SDKs 和 工具: 參考指南

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

AWS SDKs 和 工具參考指南	1
開發人員資源	2
工具組遙測通知	2
組態	4
共用 config 和 credentials 檔案	5
描述檔	5
組態檔案格式	6
登入資料檔案格式	9
共享檔案的位置	9
主目錄解析度	10
變更這些檔案的預設位置	10
環境變數	12
如何設定環境變數	12
無伺服器環境變數設定	13
JVM 系統屬性	13
如何設置 JVM 系統屬性	14
身分驗證與存取	16
AWS 建構家 ID	17
IAM 身分中心身分驗證	18
使用 IAM Identity Center 設定程式設計存取	18
瞭解 IAM 身分中心身分驗證	21
IAM Roles Anywhere	24
步驟 1：隨處設定 IAM 角色	24
步驟 2：隨處使用 IAM 角色	24
擔任角色	25
擔任IAM角色	26
擔任角色 (web)	27
與 Web 身分或 OpenID Connect 聯合	27
AWS 存取金鑰	29
使用短期憑證	29
使用長期認證	29
短期憑證	30
長期登入資料	31
IAM EC2 執行個體的角色	34

建立 IAM 角色	34
啟動 Amazon EC2 執行個體並指定您的IAM角色	35
Connect 至EC2執行個體	35
在執行個EC2體上執行應用程式	35
設定參考	37
建立服務用戶端	37
設定優先順序	37
設定頁面	38
Config 檔案設定清單	39
Credentials 檔案設定清單	43
環境變數清單	43
JVM 系統屬性清單	47
標準化憑證提供者	50
了解憑證提供者鏈	51
SDK特定和工具特定的憑證提供者鏈	52
AWS 存取金鑰	53
擔任角色提供者	55
容器供應商	61
IAM Identity Center 供應商	65
IMDS 供應商	70
流程供應商	74
標準化功能	78
帳戶型端點	78
申請識別碼	81
Amazon EC2 執行個體	83
Amazon S3 存取點	85
Amazon S3 多區域存取點	86
AWS 區域	88
AWS STS 區域端點	91
雙堆疊與端點 FIPS	95
端點探索	97
一般配置	99
IMDS 用戶端	102
重試行為	105
請求壓縮	110
服務特定端點	112

智慧型組態預設	163
一般執行期	168
CRT 相依性	168
維護政策	170
概觀	170
版本控制	170
SDK 主要版本生命週期	170
相依性週期	171
溝通方式	171
版本支援	173
文件歷史紀錄	176
AWS 詞彙表	178
.....	clxxix

AWS SDKs 和 工具參考指南

許多 SDKs 和 工具透過共用的設計規格或透過共用程式庫來共用一些常見功能。

本指南包含下列相關資訊：

- [組態](#) – 如何使用共用 config 和 credentials 檔案或環境變數來設定 AWS SDKs 和 工具。
- [身分驗證與存取](#) – 建立您的程式碼或工具在開發 AWS 時如何與進行身分驗證 AWS 服務。
- [設定參考](#) – 可用於身分驗證和組態的所有標準化設定的參考。
- [AWS 一般執行期 \(CRT \) 程式庫](#) – 幾乎所有 都可使用的共用 AWS 執行期 (CRT) 程式庫概觀 SDKs。
- [AWS SDK 和工具維護政策](#) 涵蓋 軟體開發套件 (SDKs) 和工具的 AWS 維護政策和版本控制，包括行動和物聯網 (IoT) SDKs 及其基礎相依性。

本 AWS SDKs 和 工具參考指南旨在成為適用於多個 SDKs 和 工具的資訊基礎。除了此處顯示的任何資訊之外，還應使用您使用的 SDK 或 工具的特定指南。以下是本指南中具有相關材料區段的 SDK 和 工具：

如果您使用的是：	本指南的相關章節如下：
<ul style="list-style-type: none">• 任何 SDK 或 工具	AWS SDK 和工具維護政策
<ul style="list-style-type: none">• AWS Cloud9• AWS CDK• AWS Toolkit for Azure DevOps• AWS Toolkit for JetBrains• AWS Toolkit for Visual Studio• AWS Toolkit for Visual Studio Code• AWS Serverless Application Model	組態 身分驗證與存取 AWS SDK 和工具維護政策
<ul style="list-style-type: none">• AWS CodeArtifact• AWS CodeBuild• Amazon CodeCatalyst• AWS CodeCommit	

如果您使用的是：	本指南的相關章節如下：
<ul style="list-style-type: none">• AWS CodeDeploy• AWS CodePipeline	
<ul style="list-style-type: none">• AWS CLI• AWS SDK for C++• AWS SDK for Go• AWS SDK for Java• AWS SDK for JavaScript• 適用於 Kotlin 的 AWS SDK• AWS SDK for .NET• AWS SDK for PHP• AWS SDK for Python (Boto3)• AWS SDK for Ruby• 適用於 Rust 的 AWS SDK• 適用於 Swift 的 AWS SDK• AWS Tools for Windows PowerShell	<ul style="list-style-type: none">• 組態• 身分驗證與存取• 設定參考• AWS 一般執行期 (CRT) 程式庫• AWS SDK 和工具維護政策• AWS SDKs和工具版本支援

開發人員資源

如需可協助您開發應用程式之工具的概觀 AWS，請參閱[要建置的工具 AWS](#)。如需支援的相關資訊，請參閱[AWS 知識中心](#)。

Amazon Q Developer 是採用 AI 技術的生成式對話助理，可協助您了解、建置、擴展和操作 AWS 應用程式。為了加速您的建置 AWS，Amazon Q 的模型會擴增為高品質 AWS 內容，以產生更完整、可行且參考的答案。如需詳細資訊，請參閱《[Amazon Q 開發人員使用者指南](#)》中的[什麼是 Amazon Q 開發人員？](#)。

工具組遙測通知

AWS 整合式開發環境 (IDE) Toolkit 是外掛程式和擴充功能，可讓您存取 中的 AWS 服務 IDE。Amazon Q IDE外掛程式和擴充功能可在您的 中啟用生成式 AI 協助IDE。如需每個 IDE Toolkit 的詳細資訊，請參閱上表中的 Toolkit 使用者指南。若要進一步了解如何在 中使用 Amazon QIDE，請參閱《[Amazon Q 開發人員指南](#)》中的[使用 IDE Amazon Q 主題](#)。

AWS IDE Toolkits 和 Amazon Q 可能會收集和存放用戶端遙測資料，以通知未來 AWS Toolkit 和 Amazon Q 版本的相關決策。收集的資料會量化您對 AWS Toolkit 和 Amazon Q 的使用情況。

若要進一步了解所有 AWS IDE Toolkits 和 Amazon Q 所收集的遙測資料，請參閱 `aws-toolkit-common` Github 儲存庫中的 [commonDefinitions.json](#) 文件。

如需每個 AWS IDE Toolkits 和 Amazon Q 擴充功能所收集的遙測資料的詳細資訊，請參閱下列 AWS Toolkit GitHub 儲存庫中的資源文件：

- [AWS 搭配 Amazon Q 的 Visual Studio Toolkit](#)
- [AWS Toolkit for Visual Studio Code VS 程式碼的 和 Amazon Q 延伸模組](#)
- [AWS Toolkit for JetBrains 的 和 Amazon Q 外掛程式 JetBrains](#)
- [Amazon Q for Eclipse](#)

可在 AWS Toolkits 中存取的特定 AWS 服務可能會收集額外的用戶端遙測資料。如需每個個別 AWS 服務所收集的資料類型的詳細資訊，請參閱您感興趣的特定服務 [AWS 的文件](#) 主題。

組態

使用 AWS SDK 和其他 AWS 開發人員工具 (例如 AWS Command Line Interface (AWS CLI))，您可以與 AWS 服務 API 互動。但是，在嘗試此操作之前，您必須使用執行請求操作所需的信息配置 SDK 或工具。

此資訊包括下列項目：

- 識別呼叫 API 的使用者的認證資訊。認證是用來加密對 AWS 伺服器的要求。使用此資訊可 AWS 確認您的身分，並可擷取與其相關聯的權限原則。然後，它可以確定您被允許執行的操作。
- 您用來告知 AWS CLI 或 SDK 如何處理要求、傳送要求的位置 (至哪個 AWS 服務端點)，以及如何解譯或顯示回應的其他組態詳細資料。

每個 SDK 或工具都支援多個來源，您可以使用這些來源提供必要的認證和設定資訊。某些來源對於 SDK 或工具是唯一的，您必須參閱該工具或 SDK 的文件，以瞭解如何使用該方法的詳細資訊。

但是，大多數 AWS SDK 和工具都支持來自兩個主要來源的通用設置 (代碼本身以外)：

- [共用設定 AWS 定和認證檔案](#) — 共用 config 和 credentials 檔案是您可以為 AWS SDK 或工具指定驗證和設定的最常用方式。使用這些檔案來儲存您的工具和應用程式可以使用的設定。共用 config 和 credentials 檔案中的設定與特定設定檔相關聯。使用多個配置文件，您可以創建不同的設置配置以在不同的情況下應用。當您使用 AWS 工具叫用命令或使用 SDK 叫用 AWS API 時，您可以指定要用於該動作的設定檔，以及使用哪些組態設定。其中一個描述檔會指定為設定 default 檔，並且在您未明確指定要使用的設定檔時自動使用。本參考指南說明了您可以儲存在這些檔案中的設定。
- [環境變數](#) — 某些設定也可以儲存在作業系統的環境變數中。雖然您一次只能有一組生效的環境變數，但是當程式執行和需求變更時，它們可以輕鬆地動態修改。

本區段的其他主題

- [共用 config 和 credentials 檔案](#)
- [共用 credentials 檔案 config 和檔案的位置](#)
- [環境變數支援](#)
- [JVM 系統屬性支援](#)

共用 config 和 credentials 檔案

共用 AWS config 和 credentials 檔案包含一組設定檔。設定檔是一組組態設定，以鍵值對表示，由 AWS Command Line Interface (AWS CLI) AWS SDKs、和其他工具使用。組態值會連接到設定檔，以便在使用該設定檔時設定 SDK/工具的某些層面。這些檔案是「共用」，其中的值會影響使用者的任何應用程式、程序或 SDKs 本機環境。

共用 config 和 credentials 檔案都是純文字檔案，僅包含 ASCII 字元 (UTF-8 編碼)。它們採用通常稱為 [INI 檔案](#) 的形式。

描述檔

共用 config 和 credentials 檔案內的設定與特定設定檔相關聯。您可以在 檔案中定義多個設定檔，以建立不同的設定組態，以套用至不同的開發環境。

如果未指定特定具名 [default] 設定檔，設定檔會包含 SDK 或工具操作所使用的值。您也可以建立個別的設定檔，以便依名稱明確參考。每個設定檔都可以根據您的應用程式和案例使用不同的設定和值。

Note

[default] 只是未命名的設定檔。此設定檔會命名，default 因為 SDK 如果使用者未指定設定檔，則這是使用的預設設定檔。它不會將繼承的預設值提供給其他設定檔。如果您在 [default] 設定檔中設定了某些項目，但未在具名設定檔中設定，則當您使用具名設定檔時，不會設定該值。

設定具名設定檔

[default] 設定檔和多個具名設定檔可以存在於同一個檔案中。使用下列設定來選取您的 SDK 或工具在執程式碼時使用的設定檔設定。使用時，也可以在程式碼內或每個命令內選取設定檔 AWS CLI。

透過設定下列其中一項來設定此功能：

AWS_PROFILE - 環境變數

當此環境變數設定為具名設定檔或「預設」時，所有 SDK 程式碼和 AWS CLI 命令都會使用該設定檔中的設定。

透過命令列設定環境變數的 Linux/macOS 範例：

```
export AWS_PROFILE="my_default_profile_name";
```

透過命令列設定環境變數的 Windows 範例：

```
setx AWS_PROFILE "my_default_profile_name"
```

aws.profile - JVM系統屬性

在上的 SDK for Kotlin JVM和SDK適用於 Java 2.x 的 中，您可以[設定aws.profile系統屬性](#)。SDK 建立服務用戶端時，除非在程式碼中覆寫設定，否則會使用具名設定檔中的設定。SDK for Java 1.x 不支援此系統屬性。

Note

如果您的應用程式位於執行多個應用程式的伺服器上，我們建議您一律使用具名設定檔，而非預設設定檔。預設設定檔會自動由環境中的任何 AWS 應用程式擷取，並在其中共用。因此，如果其他人更新其應用程式的預設設定檔，則可能會無意中影響其他人。為了防止這種情況，請在共用config檔案中定義具名設定檔，然後在您的程式碼中設定具名設定檔，以在應用程式中使用該具名設定檔。如果您知道命名描述檔的範圍只會影響您的應用程式，則可以使用環境變數或JVM系統屬性來設定命名描述檔。

組態檔案格式

config 檔案會組織成數個區段。區段是具名的設定集合，且會持續到發生其他區段定義列為止。

config 檔案是使用下列格式的純文字檔案：

- 區段中的所有項目均採用 `setting-name=value` 的一般形式。
- 您可以在列的開頭使用井字號 (`#`)，為列加上註解。

區段類型

區段定義是將名稱套用至設定集合的行。區段定義列以方形括號 (`[`) 開頭和結尾 `]`。在括號內，有區段類型識別符和區段的自訂名稱。您可以使用字母、數字、連字號 (`-`) 和底線 (`_`)，但不能使用空格。

區段類型：default

範例區段定義列：`[default]`

`[default]` 是唯一不需要profile區段識別碼的設定檔。

下列範例顯示具有`[default]`設定檔的基本config檔案。它會設定 [region](#)設定。此行之後直到遇到另一個區段定義為止的所有設定，都是此設定檔的一部分。

```
[default]
#Full line comment, this text is ignored.
region = us-east-2
```

區段類型：profile

範例區段定義列：`[profile dev]`

profile 區段定義列是具名的組態群組，您可以套用至不同的開發案例。若要進一步了解具名設定檔，請參閱設定檔上的上一節。

下列範例顯示具有profile區段定義列和名為 `foo` 之設定檔config的檔案。此行之後直到遇到另一個區段定義為止的所有設定，都是此具名設定檔的一部分。

```
[profile foo]
...settings...
```

有些設定有自己的巢狀子集群組，例如下列範例中s3的設定和子集。透過以一或多個空格縮排子集，將子集與群組建立關聯。

```
[profile test]
region = us-west-2
s3 =
    max_concurrent_requests=10
    max_queue_size=1000
```

區段類型：sso-session

範例區段定義列：`[sso-session my-sso]`

sso-session 區段定義列會命名一組設定，您用來設定設定檔以使用 [解析 AWS 登入資料](#) AWS IAM Identity Center。如需設定單一登入身分驗證的詳細資訊，請參閱 [IAM 您 SDK或 工具的身分中心身](#)

分驗證。描述檔由索引鍵/值對連結到sso-session區段，其中 sso-session 是索引鍵，而 sso-session區段的名稱是 值，例如 sso-session = <name-of-sso-session-section>。

下列範例會設定設定檔，該設定檔會使用來自 "my-ssoSampleRole" 的字符，取得 "111122223333" 帳戶中 "" IAM角色的短期 AWS 憑證。profile 區段中的「my-sso」sso-session區段是使用 sso-session金鑰以名稱參考。

```
[profile dev]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = SampleRole

[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://my-sso-portal.awsapps.com/start
```

區段類型：**services**

範例區段定義列：`[services dev]`

Note

services 本節支援服務特定的端點自訂，且僅適用於 SDKs和包含此功能的工具。若要查看此功能是否適用於您的 SDK，請參閱 [兼容性 AWS SDKs](#) 以取得服務特定的端點。

services 區段定義列會命名一組設定，可設定 AWS 服務 請求的自訂端點。描述檔由索引鍵值對連結到services區段，其中 services 是索引鍵，而services區段的名稱是 值，例如 services = <name-of-services-section>。

services 區段會依<SERVICE> = 行進一步分成子區段，其中 <SERVICE>是 AWS 服務 識別符索引鍵。AWS 服務 識別符是以API模型的 為基礎，serviceId方法是將所有空格替換為底線，並縮小所有字母大小寫。如需要在 services 區段中使用的所有服務識別碼金鑰的清單，請參閱 [服務特定端點的識別碼](#)。服務識別碼金鑰後面接續巢狀化設定，每個設定獨佔一列並縮排兩個空格。

下列範例使用 services定義來設定端點，以用於僅對 服務提出的 Amazon DynamoDB 請求。在 "local-dynamodb" services 區段中，會使用 services金鑰，依名稱參考 profile區段。AWS 服務 識別符索引鍵為 dynamodb。Amazon DynamoDB 服務子區段從行 開始dynamodb = 。任何緊接著縮排的列都包含在該子區段中，並適用於該服務。

當 SDK 或工具執行時，它會檢查這些檔案並載入任何可用的組態設定。如果檔案尚未存在，SDK 或工具會自動建立基本檔案。

依預設，檔案位於名為的資料夾中，.aws該資料夾位於您home或使用者資料夾中。

作業系統	預設位置和檔案名稱
Linux 和 macOS	~/.aws/config ~/.aws/credentials
Windows	%USERPROFILE%\aws\config %USERPROFILE%\aws\credentials

主目錄解析度

~僅在以下情況下用於主目錄解析：

- 開始路徑
- 後面是立即/或平台特定的分隔符號。在窗口上，~/並且~\都解析到主目錄。

決定主目錄時，會檢查下列變數：

- (所有平台) HOME 環境變數
- (視窗平台) USERPROFILE 環境變數
- (視窗平台) 連接HOMEDRIVE和HOMEPATH環境變數 (\$HOMEDRIVE\$HOMEPATH)
- (每個 SDK 或工具可選) SDK 或工具專用的主路徑解析功能或變量

如果可能的話，如果在路徑的開頭指定了使用者的主目錄 (例如，~username/)，就會解析為要求的使用者名稱的主目錄 (例如，/home/username/.aws/config)。

變更這些檔案的預設位置

您可以使用下列任何一項來覆寫 SDK 或工具載入這些檔案的位置。

使用環境變數

您可以設定下列環境變數，將這些檔案的位置或名稱從預設值變更為自訂值：

- config文件環境變量：**AWS_CONFIG_FILE**
- credentials文件環境變量：**AWS_SHARED_CREDENTIALS_FILE**

Linux/macOS

您可以在 Linux 或 macOS 上執行下列[匯出](#)指令來指定替代位置。

```
$ export AWS_CONFIG_FILE=/some/file/path/on/the/system/config-file-name
$ export AWS_SHARED_CREDENTIALS_FILE=/some/other/file/path/on/the/system/credentials-file-name
```

Windows

您可以透過在 Windows 上執行下列[setx](#)指令來指定替代位置。

```
C:\> setx AWS_CONFIG_FILE c:\some\file\path\on\the\system\config-file-name
C:\> setx AWS_SHARED_CREDENTIALS_FILE c:\some\other\file\path\on\the\system\credentials-file-name
```

如需使用環境變數配置系統的詳細資訊，請參閱[環境變數支援](#)。

使用 JVM 系統屬性

對於在 JVM 上執行的 Kotlin 適用的 SDK 以及適用於 Java 2.x 的 SDK，您可以設定下列 JVM 系統屬性，將這些檔案的位置或名稱從預設值變更為自訂值：

- config文件 JVM 系統屬性：**aws.configFile**
- credentials文件環境變量：**aws.sharedCredentialsFile**

如需有關如何設定 JVM 系統屬性的指示，請參閱[the section called “如何設置 JVM 系統屬性”](#)。適用於 Java 1.x 的開發套件不支援這些系統屬性。

環境變數支援

環境變數提供另一種方法來指定組態選項和憑證，且適合用來編寫指令碼或暫時將具名描述檔設為預設值。如需大多數 支援的環境變數清單 SDKs，請參閱 [環境變數清單](#)。

選項的優先順序

- 如果您使用環境變數指定設定，則會覆寫從共用 AWS config 和 credentials 檔案中的設定檔載入的任何值。
- 如果您在 AWS CLI 命令列上使用 參數來指定設定，它會覆寫來自對應環境變數或組態檔案中設定檔的任何值。

如何設定環境變數

下列範例說明如何為預設使用者設定環境變數。

Linux, macOS, or Unix

```
$ export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
$ export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
$ export
  AWS_SESSION_TOKEN=AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJ0gQs8IZZaIv2BXIa2R40lgk
$ export AWS_REGION=us-west-2
```

設定環境變數會變更使用的數值，直到 Shell 工作階段結束或直到您將該變數設為其他數值。您可以在 Shell 的啟動指令碼中設定變數，讓它們跨未來的工作階段持續生效。

Windows Command Prompt

```
C:\> setx AWS_ACCESS_KEY_ID AKIAIOSFODNN7EXAMPLE
C:\> setx AWS_SECRET_ACCESS_KEY wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
C:\> setx
  AWS_SESSION_TOKEN AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJ0gQs8IZZaIv2BXIa2R40lgk
C:\> setx AWS_REGION us-west-2
```

使用 `set` 來設定環境變數會變更使用的值，直到目前的命令提示詞工作階段結束，或直到您將變數設定為不同的值為止。使用 `setx` 設定環境變數，會變更目前命令提示工作階段和您在執行命令後建立的所有命令提示工作階段中使用的值。不會影響您執行命令當時已執行的其他命令 Shell。

PowerShell

```
PS C:\> $Env:AWS_ACCESS_KEY_ID="AKIAIOSFODNN7EXAMPLE"
PS C:\> $Env:AWS_SECRET_ACCESS_KEY="wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"
PS C:\>
PS C:\> $Env:AWS_SESSION_TOKEN="AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJ0gQs8IZZaIv2BXIa2R40L"
PS C:\> $Env:AWS_REGION="us-west-2"
```

如果您在 PowerShell 提示中設定環境變數，如先前範例所示，它只會在目前工作階段的持續時間內儲存該值。若要讓環境變數設定在所有 PowerShell 和 命令提示工作階段中持續存在，請使用控制面板中的系統應用程式來存放它。或者，您可以將變數新增至您的 PowerShell 設定檔，以設定所有未來 PowerShell 工作階段的變數。如需儲存環境變數或跨工作階段保留它們的詳細資訊，請參閱 [PowerShell 文件](#)。

無伺服器環境變數設定

如果您使用無伺服器架構進行開發，您還有其他設定環境變數的選項。根據您的容器，您可以針對在這些容器中執行的程式碼使用不同的策略，以查看和存取環境變數，類似於非雲端環境。

例如，使用 AWS Lambda，您可以直接設定環境變數。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 [使用 AWS Lambda 環境變數](#)。

在無伺服器架構中，您通常可以在 SDK 環境設定下的提供者金鑰下，在 `serverless.yml` 檔案中設定環境變數。如需 `serverless.yml` 檔案的資訊，請參閱 Serverless Framework 文件中的 [一般函數設定](#)。

無論您使用哪種機制來設定容器環境變數，容器都會保留一些，例如在 [定義的執行期環境變數](#) 中記錄 Lambda 的機制。請務必參閱您所使用的容器官方文件，以判斷環境變數的處理方式，以及是否有任何限制。

JVM 系統屬性支援

[JVM 系統屬性](#) 提供了另一種方法，可為在 JVM 上執行的 SDK (例如和) 指定組態選項 AWS SDK for Java 和認證。適用於 Kotlin 的 AWS SDK 如需 SDK 支援的 JVM 系統屬性清單，請參閱 [設定參考](#)。

選項的優先順序

- 如果您使用其 JVM 系統屬性指定設定，它會覆寫在環境變數中找到的任何值，或從共用 AWS `config` 和 `credentials` 檔案中的設定檔載入的任何值。

- 如果您使用其環境變數指定設定，它會覆寫從共用 AWS config 和 credentials 檔案中的設定檔載入的任何值。

如何設置 JVM 系統屬性

您可以多種方式設定 JVM 系統屬性。

在命令行上

使用參數叫用命令時，在 java 命令列上設定 JVM 系統屬性。-D 除非您明確覆寫程式碼中的值，否則下列命令會針對所有服務用戶端進行 AWS 區域 全域設定。

```
java -Daws.region=us-east-1 -jar <your_application.jar> <other_arguments>
```

如果您需要設定多個 JVM 系統屬性，請多次指定 -D 參數。

使用環境變數

如果您無法存取命令列來叫用 JVM 來執行應用程式，您可以使用 JAVA_TOOL_OPTIONS 環境變數來設定命令列選項。此方法在 Java 執行階段上執行 AWS Lambda 函式或在嵌入式 JVM 中執行程式碼等情況下非常有用。

下列範例會針對所有服務用戶端進行 AWS 區域 全域設定，除非您明確覆寫程式碼中的值。

Linux, macOS, or Unix

```
$ export JAVA_TOOL_OPTIONS="-Daws.region=us-east-1"
```

設定環境變數會變更使用的數值，直到 Shell 工作階段結束或直到您將該變數設為其他數值。您可以在 Shell 的啟動指令碼中設定變數，讓它們跨未來的工作階段持續生效。

Windows Command Prompt

```
C:\> setx JAVA_TOOL_OPTIONS -Daws.region=us-east-1
```

用 [set](#) 來設定環境變數會變更使用的值，直到目前的「指令提示字元」階段作業結束，或將變數設定為不同的值為止。用 [setx](#) 來設定環境變數，會變更目前「命令提示字元」階段作業和您在執行指令之後建立的所有「命令提示字元」階段作業中使用的值。不會影響您執行命令當時已執行的其他命令 Shell。

在運行時

您也可以通過使用如下面的例子中的 `System.setProperty` 方法在運行時在代碼中設置 JVM 系統屬性。

```
System.setProperty("aws.region", "us-east-1");
```

Important

在初始化 SDK 服務用戶端之前，請先設定任何 JVM 系統屬性，否則服務用戶端可能會使用其他值。

身分驗證與存取

當您使用開發 AWS 時，您必須建立程式碼與的身分驗證方式 AWS 服務。您可以根據環境和您可用的存取權，以不同的方式設定 AWS 資源的程式設計 AWS 存取。

在本機執行之程式碼的身分驗證選項（不在中 AWS）

- [IAM 您 SDK 或 工具的身分中心身分驗證](#) – 作為安全最佳實務，我們建議您 AWS Organizations 搭配 IAM Identity Center 使用 來管理所有的存取 AWS 帳戶。您可以在 中建立使用者 AWS IAM Identity Center、使用 Microsoft Active Directory、使用 SAML 2.0 身分提供者 IdP)，或個別聯合您的 IdP 至 AWS 帳戶。若要檢查您的區域是否支援 IAM Identity Center，請參閱 中的 [AWS IAM Identity Center 端點和配額](#) Amazon Web Services 一般參考。
- [IAM Roles Anywhere](#) – 您可以使用 IAM Roles Anywhere 在 中取得臨時安全憑證，IAM 用於在 之外執行的伺服器、容器和應用程式等工作負載 AWS。若要使用 IAM Roles Anywhere，您的工作負載必須使用 X.509 憑證。
- [使用 AWS 憑證擔任角色](#) – 您可以擔任 IAM 角色，暫時存取您可能無法存取 AWS 的資源。
- [AWS 存取金鑰](#) – 可能較不方便或可能增加 AWS 資源安全風險的其他選項。

在 AWS 環境中執行之程式碼的身分驗證選項

如果您的程式碼在 上執行 AWS，憑證可以自動提供給應用程式。例如，如果您的應用程式託管在 Amazon Elastic Compute Cloud 上，並且有一個與該資源相關聯的 IAM 角色，則登入資料會自動提供給應用程式。同樣地，如果您使用 Amazon ECS 或 Amazon EKS 容器，則透過 的憑證提供者鏈，在容器內執行的程式碼 IAM 可以自動取得角色的 SDK 憑證集。

- [使用 Amazon EC2 執行個體的 IAM 角色](#) – 使用 IAM 角色在 Amazon EC2 執行個體上安全地執行應用程式。
- 您可以使用 AWS IAM Identity Center 以程式設計方式與 互動，方法如下：
 - [AWS CloudShell](#) 使用 從主控台執行 AWS CLI 命令。
 - 若要為軟體開發團隊嘗試雲端協作空間，請考慮使用 [Amazon CodeCatalyst](#)。

透過 Web 型身分提供者進行身分驗證 - 行動或用戶端型 Web 應用程式

如果您要建立需要存取 的行動應用程式或用戶端型 Web 應用程式 AWS，請建置您的應用程式，以便使用 Web 身分聯合動態請求臨時 AWS 安全憑證。

有了 Web 聯合身分，您就不需要建立自訂登入代碼，或管理您自己的使用者身分。反之，應用程式使用者可以使用知名的外部身分提供者 (IdP) 登入，例如使用 Amazon、Facebook、Google 或任何其他 OpenID Connect (OIDC) 相容 IdP 登入。他們可以接收身分驗證權杖，然後交換 AWS 該權杖，以取得該權杖中的臨時安全憑證，以映射至 IAM 具有許可的角色，以使用您中的資源 AWS 帳戶。

若要了解如何為您的 SDK 或 工具設定此項目，請參閱 [擔任具有 Web 身分或 OpenID Connect 的角色](#)。

對於行動應用程式，請考慮使用 Amazon Cognito。Amazon Cognito 擔任身分代理程式，並為您執行大部分聯合工作。如需詳細資訊，請參閱 IAM 使用者指南 中的 [將 Amazon Cognito 用於行動應用程式](#)。

有關存取管理的詳細資訊

IAM 使用者指南包含下列有關安全控制 AWS 資源存取的資訊：

- [IAM 身分 \(使用者、使用者群組和角色\)](#) – 了解身分的基本概念 AWS。
- [中的安全最佳實務 IAM](#) – 根據 [共同責任模型](#) 開發 AWS 應用程式時應遵循的安全建議。

Amazon Web Services 一般參考 具有下列項目的基礎基本知識：

- [了解並取得您的 AWS 憑證](#) – 存取主控台和程式設計存取的金鑰選項和管理實務。

AWS 建構家 ID

您的 AWS 建構家 ID 補充 AWS 帳戶 您可能已經擁有或想要建立的任何。雖然 AWS 帳戶 充當您建立 AWS 之資源的容器，並為這些資源提供安全界限，但您的 AWS 建構家 ID 代表您作為個人。您可以使用 登入 AWS 建構家 ID，以存取開發人員工具和服務，例如 Amazon CodeWhisperer 和 Amazon CodeCatalyst。

- AWS 登入 使用 使用者指南中的 [登入 AWS 建構家 ID](#)：了解如何建立和使用 AWS 建構家 ID，並了解 Builder ID 提供的內容。
- CodeWhisperer 使用 使用者指南中的 [CodeWhisperer 和 AWS 工具組 - Builder ID 進行驗證](#) – 了解如何 CodeWhisperer 使用 AWS 建構家 ID。
- [CodeCatalyst 概念 - AWS 建構家 ID](#) 在 Amazon CodeCatalyst 使用者指南 – 了解如何 CodeCatalyst 使用 AWS 建構家 ID。

IAM 您 SDK或 工具的身分中心身分驗證

AWS IAM Identity Center 是在非AWS 運算服務上開發 時提供 AWS 登入資料的建議方法。例如，這就像您的本機開發環境一樣。如果您是在 AWS 資源上進行開發，例如 Amazon Elastic Compute Cloud (Amazon EC2) 或 AWS Cloud9，我們建議您改為從該服務取得憑證。

在本教學課程中，您會建立 IAM Identity Center 存取，並使用 AWS 存取入口網站和 來設定您的 SDK 或工具 AWS CLI。

- AWS 存取入口網站是您手動登入 IAM Identity Center 的 Web 位置。的格式URL為 `d-xxxxxxxxxx.awsapps.com/start` 或 `your_subdomain.awsapps.com/start`。登入 AWS 存取入口網站時，您可以檢視已為該使用者設定 AWS 帳戶 的角色。此程序使用 AWS 存取入口網站 來取得 SDK/tool 身分驗證程序所需的組態值。
- AWS CLI 用於設定 SDK或 工具，以使用 IAM Identity Center 身分驗證來API呼叫您的程式碼。此一次性程序會更新您的共用 AWS config檔案，然後您的 SDK或 工具會在您執行程式碼時使用。

使用 IAM Identity Center 設定程式設計存取

步驟 1：建立存取權並選擇適當的許可集

如果您尚未啟用 IAM Identity Center，請參閱 AWS IAM Identity Center 使用者指南中的[啟用 IAM Identity Center](#)。

選擇下列其中一種方法來存取您的 AWS 登入資料。

我沒有透過 IAM Identity Center 建立存取權

1. 遵循 AWS IAM Identity Center 使用者指南中的[使用預設 IAM Identity Center 目錄程序設定使用者存取權](#)，以新增使用者和新增管理許可。
2. AdministratorAccess 許可集不應用於定期開發。反之，建議您使用預先定義的PowerUserAccess許可集，除非您的雇主已為此目的建立自訂許可集。

再次遵循相同的[設定使用者存取與預設 IAM Identity Center 目錄](#)程序，但這次：

- 與其建立 *Admin team* 群組，請建立 *Dev team*群組，並在之後的說明中取代此群組。
- 您可以使用現有的使用者，但必須將使用者新增至新*Dev team*群組。
- 與其建立*AdministratorAccess*許可集，請建立*PowerUserAccess*許可集，並在之後的說明中取代此集。

完成後，您應該具備下列項目：

- Dev team 群組。
 - 附加PowerUserAccess許可集至 Dev team群組。
 - 您的使用者已新增至 Dev team群組。
3. 退出口網站並再次登入，以查看 Administrator 的 AWS 帳戶 和 選項PowerUserAccess。使用工具/ PowerUserAccess時，請選取 SDK。

我已經 AWS 透過由我的雇主管理的聯合身分提供者（例如 Microsoft Entra 或 Okta）存取

AWS 透過您的身分提供者入口網站登入。如果您的雲端管理員已授予您 PowerUserAccess（開發人員）許可，您會看到您有權存取 AWS 帳戶 的 和許可集。您會在許可集名稱旁，看到使用該許可集手動或以程式設計方式存取帳戶的選項。

若您自訂實作，可能會產生不同體驗，例如不同的許可集名稱。若您不確定要使用哪個許可集，請聯絡您的 IT 團隊尋求協助。

我已經能夠 AWS 透過由我的雇主管理的 AWS 存取入口網站存取

透過 AWS 存取入口網站登入。若雲端管理員已授予您 PowerUserAccess（開發人員）權限，您會看到您有權存取的 AWS 帳戶 和許可集。您會在許可集名稱旁，看到使用該許可集手動或以程式設計方式存取帳戶的選項。

我已經能夠 AWS 透過由我的雇主管理的聯合身分提供者存取

請聯絡您的 IT 團隊尋求協助。

步驟 2：設定 SDKs 和 工具以使用 IAM Identity Center

1. 在開發機器上安裝最新的 AWS CLI。
 - a. 請參閱AWS Command Line Interface 《使用者指南》中的[安裝或更新最新版本的 AWS CLI](#)。
 - b. （選用）若要驗證 AWS CLI 是否正常運作，請開啟命令提示字元並執行aws --version命令。
2. 登入 AWS 存取入口網站。您的雇主可能會提供此資訊，URL或者您可能會在電子郵件中取得，步驟 1：建立存取權。如果沒有，請在儀表板上尋找您的AWS 存取入口網站URL<https://console.aws.amazon.com/singlesignon/>。

- a. 在 AWS 存取入口網站的帳戶索引標籤中，選取要管理的個別帳戶。會顯示您使用者的角色。選擇存取金鑰以取得命令列的登入資料，或取得適當許可集的程式設計存取。使用預先定義的PowerUserAccess許可集，或您或您的雇主建立以套用開發最低權限許可集的任何許可集。
 - b. 在取得憑證對話方塊中，選擇MacOS 和 Linux 或 Windows，具體取決於您的作業系統。
 - c. 選擇 IAM Identity Center 登入資料方法，以取得下一個步驟所需的 Issuer URL和 SSO Region值。注意：SSO Start URL可與 互換使用Issuer URL。
3. 在 AWS CLI 命令提示中，執行 `aws configure sso` 命令。出現提示時，輸入您在上一個步驟中收集的組態值。如需此 AWS CLI 命令的詳細資訊，請參閱[使用aws configure sso精靈設定設定檔](#)。
- a. 針對提示 SSO Start URL，輸入您為 取得的值Issuer URL。
 - b. 對於CLI設定檔名稱，我們建議您在開始`default`時輸入。如需如何設定非預設（具名）設定檔及其相關聯環境變數的資訊，請參閱 [描述檔](#)。
4. （選用）在 AWS CLI 命令提示中，執行 `aws sts get-caller-identity` 命令以確認作用中的工作階段身分。回應應會顯示您設定的 IAM Identity Center 許可集。
5. 如果您使用的是 AWS SDK，SDK請在開發環境中為 建立應用程式。
- a. 對於某些 SDKs，SSOIDC您必須先將 SSO和 等其他套件新增至您的應用程式，才能使用 IAM Identity Center 身分驗證。如需詳細資訊，請參閱您的特定 SDK。
 - b. 如果您先前已設定的存取權 AWS，請檢閱共用 AWS credentials檔案是否有任何 [AWS 存取金鑰](#)。您必須移除任何靜態登入資料，SDK或 工具才會因[了解憑證提供者鏈](#)優先順序而使用 IAM Identity Center 登入資料。

如需深入了解 SDKs和 工具如何使用和重新整理使用此組態的登入資料，請參閱 [瞭解 IAM 身分中心身分驗證](#)。

視您設定的工作階段長度而定，您的存取最終會過期，而 SDK或 工具將遇到身分驗證錯誤。若要在需要時再次重新整理存取入口網站工作階段，請使用 AWS CLI 來執行 `aws sso login` 命令。

您可以同時延長 IAM Identity Center 存取入口網站工作階段持續時間和許可集工作階段持續時間。這可延長您可以執行程式碼的時間，之後您才需要再次使用 手動登入 AWS CLI。如需詳細資訊，請參閱《AWS IAM Identity Center 使用者指南》中的以下主題：

- IAM Identity Center 工作階段持續時間 – [設定使用者 AWS 存取入口網站工作階段的持續時間](#)
- 許可集工作階段持續時間 – [設定工作階段持續時間](#)

如需 SDKs 和 工具的所有 IAM Identity Center 提供者設定的詳細資訊，請參閱本指南 [IAM Identity Center 登入資料提供者](#) 中的。

瞭解 IAM 身分中心身分驗證

IAM 身分中心的相關條款

以下術語可幫助您了解背後的過程和配置 AWS IAM Identity Center。針對部分驗證概念，AWS SDK API 的說明文件會使用與 IAM 身分中心不同的名稱。知道這兩個名稱很有幫助。

下表顯示替代名稱彼此之間的關係。

IAM 身分識別中心名稱	開發套件 API 名稱	描述
識別中心	sso	雖然「AWS 單一登入」已重新命名，但 sso API 命名空間仍會保留其原始名稱，以提供回溯相容性。如需詳細資訊，請參閱 AWS IAM Identity Center 使用指南中的 IAM 身分中心重新命名 。
IAM 身分中心主控台 管理主控台		用來設定單一登入的主控台。
AWS 存取入口網站 URL		IAM 身分中心帳戶專屬的 URL，例如 <code>https://xxx.awsapps.com/start</code> 。您可以使用 IAM 身分中心登入認證登入此入口網站。
IAM 身分中心存取入口網站會話	驗證會話	提供承載訪問令牌給調用者。
權限集工作階段		SDK 在內部用來 AWS 服務撥打呼叫的 IAM 工作階段。在

IAM 身分識別中心名稱	開發套件 API 名稱	描述
權限集認證	AWS 登入資料 簽署 4 認證	非正式討論中，您可能會看到這個錯誤地稱為「角色工作階段」。 SDK 實際用於大多數AWS 服務呼叫的憑據（特別是所有 sigv4 AWS 服務 調用）。在非正式討論中，您可能會看到錯誤地稱為「角色認證」。
IAM 身分中心憑證提供者	SSO 認證提供者	如何取得認證，例如提供功能的類別或模組。

瞭解的 SDK 認證解決方案 AWS 服務

IAM 身分中心 API 會將承載權杖登入資料交換為 sigv4 登入資料。大多數AWS 服務是 sigv4 API，有一些例外情況，如Amazon CodeWhisperer和Amazon CodeCatalyst。以下說明透過支援大部分應用程式程式碼AWS 服務呼叫的認證解析程序AWS IAM Identity Center。

啟動 AWS 存取入口網站工作階段

- 透過使用您的認證登入工作階段來啟動程序。
 - 使用 AWS Command Line Interface (AWS CLI) 中的 `aws sso login` 指令。如果您還沒有使用中的工作階段，這會啟動新的 IAM 身分中心工作階段。
- 當您開始新的工作階段時，您會收到來自 IAM 身分中心的重新整理權杖和存取權杖。AWS CLI 還使用新的訪問令牌和刷新令牌更新 SSO 緩存 JSON 文件，並使其可供 SDK 使用。
- 如果您已經有作用中的工作階段，AWS CLI 命令會重複使用現有的工作階段，並在現有工作階段到期時過期。若要瞭解如何設定 IAM 身分中心工作階段的長度，請參閱 [AWS IAM Identity Center 使用者指南中的設定使用者AWS存取入口網站工作階段的持續時間](#)。
 - 工作階段長度上限已延長至 90 天，以減少頻繁登入的需求。

SDK 如何取得AWS 服務呼叫憑證

SDK 可在每個服務實例化客戶端對象AWS 服務時提供訪問權限。當選取的共用AWSconfig檔案設定為 IAM 身分中心登入資料解析時，IAM 身分中心就會用來解析應用程式的登入資料。

- 建立用戶端時，[認證解析程序](#)會在執行階段完成。

若要使用 IAM 身分中心單一登入擷取 sigv4 API 的登入資料，SDK 會使用 IAM 身分中心存取權杖取得 IAM 工作階段。此 IAM 工作階段稱為權限集工作階段，透過假設 IAM 角色來提供對 SDK 的 AWS 存取權。

- 權限集工作階段持續時間是獨立於 IAM 身分中心工作階段持續時間設定。
 - 若要瞭解如何設定權限集工作階段持續時間，請參閱 AWS IAM Identity Center 使用者指南中的 [設定工作階段持續時間](#)。
- 請注意，在大多數 AWS SDK API 文件中，權限集 AWS 認證也稱為認證和 sigv4 認證。

權限集登入資料會從呼叫 IAM 身分中心 API 傳回至 SDK。[getRoleCredentials](#) SDK 的用戶端物件會使用假設 IAM 角色來呼叫 AWS 服務，例如要求 Amazon S3 列出您帳戶中的儲存貯體。用戶端物件可以繼續使用這些權限集認證操作，直到權限集工作階段到期為止。

工作階段到期和重新

使用時 [SSO 權杖提供者組態](#)，會使用重新整理權杖自動重新整理從 IAM 身分中心取得的每小時存取權杖。

- 如果 SDK 嘗試使用它時訪問令牌已過期，則 SDK 將使用刷新令牌嘗試獲取新的訪問令牌。IAM 身分中心會比較重新整理權杖與您的 IAM 身分中心存取入口網站工作階段持續時間。如果重新整理權杖尚未過期，IAM 身分中心會以另一個存取權杖進行回應。
- 此存取權杖可用來重新整理現有用戶端的權限集工作階段，或解析新用戶端的認證。

不過，如果 IAM 身分中心存取入口網站工作階段已過期，則不會授與新的存取權杖。因此，無法更新權限集持續時間。只要現有用戶端的快取權限設定工作階段長度逾時，它就會過期 (而且存取將會遺失)。

任何建立新用戶端的程式碼都會在 IAM 身分中心工作階段到期後立即失敗驗證。這是因為不會快取權限集認證。在您擁有有效的訪問令牌之前，您的代碼將無法創建新的客戶端並完成憑據解析過程。

重述一下，當 SDK 需要新的權限集認證時，SDK 會先檢查是否有任何有效的現有認證，並使用這些認證。無論認證是用於新用戶端還是具有過期認證的現有用戶端，都適用此選項。如果找不到認證或無效，則 SDK 會呼叫 IAM 身分中心 API 以取得新的登入資料。要調用 API，它需要訪問令牌。如果存取權杖已過期，SDK 會使用重新整理權杖嘗試從 IAM 身分中心服務取得新的存取權杖。如果您的 IAM 身分中心存取入口網站工作階段尚未過期，則會授與此權杖。

IAM Roles Anywhere

您可以使用 IAM 角色隨處取得 IAM 中的臨時安全登入資料，適用於在外部執行的伺服器、容器和應用程式等工作負載AWS。若要在任何地方使用 IAM 角色，您的工作負載必須使用 X.509 憑證。您的雲端管理員應該提供憑證和私密金鑰，以便將 IAM 角色隨處設定為您的登入資料提供者。

步驟 1：隨處設定 IAM 角色

IAM 角色隨處提供一種取得在以外執行之工作負載或程序的臨時登入資料的方法AWS。信任錨點會與憑證授權單位建立，以取得關聯 IAM 角色的臨時登入資料。此角色可設定程式碼使用 IAM 角色隨處進行身份驗證時，工作負載將具有的許可。

如需設定信任錨點、IAM 角色和 IAM 角色隨處設定檔的步驟，請參閱 IAM 角AWS Identity and Access Management色隨處使用者指南中的[隨處建立信任錨點和設定檔](#)。

Note

IAM 角色隨處使用者指南中的設定檔指的是 IAM 角色任何地方服務中的獨特概念。它與共享 AWSconfig文件中的配置文件無關。

步驟 2：隨處使用 IAM 角色

若要從 IAM 角色隨處取得臨時安全登入資料，請使用 IAM 角色隨處提供的登入資料協助工具。憑證工具實作 IAM 角色在任何地方的簽署程序。

如需下載憑證協助程式工具的指示，請參閱《IAM 角色隨處使用者指南》中的「從任何AWS Identity and Access Management角色[取得臨時安全登入資料](#)」。

若要透過 AWS SDK 使用 IAM 角色隨處提供的臨時安全登入資料AWS CLI，您可以在共用AWSconfig檔案中進行credential_process設定。SDK 和AWS CLI支援用credential_process於驗證的處理程序憑證提供者。以下顯示要設定的一般結構credential_process。

```
credential_process = [path to helper tool] [command] [--parameter1 value] [--parameter2 value] [...]
```

輔助工具的credential-process命令會以與credential_process設定相容的標準 JSON 格式傳回臨時認證。請注意，指令名稱包含連字號，但設定名稱包含底線。該命令需要以下參數：

- `private-key`— 簽署要求之私密金鑰的路徑。
- `certificate`— 憑證的路徑。
- `role-arn`— 要取得暫時認證之角色的 ARN。
- `profile-arn`— 設定檔的 ARN，可為指定角色提供對應。
- `trust-anchor-arn`— 用來驗證之信任錨點的 ARN。

您的雲端管理員應該提供憑證和私密金鑰。所有三個 ARN 值都可以從複製。AWS Management Console 下列範例顯示一個共用 config 檔案，該檔案會設定從輔助工具擷取暫時認證。

```
[profile dev]
credential_process = ./aws_signing_helper credential-process --certificate /
path/to/certificate --private-key /path/to/private-key --trust-anchor-
arn arn:aws:rolesanywhere:region:account:trust-anchor/TA_ID --profile-
arn arn:aws:rolesanywhere:region:account:profile/PROFILE_ID --role-
arn arn:aws:iam::account:role/ROLE_ID
```

如需選用參數和其他協助工具的詳細資訊，請參閱上 GitHub 的 [IAM 角色隨處登入資料助手](#)

有關 SDK 配置設置本身和進程憑證提供者的詳細信息，請參閱本指南 [程序認證提供者](#) 中的。

使用 AWS 憑證擔任角色

假設角色涉及使用一組臨時安全登入資料來存取 AWS 您可能在其他情況下無法存取的資源。這些臨時登入資料由存取金鑰 ID、私密存取金鑰和安全字符組成。若要進一步了解 AWS Security Token Service (AWS STS) API 請求，請參閱 [AWS Security Token Service API 參考](#) 中的 [動作](#)。

若要設定 SDK 或 工具以擔任角色，您必須先建立或識別要擔任的特定角色。IAM 角色由角色 Amazon Resource Name () 唯一識別 [ARN](#)。角色會與其他實體建立信任關係。使用 角色的受信任實體可能是 AWS 服務 或其他 AWS 帳戶。若要進一步了解 IAM 角色，請參閱 [IAM 使用者指南](#) 中的 [使用 IAM 角色](#)。

識別 IAM 角色後，如果您受該角色信任，您可以設定 SDK 或 工具以使用角色授予的許可。

Note

AWS 最佳實務是盡可能使用區域端點並設定您的 [AWS 區域](#)。

擔任IAM角色

擔任角色時，會 AWS STS 傳回一組臨時安全憑證。這些憑證來自另一個設定檔，或來自程式碼正在執行的執行個體或容器。當您擁有一個帳戶的 AWS 憑證，但您的應用程式需要存取另一個帳戶中的資源時，通常會使用這種類型的擔任角色。

步驟 1：設定IAM角色

若要設定您的 SDK或 工具以擔任角色，您必須先建立或識別要擔任的特定角色。IAM 使用角色唯一識別角色 [ARN](#)。角色與另一個實體建立信任關係，通常在您的帳戶內或用於跨帳戶存取。若要設定，請參閱 IAM 使用者指南 中的 [建立IAM角色](#)。

步驟 2：設定 SDK或 工具

設定 SDK或 工具，從 `credential_source`或 取得憑證 `source_profile`。

使用 從 Amazon ECS容器、Amazon EC2執行個體或環境變數 `credential_source`取得憑證。

使用 `source_profile`來從另一個設定檔取得憑證。`source_profile`也支援角色鏈結，這是設定檔的階層，接著會使用擔任的角色來擔任另一個角色。

當您在設定檔中指定此項目時，SDK或 工具會自動為您進行對應的 AWS STS [AssumeRole](#) API 呼叫。若要透過擔任角色來擷取和使用臨時憑證，請在共用 AWS config檔案中指定下列組態值。如需每個設定的詳細資訊，請參閱 [擔任角色登入資料提供者設定](#) 一節。

- `role_arn` - 來自您在步驟 1 中建立IAM的角色
- 設定 `source_profile`或 `credential_source`
- (選用) `duration_seconds`
- (選用) `external_id`
- (選用) `mfa_serial`
- (選用) `role_session_name`

下列範例顯示共用config檔案中兩個擔任角色選項的組態：

```
role_arn = arn:aws:iam::123456789012:role/my-role-name
source_profile = profile-name-with-user-that-can-assume-role
```

```
role_arn = arn:aws:iam::123456789012:role/my-role-name
```

```
credential_source = Ec2InstanceMetadata
```

如需所有擔任角色憑證提供者設定的詳細資訊，請參閱本指南[擔任角色登入資料提供者](#)中的。

擔任具有 Web 身分或 OpenID Connect 的角色

假設角色涉及使用一組臨時安全登入資料來存取 AWS 您可能無法存取的資源。這些臨時登入資料由存取金鑰 ID、私密存取金鑰和安全字符組成。若要進一步了解 AWS Security Token Service (AWS STS) API 請求，請參閱 AWS Security Token Service API 參考中的[動作](#)。

若要設定您的 SDK 或 工具以擔任角色，您必須先建立或識別要擔任的特定角色。IAM 角色由角色 Amazon Resource Name ([ARN](#)) 唯一識別。角色會與其他實體建立信任關係。使用 角色的信任實體可能是 Web 身分提供者或 OpenID Connect (OIDC) SAML 或聯合。若要進一步了解 IAM 角色，請參閱 IAM 使用者指南中的[擔任角色的方法](#)。

在 中設定 IAM 角色後 SDK，如果該角色設定為信任您的身分提供者，您可以進一步設定 SDK 以擔任該角色，以取得臨時 AWS 登入資料。

Note

AWS 最佳實務是盡可能使用區域端點並設定您的 [AWS 區域](#)。

與 Web 身分或 OpenID Connect 聯合

您可以使用來自公開身分提供者的 JSON Web 權杖 (JWTs)，例如 Login With Amazon、Facebook、Google，以使用 取得臨時 AWS 登入資料 `AssumeRoleWithWebIdentity`。視其使用方式而定，這些 JWTs 可能稱為 ID 字符或存取字符。您也可以使用從與 的探索通訊協定相容的身分提供者 (IdPs) JWTs 發行 OIDC 的，例如 EntraId 或 PingFederate。

如果您使用 Amazon Elastic Kubernetes Service，此功能可讓您為 Amazon EKS 叢集中的每個服務帳戶指定不同的 IAM 角色。此 Kubernetes 功能會分佈 JWTs 到您的 Pod，然後由此登入資料提供者用來取得臨時 AWS 登入資料。如需此 Amazon EKS 組態的詳細資訊，請參閱《Amazon EKS 使用者指南》中的[IAM 服務帳戶的角色](#)。不過，對於更簡單的選項，如果您的 [SDK 支援](#)，建議您改用 [Amazon EKS Pod 身分](#)。

步驟 1：設定身分提供者和 IAM 角色

若要使用外部 IdP 設定聯合，請使用 IAM 身分提供者來通知 AWS 外部 IdP 及其組態。這可建立 AWS 帳戶 和外部 IdP 之間的信任。在設定 SDK 以使用 JSON Web 權杖 (JWT) 進行身分驗證之前，您必須

先設定身分提供者 (IdP) 和用來存取該權杖IAM的角色。若要設定這些項目，請參閱IAM《使用者指南》中的[為 Web 身分或 OpenID Connect Federation \(主控台\) 建立角色](#)。

步驟 2：設定 SDK 或 工具

將 SDK 或 工具設定為使用來自的 JSON Web 權杖 (JWT) AWS STS 進行身分驗證。

當您在設定檔中指定此項目時，SDK 或 工具會自動為您進行對應的 AWS STS [AssumeRoleWithWebIdentity](#) API 呼叫。若要使用 Web 聯合身分擷取和使用臨時登入資料，請在共用 AWS config 檔案中指定下列組態值。如需這些設定的詳細資訊，請參閱[擔任角色登入資料提供者設定](#)一節。

- `role_arn` - 來自您在步驟 1 中建立 IAM 的角色
- `web_identity_token_file` - 從外部 IdP
- (選用) `duration_seconds`
- (選用) `role_session_name`

以下是共用 config 檔案組態的範例，以擔任具有 Web 身分的角色：

```
[profile web-identity]  
role_arn=arn:aws:iam::123456789012:role/my-role-name  
web_identity_token_file=/path/to/a/token
```

Note

對於行動應用程式，請考慮使用 Amazon Cognito。Amazon Cognito 擔任身分代理程式，並為您執行大部分聯合工作。不過，Amazon Cognito 身分提供者不包含在 SDKs 和工具核心程式庫中，就像其他身分提供者一樣。若要存取 Amazon Cognito API，請在或 SDK 工具的建置或程式庫中包含 Amazon Cognito 服務用戶端。如需搭配使用 AWS SDKs，請參閱《Amazon Cognito 開發人員指南》中的[程式碼範例](#)。

如需所有擔任角色登入資料提供者設定的詳細資訊，請參閱本指南[擔任角色登入資料提供者](#)中的。

AWS 存取金鑰

使用短期憑證

我們建議您將 SDK 或工具設定為使用 [IAM 您 SDK 或 工具的身分中心身分驗證](#) 延長工作階段持續時間選項。

不過，若要直接設定 SDK 或工具的臨時認證，請參閱 [使用短期憑證進行驗證](#)。

使用長期認證

Warning

為避免安全風險，在開發專用軟體或使用真實資料時，請勿使用 IAM 使用者進行身分驗證。相反地，搭配使用聯合功能和身分提供者，例如 [AWS IAM Identity Center](#)。

管理存取 AWS 帳戶

作為安全性最佳實務，我們建議您 AWS Organizations 搭配 IAM 身分中心管理所有人的存取 AWS 帳戶。如需詳細資訊，請參閱《IAM 使用者指南》中的 [IAM 中的安全性最佳實務](#)。

您可以在身分識別中心建立使用者、使用 Microsoft 作用中目錄、使用 SAML 2.0 身分識別提供者 (IdP)，或個別將您的 IdP 聯合至。AWS 帳戶使用其中一種方法，您可以為使用者提供單一登入體驗。您也可以強制執行多重要素驗證 (MFA)，並使用臨時登入資料進行存取 AWS 帳戶。這與 IAM 使用者不同，IAM 使用者是可共用的長期登入資料，可能會增加 AWS 資源的安全風險。

僅針對沙箱環境建立 IAM 使用者

如果您不熟悉 AWS，可以建立測試 IAM 使用者，然後使用它來執行教學課程並探索可提供的內容 AWS。學習時可以使用這種類型的認證，但我們建議您避免在沙箱環境之外使用它。

對於以下使用案例，在中開始使用 IAM 使用者可能很有意義 AWS：

- 開始使用 AWS SDK 或工具，並在沙箱環境 AWS 服務 中進行探索。
- 執行排程的指令碼、工作和其他不支援人工登入程序的自動化程序，做為學習的一部分。

如果您在這些使用案例以外使用 IAM 使用者，請盡快轉換至 IAM 身分中心或聯合身分提供者。AWS 帳戶 如需詳細資訊，請參閱 [中的 AWS](#)。

安全的 IAM 使用者存取金鑰

您應該定期輪換 IAM 使用者存取金鑰。請遵循 IAM 使用者指南中[輪換存取金鑰](#)中的指引。如果您認為自己不小心共用了 IAM 使用者存取金鑰，請輪換您的存取金鑰。

IAM 使用者存取金鑰應儲存在本機電腦的共用 AWS credentials 檔案中。請勿在程式碼中儲存 IAM 使用者存取金鑰。請勿在任何原始程式碼管理軟體中包含 IAM 使用者存取金鑰的設定檔。外部工具（例如開源項目 [git-secrets](#)）可以幫助您無意中將敏感信息提交到 Git 存儲庫。如需詳細資訊，請參閱[IAM 使用者指南中的 IAM 身分 \(使用者、使用者群組和角色\)](#)。

若要設定 IAM 使用者以開始使用，請參閱 [使用長期登入資料進行驗證](#)。

使用短期憑證進行驗證

建議您設定 SDK 或 工具，以 [IAM 您 SDK 或 工具的身分中心身分驗證](#) 搭配延長工作階段持續時間選項使用。不過，您可以複製並使用 AWS 存取入口網站中提供的臨時登入資料。若這些憑證過期，便需要複製新憑證。您可以在設定檔中使用臨時憑證，或用作系統屬性和環境變數的值。

最佳實務：建議您的應用程式使用從下列位置交付的臨時憑證，而不是手動管理 登入資料檔案中的存取金鑰和權杖：

- AWS 運算服務，例如在 Amazon Elastic Compute Cloud 或 中執行應用程式 AWS Lambda。
- 登入資料提供者鏈結中的另一個選項，例如 [IAM 您 SDK 或 工具的身分中心身分驗證](#)。
- 或使用 [程序認證提供者](#) 擷取臨時登入資料。

使用從 AWS 存取入口網站擷取的短期登入資料來設定登入資料檔案

1. [建立共用登入資料檔案](#)。
2. 在登入資料檔案中，貼上下列預留位置文字，直到您貼上運作中的臨時登入資料為止。

```
[default]
aws_access_key_id=<value from AWS access portal>
aws_secret_access_key=<value from AWS access portal>
aws_session_token=<value from AWS access portal>
```

3. 儲存檔案。檔案現在 `~/.aws/credentials` 應該存在於您的本機開發系統上。此檔案包含 SDK 或工具在未指定特定具名設定檔時使用的 [【預設】](#) 設定檔。
4. [登入 AWS 存取入口網站](#)。

- 務必NOT將常值存取金鑰或登入資料資訊放入您的應用程式檔案中。如果您不小心這麼做了，則會有暴露您登入資料的風險，例如，當您上傳專案到公有儲存庫時。
- 請在NOT您的專案區域中包含包含登入資料的檔案。
- 請注意，存放在共用 AWS credentials檔案中的任何登入資料都會以純文字儲存。

安全管理憑證的其他指引

如需如何安全管理 AWS 憑證的一般討論，請參閱 [中管理 AWS 存取金鑰的最佳實務AWS 一般參考](#)。除了討論之外，請考慮下列事項：

- 針對 [IAM Amazon Elastic Container Service \(Amazon \) 任務使用 角色](#)。ECS
- 針對在 Amazon EC2執行個體上執行的應用程式使用 [IAM角色](#)。

先決條件：建立 AWS 帳戶

若要使用 IAM使用者存取 AWS 服務，您需要 AWS 帳戶和 AWS 登入資料。

1. 建立帳戶。

若要建立 AWS 帳戶，請參閱 AWS Account Management 參考指南中的 [入門：您是第一次 AWS 使用嗎？](#)。

2. 建立管理使用者。

避免使用根使用者帳戶 (您建立的初始帳戶) 來存取管理主控台與服務。反之，請建立管理使用者帳戶，如 IAM 使用者指南中的 [建立管理使用者](#) 所述。

建立管理使用者帳戶並記錄登入詳細資料之後，請務必登出您的根使用者帳戶，然後使用管理帳戶重新登入。

這些帳戶都不適合在 上執行開發，AWS 或在 上執行應用程式 AWS。最佳實務是，您需要建立適合這些任務的使用者、許可集或服務角色。如需詳細資訊，請參閱《IAM 使用者指南》中的 [套用最低權限許可](#)。

步驟 1：建立您的IAM使用者

- IAM 遵循 使用者指南中的 [建立IAM使用者 \(主控台 \)](#) 程序來建立您的使用者。IAM 建立IAM使用者時：

- 建議您選取提供使用者存取權 AWS Management Console。這可讓您檢視與在視覺化環境中執行的程式碼 AWS 服務 相關的，例如檢查 AWS CloudTrail 診斷日誌或將檔案上傳至 Amazon Simple Storage Service，這在偵錯程式碼時很有幫助。
- 針對設定許可 - 許可選項，選取直接連接政策，以了解如何將許可指派給此使用者。
 - 大多數「入門」SDK教學課程使用 Amazon S3 服務作為範例。若要讓應用程式能夠完整存取 Amazon S3，請選取要連接至此使用者的 AmazonS3FullAccess 政策。
- 您可以忽略該程序有關設定許可界限或標籤的選用步驟。

步驟 2：取得您的存取金鑰

1. 在IAM主控台的導覽窗格中，選取使用者，然後選取您先前建立的使用者 **User name** 的。
2. 在使用者頁面上，選取安全憑證頁面。接著，在存取金鑰下，選取建立存取金鑰。
3. 針對建立存取金鑰步驟 1，選擇命令列界面 (CLI) 或本機程式碼。這兩個選項都會產生與 AWS CLI 和 搭配使用的相同類型金鑰 SDKs。
4. 在建立存取金鑰步驟 2 中，輸入選用標籤並選取下一步。
5. 針對建立存取金鑰步驟 3，選取下載 .csv 檔案，以儲存具有 IAM 使用者存取金鑰和私密存取金鑰 .csv 的檔案。您之後將會用到此資訊。

Warning

使用適當的安全措施來保護這些登入資料的安全。

6. 選取 Done (完成)。

步驟 3：更新共用 **credentials** 檔案

1. 建立或開啟共用的 AWS credentials 檔案。這個檔案是位於 Linux 和 macOS 系統上的 `~/.aws/credentials` 和 Windows 上的 `%USERPROFILE%\aws\credentials`。如需詳細資訊，請參閱[登入資料檔案的位置](#)。
2. 將以下文字新增至共用的 credentials 檔案。將範例 ID 值和範例索引鍵值取代為您先前下載的 .csv 檔案中的值。

```
[default]
aws_access_key_id = AKIAIOSFODNN7EXAMPLE
```

```
aws_secret_access_key = wJa1rXUtnFEMI/K7MDENG/bPxrRfiCYEXAMPLEKEY
```

3. 儲存檔案。

共用credentials檔案是存放登入資料的最常見方式。這些也可以設定為環境變數，請參閱 [AWS 存取金鑰](#) 以取得環境變數名稱。這是讓您開始使用的方法，但我們建議您盡快轉換到 IAM Identity Center 或其他臨時登入資料。在您轉換不使用長期憑證之後，請記得從共用credentials檔案刪除這些憑證。

使用 Amazon EC2 執行個體的IAM角色

本範例涵蓋設定具有 Amazon S3 存取權的 AWS Identity and Access Management 角色，以便在部署到 Amazon EC2 執行個體的應用程式中使用。

若要在 Amazon 彈性運算雲端執行個體上執行 AWS SDK應用程式，請建立IAM角色，然後將該角色的存取權授予 Amazon EC2 執行個體。如需詳細資訊，請參閱 [Amazon EC2使用者指南EC2中的 Amazon IAM 角色](#)。

建立 IAM 角色

您開發的 AWS SDK應用程式可能會存取至少一個 AWS 服務 以執行動作。建立IAM角色，以授與執行應用程式所需的必要權限。

此程序會建立授與 Amazon S3 唯讀存取權限的角色做為範例。許多指 AWS SDK南都有從 Amazon S3 讀取的「入門」教學課程。

1. 登入 AWS Management Console 並開啟IAM主控台，位於<https://console.aws.amazon.com/iam/>。
2. 在導覽窗格中，選取角色，然後選取建立角色。
3. 針對 [選取信任的實體]，在 [信任的實體類型] 下選擇 AWS 服務
4. 在使用案例下，選擇 Amazon EC2，然後選擇下一步。
5. 對於新增許可，請從政策清單中選取 Amazon S3 唯讀存取的核取方塊，然後選取下一步。
6. 輸入角色的名稱，然後選取建立角色。請記住此名稱，因為在創建 Amazon EC2 實例時將需要它。

啟動 Amazon EC2 執行個體並指定您的IAM角色

您可以EC2執行以下操作，使用您的IAM角色建立和啟動 Amazon 執行個體：

- 遵循 Amazon EC2 使用者指南中的[快速啟動執行個體](#)。不過，在最終提交步驟之前，請執行下列動作：
 - 在 [進階詳細資料] 下方，針對 [IAM執行個體設定檔]，選擇您在上一步中建立的角色。

透過這項設定IAM和 Amazon EC2 設定，您可以將應用程式部署到 Amazon EC2 執行個體，而且您的應用程式將擁有 Amazon S3 服務的讀取存取權限。

Connect 至EC2執行個體

Connect 到 Amazon EC2 執行個體，以便將應用程式傳輸到該執行個體，然後執行應用程式。當您建立執行個體時，您需要包含您在 key pair (登入) 下使用之金鑰組私密部分的檔案；也就是PEM檔案。

您可以按照執行個體類型的指導來執行此操作：[Connect 到 Linux 執行個體](#)或 [Connect 到 Windows 執行個體](#)。連線時，請以這樣的方式將檔案從開發機器傳輸到執行個體。

Note

在 Linux 或 macOS 終端機上，您可以使用安全複製命令複製應用程式。若要搭配 key pair 使用 scp，您可以使用下列指令：`scp -i path/to/key file/to/copy ec2-user@ec2-xx-xx-xxx-xxx.compute.amazonaws.com:~`。

如需 Windows 的詳細資訊，請參閱[將檔案傳輸到 Windows 執行個體](#)。

如果您使用的是 AWS Toolkit，通常也可以使用 Toolkit 連線至執行個體。如需詳細資訊，請參閱您所使用之工具組的特定使用者指南。

在執行個EC2體上執行應用程式

1. 將您的應用程式檔案從本機磁碟機複製到 Amazon EC2 執行個體。
2. 啟動應用程式，並確認其執行結果與開發電腦上的結果相同。
3. (選擇性) 確認應用程式是否使用IAM角色提供的認證。
 - a. 登錄 AWS Management Console 並在打開 Amazon EC2 控制台<https://console.aws.amazon.com/ec2/>。

- b. 選取執行個體。
- c. 選擇動作、安全性，然後選擇修改IAM角色。
- d. 對於IAM角色，請選擇無IAM角色來卸離IAM角色。
- e. 選擇 [更新IAM角色]。
- f. 再次執行應用程式，並確認其傳回授權錯誤。

設定參考

SDKs APIs為 提供特定語言 AWS 服務。他們負責處理成功進行API呼叫所需的一些繁重任務，包括身分驗證、重試行為等。為此， SDKs具有彈性的策略，以取得用於請求的憑證、維護要與每個 服務搭配使用的設定，以及取得要用於全域設定的值。

您可以在下列各節中找到組態設定的詳細資訊：

- [AWS SDKs 和 工具標準化憑證提供者](#) – 跨多個 標準化的常見憑證提供者SDKs。
- [AWS SDKs和工具標準化功能](#) – 跨多個 標準化的常見功能SDKs。

建立服務用戶端

若要以程式設計方式存取 AWS 服務，請為每個 SDKs使用用戶端類別/物件 AWS 服務。例如，如果您的應用程式需要存取 Amazon EC2，則應用程式會建立 Amazon EC2用戶端物件以與該服務進行界面。然後，您可以使用 服務用戶端向該 提出請求 AWS 服務。在大多數 中SDKs，服務用戶端物件是不可變的，因此您必須為您提出請求的每個服務建立新的用戶端，並使用不同的組態對相同的服務提出請求。

設定優先順序

全域設定可設定大多數 支援且對 SDKs具有廣泛影響的功能、憑證提供者和其他功能 AWS 服務。所有 SDKs都有一系列位置（或來源）進行檢查，以尋找全域設定的值。以下是設定查詢優先順序：

1. 程式碼中或服務用戶端本身上設定的任何明確設定，都優先於任何其他設定。
 - 某些設定可以按每次操作進行設定，並且可以根據需要為您叫用的每個操作進行變更。對於 AWS CLI 或 AWS Tools for PowerShell，這些採用您在命令列中輸入的每個操作參數形式。對於 SDK，明確指派可以採用您在實例化 AWS 服務 用戶端或組態物件時設定的參數形式，有時也可以在呼叫個別 時設定API。
2. 僅限 Java/Kotlin：會檢查 設定的JVM系統屬性。如果已設定，該值會用來設定用戶端。
3. 檢查 環境變數。如果已設定，該值會用來設定用戶端。
4. 會SDK檢查 設定的共用credentials檔案。如果已設定，用戶端會使用它。
5. 設定的共用config檔案。如果設定存在，則 SDK會使用它。
 - AWS_PROFILE 環境變數或aws.profileJVM系統屬性可用來指定SDK載入的設定檔。
6. SDK 原始碼本身提供的任何預設值都會最後使用。

Note

有些 SDKs 和 工具可能會以不同的順序檢查。此外，部分 SDKs 和 工具支援其他儲存和擷取參數的方法。例如，AWS SDK for .NET 支援名為 [SDK Store](#) 的額外來源。如需 SDK 或 工具唯一提供者的詳細資訊，請參閱您正在使用的 SDK 或 工具的特定指南。

順序會決定哪些方法優先並覆寫其他方法。例如，如果您在共用 config 檔案中設定設定檔，只有在 SDK 或 工具先檢查其他地方之後，才會找到和使用它。這表示，如果您在 credentials 檔案中放置設定，則會使用它，而不是在 config 檔案中找到的設定。如果您使用設定和值設定環境變數，則會覆寫 credentials 和 config 檔案中的該設定。最後，個別操作（AWS CLI 命令列參數或 API 參數）或程式碼中的設定會覆寫該命令的所有其他值。

設定頁面

本指南的設定參考區段中的頁面詳細說明可透過各種機制設定的可用設定。下列資料表會列出組態和憑證檔案設定、環境變數，以及（適用於 Java 和 Kotlin SDKs）可在程式碼之外用來設定功能的 JVM 設定。每個清單中的每個連結主題都會帶您前往對應的設定頁面。

- [Config 檔案設定清單](#)
- [Credentials 檔案設定清單](#)
- [環境變數清單](#)
- [JVM 系統屬性清單](#)

每個憑證提供者或功能都有一個頁面，其中列出用於設定該功能的設定。對於每個設定，您通常可以透過將設定新增至組態檔案，或透過設定環境變數，或（僅適用於 Java 和 Kotlin）設定 JVM 系統屬性來設定值。每個設定都會列出描述詳細資訊上方區塊中值的所有支援方法。雖然 [優先順序](#) 不同，但無論如何設定，產生的功能都相同。

說明將包含預設值，如果有的話，如果不採取任何動作，就會生效。它也定義了該設定的有效值。

例如，我們來看看 [請求壓縮](#) 功能頁面中的設定。

`disable_request_compression` 範例設定的資訊會傳達下列項目：

- 控制程式碼庫外部的請求壓縮有三種等效方法。您可擇一方法：
 - 使用在組態檔案中設定 `disable_request_compression`

- 使用 將其設定為環境變數 `AWS_DISABLE_REQUEST_COMPRESSION`
- 或者，如果您使用 Java 或 Kotlin SDK，請使用 將其設定為JVM系統屬性 `aws.disableRequestCompression`

Note

也可能有直接在程式碼中設定相同功能的方法，但此參考不會涵蓋此項目，因為每個都是唯一的SDK。如果您想要在程式碼本身中設定組態，請參閱您的特定SDK指南或API參考。

- 如果您不採取任何動作，值會預設為 `false`。
- 此布林值設定的唯一有效值為 `true`和 `false`。

在每個功能頁面的底部，都有與資料表的相容性 AWS SDKs。

此資料表顯示您的 是否SDK支援頁面上列出的設定。Supported 資料欄會以下列值表示支援層級：

- Yes – SDK所寫入的設定完全受 支援。
- Partial – 支援某些設定，或行為偏離描述。對於 Partial，額外的備註會指出偏差。
- No – 不支援任何設定。這不會宣告程式碼中是否可以實現相同的功能；它只會指出不支援列出的外部組態設定。

Config 檔案設定清單

下表中列出的設定可以在共用 AWS config檔案中指派。它們是全域的，會影響所有 AWS 服務。SDKs 和 工具也可能支援唯一的設定和環境變數。若要查看僅個人或SDK工具支援的設定和環境變數，請參閱該特定SDK或工具指南。

設定名稱	詳細資訊
account_id_endpoint_mode	帳戶型端點
api_versions	一般組態設定

設定名稱	詳細資訊
aws_access_key_id	AWS 存取金鑰
aws_account_id	帳戶型端點
aws_secret_access_key	AWS 存取金鑰
aws_session_token	AWS 存取金鑰
ca_bundle	一般組態設定
credential_process	程序憑證提供者
credential_source	假設角色憑證提供者
defaults_mode	智慧組態預設值
disable_request_compression	請求壓縮
duration_seconds	假設角色憑證提供者
ec2_metadata_service_endpoint	IMDS 憑證提供者
ec2_metadata_service_endpoint_mode	IMDS 憑證提供者

設定名稱	詳細資訊
ec2_metadata_v1_disabled	IMDS 憑證提供者
endpoint_discovery_enabled	端點探索
endpoint_url	服務特定的端點
external_id	假設角色憑證提供者
ignore_configured_endpoint_urls	服務特定的端點
max_attempts	重試行為
metadata_service_num_attempts	Amazon EC2執行個體中繼資料
metadata_service_timeout	Amazon EC2執行個體中繼資料
mfa_serial	假設角色憑證提供者
output	一般組態設定
parameter_validation	一般組態設定
region	AWS 區域

設定名稱	詳細資訊
request_m in_compre ssion_siz e_bytes	請求壓縮
retry_mode	重試行為
role_arn	假設角色憑證提供者
role_sess ion_name	假設角色憑證提供者
s3_disabl e_multire gion_acce ss_points	Amazon S3 多區域存取點
s3_use_ar n_region	Amazon S3 存取點
sdk_ua_app_id	應用程式 ID
source_profile	假設角色憑證提供者
sso_account_id	IAM Identity Center 憑證提供者
sso_region	IAM Identity Center 憑證提供者
sso_regis tration_scopes	IAM Identity Center 憑證提供者
sso_role_name	IAM Identity Center 憑證提供者
sso_start_url	IAM Identity Center 憑證提供者
sts_regio nal_endpoints	AWS STS 區域端點

設定名稱	詳細資訊
use_duals_tack_endpoint	雙堆疊和FIPS端點
use_fips_endpoint	雙堆疊和FIPS端點
web_identity_token_file	假設角色憑證提供者

Credentials 檔案設定清單

下表中列出的設定可以在共用 AWS credentials 檔案中指派。它們是全域的，會影響所有 AWS 服務。SDKs 和 工具也可能支援唯一的設定和環境變數。若要查看僅個人或SDK工具支援的設定和環境變數，請參閱該特定SDK或工具指南。

設定名稱	詳細資訊
aws_access_key_id	AWS 存取金鑰
aws_secret_access_key	AWS 存取金鑰
aws_session_token	AWS 存取金鑰

環境變數清單

下表SDKs列出大多數 支援的環境變數。它們是全域的，會影響所有 AWS 服務。SDKs 和 工具也可能支援唯一的設定和環境變數。若要查看僅個人或SDK工具支援的設定和環境變數，請參閱該特定SDK或工具指南。

設定名稱	詳細資訊
AWS_ACCESS_KEY_ID	AWS 存取金鑰
AWS_ACCOUNT_ID	帳戶型端點
AWS_ACCOUNT_ID_ENDPOINT_MODE	帳戶型端點
AWS_CA_BUNDLE	一般組態設定
AWS_CONFIG_FILE	共用 config 和 credentials 檔案的位置
AWS_CONTAINER_AUTHORIZATION_TOKEN	容器憑證提供者
AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE	容器憑證提供者
AWS_CONTAINER_CREDENTIALS_FULL_URI	容器憑證提供者
AWS_CONTAINER_CREDENTIALS_RELATIVE_URI	容器憑證提供者
AWS_DEFAULTS_MODE	智慧組態預設值

設定名稱	詳細資訊
AWS_DISABLE_REQUEST_COMPRESSION	請求壓縮
AWS_EC2_METADATA_DISABLED	IMDS 憑證提供者
AWS_EC2_METADATA_SERVICE_ENDPOINT	IMDS 憑證提供者
AWS_EC2_METADATA_SERVICE_ENDPOINT_MODE	IMDS 憑證提供者
AWS_EC2_METADATA_V1_DISABLED	IMDS 憑證提供者
AWS_ENABLE_ENDPOINT_DISCOVERY	端點探索
AWS_ENDPOINT_URL	服務特定的端點
AWS_ENDPOINT_URL_<SERVICE>	服務特定的端點

設定名稱	詳細資訊
AWS_IGNORE_CONFIGURED_ENDPOINT_URLS	服務特定的端點
AWS_MAX_ATTEMPTS	重試行為
AWS_METADATA_SERVICE_NUM_ATTEMPTS	Amazon EC2執行個體中繼資料
AWS_METADATA_SERVICE_TIMEOUT	Amazon EC2執行個體中繼資料
AWS_PROFILE	共用 config 和 credentials 檔案
AWS_REGION	AWS 區域
AWS_REQUEST_MIN_COMPRESSION_SIZE_BYTES	請求壓縮
AWS_RETRY_MODE	重試行為
AWS_ROLE_ARN	假設角色憑證提供者
AWS_ROLE_SESSION_NAME	假設角色憑證提供者
AWS_S3_MULTIREGION_ACCESS_POINTS	Amazon S3 多區域存取點

設定名稱	詳細資訊
AWS_S3_US E_ARN_REGION	Amazon S3 存取點
AWS_SDK_U A_APP_ID	應用程式 ID
AWS_SECRE T_ACCESS_KEY	AWS 存取金鑰
AWS_SESSI ON_TOKEN	AWS 存取金鑰
AWS_SHARE D_CREDENT IALS_FILE	共用 config和 credentials 檔案的位置
AWS_STS_R EGIONAL_E NDPOINTS	AWS STS 區域端點
AWS_USE_D UALSTACK_ ENDPOINT	雙堆疊和FIPS端點
AWS_USE_F IPS_ENDPOINT	雙堆疊和FIPS端點
AWS_WEB_I DENTITY_T OKEN_FILE	假設角色憑證提供者

JVM 系統屬性清單

您可以針對 AWS SDK for Java 和 適用於 Kotlin 的 AWS SDK (以為目標JVM) 使用下列JVM系統屬性。如需如何設定JVM系統屬性的指示[the section called “如何設置 JVM 系統屬性”](#)，請參閱。

設定名稱	詳細資訊
<code>aws.accessKeyId</code>	AWS 存取金鑰
<code>aws.accountId</code>	帳戶型端點
<code>aws.accountIdEndpointMode</code>	帳戶型端點
<code>aws.configFile</code>	共用 config 和 credentials 檔案的位置
<code>aws.defaultsMode</code>	智慧組態預設值
<code>aws.disableEc2MetadataV1</code>	IMDS 憑證提供者
<code>aws.disableRequestCompression</code>	請求壓縮
<code>aws.ec2MetadataServiceEndpoint</code>	IMDS 憑證提供者
<code>aws.ec2MetadataServiceEndpointMode</code>	IMDS 憑證提供者
<code>aws.endpointDiscoveryEnabled</code>	端點探索
<code>aws.endpointUrl</code>	服務特定的端點

設定名稱	詳細資訊
<code>aws.endpointUrl<ServiceName></code>	服務特定的端點
<code>aws.ignoreConfiguredEndpointUrls</code>	服務特定的端點
<code>aws.maxAttempts</code>	重試行為
<code>aws.profile</code>	共用 config 和 credentials 檔案
<code>aws.region</code>	AWS 區域
<code>aws.requestMinCompressionSizeBytes</code>	請求壓縮
<code>aws.retryMode</code>	重試行為
<code>aws.roleArn</code>	假設角色憑證提供者
<code>aws.roleSessionName</code>	假設角色憑證提供者
<code>aws.s3DisableMultiRegionAccessPoints</code>	Amazon S3 多區域存取點
<code>aws.s3UseArnRegion</code>	Amazon S3 存取點
<code>aws.secretAccessKey</code>	AWS 存取金鑰

設定名稱	詳細資訊
<code>aws.sessionToken</code>	AWS 存取金鑰
<code>aws.sharedCredentialsFile</code>	共用 config和 credentials 檔案的位置
<code>aws.useDualstackEndpoint</code>	雙堆疊和FIPS端點
<code>aws.useFipsEndpoint</code>	雙堆疊和FIPS端點
<code>aws.userAgentAppId</code>	應用程式 ID
<code>aws.webIdentityTokenFile</code>	假設角色憑證提供者

AWS SDKs 和 工具標準化憑證提供者

許多憑證提供者已標準化為一致的預設值，並在許多中以相同的方式運作SDKs。此一致性可提高跨多個進行編碼時的生產力和清晰度SDKs。所有設定都可以在程式碼中覆寫。如需詳細資訊，請參閱您的特定 SDK API。

Important

並非所有供應商都SDKs支援所有供應商，甚至供應商內的所有方面。

主題

- [了解憑證提供者鏈](#)
- [SDK特定和工具特定的憑證提供者鏈](#)

- [AWS 存取金鑰](#)
- [擔任角色登入資料提供者](#)
- [容器憑證提供者](#)
- [IAM Identity Center 登入資料提供者](#)
- [IMDS 憑證提供者](#)
- [程序認證提供者](#)

了解憑證提供者鏈

所有 SDKs 都有一系列位置（或來源）進行檢查，以尋找可用於向提出請求的有效憑證 AWS 服務。找到有效的憑證後，系統就會停止搜尋。此系統搜尋稱為憑證提供者鏈。

使用其中一個標準化憑證提供者時，AWS SDKs 一律會在憑證過期時嘗試自動續約憑證。內建憑證提供者鏈可讓您的應用程式重新整理憑證，無論您在鏈中使用哪個提供者。SDK 執行此操作不需要額外的程式碼。

雖然每個所使用的不同鏈 SDK 都不同，但它們通常包含下列來源：

憑證提供者	描述
AWS 存取金鑰	AWS IAM 使用者的存取金鑰（例如 <code>AWS_ACCESS_KEY_ID</code> 和 <code>AWS_SECRET_ACCESS_KEY</code> ）。
與 Web 身分或 OpenID Connect 聯合 - 擔任角色憑證提供者	使用知名的外部身分提供者 IdP 登入，例如使用 Amazon、Facebook、Google 或任何其他 OpenID Connect (OIDC) 相容 IdP 登入。使用來自 (JWT) 的 JSON Web 權杖 AWS Security Token Service () 擔任 IAM 角色的許可 AWS STS。
IAM Identity Center 登入資料提供者	從取得憑證 AWS IAM Identity Center。
擔任角色登入資料提供者	取得 IAM 角色的許可，以取得其他資源的存取權。（擷取角色的臨時憑證，然後使用該角色的臨時憑證）。
容器憑證提供者	Amazon Elastic Container Service (Amazon ECS) 和 Amazon Elastic Kubernetes Service (Amazon EKS)

憑證提供者	描述
	憑證。容器憑證提供者會為客戶的容器化應用程式擷取憑證。
程序認證提供者	自訂憑證提供者。從外部來源或程序取得您的憑證，包括 IAM Roles Anywhere。
IMDS 憑證提供者	Amazon Elastic Compute Cloud (Amazon EC2) 執行個體設定檔憑證。將IAM角色與每個EC2執行個體建立關聯。該角色的臨時憑證可供執行個體中執行的程式碼使用。憑證是透過 Amazon EC2中繼資料服務傳遞。

對於鏈中的每個步驟，有多種指派設定值的方法。設定在程式碼中指定的值一律優先。不過，也有 [環境變數](#) 和 [共用 config 和 credentials 檔案](#)。如需詳細資訊，請參閱 [設定優先順序](#)。

SDK特定和工具特定的憑證提供者鏈

若要直接前往 SDK或工具的特定憑證提供者鏈詳細資訊，請從下列選擇您的 SDK或工具：

- [AWS CLI](#)
- [SDK 適用於 C++](#)
- [SDK for Go](#)
- [SDK 適用於 Java](#)
- [SDK 適用於 JavaScript](#)
- [SDK 適用於 Kotlin](#)
- [SDK 適用於 。NET](#)
- [SDK 適用於 PHP](#)
- [SDK for Python \(Boto3 \)](#)
- [SDK 適用於 Ruby](#)
- [SDK for Rust](#)
- [SDK 適用於 Swift](#)
- [適用於的工具 PowerShell](#)

AWS 存取金鑰

Warning

為了避免安全風險，開發專用軟體或使用真實資料時，請勿使用IAM使用者進行身分驗證。相反地，搭配使用聯合功能和身分提供者，例如 [AWS IAM Identity Center](#)。

AWS IAM使用者的存取金鑰可以用作您的 AWS 登入資料。AWS SDK 會自動使用這些 AWS 登入資料來簽署API請求 AWS，讓您的工作負載能夠安全且方便地存取您的 AWS 資源和資料。建議一律使用 `aws_session_token`讓憑證暫時且過期後不再有效。不建議使用長期登入資料。

Note

如果 AWS 無法重新整理這些臨時登入資料，AWS 可能會延長登入資料的有效性，讓您的工作負載不會受到影響。

共用 AWS `credentials`檔案是存放登入資料資訊的建議位置，因為它是在應用程式來源目錄外，並與共用`config`檔案的 SDK特定設定分開。

若要進一步了解 AWS 登入資料和使用存取金鑰，請參閱IAM《使用者指南》中的[AWS 安全登入資料和管理IAM使用者的存取金鑰](#)。

使用下列設定此功能：

`aws_access_key_id` - 共用 AWS `config`檔案設定, `aws_access_key_id` - 共用 AWS `credentials`檔案設定（建議的方法）, `AWS_ACCESS_KEY_ID` - 環境變數, `aws.accessKeyId` - JVM系統屬性：僅限 Java/Kotlin

指定做為登入資料一部分以驗證使用者的 AWS 存取金鑰。

`aws_secret_access_key` - 共用 AWS `config`檔案設定, `aws_secret_access_key` - 共用 AWS `credentials`檔案設定（建議的方法）, `AWS_SECRET_ACCESS_KEY` - 環境變數, `aws.secretAccessKey` - JVM系統屬性：僅限 Java/Kotlin

指定做為登入資料的一部分用來驗證使用者的 AWS 私密金鑰。

`aws_session_token` - 共用 `AWS config` 檔案設定, `aws_session_token` - 共用 `AWS credentials` 檔案設定 (建議的方法), `AWS_SESSION_TOKEN` - 環境變數, `aws.sessionToken` - JVM系統屬性 : 僅限 Java/Kotlin

指定 AWS 工作階段權杖, 做為驗證使用者登入資料的一部分。您會收到此值, 做為成功請求所傳回暫時登入資料的一部分, 以擔任角色。只有當您手動指定臨時的安全憑證時, 才需要工作階段字符。不過, 我們建議您一律使用臨時安全登入資料, 而非長期登入資料。如需安全建議, 請參閱 [中的安全最佳實務IAM](#)。

如需如何取得這些值的說明, 請參閱 [使用短期憑證進行驗證](#)。

在 `config` 或 `credentials` 檔案中設定這些必要值的範例 :

```
[default]
aws_access_key_id = AKIAIOSFODNN7EXAMPLE
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
aws_session_token = AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJ0gQs8IZZaIv2BXIa2R40lgk
```

透過命令列設定環境變數的 Linux/macOS 範例 :

```
export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
export
AWS_SESSION_TOKEN=AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJ0gQs8IZZaIv2BXIa2R40lgk
```

透過命令列設定環境變數的 Windows 範例 :

```
setx AWS_ACCESS_KEY_ID AKIAIOSFODNN7EXAMPLE
setx AWS_SECRET_ACCESS_KEY wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
setx
AWS_SESSION_TOKEN AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJ0gQs8IZZaIv2BXIa2R40lgk
```

與 的相容性 AWS SDKs

以下 SDKs 支援本主題所述的功能和設定。會記下任何部分例外狀況。AWS SDK for Java 和 適用於 Kotlin 的 AWS SDK 僅支援任何 JVM 系統屬性設定。

SDK	支援	備註或更多資訊
AWS CLI v2	是	
SDK 適用於 C++	是	不支援共用config檔案。
SDK for Go V2 (1.x)	是	
SDK for Go 1.x (V1)	是	若要使用共用config檔案設定，您必須開啟從組態檔案載入；請參閱 工作階段 。
SDK for Java 2.x	是	
SDK 適用於 Java 1.x	是	
SDK for JavaScript 3.x	是	
SDK for JavaScript 2.x	是	
SDK 適用於 Kotlin	是	
SDK 適用於 。NET 3.x	是	不支援環境變數。
SDK 適用於 PHP 3.x	是	
SDK for Python (Boto3)	是	
SDK 適用於 Ruby 3.x	是	
SDK for Rust	是	
SDK 適用於 Swift	是	
適用於的工具 PowerShell	是	不支援環境變數。

擔任角色登入資料提供者

假設角色涉及使用一組臨時安全登入資料來存取 AWS 您可能無法存取的資源。這些臨時登入資料由存取金鑰 ID、私密存取金鑰和安全字符組成。

若要設定您的 SDK 或 工具以擔任角色，您必須先建立或識別要擔任的特定角色。IAM 角色由角色 Amazon Resource Name (ARN) 唯一識別。角色會與其他實體建立信任關係。使用 角色的受信任實體可能是 AWS 服務、另一個 AWS 帳戶、Web 身分提供者或 OIDC 或 SAML 聯合。

識別 IAM 角色之後，如果您受該角色信任，您可以設定 SDK 或 工具來使用角色授予的許可。若要這樣做，請使用下列設定。

如需開始使用這些設定的指引，請參閱本指南 [使用 AWS 憑證擔任角色](#) 中的。

擔任角色登入資料提供者設定

使用下列項目設定此功能：

credential_source - 共用 AWS config 檔案設定

在 Amazon EC2 執行個體或 Amazon Elastic Container Service 容器內使用，指定 SDK 或 工具可以找到具有許可的登入資料，以擔任您使用 `role_arn` 參數指定的角色。

預設值：無

有效值：

- 環境 – 指定 SDK 或 工具是從環境變數 [AWS_ACCESS_KEY_ID](#) 和 [AWS_SECRET_ACCESS_KEY](#) 擷取來源憑證。
- `Ec2InstanceMetadata` – 指定 SDK 或 工具使用 [IAM 連接至 EC2 執行個體設定檔的角色](#) 來取得來源憑證。
- `EcsContainer` – 指定 SDK 或 工具使用 [IAM 連接到 ECS 容器的角色](#) 來取得來源登入資料。

您無法在同一個描述檔中同時指定 `credential_source` 和 `source_profile`。

在 `config` 檔案中設定此項目以表示登入資料應來自 Amazon 的範例 EC2：

```
credential_source = Ec2InstanceMetadata
role_arn = arn:aws:iam::123456789012:role/my-role-name
```

duration_seconds - 共用 AWS config 檔案設定

指定角色工作階段的最大持續時間 (以秒為單位)。

此設定只有在設定檔指定 擔任角色時才適用。

預設值：3600 秒（一小時）

有效值：該值的範圍可以從 900 秒（15 分鐘）到為角色設定的工作階段持續時間上限設定（最多可達 43200 秒或 12 小時）。如需詳細資訊，請參閱IAM《使用者指南》中的[檢視角色的工作階段持續時間上限設定](#)。

在 config 檔案中設定的範例：

```
duration_seconds = 43200
```

external_id - 共用 AWS config 檔案設定

指定一個唯一識別符，第三方用來在其客戶帳戶擔任角色。

此設定僅適用於設定檔指定擔任角色，且角色的信任政策需要的值時ExternalId。當設定檔指定角色時，值會映射到傳遞至 AssumeRole 操作的 ExternalId 參數。

預設值：無。

有效值：請參閱IAM《使用者指南》中的[如何將 AWS 資源的存取權授予第三方時使用外部 ID](#)。

在 config 檔案中設定的範例：

```
external_id = unique_value_assigned_by_3rd_party
```

mfa_serial - 共用 AWS config 檔案設定

指定使用者擔任角色時必須使用的多重驗證 (MFA) 裝置的識別或序號。

擔任角色時為必要，其中該角色的信任政策包含需要MFA身分驗證的條件。如需的詳細資訊MFA，請參閱IAM《使用者指南》[AWS 中的多重要素驗證IAM](#)。

預設值：無。

有效值：該值可以是硬體裝置的序號（例如 GAHT12345678），或虛擬MFA裝置的 Amazon Resource Name (ARN)。的格式ARN為：arn:aws:iam::*account-id*:mfa/*mfa-device-name*

在 config 檔案中設定的範例：

此範例假設已為帳戶MyMFADevice建立並為使用者啟用的虛擬MFA裝置，稱為。

```
mfa_serial = arn:aws:iam::123456789012:mfa/MyMFADevice
```

role_arn - 共用 AWS **config** 檔案設定, **AWS_ROLE_ARN** - 環境變數, **aws.roleArn** - JVM系統屬性：僅限 Java/Kotlin

指定您要用來執行使用此設定檔請求之操作的角色的 Amazon Resource Name IAM (ARN)。

預設值：無。

有效值：該值必須是IAM角色ARN的，格式如下：`arn:aws:iam::account-id:role/role-name`

此外，您還必須指定下列其中一項設定：

- **source_profile** – 識別另一個設定檔，用來尋找具有在此設定檔中擔任角色之許可的登入資料。
- **credential_source** – 使用目前環境變數識別的登入資料，或連接至 Amazon EC2執行個體描述檔的登入資料，或 Amazon ECS容器執行個體。
- **web_identity_token_file** – 使用公有身分提供者或任何 OpenID Connect (OIDC) 相容身分提供者，供已在行動或 Web 應用程式中驗證的使用者使用。

role_session_name - 共用 AWS **config** 檔案設定, **AWS_ROLE_SESSION_NAME** - 環境變數, **aws.roleSessionName** - JVM系統屬性：僅限 Java/Kotlin

指定要連接到角色工作階段的名稱。此名稱會出現在與此工作階段相關聯的項目 AWS CloudTrail 日誌中，這在稽核時非常有用。如需詳細資訊，請參閱 AWS CloudTrail 使用者指南中的 [CloudTrail userIdentity 元素](#)。

預設值：選用參數。如果您不提供此值，則如果設定檔擔任角色，工作階段名稱會自動產生。

有效值：當 AWS CLI 或 AWS API 代表您呼叫 AssumeRole 操作（或操作等 AssumeRoleWithWebIdentity 操作）時，提供給 RoleSessionName 參數。此值會成為您可以查詢的擔任角色使用者 Amazon Resource Name (ARN) 的一部分，並顯示為此設定檔調用之操作的 CloudTrail 日誌項目的一部分。

```
arn:aws:sts::123456789012:assumed-role/my-role-name/my-role_session_name.
```

在 config 檔案中設定的範例：

```
role_session_name = my-role-session-name
```

source_profile - 共用 AWS config 檔案設定

指定另一個設定檔，其登入資料用於擔任原始設定檔中 `role_arn` 設定所指定的角色。若要了解如何在共用 AWS config 和 `credentials` 檔案中使用設定檔，請參閱 [共用 config 和 credentials 檔案](#)。

如果您指定的設定檔也是擔任角色設定檔，則每個角色都會依序擔任，以完全解析登入資料。當 SDK 遇到具有登入資料的設定檔時，就會停止此鏈結。角色鏈結會將您的 AWS CLI 或 AWS API 角色工作階段限制為最多一小時，且無法增加。如需詳細資訊，請參閱 IAM 《使用者指南》中的 [角色術語和概念](#)。

預設值：無。

有效值：文字字串，由 `config` 和 `credentials` 檔案中定義的設定檔名稱組成。您還必須在目前的設定檔 `role_arn` 中指定的值。

您無法在同一個描述檔中同時指定 `credential_source` 和 `source_profile`。

在組態檔案中設定的範例：

```
[profile A]
source_profile = B
role_arn = arn:aws:iam::123456789012:role/RoleA
role_session_name = ProfileARoleSession

[profile B]
credential_process = ./aws_signing_helper credential-process --certificate /
path/to/certificate --private-key /path/to/private-key --trust-anchor-
arn arn:aws:rolesanywhere:region:account:trust-anchor/TA_ID --profile-
arn arn:aws:rolesanywhere:region:account:profile/PROFILE_ID --role-arn
arn:aws:iam::account:role/ROLE_ID
```

在先前的範例中，A 設定檔會告知 SDK 或 工具自動查詢連結 B 設定檔的登入資料。在此情況下，B 設定檔會使用提供的登入資料協助工具 [IAM Roles Anywhere](#) 來取得的 AWS 登入資料 SDK。然後，您的程式碼會使用這些臨時登入資料來存取 AWS 資源。指定的角色必須具有連接 IAM 許可政策，以允許請求的程式碼執行，例如 命令 AWS 服務或 API 方法。設定檔採取的每個動作 A 都包含 CloudTrail 日誌中的角色工作階段名稱。

對於第二個角色鏈結範例，如果您在 Amazon Elastic Compute Cloud 執行個體上有應用程式，且您想要讓該應用程式擔任另一個角色，則可以使用下列組態。

```
[profile A]
```

```
source_profile = B
role_arn = arn:aws:iam::123456789012:role/RoleA
role_session_name = ProfileARoleSession

[profile B]
credential_source=Ec2InstanceMetadata
```

設定檔A將使用來自 Amazon EC2執行個體的登入資料擔任指定的角色，並自動續約登入資料。

web_identity_token_file - 共用 AWS **config**檔案設定, **AWS_WEB_IDENTITY_TOKEN_FILE** - 環境變數, **aws.webIdentityTokenFile** - JVM系統屬性：僅限 Java/Kotlin

指定檔案的路徑，其中包含來自[支援 2.0 OAuth 提供者](#)或[OpenID Connect ID 身分提供者](#)的存取權杖。

此設定透過使用 [Google](#)、[Facebook](#) 和 [Amazon](#) 等 Web 聯合身分提供者來啟用身分驗證。SDK 或開發人員工具會載入此檔案的內容，並在代您呼叫AssumeRoleWithWebIdentity操作時將其做為WebIdentityToken引數傳遞。

預設值：無。

有效值：此值必須是路徑和檔案名稱。檔案必須包含 2.0 OAuth 存取字符或身分提供者提供給您的 OpenID Connect 字符。相對路徑會被視為相對於程序的工作目錄。

與的相容性 AWS SDKs

以下SDKs支援本主題所述的功能和設定。會記下任何部分例外狀況。AWS SDK for Java 和 適用於 Kotlin 的 AWS SDK 僅支援任何JVM系統屬性設定。

SDK	支援	備註或更多資訊
AWS CLI v2	是	
SDK 適用於 C++	部分	credential_source 不支援。duration_seconds 不支援。mfa_serial 不支援。
SDK for Go V2 (1.x)	是	

SDK	支援	備註或更多資訊
SDK for Go 1.x (V1)	是	若要使用共用config檔案設定，您必須開啟從組態檔案載入；請參閱 工作階段 。
SDK 適用於 Java 2.x	部分	mfa_serial 不支援。duration_seconds 不支援。
SDK 適用於 Java 1.x	部分	credential_source 不支援。mfa_serial 不支援。不支援JVM系統屬性。
SDK for JavaScript 3.x	是	
SDK for JavaScript 2.x	部分	credential_source 不支援。
SDK 適用於 Kotlin	是	
SDK 適用於 。NET 3.x	是	
SDK 適用於 PHP 3.x	是	
SDK for Python (Boto3)	是	
SDK 適用於 Ruby 3.x	是	
SDK for Rust	是	
SDK 適用於 Swift	是	
的工具 PowerShell	是	

容器憑證提供者

容器憑證提供者會為客戶的容器化應用程式擷取憑證。此憑證提供者對 Amazon Elastic Container Service (Amazon ECS) 和 Amazon Elastic Kubernetes Service (Amazon EKS) 客戶很有用。SDKs 嘗試透過GET請求從指定的HTTP端點載入憑證。

如果您使用 Amazon ECS，我們建議您使用任務IAM角色來改善憑證隔離、授權和可稽核性。設定後，Amazon 會 ECS 設定 SDKs 和 工具用來取得憑證 `AWS_CONTAINER_CREDENTIALS_RELATIVE_URI` 的環境變數。若要 ECS 為此功能設定 Amazon，請參閱 Amazon Elastic Container Service 開發人員指南 中的 [任務IAM角色](#)。

如果您使用 Amazon EKS，我們建議您使用 Amazon EKS Pod Identity 來改善憑證隔離、最低權限、可稽核性、獨立操作、可重複使用性和可擴展性。您的 Pod 和 IAM 角色都與 Kubernetes 服務帳戶相關聯，以管理應用程式的憑證。若要進一步了解 Amazon EKS Pod Identity，請參閱 [Amazon EKS 使用者指南 中的 Amazon Pod Identity](#)。EKS 設定後，Amazon 會 ECS 設定 `AWS_CONTAINER_CREDENTIALS_FULL_URI` 和 工具用來取得憑證的 SDKs 和 `AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE` 環境變數。如需設定資訊，請參閱 [Amazon EKS 使用者指南或 Amazon Pod Identity 中的設定 Amazon Pod Identity 代理程式](#)，可簡化部落格網站上的 [Amazon EKS 叢集應用程式 IAM 許可 EKS](#)。EKS AWS

使用下列設定此功能：

`AWS_CONTAINER_CREDENTIALS_FULL_URI` - 環境變數

指定在請求憑證 SDK 時使用的完整 HTTP URL 端點。這包括配置和主機。

預設值：無。

有效值：有效 URI。

注意：此設定是 的替代方案 `AWS_CONTAINER_CREDENTIALS_RELATIVE_URI`，只有在 `AWS_CONTAINER_CREDENTIALS_RELATIVE_URI` 未設定時才會使用。

透過命令列設定環境變數的 Linux/macOS 範例：

```
export AWS_CONTAINER_CREDENTIALS_FULL_URI=http://localhost/get-credentials
```

或

```
export AWS_CONTAINER_CREDENTIALS_FULL_URI=http://localhost:8080/get-credentials
```

`AWS_CONTAINER_CREDENTIALS_RELATIVE_URI` - 環境變數

指定在請求憑證時 SDK 要使用的相對 HTTP URL 端點。該值會附加到 的預設 Amazon ECS 主機名稱 `169.254.170.2`。

預設值：無。

有效值：有效相對 URI。

透過命令列設定環境變數的 Linux/macOS 範例：

```
export AWS_CONTAINER_CREDENTIALS_RELATIVE_URI=/get-credentials?a=1
```

AWS_CONTAINER_AUTHORIZATION_TOKEN - 環境變數

以純文字指定授權權杖。如果設定此變數，SDK會在具有環境變數值的HTTP請求上設定授權標頭。

預設值：無。

有效值：字串。

注意：此設定是 的替代方案 `AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE`，只有在 `AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE` 未設定時才會使用。

透過命令列設定環境變數的 Linux/macOS 範例：

```
export AWS_CONTAINER_CREDENTIALS_FULL_URI=http://localhost/get-credential  
export AWS_CONTAINER_AUTHORIZATION_TOKEN=Basic abcd
```

AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE - 環境變數

指定檔案的絕對檔案路徑，其中包含純文字的授權字符。

預設值：無。

有效值：字串。

透過命令列設定環境變數的 Linux/macOS 範例：

```
export AWS_CONTAINER_CREDENTIALS_FULL_URI=http://localhost/get-credential  
export AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE=/path/to/token
```

與 的相容性 AWS SDKs

下列 SDKs 支援本主題所述的功能和設定。會記下任何部分例外狀況。AWS SDK for Java 和 適用於 Kotlin 的 AWS SDK 僅支援任何 JVM 系統屬性設定。

SDK	支援	備註或更多資訊
AWS CLI v2	是	
SDK 適用於 C++	是	
SDK for Go V2 (1.x)	是	
SDK for Go 1.x (V1)	是	
SDK 適用於 Java 2.x	是	AWS_CONTAINER_CREDENTIALS_FULL_URI 和 AWS_CONTAINER_AUTHORIZATION_TOKEN 也用於 Lambda SnapStart for Java 。
SDK 適用於 Java 1.x	是	AWS_CONTAINER_CREDENTIALS_FULL_URI 和 AWS_CONTAINER_AUTHORIZATION_TOKEN 也用於 Lambda SnapStart for Java 。
SDK 適用於 JavaScript 3.x	是	
SDK 適用於 JavaScript 2.x	是	
SDK 適用於 Kotlin	是	
SDK 適用於 。NET 3.x	是	
SDK 適用於 PHP 3.x	是	
SDK for Python (Boto3)	是	
SDK 適用於 Ruby 3.x	是	
SDK for Rust	是	
SDK 適用於 Swift	是	
適用於 的工具 PowerShell	是	

IAM Identity Center 登入資料提供者

此身分驗證機制會使用 AWS IAM Identity Center 來取得您程式碼 AWS 服務的單一登入 (SSO) 存取權。

Note

在 AWS SDK API 文件中，IAM Identity Center 登入資料提供者稱為 SSO 登入資料提供者。

啟用 IAM Identity Center 之後，您會為共用 AWS config 檔案中的設定定義設定檔。此設定檔用於連線至 IAM Identity Center 存取入口網站。當使用者成功向 IAM Identity Center 進行身分驗證時，入口網站會傳回與該使用者相關聯 IAM 角色的短期登入資料。若要了解如何從組態 SDK 取得臨時登入資料並將其用於 AWS 服務請求，請參閱 [瞭解 IAM 身分中心身分驗證](#)。

透過 config 檔案設定 IAM Identity Center 的方式有兩種：

- (建議) SSO 權杖提供者組態 – 延長工作階段持續時間。包括對自訂工作階段持續時間的支援。
- 傳統不可重新整理組態 – 使用固定的 8 小時工作階段。

在這兩種組態中，您需要在工作階段過期時再次登入。

下列兩個指南包含有關 IAM Identity Center 的其他資訊：

- [AWS IAM Identity Center 使用者指南](#)
- [AWS IAM Identity Center 入口網站 API 參考](#)

如需深入了解 SDKs 和 工具如何使用和重新整理使用此組態的登入資料，請參閱 [瞭解 IAM 身分中心身分驗證](#)。

必要條件

您必須先啟用 IAM Identity Center。如需啟用 IAM 身分中心身分驗證的詳細資訊，請參閱 AWS IAM Identity Center 《使用者指南》中的 [啟用 AWS IAM Identity Center](#)。

Note

或者，如需完整的先決條件和此頁面中詳細說明的必要共用config檔案組態，請參閱設定的引導說明[IAM 您 SDK或 工具的身分中心身分驗證](#)。

SSO 權杖提供者組態

當您使用SSO字提供者組態時，您的 AWS SDK 或 工具會自動重新整理工作階段，直到延長工作階段期間為止。如需工作階段持續時間和最長持續時間的詳細資訊，請參閱AWS IAM Identity Center 《使用者指南》中的[設定 AWS 存取入口網站和 IAM Identity Center 整合應用程式的工作階段持續時間](#)。

config 檔案的 `sso-session`區段會用來將用來取得SSO存取權杖的組態變數分組，然後可用來取得AWS 登入資料。如需 config 檔案內本節的詳細資訊，請參閱 [組態檔案格式](#)。

下列共用config檔案範例使用dev設定檔來設定 SDK或 工具，以請求 IAM Identity Center 登入資料。

```
[profile dev]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = SampleRole

[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_registration_scopes = sso:account:access
```

上述範例顯示您定義 `sso-session`區段並將其與設定檔建立關聯。一般而言，`sso_account_id`和`sso_role_name` 必須在 `profile`區段中設定，以便 SDK可以請求 AWS 登入資料。`sso_region`、`sso_start_url`和 `sso_registration_scopes` 必須在 `sso-session`區段中設定。

`sso_account_id` 和 `sso_role_name` 並非權SSO杖組態的所有案例都需要。如果您的應用程式只使用支援承載身分驗證 AWS 服務的，則不需要傳統 AWS 登入資料。無記名身分驗證是一種身分HTTP驗證機制，使用稱為無記名符記的安全符記。在這種情況下，`sso_account_id` 和 `sso_role_name` 並非必要資訊。請參閱個別 AWS 服務 指南，判斷服務是否支援承載字符授權。

註冊範圍會設定為的一部分`sso-session`。Scope 是 中的機制 OAuth 2.0 限制應用程式對使用者帳戶的存取。先前的範例集`sso_registration_scopes`提供列出帳戶和角色的必要存取權。

下列範例示範如何在多個設定檔間重複使用相同的sso-session組態。

```
[profile dev]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = SampleRole

[profile prod]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = SampleRole2

[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_registration_scopes = sso:account:access
```

身分驗證字符會快取至~/.aws/sso/cache目錄下的磁碟，其具有以工作階段名稱為基礎的檔案名稱。

舊版不可重新整理的配置

使用舊版不可重新整理的組態，不支援自動字符重新整理。我們建議您[SSO 權杖提供者組態](#)改為使用。

若要使用舊版不可重新整理組態，您必須在設定檔中指定下列設定：

- sso_start_url
- sso_region
- sso_account_id
- sso_role_name

您可以使用 sso_start_url和 sso_region設定來指定設定檔的使用者入口網站。您可以使用 sso_account_id和 sso_role_name設定指定許可。

下列範例會設定 config 檔案中的四個必要值。

```
[profile my-sso-profile]
sso_start_url = https://my-sso-portal.awsapps.com/start
```

```
sso_region = us-west-2
sso_account_id = 111122223333
sso_role_name = SSOReadOnlyRole
```

身分驗證字符會快取至~/.aws/sso/cache目錄下的磁碟，其具有以為基礎的檔案名稱sso_start_url。

IAM Identity Center 登入資料提供者設定

使用下列各項設定此功能：

sso_start_url - 共用 AWS config檔案設定

URL 指向您組織的 IAM Identity Center 發行者URL或存取入口網站 URL。如需詳細資訊，請參閱AWS IAM Identity Center 《使用者指南》中的[使用 AWS 存取入口網站](#)。

若要尋找此值，請開啟 [IAM Identity Center 主控台](#)、檢視儀表板、尋找AWS 存取入口網站 URL。

- 或者，從 2.22.0 版開始 AWS CLI，您可以改為使用AWS 發行者 URL的值。

sso_region - 共用 AWS config檔案設定

包含 IAM Identity Center 入口網站主機 AWS 區域的，亦即您在啟用 IAM Identity Center 之前選取的區域。這與您的預設 AWS 區域無關，並且可以不同。

如需 AWS 區域 及其代碼的完整清單，請參閱 中的[區域端點](#)Amazon Web Services 一般參考。若要尋找此值，請開啟 [IAM Identity Center 主控台](#)、檢視儀表板和尋找區域。

sso_account_id - 共用 AWS config檔案設定

透過 AWS Organizations 服務 AWS 帳戶 新增用於身分驗證之 的數值 ID。

若要查看可用帳戶的清單，請前往 [IAM Identity Center 主控台](#)並開啟AWS 帳戶頁面。您也可以 在AWS IAM Identity Center 入口網站API參考中查看使用 [ListAccounts](#)API方法的可用帳戶清單。 例如，您可以呼叫 AWS CLI 方法 [list-accounts](#)。

sso_role_name - 共用 AWS config檔案設定

佈建為角色的許可集名稱IAM，定義使用者產生的許可。角色必須存在於 AWS 帳戶 指定的 中sso_account_id。使用角色名稱，而非角色 Amazon Resource Name (ARN)。

許可集已將IAM政策和自訂許可政策連接到他們，並定義使用者對其指派的存取層級 AWS 帳戶。

若要查看每個的可用許可集清單 AWS 帳戶，請前往 [IAM Identity Center 主控台](#) 並開啟 AWS 帳戶頁面。選擇 AWS 帳戶 資料表中列出的正確許可集名稱。您也可以從 AWS IAM Identity Center 入口網站 API 參考中，查看使用 [ListAccountRoles](#) API 方法的可用許可集清單。例如，您可以呼叫 AWS CLI 方法 [list-account-roles](#)。

sso_registration_scopes - 共用 AWS config 檔案設定

要授權給的有效範圍字串逗號分隔清單 sso-session。應用程式可以請求一或多個範圍，而核發給應用程式的存取權杖僅限於授予的範圍。sso:account:access 必須授予的最小範圍，才能從 IAM Identity Center 服務取得重新整理權杖。如需可用存取範圍選項的清單，請參閱 AWS IAM Identity Center 《使用者指南》中的 [存取範圍](#)。

這些範圍會定義請求授權給已註冊 OIDC 用戶端的許可，以及用戶端擷取的存取權杖。範圍授權存取 IAM Identity Center 承載符記授權端點。

此設定不適用於舊版不可重新整理組態。使用舊版組態發行的權杖僅限於 sso:account:access 隱含範圍。

與的相容性 AWS SDKs

以下 SDKs 支援本主題所述的功能和設定。會記下任何部分例外狀況。AWS SDK for Java 和適用於 Kotlin 的 AWS SDK 僅支援任何 JVM 系統屬性設定。

SDK	支援	備註或更多資訊
AWS CLI v2	是	
SDK 適用於 C++	是	
SDK for Go V2 (1.x)	是	
SDK for Go 1.x (V1)	是	若要使用共用 config 檔案設定，您必須開啟從組態檔案載入；請參閱 工作階段 。
SDK 適用於 Java 2.x	是	credentials 檔案中也支援組態值。
SDK 適用於 Java 1.x	否	
SDK for JavaScript 3.x	是	

SDK	支援	備註或更多資訊
SDK for JavaScript 2.x	是	
SDK 適用於 Kotlin	是	
SDK 適用於 。NET 3.x	是	
SDK 適用於 PHP 3.x	是	
SDK for Python (Boto3)	是	
SDK 適用於 Ruby 3.x	是	
SDK for Rust	部分	僅限舊版不可重新整理組態。
SDK 適用於 Swift	是	
適用於的工具 PowerShell	是	

IMDS 憑證提供者

Instance Metadata Service (IMDS) 提供執行個體的資料，您可以用來設定或管理執行中的執行個體。如需可用資料的詳細資訊，請參閱 Amazon EC2 使用者指南 中的 [使用執行個體中繼資料](#)。Amazon EC2 提供可供執行個體使用的本機端點，可為執行個體提供各種位元的資訊。如果執行個體已連接角色，則可以提供一組對該角色有效的憑證。SDKs 可以使用該端點來解析憑證，作為其 [預設憑證提供者鏈](#) 的一部分。預設會使用執行個體中繼資料服務第 2 版 (IMDSv2)，IMDS 這是使用工作階段權杖更安全的版本。如果由於無法重試的條件 (HTTP 錯誤碼 403、404、405) 而失敗，IMDSv1 則會用作退避。

使用下列各項設定此功能：

AWS_EC2_METADATA_DISABLED - 環境變數

是否嘗試使用 Amazon EC2 Instance Metadata Service (IMDS) 來取得憑證。

預設值：false。

有效值：

- **true** – 請勿使用 IMDS 取得憑證。
- **false** – 使用 IMDS 取得憑證。

ec2_metadata_v1_disabled - 共用 **AWS config** 檔案設定, **AWS_EC2_METADATA_V1_DISABLED** - 環境變數, **aws.disableEc2MetadataV1** - JVM系統屬性 : 僅限 Java/Kotlin

如果IMDSv2失敗，是否使用執行個體中繼資料服務第 1 版（IMDSv1）作為後援。

 Note

新 SDKs 不支援IMDSv1，因此不支援此設定。如需詳細資訊，請參閱資料表 [與的相容性 AWS SDKs](#)。

預設值：false。

有效值：

- **true** – 請勿IMDSv1用作退避。
- **false** – IMDSv1用作後援。

ec2_metadata_service_endpoint - 共用 **AWS config** 檔案設定, **AWS_EC2_METADATA_SERVICE_ENDPOINT** - 環境變數, **aws.ec2MetadataServiceEndpoint** - JVM系統屬性：僅限 Java/Kotlin

的端點IMDS。此值會覆寫 `aws.ec2MetadataServiceEndpoint` 和工具將搜尋 AWS SDKsAmazon EC2執行個體中繼資料的預設位置。

預設值：如果 `ec2_metadata_service_endpoint_mode` 等於 IPv4，則預設端點為 `http://169.254.169.254`。如果 `ec2_metadata_service_endpoint_mode` 等於 IPv6，則預設端點為 `http://[fd00:ec2::254]`。

有效值：有效 URI。

ec2_metadata_service_endpoint_mode - 共用 **AWS config** 檔案設定, **AWS_EC2_METADATA_SERVICE_ENDPOINT_MODE** - 環境變數, **aws.ec2MetadataServiceEndpointMode** - JVM系統屬性：僅限 Java/Kotlin

的端點模式IMDS。

預設值：IPv4。

有效值：IPv4、IPv6。

Note

IMDS 憑證提供者是 的一部分 [了解憑證提供者鏈](#)。不過，只有在此系列中的其他幾個提供者之後，才會檢查IMDS憑證提供者。因此，如果您希望程式使用此提供者的憑證，您必須從組態中移除其他有效的憑證提供者，或使用不同的設定檔。或者，不要依賴憑證提供者鏈自動探索哪個提供者傳回有效的憑證，而是在程式碼中指定IMDS憑證提供者的使用。您可以在建立服務用戶端時直接指定憑證來源。

IMDS 憑證的安全性

根據預設，當 AWS SDK未設定有效憑證時，SDK會嘗試使用 Amazon EC2 Instance Metadata Service (IMDS) 來擷取 AWS 角色的憑證。將AWS_EC2_METADATA_DISABLED環境變數設定為 `true`，即可停用此行為。這可防止不必要的網路活動，並增強可能模擬 Amazon EC2 Instance Metadata Service 的不受信任網路的安全性。

Note

AWS SDK 設定有效憑證的用戶端永遠不會用來IMDS擷取憑證，無論這些設定為何。

停用 Amazon EC2IMDS憑證

設定此環境變數的方式取決於正在使用的作業系統，以及您是否希望變更持續存在。

Linux 和 macOS

使用 Linux 或 macOS 的客戶可以使用下列命令來設定此環境變數：

```
$ export AWS_EC2_METADATA_DISABLED=true
```

如果您希望此設定在多個 Shell 工作階段和系統重新啟動之間持續存在，您可以將上述命令新增至 Shell 設定檔檔案，例如 `.bash_profile`、`.zsh_profile` 或 `.profile`。

Windows

使用 Windows 的客戶可以使用下列命令來設定此環境變數：

```
$ set AWS_EC2_METADATA_DISABLED=true
```

如果您希望此設定在多個 Shell 工作階段和系統重新啟動之間持續存在，則可以改用下列命令：

```
$ setx AWS_EC2_METADATA_DISABLED=true
```

Note

setx 命令不會將值套用至目前的 Shell 工作階段，因此您需要重新載入或重新開啟 Shell，變更才會生效。

與 的相容性 AWS SDKs

以下 SDKs 支援本主題所述的功能和設定。會記下任何部分例外狀況。AWS SDK for Java 和 適用於 Kotlin 的 AWS SDK 僅支援任何 JVM 系統屬性設定。

SDK	支 備註或更多資訊 援
AWS CLI v2	是
SDK 適用於 C++	是
SDK for Go V2 (1.x)	是
SDK for Go 1.x (V1)	是 若要使用共用 config 檔案設定，您必須開啟從組態檔案載入；請參閱 工作階段 。
SDK 適用於 Java 2.x	是
SDK 適用於 Java 1.x	部 JVM 系統屬性：使用 com.amazonaws.sdk.disableEc2MetadataV1 而非 aws.disableEc2MetadataV1；aws.ec2MetadataServiceEndpointMode 不支援和。
SDK 適用於 JavaScript 3.x	是

SDK	支援	備註或更多資訊
SDK 適用於 JavaScript 2.x	是	
SDK 適用於 Kotlin	是	不使用退IMDSv1避。
SDK 適用於 。NET 3.x	是	
SDK 適用於 PHP 3.x	是	
SDK for Python (Boto3)	是	
SDK 適用於 Ruby 3.x	是	
SDK for Rust	是	不使用退IMDSv1避。
SDK 適用於 Swift	是	
適用於 的工具 PowerShell	是	您可以使用 明確停用程式碼中的退IMDSv1信[Amazon.Util.EC2InstanceMetadata]::EC2MetadataV1Disabled = \$true 。

程序認證提供者

SDKs為自訂使用案例提供延伸憑證提供者鏈結的方法。此提供者可用來提供自訂實作，例如從內部部署認證存放區擷取認證，或與您的內部部署識別提供者整合。

例如，「任何IAM角色」用credential_process於代表您的應用程式取得臨時登入資料。若要credential_process針對此用途進行配置，請參閱[IAM Roles Anywhere](#)。

Note

以下說明從外部處理程序取得認證的方法，如果您在以外的地方執行軟體，則可能會使用此方法。AWS。如果您正在建立 AWS 計算資源，使用其他認證提供者。如果使用此選項，您應該使用適用於您的作業系統的安全性最佳實務來確保組態檔已盡可能鎖定。確認您的自訂認證工具不會寫入任何秘密資訊StdErr，因為SDKs和 AWS CLI 可以擷取並記錄此類資訊，進而可能會將其暴露給未經授權的使用者。

使用下列項目設定此功能：

credential_process-共享 AWS config 檔案設定

指定SDK或工具代表您執行的外部命令，以產生或擷取要使用的驗證認證。此設定SDK會指定將呼叫的程式/指令的名稱。當SDK調用該進程時，它會等待進程將JSON數據寫入。stdout自訂提供者必須以特定格式傳回資訊。該資訊包含SDK或工具可用來驗證您的認證。

Note

處理程序認證提供者是的一部分[了解憑證提供者鏈](#)。但是，只有在本系列中的其他幾個提供者之後才會檢查進程憑證提供者。因此，如果您希望程式使用此提供者的認證，您必須從組態中移除其他有效的認證提供者，或使用不同的設定檔。或者，不要依賴憑證提供者鏈來自動探索哪個提供者傳回有效的認證，而是在程式碼中指定使用處理序憑證提供者。您可以在建立服務用戶端時直接指定認證來源。

指定認證程式的路徑

設定的值是字串，其中包含SDK或開發工具代表您執行的程式路徑：

- 路徑和檔案名稱只能由下列字元組成：A-Z、a-z、0-9、連字號 (-)、底線 (_)、句點 (.)、正斜線 (/)、反斜線 (\) 和空格。
- 如果路徑或檔案名稱包含空格，完整的路徑和檔案名稱請以雙引號 (「」) 括住。
- 如果參數名稱或參數值包含空格，則該元素請以雙引號 (「」) 括住。僅括住名稱或值，而非對組。
- 請勿在字串中包含任何環境變數。例如，請勿包含\$HOME或%USERPROFILE%。
- 請勿將主資料夾指定為~。* 您必須指定完整路徑或基底檔案名稱。如果有基底檔案名稱，系統會嘗試在PATH環境變數指定的資料夾內尋找程式。根據作業系統，路徑會有所不同：

以下示例顯示了在 Linux/macOS 上的共享config文件中設置憑據 _ 進程。

```
credential_process = "/path/to/credentials.sh" parameterWithoutSpaces "parameter with spaces"
```

下列範例顯示在 Windows 上的共用config檔案中設定認證程序。

```
credential_process = "C:\Path\To\credentials.cmd" parameterWithoutSpaces "parameter with spaces"
```

- 可以在專用設定檔中指定：

```
[profile cred_process]
credential_process = /Users/username/process.sh
region = us-east-1
```

憑證程式的有效輸出

會依照設定檔中指定的方式SDK執行命令，然後從標準輸出串流讀取資料。您指定的命令 (無論是指令碼還是二進位程式) 都必須產生STDOUT符合下列語法的JSON輸出。

```
{
  "Version": 1,
  "AccessKeyId": "an AWS access key",
  "SecretAccessKey": "your AWS secret access key",
  "SessionToken": "the AWS session token for temporary credentials",
  "Expiration": "RFC3339 timestamp for when the credentials expire"
}
```

Note

截至本文編寫時，Version 索引鍵必須設定為 1。這可能隨著結構演進而逐漸遞增。

Expiration 金鑰是RFC3339格式化的時間戳記。如果Expiration金鑰不存在於工具的輸出中，則SDK假設認證是不重新整理的長期認證。否則，認證會被視為暫時證明資料，而且會在認證到期之前重新執行credential_process命令來自動重新整理這些證明資料。

Note

SDK不會像執行假設角色認證的方式一樣快取外部處理程序認證。如果需要快取，您必須在外部程序中實作它。

外部程序可能傳回非零傳回碼，以表示擷取憑證時發生錯誤。

兼容性 AWS SDKs

下列項目 SDKs 支援本主題中描述的功能和設定。注意到任何部分例外情況。任何 JVM 系統屬性設定都受到 AWS SDK for Java 和 適用於 Kotlin 的 AWS SDK 只有。

SDK	支援	備註或更多資訊
AWS CLI V2	是	
SDK 對於 C ++	是	
SDK 對於轉到 V2 (1.x)	是	
SDK 對於圍棋 1.x (V1)	是	若要使用共用 config 檔案設定，您必須開啟從設定檔載入功能；請參閱 工作階段 。
SDK 對於爪哇 2.x	是	
SDK 對於爪哇 1.x	是	
SDK 對於 JavaScript 3. x	是	
SDK 對於 JavaScript 2.x	是	
SDK 對於科特林	是	
SDK 對於 .NET 3.x	是	
SDK 對於 PHP 3. x	是	
SDK 對於 Python (肉毒桿菌 3)	是	
SDK 對於紅寶石 3.x	是	
SDK 對於銹	是	
SDK 為斯威夫特	是	
適用的工具 PowerShell	是	

AWS SDKs和工具標準化功能

許多功能已標準化為一致的預設值，並且在許多功能中都以相同的方式運作SDKs。這種一致性可以提高多個編碼時的生產力和清晰度SDKs。所有設置都可以在代碼中覆蓋，請參閱您的具體以獲取 SDKAPI詳細信息。

Important

並非所有功能都SDKs支持所有功能，甚至功能中的所有方面都支持。

主題

- [帳戶型端點](#)
- [申請識別碼](#)
- [Amazon EC2 執行個體](#)
- [Amazon S3 存取點](#)
- [Amazon S3 多區域存取點](#)
- [AWS 區域](#)
- [AWS STS 區域端點](#)
- [雙堆疊與端點 FIPS](#)
- [端點探索](#)
- [一般組態設定](#)
- [IMDS 用戶端](#)
- [重試行為](#)
- [請求壓縮](#)
- [服務特定端點](#)
- [智慧型組態預設](#)

帳戶型端點

帳戶型端點使用您的 AWS 帳戶 ID 簡化支援此功能之服務的 AWS 服務 請求路由，有助於確保高效能和可擴展性。當您使用支援以帳戶為基礎的端點的 AWS SDK、憑證提供者和服務時，SDK會自動建構和使用以帳戶為基礎的端點，而不是區域性端點。以帳戶為基礎的端點採用的形

式 `https://<account-id>.ddb.<region>.amazonaws.com`，其中 `<account-id>` 取代為您的 AWS 帳戶 ID，而 `<region>` 取代為您的 AWS 區域

根據預設，帳戶 ID 會在處理請求並用於建構端點時收集。處理請求時，也會發生憑證解析，並可能會變更端點解析的方法。根據您使用的憑證提供者，帳戶 ID 的來源可能不同。

使用下列各項設定此功能：

aws_account_id - 共用 AWS **config** 檔案設定, **AWS_ACCOUNT_ID** - 環境變數, **aws.accountId** - JVM系統屬性：僅限 Java/Kotlin

AWS 帳戶 ID。用於帳戶型端點路由。AWS 帳戶 ID 的格式如 111122223333。

帳戶型端點路由可為某些 服務提供更好的請求效能。

account_id_endpoint_mode - 共用 AWS **config** 檔案設定, **AWS_ACCOUNT_ID_ENDPOINT_MODE** - 環境變數, **aws.accountIdEndpointMode** - JVM系統屬性：僅限 Java/Kotlin

此設定用於在必要時關閉以帳戶為基礎的端點路由，並繞過以帳戶為基礎的規則。

預設值：preferred

有效值：

- **preferred** – 端點應包含可用的帳戶 ID。
- **disabled** – 已解析的端點不包含帳戶 ID。
- **required** – 端點必須包含帳戶 ID。如果帳戶 ID 無法使用，會 SDK 擲回錯誤。

與 的相容性 AWS SDKs

以下內容 SDKs 支援本主題所述的功能和設定。會記下任何部分例外狀況。AWS SDK for Java 和 適用於 Kotlin 的 AWS SDK 僅支援任何 JVM 系統屬性設定。

SDK	支援	在 SDK 版本中發行	備註或更多資訊
AWS CLI v2	否		
SDK 適用於 C++	否		

SDK	支援	在SDK版本中發行	備註或更多資訊
SDK for Go V2 (1.x)	是	1.35.0 版	
SDK for Go 1.x (V1)	否		
SDK 適用於 Java 2.x	是	v2.28.4	
SDK 適用於 Java 1.x	是	v1.12.771	
SDK 適用於 JavaScript 3.x	是	v3.656.0	
SDK 適用於 JavaScript 2.x	否		
SDK 適用於 Kotlin	是	v1.3.37	
SDK 適用於 。NET 3.x	否		
SDK 適用於 PHP 3.x	是	v3.318.0	
SDK for Python (Boto3)	否		
SDK 適用於 Ruby 3.x	是	v1.123.0	
SDK for Rust	否		
SDK 適用於 Swift	否		
適用於 的工具 PowerShell	否		

申請識別碼

單一 AWS 帳戶 可供多個客戶應用程式用來撥打電話給 AWS 服務。應用程式 ID 可讓客戶識別哪個來源應用程式使用 AWS 帳戶。AWS SDKs和服務不會使用或解釋這個值，除了在客戶通信中表現它。例如，此值可以包含在操作電子郵件中或 AWS Health Dashboard 以唯一識別哪些應用程式與通知相關聯。

使用下列項目設定此功能：

sdk_ua_app_id-共享 AWS config檔案設定, **AWS_SDK_UA_APP_ID**-環境變量,
aws.userAgentAppId-JVM 系統屬性：僅限爪哇/科特林

此設定是您指派給應用程式的唯一字串，以識別特定應用程式中的哪些應用程式 AWS 帳戶 撥打電話 AWS。

預設值：None

有效值：最大長度為 50 的字串。允許使用字母、數字和下列特殊字元：!\$%、&、*、+、-、.、/、^、_、`、|、~。

在config檔案中設定此值的範例：

```
[default]
sdk_ua_app_id=ABCDEF
```

通過命令行設置環境變量的 Linux/macOS 示例：

```
export AWS_SDK_UA_APP_ID=ABCDEF
export AWS_SDK_UA_APP_ID="ABC DEF"
```

通過命令行設置環境變量的 Windows 示例：

```
setx AWS_SDK_UA_APP_ID ABCDEF
setx AWS_SDK_UA_APP_ID="ABC DEF"
```

如果您包含對所使用的 shell 具有特殊意義的符號，請適當地逸出該值。

兼容性 AWS SDKs

下列項目 SDKs 支援本主題中描述的功能與設定。注意到任何部分例外情況。任何 JVM 系統屬性設定都受到 AWS SDK for Java 和 適用於 Kotlin 的 AWS SDK 只有。

SDK	支援	備註或更多資訊
AWS CLI V2	是	
SDK對於 C ++	是	不支援共用config檔案。
SDK對於轉到 V2 (1.x)	是	
SDK對於圍棋 1.x (V1)	否	
SDK對於爪哇 2.x	部分	不支援共用config檔案設定；不支援環境變數。
SDK對於爪哇 1.x	否	
SDK對於 JavaScript 3. x	是	
SDK對於 JavaScript 2.x	否	
SDK對於科特林	是	
SDK為 .NET3.x	是	不支援環境變數。
SDK對於 PHP 3. x	是	
SDK對於 Python (肉毒桿菌 3)	是	
SDK對於紅寶石 3.x	是	
SDK對於鏽	是	
SDK為斯威夫特	否	
用於的工具 PowerShell	否	

Amazon EC2 執行個體

Amazon 在稱為執行個體中繼資料服務 (IMDS) 的執行個體上 EC2 提供服務。若要進一步了解此服務，請參閱 Amazon EC2 使用者指南中的使用 [執行個體中繼資料](#)。嘗試在已設定 IAM 角色的 Amazon EC2 執行個體上擷取登入資料時，與執行個體中繼資料服務的連線可調整。

使用下列項目設定此功能：

metadata_service_num_attempts-共享 AWS config 檔案設定，
AWS_METADATA_SERVICE_NUM_ATTEMPTS-環境變量

此設定指定嘗試從執行個體中繼資料服務擷取資料時，放棄之前要嘗試的總嘗試次數。

預設值：1

有效值：大於或等於 1 的數字。

metadata_service_timeout-共享 AWS config 檔案設定，**AWS_METADATA_SERVICE_TIMEOUT**-
環境變量

指定嘗試從執行個體中繼資料服務擷取資料時，逾時前的秒數。

預設值：1

有效值：大於或等於 1 的數字。

在檔案中設定這些值 config 的範例：

```
[default]
metadata_service_num_attempts=10
metadata_service_timeout=10
```

通過命令行設置環境變量的 Linux/macOS 示例：

```
export AWS_METADATA_SERVICE_NUM_ATTEMPTS=10
export AWS_METADATA_SERVICE_TIMEOUT=10
```

通過命令行設置環境變量的 Windows 示例：

```
setx AWS_METADATA_SERVICE_NUM_ATTEMPTS 10
setx AWS_METADATA_SERVICE_TIMEOUT 10
```

兼容性 AWS SDKs

下列項目 SDKs 支援本主題中描述的功能和設定。注意到任何部分例外情況。任何 JVM 系統屬性設定都受到 AWS SDK for Java 和 適用於 Kotlin 的 AWS SDK 只有。

SDK	支援	備註或更多資訊
AWS CLI V2	是	
SDK 對於 C ++	否	
SDK 對於轉到 V2 (1.x)	否	
SDK 對於圍棋 1.x (V1)	否	
SDK 對於爪哇 2.x	否	
SDK 對於爪哇 1.x	部分	僅支援 <code>AWS_METADATA_SERVICE_TIMEOUT</code> 。
SDK 對於 JavaScript 3. x	否	
SDK 對於 JavaScript 2.x	否	
SDK 對於科特林	否	
SDK 為 .NET 3.x	否	
SDK 對於 PHP 3. x	是	
SDK 對於 Python (肉毒桿菌 3)	是	
SDK 對於紅寶石 3.x	否	
SDK 對於銹	否	
SDK 為斯威夫特	否	
用於的工具 PowerShell	否	

Amazon S3 存取點

Amazon S3 服務提供存取點作為與 Amazon S3 儲存貯體互動的替代方式。存取點具有唯一的原則和設定，可套用至存取點，而不是直接套用至值區。同 AWS SDKs，您可以在值區欄位中使用存取點 Amazon 資源名稱 (ARNs) 進行 API 操作，而不是明確指定值區名稱。它們可用於特定作業，例如使用 [GetObject](#) 存取點 ARN 搭配從值區擷取物件，或使用存取點 ARN 與將物件新增 [PutObject](#) 至值區。

若要進一步了解 Amazon S3 存取點 ARNs，請參閱 Amazon S3 使用者指南 [中的使用存取點](#)。

使用下列項目設定此功能：

s3_use_arn_region-共享 AWS config 檔案設定, **AWS_S3_USE_ARN_REGION**-環境變量, **aws.s3UseArnRegion**-JVM 系統屬性：僅限爪哇/科特林, 要直接在代碼中配置值，請直接諮詢您的具體 SDK 內容。

此設定控制是否 SDK 使用存取點 ARN AWS 區域 建構要求的區域端點。該 SDK 驗證 ARN AWS 區域 由相同服務 AWS 分區作為客戶端的配置 AWS 區域 以防止最有可能失敗的跨分區呼叫。如果定義了乘數，則代碼配置的設置優先，然後是環境變量設置。

預設值：false

有效值：

- **true**— SDK 使用 ARN 的 AWS 區域 構建端點而不是客戶端配置時 AWS 區域。例外：如果用戶端已設定 AWS 區域 是一個 FIPS AWS 區域，那麼它必須匹配 ARN 的 AWS 區域。否則，將導致錯誤。
- **false**— SDK 使用客戶端的配置 AWS 區域 構建端點時。

兼容性 AWS SDKs

下列項目 SDKs 支援本主題中描述的功能與設定。注意到任何部分例外情況。任何 JVM 系統屬性設定都受到 AWS SDK for Java 和 適用於 Kotlin 的 AWS SDK 只有。

SDK	支 備註或更多資訊 援
AWS CLI V2	是
SDK 對於 C ++	是

SDK	支援	備註或更多資訊
SDK對於轉到 V2 (1.x)	是	
SDK對於圍棋 1.x (V1)	是	若要使用共用config檔案設定，您必須開啟從設定檔載入功能；請參閱 工作階段 。
SDK對於爪哇 2.x	是	
SDK對於爪哇 1.x	是	JVM系統屬性不受支援。
SDK對於 JavaScript 3. x	是	
SDK對於 JavaScript 2.x	是	
SDK對於科特林	是	
SDK為 .NET 3.x	是	不遵循標準優先順序；共享config檔案值優先於環境變數。
SDK對於 PHP 3. x	是	
SDK對於 Python (肉毒桿菌 3)	是	
SDK對於紅寶石 3.x	是	
SDK對於鏽	否	
SDK為斯威夫特	否	
用於的工具 PowerShell	是	不遵循標準優先順序；共享config檔案值優先於環境變數。

Amazon S3 多區域存取點

Amazon S3 多區域存取點提供了一個全球端點，應用程式可用來滿足來自多個 Amazon S3 儲存貯體的請求 AWS 區域。您可以使用多區域存取點，使用單一區域所使用的相同架構來建置多區域應用程式，然後在世界任何地方執行這些應用程式。

若要進一步了解多區域存取點，請參閱 Amazon S3 使用者[指南中的 Amazon S3 中的多區域存取點](#)。

若要進一步了解多區域存取點 Amazon 資源名稱 (ARNs)，請參閱 [Amazon S3 使用者指南中的使用多區域存取點發出請求](#)。

若要進一步了解如何建立多區域存取點，請參閱 Amazon S3 使用者指南中的 [管理多區域存取點](#)。

SigV4a 演算法是用來簽署全域區域要求的簽署實作。此演算法是 SDK 透過對 [AWS 一般執行期 \(CRT\) 程式庫](#)。

使用下列項目設定此功能：

s3_disable_multiregion_access_points-共享 AWS config 檔

案設定, **AWS_S3_DISABLE_MULTIREGION_ACCESS_POINTS**-環境變量,

aws.s3DisableMultiRegionAccessPoints-JVM 系統屬性：僅限爪哇/科特林, 要直接在代碼中配置值，請直接諮詢您的具體 SDK 內容。

此設定可控制是否 SDK 可能嘗試跨區域要求。如果定義了乘數，則代碼配置的設置優先，然後是環境變量設置。

預設值：false

有效值：

- **true**— 停止使用跨區域請求。
- **false**— 使用多區域存取點啟用跨區域要求。

兼容性 AWS SDKs

下列項目 SDKs 支援本主題中描述的功能和設定。注意到任何部分例外情況。任何 JVM 系統屬性設定都受到 AWS SDK for Java 和 適用於 Kotlin 的 AWS SDK 只有。

SDK	支 備註或更多資訊 援
AWS CLI V2	是
SDK 對於 C ++	是
SDK 對於轉到 V2 (1.x)	是
SDK 對於圍棋 1.x (V1)	否

SDK	支援	備註或更多資訊
SDK對於爪哇 2.x	是	
SDK對於爪哇 1.x	否	
SDK對於 JavaScript 3. x	是	
SDK對於 JavaScript 2.x	否	
SDK對於科特林	是	
SDK對於 .NET3.x	是	
SDK對於 PHP 3. x	是	
SDK對於 Python (肉毒桿菌 3)	是	
SDK對於紅寶石 3.x	是	
SDK對於銹	是	
SDK為斯威夫特	否	
適用的工具 PowerShell	是	

AWS 區域

AWS 區域 是使用 時需要了解的重要概念 AWS 服務。

透過 AWS 區域，您可以存取實際位於特定地理區域 AWS 服務的。這對於讓您的資料和應用程式在接近您和您的使用者將存取它們的位置時保持執行很有用。區域提供容錯能力、穩定性和恢復能力，也可降低延遲。使用 區域，您可以建立備援資源，以保持可用且不受區域中斷影響。

大多數 AWS 服務 請求都與特定地理區域相關聯。除非您明確使用 提供的複寫功能，否則您在一個 區域中建立的資源不存在於任何其他 區域 AWS 服務。例如，Amazon S3 和 Amazon EC2支援跨區域複寫。有些服務，例如 IAM，沒有區域資源。

AWS 一般參考 包含下列資訊：

- 若要了解區域和端點之間的關係，以及檢視現有區域端點的清單，請參閱[AWS 服務端點](#)。
- 若要檢視每個 所有支援的區域和端點的目前清單 AWS 服務，請參閱[服務端點和配額](#)。

建立服務用戶端

若要以程式設計方式存取 AWS 服務，請為每個 SDKs使用用戶端類別/物件 AWS 服務。例如EC2，如果您的應用程式需要存取 Amazon，您的應用程式會建立 Amazon EC2用戶端物件以與該服務進行界面。

如果程式碼本身未明確指定用戶端的區域，則用戶端會使用透過下列設定設定的區域，預設為 region。不過，可以針對任何個別用戶端物件明確設定用戶端的作用中區域。以此方式設定區域優先於該特定服務用戶端的任何全域設定。替代區域是在該用戶端的實例化期間指定，具體針對您的 SDK (請查看您的特定SDK指南或 SDK的程式碼基礎)。

使用下列項目設定此功能：

region - 共用 AWS **config**檔案設定, **AWS_REGION** - 環境變數, **aws.region** - JVM系統屬性：僅限 Java/Kotlin

指定 AWS 區域 要用於 AWS 請求的預設值。此區域用於未隨附要使用的特定區域SDK的服務請求。

預設值：無。您必須明確指定此值。

有效值：

- 所選服務可用的任何區域代碼，如AWS 一般參考 中的[AWS 服務端點](#)中所列。例如，值us-east-1會將端點設定為 AWS 區域 美國東部 (維吉尼亞北部)。
- aws-global 指定除了區域端點之外，支援個別全域端點的服務全域端點，例如 AWS Security Token Service (AWS STS) 和 Amazon Simple Storage Service (Amazon S3)。

在 config 檔案中設定此值的範例：

```
[default]
region = us-west-2
```

透過命令列設定環境變數的 Linux/macOS 範例：

```
export AWS_REGION=us-west-2
```

透過命令列設定環境變數的 Windows 範例：

```
setx AWS_REGION us-west-2
```

大多數 SDKs 都具有「組態」物件，可用於從應用程式程式碼內設定預設區域。如需詳細資訊，請參閱您的特定 AWS SDK 開發人員指南。

與的相容性 AWS SDKs

以下 SDKs 支援本主題所述的功能和設定。會記下任何部分例外狀況。AWS SDK for Java 和 適用於 Kotlin 的 AWS SDK 僅支援任何 JVM 系統屬性設定。

SDK	支援	備註或更多資訊
AWS CLI v2	是	AWS CLI v2 在 中的任何值 <code>AWS_REGION</code> 之前，會先使用 中的任何值 <code>AWS_DEFAULT_REGION</code> （會檢查兩個變數）。
AWS CLI v1	是	AWS CLI v1 <code>AWS_DEFAULT_REGION</code> 會為此目的使用名為的環境變數。
SDK 適用於 C++	是	
SDK for Go V2 (1.x)	是	
SDK for Go 1.x (V1)	是	若要使用共用 config 檔案設定，您必須開啟從組態檔案載入；請參閱 工作階段 。
SDK 適用於 Java 2.x	是	
SDK 適用於 Java 1.x	是	
SDK 適用於 JavaScript 3.x	是	
SDK 適用於 JavaScript 2.x	是	
SDK 適用於 Kotlin	是	
SDK 適用於 。NET 3.x	是	
SDK 適用於 PHP 3.x	是	

SDK	支援	備註或更多資訊
SDK for Python (Boto3)	是	這 SDK 會使用名為 <code>AWS_DEFAULT_REGION</code> 的環境變數來達成此目的。
SDK 適用於 Ruby 3.x	是	
SDK for Rust	是	
SDK 適用於 Swift	是	
適用於的工具 PowerShell	是	

AWS STS 區域端點

AWS Security Token Service (AWS STS) 提供全球和區域服務。有些 AWS SDKs 和 依預設 CLIs 使用全域服務端點 (<https://sts.amazonaws.com>)，有些 則使用區域服務端點 (https://sts.{region_identifier}.{partition_domain})。全域請求對應至美國東部 (維吉尼亞北部) 區域。如需 AWS STS 端點的詳細資訊，請參閱 AWS Security Token Service API 參考 中的 [端點](#)。或者，您也可以在使用者指南 [AWS STS 中的 中學習管理 AWS 區域](#)。AWS Identity and Access Management

AWS 最佳實務是盡可能使用區域端點並設定您的 [AWS 區域](#)。商業以外 [分割區](#) 中的客戶必須使用區域端點。並非所有 SDKs 和 工具都支援此設定，但所有 都已定義全球和區域端點的行為。如需詳細資訊，請參閱下一節。

對於支援此設定的 SDKs 和 工具，客戶可以使用下列方式設定 功能：

sts_regional_endpoints - 共用 AWS **config** 檔案設定, **AWS_STS_REGIONAL_ENDPOINTS** - 環境變數

此設定指定 SDK 或 工具如何決定 AWS 服務 用來與 AWS Security Token Service () 交談的端點 AWS STS。

預設值：legacy

Note

2022 年 7 月之後發行的所有新 SDK 主要版本都將預設為 `regional`。新的 SDK 主要版本可能會移除此設定並使用 `regional` 行為。為了減少此變更的未來影響，我們建議您盡可能 `regional` 在應用程式中開始使用。

有效值：（建議值：`regional`）

- **legacy** – 使用全域 AWS STS 端點 `sts.amazonaws.com`。
- **regional** – SDK 或 工具一律使用目前設定區域的 AWS STS 端點。例如，如果用戶端設定為使用 `us-west-2`，則的所有呼叫 AWS STS 都會對區域端點進行 `sts.us-west-2.amazonaws.com`，而不是全域 `sts.amazonaws.com` 端點。若要在啟用此設定的同時，將請求傳送至全域端點，您可以將區域設為 `aws-global`。

在 `config` 檔案中設定這些值的範例：

```
[default]
sts_regional_endpoints = regional
```

透過命令列設定環境變數的 Linux/macOS 範例：

```
export AWS_STS_REGIONAL_ENDPOINTS=regional
```

透過命令列設定環境變數的 Windows 範例：

```
setx AWS_STS_REGIONAL_ENDPOINTS regional
```

與的相容性 AWS SDKs

Note

AWS 最佳實務是盡可能使用區域端點並設定您的 [AWS 區域](#)。

以下資料表摘要說明 SDK 或 工具：

- **支援設定**：是否支援 STS 區域端點的共用 `config` 檔案變數和環境變數。

- 預設設定值：如果支援，則設定的預設值。
- 預設服務用戶端目標STS端點：用戶端使用什麼預設端點，即使無法進行變更設定。
- 服務用戶端退避行為：當應該使用區域端點，但未設定區域時，會SDK做什麼。這是行為，無論其是否因為預設或設定regional已選取，而使用區域端點。

資料表也會使用下列值：

- 全域端點：<https://sts.amazonaws.com>。
- 區域端點：根據您應用程式[AWS 區域](#)所使用的設定。
- **us-east-1** (區域)：使用us-east-1區域端點，但具有比一般全域請求更長的工作階段權杖。

SDK	預設設定值	預設服務用戶端目標STS端點	服務用戶端退避行為	備註或更多資訊
AWS CLI v2	否 N/A	區域 (Region) 端點	全域端點	
AWS CLI v1	是 legacy	全域端點	全域端點	
SDK 適用於 C++	否 N/A	區域 (Region) 端點	us-east-1 (區域)	
SDK for Go V2 (1.x)	否 N/A	區域 (Region) 端點	請求失敗	
SDK for Go 1.x (V1)	是 legacy	全域端點	全域端點	若要使用共用config檔案設定，您必須開啟從組態檔案載入；請參閱 工作階段 。
SDK 適用於 Java 2.x	否 N/A	區域 (Region) 端點	請求失敗	如果未設定區域，AssumeRole 和 AssumeRol

SDK	預設設定值	預設服務用戶端目標STS端點	服務用戶端退信行為	備註或更多資訊
				eWithWebIdentity 將使用全域STS端點
SDK 適用於 Java 1.x	是 legacy	全域端點	全域端點	
SDK 適用於 JavaScript 3.x	否 N/A	區域 (Region) 端點	請求失敗	
SDK 適用於 JavaScript 2.x	是 legacy	全域端點	全域端點	
SDK 適用於 Kotlin	否 N/A	區域 (Region) 端點	全域端點	
SDK 適用於 .NET 3.x	是 legacy	全域端點	全域端點	
SDK 適用於 PHP 3.x	是 legacy	全域端點	請求失敗	
SDK for Python (Boto3)	是 legacy	全域端點	全域端點	
SDK 適用於 Ruby 3.x	是 regional	區域 (Region) 端點	請求失敗	

SDK	預設設定值	預設服務用戶端目標STS端點	服務用戶端退信行為	備註或更多資訊
SDK for Rust	否 N/A	區域 (Region) 端點	請求失敗	
SDK 適用於 Swift	否 N/A	區域 (Region) 端點	請求失敗	
適用於的工具 PowerShell	是 legacy	全域端點	全域端點	

雙堆疊與端點 FIPS

使用下列項目設定此功能：

use_dualstack_endpoint-共享 AWS `config` 檔案設定, **AWS_USE_DUALSTACK_ENDPOINT**-環境變量, **aws.useDualstackEndpoint**-JVM 系統屬性：僅限爪哇/科特林

開啟或關閉是否 SDK 將要求傳送至雙堆疊端點。若要進一步了解支援 IPv4 和 IPv6 流量的雙堆疊端點，請參閱 [Amazon 簡單儲存服務使用者指南中的使用 Amazon S3 雙堆疊端點](#)。雙堆疊端點適用於部分區域的某些服務。

預設值：false

有效值：

- **true**— SDK 或工具會嘗試使用雙堆疊端點提出網路要求。如果服務和/或不存在雙堆疊端點 AWS 區域，請求將失敗。
- **false**— SDK 或工具不會使用雙堆疊端點發出網路要求。

use_fips_endpoint-共享 AWS **config**檔案設定, **AWS_USE_FIPS_ENDPOINT**-環境變量,
aws.useFipsEndpoint-JVM 系統屬性：僅限爪哇/科特林

開啟或關閉SDK或工具是否會傳送要求至FIPS符合標準的端點。聯邦資訊處理標準 (FIPS) 是一組美國政府對資料及其加密的安全要求。政府機構, 合作夥伴, 和那些想要與聯邦政府做生意的人都必須遵守FIPS準則。不同於標準 AWS 端點, FIPS端點使用符合 FIPS 140-2 的TLS軟體程式庫。如果已啟用此設定, 且您的服務中不存在FIPS端點 AWS 區域, 該 AWS 呼叫可能會失敗。 [服務特定端點](#)和 `--endpoint-url`選項 AWS Command Line Interface 覆寫此設定。

若要深入了解透過以下方式指定FIPS端點的其他方法 AWS 區域, 請參閱[依服務區分的FIPS端點](#)。如需 Amazon 彈性運算雲端服務端點的詳細資訊, 請參閱 Amazon EC2 API 參考中的[雙堆疊 \(IPv4 和IPv6\) 端點](#)。

預設值：false

有效值：

- **true**— SDK 或工具會將要求傳送至FIPS符合標準的端點。
- **false**— SDK 或工具不會將要求傳送至FIPS符合標準的端點。

兼容性 AWS SDKs

下列項目SDKs支援本主題中描述的功能和設定。注意到任何部分例外情況。任何JVM系統屬性設定都受到 AWS SDK for Java 和 適用於 Kotlin 的 AWS SDK 只有。

SDK	支援	備註或更多資訊
AWS CLI V2	是	
SDK對於 C ++	是	
SDK對於轉到 V2 (1.x)	是	
SDK對於圍棋 1.x (V1)	是	若要使用共用config檔案設定, 您必須開啟從設定檔載入功能; 請參閱 工作階段 。
SDK對於爪哇 2.x	是	
SDK對於爪哇 1.x	否	

SDK	支援	備註或更多資訊
SDK對於 JavaScript 3. x	是	
SDK對於 JavaScript 2.x	是	
SDK對於科特林	是	
SDK對於 .NET3.x	是	
SDK對於 PHP 3. x	是	
SDK對於 Python (肉毒桿菌 3)	是	
SDK對於紅寶石 3.x	是	
SDK對於銹	是	
SDK為斯威夫特	是	
適用的工具 PowerShell	是	

端點探索

SDKs使用端點探索存取服務端點 (URLs以存取各種資源)，同時仍保持以下項目的彈性 AWS 根據需要URLs進行更改。這樣，您的代碼可以自動檢測新的端點。某些服務沒有固定端點。相反地，您可以在執行階段期間透過提出要求先取得端點來取得可用端點。擷取可用端點後，程式碼會使用端點存取其他作業。例如，對於 Amazon Timestream，會提出SDK擷取可用端點的DescribeEndpoints請求，然後使用這些端點完成特定操作，例如CreateDatabase或CreateTable。

使用下列項目設定此功能：

endpoint_discovery_enabled-共享 AWS config檔案設定，
AWS_ENABLE_ENDPOINT_DISCOVERY-環境變量，**aws.endpointDiscoveryEnabled**-JVM 系統屬性：僅限爪哇/科特林，要直接在代碼中配置值，請直接諮詢您的具體SDK內容。

開啟或關閉 DynamoDB 的端點探索。

在時間流中需要端點探索，而在 Amazon DynamoDB 中則是選用的。此設定預設為 `true` 或 `false` 視服務是否需要端點探索而定。時間串流請求預設為 `true`，而 Amazon DynamoDB 請求預設為 `false`。

有效值：

- **true**— SDK 應該會自動嘗試探索端點為選用的服務探索端點。
- **false**— 不 SDK 應自動嘗試探索端點為選用的服務探索端點。

兼容性 AWS SDKs

下列項目 SDKs 支援本主題中描述的功能與設定。注意到任何部分例外情況。任何 JVM 系統屬性設定都受到 AWS SDK for Java 和 適用於 Kotlin 的 AWS SDK 只有。

SDK	支援	備註或更多資訊
AWS CLI V2	是	
SDK 對於 C ++	是	
SDK 對於轉到 V2 (1.x)	是	
SDK 對於圍棋 1.x (V1)	是	若要使用共用 config 檔案設定，您必須開啟從設定檔載入功能；請參閱 工作階段 。
SDK 對於爪哇 2.x	是	對 SDK 於 Java 2.x 使用 <code>AWS_ENDPOINT_DISCOVERY_ENABLED</code> 的環境變量名稱。
SDK 對於爪哇 1.x	部分	JVM 系統屬性不受支援。
SDK 對於 JavaScript 3. x	是	
SDK 對於 JavaScript 2.x	是	
SDK 對於科特林	是	
SDK 對於 .NET 3.x	是	

SDK	支援	備註或更多資訊
SDK對於 PHP 3. x	是	
SDK對於 Python (肉毒桿菌 3)	是	
SDK對於紅寶石 3.x	是	
SDK對於鏽	部分	僅支援「時間流」。
SDK為斯威夫特	否	
適用的工具 PowerShell	是	

一般組態設定

SDKs支援一些設定整體SDK行為的一般設定。

使用下列項目設定此功能：

api_versions-共享 AWS config檔案設定

一些 AWS 服務會維護多個API版本，以支援回溯相容性。默認情況下，SDK和 AWS CLI 作業會使用最新的可用API版本。要要求特定API版本用於您的請求，請在您的個人資料中包含該api_versions設置。

預設值：無。(最新API版本由使用SDK。)

有效值：這是一個嵌套設置，後跟一個或多個縮進行每行標識 AWS 服務和要使用的API版本。請參閱文件以了解 AWS 服務，以了解哪些API版本可用。

這個例子為兩個設置了一個特定的API版本 AWS config檔案中的服務。這些API版本僅用於在包含這些設定的設定檔下執行的命令。任何其他服務的命令都會使用該服務的最新版本API。

```
api_versions =
  ec2 = 2015-03-01
  cloudfront = 2015-09-017
```

ca_bundle-共享 AWS config檔案設定, AWS_CA_BUNDLE-環境變量

指定建立SSL/TLS連線時要使用的自訂憑證套裝軟體 (.pem副檔名為副檔名的檔案) 的路徑。

預設值：無

有效值：指定完整路徑或基礎檔案名稱。如果有基底檔案名稱，系統會嘗試在PATH環境變數指定的資料夾內尋找程式。

在config檔案中設定此值的範例：

```
[default]
ca_bundle = dev/apps/ca-certs/cabundle-2019mar05.pem
```

由於作業系統處理路徑和路徑字元逸出的方式有所不同，以下是在 Windows 上的config檔案中設定此值的範例：

```
[default]
ca_bundle = C:\\Users\\username\\.aws\\aws-custom-bundle.pem
```

通過命令行設置環境變量的 Linux/macOS 示例：

```
export AWS_CA_BUNDLE=/dev/apps/ca-certs/cabundle-2019mar05.pem
```

通過命令行設置環境變量的 Windows 示例：

```
setx AWS_CA_BUNDLE C:\dev\apps\ca-certs\cabundle-2019mar05.pem
```

output-共享 AWS config檔案設定

指定結果的格式化方式 AWS CLI 和其他 AWS SDKs和工具。

預設值：json

有效值：

- **json**— 輸出格式為字JSON串。
- **yaml**— 輸出格式為字YAML串。
- **yaml-stream**— 輸出會串流並格式化為YAML字串。串流可加速處理大型資料類型。

- **text** – 輸出的格式是多行以 Tab 分隔的字串值。這對於將輸出傳遞給文字處理器 (如 grep、sed 或 awk) 非常有用。
- **table** – 輸出的格式為使用字元 +|- 形成儲存格框線的表格。它通常以「方便人類使用」的格式來呈現資訊，這種格式比其他格式更容易閱讀，但在編寫程式方面較不有用。

parameter_validation-共享 AWS config 檔案設定

指定 SDK 或工具是否嘗試驗證命令列參數，然後再將其傳送至 AWS 服務端點。

預設值：true

有效值：

- **true** – 預設值。SDK 或工具會執行命令列參數的用戶端驗證。這有助於 SDK 或工具確認參數是否有效，並捕獲一些錯誤。SDK 或工具可以拒絕無效的請求，然後再將請求傳送至 AWS 服務端點。
- **false** - SDK 或工具在將命令行參數發送到之前不驗證命令行參數 AWS 服務端點。所以此 AWS 服務端點負責驗證所有請求並拒絕無效的請求。

兼容性 AWS SDKs

下列項目 SDKs 支援本主題中描述的功能和設定。注意到任何部分例外情況。任何 JVM 系統屬性設定都受到 AWS SDK for Java 和 適用於 Kotlin 的 AWS SDK 只有。

SDK	支 備註或更多資訊 援
AWS CLI V2	部 api_versions 不支援。 分
SDK 對於 C ++	是
SDK 對於轉到 V2 (1.x)	部 api_versions 並且 parameter_validation 不支持。 分
SDK 對於圍棋 1.x (V1)	部 api_versions 並且 parameter_validation 不支持。 分 若要使用共用 config 檔案設定，您必須開啟從設定檔載入功能；請參閱 工作階段 。
SDK 對於爪哇 2.x	否

SDK	支援	備註或更多資訊
SDK對於爪哇 1.x	否	
SDK對於 JavaScript 3. x	是	
SDK對於 JavaScript 2.x	是	
SDK對於科特林	否	
SDK對於 .NET3.x	否	
SDK對於 PHP 3. x	是	
SDK對於 Python (肉毒桿菌 3)	是	
SDK對於紅寶石 3.x	是	
SDK對於鏽	否	
SDK為斯威夫特	否	
適用的工具 PowerShell	否	

IMDS 用戶端

SDKs 使用工作階段導向請求實作執行個體中繼資料服務第 2 版 (IMDSv2) 用戶端。如需的詳細資訊IMDSv2，請參閱 Amazon 使用者指南 中的[使用 IMDSv2](#)。EC2 IMDS 用戶端可透過SDK程式碼基礎中提供的用戶端組態物件進行設定。

使用下列設定此功能：

retries - 用戶端組態物件成員

任何失敗請求的額外重試嘗試次數。

預設值：3

有效值：大於 0 的數字。

port - 用戶端組態物件成員

端點的連接埠。

預設值：80

有效值：數字。

token_ttl - 用戶端組態物件成員

權杖TTL的。

預設值：21,600 秒（6 小時，最大分配時間）。

有效值：數字。

endpoint - 用戶端組態物件成員

的端點IMDS。

預設值：如果 `endpoint_mode` 等於 IPv4，則預設端點為 `http://169.254.169.254`。如果 `endpoint_mode` 等於 IPv6，則預設端點為 `http://[fd00:ec2::254]`。

有效值：有效 URI。

大多數 支援下列選項SDKs。如需詳細資訊，請參閱您的特定SDK程式碼基礎。

endpoint_mode - 用戶端組態物件成員

的端點模式IMDS。

預設值：IPv4

有效值：IPv4、IPv6

http_open_timeout - 用戶端組態物件成員（名稱可能有所不同）

等待連線開啟的秒數。

預設值：1 秒。

有效值：大於 0 的數字。

http_read_timeout - 用戶端組態物件成員（名稱可能有所不同）

要讀取的資料塊秒數。

預設值：1 秒。

有效值：大於 0 的數字。

http_debug_output - 用戶端組態物件成員（名稱可能有所不同）

設定用於偵錯的輸出串流。

預設值：無。

有效值：有效的 I/O 串流，例如 STDOUT。

backoff - 用戶端組態物件成員（名稱可能有所不同）

在重試或客戶提供的退避函數之間進入休眠的秒數。這會覆寫預設指數退避策略。

預設值：依 而異 SDK。

有效值：依 而異 SDK。可以是數值或呼叫自訂函數。

與 的相容性 AWS SDKs

以下 SDKs 支援本主題所述的功能和設定。會記下任何部分例外狀況。AWS SDK for Java 和 適用於 Kotlin 的 AWS SDK 僅支援任何 JVM 系統屬性設定。

SDK	支 備註或更多資訊 援
AWS CLI v2	是
SDK 適用於 C++	否
SDK for Go V2 (1.x)	是
SDK for Go 1.x (V1)	是
SDK 適用於 Java 2.x	是
SDK 適用於 Java 1.x	是
SDK 適用於 JavaScript 3.x	是
SDK 適用於 JavaScript 2.x	是

SDK	支援 備註或更多資訊
SDK 適用於 Kotlin	否
SDK 適用於 。NET 3.x	是
SDK 適用於 PHP 3.x	是
SDK for Python (Boto3)	是
SDK 適用於 Ruby 3.x	是
SDK for Rust	是
SDK 適用於 Swift	是
適用於 的工具 PowerShell	是

重試行為

重試行為包括有關SDKs嘗試從失敗中復原的設定 AWS 服務。

使用下列項目設定此功能：

retry_mode-共享 AWS config檔案設定, **AWS_RETRY_MODE**-環境變量, **aws.retryMode**-JVM 系統屬性：僅限爪哇/科特林

指定SDK或開發人員工具嘗試重試的方式。

預設值：此值特定於您的SDK. 檢查您的特定SDK指南或SDK代碼庫的默認值**retry_mode**。

有效值：

- **standard**— (建議) 跨越建議的重試規則集 AWS SDKs。此模式包含重試的一組標準錯誤，並自動調整重試次數以最大化可用性和穩定性。此模式適用於多租戶應用程式。除非明確設定，否則使用此模式的預設最大嘗試次數**max_attempts**為三次。
- **adaptive**— 僅適用於特殊使用案例的重試模式，其中包括標準模式的功能以及自動用戶端速率限制。除非您小心隔離應用程式承租人，否則不建議多租用戶應用程式使用此重試模式。如需更多資訊，請參閱[選擇standard和adaptive重試模式](#)。這種模式是實驗性的，它可能會改變future的行為。

- `legacy-` (不推薦) 特定於您的 SDK (請檢查您的具體 SDK 指南或您的 SDK 的代碼庫)。

`max_attempts-` 共享 `AWS config` 檔案設定, `AWS_MAX_ATTEMPTS-` 環境變量, `aws.maxAttempts-` JVM 系統屬性：僅限爪哇/科特林

指定對請求進行的嘗試次數上限。

預設值：如果未指定此值，其預設值取決於 `retry_mode` 設定值：

- 如果 `retry_mode` 是 `legacy` — 使用特定於您的默認值 SDK (`max_attempts` 默認情況下檢查您的 SDK 的特定 SDK 指南或代碼庫)。
- 如果 `retry_mode` 是 `standard` — 進行三次嘗試。
- 如果 `retry_mode` 是 `adaptive` — 進行三次嘗試。

有效值：大於 0 的數字。

選擇 `standard` 和 `adaptive` 重試模式

我們建議您使用 `standard` 重試模式，除非您確定您的使用方式更適合 `adaptive`。

Note

此 `adaptive` 模式假設您根據後端服務可能會限制要求的範圍來共用用戶端。如果不這樣做，如果您對兩個資源使用相同的用戶端，則一個資源中的節流可能會延遲對不相關資源的請求。

標準	自适应
應用程式使用案例：全部。	應用程序用例： <ol style="list-style-type: none"> 1. 對延遲不敏感。 2. 用戶端只會存取單一資源，或者您提供邏輯，以便依據正在存取的服務資源來分別集區用戶端。
支持斷路，以防止在 SDK 中斷期間重試。	支持斷路，以防止在 SDK 中斷期間重試。
在發生故障時使用抖動的指數輪詢。	使用動態輪詢持續時間來嘗試將失敗的要求數目降到最低，以換取延遲增加的可能性。

標準	自适应
永遠不會延遲第一次請求嘗試，只有重試。	可以限制或延遲初始請求嘗試。

如果您選擇使用adaptive模式，您的應用程式必須建構圍繞可能限制的每個資源所設計的用户端。在這種情況下，資源比只是思考每個資源更精細地調整 AWS 服務。AWS 服務 可以有額外的維度，它們用來限制請求。讓我們以 Amazon DynamoDB 服務為例。DynamoDB 使用 AWS 區域 加上被訪問的表以節流請求。這意味著您的代碼正在訪問的一個表可能比其他表進行限制。如果您的程式碼使用相同的用戶端來存取所有資料表，且對其中一個資料表的要求會受到限制，則調適性重試模式會降低所有資料表的要求率。您的代碼應該設計為每個 Region-and-table 對有一個客戶端。如果您在使用adaptive模式時遇到意外延遲，請參閱具體內容 AWS 您所使用之服務的文件指南。

重試模式實施細節

所以此 AWS SDKs利用令牌桶來決定是否應該重試請求以及（在adaptive重試模式的情況下）應該多快地發送請求。兩個令牌儲存桶由使用SDK：重試令牌儲存桶和請求率令牌儲存桶。

- 重試 Token 值區可用來判斷是否SDK應暫時停用重試，以便在中斷期間保護上游和下游服務。在嘗試重試之前從儲存桶中獲取令牌，並在請求成功時將令牌返回到儲存桶。如果嘗試重試時值區為空，則不SDK會重試要求。
- 請求率 Token 儲存桶僅用於adaptive重試模式，以確定發送請求的速率。在發送請求之前，會從儲存桶中獲取令牌，並根據服務返回的節流響應，以動態確定的速率將令牌返回到儲存桶。

以下是standard和adaptive重試模式的高級偽代碼：

```

MakeSDKRequest() {
  attempts = 0
  loop {
    GetSendToken()
    response = SendHTTPRequest()
    RequestBookkeeping(response)
    if not Retryable(response)
      return response
    attempts += 1
    if attempts >= MAX_ATTEMPTS:
      return response
    if not HasRetryQuota(response)
      return response
    delay = ExponentialBackoff(attempts)
  }
}

```

```
    sleep(delay)
  }
}
```

以下是關於偽代碼中使用的組件的更多細節：

GetSendToken:

此步驟僅用於adaptive重試模式。此步驟從請求率令牌存儲桶中獲取令牌。如果令牌不可用，它將等待一個可用。您SDK可能有配置選項可以使請求失敗，而不是等待。值區中的Token會根據用戶端收到的節流回應數量，以動態決定的速率重新填入。

SendHTTPRequest:

此步驟會將要求傳送至AWS。最多AWS SDKs使用使用連線集區的程式HTTP庫，在提出HTTP要求時重複使用現有連線。一般而言，如果要求因為節流錯誤而失敗，但如果要求因暫時性錯誤而失敗，則會重複使用連線。

RequestBookkeeping:

如果請求成功，令牌將添加到令牌存儲桶中。僅對於adaptive重試模式，請求率Token存儲桶的填充率會根據收到的響應類型更新。

Retryable:

此步驟會根據下列項目決定是否可以重試回應：

- 狀HTTP態碼。
- 從服務傳回的錯誤碼。
- 連線錯誤，定義為未收到來自服務SDK的HTTP回應所收到的任何錯誤。

暫態錯誤 (HTTP狀態碼 400、408、500、502、503 和 504) 和節流錯誤 (HTTP狀態碼 400、403、429、502、503 和 509) 都可能重試。SDK重試行為與錯誤代碼或服務中的其他數據一起確定。

MAX_ATTEMPTS:

預設的最大嘗試次數由設定retry_mode設定，除非被max_attempts設定覆蓋。

HasRetryQuota

此步驟從重試令牌存儲桶獲取令牌。如果重試令牌存儲桶為空，則不會重試請求。

ExponentialBackoff

對於可重試的錯誤，會使用截斷的指數輪詢來計算重試延遲。SDKs使用截斷的二進制指數輪詢與抖動。下列演算法顯示如何針對要求回應定義睡眠的時間量 (以秒為單位) i ：

```
seconds_to_sleep_i = min(b*r^i, MAX_BACKOFF)
```

在上述演算法中，下列值適用：

b = random number within the range of: $0 \leq b \leq 1$

r = 2

$MAX_BACKOFF$ = 20 seconds對於大多數SDKs。請參閱您的具體SDK指南或源代碼以進行確認。

兼容性 AWS SDKs

下列項目SDKs支援本主題中描述的功能與設定。注意到任何部分例外情況。任何JVM系統屬性設定都受到 AWS SDK for Java 和 適用於 Kotlin 的 AWS SDK 只有。

SDK	支 備註或更多資訊 援
AWS CLI V2	是
SDK對於 C ++	是
SDK對於轉到 V2 (1.x)	是
SDK對於圍棋 1.x (V1)	否
SDK對於爪哇 2.x	是
SDK對於爪哇 1.x	是 JVM系統屬性：使用com.amazonaws.sdk.maxAttempts 代替aws.maxAttempts；使用com.amazonaws.sdk.retryMode 代替aws.retryMode。
SDK對於 JavaScript 3. x	是

SDK	支援	備註或更多資訊
SDK對於 JavaScript 2.x	否	支援最大重試次數、含抖動的指數輪詢，以及重試輪詢的自訂方法選項。
SDK對於科特林	是	
SDK對於 .NET3.x	是	
SDK對於 PHP 3. x	是	
SDK對於 Python (肉毒桿菌 3)	是	
SDK對於紅寶石 3.x	是	
SDK對於鏽	是	
SDK為斯威夫特	是	
適用的工具 PowerShell	是	

請求壓縮

Note

有關了解設置頁面佈局或解釋兼容性的幫助 AWS SDKs表格如下，請參閱[設定頁面](#)。

AWS SDKs和工具可以在將請求發送到時自動壓縮有效載荷 AWS 服務 支持接收壓縮的有效載荷。在將裝載傳送至服務之前，先壓縮用戶端上的裝載，可能會減少傳送資料至服務所需的要求總數和頻寬，以及減少因為承載大小的服務限制而導致失敗的要求。對於壓縮，SDK或工具會選取服務和支援的編碼演算法SDK。但是，當前可能的編碼列表僅由 gzip 組成，但將 future 可能會擴展。

如果您的應用程式使用 [Amazon](#)，請求壓縮會特別有用 CloudWatch。CloudWatch 是一種監視和可觀察性服務，以日誌，指標和事件的形式收集監視和操作數據。支援壓縮的服務作業範例之一 CloudWatch就是[PutMetricDataAPI](#)方法。

使用下列項目設定此功能：

disable_request_compression-共享 AWS config 檔案設定,
AWS_DISABLE_REQUEST_COMPRESSION-環境變量, **aws.disableRequestCompression**-JVM 系統屬性：僅限爪哇/科特林

開啟或關閉 SDK 或工具是否會在傳送要求之前壓縮承載。

預設值：false

有效值：

- **true**— 關閉請求壓縮。
- **false**— 盡可能使用請求壓縮。

request_min_compression_size_bytes-共享 AWS config 檔案設定, **AWS_REQUEST_MIN_COMPRESSION_SIZE_BYTES**-環境變量,
aws.requestMinCompressionSizeBytes-JVM 系統屬性：僅限爪哇/科特林

設置 SDK 或工具應壓縮的請求主體的最小大小 (以字節為單位)。壓縮時，小的有效載荷可能會變得更長，因此，執行壓縮的下限是有意義的。此值包含在內，壓縮大於或等於該值的請求大小。

預設值：

有效值：介於 0 到 10485760 位元組之間的整數值 (包含在內)。

兼容性 AWS SDKs

下列項目 SDKs 支援本主題中描述的功能和設定。注意到任何部分例外情況。任何 JVM 系統屬性設定都受到 AWS SDK for Java 和 適用於 Kotlin 的 AWS SDK 只有。

SDK	支援	備註或更多資訊
AWS CLI V2	是	
SDK 對於 C ++	是	
SDK 對於轉到 V2 (1.x)	是	
SDK 對於圍棋 1.x (V1)	否	

SDK	支援	備註或更多資訊
SDK對於爪哇 2.x	是	
SDK對於爪哇 1.x	否	
SDK對於 JavaScript 3. x	是	
SDK對於 JavaScript 2.x	否	
SDK對於科特林	是	
SDK對於 .NET3.x	是	
SDK對於 PHP 3. x	是	
SDK對於 Python (肉毒桿菌 3)	是	
SDK對於紅寶石 3.x	是	
SDK對於銹	是	
SDK為斯威夫特	否	
適用的工具 PowerShell	是	

服務特定端點

服務特定端點設定可讓您選擇使用您選擇的端點來處理API要求，並讓該選擇持續存在。這些設定提供支援本機端點、VPC端點和協力廠商本機的彈性 AWS 開發環境。不同的端點可用於測試和生產環境。您可以URL為個人指定端點 AWS 服務。

使用下列項目設定此功能：

endpoint_url-共享 AWS config 檔案設定, **AWS_ENDPOINT_URL**-環境變量, **aws.endpointUrl**-JVM 系統屬性：僅限爪哇/科特林

當直接在設定檔中指定或作為環境變數指定時，此設定會指定用於所有服務要求的端點。任何已設定的服務特定端點會覆寫此端點。

您也可以共享的 `services` 部分中使用此設置 AWS config 用於為特定服務設定自訂端點的檔案。如需要小 `services` 節中子區段使用的所有服務識別碼金鑰清單，請參閱 [服務特定端點的識別碼](#)。

預設值：none

有效值：A URL 包括端點的配置和主機。URL 可以選擇性地包含包含一或多個路徑區段的路徑元件。

AWS_ENDPOINT_URL_<SERVICE>-環境變量, **aws.endpointUrl<ServiceName>**-JVM 系統屬性：僅限爪哇/科特林

AWS_ENDPOINT_URL_<SERVICE>，在 `<SERVICE>` 哪裡 AWS 服務 識別碼，設定特定服務的自訂端點。如需所有服務特定環境變數的清單，請參閱 [服務特定端點的識別碼](#)。

此服務特定端點會覆寫中 **AWS_ENDPOINT_URL** 設定的任何全域端點。

預設值：none

有效值：A URL 包括端點的配置和主機。URL 可以選擇性地包含包含一或多個路徑區段的路徑元件。

ignore_configured_endpoint_urls-共享 AWS config 檔案設定, **AWS_IGNORE_CONFIGURED_ENDPOINT_URLS**-環境變量, **aws.ignoreConfiguredEndpointUrls**-JVM 系統屬性：僅限爪哇/科特林

此設定用於忽略所有自訂端點組態。

請注意，無論此設定為何，都會使用程式碼或服務用戶端本身中設定的任何明確端點。例如，包括指 `--endpoint-url` 命令行參數與 AWS CLI 命令或將端點傳遞 URL 到客戶端構造函數將始終生效。

預設值：false

有效值：

- **true**— SDK 或工具不會從共用 config 檔案或環境變數讀取任何自訂組態選項以設定端點 URL。
- **false**— SDK 或工具會使用共用 config 檔案或環境變數中的任何可用使用者提供的端點。

使用環境變數設定端點

若要將所有服務的要求路由至自訂端點URL，請設定AWS_ENDPOINT_URL全域環境變數。

```
export AWS_ENDPOINT_URL=http://localhost:4567
```

若要路由特定的要求 AWS 服務 至自訂端點URL，請使用AWS_ENDPOINT_URL_<SERVICE>環境變數。Amazon DynamoDB 有一個serviceId的[DynamoDB](#)。對於此服務，端點URL環境變數為AWS_ENDPOINT_URL_DYNAMODB。此端點的優先順序高於AWS_ENDPOINT_URL針對此服務設定的全域端點。

```
export AWS_ENDPOINT_URL_DYNAMODB=http://localhost:5678
```

作為另一個例子，AWS Elastic Beanstalk 有一個serviceId的[Elastic Beanstalk](#)。所以此 AWS 服務 標識符是基於API模型的，serviceId通過用下劃線替換所有空格並大寫所有字母。若要設定此服務的端點，對應的環境變數為AWS_ENDPOINT_URL_ELASTIC_BEANSTALK。如需所有服務特定環境變數的清單，請參閱 [服務特定端點的識別碼](#)。

```
export AWS_ENDPOINT_URL_ELASTIC_BEANSTALK=http://localhost:5567
```

使用共用config檔案設定端點

在共享config文件中，用endpoint_url於不同的地方用於不同的功能。

- endpoint_url直接在一個內指定，profile使該端點成為全局端點。
- endpoint_url嵌套在services區段內的服務識別碼金鑰下，可讓該端點套用至僅對該服務發出的要求。如需在共用 config 檔案中定義 services 區段的詳細資訊，請參閱 [組態檔案格式](#)。

下列範例使用定services義來設定用於 Amazon S3 的服務特URL定端點，以及設定用於所有其他服務的自訂全球端點：

```
[profile dev-s3-specific-and-global]  
endpoint_url = http://localhost:1234  
services = s3-specific  
  
[services s3-specific]  
s3 =  
  endpoint_url = https://play.min.io:9000
```

單一設定檔可以設定多個服務的端點。此範例顯示如何URLs為 Amazon S3 和設定服務特定端點 AWS Elastic Beanstalk 在同一個配置文件中。AWS Elastic Beanstalk 有一個serviceId的[Elastic Beanstalk](#)。所以此 AWS 服務 標識符是基於API模型的，serviceId通過用下劃線替換所有空格並降低所有字母。因此，服務標識符密鑰變為，elastic_beanstalk並且此服務的設置在行開始elastic_beanstalk = 。如需要在 services 區段中使用的所有服務識別碼金鑰的清單，請參閱 [服務特定端點的識別碼](#)。

```
[services testing-s3-and-eb]
s3 =
  endpoint_url = http://localhost:4567
elastic_beanstalk =
  endpoint_url = http://localhost:8000

[profile dev]
services = testing-s3-and-eb
```

服務配置部分可以從多個配置文件中使用。例如，兩個縱斷面可以在變更其他縱斷面性質時使用相同的services定義：

```
[services testing-s3]
s3 =
  endpoint_url = https://localhost:4567

[profile testing-json]
output = json
services = testing-s3

[profile testing-text]
output = text
services = testing-s3
```

使用角色型認證在設定檔中設定端點

如果您的設定檔具有透過IAM假設角色功能的source_profile參數設定以角色為基礎的認證，則 SDK只會針對指定的設定檔使用服務組態。它不會使用與其連結的角色的設定檔。例如，使用下列共用 config 檔案：

```
[profile A]
credential_source = Ec2InstanceMetadata
endpoint_url = https://profile-a-endpoint.aws/
```

```
[profile B]
source_profile = A
role_arn = arn:aws:iam::123456789012:role/roleB
services = profileB

[services profileB]
ec2 =
  endpoint_url = https://profile-b-ec2-endpoint.aws
```

如果您使用設定檔B並在程式碼中呼叫 AmazonEC2，端點會解析為https://profile-b-ec2-endpoint.aws。如果您的程式碼向任何其他服務發出要求，端點解析將不會遵循任何自訂邏輯。端點未解析為設定檔 A 中定義的全域端點。若要讓全域端點對設定檔 B 生效，您需要直接在 B 設定檔中設定 endpoint_url。如需 source_profile 設定的詳細資訊，請參閱[擔任角色登入資料提供者](#)。

設定的優先順序

此功能的設定可以同時使用，但每個服務只會優先使用一個值。對於給定的API呼叫 AWS 服務，下面的順序用於選擇一個值：

1. 在代碼或服務客戶端上設置的任何明確設置本身都優先於其他任何設置。
 - 對於 AWS CLI，這是由--endpoint-url命令行參數提供的值。對於SDK，明確的分配可以採用您在實例化時設置的參數的形式 AWS 服務 客戶端或配置對象。
2. 服務特定環境變數所提供的值，例如AWS_ENDPOINT_URL_DYNAMODB。
3. AWS_ENDPOINT_URL 全域端點環境變數所提供的值。
4. endpoint_url設定巢狀於共用config檔案services區段內的服務識別碼金鑰下所提供的值。
5. 由共用config檔案中直接指endpoint_url定profile的設定所提供的值。
6. 相應的任何預設端點 URL AWS 服務 最後使用。

兼容性 AWS SDKs

下列項目SDKs支援本主題中描述的功能與設定。注意到任何部分例外情況。任何JVM系統屬性設定都受到 AWS SDK for Java 和 適用於 Kotlin 的 AWS SDK 只有。

SDK	支 備註或更多資訊 援
AWS CLI V2	是

SDK	支援	備註或更多資訊
SDK對於 C ++	否	
SDK對於轉到 V2 (1.x)	是	
SDK對於圍棋 1.x (V1)	否	
SDK對於爪哇 2.x	是	
SDK對於爪哇 1.x	否	
SDK對於 JavaScript 3. x	是	
SDK對於 JavaScript 2.x	否	
SDK對於科特林	是	
SDK為 .NET3.x	是	
SDK對於 PHP 3. x	是	
SDK對於 Python (肉毒桿菌 3)	是	
SDK對於紅寶石 3.x	是	
SDK對於銹	否	
SDK為斯威夫特	否	
適用的工具 PowerShell	是	

服務特定端點的識別碼

如需有關如何及在何處使用下表中識別碼的資訊，請參閱[服務特定端點](#)。

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
AccessAnalyzer	aws_endpoint_url_accessanalyzer
Account	aws_endpoint_url_account
ACM	aws_endpoint_url_acm
ACM PCA	aws_endpoint_url_acm_pca
Alexa For Business	aws_endpoint_url_alexa_for_business
amp	aws_endpoint_url_amp
Amplify	aws_endpoint_url_amplify
AmplifyBackend	aws_endpoint_url_amplifybackend
AmplifyUIBuilder	aws_endpoint_url_amplifyuibuilder

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
API Gateway	aws_endpoint_url_api_gateway
ApiGatewayManagementApi	aws_endpoint_url_apigatewaymanagementapi
ApiGatewayV2	aws_endpoint_url_apigatewayv2
AppConfig	aws_endpoint_url_appconfig
AppConfigData	aws_endpoint_url_appconfigdata
AppFabric	aws_endpoint_url_appfabric
Appflow	aws_endpoint_url_appflow
AppIntegrations	aws_endpoint_url_appintegrations

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
Application Auto Scaling	aws:application-auto-scaling:AWS_ENDPOINT_URL_APPLICATION_AUTO_SCALING
Application Insights	aws:application-insights:AWS_ENDPOINT_URL_APPLICATION_INSIGHTS
ApplicationCostProfiler	aws:application-cost-profiler:AWS_ENDPOINT_URL_APPLICATIONCOSTPROFILER
App Mesh	aws:app-mesh:AWS_ENDPOINT_URL_APP_MESH
AppRunner	aws:app-runner:AWS_ENDPOINT_URL_APPRUNNER
AppStream	aws:appstream:AWS_ENDPOINT_URL_APPSTREAM
AppSync	aws:appsync:AWS_ENDPOINT_URL_APPSVC
ARC Zonal Shift	aws:arc-zonal-shift:AWS_ENDPOINT_URL_ARC_ZONAL_SHIFT

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
Artifact	a: AWS_ENDPOINT_URL_ARTIFACT
Athena	a: AWS_ENDPOINT_URL_ATHENA
AuditManager	a: AWS_ENDPOINT_URL_AUDITMANAGER
Auto Scaling	a: AWS_ENDPOINT_URL_AUTO_SCALING
Auto Scaling Plans	a: AWS_ENDPOINT_URL_AUTO_SCALING_PLANS
b2bi	b: AWS_ENDPOINT_URL_B2BI
Backup	b: AWS_ENDPOINT_URL_BACKUP
Backup Gateway	b: AWS_ENDPOINT_URL_BACKUP_GATEWAY
BackupStorage	b: AWS_ENDPOINT_URL_BACKUPSTORAGE
Batch	b: AWS_ENDPOINT_URL_BATCH

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
BCM Data Exports	b: AWS_ENDPOINT_URL_BCM_DATA_EXPORTS e:
Bedrock	b: AWS_ENDPOINT_URL_BEDROCK
Bedrock Agent	b: AWS_ENDPOINT_URL_BEDROCK_AGENT g:
Bedrock Agent Runtime	b: AWS_ENDPOINT_URL_BEDROCK_AGENT_RUNTIME g: ir
Bedrock Runtime	b: AWS_ENDPOINT_URL_BEDROCK_RUNTIME ur
billingconductor	b: AWS_ENDPOINT_URL_BILLINGCONDUCTOR nc
Braket	b: AWS_ENDPOINT_URL_BRAKET
Budgets	b: AWS_ENDPOINT_URL_BUDGETS
Cost Explorer	c: AWS_ENDPOINT_URL_COST_EXPLORER o:

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
chatbot	cl AWS_ENDPOINT_URL_CHATBOT
Chime	cl AWS_ENDPOINT_URL_CHIME
Chime SDK Identity	cl AWS_ENDPOINT_URL_CHIME_SDK_IDENTITY _i
Chime SDK Media Pipelines	cl AWS_ENDPOINT_URL_CHIME_SDK_MEDIA_PIPELINES _r p
Chime SDK Meetings	cl AWS_ENDPOINT_URL_CHIME_SDK_MEETINGS _r
Chime SDK Messaging	cl AWS_ENDPOINT_URL_CHIME_SDK_MESSAGING _r g
Chime SDK Voice	cl AWS_ENDPOINT_URL_CHIME_SDK_VOICE _r

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
CleanRooms	c: AWS_ENDPOINT_URL_CLEANROOMS
CleanRoomsML	c: AWS_ENDPOINT_URL_CLEANROOMSML
Cloud9	c: AWS_ENDPOINT_URL_CLOUD9
CloudControl	c: AWS_ENDPOINT_URL_CLOUDCONTROL
CloudDirectory	c: AWS_ENDPOINT_URL_CLOUDDIRECTORY
CloudFormation	c: AWS_ENDPOINT_URL_CLOUDFORMATION
CloudFront	c: AWS_ENDPOINT_URL_CLOUDFRONT
CloudFront KeyValuesStore	c: AWS_ENDPOINT_URL_CLOUDFRONT_KEYVALUESTORE

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
CloudHSM	c: AWS_ENDPOINT_URL_CLOUDHSM
CloudHSM V2	c: AWS_ENDPOINT_URL_CLOUDHSM_V2 v:
CloudSearch	c: AWS_ENDPOINT_URL_CLOUDSEARCH cl
CloudSearch Domain	c: AWS_ENDPOINT_URL_CLOUDSEARCH_DOMAIN cl
CloudTrail	c: AWS_ENDPOINT_URL_CLOUDTRAIL l
CloudTrail Data	c: AWS_ENDPOINT_URL_CLOUDTRAIL_DATA l_
CloudWatch	c: AWS_ENDPOINT_URL_CLOUDWATCH h
codeartifact	c: AWS_ENDPOINT_URL_CODEARTIFACT a:

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_ <SERVICE> 環境變數
CodeBuild	AWS_ENDPOINT_URL_CODEBUILD
CodeCatalyst	AWS_ENDPOINT_URL_CODECATALYST
CodeCommit	AWS_ENDPOINT_URL_CODECOMMIT
CodeDeploy	AWS_ENDPOINT_URL_CODEDEPLOY
CodeGuru Reviewer	AWS_ENDPOINT_URL_CODEGURU_REVIEWER
CodeGuru Security	AWS_ENDPOINT_URL_CODEGURU_SECURITY
CodeGuruProfiler	AWS_ENDPOINT_URL_CODEGURUPROFILER
CodePipeline	AWS_ENDPOINT_URL_CODEPIPELINE

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
CodeStar	code AWS_ENDPOINT_URL_CODESTAR
CodeStar connections	code AWS_ENDPOINT_URL_CODESTAR_CONNECTIONS code
codestar notifications	code AWS_ENDPOINT_URL_CODESTAR_NOTIFICATIONS code
Cognito Identity	code AWS_ENDPOINT_URL_COGNITO_IDENTITY code
Cognito Identity Provider	code AWS_ENDPOINT_URL_COGNITO_IDENTITY_PROVIDER code
Cognito Sync	code AWS_ENDPOINT_URL_COGNITO_SYNC code
Comprehend	code AWS_ENDPOINT_URL_COMPREHEND code

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
ComprehendMedical	AWS_ENDPOINT_URL_COMPREHENDMEDICAL
Compute Optimizer	AWS_ENDPOINT_URL_COMPUTE_OPTIMIZER
Config Service	AWS_ENDPOINT_URL_CONFIG_SERVICE
Connect	AWS_ENDPOINT_URL_CONNECT
Connect Contact Lens	AWS_ENDPOINT_URL_CONNECT_CONTACT_LENS
ConnectCampaigns	AWS_ENDPOINT_URL_CONNECTCAMPAIGNS
ConnectCases	AWS_ENDPOINT_URL_CONNECTCASES
ConnectParticipant	AWS_ENDPOINT_URL_CONNECTPARTICIPANT

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
ControlTower	controltower: AWS_ENDPOINT_URL_CONTROLTOWER
Cost Optimization Hub	costoptimizationhub: AWS_ENDPOINT_URL_COST_OPTIMIZATION_HUB
Cost and Usage Report Service	costandusagereport: AWS_ENDPOINT_URL_COST_AND_USAGE_REPORT_SERVICE
Customer Profiles	customerprofiles: AWS_ENDPOINT_URL_CUSTOMER_PROFILES
DataBrew	databrew: AWS_ENDPOINT_URL_DATABREW
DataExchange	dataexchange: AWS_ENDPOINT_URL_DATAEXCHANGE
Data Pipeline	datapipeline: AWS_ENDPOINT_URL_DATA_PIPELINE
DataSync	datasync: AWS_ENDPOINT_URL_DATASYNC

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_ <SERVICE> 環境變數
DataZone	d: AWS_ENDPOINT_URL_DATAZONE
DAX	d: AWS_ENDPOINT_URL_DAX
Detective	d: AWS_ENDPOINT_URL_DETECTIVE
Device Farm	d: AWS_ENDPOINT_URL_DEVICE_FARM
DevOps Guru	d: AWS_ENDPOINT_URL_DEVOPS_GURU
Direct Connect	d: AWS_ENDPOINT_URL_DIRECT_CONNECT
Application Discovery Service	a: AWS_ENDPOINT_URL_APPLICATION_DISCOVERY_SERVICE
DLM	d: AWS_ENDPOINT_URL_DLM

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_ <SERVICE> 環境變數
Database Migration Service	d: AWS_ENDPOINT_URL_DATABASE_MIGRATION_ m: SERVICE _:
DocDB	d: AWS_ENDPOINT_URL_DOCDB
DocDB Elastic	d: AWS_ENDPOINT_URL_DOCDB_ELASTIC s:
drs	d: AWS_ENDPOINT_URL_DRS
Directory Service	d: AWS_ENDPOINT_URL_DIRECTORY_SERVICE _:
DynamoDB	d: AWS_ENDPOINT_URL_DYNAMODB
DynamoDB Streams	d: AWS_ENDPOINT_URL_DYNAMODB_STREAMS s:
EBS	e: AWS_ENDPOINT_URL_EBS
EC2	e: AWS_ENDPOINT_URL_EC2

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
EC2 Instance Connect	environmental AWS_ENDPOINT_URL_EC2_INSTANCE_CONNECT
ECR	environmental AWS_ENDPOINT_URL_ECR
ECR PUBLIC	environmental AWS_ENDPOINT_URL_ECR_PUBLIC
ECS	environmental AWS_ENDPOINT_URL_ECS
EFS	environmental AWS_ENDPOINT_URL_EFS
EKS	environmental AWS_ENDPOINT_URL_EKS
EKS Auth	environmental AWS_ENDPOINT_URL_EKS_AUTH
Elastic Inference	environmental AWS_ENDPOINT_URL_ELASTIC_INFERENCE
ElastiCache	environmental AWS_ENDPOINT_URL_ELASTICACHE
Elastic Beanstalk	environmental AWS_ENDPOINT_URL_ELASTIC_BEANSTALK

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
Elastic Transcoder	e: AWS_ENDPOINT_URL_ELASTIC_TRANSCODER i:
Elastic Load Balancing	e: AWS_ENDPOINT_URL_ELASTIC_LOAD_BALANCING o: c:
Elastic Load Balancing v2	e: AWS_ENDPOINT_URL_ELASTIC_LOAD_BALANCING_V2 o: c:
EMR	er: AWS_ENDPOINT_URL_EMR
EMR containers	er: AWS_ENDPOINT_URL_EMR_CONTAINERS i:
EMR Serverless	er: AWS_ENDPOINT_URL_EMR_SERVERLESS r: i:
EntityResolution	er: AWS_ENDPOINT_URL_ENTITYRESOLUTION o:

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
Elasticsearch Service	e: AWS_ENDPOINT_URL_ELASTICSEARCH_SERVICE a: i:
EventBridge	e: AWS_ENDPOINT_URL_EVENTBRIDGE g:
Evidently	e: AWS_ENDPOINT_URL_EVIDENTLY
finspace	f: AWS_ENDPOINT_URL_FINSPLACE
finspace data	f: AWS_ENDPOINT_URL_FINSPLACE_DATA d:
Firehose	f: AWS_ENDPOINT_URL_FIREHOSE
fis	f: AWS_ENDPOINT_URL_FIS
FMS	fr: AWS_ENDPOINT_URL_FMS
forecast	f: AWS_ENDPOINT_URL_FORECAST
forecastquery	f: AWS_ENDPOINT_URL_FORECASTQUERY u:

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
FraudDetector	f: AWS_ENDPOINT_URL_FRAUDETECTOR c1
FreeTier	f: AWS_ENDPOINT_URL_FREETIER
FSx	f: AWS_ENDPOINT_URL_FSX
GameLift	g: AWS_ENDPOINT_URL_GAMELIFT
Glacier	g: AWS_ENDPOINT_URL_GLACIER
Global Accelerator	g: AWS_ENDPOINT_URL_GLOBAL_ACCELERATOR c6
Glue	g: AWS_ENDPOINT_URL_GLUE
grafana	g: AWS_ENDPOINT_URL_GRAFANA
Greengrass	g: AWS_ENDPOINT_URL_GREENGRASS s
GreengrassV2	g: AWS_ENDPOINT_URL_GREENGRASSV2 s\

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_ <SERVICE> 環境變數
GroundStation	g: AWS_ENDPOINT_URL_GROUNDSTATION t:
GuardDuty	g: AWS_ENDPOINT_URL_GUARDDUTY
Health	h: AWS_ENDPOINT_URL_HEALTH
HealthLake	h: AWS_ENDPOINT_URL_HEALTHLAKE e
Honeycode	h: AWS_ENDPOINT_URL_HONEYCODE
IAM	i: AWS_ENDPOINT_URL_IAM
identitystore	i: AWS_ENDPOINT_URL_IDENTITYSTORE t:
imagebuilder	i: AWS_ENDPOINT_URL_IMAGEBUILDER d:
ImportExport	i: AWS_ENDPOINT_URL_IMPORTEXPORT o:

serviceId	AWS_ENDPOINT_URL_<SERVICE>	環境變數
	共用的服務識別碼金鑰	環境變數
Inspector	inspector	AWS_ENDPOINT_URL_INSPECTOR
Inspector Scan	inspector-scan	AWS_ENDPOINT_URL_INSPECTOR_SCAN
Inspector2	inspector2	AWS_ENDPOINT_URL_INSPECTOR2
InternetMonitor	internetmonitor	AWS_ENDPOINT_URL_INTERNETMONITOR
IoT	iot	AWS_ENDPOINT_URL_IOT
IoT Data Plane	iot-data-plane	AWS_ENDPOINT_URL_IOT_DATA_PLANE
IoT Jobs Data Plane	iot-jobs-data-plane	AWS_ENDPOINT_URL_IOT_JOBS_DATA_PLANE
IoT 1Click Devices Service	iot-1click-devices	AWS_ENDPOINT_URL_IOT_1CLICK_DEVICES_SERVICE

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
IoT 1Click Projects	AWS_ENDPOINT_URL_IOT_1CLICK_PROJECTS
IoTAnalytics	AWS_ENDPOINT_URL_IOTANALYTICS
IotDeviceAdvisor	AWS_ENDPOINT_URL_IOTDEVICEADVISOR
IoT Events	AWS_ENDPOINT_URL_IOT_EVENTS
IoT Events Data	AWS_ENDPOINT_URL_IOT_EVENTS_DATA
IoTFleetHub	AWS_ENDPOINT_URL_IOTFLEETHUB
IoTFleetWise	AWS_ENDPOINT_URL_IOTFLEETWISE

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
IoTSecureTunneling	i\ AWS_ENDPOINT_URL_IOTSECURETUNNELING t\
IoTSiteWise	i\ AWS_ENDPOINT_URL_IOTSITWISE s\
IoTThingsGraph	i\ AWS_ENDPOINT_URL_IOTTHINGSGRAPH g\
IoTTwinMaker	i\ AWS_ENDPOINT_URL_IOTTWINMAKER k\
IoT Wireless	i\ AWS_ENDPOINT_URL_IOT_WIRELESS e\
ivs	i\ AWS_ENDPOINT_URL_IVS
IVS RealTime	i\ AWS_ENDPOINT_URL_IVS_REALTIME i\
ivschat	i\ AWS_ENDPOINT_URL_IVSCHAT
Kafka	k\ AWS_ENDPOINT_URL_KAFKA

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
KafkaConnect	k: AWS_ENDPOINT_URL_KAFKACONNECT e:
kendra	k: AWS_ENDPOINT_URL_KENDRA
Kendra Ranking	k: AWS_ENDPOINT_URL_KENDRA_RANKING n:
Keyspaces	k: AWS_ENDPOINT_URL_KEYSPACES
Kinesis	k: AWS_ENDPOINT_URL_KINESIS
Kinesis Video Archived Media	k: AWS_ENDPOINT_URL_KINESIS_VIDEO_ARCHIVED_MEDIA i: a:
Kinesis Video Media	k: AWS_ENDPOINT_URL_KINESIS_VIDEO_MEDIA i: a:

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
Kinesis Video Signaling	k: AWS_ENDPOINT_URL_KINESIS_VIDEO_SIGNALING i: a:
Kinesis Video WebRTC Storage	k: AWS_ENDPOINT_URL_KINESIS_VIDEO_WEBRTC_STORAGE i: C_STORAGE t: e:
Kinesis Analytics	k: AWS_ENDPOINT_URL_KINESIS_ANALYTICS n: v:
Kinesis Analytics V2	k: AWS_ENDPOINT_URL_KINESIS_ANALYTICS_V2 n: v:
Kinesis Video	k: AWS_ENDPOINT_URL_KINESIS_VIDEO i: a:
KMS	k: AWS_ENDPOINT_URL_KMS i: a:
LakeFormation	l: AWS_ENDPOINT_URL_LAKEFORMATION t: a:

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數 Application Configuration file
Lambda	AWS_ENDPOINT_URL_LAMBDA
Launch Wizard	AWS_ENDPOINT_URL_LAUNCH_WIZARD
Lex Model Building Service	AWS_ENDPOINT_URL_LEX_MODEL_BUILDING_SERVICE
Lex Runtime Service	AWS_ENDPOINT_URL_LEX_RUNTIME_SERVICE
Lex Models V2	AWS_ENDPOINT_URL_LEX_MODELS_V2
Lex Runtime V2	AWS_ENDPOINT_URL_LEX_RUNTIME_V2
License Manager	AWS_ENDPOINT_URL_LICENSE_MANAGER

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
License Manager Linux Subscriptions	l: AWS_ENDPOINT_URL_LICENSE_MANAGER_LINUX_SUBSCRIPTIONS
License Manager User Subscriptions	l: AWS_ENDPOINT_URL_LICENSE_MANAGER_USER_SUBSCRIPTIONS
Lightsail	l: AWS_ENDPOINT_URL_LIGHTSAIL
Location	l: AWS_ENDPOINT_URL_LOCATION
CloudWatch Logs	c: AWS_ENDPOINT_URL_CLOUDWATCH_LOGS
CloudWatch Logs	c: AWS_ENDPOINT_URL_CLOUDWATCH_LOGS
LookoutEquipment	l: AWS_ENDPOINT_URL_LOOKOUTEQUIPMENT

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
LookoutMetrics	LookoutMetrics: AWS_ENDPOINT_URL_LOOKOUTMETRICS
LookoutVision	LookoutVision: AWS_ENDPOINT_URL_LOOKOUTVISION
m2	m2: AWS_ENDPOINT_URL_M2
Machine Learning	Machine Learning: AWS_ENDPOINT_URL_MACHINE_LEARNING
Macie2	Macie2: AWS_ENDPOINT_URL_MACIE2
ManagedBlockchain	ManagedBlockchain: AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN
ManagedBlockchain Query	ManagedBlockchain Query: AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN_QUERY
Marketplace Agreement	Marketplace Agreement: AWS_ENDPOINT_URL_MARKETPLACE_AGREEMENT

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
Marketplace Catalog	m: AWS_ENDPOINT_URL_MARKETPLACE_CATALOG c: g
Marketplace Deployment	m: AWS_ENDPOINT_URL_MARKETPLACE_DEPLOYMENT c: m:
Marketplace Entitlement Service	m: AWS_ENDPOINT_URL_MARKETPLACE_ENTITLEMENT_SERVICE c: e: v:
Marketplace Commerce Analytics	m: AWS_ENDPOINT_URL_MARKETPLACE_COMMERCE_ANALYTICS c: c: i:
MediaConnect	m: AWS_ENDPOINT_URL_MEDIACONNECT e:
MediaConvert	m: AWS_ENDPOINT_URL_MEDIACONVERT e:

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
MediaLive	media-live AWS_ENDPOINT_URL_MEDIALIVE
MediaPackage	media-package AWS_ENDPOINT_URL_MEDIAPACKAGE
MediaPackage Vod	media-package-vod AWS_ENDPOINT_URL_MEDIAPACKAGE_VOD
MediaPackageV2	media-package-v2 AWS_ENDPOINT_URL_MEDIAPACKAGEV2
MediaStore	media-store AWS_ENDPOINT_URL_MEDIASTORE
MediaStore Data	media-store-data AWS_ENDPOINT_URL_MEDIASTORE_DATA
MediaTailor	media-tailor AWS_ENDPOINT_URL_MEDIATAILOR
Medical Imaging	medical-imaging AWS_ENDPOINT_URL_MEDICAL_IMAGING

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
MemoryDB	m: AWS_ENDPOINT_URL_MEMORYDB
Marketplace Metering	m: AWS_ENDPOINT_URL_MARKETPLACE_METERING c: n:
Migration Hub	m: AWS_ENDPOINT_URL_MIGRATION_HUB _I
mgn	m: AWS_ENDPOINT_URL_MGN
Migration Hub Refactor Spaces	m: AWS_ENDPOINT_URL_MIGRATION_HUB_REFACTOR_SPACES c: e:
MigrationHub Config	m: AWS_ENDPOINT_URL_MIGRATIONHUB_CONFIG h: g
MigrationHubOrchestrator	m: AWS_ENDPOINT_URL_MIGRATIONHUBORCHESTRATOR h: t:

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
MigrationHubStrategy	m: AWS_ENDPOINT_URL_MIGRATIONHUBSTRATEGY h g)
Mobile	m: AWS_ENDPOINT_URL_MOBILE
mq	m: AWS_ENDPOINT_URL_MQ
MTurk	m: AWS_ENDPOINT_URL_MTURK
MWAA	m: AWS_ENDPOINT_URL_MWAA
Neptune	n: AWS_ENDPOINT_URL_NEPTUNE
Neptune Graph	n: AWS_ENDPOINT_URL_NEPTUNE_GRAPH r:
neptunedata	n: AWS_ENDPOINT_URL_NEPTUNEDATA t:
Network Firewall	n: AWS_ENDPOINT_URL_NETWORK_FIREWALL i:
NetworkManager	n: AWS_ENDPOINT_URL_NETWORKMANAGER n:

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_ <SERVICE> 環境變數
NetworkMonitor	n: AWS_ENDPOINT_URL_NETWORKMONITOR
nimble	n: AWS_ENDPOINT_URL_NIMBLE
OAM	o: AWS_ENDPOINT_URL_OAM
Omics	or: AWS_ENDPOINT_URL_OMICS
OpenSearch	o: AWS_ENDPOINT_URL_OPENSEARCH
OpenSearchServerless	o: AWS_ENDPOINT_URL_OPENSEARCHSERVERLESS
OpsWorks	o: AWS_ENDPOINT_URL_OPSWORKS
OpsWorksCM	o: AWS_ENDPOINT_URL_OPSWORKSCM
Organizations	o: AWS_ENDPOINT_URL_ORGANIZATIONS
OSIS	o: AWS_ENDPOINT_URL_OSIS

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
Outposts	AWS_ENDPOINT_URL_OUTPOSTS
p8data	AWS_ENDPOINT_URL_P8DATA
p8data	AWS_ENDPOINT_URL_P8DATA
Panorama	AWS_ENDPOINT_URL_PANORAMA
Payment Cryptography	AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY
Payment Cryptography Data	AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY_DATA
Pca Connector Ad	AWS_ENDPOINT_URL_PCA_CONNECTOR_AD
Personalize	AWS_ENDPOINT_URL_PERSONALIZE

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
Personalize Events	p: AWS_ENDPOINT_URL_PERSONALIZE_EVENTS z:
Personalize Runtime	p: AWS_ENDPOINT_URL_PERSONALIZE_RUNTIME z: e
PI	p: AWS_ENDPOINT_URL_PI
Pinpoint	p: AWS_ENDPOINT_URL_PINPOINT
Pinpoint Email	p: AWS_ENDPOINT_URL_PINPOINT_EMAIL er
Pinpoint SMS Voice	p: AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE sr
Pinpoint SMS Voice V2	p: AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE_V2 sr _\'
Pipes	p: AWS_ENDPOINT_URL_PIPES

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_ <SERVICE> 環境變數
Polly	polly: AWS_ENDPOINT_URL_POLLY
Pricing	pricing: AWS_ENDPOINT_URL_PRICING
PrivateNetworks	private-networks: AWS_ENDPOINT_URL_PRIVATENETWORKS
Proton	proton: AWS_ENDPOINT_URL_PROTON
QBusiness	qbusiness: AWS_ENDPOINT_URL_QBUSINESS
QConnect	qconnect: AWS_ENDPOINT_URL_QCONNECT
QLDB	qldb: AWS_ENDPOINT_URL_QLDB
QLDB Session	qldb-session: AWS_ENDPOINT_URL_QLDB_SESSION
QuickSight	quicksight: AWS_ENDPOINT_URL_QUICKSIGHT
RAM	ram: AWS_ENDPOINT_URL_RAM
rbin	rbin: AWS_ENDPOINT_URL_RBIN

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
RDS	AWS_ENDPOINT_URL_RDS
RDS Data	AWS_ENDPOINT_URL_RDS_DATA
Redshift	AWS_ENDPOINT_URL_REDSHIFT
Redshift Data	AWS_ENDPOINT_URL_REDSHIFT_DATA
Redshift Serverless	AWS_ENDPOINT_URL_REDSHIFT_SERVERLESS
Rekognition	AWS_ENDPOINT_URL_REKOGNITION
repostspace	AWS_ENDPOINT_URL_REPOSTSPACE
resiliencehub	AWS_ENDPOINT_URL_RESILIENCEHUB

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
Resource Explorer 2	AWS_ENDPOINT_URL_RESOURCE_EXPLORER_2
Resource Groups	AWS_ENDPOINT_URL_RESOURCE_GROUPS
Resource Groups Tagging API	AWS_ENDPOINT_URL_RESOURCE_GROUPS_TAGGING_API
RoboMaker	AWS_ENDPOINT_URL_ROBOMAKER
RolesAnywhere	AWS_ENDPOINT_URL_ROLESEANYWHERE
Route 53	AWS_ENDPOINT_URL_ROUTE_53
Route53 Recovery Cluster	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CLUSTER

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數 Configuration file
Route53 Recovery Control Config	i: AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CONTROL_CONFIG
Route53 Recovery Readiness	i: AWS_ENDPOINT_URL_ROUTE53_RECOVERY_READINESS
Route 53 Domains	i: AWS_ENDPOINT_URL_ROUTE_53_DOMAINS
Route53Resolver	i: AWS_ENDPOINT_URL_ROUTE53RESOLVER
RUM	i: AWS_ENDPOINT_URL_RUM
S3	s: AWS_ENDPOINT_URL_S3
S3 Control	s: AWS_ENDPOINT_URL_S3_CONTROL
S3Outposts	s: AWS_ENDPOINT_URL_S3OUTPOSTS

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
SageMaker	s: AWS_ENDPOINT_URL_SAGEMAKER
SageMaker A2I Runtime	s: AWS_ENDPOINT_URL_SAGEMAKER_A2I_RUNTIME _i
Sagemaker Edge	s: AWS_ENDPOINT_URL_SAGEMAKER_EDGE _e
SageMaker FeatureStore Runtime	s: AWS_ENDPOINT_URL_SAGEMAKER_FEATURESTORE_RUNTIME _f _i
SageMaker Geospatial	s: AWS_ENDPOINT_URL_SAGEMAKER_GEOSPATIAL _g _a
SageMaker Metrics	s: AWS_ENDPOINT_URL_SAGEMAKER_METRICS _m
SageMaker Runtime	s: AWS_ENDPOINT_URL_SAGEMAKER_RUNTIME _r

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_ <SERVICE>	環境變數
savingsplans	savingsplans	AWS_ENDPOINT_URL_SAVINGSPLANS
Scheduler	scheduler	AWS_ENDPOINT_URL_SCHEDULER
schemas	schemas	AWS_ENDPOINT_URL_SCHEMAS
SimpleDB	simpledb	AWS_ENDPOINT_URL_SIMPLEDB
Secrets Manager	secretsmanager	AWS_ENDPOINT_URL_SECRETS_MANAGER
SecurityHub	securityhub	AWS_ENDPOINT_URL_SECURITYHUB
SecurityLake	securitylake	AWS_ENDPOINT_URL_SECURITYLAKE
ServerlessApplicationRepository	serverlessapplicationrepository	AWS_ENDPOINT_URL_SERVERLESSAPPLICATIONREPOSITORY

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_ <SERVICE> 環境變數
Service Quotas	shared: AWS_ENDPOINT_URL_SERVICE_QUOTAS
Service Catalog	shared: AWS_ENDPOINT_URL_SERVICE_CATALOG
Service Catalog AppRegistry	shared: AWS_ENDPOINT_URL_SERVICE_CATALOG_APP_REGISTRY
ServiceDiscovery	shared: AWS_ENDPOINT_URL_SERVICEDISCOVERY
SES	shared: AWS_ENDPOINT_URL_SES
SESV2	shared: AWS_ENDPOINT_URL_SESV2
Shield	shared: AWS_ENDPOINT_URL_SHIELD
signer	shared: AWS_ENDPOINT_URL_SIGNER
SimSpaceWeaver	shared: AWS_ENDPOINT_URL_SIMSPACEWEAVER

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
SMS	shared: AWS_ENDPOINT_URL_SMS
Snow Device Management	shared: AWS_ENDPOINT_URL_SNOW_DEVICE_MANAGEMENT
Snowball	shared: AWS_ENDPOINT_URL_SNOWBALL
SNS	shared: AWS_ENDPOINT_URL_SNS
SQS	shared: AWS_ENDPOINT_URL_SQS
SSM	shared: AWS_ENDPOINT_URL_SSM
SSM Contacts	shared: AWS_ENDPOINT_URL_SSM_CONTACTS
SSM Incidents	shared: AWS_ENDPOINT_URL_SSM_INCIDENTS
Ssm Sap	shared: AWS_ENDPOINT_URL_SSM_SAP
SSO	shared: AWS_ENDPOINT_URL_SSO

serviceId	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_ <SERVICE> 環境變數
SSO Admin	aws: AWS_ENDPOINT_URL_SSO_ADMIN
SSO OIDC	aws: AWS_ENDPOINT_URL_SSO_OIDC
SFN	aws: AWS_ENDPOINT_URL_SFN
Storage Gateway	aws: AWS_ENDPOINT_URL_STORAGE_GATEWAY
STS	aws: AWS_ENDPOINT_URL_STS
SupplyChain	aws: AWS_ENDPOINT_URL_SUPPLYCHAIN
Support	aws: AWS_ENDPOINT_URL_SUPPORT
Support App	aws: AWS_ENDPOINT_URL_SUPPORT_APP
SWF	aws: AWS_ENDPOINT_URL_SWF
synthetics	aws: AWS_ENDPOINT_URL_SYNTHETICS

serviceId	共用的服務識別碼金鑰 AWS_EndpointURL_ environment	AWS_ENDPOINT_URL_<SERVICE> 環境變數
Textract	t:	AWS_ENDPOINT_URL_TEXTRACT
Timestream InfluxDB	t: m: b:	AWS_ENDPOINT_URL_TIMESTREAM_INFLUXDB
Timestream Query	t: m:	AWS_ENDPOINT_URL_TIMESTREAM_QUERY
Timestream Write	t: m:	AWS_ENDPOINT_URL_TIMESTREAM_WRITE
tnb	t:	AWS_ENDPOINT_URL_TNB
Transcribe	t: e:	AWS_ENDPOINT_URL_TRANSCRIBE
Transfer	t:	AWS_ENDPOINT_URL_TRANSFER
Translate	t:	AWS_ENDPOINT_URL_TRANSLATE
TrustedAdvisor	t: v:	AWS_ENDPOINT_URL_TRUSTEDADVISOR

serviceId	共 用 的 服 務 識 別 碼 金 鑰 A C C e s s f i l e	AWS_ENDPOINT_URL_<SERVICE>	環境變數
VerifiedPermissions	ve	AWS_ENDPOINT_URL_VERIFIEDPERMISSIONS	
Voice ID	vi	AWS_ENDPOINT_URL_VOICE_ID	
VPC Lattice	vp	AWS_ENDPOINT_URL_VPC_LATTICE	
WAF	wi	AWS_ENDPOINT_URL_WAF	
WAF Regional	wi	AWS_ENDPOINT_URL_WAF_REGIONAL	
WAFV2	wi	AWS_ENDPOINT_URL_WAFV2	
WellArchitected	wi	AWS_ENDPOINT_URL_WELLARCHITECTED	
Wisdom	wi	AWS_ENDPOINT_URL_WISDOM	
WorkDocs	wi	AWS_ENDPOINT_URL_WORKDOCS	
WorkLink	wi	AWS_ENDPOINT_URL_WORKLINK	

<code>serviceId</code>	共用的服務識別碼金鑰 AWS_ENDPOINT_URL_ <code><SERVICE></code> 環境變數
WorkMail	<code>wc: AWS_ENDPOINT_URL_WORKMAIL</code>
WorkMailMessageFlow	<code>wc: AWS_ENDPOINT_URL_WORKMAILMESSAGEFLOW</code> <code>e: w</code>
WorkSpaces	<code>wc: AWS_ENDPOINT_URL_WORKSPACES</code> <code>s</code>
WorkSpaces Thin Client	<code>wc: AWS_ENDPOINT_URL_WORKSPACES_THIN_CLIENT</code> <code>s_</code> <code>i:</code>
WorkSpaces Web	<code>wc: AWS_ENDPOINT_URL_WORKSPACES_WEB</code> <code>s_</code>
XRay	<code>x: AWS_ENDPOINT_URL_XRAY</code>

智慧型組態預設

使用智慧型組態預設值功能，AWS SDKs 可以為其他組態設定提供預先定義、最佳化的預設值。

使用下列項目設定此功能：

defaults_mode-共享 AWS config檔案設定, AWS_DEFAULTS_MODE-環境變量, aws.defaultsMode-JVM 系統屬性：僅限爪哇/科特林

使用此設定，您可以選擇與應用程式架構保持一致的模式，然後為您的應用程式提供最佳化的預設值。如果 AWS SDK 設置具有明確設置的值，那麼該值始終優先。如果 AWS SDK 設置沒有明確設置的值，並且 defaults_mode 不等於傳統，那麼這個功能可以為針對您的應用程序優化的各種設置提供不同的默認值。設定可能包括下列項目：HTTP 通訊設定、重試行為、服務地區端點設定，以及可能的任何 SDK 相關組態。使用此功能的客戶可以取得針對常見使用案例量身打造的新組態預設值。如果您 defaults_mode 不等於 legacy，建議您在升級時對應用程式執行測試 SDK，因為提供的預設值可能會隨著最佳作法的發展而變更。

預設值：legacy

注意：新的主要版 SDKs 本預設為 standard。

有效值：

- legacy— 提供在建立之前有所不同 SDK 且已存在的預設設定 defaults_mode。
- standard— 提供在大多數情況下應安全執行的最新建議預設值。
- in-region— 建立在標準模式上，包括針對呼叫的應用程式量身打造的最佳化 AWS 服務 從同一個 AWS 區域。
- cross-region— 建立在標準模式上，包括針對呼叫的應用程式量身打造的最佳化 AWS 服務 在不同的區域。
- mobile— 建立在標準模式上，包括針對移動應用程序量身定制的優化。
- auto— 基於標準模式構建，並包括實驗功能。SDK 嘗試探索執行階段環境以自動判斷適當的設定。auto 檢測是基於啟發式的，不提供 100% 的準確性。如果無法確定運行時環境，則使用 standard 模式。auto 偵測可能會查詢 [執行個體中繼資料](#)，這可能會導致延遲。如果啟動延遲對您的應用程式很重要，我們建議您 defaults_mode 改為選擇明確的。

在 config 檔案中設定此值的範例：

```
[default]
defaults_mode = standard
```

下列參數可能會根據選取項目進行最佳化 defaults_mode：

- retryMode— 指定嘗試重 SDK 試的方式。請參閱 [重試行為](#)。
- stsRegionalEndpoints— 指定如何 SDK 決定 AWS 服務 它用來與之交談的端點 AWS Security Token Service (AWS STS)。請參閱 [AWS STS 區域端點](#)。

- `s3UsEast1RegionalEndpoints`— 指定如何SDK決定 AWS 它用來與該us-east-1區域的 Amazon S3 通訊的服務端點。
- `connectTimeoutInMillis`-在套接字上進行初始連接嘗試後，超時之前的時間量。如果用戶端沒有收到連線握手的完成，用戶端就會放棄並失敗作業。
- `tlsNegotiationTimeoutInMillis`— TLS 握手從傳送CLIENTHELLO訊息到用戶端和伺服器已完全協商加密和交換金鑰之後所需的時間上限。

每個設定的預設值會根據您為應用程式`defaults_mode`選取的項目而變更。這些值目前設定如下 (可能會變更)：

參數	standard 模式	in-region 模式	cross-region 模式	mobile 模式
<code>retryMode</code>	standard	standard	standard	standard
<code>stsRegionalEndpoints</code>	regional	regional	regional	regional
<code>s3UsEast1RegionalEndpoints</code>	regional	regional	regional	regional
<code>connectTimeoutInMillis</code>	3100	1100	3100	30000
<code>tlsNegotiationTimeoutInMillis</code>	3100	1100	3100	30000

例如，如果`defaults_mode`您選取的是`standard`，則`standard`會為 `retry_mode` (從有效`retry_mode`選項中) 指派的`regional`值，並為 `stsRegionalEndpoints` (從有效`stsRegionalEndpoints`選項) 指派的值。

兼容性 AWS SDKs

下列項目SDKs支援本主題中描述的功能和設定。注意到任何部分例外情況。任何JVM系統屬性設定都受到 AWS SDK for Java 和 適用於 Kotlin 的 AWS SDK 只有。

SDK	支援	備註或更多資訊
AWS CLI V2	否	
SDK對於 C ++	是	未最佳化的參數： <code>stsRegionalEndpoints s3UsEast1 RegionaleEndpoints</code> 、 <code>tlsNegotiationTimeoutInMillis</code> 。
SDK對於轉到 V2 (1.x)	是	未最佳化的參數： <code>retryMode stsRegionalEndpoints</code> 、 <code>s3UsEast1 RegionalEndpoints</code> 。
SDK對於圍棋 1.x (V1)	否	
SDK對於爪哇 2.x	是	未最佳化的參數： <code>stsRegionalEndpoints</code> 。
SDK對於爪哇 1.x	否	
SDK對於 JavaScript 3. x	是	未最佳化的參數： <code>stsRegionalEndpoints s3UsEast1 RegionaleEndpoints</code> 、 <code>tlsNegotiationTimeoutInMillis</code> 。 <code>connectTimeoutInMillis</code> 被稱為 <code>connectionTimeout</code> 。
SDK對於 JavaScript 2.x	否	

SDK	支援	備註或更多資訊
SDK對於科特林	否	
SDK為 .NET 3.x	是	未最佳化的參數：connectTimeoutInMillis、tlsNegotiationTimeoutInMillis。
SDK對於 PHP 3.x	是	未最佳化的參數：tlsNegotiationTimeoutInMillis。
SDK對於 Python (肉毒桿菌 3)	是	未最佳化的參數：tlsNegotiationTimeoutInMillis。
SDK對於紅寶石 3.x	是	
SDK對於銹	否	
SDK為斯威夫特	否	
用於的工具 PowerShell	是	未最佳化的參數：connectTimeoutInMillis、tlsNegotiationTimeoutInMillis。

AWS 一般執行期 (CRT) 程式庫

AWS 通用執行期 (CRT) 程式庫是 的基礎程式庫SDKs。CRT 是以 C 編寫的獨立套件的模組化系列。每個套件都為不同的必要功能提供良好的效能和最小的佔用空間。這些功能是常見且跨所有功能共用的，SDKs可提供更好的程式碼重複使用、最佳化和準確性。這些套件包括：

- [aws-labs/aws-c-auth](#)：AWS 用戶端身分驗證 (標準憑證提供者和簽署 (sigv4))
- [aws-labs/aws-c-cal](#)：密碼編譯原始類型、雜湊 (MD5、SHA256、SHA256HMAC)、簽署者、AES
- [aws-labs/aws-c-common](#)：基本資料結構、執行緒/同步原始類型、緩衝區管理、stdlib 相關函數
- [aws-labs/aws-c-compression](#)：壓縮演算法 (Huffman 編碼/解碼)
- [aws-labs/aws-c-event-stream](#)：事件串流訊息處理 (標頭、排除、承載、crc/追蹤器)、透過事件串流實作遠端程序呼叫 (RPC)
- [aws-labs/aws-c-http](#)：C99 實作 HTTP/1.1 和 HTTP/2 規格
- [aws-labs/aws-c-io](#)：通訊端 (TCP、UDP)、DNS、管道、事件迴路、頻道、SSL/TLS
- [aws-labs/aws-c-iot](#)：C99 實作 AWS IoT 雲端服務與 裝置整合
- [aws-labs/aws-c-mqtt](#)：適用於物聯網 (IoT的標準輕量型傳訊通訊協定
- [aws-labs/aws-c-s3](#)：用於與 Amazon S3 服務通訊的 C99 程式庫實作，旨在最大限度地提高高頻寬 Amazon EC2執行個體的輸送量
- [aws-labs/aws-c-sdkutils](#)：用於剖析和管理 AWS 設定檔的公用程式程式庫
- [aws-labs/aws-checksums](#)：跨平台硬體加速CRC32c，CRC32並退至高效的軟體實作
- [aws-labs/aws-lc](#)：密碼 AWS 編譯團隊為 AWS 及其客戶維護的一般用途密碼編譯程式庫，以 Google BoringSSL 專案和 OpenSSL 專案的程式碼為基礎
- [aws-labs/s2n](#)：C99 實作 TLS/SSL 通訊協定，設計為小型且快速，以安全性為優先

除了 Go 和 Rust SDKs之外，所有 CRT 都可以使用。

CRT 相依性

這些CRT程式庫形成複雜的關係和相依性網。如果您需要CRT直接從來源建置，了解這些關係會很有幫助。不過，大多數使用者透過其語言 SDK (例如 AWS SDK C++ 或 AWS SDK Java) 或語言 IoT 裝置 SDK (例如 AWS IoT++ SDK 的 IoT 或 Java SDK 的 AWS IoT) 存取CRT功能。在下圖中，語

言CRT繫結方塊是指包裝特定語言CRT程式庫的套件SDK。這是表單的套件集合aws-crt-*，其中 '*' 是SDK語言（例如 [aws-crt-cpp](#)或 [aws-crt-java](#)）。

以下是CRT程式庫的階層相依性圖示。

AWS SDK 和工具維護政策

概觀

本文件概述了 AWS 軟體開發套件 (SDK) 和工具 (包括行動和 IoT SDK) 的維護政策，以及其基礎相依性。AWS 定期為 AWS SDK 和工具提供更新，這些更新可能包含對新的或更新的 AWS API、新功能、增強功能、錯誤修正、安全性修補程式或文件更新的支援。更新也可以解決具有相依性、語言執行階段和作業系統的變更。AWS SDK 發行版本會發佈至套件管理員 (例如 Maven NuGet、PyPI)，並且可在上做為原始程式碼使用。GitHub

我們建議使用者繼續使用 up-to-date 用 SDK 版本，以跟上最新功能、安全性更新和基礎相依性。不建議繼續使用不受支援的 SDK 版本，並由使用者自行決定。

版本控制

AWS SDK 發行版本採用 X.Y.Z 的形式，其中 X 代表主要版本。增加 SDK 的主要版本表示此 SDK 進行了重大而重大的變化，以支持該語言中的新習語和模式。當公共接口 (例如類，方法，類型等)，行為或語義發生變化時，會引入主要版本。應用程式必須更新，才能使用最新的 SDK 版本。請務必根據提供的升級準則仔細更新主要版本 AWS。

SDK 主要版本生命週期

主要 SDK 和工具版本的生命週期由 5 個階段組成，概述如下。

- 開發人員預覽 (階段 0)- 在此階段，SDK 不受支援，不應在生產環境中使用，並且僅用於早期存取和意見反應。future 的版本可能會引入重大變更。一旦將某個版本 AWS 識別為穩定的產品，它可能會將其標記為候選版本。除非出現重大錯誤，否則候選發布版本已準備好發布 GA，並將獲得全面 AWS 支持。
- 一般可用性 (GA) (階段 1)- 在此階段，完整支援 SDK。AWS 將提供常規 SDK 版本，其中包括對新服務的支援、現有服務的 API 更新，以及錯誤和安全性修正。對於工具，AWS 將提供包含新功能更新和錯誤修復的常規版本。AWS 將支持一個 SDK 的 GA 版本至少 24 個月。
- 維護公告 (第二階段) - AWS 將在 SDK 進入維護模式前至少 6 個月公佈。在此期間，SDK 將繼續獲得完整支援。通常，維護模式會在下一個主要版本轉換為 GA 的同時宣布。
- 維護 (階段 3) - 在維護模式下，AWS 限制 SDK 版本以僅解決重大錯誤修復和安全問題。SDK 不會收到新服務或現有服務的 API 更新，也不會更新以支援新區域。除非另有說明，否則維護模式的預設持續時間為 12 個月。

- 終止支援 (階段 4)- 當 SDK 終止 Support 時，將不再收到更新或發行版本。先前發佈的版本將繼續透過公開套件管理員提供，而程式碼仍會保持開啟狀態 GitHub。存放 GitHub 庫可能已封存。使用者可自行決定使用已達到 end-of-support 的 SDK。我們建議使用者升級至新的主要版本。

以下是 SDK 主要版本生命週期的視覺說明。請注意，下面顯示的時間表是說明性的，沒有約束力。

相依性週期

大多數 AWS SDK 都具有基礎依賴關係，例如語言運行時，操作系統或第三方庫和框架。這些相依性通常與語言社群或擁有該特定元件的廠商相關聯。每個社區或供應商都會為其產品發布自己的 end-of-support 時間表。

下列術語用於分類基礎第三方相依性：

- 操作系統 (OS): 例子包括 Amazon Linux AMI, Amazon Linux 2, 視窗 2008, 視窗 2012, 視窗 2016, 等等。
- 語言運行時：例子包括 Java 7，Java 8，Java 11，淨核心，淨標準，淨網絡 PCL 等
- 第三方庫/框架：示例包括 OpenSSL，.NET 框架 4.5，Java EE 等。

我們的政策是在社群或廠商結束對相依性的支援後，持續支援 SDK 相依性至少 6 個月。不過，此原則可能會因特定相依性而有所不同。

Note

AWS 保留在不增加主要 SDK 版本的情況下停止支持基礎依賴項的權利

溝通方式

維護公告以下列幾種方式傳達：

- 系統會向受影響的帳戶傳送電子郵件通知，宣布我們計劃終止對特定 SDK 版本的支援。電子郵件會概述到的路徑 end-of-support、指定行銷活動時間表，並提供升級指引。
- AWS SDK 文件 (例如 API 參考文件、使用者指南、SDK 產品行銷頁面和 GitHub README) 已更新，以指出行銷活動時間表，並提供升級受影響應用程式的指引。
- 發布了一篇 AWS 博客文章，其中概述了路徑 end-of-support，並重申了活動時間表。

- 棄用警告添加到 SDK，概述了 SDK 文檔的路徑 end-of-support 和鏈接。

若要查看 AWS SDK 和工具的可用主要版本清單，以及它們在維護生命週期中的位置，請參閱[版本支援](#)。

AWS SDKs和工具版本支援

下表顯示了可用的列表 AWS 軟體開發套件 (SDK) 主要版本，以及它們在維護生命週期中的位置，以及相關時間表。有關主要版本的生命週期的詳細信息 AWS SDKs和工具及其基礎依賴關係，請參閱[維護政策](#)。

SDK	主要版本	目前相位	一般可用日期	備註
AWS CLI	1.x	一般可用性	9/2/2013	
AWS CLI	2.x	一般可用性	2/10/2020	
SDK對於 C ++	1.x	一般可用性	2015 年 9 月 2 日	
SDK對於轉到 V2	V2	一般可用性	1/19/2021	
SDK為了去	1.x	維護	11/19/2015	詳情及日期請參閱 公告
SDK對於爪哇	1.x	維護	3/25/2010	詳情及日期請參閱 公告
SDK對於爪哇	2.x	一般可用性	11/20/2018	
SDK對於 JavaScript	1.x	終止 Support	5/6/2013	
SDK對於 JavaScript	2.x	維護	6/19/2014	詳情及日期請參閱 公告
SDK對於 JavaScript	3.x	一般可用性	12/15/2020	
SDK對於科特林	1.x	一般可用性	11/27/2023	
SDK對於 .NET	1.x	終止 Support	11/2009	
SDK對於 .NET	2.x	終止 Support	11/8/2013	

SDK	主要版本	目前相位	一般可用日期	備註
SDK對於 .NET	3.x	一般可用性	7/28/2015	
SDK對於 PHP	2.x	終止 Support	11/2/2012	
SDK對於 PHP	3.x	一般可用性	5/27/2015	
SDK對於 Python (肉毒桿菌 2)	1.x	終止 Support	7/13/2011	
SDK對於 Python (肉毒桿菌 3)	1.x	一般可用性	6/22/2015	
SDK對於 Python (肉毒核)	1.x	一般可用性	6/22/2015	
SDK對於紅寶石	1.x	終止 Support	7/14/2011	
SDK對於紅寶石	2.x	終止 Support	2015 年 2 月 15 日	
SDK對於紅寶石	3.x	一般可用性	8/29/2017	
SDK對於銹	1.x	一般可用性	11/27/2023	
SDK為斯威夫特	1.x	一般可用性	9/17/2024	
適用的工具 PowerShell	2.x	終止 Support	11/8/2013	
適用的工具 PowerShell	3.x	終止 Support	7/29/2015	
適用的工具 PowerShell	4.x	一般可用性	11/21/2019	

搜索未提及的SDK或工具？例如SDKs，加密SDKs、IoT 裝置和行動SDKs裝置不包含在本指南中。要查找有關這些其他工具的文檔，請參閱構建的[工具 AWS](#)。

的文件歷史記錄 AWS SDKs和工具參考指南

下表說明重要的新增和更新 AWS SDKs和工具參考指南。如需有關此文件更新的通知，您可以訂閱 RSS摘要。

變更	描述	日期
添加斯威夫特SDK到設置參考	向所有設置引用添加 Swift SDK 支持兼容 AWS SDKs表格。	2024年9月17日
SDK用於 Java 1.x 系統屬性	新增有關支援的JVM系統組態設定的詳細資料 AWS SDK for Java 1. x。	2024年5月30日
設定更新	新增JVM系統組態設定。	2024年3月27日
相容性表格更新	SDK支援相容性的更新、IAM 身分識別中心程序的更新。	2024年2月20日
容器認證更新。IMDS更新。	添加對 Amazon 的支持EKS。添加設置以禁用IMDSv1後備。	2023 年 12 月 29 日
請求壓縮	添加請求壓縮功能的設置。	2023 年 12 月 27 日
相容性表	更新的相容性表SDK和工具功能以包括 SDK Kotlin , SD K Rust 和 AWS Tools for PowerShell.	2023 年 12 月 10 日
驗證更新	支援的驗證方法SDKs和工具的更新。	2023 年 7 月 1 日
IAM最佳實踐更新	更新指南以符合最IAM佳做法。 如需詳細資訊，請參閱 IAM.	2023 年 2 月 27 日
SSO更新	新SSO權杖組態的SSO認證更新。	2022 年 11 月 19 日

設定更新	一般組態和 Amazon S3 多區域存取點支援表的更新。	2022 年 11 月 17 日
設定更新	更新IMDS客戶端和IMDS憑據的清晰度。環境變數的更新。	2022 年 11 月 4 日
更新歡迎頁面	宣布 Amazon CodeWhisperer。	2022 年 9 月 22 日
單一登入的服務名稱變更	更新以反映這一點 AWS SSO 現在被稱為 AWS IAM Identity Center.	2022 年 7 月 26 日
設定更新	對配置文件詳細信息和支持的設置進行次要更新。	2022 年 6 月 15 日
更新	本指南幾乎所有部分的大量更新。	2022 年 2 月 1 日
初始版本	本指南的第一個版本向公眾發布。	2020 年 3 月 13 日

AWS 詞彙表

如需最新的 AWS 術語，請參閱《AWS 詞彙表 參考》中的 [AWS 詞彙表](#)。

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。