



參考指南

AWS 軟體開發套件和工具



AWS 軟體開發套件和工具: 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，這些擁有者可能隸屬於 Amazon，或與 Amazon 有合作關係，或由 Amazon 贊助。

Table of Contents

AWS SDK 和工具參考指南	1
開發者資源	2
工具包遙測通知	2
組態	4
共用config和credentials檔案	5
描述檔	5
配置文件的格式	6
認證檔案的格式	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	11
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	13
身分驗證與存取	15
AWS 建構家 ID	16
IAM Identity Center 驗證	16
使用 IAM 身分中心設定程式設計存取	17
瞭解 IAM 身分中心身分驗證	19
IAM Roles Anywhere	22
步驟 1：隨處設定 IAM 角色	23
步驟 2：隨處使用 IAM 角色	23
擔任角色	24
假設 IAM 角色	24
與網絡身份或 OpenID Connect 聯盟	25
AWS 存取金鑰	27
使用短期憑證	27
使用長期認證	27
短期憑證	28
長期認證	29
Amazon EC2 執行個體的 IAM 角色	32
建立 IAM 角色	32

啟動 Amazon EC2 執行個體並指定您的 IAM 角色	32
Connect 至 EC2 執行個體	33
在 EC2 執行個體上執行範例應用程式	33
設定參考	34
建立服務用戶端	34
設定的優先順序	34
Config檔案設定清單	35
Credentials檔案設定清單	38
環境變數清單	39
JVM 系統屬性清單	43
標準化認證提供	46
憑證提供者鏈	46
AWS 存取金鑰	47
假設角色提供者	50
容器供應商	55
IAM 身分中心提供者	58
國際證券供應商	64
流程提供者	68
標準化功能	71
申請識別碼	72
Amazon EC2 執行個體中繼資料	74
Amazon S3 存取點	76
Amazon S3 多區域存取點	78
AWS 區域	80
AWS STS 區域化端點	83
雙堆疊和 FIPS 端點	85
端點探索	87
一般配置	89
法團客戶端	92
重試行為	95
請求壓縮	100
服務特定端點	102
智慧型組態預設	153
通用運行時	158
CRT 相依性	158
維護政策	160

概觀	160
版本控制	160
SDK 主要版本生命週期	160
相依性週期	161
溝通方式	161
版本支援矩陣	163
文件歷史紀錄	165
AWS 詞彙表	167
.....	clxviii

AWS SDK 和工具參考指南

許多 SDK 和工具共用一些通用功能，無論是透過共用設計規格還是透過共用程式庫。

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 SDK 和工具。
- [身分驗證與存取](#)— 建立程式碼或工具在開發 AWS 時的驗證方式。AWS 服務
- [設定參考](#)— 可用於驗證和組態的所有標準化設定的參考資料。
- [AWS通用執行階段 \(CRT\) 程式庫](#)— 幾乎所有 SDK 都可用的 AWS 共用通用執行階段 (CRT) 程式庫概觀。
- [AWS SDK 和工具維護政策](#)涵蓋 AWS 軟體開發套件 (SDK) 和工具的維護政策和版本控制，包括行動和物聯網 (IoT) SDK，以及其基礎相依性。

本 AWS SDK 和工具參考指南旨在成為適用於多個 SDK 和工具的信息基礎。除了此處提供的任何資訊外，還應使用您正在使用的 SDK 或工具的特定指南。以下是 SDK 和工具，其中包含本指南中的相關材料部分：

如果您正在使用：	本指南適合您的相關章節包括：
<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 Cloud Development Kit (AWS CDK) 開發人員指南 • AWS Serverless Application Model 開發人員指南 • AWS Toolkit for Eclipse 使用者指南 • AWS Toolkit for JetBrains 使用者指南 • AWS Toolkit for Visual Studio 使用者指南 • AWS Toolkit for Visual Studio Code 使用者指南 	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 Command Line Interface 使用者指南 • AWS SDK for C++ 開發人員指南 	組態 身分驗證與存取

如果您正在使用：

- [AWS SDK for Go 開發人員指南](#)
- [AWS SDK for Java 開發人員指南](#)
- [AWS SDK for JavaScript 開發人員指南](#)
- [適用於 Kotlin 的 AWS SDK](#)
- [AWS SDK for .NET 開發人員指南](#)
- [AWS SDK for PHP 開發人員指南](#)
- [AWS 開始使用軟體開發套件](#)
- [AWS SDK for Ruby 開發人員指南](#)
- [適用於 Rust 的 AWS SDK](#)
- [適用於 Swift 的 AWS SDK](#)
- [AWS Tools for Windows PowerShell 使用者指南](#)

本指南適合您的相關章節包括：

- [設定參考](#)
- [AWS 通用執行階段 \(CRT\) 程式庫](#)
- [AWS SDK 和工具維護政策](#)
- [AWS SDK 和工具版本支援對照表](#)

開發者資源

如需可協助您開發應用程式的工具概觀 AWS，請參閱[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、擴充和操作 AWS 應用程式。為了加快建置速度 AWS，支援 Amazon Q 的模型已透過高品質 AWS 內容進行擴充，以產生更完整、可操作且參考的答案。如需詳細資訊，請參閱[什麼是 Amazon Q 開發人員？](#) 在 Amazon Q 開發人員用戶指南中。

工具包遙測通知

AWS 整合式開發環境 (IDE) 工具組是可從 IDE 存取 AWS 服務的外掛程式和擴充功能。如需有關每個 IDE 工具組的詳細資訊，請參閱上表中的「工具組使用者指南」。

AWS IDE Toolkit 可能會收集並儲存用戶端遙測資料，以通知 future AWS Toolkit 版本的決策。收集到的資料會量化您對 AWS 工具組的使用情況。

若要深入了解所有 AWS IDE 工具組收集的遙測資料，請參閱 Github 存放[庫中的通用定義 .JSON](#) 文件。aws-toolkit-common

如需每個 AWS IDE 工具組所收集之遙測資料的詳細資訊，請參考下列工具組 Github 存放庫中 AWS 的資源文件：

- [AWS Toolkit for Visual Studio](#)
- [AWS Toolkit for Visual Studio Code](#)
- [AWS Toolkit for JetBrains](#)

AWS Toolkit 中可存取的某些 AWS 服務可能會收集其他用戶端遙測資料。如需每個個別 AWS 服務所收集之資料類型的詳細資訊，請參閱您感興趣之特定服務的[AWS 說明文件](#)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 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](#)
- [環境變量支持](#)
- [JVM 系統屬性支援](#)

共用config和credentials檔案

共用檔credentials案 AWS config和檔案包含一組設定檔。設定檔是由 ()、AWS SDK 和其他工具所使用的一組組態設定 AWS Command Line Interface (以鍵值配對表示AWS CLI)。設定值會附加至設定檔，以便在使用該設定檔時配置 SDK/ 工具的某些方面。這些文件是「共享」的，因為這些值會影響用戶本地環境中的任何應用程式，進程或 SDK。

共用credentials檔案config和檔案都是僅包含 ASCII 字元 (UTF-8 編碼) 的純文字檔案。他們採取什麼通常被稱為 [INI 文件](#) 的形式。

描述檔

共用config和credentials檔案中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，以便在不同的開發環境中套用。

如果未指定特定的具名[default]描述檔，設定檔會包含 SDK 或工具作業所使用的值。您也可以建立可依名稱明確參照的個別設定檔。每個設定檔都可以根據您的應用程式和案例需要使用不同的設定和值。

Note

[default]只是一個未命名的配置文件。此設定檔之所以命名，default因為如果使用者未指定設定檔，它是 SDK 使用的預設設定檔。它不會為其他設定檔提供繼承的預設值。如果您在[default]描述檔中設定某些內容，但未在具名描述檔中設定，則在使用具名描述檔時不會設定該值。

設定具名的設定檔

[default]縱斷面和多個具名輪廓可以存在於同一個檔案中。執行程式碼時，請使用下列設定來選取 SDK 或工具要使用的設定檔設定。也可以在程式碼中選取設定檔，或在使用 AWS CLI。

透過設定下列其中一項來設定此功能：

AWS_PROFILE-環境變量

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 指令都會使用該設定檔中的設定。

通過命令行設置環境變量的 Linux/macOS 示例：

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

通過命令行設置環境變量的 Windows 示例：

```
setx AWS_PROFILE "my_default_profile_name"
```

aws.profile-JVM 系統屬性

對於 JVM 上的 Kotlin 的 SDK 以及適用於 Java 2.x 的 SDK，您可以[設置系統屬性](#) `aws.profile`。SDK 建立服務用戶端時，會使用具名設定檔中的設定，除非程式碼中的設定遭到覆寫。適用於 Java 1.x 的開發套件不支援此系統屬性。

配置文件的格式

該config文件被組織成幾個部分。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。

該config文件是使用以下格式的純文本文件：

- 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。

剖面類型

橫斷面定義是將名稱套用至設定集合的一行。剖面定義行以方括號 (`[]`) 開始和結束。在括號內，有一個區段類型識別元和區段的自訂名稱。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**

剖面定義線範例：`[default]`

`[default]`是唯一不需要profile截面標識符的輪廓。

下列範例顯示具有`[default]`設定config檔的基本檔案。它設[region](#)置的設置。在遇到其他橫斷面定義之前，遵循此直線的所有設定都是此縱斷面的一部分。

```
[default]
```

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

區段類型：**profile**

剖面定義線範例：`[profile dev]`

`profile`區段定義行是具名的組態群組，您可以套用於不同的開發案例。若要更好地瞭解命名的設定檔，請參閱上一節「設定檔」。

以下範例展示了具有`profile`剖面定義線和名為的具名縱斷面的`config`檔案`foo`。在遇到另一個橫斷面定義之前，遵循此行的所有設定都是此具名縱斷面的一部分。

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

某些設定有自己的巢狀子設定群組，例如以下範例中的`s3`設定和子設定。藉由縮排一或多個空格，將子設定與群組相關聯。

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

區段類型：**sso-session**

剖面定義線範例：`[sso-session my-sso]`

`sso-session`剖面定義行會命名一組設定，您用來設定設定檔以解析 AWS 身分證明 AWS IAM Identity Center。如需設定單一登入驗證的詳細資訊，請參閱[IAM Identity Center 驗證](#)。配置文件通過鍵值對鏈接到一個`sso-session`部分，其中`sso-session`是鍵，您的`sso-session`部分的名稱是值，例如`sso-session = <name-of-sso-session-section>`。

下列範例會設定一個設定檔，使用「`my-sso`」的權杖取得「111122223333」帳戶中「`SampleRole`」IAM 角色的短期 AWS 登入資料。「`my-sso`」`sso-session`部分在部分中使用`profile sso-session`密鑰按名稱引用。

```
[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**

剖面定義線範例：`[services dev]`

Note

本 `services` 節支援服務特定端點自訂，且僅適用於包含此功能的 SDK 和工具。若要查看此功能是否適用於您的 SDK，請參閱 [與 AWS 開發套件的相容性](#) 取得服務特定端點。

`services` 區段定義行會命名一組設定要求 AWS 服務的自訂端點的設定。配置文件通過鍵值對鏈接到一個 `services` 部分，其中 `services` 是鍵，您的 `services` 部分的名稱是值，例如 `services = <name-of-services-section>`。

該 `services` 部分由行進一步分成子部 `<SERVICE> =` 分，其中 `<SERVICE>` 是標 AWS 服務 識鍵。標 AWS 服務 識符基於 API 模型的 `serviceId` 方法是用下劃線替換所有空格並將所有字母小寫。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。

下列範例會使用 `services` 定義來設定端點，以用於僅對 Amazon DynamoDB 服務發出的要求。在 `"local-dynamodb"` `services` 區段中依名稱使用 `services` 金鑰參照 `profile` 區段。AWS 服務 識別碼金鑰為 `dynamodb`。服 Amazon DynamoDB 務子區段從該行開始 `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。

```
[profile dev]
services = local-dynamodb

[services local-dynamodb]
dynamodb =
  endpoint_url = http://localhost:8000
```

如需自訂端點組態的詳細資訊，請參閱 [服務特定端點](#)。

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[環境變量支持](#)。

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。

環境變量支持

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 SDK 支援的環境變數清單，請參閱[環境變數清單](#)。

選項的優先順序

- 如果您使用其環境變數指定設定，它會覆寫從共用AWSconfig和檔案中設定credentials檔載入的任何值。

- 如果您使用指AWS CLI命令行上的參數來指定設定，它會取代組態檔案中對應環境變數或設定檔中的任何值。

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
```

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。

PowerShell

```
PS C:\> $Env:AWS_ACCESS_KEY_ID="AKIAIOSFODNN7EXAMPLE"
PS C:\> $Env:AWS_SECRET_ACCESS_KEY="wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY"
PS C:\>
  \> $Env:AWS_SESSION_TOKEN="AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJ0gQs8IZZaIv2BXIa2R40lgk"
PS C:\> $Env:AWS_REGION="us-west-2"
```

如果您在 PowerShell 提示中設定環境變數 (如前面範例所示)，則只會儲存目前工作階段持續時間的值。若要讓環境變數設定在所有 PowerShell 和 [命令提示字元] 工作階段中持續存放，請使用 [控

制台] 中的系統應用程式來儲存。或者，您可以將變量添加到您的個 PowerShell 人資料中，為所有 future 的 PowerShell 會話設置該變量。如需有關儲存環境變數或在工作階段中保留環境變數的詳細資訊，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。視您的容器而定，您可以針對在這些容器中執行的程式碼使用不同的策略來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](#)。

在無服務器框架中，您通常可以在環境設置下的提供者密鑰下的 `serverless.yml` 文件中設置 SDK 環境變量。如需 `serverless.yml` 檔案的相關資訊，請參閱無伺服器架構文件中的[一般功能設定](#)。

無論您使用哪種機制來設定容器環境變數，都有一些是由容器保留的，例如在[定義的執行階段環境變數](#)中針對 Lambda 記錄的那些機制。請務必查閱您正在使用的容器的官方文檔，以確定如何處理環境變量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

在開發AWS時，您必須建立程式碼的驗證方式。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。

本機執行程式碼的驗證選項 (不在中AWS)

- [IAM Identity Center 驗證](#)— 為了安全性最佳做法，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。您可以在中建立使用者AWS IAM Identity Center、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[若要檢查您的區域是否支援 IAM 身分中心，請參閱 AWS IAM Identity CenterAmazon Web Services 一般參考](#)。
- [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，以處理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。
- [擔任角色](#)— 您可以假設 IAM 角色暫時存取您可能無法存取的AWS資源。
- [AWS 存取金鑰](#)— 可能不太方便或可能增加AWS資源安全風險的其他選項。

在AWS環境中執行之程式碼的驗證選項

- [針對 Amazon EC2 執行個體使用 IAM 角色](#)— 使用 IAM 角色在 Amazon EC2 執行個體上安全地執行您的應用程式。
- 您可以透過下列方式以程式設計方式與AWS使用 IAM 身分中心進行互動：
 - 用[AWS CloudShell](#)於從控制台運行AWS CLI命令。
 - 用於[AWS Cloud9](#)開始AWS使用具有AWS資源的整合式開發環境 (IDE) 程式設計。
 - 若要為軟體開發團隊試用雲端協作空間，請考慮使用 [Amazon CodeCatalyst](#)。

透過 Web 型身分識別提供者進行驗證-行動裝置或用戶端 Web 應用

如果您要建立需要存取權的行動應用程式或以用戶端為基礎的 Web 應用程式AWS，請建置您的應用程式，使其使用 Web 身分同盟動態要求臨時AWS安全性登入資料。

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。相反，應用程序用戶可以使用知名的外部身份提供者 (IdP) 登錄，例如使用亞馬遜，Facebook，谷歌或任何其他 OpenID Connect (OIDC) 兼容的 IdP 登錄。他們可以收到身分驗證權杖，然後交換取得用於 AWS 的暫時安全憑證，並藉其對應到可使用您的 AWS 帳戶 下資源之許可的 IAM 角色。

若要瞭解如何針對 SDK 或工具進行設定，請參閱[與網絡身份或 OpenID Connect 聯盟](#)。

對於行動應用程式，請考慮使用 Amazon Cognito。Amazon Cognito 充當身分識別代理人，為您完成大部分聯合會工作。如需詳細資訊，請參閱 IAM 使用者指南中的[行動應用程式使用 Amazon Cognito](#)。

關於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 \(使用者、使用者群組和角色\)](#) — 瞭解AWS。
- [IAM 中的安全最佳實務](#) — 根據[共享責任模型](#)開發AWS應用程式時應遵循的安全建議。

具Amazon Web Services 一般參考有以下基礎知識：

- [瞭解並取得您的AWS認證](#) — 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

您可以AWS 建構家 ID補充AWS 帳戶您可能已經擁有或想要創建的任何內容。雖然AWS 帳戶作為您建立之AWS資源的容器，並為這些資源提供安全性界限，但您的AWS 建構家 ID代表您是個人。您可以使用您的登錄AWS 建構家 ID以訪問 Amazon 和 Amazon 等開發人員工具 CodeWhisperer 和服務 CodeCatalyst。

- 使用[用AWS 登入戶指南登錄](#) — 了解如何創建和使用，AWS 建構家 ID並了解生成器 ID 提供的AWS 建構家 ID內容。
- 使用 [CodeWhisperer 和進行驗證 AWS 工具組-使用CodeWhisperer 者指南中的產生器 ID](#) — 瞭解如何 CodeWhisperer 使用AWS 建構家 ID。
- [CodeCatalyst概念-AWS 建構家 ID](#) 請參閱 Amazon 使用 CodeCatalyst 者指南 — 了解如何 CodeCatalyst 使用AWS 建構家 ID。

IAM Identity Center 驗證

AWS IAM Identity Center 是在非AWS 計算服務上進行開發時提供 AWS 憑證的建議方法。例如，這會類似於您的本機開發環境。如果您要在 AWS 資源上開發，例如 Amazon Elastic Compute Cloud (Amazon EC2) AWS Cloud9，或者，我們建議您改為從該服務取得登入資料。

在本教學課程中，您會建立 IAM 身分中心存取權，並使用 AWS 存取入口網站和 AWS CLI。

- AWS 存取入口網站是您手動登入 IAM 身分中心的網路位置。網址的格式為d-xxxxxxx.awsapps.com/start或`your_subdomain.awsapps.com/start`。登入 AWS 存取入口網站時，您可以檢視 AWS 帳戶 並為該使用者設定的角色。此程序會使用 AWS 存取入口網站來取得 SDK/ 工具驗證程序所需的組態值。
- 可 AWS CLI 用來設定您的 SDK 或工具，以針對程式碼所發出的 API 呼叫使用 IAM 身分中心驗證。這個一次性程序會更新您的共用 AWS config 檔案，然後當您執行程式碼時，SDK 或工具會使用該檔案。

使用 IAM 身分中心設定程式設計存取

步驟 1：建立存取權並選取適當的權限集

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

選擇下列其中一種方法來存取您的 AWS 認證。

我沒有透過 IAM Identity Center 建立存取權

1. 按照使用者指南中的[預設 IAM Identity Center 目錄程序設定使用者存取](#)權限，以新增 AWS IAM Identity Center 使用者並新增管理許可。
2. AdministratorAccess 權限集不應用於定期開發。除非您的僱主為此目的建立了自訂 PowerUserAccess 權限集，否則我們建議您使用預先定義的權限集。

再次[使用預設的 IAM 身分中心目錄程序進行相同的設定使用者存取](#)權限，但這一次：

- 不要建立 *Admin team* 群組，而是建立 *Dev team* 群組，然後在指示中取代此群組。
- 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。
- 不要建立 *AdministratorAccess* 權限集，而是建立 *PowerUserAccess* 權限集，然後在指示中取代此權限集。

完成後，您應該具備以下內容：

- 一個 Dev team 小組。
- 群組的附加 PowerUserAccess 權限 Dev team 集。
- 您的使用者已新增至 Dev team 群組。

3. 退出口網站，然後再次登入以查看Administrator或的選項PowerUserAccess。AWS 帳戶在使用工具 /SDK PowerUserAccess 時選取此選項。

我已經 AWS 通過雇主管理的聯合身份提供商 (例如 Microsoft Entra 或 Okta) 訪問

AWS 透過身分提供者的入口網站登入。如果您的 Cloud 管理員已授予您 PowerUserAccess (開發人員) 權限，您會看到您 AWS 帳戶有權存取的權限以及您的權限集。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。

我已經可以通 AWS 過雇主管理的 AWS 訪問門戶訪問

AWS 透過 AWS 存取入口網站登入。如果您的 Cloud 管理員已授予您 PowerUserAccess (開發人員) 權限，您會看到您 AWS 帳戶有權存取的權限以及您的權限集。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。

我已經可以通 AWS 過雇主管理的聯合自定義身份提供商訪問

請聯絡您的 IT 團隊尋求協助。

步驟 2：設定 SDK 和工具以使用 IAM 身分中心

1. 在您的開發機器上，安裝最新的 AWS CLI。
 - a. 請參閱 [《AWS Command Line Interface 使用指南》AWS CLI 中的〈安裝或更新〉的最新版本](#)。
 - b. (選擇性) 若要確認 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 URL，或者您可以通過以下步驟 1：建立訪問權限的電子郵件中獲得它。如果沒有，請在 <https://console.aws.amazon.com/singlesignon/> 的儀表板上找到您的AWS 訪問門戶網址。
 - 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。此時會顯示您使用者的角色。選擇 [存取金鑰] 以取得命令列的認證或適當權限集的程式設計存取。使用預先定義的PowerUserAccess權限集，或您或您的雇主建立的任何權限集來套用最低權限權限進行開發。
 - 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 身分中心登入資料方法，以取得下一個步驟所需的SSO Start URL和SSO Region值。
3. 在 AWS CLI 命令提示字元中，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。如需有關此 AWS CLI 命令的詳細資訊，請參閱[使用aws configure sso精靈設定您的設定檔](#)。
 - 對於 CLI 設定檔名稱，我們建議您在開始使用時輸入###。若要取得有關如何設定非預設 (具名) 設定檔及其關聯環境變數的資訊，請參閱 [〈〉 描述檔](#)。
 4. (選擇性) 在 AWS CLI 命令提示字元中，執行aws sts get-caller-identity令來確認作用中階段作業識別。回應應會顯示您設定的 IAM 身分中心權限集。
 5. 如果您使用的是 AWS SDK，請在開發環境中為 SDK 建立應用程式。
 - a. 對於某些 SDK，SSO0IDC必須先將其他套件 (例如SSO和) 新增至您的應用程式，才能使用 IAM 身分中心驗證。如需詳細資訊，請參閱您的特定 SDK。
 - b. 如果您先前已設定存取權 AWS，請檢閱您的共用 AWS credentials檔案是否有任何檔案[AWS 存取金鑰](#)。您必須先移除任何靜態登入資料，SDK 或工具因[憑證提供者鏈](#)優先順序而使用 IAM 身分中心登入資料。

有關 SDK 和工具如何使用此配置使用和重新整理認證的深入研究，請參閱[瞭解 IAM 身分中心身分驗證](#)。

根據您設定的工作階段長度，您的存取最終會過期，SDK 或工具會遇到驗證錯誤。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。

您可以延長 IAM 身分中心存取入口網站工作階段持續時間和權限集工作階段持續時間。這會延長您可以執程式碼的時間，然後您需要使用 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 身分識別中心工作階段持續時間 — [設定使用者存取 AWS 取入口網站工作階段](#)
- 權限設置會話持續時間-[設置會話持續](#)

如需 SDK 和工具的所有 IAM 身分中心提供者設定的詳細資訊，請參閱本指南[IAM 身分中心憑證提供者](#)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，AWSSDK 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」。
權限集認證	AWS 登入資料	SDK 實際用於大多數AWS 服務呼叫的憑據（特別是所有

IAM 身分識別中心名稱	開發套件 API 名稱	描述
	簽署 4 認證	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。

擔任角色

假設角色涉及使用一組臨時安全認證來存取您可能無法存取的 AWS 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。要了解有關 AWS Security Token Service (AWS STS) API 請求的更多信息，請參閱 AWS Security Token Service API 參考中的 [操作](#)。

若要將 SDK 或工具設定為擔任角色，您必須先建立或識別要擔任的特定角色。IAM 角色是由角色 Amazon 資源名稱 (ARN) 唯一識別的。角色與另一個實體建立信任關係。使用此角色的受信任實體可能是 AWS 服務網頁身分識別提供者或 OIDC 或 SAML 聯盟。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，如果您受到該角色的信任，您可以將 SDK 或工具設定為使用角色授與的權限。要做到這一點，要么 [假設 IAM 角色](#) 或 [與網絡身份或 OpenID Connect 聯盟](#)。

假設 IAM 角色

假設角色時，會 AWS STS 傳回一組暫時的安全性認證。這些認證來自其他設定檔，或來自您的程式碼執行的執行個體或容器。假設角色的其他示例包括管理 AWS 帳戶來自 Amazon EC2 的多個 AWS 帳戶，AWS CodeCommit 跨越使用或從中訪問另一個帳戶 AWS CodeBuild。

步驟 1：設定 IAM 角色

若要將 SDK 或工具設定為擔任角色，您必須先建立或識別要擔任的特定角色。IAM 角色是使用角色 [ARN](#) 進行唯一識別的。角色會與其他實體建立信任關係，通常是在您的帳戶內或跨帳戶存取。若要進行設定，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 或工具以從 `credential_source` 或取得認證 `source_profile`。

用 `credential_source` 於從 Amazon ECS 容器、Amazon EC2 執行個體或環境變數取得登入資料。

用 `source_profile` 於從其他設定檔取得身分證明。 `source_profile` 也支援角色鏈結，這是設定檔的階層，其中假定的角色會用來擔任其他角色。

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。若要透過假設角色來擷取和使用臨時認證，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 [假設角色認證提供者設定](#) 章節。

- `role_arn` - 從您在步驟 1 中建立的 IAM 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 檔案中兩種假設角色選項的組態：

```
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
```

如需所有假設角色認證提供者設定的詳細資訊，請參閱本指南 [假設角色認證提供者](#) 中的。

與網絡身份或 OpenID Connect 聯盟

建立需要存取權的行動應用程式或以用戶端為基礎的 Web 應用程式時 AWS，會針對透過公用身分識別提供者 (IdP) 驗證的聯合身分使用者 AWS STS 傳回一組臨時安全登入資料。公有身分提供者的範例包含 Login with Amazon、Facebook、Google 或任何與身分提供者相容的 OpenID Connect (OIDC)。使用此方法，您的使用者不需要自己的身分 AWS 或 IAM 身分。

如果您使用的是 Amazon Elastic Kubernetes Service，則此功能可讓您為每個容器指定不同的 IAM 角色。Kubernetes 提供將 OIDC 權杖散發至此認證提供者用來取得暫時認證的容器的功能。如需此 Amazon EKS 組態的詳細資訊，請參閱 Amazon EKS 使用者指南中的[服務帳戶適用的 IAM 角色](#)。不過，如果您的[開發套件支援](#)，我們建議您改用 Amazon EKS Pod 身分識別，以取代此選項，以取代此選項。

步驟 1：設定身分識別提供者和 IAM 角色

若要設定與外部 IdP 的聯合，請使用 IAM 身分供應商來通AWS知外部 IdP 及其組態。這會在您AWS帳戶與外部 IdP 之間建立信任。在將 SDK 設定為使用 Web 身分權杖進行驗證之前，您必須先設定身分識別提供者 (IdP) 和用於存取它的 IAM 角色。若要進行設定，請參閱 IAM 使用者指南中的[建立網頁身分角色或 OpenID Connect 同盟 \(主控台\)](#)。

步驟 2：設定 SDK 或工具

將 SDK 或工具配置為使用來自的 Web 身份權杖進行身AWS STS份驗證。

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 身分同盟擷取和使用臨時認證，請在共用 AWSconfig 檔案中指定下列組態值。如需這些設定的詳細資訊，請參閱[假設角色認證提供者設定](#)章節。

- `role_arn`-從您在步驟 1 中建立的 IAM 角色
- `web_identity_token_file`-來自外部 IdP
- (選用) `duration_seconds`
- (選用) `role_session_name`

以下是假設具有 Web 身份的角色共享 config 文件配置的示例：

```
[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 充當身分識別代理人，為您完成大部分聯合會工作。不過，Amazon Cognito 身分供應商不會像其他身分供應商一樣包含在 SDK 和工具核心程式庫中。若要存取 Amazon Cognito API，請在組建中包含 Amazon

Cognito 服務用戶端，或為您的開發套件或工具包含程式庫。如需使用 AWS SDK 的相關資訊，請參閱 Amazon Cognito 開發人員指南中的[程式碼範例](#)。

如需所有假設角色認證提供者設定的詳細資訊，請參閱本指南[假設角色認證提供者](#)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 Identity Center 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](#)。

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[使用長期認證進行驗](#)。

使用短期憑證進行驗

我們建議您將 SDK 或工具設定為[IAM Identity Center 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 存取入口網站的臨時憑證。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。

使用從 AWS 存取入口網站擷取的短期憑證來設定認證檔案

1. [建立共用認證檔案](#)。
2. 在認證檔案中，貼上下列預留位置文字，直到您貼上工作中的暫時認證為止。

```
[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[描述檔時使用的 \[預設\] 設定檔](#)。
4. [登入 AWS 存取入口網站](#)。
5. 請遵循以下[手動登入資料重新整理](#)指示，從存取入口網站複製 IAM 角色登 AWS 入資料。
 - a. 對於連結指示中的步驟 4，請選擇針對您的開發需求授予存取權的 IAM 角色名稱。此角色通常具有類似 `PowerUserAccess` 或開發人員的名稱。

- 請注意，儲存在共用 AWS credentials 檔案中的任何認證都會以純文字儲存。

安全管理憑證的其他指引

如需如何安全管理 [AWS](#) 認證的一般討論，請參閱 [AWS 一般參考](#)。除了討論之外，請考慮下列事項：

- 使用適用於 Amazon Elastic Container Service (Amazon ECS) 任務的 [任務 IAM 角色](#)。
- 使用在 Amazon EC2 執行個體上執行的應用程式的 [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 [開始使用：您是第一次 AWS 使用嗎？](#) 在《參 AWS Account Management 考指南》中。

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，請依據 IAM 使用者指南的 [建立管理使用者](#) 中的說明，建立管理使用者帳戶。

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應用程式 AWS 或執行應用程式的開發 AWS。最佳作法是，您需要建立適合這些工作的使用者、權限集或服務角色。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 IAM 使用者

- 按照 IAM 使用者指南中的 [建立 IAM 使用者 \(主控台\)](#) 程序來建立 IAM 使用者。建立 IAM 使用者時：
 - 我們建議您選取 [提供使用者存取權限] 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 相關的資訊，例如檢查 AWS CloudTrail 診斷日誌或將檔案上傳到 Amazon Simple Storage 服務，這在偵錯程式碼時很有幫助。

- 對於 [設定權限-權限選項]，選取 [直接附加原則] 以瞭解您要如何指派權限給此使用者。
- 多數「入門」SDK 教學都使用 Amazon S3 服務做為範例。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權限界限或標籤的選擇性步驟。

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使用者的 **User name**。
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. 在 [建立存取金鑰步驟 1] 中，選擇 [命令列介面 (CLI)] 或 [本機程式碼]。這兩個選項都會產生與 SDK 一起使用的相同類型的金鑰。AWS CLI
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. 在建立存取金鑰步驟 3 中，選取下載 .csv 檔案，以儲存包含 IAM 使用者存取金鑰和私密存取金鑰的 .csv 檔案。您之後將會用到此資訊。

Warning

使用適當的安全措施來確保這些憑證的安全。

6. 選取 Done (完成)。

步驟 3：更新共享 **credentials** 文件

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[認證檔案的位置](#)。
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 = wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
```

3. 儲存檔案。

共用credentials檔案是儲存認證的最常用方式。這些也可以設置為環境變量，請參閱[AWS 存取金鑰](#)閱環境變量名稱。這是讓您開始使用的一種方式，但我們建議您盡快轉換至 IAM 身分中心或其他臨時登入資料。避免使用長期認證之後，請記得從共用credentials檔案中刪除這些認證。

針對 Amazon EC2 執行個體使用 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。

對於 Amazon 彈性運算雲端執行個體，請建立 IAM 角色，然後將該角色的存取權授予您的 Amazon EC2 執行個體。如需詳細資訊，請參閱 [Amazon EC2 使用者指南中的適用於 Amazon EC2 的 IAM 角色](#)或 [Amazon EC2 使用者指南中的適用於 Amazon EC2 的 IAM 角色](#)。

建立 IAM 角色

建立可授與 Amazon S3 唯讀存取權的 IAM 角色。

1. 登入 AWS Management Console 並開啟身分與存取權管理主控台，網址為 <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，然後選取下一步。
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，因為啟動 Amazon EC2 執行個體時會需要這個名稱。

啟動 Amazon EC2 執行個體並指定您的 IAM 角色

您可以使用 Amazon EC2 主控台啟動具有 IAM 角色的 Amazon EC2 執行個體。

按照 [Amazon EC2 使用者指南](#)或 [Amazon EC2 使用者指南](#)中的指示啟動執行個體。

當您到達 Review Instance Launch (檢閱執行個體啟動) 頁面時，選取 Edit instance details (編輯執行個體詳細資訊)。在 IAM 角色中，選擇您先前建立的 IAM 角色。依照指示完成程序。

Note

您需要建立或使用現有的安全群組和金鑰對，以連接到執行個體。

透過這個 IAM 和 Amazon EC2 設定，您可以將應用程式部署到 Amazon EC2 執行個體，並具有對 Amazon S3 服務的讀取存取權限。

Connect 至 EC2 執行個體

Connect 至 EC2 執行個體，以便將範例應用程式傳輸到該執行個體，然後執行應用程式。您需要包含用來啟動執行個體之 key pair 私密部分的檔案；也就是 PEM 檔案。

您可以按照 [Amazon EC2 使用者指南](#)或 [Amazon EC2 使用者指南](#)中的連線程序執行此操作。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。

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 EC2 執行個體上執行範例應用程式

1. 將應用程式檔案從本機磁碟複製到執行個體。

如需有關如何將檔案傳輸到執行個體的詳細資訊，請參閱 [Amazon EC2 使用者指南](#)或 [Amazon EC2 使用者指南](#)。

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) 確認應用程式使用 IAM 角色提供的登入資料。
 - a. 登入 AWS Management Console 並開啟 Amazon EC2 主控台，網址為 <https://console.aws.amazon.com/ec2/>。
 - b. 選取執行個體，然後透過動作、執行個體設定、附加/取代 IAM 角色分離 IAM 角色。
 - c. 再次執行應用程式，並確認其傳回授權錯誤。

設定參考

SDK 提供 AWS 服務他們負責成功進行 API 調用所需的一些繁重工作，包括身份驗證，重試行為等。為此，SDK 具有靈活的策略來獲取用於請求的憑據，維護用於每個服務的設置，以及獲取用於全局設置的值。

您可以在下列各節中找到有關組態設定的詳細資訊：

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

建立服務用戶端

為了以編程方式訪問 AWS 服務，SDK 使用每個客戶端類/對象。AWS 服務例如，如果您的應用程式需要存取 Amazon EC2，您的應用程式會建立一個 Amazon EC2 用戶端物件，以便與該服務進行交互。然後，您可以使用服務客戶端向其發出請求 AWS 服務。在大多數 SDK 中，服務客戶端對象是不可變的，因此您必須為向其發出請求的每個服務創建一個新的客戶端，並使用不同的配置向同一服務發出請求。

設定的優先順序

全域設定會設定大多數 SDK 支援的功能、認證提供者和其他功能，並在各方面產生廣泛 AWS 服務影響。所有 SDK 都有一系列位置（或來源），它們會檢查以查找全局設置的值。以下是設定查閱優先順序：

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。
 - 某些設定可以根據每個作業進行設定，並且可以根據需要針對您呼叫的每個作業進行變更。對於 AWS CLI 或 AWS Tools for PowerShell，這些參數採用您在指令行中輸入的每個作業參數的形式。對於 SDK，明確指派可以採用您在實例化 AWS 服務用戶端或設定物件時所設定的參數形式，或者有時在呼叫個別 API 時設定的參數形式。
2. 僅限 Java /Kotlin：已檢查該設定的 JVM 系統屬性。如果已設定，則會使用該值來設定用戶端。
3. 檢查環境變數。如果已設定，則會使用該值來設定用戶端。
4. SDK 會檢查共用credentials檔案的設定。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，SDK 會使用它。
 - AWS_PROFILE環境變數或 aws.profile JVM 系統屬性可用來指定 SDK 載入的設定檔。

6. 最後使用 SDK 原始程式碼本身提供的任何預設值。

Note

某些 SDK 和工具可能會以不同的順序檢查。此外，一些 SDK 和工具支持其他存儲和檢索參數的方法。例如，AWS SDK for .NET 支援稱為 [SDK 存放區](#) 的其他來源。如需 SDK 或工具唯一提供者的詳細資訊，請參閱您所使用之 SDK 或工具的特定指南。

順序決定哪些方法優先順序並覆寫其他方法。例如，如果您在共享 config 檔案中設定描述檔，則只有在 SDK 或工具首先檢查其他位置後，才會找到並使用該設定檔。這意味著，如果您在 credentials 文件中放置一個設置，則會使用該設置而不是在 config 文件中找到的設置。如果您使用設定和值來設定環境變數，它會覆寫 credentials 和 config 檔案中的該設定。最後，單個操作（AWS CLI 命令行參數或 API 參數）或代碼中的設置將覆蓋該命令的所有其他值。

Config 檔案設定清單

可以在共用 AWS config 檔案中指定下表中列出的設定。它們是全球性的，並影響所有人 AWS 服務。SDK 和工具也可能支援唯一的設定和環境變數。若要查看僅由個別 SDK 或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
api_versions	一般組態設定
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
ca_bundle	一般組態設定
credential_process	程序認證提供者

設定名稱	詳細資訊
credential_source	假設角色認證提供者
defaults_mode	智慧型組態預設
disable_request_compression	請求壓縮
duration_seconds	假設角色認證提供者
ec2_metadata_service_endpoint	IMDS 認證提供者
ec2_metadata_service_endpoint_mode	IMDS 認證提供者
ec2_metadata_v1_disabled	IMDS 認證提供者
endpoint_discovery_enabled	端點探索
endpoint_url	服務特定端點
external_id	假設角色認證提供者
ignore_configured_endpoint_urls	服務特定端點

設定名稱	詳細資訊
max_attempts	重試行為
metadata_service_num_attempts	Amazon EC2 實例元數據
metadata_service_timeout	Amazon EC2 實例元數據
mfa_serial	假設角色認證提供者
output	一般組態設定
parameter_validation	一般組態設定
region	AWS 區域
request_max_in_compression_size_bytes	請求壓縮
retry_mode	重試行為
role_arn	假設角色認證提供者
role_session_name	假設角色認證提供者
s3_disable_multiregion_access_points	Amazon S3 多區域存取點
s3_use_arn_region	Amazon S3 存取點

設定名稱	詳細資訊
sdk_ua_app_id	申請識別碼
source_profile	假設角色認證提供者
sso_account_id	IAM 身分中心憑證提供者
sso_region	IAM 身分中心憑證提供者
sso_registration_scopes	IAM 身分中心憑證提供者
sso_role_name	IAM 身分中心憑證提供者
sso_start_url	IAM 身分中心憑證提供者
sts_regional_endpoints	AWS STS 區域化端點
use_dualstack_endpoint	雙堆疊和 FIPS 端點
use_fips_endpoint	雙堆疊和 FIPS 端點
web_identity_token_file	假設角色認證提供者

Credentials 檔案設定清單

可以在共用 AWS credentials 檔案中指定下表中列出的設定。它們是全球性的，並影響所有人 AWS 服務。SDK 和工具也可能支援唯一的設定和環境變數。若要查看僅由個別 SDK 或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

下表列出了大多數 SDK 支援的環境變數。它們是全球性的，並影響所有人 AWS 服務。SDK 和工具也可能支援唯一的設定和環境變數。若要查看僅由個別 SDK 或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_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	容器認證提供者
AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE	容器認證提供者
AWS_CONTAINER_CRED	容器認證提供者

設定名稱	詳細資訊
ENTIALS_F ULL_URI	
AWS_CONTA INER_CRED ENTIALS_R ELATIVE_URI	容器認證提供者
AWS_DEFAU LTS_MODE	智慧型組態預設
AWS_DISAB LE_REQUES T_COMPRESSION	請求壓縮
AWS_EC2_M ETADATA_D ISABLED	IMDS 認證提供者
AWS_EC2_M ETADATA_S ERVICE_EN DPOINT	IMDS 認證提供者
AWS_EC2_M ETADATA_S ERVICE_EN DPOINT_MODE	IMDS 認證提供者
AWS_EC2_M ETADATA_V 1_DISABLED	IMDS 認證提供者
AWS_ENABL E_ENDPOIN T_DISCOVERY	端點探索

設定名稱	詳細資訊
AWS_ENDPOINT_URL	服務特定端點
AWS_ENDPOINT_URL_<SERVICE>	服務特定端點
AWS_IAM_ROLE_ARN	假設角色認證提供者
AWS_IAM_ROLE_SESSION_NAME	假設角色認證提供者
AWS_IGNORE_ENDPOINT_URLS	服務特定端點
AWS_MAX_ATTEMPTS	重試行為
AWS_METADATA_SERVICE_NUM_ATTEMPTS	Amazon EC2 實例元數據
AWS_METADATA_SERVICE_TIMEOUT	Amazon EC2 實例元數據
AWS_PROFILE	共用config和credentials 檔案
AWS_REGION	AWS 區域

設定名稱	詳細資訊
AWS_REQUEST_MIN_COMPRESSION_SIZE_BYTES	請求壓縮
AWS_RETRY_MODE	重試行為
AWS_S3_DISABLE_MULTIREGION_ACCESS_POINTS	Amazon S3 多區域存取點
AWS_S3_USE_ARN_REGION	Amazon S3 存取點
AWS_SDK_UA_APP_ID	申請識別碼
AWS_SECRET_ACCESS_KEY	AWS 存取金鑰
AWS_SESSION_TOKEN	AWS 存取金鑰
AWS_SHARED_CREDENTIALS_FILE	共用config和credentials 檔案的位置
AWS_STS_REGIONAL_ENDPOINTS	AWS STS 區域化端點
AWS_USE_DUALSTACK_ENDPOINT	雙堆疊和 FIPS 端點

設定名稱	詳細資訊
AWS_USE_FIPS_ENDPOINT	雙堆疊和 FIPS 端點
AWS_WEB_IDENTITY_TOKEN_FILE	假設角色認證提供者

JVM 系統屬性清單

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

設定名稱	詳細資訊
aws.accessKeyId	AWS 存取金鑰
aws.configFile	共用config和credentials 檔案的位置
aws.defaultMode	智慧型組態預設
aws.disableEc2MetadataV1	IMDS 認證提供者
aws.disableRequestCompression	請求壓縮
aws.ec2MetadataServiceEndpoint	IMDS 認證提供者
aws.ec2MetadataServiceEndpoint	IMDS 認證提供者

設定名稱	詳細資訊
<code>viceEndpointMode</code>	
<code>aws.endpointDiscoveryEnabled</code>	端點探索
<code>aws.endpointUrl</code>	服務特定端點
<code>aws.endpointUrl<ServiceName></code>	服務特定端點
<code>aws.ignoreConfiguredEndpointUrls</code>	服務特定端點
<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>	假設角色認證提供者
<code>aws.roleSessionName</code>	假設角色認證提供者

設定名稱	詳細資訊
<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
<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>	申請識別碼
<code>aws.webIdentityTokenFile</code>	假設角色認證提供者

AWS SDK 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，並且在許多 SDK 中以相同的方式運作。跨多個 SDK 編碼時，這種一致性可提高生產力和清晰度。所有設置都可以在代碼中覆蓋。如需詳細資訊，請參閱您的特定 SDK API。

Important

並非所有 SDK 都支援所有提供者，甚至是提供者內的所有層面。

主題

- [憑證提供者鏈](#)
- [AWS 存取金鑰](#)
- [假設角色認證提供者](#)
- [容器認證提供者](#)
- [IAM 身分中心憑證提供者](#)
- [IMDS 認證提供者](#)
- [程序認證提供者](#)

憑證提供者鏈

所有 SDK 都有一系列位置（或來源），它們會檢查這些位置（或來源），以便找到用於向 AWS 服務找到有效的憑證後，系統就會停止搜尋。此系統搜尋稱為預設認證提供者鏈結。

儘管每個 SDK 使用的不同鏈結各不相同，但它們通常包括以下來源：

憑證提供者	描述
AWS 存取金鑰	AWS IAM 使用者的存取金鑰 (例如AWS_ACCESS_KEY_ID 和AWS_SECRET_ACCESS_KEY)。
與網絡身份或 OpenID Connect 聯盟-假設角色認證提供者	使用知名的外部身份提供商 (IdP) 登錄，例如使用亞馬遜，臉書，谷歌或任何其他與 OpenID Connect (OIDC) 兼容的 IdP 登錄。使用 AWS Security Token Service (AWS STS) 中的 Web 身分權杖，假設 IAM 角色的許可。

憑證提供者	描述
IAM 身分中心憑證提供者	從中獲取憑據 AWS IAM Identity Center。
假設角色認證提供者	透過假設 IAM 角色的許可來存取其他資源。(擷取並使用角色的臨時認證)。
容器認證提供者	Amazon Elastic Container Service (Amazon ECS) 和 Amazon Elastic Kubernetes Service (Amazon EKS) 憑據。容器認證提供者會擷取客戶容器化應用程式的認證。
程序認證提供者	自訂認證提供者。從外部來源或程序取得您的登入資料，包括隨處存取 IAM 角色。
IMDS 認證提供者	亞馬遜彈性運算雲端 (Amazon EC2) 執行個體設定檔登入資料。將 IAM 角色與每個 EC2 執行個體建立關聯。該角色的臨時認證可用於在執行個體中執行的程式碼。憑證是透過 Amazon EC2 中繼資料服務傳遞。

對於鏈中的每個步驟，有多種方法可以指定設定值。程式碼中指定的設定值永遠具有優先順序。但是，也有[環境變數](#)和[共用config和credentials檔案](#)。如需更多詳細資訊，請參閱[設定的優先順序](#)。

AWS 存取金鑰

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](#)。

AWS IAM 使用者的存取金鑰可用作您的 AWS 登入資料。AWS SDK 會自動使用這些 AWS 認證來簽署 API 請求 AWS，以便您的工作負載可以安全方便地存取您的 AWS 資源和資料。建議始終使用，`aws_session_token`以便認證是臨時的，並且在過期後不再有效。不建議使用長期認證。

Note

如果 AWS 無法重新整理這些臨時登入資料，AWS 可能會延長認證的有效性，以便您的工作負載不受影響。

共用 AWS credentials 檔案是儲存身分證明資訊的建議位置，因為它安全不在應用程式來源目錄之外，而且與共用 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](#)。

使用下列項目設定此功能：

aws_access_key_id-共享 AWS config 文件設置, **aws_access_key_id**-共享 AWS credentials 文件設置 (推薦方法), **AWS_ACCESS_KEY_ID**-環境變量, **aws.accessKeyId**-JVM 系統屬性：僅限爪哇/科特林

指定用作認證一部分的 AWS 存取金鑰，以驗證使用者。

aws_secret_access_key-共享 AWS config 文件設置, **aws_secret_access_key**-共享 AWS credentials 文件設置 (推薦方法), **AWS_SECRET_ACCESS_KEY**-環境變量, **aws.secretAccessKey**-JVM 系統屬性：僅限爪哇/科特林

指定作為認證一部分用來驗證使用者的 AWS 秘密金鑰。

aws_session_token-共享 AWS config 文件設置, **aws_session_token**-共享 AWS credentials 文件設置 (推薦方法), **AWS_SESSION_TOKEN**-環境變量, **aws.sessionToken**-JVM 系統屬性：僅限爪哇/科特林

指定 AWS 作為認證一部分用來驗證使用者的工作階段權杖。您會收到此值，做為成功要求擔任角色所傳回之暫時認證的一部分。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。如需安全性建議，請參閱 [IAM 中的安全性最佳實務](#)。

如需如何取得這些值的指示，請參閱 [使用短期憑證進行驗](#)。

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
```

通過命令行設置環境變量的 Linux/macOS 示例：

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

通過命令行設置環境變量的 Windows 示例：

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

與 AWS 開發套件的相容性

下列 SDK 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。和唯一支援任何 JVM 系統屬 AWS SDK for Java 性設 適用於 Kotlin 的 AWS SDK 定。

SDK	支 備註或更多資訊 援
AWS CLI v2	是
SDK for C++	是 不支援共用config檔案。
適用於轉到 V2 的開發套件 (1.x)	是
適用於 Go 1.x 的開發套件 (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 。
適用於 Java 2.x 的開發套件	是
適用於 Java 1.x 的開發套件	是
適用於 JavaScript 3.x 的開發套	是

SDK	支援	備註或更多資訊
適用於 JavaScript 2.x 的 SDK	是	
SDK 的科特林	是	
適用於 .NET 3.x 的開發套	是	不支援環境變數。
適用於 PHP 3.x 的開發套件	是	
SDK for Python (Boto3)	是	
適用於紅寶石 3.x 的 SDK	是	
適用於鏽蝕的 SDK	是	
適用的工具 PowerShell	是	不支援環境變數。

假設角色認證提供者

假設角色涉及使用一組臨時安全認證來存取您可能無法存取的 AWS 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。

若要將 SDK 或工具設定為擔任角色，您必須先建立或識別要擔任的特定角色。IAM 角色是由角色 Amazon 資源名稱 (ARN) 唯一識別的。角色與另一個實體建立信任關係。使用此角色的受信任實體可能是 AWS 服務網頁身分識別提供者或 OIDC 或 SAML 聯盟。AWS 帳戶

識別 IAM 角色後，如果您受到該角色的信任，您可以將 SDK 或工具設定為使用角色授與的權限。若要這麼做，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[擔任角色](#)中的。

假設角色認證提供者設定

使用下列項目設定此功能：

credential_source-共享 AWS config 文件設置

在 Amazon EC2 執行個體或 Amazon 彈性容器服務容器中使用，以指定開發套件或工具可以在何處找到有權承擔您使用該 `role_arn` 參數指定角色的登入資料。

預設值：無

有效值：

- 環境 — 指定 SDK 或工具從環境變數 [AWS_ACCESS_KEY_ID](#) 和擷取來源認證 [AWS_SECRET_ACCESS_KEY](#)。
- EC2 InstanceMetadata — 指定 SDK 或工具使用 [附加到 EC2 執行個體設定檔的 IAM 角色](#) 來取得來源登入資料。
- EcsContainer — 指定 SDK 或工具使用 [附加至 ECS 容器的 IAM 角色](#) 來取得來源認證。

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

在 `config` 檔案中設定此值以指出登入資料應該來自 Amazon EC2 的範例：

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

`duration_seconds`-共享 AWS `config` 文件設置

指定角色工作階段的最大持續時間 (以秒為單位)。

只有當設定檔指定擔任角色時，才會套用此設定。

預設值：3600 秒 (一小時)

有效值：此值的範圍從 900 秒 (15 分鐘) 到為角色設定的工作階段持續時間上限設定 (最多可為 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` 文件中設置此值的示例：

```
duration_seconds = 43200
```

`external_id`-共享 AWS `config` 文件設置

指定一個唯一識別符，第三方用來在其客戶帳戶擔任角色。

只有當設定檔指定為擔任角色，且角色的信任原則需要的值時，才會套用此設定 `ExternalId`。值會對應至設定檔指定角色時傳遞至 `AssumeRole` 作業的 `ExternalId` 參數。

預設值：無。

有效值：請參閱 [IAM 使用者指南中的向第三方授予資 AWS 源存取權時如何使用外部 ID](#)。

在config文件中設置此值的示例：

```
external_id = unique_value_assigned_by_3rd_party
```

mfa_serial-共享 AWS config文件設置

指定使用者在擔任角色時必須使用的多重要素驗證 (MFA) 裝置的識別碼或序號。

假設該角色的信任原則包含需要 MFA 驗證之條件的角色時，需要此選項。

預設值：無。

有效值：此值可以是硬體裝置 (例如GAHT12345678) 的序號，也可以是虛擬 MFA 裝置的 Amazon 資源名稱 (ARN)。如需 MFA 的詳細資訊，請參閱 IAM 使用者指南中的[設定受 MFA 保護的 API 存取](#)。

在config文件中設置此值的示例：

```
mfa_serial = arn:aws:iam::123456789012:mfa/my-user-name
```

role_arn-共享 AWS config文件設置, **AWS_IAM_ROLE_ARN**-環境變量, **aws.roleArn**-JVM 系統屬性：僅限爪哇/科特林

指定要用來執行使用此設定檔請求的操作的 IAM 角色的 Amazon 資源名稱 (ARN)。

預設值：無。

有效值：該值必須是 IAM 角色的 ARN，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個設定：

- `source_profile`— 識別另一個設定檔，用來尋找有權在此設定檔中擔任角色的認證。
- `credential_source`— 使用由目前環境變數識別的登入資料或連接至 Amazon EC2 執行個體設定檔的登入資料，或 Amazon ECS 容器執行個體。
- `web_identity_token_file`— 為已在移動或網絡應用程序中進行身份驗證的用戶使用公共身份提供商或任何 OpenID Connect (OIDC) 兼容的身份提供商。

role_session_name-共享 AWS config文件設置, **AWS_IAM_ROLE_SESSION_NAME**-環境變量, **aws.roleSessionName**-JVM 系統屬性：僅限爪哇/科特林

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的 AWS CloudTrail 記錄中，這在稽核時很有用。

在上述範例中，A 描述檔使用 B 描述檔中的憑證。當您指定 SDK 或工具應使用設定A檔時，SDK 或工具會自動查詢連結B設定檔的認證，並使用它們為指定的 IAM 角色要求臨時登入資料。SDK 或工具會在背景中使用 [sts: AssumeRole](#) 作業來完成此操作。然後，您的代碼將使用這些臨時憑據來訪問 AWS 資源。指定的角色必須附加 IAM 許可政策，以允許執行請求的程式碼 AWS 服務，例如命令或 API 方法。

`web_identity_token_file`-共享 AWS config 文件設置, `AWS_WEB_IDENTITY_TOKEN_FILE`-環境變量, `aws.webIdentityTokenFile`-JVM 系統屬性：僅限爪哇/科特林

指定包含來自 [支援 OAuth 2.0 提供者或 OpenID 身分識別提供者之存取權杖的檔案 Connect](#) 徑。

此設定可透過使用網路身分聯合提供者 (例如 [Google](#)、[Facebook](#) 和 [Amazon](#) 等) 來啟用身份驗證。SDK 或開發人員工具會載入此檔案的內容，並在代表您呼叫作業時將其 `AssumeRoleWithWebIdentity` 作為 `WebIdentityToken` 引數傳遞。

預設值：無。

有效值：此值必須是路徑和檔案名稱。該文件必須包含身份提供者提供給您的 OAuth 2.0 訪問令牌或 OpenID Connect 令牌。相對路徑會被視為相對於流程的工作目錄。

與 AWS 開發套件的相容性

下列 SDK 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。和唯一支援任何 JVM 系統屬 AWS SDK for Java 性設 適用於 Kotlin 的 AWS SDK 定。

SDK	支 備註或更多資訊 援
AWS CLI v2	是
SDK for C++	部 分 <code>credential_source</code> 不支援。 <code>duration_seconds</code> 不支援。 <code>mfa_serial</code> 不支援。
適用於轉到 V2 的開發套件 (1.x)	是
適用於 Go 1.x 的開發套件 (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	支援	備註或更多資訊
適用於 Java 2.x 的開發套件	部分	mfa_serial 不支援。AWS_ROLE_ARN 而不是使用AWS_IAM_ROLE_ARN 。AWS_ROLE_SESSION_NAME 而不是使用AWS_IAM_ROLE_SESSION_NAME 。
適用於 Java 1.x 的開發套件	部分	mfa_serial 不支援。不支援 JVM 系統屬性。
適用於 JavaScript 3.x 的開發套件	是	
適用於 JavaScript 2.x 的開發套件	部分	credential_source 不支援。
SDK 的科特林	是	AWS_ROLE_ARN 而不是使用AWS_IAM_ROLE_ARN 。AWS_ROLE_SESSION_NAME 而不是使用AWS_IAM_ROLE_SESSION_NAME 。
適用於 .NET 3.x 的開發套件	是	
適用於 PHP 3.x 的開發套件	是	
SDK for Python (Boto3)	是	
適用於紅寶石 3.x 的 SDK	是	
適用於鏽蝕的 SDK	是	
適用的工具 PowerShell	是	

容器認證提供者

容器認證提供者會擷取客戶容器化應用程式的認證。此認證提供商對 Amazon Elastic Container Service (Amazon ECS) 和 Amazon Elastic Kubernetes Service (Amazon EKS) 客戶非常有用。開發套件會嘗試透過 GET 要求從指定的 HTTP 端點載入認證。

如果您使用 Amazon ECS，建議您使用任務 IAM 角色來改善登入資料隔離、授權和可稽核性。設定完成後，Amazon ECS 會設定開發套件和工具用來取得登入資料的 `AWS_CONTAINER_CREDENTIALS_RELATIVE_URI` 環境變數。若要針對此功能設定 Amazon ECS，請參閱 Amazon 彈性容器服務開發人員指南中的任務 [IAM 角色](#)。

如果您使用 Amazon EKS，建議您使用 Amazon EKS Pod 身分來改善登入資料隔離、最低特權、可稽核性、獨立操作、可重複使用性和可擴展性。您的網繭和 IAM 角色都與 Kubernetes 服務帳戶相關聯，以管理應用程式的登入資料。若要進一步了解有關 Amazon EKS 網繭身分識別的資訊，請參閱 [Amazon EKS 網繭使用者指南中的 Amazon EKS 網繭身分識別](#)。設定完成後，Amazon EKS 會設定開發套件 `AWS_CONTAINER_CREDENTIALS_FULL_URI` 和工具用來取得登入資料的 `AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE` 環境變數。如需設定資訊，請參閱 [Amazon EKS 使用者指南中的設定 Amazon EKS 網繭身分代理](#) 程式或 Amazon EKS [網繭身分識別可簡化部落格](#) 網站上 [Amazon EKS 叢集上應用程式的 IAM 許可](#)。AWS

使用下列項目設定此功能：

`AWS_CONTAINER_CREDENTIALS_FULL_URI`-環境變量

指定提出認證要求時所要使用的 SDK 完整 HTTP URL 端點。這包括方案和主機。

預設值：無。

有效值：有效的 URI。

注意：此設定是替代設定 `AWS_CONTAINER_CREDENTIALS_RELATIVE_URI`，只有在未設定時才會使用。 `AWS_CONTAINER_CREDENTIALS_RELATIVE_URI`

通過命令行設置環境變量的 Linux/macOS 示例：

```
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`-環境變量

指定發出認證請求時所要使用的 SDK 相對 HTTP URL 端點。該值會附加到的預設 Amazon ECS 主機名 `169.254.170.2` 稱。

預設值：無。

有效值：有效的相對 URI。

通過命令行設置環境變量的 Linux/macOS 示例：

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

AWS_CONTAINER_AUTHORIZATION_TOKEN-環境變量

以純文字指定授權權杖。如果設定此變數，SDK 會使用環境變數的值在 HTTP 要求上設定授權標頭。

預設值：無。

有效值：字串。

注意：此設定是替代設定 `AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE`，只有在未設定時才會使用。 `AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE`

通過命令行設置環境變量的 Linux/macOS 示例：

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

AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE-環境變量

指定包含純文字授權 Token 之檔案的絕對檔案路徑。

預設值：無。

有效值：字串。

通過命令行設置環境變量的 Linux/macOS 示例：

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

與 AWS 開發套件的相容性

下列 SDK 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。和唯一支援任何 JVM 系統屬 AWS SDK for Java 性設 適用於 Kotlin 的 AWS SDK 定。

SDK	支援	備註或更多資訊
AWS CLI v2	是	
SDK for C++	是	
適用於轉到 V2 的開發套件 (1.x)	是	
適用於 Go 1.x 的開發套件 (V1)	是	
適用於 Java 2.x 的開發套件	是	
適用於 Java 1.x 的開發套件	部分	AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE 不支援 Amazon EKS 網繭身分識別。
適用於 JavaScript 3.x 的開發套	是	
適用於 JavaScript 2.x 的 SDK	是	
SDK 的科特林	是	
適用於 .NET 3.x 的開發套	是	
適用於 PHP 3.x 的開發套件	是	
SDK for Python (Boto3)	是	
適用於紅寶石 3.x 的 SDK	是	
適用於鏽蝕的 SDK	是	
適用的工具 PowerShell	是	

IAM 身分中心憑證提供者

此驗證機制會用 AWS IAM Identity Center 來取得程式碼的單一登入 (SSO) 存取權。AWS 服務

Note

在 AWS SDK API 文件中，IAM 身分中心登入資料提供者稱為 SSO 登入資料提供者。

啟用 IAM 身分中心後，您可以在共用檔案中為其設定定義設定 AWS config 檔。此設定檔用於連線至 IAM 身分中心存取入口網站。當使用者成功透過 IAM 身分中心進行驗證時，入口網站會傳回與該使用者關聯之 IAM 角色的短期登入資料。若要瞭解 SDK 如何從設定取得暫時認證，並將其用於 AWS 服務要求，請參閱[瞭解 IAM 身分中心身分驗證](#)。

透過檔案設定 IAM 身分中心有兩種方config式：

- SSO 權杖提供者組態 (建議) — 延長工作階段持續時間。
- 舊版不可重新整理的組態 — 使用固定的八小時工作階段。

在這兩種設定中，您都需要在工作階段到期時再次登入。

若要設定自訂工作階段持續時間，您必須使用 SSO 權杖提供者組態。

以下兩個指南包含 IAM 身分中心的其他相關資訊：

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

必要條件

您必須先啟用 IAM 身分中心。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 使用者指南中的入門指南](#)。

或者，請按照本指南中的[IAM Identity Center 驗證](#)說明進行操作。這些指示提供完整的指導，從啟用 IAM 身分中心到完成此處所需的共用config檔案設定。

SSO 權杖提供者組態

Note

若要使用 AWS CLI 建立此組態，請參閱使用中的[aws configure sso](#)精靈設定您的設定檔 AWS CLI。

當您使用 SSO 權杖提供者設定時，AWS SDK 或工具會自動重新整理工作階段，直到延長的工作階段期間為止。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](#)

該config文件的sso-session部分用於分組配置變量以獲取 SSO 訪問令牌，然後可以用來獲取 AWS 憑據。如需有關格式化config檔案中區段的更多詳細資訊，請參閱[配置文件的格式](#)。

定義 sso-session 區段並將其與設定檔相關聯。sso_region 和 sso_start_url 必須在 sso-session 區段內設定。一般而言，sso_account_id且sso_role_name必須在profile區段中設定，以便 SDK 可以要求 AWS 認證。

Note

有關 SDK 和工具如何使用此配置使用和刷新憑據的深入研究，請參閱[瞭解 IAM 身分中心身分驗證](#)。

下列範例會將 SDK 設定為要求 IAM 身分中心登入資料。它還支持自動令牌刷新。

```
[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
```

您可以在多個設定檔中重複使用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
```

sso_account_id 並 sso_role_name 不是 SSO 令牌配置的所有場景都是必需的。如果您的應用程式只 AWS 服務 使用支援承載驗證，則不需要傳統 AWS 認證。承載身分驗證是一種 HTTP 身分驗證結構描述，使用稱為承載字符的安全字符。在這種情況下，sso_account_id 和 sso_role_name 並非必要資訊。請參閱單獨指南，AWS 服務 以確定它是否支持承載令牌授權。

註冊範圍設定為 sso-session。範圍是限制應 OAuth 2.0 應用程式存取使用者帳戶的機制。應用程式可以請求一個或多個範圍，並且發行給應用程式的訪問令牌僅限於授予的範圍。這些範圍定義為已註冊 OIDC 用戶端和用戶端擷取的存取字符進行授權所需請求的許可。如需支援的存取範圍選項，請參閱《AWS IAM Identity Center 使用指南》中的「[存取範圍](#)」。下列範例設定為 sso_registration_scopes 提供列出帳戶和角色的存取權。

```
[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
```

驗證 Token 會快取至 ~/.aws/sso/cache 目錄下的磁碟，且檔案名稱根據工作階段名稱。

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](#) 改用。

若要使用舊式的不可重新整理規劃，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 和設定來指定 sso_region 設定檔的使用者入口網站。您可以使用 sso_account_id 和 sso_role_name 設定來指定權限。

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
```

驗證 Token 會快取至 `~/.aws/sso/cache` 目錄下的磁碟，檔案名稱是以 `sso_start_url`。

IAM 身分中心憑證提供者設定

使用下列項目設定此功能：

`sso_start_url`-共享 AWS `config` 文件設置

指向組織的 IAM 身分中心存取入口網站的 URL。如需 IAM 身分中心存取入口網站的詳細資訊，請參閱 [使用 AWS IAM Identity Center 者指南中的使用 AWS 存取入口網站](#)。

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

`sso_region`-共享 AWS `config` 文件設置

包含您的 AWS 區域 IAM 身分中心入口網站主機；也就是您在啟用 IAM 身分中心之前選取的區域。這與您的預設 AWS 區域無關，而且可以是不同的。

如需完整清單 AWS 區域 及其代碼，請參閱 Amazon Web Services 一般參考。若要尋找此值，請開啟 [IAM 身分中心主控台](#)、檢視儀表板，然後尋找 [區域]。

`sso_account_id`-共享 AWS `config` 文件設置

透過 AWS Organizations 服務新增以用於驗證的數值 ID。AWS 帳戶

若要查看可用帳戶的清單，請前往 [IAM 身分中心主控台](#) 並開啟 AWS 帳戶頁面。您也可以在 AWS IAM Identity Center 入口網站 API 參考資料中查看使用 [ListAccounts](#) API 方法的可用帳戶清單。例如，您可以調用 AWS CLI 方法 [列表帳戶](#)。

`sso_role_name`-共享 AWS `config` 文件設置

佈建為 IAM 角色的權限集名稱，可定義使用者產生的許可。角色必須存在於由 AWS 帳戶 指定的 `sso_account_id`。使用角色名稱，而不是角色 Amazon 資源名稱 (ARN)。

權限集已附加 IAM 政策和自訂許可政策，並定義使用者對其指派的存取層級 AWS 帳戶。

若要查看每個可用權限集的清單 AWS 帳戶，請前往 [IAM 身分中心主控台](#) 並開啟 AWS 帳戶頁面。選擇表格中列出的正確權限集名 AWS 帳戶 稱。您也可以在 AWS IAM Identity Center 入口網站 API 參考中查看使用 [ListAccountRoles](#) API 方法的可用權限集清單。例如，您可以呼叫方 AWS CLI 方法 [list-account-roles](#)。

sso_registration_scopes-共享 AWS config 文件設置

要授權的有效範圍字串的逗號分隔清單。sso-session 範圍授權對 IAM Identity Center 承載字符授權端點的存取。sso:account:access 必須授予的最低範圍，才能從 IAM 身分中心服務取回重新整理權杖。如需支援的存取範圍字串，請參閱《AWS IAM Identity Center 使用指南》中的「[存取範圍](#)」。此設定不適用於舊版不可重新整理的組態。使用舊配置發行的令牌僅限於 sso:account:access 隱式範圍。

與 AWS 開發套件的相容性

下列 SDK 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。和唯一支援任何 JVM 系統屬 AWS SDK for Java 性設 適用於 Kotlin 的 AWS SDK 定。

SDK	支 援	備註或更多資訊
AWS CLI v2	是	
SDK for C++	是	
適用於轉到 V2 的開發套件 (1.x)	是	
適用於 Go 1.x 的開發套件 (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 。
適用於 Java 2.x 的開發套件	是	credentials 檔案中也支援組態值。
適用於 Java 1.x 的開發套件	否	
適用於 JavaScript 3.x 的開發套	是	
適用於 JavaScript 2.x 的開發套	是	
SDK 的科特林	是	
適用於 .NET 3.x 的開發套	是	

SDK	支援	備註或更多資訊
適用於 PHP 3.x 的開發套件	是	
SDK for Python (Boto3)	是	
適用於紅寶石 3.x 的 SDK	是	
適用於鏽蝕的 SDK	部分	僅限舊式不可重新整理的組態。
適用的工具 PowerShell	是	

IMDS 認證提供者

執行個體中繼資料服務 (IMDS) 提供執行個體的相關資料，讓您用來設定或管理執行中的執行個體。[如需有關可用資料的詳細資訊，請參閱 Amazon EC2 使用者指南中的執行個體中繼資料和使用者資料](#)，或 Amazon EC2 使用者指南中的[執行個體中繼資料和使用者資料](#)。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資訊。如果執行個體附加了角色，它可以提供一組對該角色有效的認證。SDK 可以使用該端點將認證解析為其[預設憑證提供者鏈](#)的一部分。依預設，會使用執行個體中繼資料服務第 2 版 (IMDSv2)，這是使用工作階段 Token 的更安全的 IMDS 版本。如果由於不可重試的條件 (HTTP 錯誤代碼 403, 404, 405) 而失敗，則使用 IMDSv1 作為後備。

使用下列項目設定此功能：

AWS_EC2_METADATA_DISABLED-環境變量

是否嘗試使用 Amazon EC2 執行個體中繼資料服務 (IMDS) 取得登入資料。

預設值：false。

有效值：

- **true**— 請勿使用 IMDS 取得認證。
- **false**— 使用 IMDS 取得認證。

ec2_metadata_v1_disabled-共享 AWS config 文件設置, **AWS_EC2_METADATA_V1_DISABLED**-環境變量, **aws.disableEc2MetadataV1**-JVM 系統屬性：僅限爪哇/科特林

如果 IMDSv2 失敗，是否要使用執行個體中繼資料服務版本 1 (IMDSv1) 作為後援。

Note

新的開發套件不支援 IMDSv1，因此不支援此設定。如需詳細資訊，請參閱表格[與 AWS 開發套件的相容性](#)。

預設值：false。

有效值：

- **true**— 請勿使用 IMDSv1 作為後援。
- **false**— 使用 IMDSv1 做為後援。

ec2_metadata_service_endpoint-共享 AWS config 文件設置, **AWS_EC2_METADATA_SERVICE_ENDPOINT**-環境變量, **aws.ec2MetadataServiceEndpoint**-JVM 系統屬性：僅限爪哇/科特林

IMDS 的端點。

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：有效的 URI。

ec2_metadata_service_endpoint_mode-共享 AWS config 文件設置, **AWS_EC2_METADATA_SERVICE_ENDPOINT_MODE**-環境變量, **aws.ec2MetadataServiceEndpointMode**-JVM 系統屬性：僅限爪哇/科特林

IMDS 的端點模式。

預設值：IPv4。

有效值：IPv4、IPv6。

Note

IMDS 認證提供者是 [憑證提供者鏈](#)。不過，IMDS 認證提供者只會在本系列中的其他數個提供者之後進行檢查。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，請在程式碼中指定使用 IMDS 認證提供者，而不是

依賴認證提供者鏈來自動探索哪個提供者傳回有效認證。您可以在建立服務用戶端時直接指定認證來源。

IMDS 認證的安全性

依預設，如果 AWS SDK 未設定有效的登入資料，SDK 將嘗試使用 Amazon EC2 執行個體中繼資料服務 (IMDS) 來擷取 AWS 角色的登入資料。您可以將 `AWS_EC2_METADATA_DISABLED` 環境變數設定為來停用此行為 `true`。如此可避免不必要的網路活動，並在可能模擬 Amazon EC2 執行個體中繼資料服務的不受信任網路上增強安全性。

Note

AWS 使用有效認證設定的 SDK 用戶端永遠不會使用 IMDS 擷取認證，無論這些設定為何。

停用 Amazon EC2 IMDS 登入資料的使用

設定此環境變數的方式取決於使用中的作業系統，以及是否要持續變更。

Linux 和 macOS

使用 Linux 或 macOS 的客戶可以使用下列指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您的 shell 設定檔 `.bash_profile`，例如 `.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
```

如果您希望在多個 shell 會話中保持此設置，並且系統重新啟動可以使用以下命令：

```
$ setx AWS_EC2_METADATA_DISABLED=true
```

Note

該setx命令不會將該值應用於當前 shell 會話，因此您需要重新加載或重新打開 shell 以使更改生效。

與 AWS 開發套件的相容性

下列 SDK 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。和唯一支援任何 JVM 系統屬 AWS SDK for Java 性設 適用於 Kotlin 的 AWS SDK 定。

SDK	支 備註或更多資訊 援
AWS CLI v2	是
SDK for C++	是
適用於轉到 V2 的開發套件 (1.x)	是
適用於 Go 1.x 的開發套件 (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 。
適用於 Java 2.x 的開發套件	是
適用於 Java 1.x 的開發套件	部分 JVM 系統屬性：使用com.amazonaws.sdk.disableEc2MetadataV1 而不是aws.disableEc2MetadataV1 ；aws.ec2MetadataServiceEndpoint 且aws.ec2MetadataServiceEndpointMode 不支援。
適用於 JavaScript 3.x 的開發套件	是
適用於 JavaScript 2.x 的開發套件	是
SDK 的科特林	是 不使用 IMDSv1 後援。

SDK	支援	備註或更多資訊
適用於 .NET 3.x 的開發套	是	
適用於 PHP 3.x 的開發套件	是	
SDK for Python (Boto3)	是	
適用於紅寶石 3.x 的 SDK	是	
適用於鏽蝕的 SDK	是	不使用 IMDSv1 後援。
適用的工具 PowerShell	是	您可以在程式碼中使用. <code>[Amazon.Util.EC2InstanceMetadata]::EC2MetadataV1Disabled = \$true</code>

程序認證提供者

SDK 提供了一種擴展自定義用例的憑證提供者鏈的方法。

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。若要 `credential_process` 針對此用途進行配置，請參閱 [IAM Roles Anywhere](#)。

Warning

下列說明從外部處理程序取得認證的方法。這可能是潛在的危險，所以謹慎行事。如果可能的話，應該優先選擇其他認證提供者。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 `config` 檔案已盡可能鎖定。確認您的自訂認證工具不會將任何秘密資訊寫入 `StdErr`，因為 SDK 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 文件設置

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。該設置指定 SDK 將調用的程序/命令的名稱。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 [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，您可以在程式碼中指定使用處理序憑證提供者，而不是依賴認證提供者鏈來自動探索哪個提供者傳回有效的認證。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

該設置的值是一個字符串，其中包含 SDK 或開發工具代表您運行的程序路徑：

- 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。

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"
```

憑證程式的有效輸出

SDK 會依照設定檔中指定的方式執行命令，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 格式的時間戳記。如果 Expiration 金鑰不存在於工具的輸出中，SDK 會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 不會像假設角色認證那樣快取外部處理序認證。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。

與 AWS 開發套件的相容性

下列 SDK 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。和唯一支援任何 JVM 系統屬 AWS SDK for Java 性設 適用於 Kotlin 的 AWS SDK 定。

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 for C++	是	
適用於轉到 V2 的開發套件 (1.x)	是	
適用於 Go 1.x 的開發套件 (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 。
適用於 Java 2.x 的開發套件	是	
適用於 Java 1.x 的開發套件	是	
適用於 JavaScript 3.x 的開發套件	是	
適用於 JavaScript 2.x 的 SDK	是	
SDK 的科特林	是	
適用於 .NET 3.x 的開發套	是	
適用於 PHP 3.x 的開發套件	是	
SDK for Python (Boto3)	是	
適用於紅寶石 3.x 的 SDK	是	
適用於鏽蝕的 SDK	是	
適用的工具 PowerShell	是	

AWS SDK 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 SDK 中以相同的方式運作。跨多個 SDK 編碼時，這種一致性可提高生產力和清晰度。所有設置都可以在代碼中覆蓋，請參閱您的特定 SDK API 以獲取詳細信息。

⚠ Important

並非所有 SDK 都支援所有功能，甚至是功能中的所有層面。

主題

- [申請識別碼](#)
- [Amazon EC2 執行個體中繼資料](#)
- [Amazon S3 存取點](#)
- [Amazon S3 多區域存取點](#)
- [AWS 區域](#)
- [AWS STS 區域化端點](#)
- [雙堆疊和 FIPS 端點](#)
- [端點探索](#)
- [一般組態設定](#)
- [法團客戶端](#)
- [重試行為](#)
- [請求壓縮](#)
- [服務特定端點](#)
- [智慧型組態預設](#)

申請識別碼

一個 AWS 帳戶 可以由多個客戶應用程式使用，以撥打電話 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 SDK 和服務不會使用或解釋這個值，除了在客戶通訊中顯現它。例如，此值可包含在操作電子郵件中，或包含在中，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 文件設置, **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 進行呼叫 AWS。

預設值：None

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

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。

與 AWS 開發套件的相容性

下列 SDK 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。和唯一支援任何 JVM 系統屬 AWS SDK for Java 性設 適用於 Kotlin 的 AWS SDK 定。

SDK	支 援	備註或更多資訊
AWS CLI v2	否	
SDK for C++	是	不支援共用config檔案。
適用於轉到 V2 的開發套件 (1.x)	是	
適用於 Go 1.x 的開發套件 (V1)	否	

SDK	支援	備註或更多資訊
適用於 Java 2.x 的開發套件	部分	不支援共用config檔案設定；不支援環境變數。
適用於 Java 1.x 的開發套件	否	
適用於 JavaScript 3.x 的開發套件	是	
適用於 JavaScript 2.x 的 SDK	否	
SDK 的科特林	是	
適用於 .NET 3.x 的開發套件	是	不支援環境變數。
適用於 PHP 3.x 的開發套件	是	
SDK for Python (Boto3)	是	
適用於紅寶石 3.x 的 SDK	是	
適用於鏽蝕的 SDK	是	
適用的工具 PowerShell	否	

Amazon EC2 執行個體中繼資料

Amazon EC2 在稱為執行個體中繼資料服務 (IMDS) 的執行個體上提供服務。若要進一步了解此服務，請參閱 Amazon EC2 [使用者指南中的執行個體中繼資料和使用者資料](#)，或 Amazon EC2 [使用者指南中的執行個體中繼資料和使用者資料](#)。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 文件設置，
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 文件設置, **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
```

與 AWS 開發套件的相容性

下列 SDK 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。和唯一支援任何 JVM 系統屬 AWS SDK for Java 性設適用於 Kotlin 的 AWS SDK 定。

SDK	支援 備註或更多資訊
AWS CLI v2	是

SDK	支援	備註或更多資訊
SDK for C++	否	
適用於轉到 V2 的開發套件 (1.x)	否	
適用於 Go 1.x 的開發套件 (V1)	否	
適用於 Java 2.x 的開發套件	否	
適用於 Java 1.x 的開發套件	部分	僅支援 <code>AWS_METADATA_SERVICE_TIMEOUT</code> 。
適用於 JavaScript 3.x 的開發套	否	
適用於 JavaScript 2.x 的開發套	否	
SDK 的科特林	否	
適用於 .NET 3.x 的開發套	否	
適用於 PHP 3.x 的開發套件	是	
SDK for Python (Boto3)	是	
適用於紅寶石 3.x 的 SDK	否	
適用於鏽蝕的 SDK	否	
適用的工具 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。透過 AWS SDK，您可以在儲存貯體欄位中使用存

取點 Amazon 資源名稱 (ARN) 進行 API 操作，而不必明確指定值區名稱。它們用於特定操作，例如使用存取點 ARN 搭配 [GetObject](#) 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 存取點和 ARN，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 文件設置, **AWS_S3_USE_ARN_REGION**-環境變量, **aws.s3UseArnRegion**-JVM 系統屬性：僅限爪哇/科特林, 若要直接在程式碼中設定值，請直接洽詢您的特定 SDK。

此設定可控制 SDK 是否使用存取點 ARN AWS 區域 來建構要求的地區端點。SDK 會驗證 ARN AWS 區域 是否由與用戶端設定 AWS 區域 的相同 AWS 分割區提供服務，以防止跨分割區呼叫最有可能失敗。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 在構建端點而不是客戶端配置 AWS 區域 時使用 ARN。AWS 區域例外狀況：如果用戶端設定 AWS 區域 為 FIPS AWS 區域，則必須與 ARN 相符。AWS 區域否則將會發生錯誤。
- **false**— SDK 在構建端點 AWS 區域 時使用客戶端配置的。

與 AWS 開發套件的相容性

下列 SDK 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。和唯一支援任何 JVM 系統屬 AWS SDK for Java 性設 適用於 Kotlin 的 AWS SDK 定。

SDK	支 備註或更多資訊 援
AWS CLI v2	是
SDK for C++	是
適用於轉到 V2 的開發套件 (1.x)	是

SDK	支援	備註或更多資訊
適用於 Go 1.x 的開發套件 (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 。
適用於 Java 2.x 的開發套件	是	
適用於 Java 1.x 的開發套件	是	不支援 JVM 系統屬性。
適用於 JavaScript 3.x 的開發套件	是	
適用於 JavaScript 2.x 的 SDK	是	
SDK 的科特林	是	
適用於 .NET 3.x 的開發套件	是	不遵循標準優先順序；共享config檔案值優先於環境變數。
適用於 PHP 3.x 的開發套件	是	
SDK for Python (Boto3)	是	
適用於紅寶石 3.x 的 SDK	是	
適用於鏽蝕的 SDK	否	
適用的工具 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 AWS 區域 Amazon S3 儲存貯體的請求。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 (ARN)，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。SDK 透過相依性取得此演算法 [AWS 通用執行階段 \(CRT\) 程式庫](#)。

使用下列項目設定此功能：

s3_disable_multiregion_access_points-共享 AWS config 文件設置, **AWS_S3_DISABLE_MULTIREGION_ACCESS_POINTS**-環境變量, **aws.s3DisableMultiRegionAccessPoints**-JVM 系統屬性：僅限爪哇/科特林, 若要直接在程式碼中設定值，請直接洽詢您的特定 SDK。

此設定可控制 SDK 是否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。

與 AWS 開發套件的相容性

下列 SDK 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。和唯一支援任何 JVM 系統屬 AWS SDK for Java 性設 適用於 Kotlin 的 AWS SDK 定。

SDK	支 備註或更多資訊 援
AWS CLI v2	是
SDK for C++	是
適用於轉到 V2 的開發套件 (1.x)	是
適用於 Go 1.x 的開發套件 (V1)	否
適用於 Java 2.x 的開發套件	是

SDK	支援	備註或更多資訊
適用於 Java 1.x 的開發套件	否	
適用於 JavaScript 3.x 的開發套件	是	
適用於 JavaScript 2.x 的 SDK	否	
SDK 的科特林	是	
適用於 .NET 3.x 的開發套	是	
適用於 PHP 3.x 的開發套件	是	
SDK for Python (Boto3)	是	
適用於紅寶石 3.x 的 SDK	是	
適用於鏽蝕的 SDK	是	
適用的工具 PowerShell	是	

AWS 區域

AWS 區域 是使用時要理解的重要概念 AWS 服務。

使用 AWS 區域，您可以存取 AWS 服務 實際位於特定地理區域中的內容。這有助於讓您的資料和應用程式在接近您和使用者存取資料的位置執行。區域提供容錯能力、穩定性和恢復能力，也可降低延遲。使用 Region，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。您在一個區域中建立的資源不存在於任何其他區域中，除非您明確使用 AWS 服務。例如，Amazon S3 和 Amazon EC2 支援跨區域複寫。某些服務 (例如 IAM) 沒有區域資源。

AWS 一般參考包含有關以下內容的資訊：

- 若要瞭解區域與端點之間的關係，並檢視現有地區端點的清單，請參閱[AWS 服務端點](#)。
- 若要檢視每個區域和端點的目前所有支援區域和端點清單 AWS 服務，請參閱[服務端點和配額](#)。

建立服務用戶端

為了以編程方式訪問 AWS 服務，SDK 使用每個客戶端類/對象。AWS 服務例如，如果您的應用程式需要存取 Amazon EC2，您的應用程式會建立一個 Amazon EC2 用戶端物件，以便與該服務進行交流。

如果沒有為用戶端明確指定 [區域]，用戶端會預設使用透過下列 region 設定所設定的 [地區]。但是，可以為任何單獨的客戶端對象明確設置客戶端的活動區域。以這種方式設定區域的優先順序高於該特定服務用戶端的任何全域設定。替代區域是在實例化該客戶端期間指定的，特定於您的 SDK (請查看您的特定 SDK 指南或 SDK 的代碼庫)。

使用下列項目設定此功能：

region-共享 AWS config 文件設置, **AWS_REGION**-環境變量, **aws.region**-JVM 系統屬性：僅限爪哇/科特林

指定要用於 AWS 請求的預設值 AWS 區域。此區域用於未提供特定區域以供使用的 SDK 服務請求。

預設值：無。您必須明確指定此值。

有效值：

- 適用於所選服務的任何區域代碼，如「AWS 一般參考」中的[AWS 服務端點](#)所列。例如，此值會將端點 us-east-1 設定為 AWS 區域 美國東部 (維吉尼亞北部)。
- aws-global 指定除區域端點 AWS Security Token Service (AWS STS) 和 Amazon 簡單儲存服務 (Amazon S3) 等區域端點之外，還支援個別全球端點的服務的全球端點。

在 config 檔案中設定此值的範例：

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

通過命令行設置環境變量的 Linux/macOS 示例：

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

通過命令行設置環境變量的 Windows 示例：

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

大多數 SDK 都有一個「配置」對象，可用於從應用程序代碼中設置默認區域。如需詳細資訊，請參閱特定的 AWS SDK 開發人員指南。

與 AWS 開發套件的相容性

下列 SDK 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。和唯一支援任何 JVM 系統屬 AWS SDK for Java 性設 適用於 Kotlin 的 AWS SDK 定。

SDK	支援	備註或更多資訊
AWS CLI v2	是	AWS CLI v2 在中的任何值之AWS_REGION 前使用中的任何值AWS_DEFAULT_REGION （兩個變量都被檢查）。
AWS CLI V1	是	AWS CLI v1 使用AWS_DEFAULT_REGION 為此目的命名的環境變量。
SDK for C++	是	
適用於轉到 V2 的開發套件 (1.x)	是	
適用於 Go 1.x 的開發套件 (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 。
適用於 Java 2.x 的開發套件	是	
適用於 Java 1.x 的開發套件	是	
適用於 JavaScript 3.x 的開發套件	是	
適用於 JavaScript 2.x 的 SDK	是	
SDK 的科特林	是	
適用於 .NET 3.x 的開發套	是	
適用於 PHP 3.x 的開發套件	是	

SDK	支援	備註或更多資訊
SDK for Python (Boto3)	是	此 SDK 使用 <code>AWS_DEFAULT_REGION</code> 為此目的命名的環境變數。
適用於紅寶石 3.x 的 SDK	是	
適用於鏽蝕的 SDK	是	
適用的工具 PowerShell	是	

AWS STS 區域化端點

預設情況下，AWS Security Token Service (AWS STS) 可作為全域服務使用，且所有 AWS STS 要求都會傳送至位於的單一端點 `https://sts.amazonaws.com`。全域要求對應至美國東部 (維吉尼亞北部) 區域。AWS 建議使用區域 AWS STS 端點而非全域端點。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](#)。

使用下列項目設定此功能：

sts_regional_endpoints-共享 AWS `config` 文件設置, **AWS_STS_REGIONAL_ENDPOINTS**-環境變量

此設定指定 SDK 或工具如何判斷它用來與 AWS Security Token Service (AWS STS) 通訊的 AWS 服務端點。

預設值：`legacy`

Note

在 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 future 對此變更的影響，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**— 針對下列 AWS 區域使用全域 AWS STS 端點：`ap-northeast-1`、`ap-south-1`、`ap-southeast-1`、`ap-southeast-2`、`aws-global`、`ca-central-1`、`eu-`

central-1、eu-north-1、eu-west-1、eu-west-2、eu-west-3、sa-east-1、us-east-1、us-east-2、us-west-1、和us-west-2。sts.amazonaws.com所有其他區域會自動使用其各自的區域端點。

- **regional**— SDK 或工具一律會針對目前設定的「區域」使用 AWS STS 端點。例如，如果用戶端設定為使用us-west-2，則會對區域端點進 AWS STS 行的所有呼叫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。

在檔案中設定這些值config的範例：

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

通過命令行設置環境變量的 Linux/macOS 示例：

```
export AWS_STS_REGIONAL_ENDPOINTS=regional
```

通過命令行設置環境變量的 Windows 示例：

```
setx AWS_STS_REGIONAL_ENDPOINTS regional
```

與 AWS 開發套件的相容性

下列 SDK 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。和唯一支援任何 JVM 系統屬 AWS SDK for Java 性設 適用於 Kotlin 的 AWS SDK 定。

SDK	支 備註或更多資訊 援
AWS CLI v2	部 預設值為 regional。 分
SDK for C++	部 不支援環境變數和config檔案設定。SDK 與regional設置一 分 起執行。
適用於轉到 V2 的開發套件 (1.x)	是

SDK	支援	備註或更多資訊
適用於 Go 1.x 的開發套件 (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 。
適用於 Java 2.x 的開發套件	是	
適用於 Java 1.x 的開發套件	是	
適用於 JavaScript 3.x 的開發套件	是	
適用於 JavaScript 2.x 的開發套件	是	
SDK 的科特林	否	
適用於 .NET 3.x 的開發套	是	
適用於 PHP 3.x 的開發套件	是	
SDK for Python (Boto3)	是	
適用於紅寶石 3.x 的 SDK	是	
適用於鏽蝕的 SDK	是	
適用的工具 PowerShell	是	

雙堆疊和 FIPS 端點

使用下列項目設定此功能：

use_dualstack_endpoint-共享 AWS config 文件設置, **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 SDK 是否會傳送要求至雙堆疊端點。若要進一步了解支援 IPv4 和 IPv6 流量的雙堆疊端點，請參閱《Amazon Simple Storage Service 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/或雙堆疊端點不存在 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 文件設置, **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 或工具是否會將要求傳送至 FIPS 相容的端點。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、合作夥伴以及那些想要與聯邦政府開展業務往來的人都必須遵守 FIPS 準則。與標準 AWS 端點不同，FIPS 端點使用符合 FIPS 140-2 的 TLS 軟體程式庫。如果啟用此設定，且您的服務不存在 FIPS 端點 AWS 區域，則 AWS 呼叫可能會失敗。[服務特定端點](#)和用於 `--endpoint-url` AWS Command Line Interface 覆蓋此設置的選項。

若要深入了解指定 FIPS 端點的其他方法 AWS 區域，請參閱 [FIPS 端點 \(依服務分類\)](#)。如需有關 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 相容的端點。
- **false**— SDK 或工具不會將要求傳送至 FIPS 相容的端點。

與 AWS 開發套件的相容性

下列 SDK 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。和唯一支援任何 JVM 系統屬 AWS SDK for Java 性設 適用於 Kotlin 的 AWS SDK 定。

SDK	支 備註或更多資訊 援
AWS CLI v2	是
SDK for C++	是

SDK	支援	備註或更多資訊
適用於轉到 V2 的開發套件 (1.x)	是	
適用於 Go 1.x 的開發套件 (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 。
適用於 Java 2.x 的開發套件	是	
適用於 Java 1.x 的開發套件	否	
適用於 JavaScript 3.x 的開發套件	是	
適用於 JavaScript 2.x 的 SDK	是	
SDK 的科特林	是	
適用於 .NET 3.x 的開發套件	是	
適用於 PHP 3.x 的開發套件	是	
SDK for Python (Boto3)	是	
適用於紅寶石 3.x 的 SDK	是	
適用於鏽蝕的 SDK	是	
適用的工具 PowerShell	是	

端點探索

SDK 使用端點探索來存取服務端點 (用於存取各種資源的 URL)，同時依然保持視需 AWS 要變更 URL 的彈性。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，SDK 會提出擷取可用端點的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 文件設置,
AWS_ENABLE_ENDPOINT_DISCOVERY-環境變量, **aws.endpointDiscoveryEnabled**-JVM 系統屬性：僅限爪哇/科特林, 若要直接在程式碼中設定值, 請直接洽詢您的特定 SDK。

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 應自動嘗試探索端點為選用的服務探索端點。
- **false**— SDK 不應自動嘗試探索端點為選用的服務探索端點。

與 AWS 開發套件的相容性

下列 SDK 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。和唯一支援任何 JVM 系統屬 AWS SDK for Java 性設 適用於 Kotlin 的 AWS SDK 定。

SDK	支 援	備註或更多資訊
AWS CLI v2	是	
SDK for C++	是	
適用於轉到 V2 的開發套件 (1.x)	是	
適用於 Go 1.x 的開發套件 (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 。
適用於 Java 2.x 的開發套件	是	適用於 Java 2.x 的 SDK 用 <code>AWS_ENDPOINT_DISCOVERY_ENABLED</code> 於環境變數名稱。
適用於 Java 1.x 的開發套件	部分	不支援 JVM 系統屬性。

SDK	支援	備註或更多資訊
適用於 JavaScript 3.x 的開發套	是	
適用於 JavaScript 2.x 的 SDK	是	
SDK 的科特林	是	
適用於 .NET 3.x 的開發套	是	
適用於 PHP 3.x 的開發套件	是	
SDK for Python (Boto3)	是	
適用於紅寶石 3.x 的 SDK	是	
適用於鏽蝕的 SDK	部分	僅支援「時間流」。
適用的工具 PowerShell	是	

一般組態設定

SDK 支援一些用來設定整體 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 文件設置

有些 AWS 服務會維護多個 API 版本，以支援回溯相容性。根據預設，SDK 和 AWS CLI 作業會使用最新的可用 API 版本。要要求特定的 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。（ SDK 使用了最新的 API 版本。 ）

有效值：這是巢狀設定，後面接著一個或多個縮排行，每個行都會識別一個 AWS 服務和要使用的 API 版本。請參閱 AWS 服務的說明文件，以瞭解哪些 API 版本可用。

此範例會為檔案中的兩個 AWS 服務設定特定的 API 版本。這些 API 版本僅適用於在包含這些設定的描述檔下執行的命令。任何其他服務的命令都會使用該服務 API 的最新版本。

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

ca_bundle-共享 AWS config 文件設置, AWS_CA_BUNDLE-環境變量

指定建立 SSL/TLS 連線時要使用的自訂憑證服務包 (.pem 副檔名為副檔名的檔案) 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
```

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 文件設置

指定如何在以 AWS CLI 及其他 AWS SDK 和工具中格式化結果。

預設值：json

有效值：

- **json** – 輸出的格式為 [JSON](#) 字串。
- **yaml** – 輸出的格式為 [YAML](#) 字串。
- **yaml-stream** – 輸出採用串流方式且格式為 [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 文件設置

指定 SDK 或工具在將命令列參數傳送至 AWS 服務端點之前是否嘗試驗證命令列參數。

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 或工具可以在將要求傳送至 AWS 服務端點之前拒絕無效的要求。
- **false**— SDK 或工具在將命令列參數傳送至 AWS 服務端點之前，不會驗證命令列參數。AWS 服務端點負責驗證所有要求，並拒絕無效的要求。

與 AWS 開發套件的相容性

下列 SDK 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。和唯一支援任何 JVM 系統屬 AWS SDK for Java 性設 適用於 Kotlin 的 AWS SDK 定。

SDK	支 備註或更多資訊 援
AWS CLI v2	部 api_versions 不支援。 分
SDK for C++	是
適用於轉到 V2 的開發套件 (1.x)	部 api_versions 並且parameter_validation 不支持。 分
適用於 Go 1.x 的開發套件 (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 。
適用於 Java 2.x 的開發套件	否

SDK	支援	備註或更多資訊
適用於 Java 1.x 的開發套件	否	
適用於 JavaScript 3.x 的開發套件	是	
適用於 JavaScript 2.x 的開發套件	是	
SDK 的科特林	否	
適用於 .NET 3.x 的開發套件	否	
適用於 PHP 3.x 的開發套件	是	
SDK for Python (Boto3)	是	
適用於紅寶石 3.x 的 SDK	是	
適用於鏽蝕的 SDK	否	
適用的工具 PowerShell	否	

法團客戶端

SDK 會使用工作階段導向要求來實作執行個體中繼資料服務第 2 版 (IMDSv2) 用戶端。如需有關 IMDSv2 的詳細資訊，請參閱 Amazon EC2 [使用者指南中的使用 ImDSv2](#) 或亞馬 Amazon EC2 使用者指南中的[使用 ImDSv2](#)。IMDS 用戶端可透過 SDK 程式碼庫中提供的用戶端設定物件進行設定。

使用下列項目設定此功能：

retries-客戶端配置對象成員

任何失敗要求的其他重試嘗試次數。

預設值：3

有效值：大於 0 的數字。

port-客戶端配置對象成員

端點的連接埠。

預設值：80

有效值：數字。

token_ttl-客戶端配置對象成員

權杖的 TTL。

默認值：21,600 秒（6 小時，最大分配的時間）。

有效值：數字。

endpoint-客戶端配置對象成員

IMDS 的端點。

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：有效的 URI。

大多數 SDK 都支援下列選項。如需詳細資訊，請參閱特定的 SDK 程式碼庫。

endpoint_mode-客戶端配置對象成員

IMDS 的端點模式。

預設值：IPv4

有效值：IPv4、IPv6

http_open_timeout-客戶端配置對象成員（名稱可能會有所不同）

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-客戶端配置對象成員（名稱可能會有所不同）

讀取一個資料區塊的秒數。

預設值：1 秒。

有效值：大於 0 的數字。

http_debug_output-客戶端配置對象成員 (名稱可能會有所不同)

設置調試輸出流。

預設值：無。

有效值：一個有效的 I/O 流，如標準輸出。

backoff-客戶端配置對象成員 (名稱可能會有所不同)

重試或客戶提供的輪詢功能進行呼叫之間的睡眠時間秒數。這會覆寫預設的指數輪詢策略。

預設值：依 SDK 而異。

有效值：依 SDK 而異。可以是數值，也可以是對自訂函數的呼叫。

與 AWS 開發套件的相容性

下列 SDK 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。和唯一支援任何 JVM 系統屬 AWS SDK for Java 性設 適用於 Kotlin 的 AWS SDK 定。

SDK	支 備註或更多資訊 援
AWS CLI v2	是
SDK for C++	否 僅在內部使用。請參閱 IMDS 認證提供者 。
適用於轉到 V2 的開發套件 (1.x)	是
適用於 Go 1.x 的開發套件 (V1)	是
適用於 Java 2.x 的開發套件	是
適用於 Java 1.x 的開發套件	是

SDK	支援	備註或更多資訊
適用於 JavaScript 3.x 的開發套	是	
適用於 JavaScript 2.x 的開發套	是	
SDK 的科特林	是	
適用於 .NET 3.x 的開發套	是	
適用於 PHP 3.x 的開發套件	是	
SDK for Python (Boto3)	是	
適用於紅寶石 3.x 的 SDK	是	
適用於鏽蝕的 SDK	是	
適用的工具 PowerShell	是	

重試行為

重試行為包括有關 SDK 如何嘗試從發出請求而導致的失敗中恢復的設置。AWS 服務

使用下列項目設定此功能：

max_attempts-共享 AWS config 文件設置, **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，則其預設值取決於 `retry_mode` 設定值：

- 如果 `retry_mode` 是 `legacy` — 使用特定於您的 SDK 的默認值 (`max_attempts` 默認情況下檢查您的特定 SDK 指南或 SDK 的代碼庫)。
- 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。

retry_mode-共享 AWS config 文件設置, **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：legacy 是預設的重試策略。

有效值：

- legacy— 特定於您的 SDK (請查看您的特定 SDK 指南或 SDK 的代碼庫)。
- standard— 跨 AWS SDK 的標準重試規則集。此模式包含重試的一組標準錯誤，並支援重試配額。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— 實驗性重試模式，其中包含標準模式的功能，但包含自動用戶端節流。由於此模式是實驗性的，因此 future 可能會改變行為。

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，您提供的邏輯可依據正在存取的服務資源來分別集區用戶端。
支援斷路，以防止 SDK 在中斷期間重試。	支援斷路，以防止 SDK 在中斷期間重試。
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。資源，在這種情況下，比只是思考每一個更精細的調整。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 會使用 AWS 區域 加上正在存取的表格來限制請求。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。您的代碼應該被設計為每個 R egion-and-table 對有一個客戶端。如果您在使用adaptive模式時遇到非預期的延遲，請參閱所使用服務的特定 AWS 文件指南。

重試模式實作詳細資

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:

令牌存儲桶僅在adaptive重試模式下使用。令牌存儲桶通過要求可用令牌以啟動請求來強制執行最大請求率。SDK 客戶端可配置為快速失敗請求或阻止，直到令牌變為可用。

客戶端速率限制是一種算法，最初允許以任何速率發出請求，最高可達令牌限額。但是，在偵測到節流回應之後，用戶端會受到相應 rate-of-request 的限制。如果收到成功的回應，令牌額度也會相應地增加。

使用自適應速率限制，SDK 可以減慢傳送要求的速率，以便更好地容納的 AWS 服務容量。

SendHTTPRequest:

大多數 AWS SDK 使用使用連接池的 HTTP 庫，以便您可以在發出 HTTP 請求時重複使用現有連接。一般而言，由於節流錯誤而重試要求時，會重複使用連線。由於暫時性錯誤而重試時，請求不會重複使用。

RequestBookkeeping:

如果請求成功，則應更新重試配額。僅針對adaptive重試模式，state 變數maxsendrate會根據收到的回應類型進行更新。

Retryable:

此步驟會根據下列項目決定是否可以重試回應：

- HTTP 狀態碼。
- 從服務傳回的錯誤碼。
- 連接錯誤，定義為 SDK 收到的任何錯誤，其中未收到來自服務的 HTTP 響應。

暫時性錯誤 (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) 都可能會重試。SDK 重試行為是與服務中的錯誤代碼或其他數據一起確定的。

MAX_ATTEMPTS:

由config檔案設定或環境變數指定。

HasRetryQuota

此步驟通過要求重試配額存儲桶中提供令牌來限制重試請求。重試配額值區是一種機制，可防止不太可能成功的重試。這些配額依賴於 SDK，通常依賴於用戶端，有時甚至依賴於服務端點。當請求因各種原因而失敗時，可用的重試配額令牌將被刪除，並在成功時進行補充。當沒有令牌保留時，重試循環將退出。

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。SDK 使用具有抖動的截斷二進制指數輪詢。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 適用於大多數 SDK。請參閱您的特定 SDK 指南或原始碼以進行確認。

與 AWS 開發套件的相容性

下列 SDK 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。和唯一支援任何 JVM 系統屬 AWS SDK for Java 性設 適用於 Kotlin 的 AWS SDK 定。

SDK	支 備註或更多資訊 援
AWS CLI v2	是
SDK for C++	是
適用於轉到 V2 的開發套件 (1.x)	是
適用於 Go 1.x 的開發套件 (V1)	否
適用於 Java 2.x 的開發套件	是
適用於 Java 1.x 的開發套件	是 JVM 系統屬性：使用 <code>com.amazonaws.sdk.maxAttempts</code> 而不是 <code>aws.maxAttempts</code> ；使用 <code>com.amazonaws.sdk.retryMode</code> 而不是 <code>aws.retryMode</code> 。
適用於 JavaScript 3.x 的開發套	是

SDK	支援	備註或更多資訊
適用於 JavaScript 2.x 的開發套件	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 的科特林	是	
適用於 .NET 3.x 的開發套件	是	
適用於 PHP 3.x 的開發套件	是	
SDK for Python (Boto3)	是	
適用於紅寶石 3.x 的 SDK	是	
適用於鏽蝕的 SDK	是	
適用的工具 PowerShell	是	

請求壓縮

AWS 當向支持接收壓縮有效負載的請求發送請求時，SDK 和工具可以自動壓縮有效負載。AWS 服務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。針對壓縮，SDK 或工具會選取服務和 SDK 都支援的編碼演算法。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 是 [PutMetricDataAPI](#) 方法。

使用下列項目設定此功能：

disable_request_compression-共享 AWS config 文件設置，
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** 文件設置, **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)。

與 AWS 開發套件的相容性

下列 SDK 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。和唯一支援任何 JVM 系統屬 AWS SDK for Java 性設 適用於 Kotlin 的 AWS SDK 定。

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

SDK	支援	備註或更多資訊
適用於 JavaScript 2.x 的開發套	否	
SDK 的科特林	是	
適用於 .NET 3.x 的開發套	是	
適用於 PHP 3.x 的開發套件	是	
SDK for Python (Boto3)	是	
適用於紅寶石 3.x 的 SDK	是	
適用於鏽蝕的 SDK	是	
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 端點和第三方本機 AWS 開發環境的彈性。不同的端點可用於測試和生產環境。您可以為個人 AWS 服務指定端點 URL。

使用下列項目設定此功能：

endpoint_url-共享 AWS config 文件設置, **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。

您也可以使用共用 AWS config 檔案的某個 services 區段內使用此設定，為特定服務設定自訂端點。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：包括端點的配置和主機 URL。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>(其中<SERVICE>是 AWS 服務 識別碼) 會為特定服務設定自訂端點。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：包括端點的配置和主機 URL。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 文件設置, **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設定。

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 `--endpoint-url` 命令行參數與 AWS CLI 命令或將端點 URL 傳遞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
```

若要路由傳送特定 AWS 服務 至自訂端點 URL 的要求，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serviceId一個 [Elastic Beanstalk](#)。標 AWS 服務 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 [配置文件的格式](#)。

下列範例使用定services義來設定用於 Amazon S3 的服務特定端點 URL，以及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。此範例顯示如何為 Amazon S3 和 AWS Elastic Beanstalk 同一個設定檔設定服務特定端點 URL。AWS Elastic Beanstalk 有一個serviceId的[Elastic Beanstalk](#)。標 AWS 服務 識符基於 API 模型的serviceId方法是用下劃線替換所有空格並將所有字母小寫。因此，服務標識符密鑰變為，elastic_beanstalk並且此服務的設置開始在線上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 `source_profile` 參數設定用於 IAM 假設角色功能，以角色為基礎的憑證，則 SDK 只會使用指定設定檔的服務組態。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 Amazon EC2，則端點會解析為 `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[假設角色認證提供者](#)。

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。對於對給定的 API 調用 AWS 服務，使用以下順序來選擇一個值：

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，這是指 `--endpoint-url` 命令行參數所提供的值。對於 SDK，明確指派可以採用您在實例化 AWS 服務用戶端或配置物件時所設定的參數形式。
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. 最後會使用相應 AWS 服務的任何預設端點 URL。

與 AWS 開發套件的相容性

下列 SDK 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。和唯一支援任何 JVM 系統屬 AWS SDK for Java 性設 適用於 Kotlin 的 AWS SDK 定。

SDK	支 備註或更多資訊 援
AWS CLI v2	是
SDK for C++	否
適用於轉到 V2 的開發套件 (1.x)	是
適用於 Go 1.x 的開發套件 (V1)	否

SDK	支援	備註或更多資訊
適用於 Java 2.x 的開發套件	否	
適用於 Java 1.x 的開發套件	否	
適用於 JavaScript 3.x 的開發套件	是	
適用於 JavaScript 2.x 的開發套件	否	
SDK 的科特林	是	
適用於 .NET 3.x 的開發套件	是	
適用於 PHP 3.x 的開發套件	是	
SDK for Python (Boto3)	是	
適用於紅寶石 3.x 的 SDK	是	
適用於鏽蝕的 SDK	否	
適用的工具 PowerShell	是	

服務特定端點的識別碼

如需有關如何以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 AWS_ENDPOINT_URL_<SERVICE> 環境變數 AWS 帳戶的 服務識別碼 金鑰
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 AWS_ENDPOINT_URL_<SERVICE> 環境變數 API 方案的服務識別碼金鑰
API Gateway	api AWS_ENDPOINT_URL_API_GATEWAY
ApiGatewayManagementApi	api AWS_ENDPOINT_URL_APIGATEWAYMANAGEMENTAPI
ApiGatewayV2	api AWS_ENDPOINT_URL_APIGATEWAYV2
AppConfig	api AWS_ENDPOINT_URL_APPCONFIG
AppConfigData	api AWS_ENDPOINT_URL_APPCONFIGDATA
AppFabric	api AWS_ENDPOINT_URL_APPFABRIC
Appflow	api AWS_ENDPOINT_URL_APPFLOW
AppIntegrations	api AWS_ENDPOINT_URL_APPINTEGRATIONS

serviceId	共用 AWS_ENDPOINT_URL_<SERVICE> 環境變數 案例的服務識別碼金鑰
Application Auto Scaling	a AWS_ENDPOINT_URL_APPLICATION_AUTO_SCALING o c
Application Insights	a AWS_ENDPOINT_URL_APPLICATION_INSIGHTS o t
ApplicationCostProfiler	a AWS_ENDPOINT_URL_APPLICATIONCOSTPROFILER o f
App Mesh	a AWS_ENDPOINT_URL_APP_MESH
AppRunner	a AWS_ENDPOINT_URL_APPRUNNER
AppStream	a AWS_ENDPOINT_URL_APPSTREAM
AppSync	a AWS_ENDPOINT_URL_APPSYNC
ARC Zonal Shift	a: AWS_ENDPOINT_URL_ARC_ZONAL_SHIFT _:

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

serviceId	共用 API 案例 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
Bedrock	b:	AWS_ENDPOINT_URL_BEDROCK
Bedrock Agent	b: g:	AWS_ENDPOINT_URL_BEDROCK_AGENT
Bedrock Agent Runtime	b: g: i:	AWS_ENDPOINT_URL_BEDROCK_AGENT_RUNTIME
Bedrock Runtime	b: u:	AWS_ENDPOINT_URL_BEDROCK_RUNTIME
billingconductor	b: n:	AWS_ENDPOINT_URL_BILLINGCONDUCTOR
Braket	b:	AWS_ENDPOINT_URL_BRAKET
Budgets	b:	AWS_ENDPOINT_URL_BUDGETS
Cost Explorer	c: o:	AWS_ENDPOINT_URL_COST_EXPLORER

serviceId	共用 AWS_ENDPOINT_URL_<SERVICE> 環境變數 Amazon CloudFormation 的 服務識別碼金鑰
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 _v

serviceId	共用 AWS_ENDPOINT_URL_<SERVICE> 環境變數 案例的服務識別碼金鑰
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 KeyValueStore	c: AWS_ENDPOINT_URL_CLOUDFRONT_KEYVALUESTORE

serviceId	共用 AWS_ENDPOINT_URL_<SERVICE> 環境變數 案例的服務識別碼金鑰
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 AWS_ENDPOINT_URL_<SERVICE> 環境變數 AWS 方案的服務識別碼金鑰
CodeBuild	codebuild AWS_ENDPOINT_URL_CODEBUILD
CodeCatalyst	codecatalyst AWS_ENDPOINT_URL_CODECATALYST
CodeCommit	codecommit AWS_ENDPOINT_URL_CODECOMMIT
CodeDeploy	codedeploy AWS_ENDPOINT_URL_CODEDEPLOY
CodeGuru Reviewer	codeguru-reviewer AWS_ENDPOINT_URL_CODEGURU_REVIEWER
CodeGuru Security	codeguru-security AWS_ENDPOINT_URL_CODEGURU_SECURITY
CodeGuruProfiler	codeguru-profiler AWS_ENDPOINT_URL_CODEGURUPROFILER
CodePipeline	codepipeline AWS_ENDPOINT_URL_CODEPIPELINE

serviceId	共用 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 AWS_ENDPOINT_URL_<SERVICE> 環境變數 Amazon Connect 的 服務識別碼金鑰
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 AWS_ENDPOINT_URL_<SERVICE> 環境變數 方案的服務識別碼金鑰
ControlTower	c: AWS_ENDPOINT_URL_CONTROLTOWER w:
Cost Optimization Hub	c: AWS_ENDPOINT_URL_COST_OPTIMIZATION_HUB m: h:
Cost and Usage Report Service	c: AWS_ENDPOINT_URL_COST_AND_USAGE_REPO u: RT_SERVICE o: c:
Customer Profiles	c: AWS_ENDPOINT_URL_CUSTOMER_PROFILES p:
DataBrew	d: AWS_ENDPOINT_URL_DATABREW
DataExchange	d: AWS_ENDPOINT_URL_DATAEXCHANGE n:
Data Pipeline	d: AWS_ENDPOINT_URL_DATA_PIPELINE l:
DataSync	d: AWS_ENDPOINT_URL_DATASYNC

serviceId	共用 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 A W S C I A S 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 AWS_ENDPOINT_URL_<SERVICE> 環境變數 方案的服務識別碼金鑰
EC2 Instance Connect	e: AWS_ENDPOINT_URL_EC2_INSTANCE_CONNECT
ECR	e: AWS_ENDPOINT_URL_ECR
ECR PUBLIC	e: AWS_ENDPOINT_URL_ECR_PUBLIC
ECS	e: AWS_ENDPOINT_URL_ECS
EFS	e: AWS_ENDPOINT_URL_EFS
EKS	e: AWS_ENDPOINT_URL_EKS
EKS Auth	e: AWS_ENDPOINT_URL_EKS_AUTH
Elastic Inference	e: AWS_ENDPOINT_URL_ELASTIC_INFERENCE
ElastiCache	e: AWS_ENDPOINT_URL_ELASTICACHE
Elastic Beanstalk	e: AWS_ENDPOINT_URL_ELASTIC_BEANSTALK

serviceId	共用 AWS_ENDPOINT_URL_<SERVICE> 環境變數 Amazon CloudFormation 的 服務識別碼金鑰
Elastic Transcoder	e: AWS_ENDPOINT_URL_ELASTIC_TRANSCODER r:
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:
EntityResolution	er: AWS_ENDPOINT_URL_ENTITYRESOLUTION o:

serviceId	共用 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 AWS_ENDPOINT_URL_<SERVICE> 環境變數 Amazon CloudFormation 的 服務識別碼金鑰
FraudDetector	f: AWS_ENDPOINT_URL_FRAUDETECTOR cf
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 cf
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 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	共用 AWS_ENDPOINT_URL_<SERVICE> 環境變數 方案的服務識別碼金鑰
Inspector	i AWS_ENDPOINT_URL_INSPECTOR
Inspector Scan	i AWS_ENDPOINT_URL_INSPECTOR_SCAN_
Inspector2	i AWS_ENDPOINT_URL_INSPECTOR2 2
InternetMonitor	i AWS_ENDPOINT_URL_INTERNETMONITOR o
IoT	i AWS_ENDPOINT_URL_IOT
IoT Data Plane	i AWS_ENDPOINT_URL_IOT_DATA_PLANE p
IoT Jobs Data Plane	i AWS_ENDPOINT_URL_IOT_JOBS_DATA_PLANE d e
IoT 1Click Devices Service	i AWS_ENDPOINT_URL_IOT_1CLICK_DEVICES_ k_SERVICE_

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

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

serviceId	共用 AWS_ENDPOINT_URL_<SERVICE> 環境變數 方案的服務識別碼金鑰
Lambda	1: AWS_ENDPOINT_URL_LAMBDA
Launch Wizard	1: AWS_ENDPOINT_URL_LAUNCH_WIZARD 2:
Lex Model Building Service	1: AWS_ENDPOINT_URL_LEX_MODEL_BUILDING_SERVICE 2:
Lex Runtime Service	1: AWS_ENDPOINT_URL_LEX_RUNTIME_SERVICE me
Lex Models V2	1: AWS_ENDPOINT_URL_LEX_MODELS_V2 S_
Lex Runtime V2	1: AWS_ENDPOINT_URL_LEX_RUNTIME_V2 me
License Manager	1: AWS_ENDPOINT_URL_LICENSE_MANAGER at

serviceId	共用 AWS_ENDPOINT_URL_<SERVICE> 環境變數 AWS 方案的服務識別碼金鑰
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 AWS_ENDPOINT_URL_<SERVICE> 環境變數 AI 方案的服務識別碼金鑰
LookoutMetrics	lookoutmetrics: AWS_ENDPOINT_URL_LOOKOUTMETRICS
LookoutVision	lookoutvision: AWS_ENDPOINT_URL_LOOKOUTVISION
m2	m2: AWS_ENDPOINT_URL_M2
Machine Learning	machinelearning: 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	marketplaceagreement: AWS_ENDPOINT_URL_MARKETPLACE_AGREEMENT

serviceId	共用 AWS_ENDPOINT_URL_<SERVICE> 環境變數 AWS 方案的服務識別碼金鑰
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: i:
MediaConnect	m: AWS_ENDPOINT_URL_MEDIACONNECT e:
MediaConvert	m: AWS_ENDPOINT_URL_MEDIACONVERT e:

serviceId	共用 AWS_ENDPOINT_URL_<SERVICE> 環境變數 案例的服務識別碼金鑰
MediaLive	media AWS_ENDPOINT_URL_MEDIALIVE
MediaPackage	media AWS_ENDPOINT_URL_MEDIAPACKAGE
MediaPackage Vod	media AWS_ENDPOINT_URL_MEDIAPACKAGE_VOD
MediaPackageV2	media AWS_ENDPOINT_URL_MEDIAPACKAGEV2
MediaStore	media AWS_ENDPOINT_URL_MEDIASTORE
MediaStore Data	media AWS_ENDPOINT_URL_MEDIASTORE_DATA
MediaTailor	media AWS_ENDPOINT_URL_MEDIATAILOR
Medical Imaging	media AWS_ENDPOINT_URL_MEDICAL_IMAGING

serviceId	共用 AWS_ENDPOINT_URL_<SERVICE> 環境變數 案例的服務識別碼金鑰
MemoryDB	memorydb: AWS_ENDPOINT_URL_MEMORYDB
Marketplace Metering	marketplacemetering: AWS_ENDPOINT_URL_MARKETPLACE_METERING
Migration Hub	mghub: AWS_ENDPOINT_URL_MIGRATION_HUB
mgn	mgn: AWS_ENDPOINT_URL_MGN
Migration Hub Refactor Spaces	mghubrefactorspaces: AWS_ENDPOINT_URL_MIGRATION_HUB_REFAC TOR_SPACES
MigrationHub Config	mghubconfig: AWS_ENDPOINT_URL_MIGRATIONHUB_CONFIG
MigrationHubOrchestrator	mghuborchestrator: AWS_ENDPOINT_URL_MIGRATIONHUBORCHESTRATOR

serviceId	共用 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 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	o: 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 AWS Cloud 方案的 服務的 識別碼 金鑰 AWS_ENDPOINT_URL_<SERVICE> 環境變數
Outposts	outposts AWS_ENDPOINT_URL_OUTPOSTS
p8data	p8data AWS_ENDPOINT_URL_P8DATA
p8data	p8data AWS_ENDPOINT_URL_P8DATA
Panorama	panorama AWS_ENDPOINT_URL_PANORAMA
Payment Cryptography	payment-cryptography AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY
Payment Cryptography Data	payment-cryptography-data AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY_DATA
Pca Connector Ad	pca-connector-ad AWS_ENDPOINT_URL_PCA_CONNECTOR_AD
Personalize	personalize AWS_ENDPOINT_URL_PERSONALIZE

serviceId	共用 AWS_ENDPOINT_URL_<SERVICE> 環境變數 API 方案的服務識別碼金鑰
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 AWS_ENDPOINT_URL_<SERVICE> 環境變數 API 方案的服務識別碼金鑰
Polly	p: AWS_ENDPOINT_URL_POLLY
Pricing	p: AWS_ENDPOINT_URL_PRICING
PrivateNetworks	p: AWS_ENDPOINT_URL_PRIVATENETWORKS t:
Proton	p: AWS_ENDPOINT_URL_PROTON
QBusiness	q: AWS_ENDPOINT_URL_QBUSINESS
QConnect	q: AWS_ENDPOINT_URL_QCONNECT
QLDB	q: AWS_ENDPOINT_URL_QLDB
QLDB Session	q: AWS_ENDPOINT_URL_QLDB_SESSION i:
QuickSight	q: AWS_ENDPOINT_URL_QUICKSIGHT t:
RAM	r: AWS_ENDPOINT_URL_RAM
rbin	r: AWS_ENDPOINT_URL_RBIN

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

serviceId	共用 AWS_ENDPOINT_URL_<SERVICE> 環境變數 案例的服務識別碼金鑰
Route53 Recovery Control Config	r: AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CONTROL_CONFIG
Route53 Recovery Readiness	r: AWS_ENDPOINT_URL_ROUTE53_RECOVERY_READINESS
Route 53 Domains	r: AWS_ENDPOINT_URL_ROUTE_53_DOMAINS
Route53Resolver	r: AWS_ENDPOINT_URL_ROUTE53RESOLVER
RUM	r: 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 AWS CLI 案 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 ir
Sagemaker Edge	s: AWS_ENDPOINT_URL_SAGEMAKER_EDGE _e
SageMaker FeatureStore Runtime	s: AWS_ENDPOINT_URL_SAGEMAKER_FEATURESTORE_RUNTIME _f t i r
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 AWS_ENDPOINT_URL_<SERVICE> 環境變數 案例的服務識別碼金鑰
savingsplans	s: AWS_ENDPOINT_URL_SAVINGSPLANS a:
Scheduler	s: AWS_ENDPOINT_URL_SCHEDULER
schemas	s: AWS_ENDPOINT_URL_SCHEMAS
SimpleDB	s: AWS_ENDPOINT_URL_SIMPLEDB
Secrets Manager	s: AWS_ENDPOINT_URL_SECRETS_MANAGER a:
SecurityHub	s: AWS_ENDPOINT_URL_SECURITYHUB u:
SecurityLake	s: AWS_ENDPOINT_URL_SECURITYLAKE a:
ServerlessApplicationRepository	s: AWS_ENDPOINT_URL_SERVERLESSAPPLICATI s: ONREPOSITORY i: t:

serviceId	共用 AWS_ENDPOINT_URL_<SERVICE> 環境變數 AWS 方案的服務識別碼金鑰
Service Quotas	s: AWS_ENDPOINT_URL_SERVICE_QUOTAS u:
Service Catalog	s: AWS_ENDPOINT_URL_SERVICE_CATALOG a:
Service Catalog AppRegistry	s: AWS_ENDPOINT_URL_SERVICE_CATALOG_APP a: REGISTRY p:
ServiceDiscovery	s: AWS_ENDPOINT_URL_SERVICEDISCOVERY s:
SES	s: AWS_ENDPOINT_URL_SES
SESV2	s: AWS_ENDPOINT_URL_SESV2
Shield	s: AWS_ENDPOINT_URL_SHIELD
signer	s: AWS_ENDPOINT_URL_SIGNER
SimSpaceWeaver	s: AWS_ENDPOINT_URL_SIMSPACEWEAVER e:

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

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

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

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

智慧型組態預設

借助智能配置默認功能，AWS SDK 可以為其他配置設置提供預定義的優化默認值。

使用下列項目設定此功能：

defaults_mode-共享 AWS **config**文件設置, **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 設定有明確設定的值，則該值永遠優先。如果 AWS SDK 設定沒有明確設定值，且 `defaults_mode` 不等於舊版，則此功能可為針對應用程式最佳化的各種設定提供不同的預設值。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。如果不等於 `legacy`，我們建議您 `defaults_mode` 在升級 SDK 時對應用程式執行測試，因為提供的預設值可能會隨著最佳實務的發展而改變。

預設值：`legacy`

注意：SDK 的新主要版本預設為 `standard`。

有效值：

- `legacy`— 提供依 SDK 而異且在建立之前已存在的預設設定 `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`— 以標準模式為基礎建置，並包含針對 AWS 服務 從內部呼叫的應用程式量身打造的最佳化 AWS 區域。
- `cross-region`— 以標準模式為基礎，包括針對不同區域呼叫的應用程 AWS 服務 式量身打造的最佳化。
- `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 會嘗試探索執行階段環境，以自動決定適當的設定。`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 [執行個體中繼資料和使用者資料](#)，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 SDK 嘗試重試的方式。請參閱 [重試行為](#)。
- `stsRegionalEndpoints`— 指定 SDK 如何判斷它用來與 AWS Security Token Service (AWS STS) 通訊的 AWS 服務 端點。請參閱 [AWS STS 區域化端點](#)。

- `s3UsEast1RegionalEndpoints`— 指定 SDK 如何判斷它用來與 `us-east-1` 區域的 Amazon S3 通訊的 AWS 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 交換從 CLIENT HELLO 訊息傳送到用戶端與伺服器完全交涉密碼並交換金鑰的時間之後，TLS 交換所需的時間上限。

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。

與 AWS 開發套件的相容性

下列 SDK 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。和唯一支援任何 JVM 系統屬 AWS SDK for Java 性設 適用於 Kotlin 的 AWS SDK 定。

SDK	支援	備註或更多資訊
AWS CLI v2	否	
SDK for C++	是	未最佳化的參數： <code>stsRegionalEndpoints s3UsEast1RegionaleEndpoints</code> 、 <code>tlsNegotiationTimeoutInMillis</code> 。
適用於轉到 V2 的開發套件 (1.x)	是	未最佳化的參數： <code>retryMode stsRegionalEndpoints</code> 、 <code>s3UsEast1RegionalEndpoints</code> 。
適用於 Go 1.x 的開發套件 (V1)	否	
適用於 Java 2.x 的開發套件	是	未最佳化的參數： <code>stsRegionalEndpoints</code> 。
適用於 Java 1.x 的開發套件	否	
適用於 JavaScript 3.x 的開發套	是	未最佳化的參數： <code>stsRegionalEndpoints s3UsEast1RegionaleEndpoints</code> 、 <code>tlsNegotiationTimeoutInMillis</code> 。 <code>connectTimeoutInMillis</code> 被稱為 <code>connectionTimeout</code> 。
適用於 JavaScript 2.x 的 SDK	否	

SDK	支援	備註或更多資訊
SDK 的科特林	否	
適用於 .NET 3.x 的開發套	是	未最佳化的參數：connectTimeoutInMilliseconds、tlsNegotiationTimeoutInMilliseconds。
適用於 PHP 3.x 的開發套件	是	未最佳化的參數：tlsNegotiationTimeoutInMilliseconds。
SDK for Python (Boto3)	是	未最佳化的參數：tlsNegotiationTimeoutInMilliseconds。
適用於紅寶石 3.x 的 SDK	是	
適用於鏽蝕的 SDK	否	
適用的工具 PowerShell	是	未最佳化的參數：connectTimeoutInMilliseconds、tlsNegotiationTimeoutInMilliseconds。

AWS通用執行階段 (CRT) 程式庫

AWS通用運行時 (CRT) 庫是 SDK 的基本庫。CRT 是獨立封裝的模組化系列，以 C 撰寫，每個封裝都提供良好的效能和最小的佔用空間，適用於不同的所需功能。這些功能通用於所有 SDK，可提供更好的代碼重複使用，優化和準確性。這些軟件包是：

- [awslabs/aws-c-auth](#)：AWS用戶端驗證 (標準認證提供者和簽署 (sigv4))
- [awslabs/aws-c-cal](#)：加密原始類型，哈希 (MD5, SHA256, SHA256 HMAC)，簽名者，AES
- [awslabs/aws-c-common](#)：基本數據結構，線程/同步原始類型，緩衝區管理，stdlib 相關函數
- [awslabs/aws-c-compression](#)：壓縮算法 (赫夫曼編碼/解碼)
- [awslabs/aws-c-event-stream](#)：事件串流訊息處理 (標頭、序曲、有效負載、crc/拖車)、事件串流上的遠端程序呼叫 (RPC) 實作
- [awslabs/aws-c-http](#): C99 實作的 HTTP/1.1 和 HTTP/2 規格
- [awslabs/aws-c-io](#): 通訊端 (TCP、UDP)、DNS、管道、事件迴圈、通道、SSL/TLS
- [awslabs/aws-c-iot](#): C99 實作 AWS IoT 雲端服務與裝置整合
- [awslabs/aws-c-mqtt](#)：適用於物聯網 (IoT) 的標準輕量型訊息通訊協定
- [awslabs/aws-c-s3](#)：用於與 Amazon S3 服務通訊的 C99 程式庫實作，專為在高頻寬 Amazon EC2 執行個體上最大化輸送量而設計
- [awslabs/aws-c-sdkutils](#): 剖析和管理AWS設定檔的公用程式庫
- [awslabs/aws-checksums](#)：跨平台的硬體加速 CRC32c 和 CRC32，具備有效率軟體實作的後援功能
- [awslabs/aws-lc](#)：由密碼AWS學團隊AWS及其客戶維護的一般用途密碼編譯程式庫，根據谷歌 BoringSSL 專案和 OpenSSL 專案的程式碼
- [awslabs/s2n](#)：C99 實作 TLS/SSL 通訊協定，其設計既小巧又快速，並以安全為優先考量

CRT 可透過除 Go 以外的所有開發套件使用。

CRT 相依性

CRT 庫形成了一個複雜的關係和依賴關係網絡。如果您需要直接從源代碼構建 CRT，那麼了解這些關係很有幫助。但是，大多數使用者都可以透過其語言 SDK (例如 AWS SDK for C++ 或 AWS SDK for Java) 或其語言 IoT 裝置 SDK (例如用於 C++ 的 AWS IoT SDK 或 Java 的 AWS IoT SDK) 存取 CRT

功能。在下圖中，[語言 CRT 繫結] 方塊是指包裝特定語言 SDK CRT 程式庫的套件。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 程式庫的階層式相依性圖解。

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[版本支援矩陣](#)。

AWS SDK 和工具版本支援對照表

下表顯示可用 AWS 軟體開發套件 (SDK) 主要版本的清單，以及它們在維護生命週期中的位置及相關時間表。如需 AWS SDK 和工具主要版本的生命週期及其基礎相依性的詳細資訊，請參閱[維護政策](#)。

SDK	主要版本	目前階段	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 for C++	1.x	一般可用性	2015 年 9 月 2 日	
適用於轉到 V2 的 SDK	V2	一般可用性	1/19/2021	
SDK for Go	1.x	維護公告	11/19/2015	詳情及日期請參閱 公告
適用於 Java 的開發套件	1.x	維護公告	3/25/2010	詳情及日期請參閱 公告
適用於 Java 的開發套件	2.x	一般可用性	11/20/2018	
適用於的 SDK JavaScript	1.x	終止 Support	5/6/2013	
適用於的 SDK JavaScript	2.x	維護公告	6/19/2014	詳情及日期請參閱 公告
適用於的 SDK JavaScript	3.x	一般可用性	12/15/2020	
SDK 的科特林	1.x	一般可用性	11/27/2023	
SDK for .NET	1.x	終止 Support	11/2009	

SDK	主要版本	目前階段	一般可用日期	備註
SDK for .NET	2.x	終止 Support	11/8/2013	
SDK for .NET	3.x	一般可用性	7/28/2015	
SDK for PHP	2.x	終止 Support	11/2/2012	
SDK for PHP	3.x	一般可用性	5/27/2015	
Python 的 SDK (肉毒桿菌 2)	1.x	終止 Support	7/13/2011	
SDK for Python (Boto3)	1.x	一般可用性	6/22/2015	
Python (肉核) 的 SDK	1.x	一般可用性	6/22/2015	
SDK for Ruby	1.x	終止 Support	7/14/2011	
SDK for Ruby	2.x	終止 Support	2015 年 2 月 15 日	
SDK for Ruby	3.x	一般可用性	8/29/2017	
適用於鏽蝕的 SDK	1.x	一般可用性	11/27/2023	
SDK 為斯威夫特	1.x	開發人員預覽版		
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	

AWS SDK 和工具參考指南的文件歷史記錄

下表說明 AWS SDK 和工具參考指南的重要新增項目和更新。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
適用於 Java 1.x 系統屬性的開發套件	通過 AWS SDK for Java 1.x 添加有關支持的 JVM 系統配置設置的詳細信息。	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 支援的相容性更新、IAM 身分中心程序的更新。	2024年2月20日
容器認證更新。國際移動部門更新。	添加對 Amazon EKS 的支持。新增設定以停用 IMDSv1 後援。	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	SDK 和工具功能的相容性表已更新，包括 Kotlin 的 SDK、適用於 Rust 的 SDK 和 AWS Tools for PowerShell	2023 年 12 月 10 日
驗證更新	SDK 和工具支援的驗證方法的更新。	2023 年 7 月 1 日
IAM 最佳實務更新	更新了指南以符合 IAM 最佳實務。如需更多詳細資訊，請參閱 IAM 中的安全最佳實務 。	2023 年 2 月 27 日
SSO 更新	新 SSO 憑證組態的 SSO 認證的更新。	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 用戶端和 IMDS 憑證的清晰度。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	反映 AWS SSO 現在稱為的更新 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。