



使用者指南

# AWS Toolkit for VS Code



# AWS Toolkit for VS Code: 使用者指南

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

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

# Table of Contents

AWS Toolkit for Visual Studio Code .....	1
什麼是 AWS Toolkit for Visual Studio Code .....	1
相關資訊 .....	1
Amazon Q 開發者和 Amazon CodeWhisperer .....	2
下載工具包 .....	3
從 VS 代碼 Marketplace 下載工具包 .....	3
來自的其他 IDE 工具包 AWS .....	3
入門 .....	4
安裝 Toolkit for VS Code .....	4
必要條件 .....	4
下載並安裝 AWS Toolkit for Visual Studio Code .....	4
選擇性必要 .....	5
連接到 AWS .....	6
必要條件 .....	6
開啟「登入」面板 .....	6
AWS 從工具組連接到 .....	7
Amazon 身份驗證 CodeCatalyst .....	8
變更 AWS 區域 .....	8
將區域新增至 AWS 總管 .....	8
從 AWS 總管隱藏區域 .....	9
設定您的工具包裝您 .....	9
設定 .NET 核心的工具鏈 .....	9
設定 Node.js 的工具包括一個工具箱 .....	9
為 Python 配置一個工具鏈 .....	10
設定 Java 的工具鏈 .....	10
為 Go 配置工具鏈 .....	11
使用您的工具鏈 .....	11
身分驗證與存取 .....	12
IAM Identity Center .....	12
IAM 認證 .....	12
建立使 IAM 用者 .....	13
從建立共用認證檔案 AWS Toolkit for Visual Studio Code .....	13
新增其他認證設定檔 .....	14
AWS 產生器 ID .....	15

使用外部認證程序 .....	15
使用 AWS .....	16
實驗功能 .....	17
AWS 瀏覽器 .....	17
Amazon CodeCatalyst .....	18
什麼是 Amazon CodeCatalyst ? .....	18
開始使用 CodeCatalyst .....	19
使用 CodeCatalyst 資源 .....	19
使用開發環境 .....	22
故障診斷 .....	25
Amazon API Gateway .....	26
AWS App Runner .....	26
必要條件 .....	26
定價 .....	29
建立 App Runner 服務 .....	30
管理 App Runner 服務 .....	32
AWS應用程式編 .....	34
使 AWS 用應用程式撰寫器 .....	35
AWS CDK .....	36
AWS CDK 應用 .....	36
AWS CloudFormation 堆疊 .....	38
刪除AWS CloudFormation堆 .....	38
建立 CloudFormation 範本 .....	39
亞馬遜CloudWatch日誌 .....	40
檢視 CloudWatch 日誌群組和日誌串流 .....	41
使用 CloudWatch 日誌事件 .....	42
搜尋記錄群組 .....	43
Amazon ECR .....	45
先決條件 .....	46
搭配使用 Amazon ECR 和 Toolkit for VS Code .....	47
Amazon ECS .....	55
對任務定義文件使用 IntelliSense .....	55
Amazon ECS Exec .....	56
Amazon EventBridge .....	59
使用 Amazon EventBridge 模式 .....	59
AWS IAM Access Analyzer .....	61

AWS IAM使用存取分析器 .....	62
AWS IoT .....	65
AWS IoT 先決條件 .....	65
AWS IoT實物 .....	66
AWS IoT 憑證 .....	67
AWS IoT 政策 : .....	70
AWS Lambda 函數 .....	72
與遠端 Lambda 函數互動 .....	73
Amazon Redshift .....	78
與 Amazon Redshift 工作 .....	79
Amazon S3 .....	83
使用 S3 資源 .....	83
使用 S3 物件 .....	84
AWS 無伺服器應用程式 .....	88
開始 .....	88
執行和偵錯 Lambda 函數，同時排除 AWS SAM 範本資源 .....	95
執行並除錯本機 Amazon API Gateway 資源 .....	98
除錯無伺服器應用程式的組態選項 .....	101
疑難排解 .....	107
AWS Systems Manager .....	109
假設和先決條件 .....	109
Systems Manager 自動化文件的 IAM 許可 .....	110
建立新的 Systems Manager 自動化文件 .....	110
開啟現有的 Systems Manager 自動化文件 .....	111
編輯 Systems Manager 自動化文件 .....	111
發佈 Systems Manager 自動化文件 .....	112
刪除 Systems Manager 自動化文件 .....	113
執行 Systems Manager 自動化文件 .....	113
故障診斷 .....	113
AWS Step Functions .....	114
AWS Step Functions 和 VS 代碼 .....	114
威脅撰寫器 .....	123
使用威脅撰寫器 .....	124
資源 .....	124
IAM存取資源的權限 .....	125
新增並與現有資源互動 .....	126

---

建立和編輯資源 .....	127
安全性 .....	129
資料保護 .....	129
文件歷史紀錄 .....	130
.....	CXXXiv

# AWS Toolkit for Visual Studio Code

這是 AWS Toolkit for VS Code 的用戶指南。如果您要尋找 AWS Toolkit for Visual Studio，請參閱 [AWS Toolkit for Visual Studio 使用者指南](#)。

## 什麼是 AWS Toolkit for Visual Studio Code

VS 程式碼的工具組是適用於 Visual Studio 程式碼 (VS 程式碼) 編輯器的開放原始碼擴充功能。此擴充功能可讓開發人員更輕鬆地在本機開發、偵錯，以及部署使用 Amazon Web Services (AWS) 的無伺服器應用程式。

### 主題

- [AWS Toolkit for Visual Studio Code 入門](#)
- [使用 AWS 服務和工具](#)

## 相關資訊

使用下列資源存取工具組的原始程式碼，或查看目前未解決的問題。

- [原始碼](#)
- [問題追蹤器](#)

若要進一步了解視覺工作室程式碼編輯器，請造訪 <https://code.visualstudio.com/>。

# Amazon Q 開發者和 Amazon CodeWhisperer

從 2024 年 4 月 30 日起，Amazon 現在已成 CodeWhisperer 為 Amazon Q 開發人員的一員，其中包括內嵌代碼建議和 Amazon Q 開發人員安全掃描。[從 VS 程式碼 Marketplace 下載 Amazon Q 開發人員 IDE 擴充功能](#)以開始使用。

如需 Amazon Q 開發人員服務的詳細資訊，請參閱 [Amazon Q 開發人員](#) 使用者指南。如需 Amazon Q 計劃和定價的詳細資訊，請參閱 [Amazon Q 定價](#) 指南。

# 下載 Toolkit for VS Code

您可以AWS Toolkit for Visual Studio Code通過 IDE 中的 VS 代碼 Marketplace 下載，安裝和設置。如需詳細指示，請參閱本使用指南的入門主題中的「[下載並安裝](#)」一節。

## 從 VS 代碼 Marketplace 下載工具包

或者，您也可以從網頁瀏覽器導覽至 [VS Code Marketplace](#)，以下載AWS Toolkit for Visual Studio Code安裝檔案。

## 來自的其他 IDE 工具包 AWS

除了AWS Toolkit for Visual Studio Code, AWS 還提供 IDE 工具包 JetBrains 和視覺工作室。

### AWS Toolkit for JetBrains連結

- 請按照此鏈接AWS Toolkit for JetBrains從 JetBrains Marketplace [下載](#)。
- 若要進一步瞭解AWS Toolkit for JetBrains，請參閱[AWS Toolkit for JetBrains](#)使用者指南。

### Toolkit for Visual Studio 連結的工具組

- 請按照此鏈接從 [Toolkit for Visual Studio Marketplace](#) [下載](#)工具包。
- 若要深入了解適用 Toolkit for Visual Studio 組，請參閱 [Toolkit for Visual Studio](#) 使用者指南。

# AWS Toolkit for Visual Studio Code 入門

AWS Toolkit for Visual Studio Code使您的AWS服務和資源可用，直接從您的 VS 代碼集成開發環境 ( IDE )。

若要開始使用，下列主題說明如何設定、安裝和設定AWS Toolkit for Visual Studio Code。

## 主題

- [安裝 AWS Toolkit for Visual Studio Code](#)
- [連接到 AWS](#)
- [變更 AWS 區域](#)
- [設定您的工具包裝您](#)

## 安裝 AWS Toolkit for Visual Studio Code

### 必要條件

若要開始使用 AWS Toolkit for Visual Studio Code VS Code，必須符合下列先決條件。若要進一步了解如何存取所有可用的 AWS 服務和資源 AWS Toolkit for Visual Studio Code，請參閱本指南—[the section called “選擇性必要”節](#)。

- VS 程式碼需要視窗、macOS 或 Linux 作業系統。
- 要 AWS Toolkit for Visual Studio Code 求您使用 VS 代碼版本 1.73.0 或更高版本。

如需有關 VS Code 的其他資訊，或是要下載最新版本的 VS Code，請參閱 [VS 程式碼下載](#) 網站。

### 下載並安裝 AWS Toolkit for Visual Studio Code

您可以下載，安裝和設置 AWS Toolkit for Visual Studio Code 通過 VS 代碼 Marketplace 在您的IDE。或者，您也可以從網頁瀏覽器導覽至 [VS Code Marketplace](#)，以下載 AWS Toolkit for Visual Studio Code 安裝檔案。

AWS Toolkit for Visual Studio Code 從 VS 代碼 IDE Marketplace 安裝

1. IDE使用以下鏈接在 VS 代碼中打開 AWS Toolkit for Visual Studio Code 擴展程序：[打開 VS 代碼 Marketplace](#)。

**Note**

如果 VS Code 尚未在您的計算機上運行，則在加載 VS Code 時，此操作可能需要一些時間。

2. 從 VS 程式碼 Marketplace 的 AWS Toolkit for Visual Studio Code 擴充功能中，選擇 [安裝] 以開始安裝程序。
3. 出現提示時，選擇重新啟動 VS Code 以完成安裝過程。

## 選擇性必要

您必須具備下列條件 AWS Toolkit for Visual Studio Code，才能使用的某些功能：

- Amazon Web Services (AWS) 帳戶不是使用的必要條件 AWS Toolkit for Visual Studio Code，但如果沒有它，功能會受到很大限制。AWS 要獲取 AWS 帳戶，請轉到[AWS 首頁](#)。選擇「創建 AWS 帳戶」或「完成註冊」（如果您之前訪問過該網站）。
- 程式碼開發 — 與 SDK 您要使用的語言相關。您可以從以下連結下載，或使用您最愛的套件管理工具：
  - .NET SDK: <https://dotnet.microsoft.com/download>
  - Node.js SDK: <https://nodejs.org/en/download>
  - Python SDK : <https://www.python.org/downloads>
  - 爪 SDK 哇 <https://aws.amazon.com/corretto/>
  - 前往 SDK : <https://golang.org/doc/install>
- AWS SAM CLI— 此 AWS CLI 工具可協助您在本機開發、測試和分析無伺服器應用程式。安裝工具組不需要此操作。但是，我們建議您安裝它（以及接下來所述的 Docker），因為它對於任何 AWS Serverless Application Model (AWS SAM) 功能都是必需的，例如[建立新的無伺服器應用程式 \(本機\)](#)。

如需詳細資訊，請參閱[AWS Serverless Application Model 開發人員指南 AWS SAM CLI](#)中的安裝。

- Docker — AWS SAM CLI 需要此開放原始碼軟體容器平台。如需更多詳細資訊和下載指示，請參閱[Docker](#)。
- P@@ ackage 管理器-軟件包管理器，以便您可以下載和共享應用程序代碼。
  - .NET: [NuGet](#)
  - Node.js: [故宮](#)

- Python : [點子](#)
- 爪哇 : [搖籃或釋界](#)

## 連接到 AWS

大多數 Amazon Web Services (AWS) 資源都是透過 AWS 帳戶管理的。不需要 AWS 帳戶即可使用 AWS Toolkit for Visual Studio Code，但是 Toolkit 功能在沒有連接的情況下受到限制。

如果您先前已透過其他 AWS 服務 (例如 AWS Command Line Interface) 設定 AWS 帳戶和驗證，則 AWS Toolkit for Visual Studio Code 會自動偵測您的認證。

### 必要條件

如果您是新帳戶 AWS 或尚未創建帳戶，則有 3 個主要步驟可以 AWS Toolkit for Visual Studio Code 與您的 AWS 帳戶連接：

1. 註冊 AWS 帳戶：您可以從註冊入口網站[AWS 註冊 AWS 帳戶](#)。如需設定新 AWS 帳戶的詳細資訊，請參閱《AWS 設定使用指南》中的[概觀](#)主題。
2. 設置身份驗證：有 3 種主要方法可以從您的 AWS 帳戶進行身份驗證 AWS Toolkit for Visual Studio Code。若要進一步了解這些方法，請參閱本使用者指南中的「[驗證與存取](#)」主題。
3. 透過「AWS 工具組」進行驗證：您可以透過完成本使用者指南下列各節中的程序，從「工具組」與您的 AWS 帳戶連線。

### 開啟「登入」面板

完成下列其中一個程序，以開啟「AWS 工具組登入」面板。

若要從 AWS 檔案總管開啟「AWS 工具組登入」面板：

1. 從 AWS Toolkit for Visual Studio Code，展開 EXPLORER。
2. 展開更多動作... 菜單通過選擇... 圖示。
3. 從更多動作... 選單中，選擇「Connect AWS至」以開啟「AWS 工具組登入」面板。

若要使用「VS 程式碼」指令列開啟「AWS 工具組登入」面板：

1. 按下 **Shift+Command+P** ( **Ctrl+Shift+P** 視窗 ) 開啟指令棧板。

2. **AWS: Add a New Connection** 進入搜索字段。
3. 選取 **AWS: Add a New Connection** 以開啟「AWS 工具組登入」面板。

## AWS 從工具組連接到

### 驗證並連接 SSO

若要驗證並 AWS 使用連線 AWS IAM Identity Center，請完成以下程序。

#### Note

使用 AWS 產生器 ID 或 IAM 身分識別中心進行驗證會在預設網頁瀏覽器中啟動 AWS 授權入口網站。每當您的憑證過期時，都必須重複執行此程序，以更新您的 AWS 帳戶與 AWS Toolkit for Visual Studio Code。

### 驗證 AWS IAM 身分識別中心並連線

1. 從「AWS 工具組登入」面板中，選擇「人力」標籤，然後選取「繼續」按鈕以繼續。
2. 從「使用 IAM 身分識別中心登入」面板中，輸入組織 URL 的「開始」。這 URL 是由您的管理員或服務台提供給您。
3. 從下拉式選單中選取您的 AWS 地區。這是代管您身分識別目錄的 AWS 區域。
4. 選擇「繼續」按鈕，並確認您要在預設網頁瀏覽器中開啟 AWS 授權要求網站。
5. 按照默認 Web 瀏覽器中的提示進行操作，授權過程完成時會收到通知，關閉瀏覽器並返回 VS Code 是安全的。

### 使用認證進行 IAM 證並連線

若要 AWS 使用身分證 IAM 明進行驗證並連線，請完成以下程序。

#### 使用認證進行 IAM 證並連線

1. 從「AWS 工具組登入」面板中，選擇「IAM 認證」，然後選取「繼續」按鈕以繼續。
2. 在提供的欄位中輸入您 AWS 帳戶 **Secret Key** 的 **Profile Name** **Access Key**、和，然後選擇「繼續」按鈕，將設定檔新增至您的設定檔，並將 Toolkit 與您的 AWS 帳戶連線。
3. 當驗證完成並建立連線時，Toolkit AWS Explorer 會更新以顯示您的 AWS 服務和資源。

## Amazon 身份驗證 CodeCatalyst

若要 CodeCatalyst 從工具組開始使用，請使用您的 AWS 產生器 ID 或 IAM 身分識別中心認證進行驗證並連線。

下列程序說明如何驗證 Toolkit 並將其與您的 AWS 帳戶連線。

### 使用 AWS 生成器 ID 進行身份驗證並連接

1. 從「AWS 工具組登入」面板中，選擇「人力」標籤，然後選取「繼續」按鈕以繼續。
2. 在「登入方式SSO」面板頂端，選擇「跳至登入」連結。
3. 按照默認 Web 瀏覽器中的提示進行操作，授權過程完成時會收到通知，關閉瀏覽器並返回 VS Code 是安全的。

### 驗證IAM身分識別中心並連線

1. 從「AWS 工具組登入」面板中，選擇「人力」標籤，然後選取「繼續」按鈕以繼續。
2. 從「使用IAM身分識別中心登入」面板中，輸入組織URL的「開始」。這URL是由您的公司的管理員或服務台提供給您。
3. 從下拉式選單中選取您的 AWS 地區。這是代管您身分識別目錄的 AWS 區域。
4. 選擇「繼續」按鈕，並確認您要在預設網頁瀏覽器中開啟AWS 授權要求網站。
5. 按照默認 Web 瀏覽器中的提示進行操作，授權過程完成時會收到通知，關閉瀏覽器並返回 VS Code 是安全的。

## 變更 AWS 區域

「AWS 區域」會指定管理資 AWS 源的位置。當您從「AWS 檔案總管」中自動顯示連線到您的 AWS 帳戶時 AWS Toolkit for Visual Studio Code，會偵測到您的預設 AWS 區域。

以下各節說明如何從AWS 總管中加入或隱藏「區域」。

### 將區域新增至 AWS 總管

完成下列程序，以將「區域」新增至「AWS 總管」。

1. 在 VS Code 中，展開主功能表上的 [檢視] 並選擇 [命令面板]，以開啟命令面板。或使用下列快速鍵：

- 視窗和 Linux — 新聞界 **Ctrl+Shift+P**。
  - macOS — 按 **Shift+Command+P**。
2. 從「指令選項板」中，搜尋 **AWS: Show or Hide Regions** 並選擇 AWS：「展示」或「隱藏區域」以顯示可用區域的清單。
  3. 從清單中，選取您要新增至 AWS 總管的 AWS 區域。
  4. 選擇「確定」按鈕以確認您的選擇並更新 AWS 檔案總管。

## 從 AWS 總管隱藏區域

若要在「AWS 總管」檢視中隱藏「區域」，請完成下列程序。

1. 從 AWS 檔案總管中，找到您要隱藏的 AWS 區域。
2. 開啟 (按一下滑鼠右鍵) 要隱藏之區域的關聯式功能表。
3. 選擇「顯示」或「隱藏區域」以開啟「AWS VS 程式碼」中的「顯示或隱藏區域」選項。
4. 取消選取您要在 AWS 檔案總管檢視中隱藏的區域。

## 設定您的工具包裝您

AWS Toolkit for Visual Studio Code 支援跨所有 AWS 服務的多種語言。以下章節說明如何設定您的工具鏈境內的工具鏈境內的工具鏈境

### 設定 .NET 核心的工具鏈

1. 確保您已 [安裝](#) AWS Toolkit for VS Code。
2. 安裝 [C# 擴充功能](#)。這個擴展名使 VS 代碼調試 .NET 核心應用程序的 VS 代碼
3. 開啟 AWS Serverless Application Model (AWS SAM) 應用程式，或 [建立應用程式](#)。
4. 開啟包含的資料夾 `template.yaml`。

### 設定 Node.js 的工具包括一個工具箱

1. 確保您已 [安裝](#) AWS Toolkit for VS Code。
2. 開啟 AWS SAM 應用程式，或 [建立應用程式](#)。
3. 開啟包含的資料夾 `template.yaml`。

**Note**

直接從原始程式碼偵錯 TypeScript Lambda 函數時 (啟動組態具有 "target": "code")，TypeScript 編譯器必須安裝在全域或專案中 package.json。

## 為 Python 配置一個工具鏈

1. 確保您已[安裝](#) TAWS oolkit for VS Code。
2. 安裝 [Python 擴展的可視化工作室代碼](#)。此擴充功能可讓 VS 程式碼偵錯 Python 應用程式。
3. 開啟AWS SAM應用程式，或[建立應用程式](#)。
4. 開啟包含的資料夾template.yaml。
5. 在應用程序的根目錄中打開一個終端，並virtualenv通過運行進行配置python -m venv ./venv。

**Note**

每個系統只需要設定virtualenv一次。

6. 請執virtualenv行下列其一個項目的啟用下列一個
  - Bash Shell Shell : `./venv/Scripts/activate`
  - PowerShell: `./venv/Scripts/Activate.ps1`

## 設定 Java 的工具鏈

1. 確保您已[安裝](#) TAWS oolkit for VS Code。
2. 安裝 [Java 擴展和 Java 11](#)。此擴展使 VS 代碼能夠識別 Java 函數。
3. 安裝 [Java 除錯程式擴充功能](#)。此擴充功能可讓 VS 程式碼偵錯 Java 應用程式。
4. 開啟AWS SAM應用程式，或[建立應用程式](#)。
5. 開啟包含的資料夾template.yaml。

## 為 Go 配置工具鏈

1. 確保您已[安裝](#) AWS Toolkit for VS Code。
2. 調試 Go Lambda 函數需要 1.14 或更高版本。
3. 安裝 [Go 擴展](#) 程序。

### Note

除錯 Go1.15 以上執行階段所需的版本為 0.25.0 或更高版本。

4. 使用[命令面板](#)安裝 Go 工具：
  - a. 從指令面板中選擇Go: Install/Update Tools。
  - b. 從核取方塊集中，選取dlv和gopl5。
5. 開啟AWS SAM應用程式，或[建立應用程式](#)。
6. 開啟包含的資料夾template.yaml。

## 使用您的工具鏈

設定好工具鏈之後，您就可以使用它來[執行或偵錯](#) AWS SAM應用程式。

# AWS Toolkit for Visual Studio Code的身分驗證與存取

您不需要驗證即可 AWS 開始使用 AWS Toolkit for Visual Studio Code. 但是，大多數 AWS 資源都是通過 AWS 帳戶管理的。若要存取所有 AWS Toolkit for Visual Studio Code 服務和功能，您需要使用 AWS IAM Identity Center AWS Builder ID 或 IAM 登入資料進行驗證。

下列主題包含有關每個認證類型的其他詳細資料。

如需如何使用現有認證連線到 AWS 中 AWS Toolkit for Visual Studio Code 的詳細資訊，請參閱本使用者指南中的 [〈連線至〉](#) AWS主題。

## 主題

- [AWS IAM 身分識別中心](#)
- [AWS IAM認證](#)
- [AWS 開發人員的生成器 ID](#)
- [使用外部認證程序](#)

## AWS IAM 身分識別中心

AWS IAM Identity Center 是管理 AWS 帳戶驗證的建議最佳作法。

如需有關如何設定軟體開發套件 (SDK) 的 IAM 身分中心的詳細指示，請參閱 AWS SDK 和工具參考指南的 [IAM 身分中心身分驗證](#) 一節。

如需如何驗證 AWS 工具組以及如何使用現有 IAM 身分中心登入資料 Connect 線的詳細資訊，請參閱本使用者指南中的 [「連線至」](#) AWS主題。

## AWS IAM認證

AWS IAM透過本機儲存的存取金鑰，使用您的 AWS 帳戶進行認證。

如需如何驗證 AWS 工具組並使用現有 AWS IAM認證 Connect 的詳細資訊，請參閱本使用者指南中的 [「連線至」](#) AWS主題。

以下各節說明如何設定IAM認證，以便從 AWS 帳戶進行驗證 AWS Toolkit for Visual Studio Code。

### ⚠ Important

在設定IAM憑證以使用您的 AWS 帳戶進行驗證之前，請注意：

- 如果您已經透過其他 AWS 服務 (例如 AWS CLI) 設定IAM認證，則 AWS Toolkit for Visual Studio Code 會自動偵測這些認證，並在 VS Code 中提供這些認證。
- AWS 建議使用IAM身分識別中心驗證。如需有關 AWS IAM最佳做法的其他資訊，請參閱 AWS Identity and Access Management 使用者指南—IAM節[中的安全性最佳作法](#)。
- 為避免安全風險，在開發專用軟件或使IAM用真實數據時，請勿使用用戶進行身份驗證。請改為使用身分識別提供者的聯合，例如[IAM身分識別中心是什麼？](#) 在《AWS IAM Identity Center 使用者指南》中。

## 建立使IAM用者

您必須先完成「AWS Toolkit for Visual Studio Code 和工具參考指南」中的「[使用長期憑證驗證](#)」主題中的「[步驟 1：建立使用IAM者](#)」和「[步驟 2：取得您的存取金鑰](#)」，才能設定為使用您的 [AWS SDKs帳戶進行驗證](#)。

### 📘 Note

步驟 3：更新共享憑據文件AWS SDKs和工具參考指南是可選的。

如果您完成步驟 3，會在以下位置期間 AWS Toolkit for Visual Studio Code 自動偵測您的身[the section called “從建立共用認證檔案 AWS Toolkit for Visual Studio Code”](#)分證明。

如果您尚未完成步驟 3，則會逐 AWS Toolkit for Visual Studio Code 步引導您完成建立程序，credentials file如下[the section called “從建立共用認證檔案 AWS Toolkit for Visual Studio Code”](#)所述。

## 從建立共用認證檔案 AWS Toolkit for Visual Studio Code

您的共享配置文件和共享憑據文件存儲配置和 AWS 帳戶的憑據信息。如需共用組態和憑證的詳細資訊，請參閱《AWS Command Line Interface 使用者指南》中的[組態設定儲存在哪裡？](#)一節。

透過建立共用認證檔案 AWS Toolkit for Visual Studio Code

1. 按下**Shift+Command+P** ( **Ctrl+Shift+P**視窗 ) 開啟指令棧板。
2. **AWS: Add a New Connection**進入搜索字段。

3. 選取**AWS: Add a New Connection**以開啟「AWS 工具組登入」面板。
4. 從「AWS 工具組登入」面板中，選擇「IAM認證」，然後選取「繼續」按鈕以繼續。
5. 在提供的欄位中輸入您 AWS 帳戶**Secret Key**的**Profile Name****Access Key**、和，然後選擇「繼續」按鈕，將設定檔新增至您的設定檔，並將 Toolkit 與您的 AWS 帳戶連線。
6. 當驗證完成並建立連線時，Toolkit AWS Explorer 會更新以顯示您的 AWS 服務和資源。

### Note

在此範例中，假設 `[Profile_Name]` 包含語法錯誤並導致驗證失敗。

```
[Profile_Name]
xaws_access_key_id= AKIAI44QH8DHBEXAMPLE
xaws_secret_access_key= wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
```

以下是為了回應失敗的驗證嘗試而產生的記錄訊息範例。

```
2022-11-02 22:01:54 [ERROR]: Profile [Profile_Name] is not a valid Credential
Profile: not supported by the Toolkit
2022-11-02 22:01:54 [WARN]: Shared Credentials Profile [Profile_Name] is not
valid. It will not be used by the toolkit.
```

## 新增其他認證設定檔

您可以將多個認證新增至您的組態檔。若要這麼做，請開啟命令面板，然後選擇「AWS 工具組」「建立認證設定檔」這將打開憑據文件。在此頁面上，您可以在第一個設定檔下方新增設定檔，如下列範例所示：

```
# Amazon Web Services Credentials File used by AWS CLI, SDKs, and tools
# This file was created by the AWS Toolkit for Visual Studio Code extension.
#
# Your AWS credentials are represented by access keys associated with IAM users.
# For information about how to create and manage AWS access keys for a user, see:
# https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html
#
# This credential file can store multiple access keys by placing each one in a
```

```
# named "profile". For information about how to change the access keys in a
# profile or to add a new profile with a different access key, see:
# https://docs.aws.amazon.com/cli/latest/userguide/cli-config-files.html
#
[Profile1_Name]
# The access key and secret key pair identify your account and grant access to AWS.
aws_access_key_id = AKIAIOSFODNN7EXAMPLE
# Treat your secret key like a password. Never share your secret key with anyone. Do
# not post it in online forums, or store it in a source control system. If your secret
# key is ever disclosed, immediately use IAM to delete the access key and secret key
# and create a new key pair. Then, update this file with the replacement key details.
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
[Profile2_Name]
aws_access_key_id = AKIAI44QH8DHBEXAMPLE
aws_secret_access_key = je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY
```

## AWS 開發人員的生成器 ID

AWS 產生器 ID 是特定 AWS 服務選用或必要的額外 AWS 帳戶。[有關 AWS Builder ID 驗證方法的詳細資訊，請參閱AWS 登入使用者指南中的使用 AWS Builder ID 登入主題。](#)

如需如何驗證 AWS 工具組並使用現有 AWS Builder ID Connect 的詳細資訊，請參閱本使用者指南中的「[連線至](#)」AWS主題。

## 使用外部認證程序

您可AWS Toolkit for Visual Studio Code以透過AWS修改您的shared config file.

修改您shared config file的憑證程序對於AWS Toolkit for Visual Studio Code和是相同的AWS Command Line Interface。如需有關如何設定外部認證的詳細資訊，請參閱《AWS Command Line Interface使用者指南》中的[具有外部處理程序的 Sourcing 認證](#)主題。

# 使用 AWS 服務和工具

直接在 VS Code 中為您提供 AWS 服務，工具和資源。AWS Toolkit for Visual Studio Code 以下是涵蓋 VS Code 服務的每個工具組及其功能的指南主題清單。選擇服務或工具，以取得有關其功能、如何設定以及使用基本功能的詳細資訊。

## 主題

- [使用實驗功能](#)
- [在 AWS 總管中使用 AWS 服務](#)
- [Amazon CodeCatalyst VS 代碼](#)
- [Amazon API Gateway](#)
- [AWS App Runner 搭配使用 AWS Toolkit for Visual Studio Code](#)
- [AWS 應用程式編](#)
- [AWS CDK 適用於 VS 程式碼](#)
- [使用 AWS CloudFormation 堆疊](#)
- [使用 CloudWatch 記錄檔使用 AWS Toolkit for Visual Studio Code](#)
- [使用 Amazon Elastic Registry](#)
- [使用 Amazon Elastic Container Service](#)
- [使用 Amazon EventBridge](#)
- [AWS IAM Access Analyzer](#)
- [在 AWS Toolkit for Visual Studio Code 中使用 AWS IoT](#)
- [使用 AWS Lambda 函數](#)
- [Toolkit for VS Code 中的 Amazon Redshift](#)
- [與 Amazon S3 合作](#)
- [使用無伺服器應用程式](#)
- [使用 Systems Manager 自動化文件](#)
- [使用 AWS Step Functions](#)
- [使用威脅撰寫器](#)
- [使用 資源](#)

## 使用實驗功能

實驗性功能可提早使用中的功能AWS Toolkit for Visual Studio Code在正式推出前。

### Warning

由於實驗性功能會持續進行測試和更新，因此它們可能存在可用性問題。並且實驗功能可能會從中刪除AWS Toolkit for Visual Studio Code恕不另行通知。

您可以啟用特定的實驗性功能AWS在中的服務AWS工具箱的 區段設定VS 代碼 IDE 中的窗格。

1. 編輯AWSVS 代碼中的設置，選擇File (檔案)、偏好設定、設定。
2. 在「中設定」窗格中，展開擴充並選擇AWS工具箱。
3. Under「AWS : Experiments」中，選取您要在發行前存取的實驗性功能的核取方塊。如果您想要關閉實驗性功能，請清除相關核取方塊。

## 在 AWS 總管中使用 AWS 服務

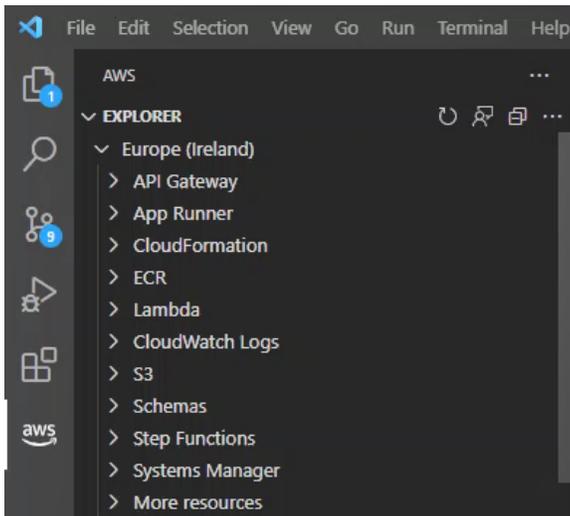
AWS 檔案總管可讓您檢視使用時可以使用的一些 AWS 服務 AWS Toolkit for Visual Studio Code。

本節提供有關如何在 VS Code 中存取和使用AWS 檔案總管的資訊。它假設您已經在系統上[安裝並配置](#)了 VS Code 的工具包。

部分重點提示：

- 如果已安裝並正確設定工具組，您應該會在檔案總管中看到項目。AWS 若要查看AWS 檔案總管，請選擇 [活動] 列中的AWS圖示。

例如：



- 某些功能需要特定 AWS 權限。例如，若要查看 AWS 帳戶中的 AWS Lambda 函數，您在其中設定的認證[身分驗證與存取](#)必須至少包含唯讀 Lambda 許可。如需每個功能所需許可的詳細資訊，請參閱以下主題。
- 如果您想要與在 AWS 檔案總管中無法立即顯示的 AWS 服務互動，您可以移至 [更多資源]，然後從可新增至介面的數百個資源中進行選擇。

例如，您可以從可用的資源類型中選擇 AWS ToolkitCodeArtifact::: 存放庫。將此資源類型新增至 [更多資源] 後，您可以展開項目以檢視使用自己的屬性和屬性建立不同 CodeArtifact 儲存庫的資源清單。此外，您可以在 JSON 格式的範本中描述資源的屬性和屬性，這些範本可以儲存以在雲端中建立新資源。AWS

## Amazon CodeCatalyst VS 代碼

### 什麼是 Amazon CodeCatalyst ？

Amazon CodeCatalyst 是適用於軟體開發團隊的雲端協作空間。透過 AWS Toolkit for Visual Studio Code，您可以直接從 VS Code 檢視和管理您的 CodeCatalyst 資源。您也可以啟動 CodeCatalyst 開發環境，直接在雲端中工作。如需有關 CodeCatalyst 服務的詳細資訊，請參閱 [Amazon CodeCatalyst 使用者指南](#)。

下列主題說明如何連接 VS Code CodeCatalyst，以及如何 CodeCatalyst 從 VS 程式碼的工具組中使用。

#### 主題

- [開始使用 CodeCatalyst Toolkit for VS Code](#)

- [在 VS 代碼中使用 Amazon CodeCatalyst 資源](#)
- [在開發環境中使用工具包](#)
- [Amazon CodeCatalyst 和 VS 代碼故障排除](#)

## 開始使用 CodeCatalyst Toolkit for VS Code

若要開始在 VS 程式碼 CodeCatalyst 中使用，請遵循下列程序。

### 主題

- [創建一個 CodeCatalyst 帳戶](#)
- [連接 AWS 工具包 CodeCatalyst](#)

## 創建一個 CodeCatalyst 帳戶

您必須具有作用中的 AWS 產生器 ID 或 AWS IAM Identity Center 認證，才能 CodeCatalyst 從 Toolkit for VS Code 連線到。若要進一步瞭解 AWS 產生器 ID、IAM 身分識別中心和 CodeCatalyst 認證，請參閱 CodeCatalyst 使用者指南中的 [〈設定方式〉](#) 一節。

## 連接 AWS 工具包 CodeCatalyst

若要將 AWS 工具組與您的 CodeCatalyst 帳戶連接，請參閱本使用者指南中的連線至 AWS 主題中的 [Amazon 身份驗證](#) — CodeCatalyst 節。

## 在 VS 代碼中使用 Amazon CodeCatalyst 資源

以下各節提供 Toolkit for VS Code 中可用之 Amazon CodeCatalyst 資源管理功能的概觀。

有關開發環境以及如何從中存取它們的詳細資訊 CodeCatalyst，請參閱 Amazon CodeCatalyst 使用者指南中的 [開發環境](#) 一節。

以下各節說明如何從 VS Code 建立、開啟和使用開發環境。

### 主題

- [複製儲存庫](#)
- [開啟開發環境](#)
- [建立開 CodeCatalyst 發環境](#)
- [從第三方儲存庫建立開發環境](#)

- [CodeCatalyst VS 代碼中的命令](#)

## 複製儲存庫

CodeCatalyst 是一種基於雲的服務，需要您連接到雲才能處理 CodeCatalyst 項目。如果您偏好在本機上處理專案，則可以將 CodeCatalyst 儲存庫複製到本機電腦，並在下次連線至雲端時將其與 CodeCatalyst 專案線上同步。

若要使用 AWS 工具組將存放庫從您的 CodeCatalyst 帳戶複製到 VS Code，請完成以下步驟：

### Note

如果您從第三方服務複製存放庫，系統可能會提示您使用該服務的認證進行驗證。複製儲存區域時，VS Code 會在「複製儲存區域」狀態視窗中顯示進度。複製儲存庫之後，您要開啟複製的儲存庫嗎？出現消息。

1. 從 Toolkit for VS Code 中，展開 DEVELOPERTOOLS 資源管理器。
2. 展開 CodeCatalyst，選擇複製儲存庫。
3. 在「選取 CodeCatalyst 儲存庫」對話方塊中，搜尋您要複製的存放庫，然後選取它以開啟「選擇要複製的資料夾」對話方塊。
4. 選擇「選取儲存區域位置」以關閉提示並開始複製儲存區域。
5. 從對話方塊視窗中，選擇下列其中一項來完成複製程序：
  - 若要在目前的「VS 程式碼」視窗中開啟儲存庫，請選擇「開啟」。
  - 若要在新的「VS 程式碼」視窗中開啟儲存庫，請選擇「在新視窗開啟」。
  - 若要在不開啟存放庫的情況下完成複製程序，請關閉對話方塊視窗。

## 開啟開發環境

若要在 VS Code 中開啟現有的開發環境，請完成下列步驟。

### Note

選擇開發環境開始 CodeCatalyst 通過打開您的開發環境來連接 VS Code 的過程。在此過程中，VS Code 會在 CodeCatalyst 狀態視窗中顯示進度更新。當程序完成時，狀態視窗會更新。

- 如果開發環境無法開啟，狀態會更新，並顯示程序失敗原因的資訊，以及開啟處理程序記錄的連結。
- 如果程序成功，您的開發環境會從 VS Code 在新視窗中開啟。

1. 從 Toolkit for VS Code 中，展開DEVELOPERTOOLS資源管理器。
2. 展開CodeCatalyst並選擇 [開啟開發環境]，以在 VS Code 中開啟 [選取開 CodeCatalyst發人員環境] 對話方塊。
3. 從 [選取開 CodeCatalyst 發環境] 對話方塊中，選擇您要開啟的開發環境。

## 建立開 CodeCatalyst 發環境

若要建立新的開發環境，請完成以下步驟：

### Note

建立新的開發環境時，請注意下列事項：

- AWS 建議您指定別名，因為它可簡化組織並改善開發環境的搜尋功能。
- 開發環境會持續保存您的工作。這意味著您的開發環境可以在不丟失工作的情況下停止。停止開發環境可降低開發環境保持正常運作所需的成本。
- 儲存是建立開發環境之後唯一無法變更的設定。
- VS Code 會在狀態視窗中顯示正在建立的開發環境的進度。建立開發環境之後，VS Code 會在新視窗中開啟開發環境，而您是否信任此資料夾中檔案的作者？提示也會出現。同意條款與條件，以繼續在您的開發環境中運作。

1. 從 Toolkit for VS Code 中，展開DEVELOPERTOOLS資源管理器。
2. 展開 CodeCatalyst，然後選擇 [建立開發環境] 選項以開啟 [VS Code] 中的 [建立開 CodeCatalyst 發環境] 功能表。
3. 從「原始程式碼」區段中，選擇下列其中一個選項：
  - 使用現有的 CodeCatalyst 存放庫：從現有存 CodeCatalyst 放庫建立開發環境。您必須選取「CodeCatalyst 專案」和「分支」。
  - 創建一個空的開發環境：創建一個空的開發環境。

4. (選擇性) 在「別名」區段中，輸入開發環境的替代名稱。
5. (可選) 從開發環境配置部分中，更改以下設置以滿足您的特定需求。
  - 運算：選擇「編輯運算」以變更指派給系統的處理能力量。RAM
  - 逾時：選擇「編輯逾時」以變更「開發環境」停止之前允許的系統閒置時間量。
  - 儲存空間：選擇編輯儲存空間大小以變更指派給系統的儲存空間量。
6. 選擇 [建立開發環境] 以建立新的雲端開發環境。

## 從第三方儲存庫建立開發環境

您可以透過連結至儲存庫做為來源，從第三方儲存庫建立開發環境。

連結至協力廠商儲存庫做為來源會在中的專案層級處理 CodeCatalyst。有關如何將第三方存儲庫連接到開發環境的說明和其他詳細資訊，請參閱 Amazon CodeCatalyst 使用者指南中的[連結來源儲存庫](#)主題。

## CodeCatalyst VS 代碼中的命令

還有其他 VS Code 指令會指派給 CodeCatalyst 相關功能，這些功能不會直接顯示在「AWS 工具組」中。

若要從指令選項板檢視指定給 CodeCatalyst 的指令清單，請完成以下步驟：

1. 從 Toolkit for VS Code 中，展開 DEVELOPERTOOLS 資源管理器。
2. 選擇「展示 CodeCatalyst 指令」以開啟具有預先填入搜尋的「指令選項板」。CodeCatalyst
3. 從清單中選取要啟用的 CodeCatalyst 指令。

## 在開發環境中使用工具包

開發環境是 Amazon 的虛擬運算環境 CodeCatalyst。下列各節說明如何使用 AWS Toolkit for Visual Studio Code。

如需有關開發環境的詳細資訊，請參閱 Amazon CodeCatalyst 使用者指南中的[開發環境](#)主題。

## 使用開發文件配置您的開發環境

該 devfile 規格是一種開放標準格式 YAML，可用於定義開發環境的配置。每個開發環境都有一個開發文件。如果您創建一個沒有存儲庫的開發環境或從不包含 devfile 的存儲庫，默認值將自動應用於源。

開發文件可以從 CodeCatalyst 或您的 IDE。在 VS Code 的本機或遠端執行個體中更新 devfile 的程序是相同的，但是如果您在本機更新 devfile，則必須在更新生效之前將更新推送至來源儲存庫。

如需使用 devfiles 設定開發環境的詳細資訊，請參閱 Amazon CodeCatalyst 使用者指南中的[設定開發環境](#)主題。

下列程序說明如何在開發環境中執行時，從 Toolkit 的遠端執行個體編輯 devfile。

### Important

如果您編輯來自 Devfile 的 VS 代碼，請注意以下幾點：

- 變更 devfile 的名稱或 devfile 元件名稱會取代根目錄的內容。所有先前的內容都會遺失且無法復原。
- 如果您在根資料夾或未與來源儲存庫關聯的開發環境中建立沒有 devfile 的開發環境，則當您建立開發環境時，會為您的開發環境產生具有預設組態設定的 devfile。
- 有關如何定義和配置的指示 Devfile，請參閱 [devfile.io](https://devfile.io) 網站上的[添加命令](#)文檔。

1. 從 Toolkit for VS Code 中，展開 DEVELOPERTOOLS 資源管理器。
2. 展開 CodeCatalyst 並選擇 [開啟 Devfile]，devfile.yaml 在目前的開發環境中的新編輯器視窗中開啟。
3. 在 VS 程式碼編輯器中，更新您的開發檔案，然後儲存變更。
4. 下次啟動開發環境時，組態會更新，以符合在 Devfile。

## 驗證並 AWS 從您的開發環境連接

若要從您的開發環境存取所有 AWS 資源，您必須驗證 Toolkit 的遠端執行個體，並將其與您的 AWS 帳戶連線。啟動開發環境時，Toolkit 的遠端執行個體會自動使用繼承自 Toolkit 本機執行個體的認證進行驗證。

更新 Toolkit 遠端執行個體認證的程序與 Toolkit 本機執行個體中的驗證體驗相同。如需如何 AWS 從工具組更新認證、驗證和連線的詳細指示，請參閱本使用者指南的入門主題中的〈[連線至](#)〉— AWS 節。

如需與相容之每種 AWS 驗證方法的詳細資訊 AWS Toolkit for Visual Studio Code，請參閱本使用者指南中的「[驗證與存取](#)」主題。

## 在開發環境中使用 Toolkit for VS Code

在 VS Code 中開啟或建立開發環境之後，您可以從 VS Code 的工具組中工作，類似於您可以從 VS Code 的本機執行個體執行個體執行個體執行個體。運行 VS 代碼的開發環境被配置為自動安裝 AWS 工具包並與您的 AWS 生成器 ID 連接。

### 停止開發環境

要停止當前的開發環境：

1. 從 Toolkit for VS Code 中，展開DEVELOPERTOOLS資源管理器。
2. 展開CodeCatalyst並選擇停止開發環境。
3. 當 VS Code 提示時，請確認您要停止開發環境。
4. 當 VS Code 關閉遠端連線並返回本機開發執行個體時，您的開發環境已成功停止。

### 開啟開發環境設定

若要開啟目前開發環境的設定，請完成以下步驟：

#### Note

一旦建立之後，就無法變更您開發環境的儲存體空間。

1. 從 Toolkit for VS Code 中，展開DEVELOPERTOOLS資源管理器。
2. 展開CodeCatalyst並選擇 [開啟設定]，以針對您目前的開發環境開啟 [開發環境設定] 檢視。
3. 從 Dev Environment Settings (開發環境設定) 檢視中，下列各章節包含適用於您開發環境的選項：
  - Alias: (別名：) 檢視和變更指派給開發環境的 Alias (別名)。
  - 狀態：檢視您目前的開發環境狀態、其指派給的專案，並停止您的環境。
  - Devfile：檢視您開發環境的Devfile名稱和位置。Devfile通過選擇在編輯器中打開按鈕打開您的。
  - Compute Settings: (運算設定：) 變更開發環境的 Timeout Length (逾時長度) 長短和預設值。

## Amazon CodeCatalyst 和 VS 代碼故障排除

下列主題解決了使用 Amazon CodeCatalyst 和 VS 程式碼時潛在的技術問題。

### 主題

- [VS 程式碼版本](#)
- [Amazon 的許可 CodeCatalyst](#)
- [從 Toolkit for VS Code 連接到開發環境](#)

### VS 程式碼版本

您的 VS Code 版本預計將在您的系統 `vscode://URIs` 上設置一個處理程序。如果沒有這個處理常式，您就無法從 AWS 工具組存取所有 CodeCatalyst 功能。例如，從 VS Code 測試人員啟動開發環境時，您會遇到錯誤。這是因為 VS Code 內部人員處理 `vscode-insiders://URIs` 並不處理 `vscode://URIs`。

### Amazon 的許可 CodeCatalyst

以下是 CodeCatalyst 從中使用的檔案權限需求 AWS Toolkit for Visual Studio Code：

- 將您自己的 `~/.ssh/config` 檔案存取權限設定為 `read` 和 `write`。 `write` 限制所有其他使用者的權限。
- 將 `~/.ssh/id_dsa` 和 `~/.ssh/id_rsa` 檔案的存取權限設定為「`read` 僅」。限制 `read` 所有其他使用者的 `execute` 權限。 `write`
- 您的 `globals.context.globalStorageUri.fsPath` 檔案必須位於可寫入的位置。

### 從 Toolkit for VS Code 連接到開發環境

如果您在嘗試從以下位置連線至開發環境時收到下列錯誤 AWS Toolkit for Visual Studio Code：

您 `~/.ssh/config` 的 `aws-devenv-*` 區段可能已過期。

- 選擇「開啟」設定。按鈕以在 VS 代碼編輯器中打開 `~/.ssh/config` 文件。
- 從編輯器中，選取並刪除 `Host aws-devenv-*` 區段的內容。
- 儲存您對中所做 `Host aws-devenv-*` 的變更 `~/.ssh/config`。然後，關閉檔案。
- 重新嘗試從 Toolkit for VS Code 連接到開發環境。

# Amazon API Gateway

您可以瀏覽和運行遠程 API Gateway 資源，並在已連接AWS帳戶使用AWS Toolkit for Visual Studio Code。

## Note

此功能不支援調試。

## 瀏覽和運行遠程 API Gateway 資源

1. 在 中AWS探險者，選擇API Gateway以展開菜單。列出了遠程 API Gateway 資源。
2. 開啟操作功能表 (按右鍵) 選單，開啟其內容 (按右鍵) 選單，然後選擇開啟操作功能AWS。
3. 在參數窗體中，指定調用參數。
4. 要運行遠程 API Gateway 資源，請選擇呼叫。其結果將在VS 代碼輸出視圖。

## AWS App Runner 搭配使用 AWS Toolkit for Visual Studio Code

[AWS App Runner](#)提供快速、簡單且符合成本效益的方式，將原始程式碼或容器映像直接部署到 AWS 雲端中可擴充且安全的 Web 應用程式。使用它，您不需要學習新技術、決定要使用哪個運算服務，或知道如何佈建和設定 AWS 資源。

您可以使用 AWS App Runner 根據來源映像檔或原始程式碼建立和管理服務。如果您使用來源映像，則可以選擇存放在映像儲存庫中的公有或私有容器映像。App Runner 支援下列映像儲存庫提供者：

- Amazon Elastic Container Registry (Amazon ECR)：在您的 AWS 帳戶中存儲私有映像。
- Amazon Elastic Container Registry Public (Amazon ECR Public)：存放公有的可讀取映像。

如果您選擇原始碼選項，則可以從由支援的儲存庫提供者所維護的原始碼儲存庫進行部署。目前，App Runner 支持[GitHub](#)作為源代碼存儲庫提供程序。

## 必要條件

要使用與應用程式運行器進行交互，AWS Toolkit for Visual Studio Code 需要以下內容：

- 一個 AWS 帳戶

- AWS Toolkit for Visual Studio Code 該功能的版本 AWS App Runner

除了這些核心需求之外，請確定所有相關的 IAM 使用者都具有與 App Runner 服務互動的許可。此外，您還需要獲取有關服務源的特定信息，例如容器映像 URI 或與 GitHub 存儲庫的連接。當您建立 App Runner 服務時需要此資訊。

### 設定 App Runner 的 IAM 許可

授予 App Runner 所需權限的最簡單方法是將現有的 AWS 受管政策附加到相關 AWS Identity and Access Management (IAM) 實體，特別是使用者或群組。App Runner 提供兩個您可以連接至 IAM 使用者的受管政策：

- `AWSAppRunnerFullAccess`：允許使用者執行所有 App Runner 動作。
- `AWSAppRunnerReadOnlyAccess`：允許使用者列出和檢視 App Runner 資源的詳細資訊。

此外，如果您從 Amazon Elastic Container Registry (Amazon ECR) 選擇私有儲存庫做為服務來源，則必須為 App Runner 服務建立下列存取角色：

- `AWSAppRunnerServicePolicyForECRAccess`：允許 App Runner 存取您帳戶中的 Amazon Elastic Container Registry (Amazon ECR) 映像。

使用 VS Code 的命令面板設定服務執行個體時，您可以自動建立此角色。

#### Note

`AWSServiceRoleForAppRunner` 服務連結角色可 AWS App Runner 讓您完成下列工作：

- 將日誌推送到 Amazon CloudWatch 日誌日誌群組。
- 創建 Amazon CloudWatch 活動規則以訂閱 Amazon Elastic Container Registry (Amazon ECR) 映像推送。

您不需要手動建立 服務連結角色。當您 AWS App Runner 在 AWS Management Console 或使用呼叫的 API 作業中建立時 AWS Toolkit for Visual Studio Code，會為您建 AWS App Runner 立此服務連結角色。

如需詳細資訊，請參閱《AWS App Runner 開發人員指南》中的「[App Runner 的 Identity and Access Management](#)」。

取得 App Runner 的服務來源

您可以使用 AWS App Runner 從來源映像檔或原始程式碼部署服務。

Source image

如果您是從來源映像檔進行部署，您可以從私人或公用映像登錄取得該映 AWS 像存放庫的連結。

- Amazon ECR 私有登錄：複製使用 Amazon ECR 主控台之私有儲存庫的 URI，網址為 <https://console.aws.amazon.com/ecr/repositories>。
- Amazon ECR 公有登錄：複製使用 Amazon ECR Public Gallery (Amazon ECR 公有映像庫) 的公有儲存庫的 URI，網址為 <https://gallery.ecr.aws/>。

#### Note

您也可以直接從 VS 程式碼工具組中的 AWS 資源管理器取得私有 Amazon ECR 儲存庫的 URI：

- 開啟 AWS 檔案總管並展開 ECR 節點，以檢視該 AWS 區域的存放庫清單。
- 在儲存庫上按一下滑鼠右鍵，然後選擇 Copy Repository URI (複製儲存庫 URI) 將連結複製到剪貼簿。

使用 VS Code 的命令調色板配置服務實例時，您可以指定映像儲存庫的 URI

如需詳細資訊，請參閱《AWS App Runner 開發人員指南》中的「[以來源映像為基礎的 App Runner 服務](#)」。

Source code

若要將原始程式碼部署至 AWS App Runner 服務，該程式碼必須儲存在由支援的儲存庫提供者維護的 Git 儲存庫中。應用程式運行器支持一個源代碼庫提供商：[GitHub](#)。

如需有關設定 GitHub 存放庫的資訊，請參閱上的 [入門文件](#) GitHub。

要從 GitHub 儲存庫將源代碼部署到應用程式運行器服務，應用程式運行器建立了一個連接到 GitHub。如果您的儲存庫是私有的（也就是說，它在上不可公開訪問 GitHub），則必須向 App Runner 提供連接詳細信息。

### ⚠ Important

若要建立 GitHub 連線，您必須使用應用程式執行器主控台 (<https://console.aws.amazon.com/apprunner>) 建立連結的 GitHub 連線 AWS。使用 VS Code 的命令面板設定服務執行個體時，您可以選取 GitHub 連線頁面上可用的連線。

如需詳細資訊，請參閱 AWS App Runner 開發人員指南中的「[管理 App Runner 連線](#)」。

App Runner 服務實例提供了一個託管運行時，允許您的代碼構建和運行。AWS App Runner 目前支援下列執行階段：

- Python 受管的執行時間
- Node.js 受管的執行時間

做為服務設定的一部分，您會提供 App Runner 服務如何建置和開始服務的相關資訊。您可以使用 Command Palette (命令選擇區) 或指定一個 YAML 格式化的 [App Runner 組態檔案](#) 來輸入此資訊。此檔案中的值會指示 App Runner 如何建置和開始您的服務，並提供執行時間內容。這包括相關的網路設定和環境變數。組態檔案命名為 `apprunner.yaml`。它會自動新增到您應用程式儲存庫的根目錄中。

## 定價

您需支付應用程式使用的運算和記憶體資源的費用。此外，如果您將部署自動化，您也要針對每個涵蓋當月所有自動化部署的應用程式來支付固定月費。如果您選擇從原始碼進行部署，您還需要根據 App Runner 從原始碼建置容器所需的時間來支付建置費用。

如需詳細資訊，請參閱 [AWS App Runner 定價](#)。

### 主題

- [建立 App Runner 服務](#)
- [管理 App Runner 服務](#)

## 建立 App Runner 服務

您可以使用資源管理器和 VS 代碼的命令調色板在工具包中為 VS 代碼創建應用程式運行 AWS 服務。選擇在特定 AWS 區域中建立服務之後，Command Console 提供的編號步驟會引導您完成設定應用程式執行所在服務執行個體的程序。

在建立 App Runner 服務之前，請務必先完成[先決條件](#)。這包括提供相關的 IAM 許可，以及確認您要部署的特定來源儲存庫。

### 建立 App Runner 服務

1. 打開 AWS 資源管理器，如果它尚未打開。
2. 在 App Runner 節點上按一下滑鼠右鍵並選擇 Create Service (建立服務)。

將顯示「指令選項板」。

3. 對於 Select a source code location type (選取原始碼位置類型)，請選擇 ECR 或 Repository (儲存庫)。

如果選擇 ECR，您可以在由 Amazon Elastic Container Registry (Amazon ECR) 維護的儲存庫中指定容器映像。如果選擇 Repository (儲存庫)，您可以指定由支援的儲存庫提供者維護的原始碼儲存庫。目前，App Runner 支持[GitHub](#)作為源代碼儲存庫提供程序。

### 從 ECR 部署

1. 若為 Select or enter an image repository (選取或輸入映像儲存庫)，請選擇或輸入由您的 Amazon ECR 私有登錄或 Amazon ECR 公有映像庫維護的映像儲存庫 URL。

#### Note

如果您從 Amazon ECR 公有映像庫指定儲存庫，請確定自動部署已關閉，因為 App Runner 不支援 ECR 公有儲存庫中映像的自動部署。

依預設，自動部署處於關閉狀態，並且當指令選項板標頭上的圖示以對角線穿過它時表示。如果您選擇開啟自動部署，則會顯示一則訊息，通知您此選項可能產生額外費用。

2. 如果「命令調色盤」步驟報告找不到標籤，則您需要返回一個步驟以選取包含標記容器映像檔的存放庫。
3. 如果您使用的是 Amazon ECR 私有登錄，則需要 ECR 存取角色 ECR，AppRunner 以允許應用程式執行器存取您帳戶中的 Amazon 彈性容器登錄 (Amazon ECR) 映像。AccessRole 選擇命令面

板標題上的「+」圖標以自動創建此角色。(如果您的映像存放在 Amazon ECR 公有 (其映像可公開存取) 中，則不需要存取角色。)

4. 對於 Port (連接埠)，請輸入服務所使用的 IP 連接埠 (例如：連接埠 8000)。
5. 對於 Configure environment variables (設定環境變數)，您可以指定包含用來自訂服務執行個體中行為的環境變數的檔案。或者，您可以略過此步驟。
6. 對於 Name your service (為您的服務命名)，請輸入唯一的名稱 (不含空格)，然後按下 Enter 鍵。
7. 對於 Select instance configuration (選取執行個體組態)，請為您的服務執行個體選擇 CPU 單位和以 GB 為單位的記憶體組合。

建立您的服務時，其狀態會從建立中變更至執行中。

8. 服務開始執行後，在該服務上按一下滑鼠右鍵，然後選擇 Copy Service URL (複製服務 URL)。
9. 若要存取您部署的應用程式，請將複製的 URL 貼上至您 Web 瀏覽器的網址列中。

### 從遠端儲存庫部署

1. 對於選取連線，請選擇連結的 GitHub 連線 AWS。可供選取的連線會列在 App Runner 主控台的 GitHub 連線頁面上。
2. 針對選取遠端 GitHub 存放庫，選擇或輸入遠端存放庫的 URL。

已使用 Visual Studio 程式碼的原始程式碼控制管理 (SCM) 設定的遠端存放庫可供選取。如果沒有列出儲存庫的連結，您也可以貼上該儲存庫的連結。

3. 對於 Select a branch (選取分支)，請選擇您要部署之原始碼中的 Git 分支。
4. 對於 Choose configuration source (選擇組態來源)，請指定您要如何定義執行時間組態。

如果選擇 Use configuration file (使用組態檔案)，則您的服務執行個體是由 `apprunner.yaml` 組態檔案所定義之設定來設定。此檔案位於應用程式儲存庫的根目錄中。

如果您在此選擇 [設定所有設定]，請使用 [指令] 選項板來指定下列項目：

- Runtime (執行時間)：選擇 Python 3 或 Nodejs 12。
  - Build command (建置命令)：輸入命令以在服務執行個體的執行階段環境中建置應用程式。
  - Start command (開始命令)：輸入命令以在服務執行個體的執行階段環境中啟動應用程式。
5. 若為 Port (連接埠)，請輸入服務所使用的 IP 連接埠 (例如：連接埠 8000)。
  6. 對於 Configure environment variables (設定環境變數)，您可以指定包含用來自訂服務執行個體中行為的環境變數的檔案。或者，您可以略過此步驟。

7. 對於 Name your service (為您的服務命名)，請輸入唯一的名稱 (不含空格)，然後按下 Enter 鍵。
8. 對於 Select instance configuration (選取執行個體組態)，請為您的服務執行個體選擇 CPU 單位和以 GB 為單位的記憶體組合。

建立您的服務時，其狀態會從建立中變更至執行中。

9. 服務開始執行後，在該服務上按一下滑鼠右鍵，然後選擇 Copy Service URL (複製服務 URL)。
10. 若要存取您部署的應用程式，請將複製的 URL 貼上至您 Web 瀏覽器的網址列中。

#### Note

如果您嘗試建立 App Runner 服務失敗，服務會在 AWS Explorer 中顯示建立失敗狀態。如需疑難排解祕訣，請參閱 App Runner 開發人員指南中的「[建立服務失敗時](#)」。

## 管理 App Runner 服務

建立 App Runner 服務之後，您可以使用 [AWS 檔案總管] 窗格執行下列活動來管理它：

- [暫停和繼續 App Runner 服務](#)
- [部署 App Runner 服務](#)
- [檢視 App Runner 的日誌串流](#)
- [刪除 App Runner 服務](#)

### 暫停和繼續 App Runner 服務

如果您需要暫時禁用 Web 應用程式並停止運行代碼，則可以暫停 AWS App Runner 服務。App Runner 會將服務的運算容量縮減為零。當您準備好再次執行應用程式時，請繼續您的 App Runner 服務。App Runner 會佈建新的運算容量、將您的應用程式部署到該容量，並執行應用程式。

#### Important

只有在執行 App Runner 時，才會向您收取費用。因此，您可以視需要暫停及繼續應用程式以管理成本。這對開發和測試情境特別有幫助。

## 暫停您的 App Runner 服務

1. 打開 AWS 資源管理器，如果它尚未打開。
2. 展開 App Runner 以檢視服務清單。
3. 在服務上按一下滑鼠右鍵並選擇 Pause (暫停)。
4. 在顯示的對話方塊中，選擇 Confirm (確認)。

服務暫停時，服務狀態會從 Running (執行中) 變更至 Pausing (暫停中)，然後到 Paused (已暫停)。

## 繼續您的 App Runner 服務

1. 打開 AWS 資源管理器，如果它尚未打開。
2. 展開 App Runner 以檢視服務清單。
3. 在服務上按一下滑鼠右鍵並選擇 Resume (繼續)。

服務繼續時，服務狀態會從 Resuming (繼續中) 變更為 Running (執行中)。

## 部署 App Runner 服務

如果您為服務選擇手動部署選項，則需要明確起始每個服務的部署。

1. 打開 AWS 資源管理器，如果它尚未打開。
2. 展開 App Runner 以檢視服務清單。
3. 在服務上按一下滑鼠右鍵並選擇 Start Deployment (開始部署)。
4. 部署應用程式時，服務狀態會從 Deploying (部署中) 變更為 Running (執行中)。
5. 若要確認您的應用程式已成功部署，請在相同的服務上按一下滑鼠右鍵並選擇 Copy Service URL (複製服務 URL)。
6. 若要存取您部署的 Web 應用程式，請將複製的 URL 貼上至您 Web 瀏覽器的網址列中。

## 檢視 App Runner 的日誌串流

使用 CloudWatch 日誌來監視，存儲和訪問您的日誌流服務，例如應用程序運行器。日誌串流是共享相同來源的一系列日誌事件。

1. 展開 App Runner 以檢視服務執行個體清單。

2. 展開特定服務執行個體以檢視日誌群組清單。(日誌群組是共享相同保留、監控和存取控制設定的日誌串流群組。)
3. 在日誌群組上按一下滑鼠右鍵並選擇 View Log Streams (檢視日誌串流)。
4. 從命令面板中，從群組中選擇記錄資料流。

VS 程式碼編輯器會顯示組成串流的記錄事件清單。您可以選擇將較舊或較新的事件載入編輯器中。

## 刪除 App Runner 服務

### Important

如果您刪除 App Runner 服務，系統會永久移除該服務，並刪除存放的資料。當您需要重新建立服務時，若來源為程式碼儲存庫，則 App Runner 需要再次擷取您的來源並建置它。您的 Web 應用程式取得一個新的 App Runner 網域。

1. 打開 AWS 資源管理器，如果它尚未打開。
2. 展開 App Runner 以檢視服務清單。
3. 在服務上按一下滑鼠右鍵並選擇 Delete Service (刪除服務)。
4. 在指令選項板中，輸入刪除，然後按 Enter 確認。

刪除的服務會顯示 Deleting (刪除中) 狀態，然後服務就會從清單中消失。

## AWS 應用程式編

您可以使用與應 AWS Toolkit for Visual Studio Code 用 AWS 程式撰寫器服務搭配使用。AWS 應用程式撰寫器是 AWS 應用程式的視覺化建置器，可協助您設計應用程式架構並視覺化您的 AWS CloudFormation 基礎結構。

如需 AWS 應用程式撰寫器服務的詳細資訊，請參閱 [AWS 應用程式撰寫器](#) 使用指南。

下列主題說明如何使用「AWS 應用程式撰寫器」AWS Toolkit for Visual Studio Code。

### 主題

- [從工具組使 AWS 用應用程式撰寫器](#)

## 從工具組使 AWS 用應用程式撰寫器

AWS 的應用程式撰寫器 AWS Toolkit for Visual Studio Code 可讓您透過互動式畫布以視覺化方式設計應用程式。您也可以使用「應用程式撰寫器」來視覺化 AWS CloudFormation 及修改和 AWS Serverless Application Model (AWS SAM) 範本。使用「應用程式撰寫器」時，您的變更會持續儲存，讓您可以直接在 VS Code 編輯器中或使用互動式畫布在編輯檔案之間順暢切換。

如需有關 AWS 應用程式撰寫器服務、入門資訊和教學課程的詳細資訊，請參閱[AWS 應用程式撰寫器服務](#)使用者指南。

下列各節說明如何從存取 AWS 應用程式撰寫器服務 AWS Toolkit for Visual Studio Code。

### 從工具組存取 AWS 應用程式撰寫器

您可以從「工具組」存取「AWS 應用程式撰寫器」的 3 種主要方式。

從現有範本存取 AWS 應用程式撰寫器

1. 在 VS Code 中，在 VS 程式碼編輯器中開啟現有的範本檔案。
2. 在編輯器視窗中，按一下位於編輯器視窗右上角的「AWS 應用程式撰寫器」按鈕。
3. AWS 應用程式撰寫器會在 VS 程式碼編輯器視窗中開啟並視覺化您的範本檔案。

從內容功能表存取 AWS 應用程式撰寫器 (按一下滑鼠)

1. 在 VS Code 中，以滑鼠右鍵按一下要使 AWS 用應用程式撰寫器開啟的範本檔案
2. 在內容選單中，選擇「以應用程式撰寫器開啟」選項。
3. AWS 應用程式撰寫器會在新的 VS 程式碼編輯器視窗中開啟並視覺化您的範本檔案。

從指令選項板存取 AWS 應用程式撰寫器

1. 從 VS 代碼通過按 **Cmd + Shift + P** 或打開命令調色板 **Ctrl + Shift + P** (視窗)
2. 在搜尋欄位中，輸入「AWS 應用程式撰寫器」**AWS Application Composer** 並在結果中填入應用程式撰寫器時選擇它。
3. 選擇您要開啟的範本檔案，AWS 應用程式撰寫器會在新的 VS Code 編輯器視窗中開啟並視覺化您的範本檔案。

# AWS CDK適用於 VS 程式碼

這是預覽版本之服務的發行前版本文件。內容可能變動。

所以此AWS CDK服務使您能夠使用[AWS Cloud Development Kit \(AWS CDK\)](#)應用程式，或應用程式。您可以找到有關的詳細資訊AWS CDK中的[AWS Cloud Development Kit \(AWS CDK\)開發人員指南](#)。

AWS CDK 應用程式由稱為[建構](#)的建置區塊組成，內含其中的 AWS CloudFormation 堆疊和 AWS 資源的定義。使用AWS CDK探險者，您可以視化[堆疊](#)和[資源](#)定義於AWS CDK建構。此視覺效果以樹檢視在開發人員工具窗格中的 Visual Studio 程式碼 (VS 程式碼) 編輯器。

本節提供如何存取和使用的相關資訊AWS CDK 在 VS 代碼編輯器中。它假設您已經[安裝並設定](#)適用於本地 IDE 的 VS 代碼工具包。

## 主題

- [使 AWS CDK 用應用程式](#)

## 使 AWS CDK 用應用程式

這是預覽版中某項功能的搶鮮版說明文件。內容可能變動。

使用 VS 程式碼工 AWS 具組中的AWS CDK 檔案總管來視覺化和使用 AWS CDK 應用程式。

## 必要條件

- 請確定您的系統符合[安裝 VS Code 工具組中指定的](#)先決條件。
- 安裝命 AWS CDK 令列介面，如《[開AWS Cloud Development Kit \(AWS CDK\) 發人員指南](#)》中的 [〈入門使用〉的 AWS CDK](#)前幾節所述。

### Important

AWS CDK 版本必須是 1.17.0 或更新版本。在命令列使用 `cdk --version` 查看您正在執行的版本。

## 視覺化 AWS CDK 應用程式

使用 VS 程式碼 AWS CDK 總管的 AWS 工具組，您可以管理儲存在應用程式CDK建構中的[堆疊](#)和[資源](#)。Ex AWS CDK plorer 會使用tree.json檔案中定義的資訊 (在您執行**cdk synth**命令時建立) 在樹狀檢視中顯示您的資源。默認情況下，該tree.json文件位於應用程序的cdk.out目錄中。

若要開始使用工具組 AWS CDK 總管，您需要建立CDK應用程式。

1. 完成位於[AWS CDK 開發人員指南](#)中的 [Hello World 教程](#)的前幾個步驟。

### Important

當您到達教學課程步驟部署堆疊時，請停止並返回本指南。

### Note

例如，您可以在作業系統命令列或 VS Code 編輯器內的「終端機」視窗中執行教學課程中提供的命令。**mkdircdk init**

2. 完成CDK教學課程的必要步驟之後，請開啟您在 VS Code 編輯器中建立的CDK內容。
3. 在 AWS 導覽窗格中，展開 CDK(預覽) 標題。您的CDK應用程式及其相關聯的資源現在會顯示在CDK檔案總管樹狀檢視中。

## 重要說明

- 當您將CDK應用程式載入 VS Code 編輯器時，您可以一次載入多個資料夾。每個資料夾都可以包含多個CDK應用程式，如上圖所示。AWS CDK 檔案總管會在專案根目錄及其直接子目錄中尋找應用程式。
- 當您執行教學課程的開頭幾個步驟時，您可能注意到您執行的最後一個命令是 **cdk synth**，此命令會產生 tree.json 檔案。如果您更改CDK應用程序的各個方面，例如添加更多資源，則需要再次執行該命令以查看樹視圖中反映的更改。

## 在 AWS CDK 應用程式上執行其他作業

您可以使用 VS Code 編輯器在應用CDK程式上執行其他作業，就像使用作業系統的命令列或其他工具一樣。例如，您可以更新編輯器中的程式碼檔案，並使用 VS 程式碼終端機視窗來部署應用程式。

要嘗試這些類型的操作，請使用 VS 代碼編輯器繼續AWS CDK 開發人員指南中的 [Hello World 教程](#)。確保執行最後一步，銷毀應用程序的資源，以免您的 AWS 帳戶產生意外費用。

## 使用AWS CloudFormation堆疊

AWS Toolkit for Visual Studio Code 支援使用 [AWS CloudFormation](#) 堆疊。使用 Toolkit for VS Code，您可以通過AWS CloudFormation堆疊，例如將其刪除。

### 主題

- [刪除AWS CloudFormation堆](#)
- [使用建立 AWS CloudFormation 範本 AWS Toolkit for Visual Studio Code](#)

## 刪除AWS CloudFormation堆

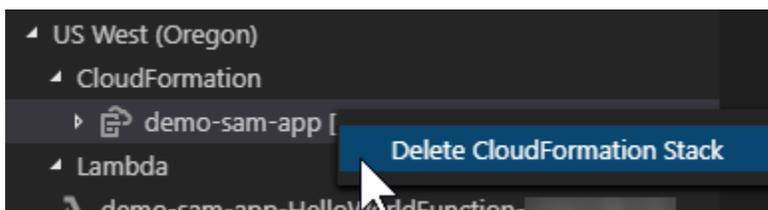
您可以使用 AWS Toolkit for Visual Studio Code 來刪除 AWS CloudFormation 堆疊。

### 先決條件

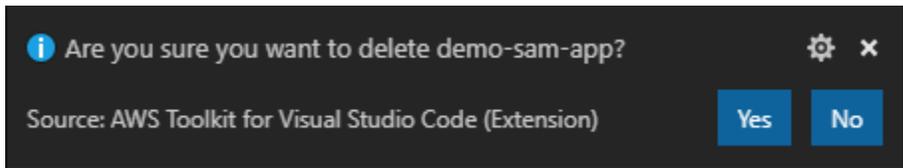
- 請確定您的系統符合[安裝 Toolkit for VS Code](#)。
- 確認您在 [身分驗證與存取](#) 中設定的登入資料包括 AWS CloudFormation 服務的適合讀取/寫入存取。如果在 AWS Explorer 的 CloudFormation 下方，您看到類似「Error loading CloudFormation resources」(載入 CloudFormation 資源時發生錯誤) 的訊息，請檢查與這些憑證連接的許可。您對這些許可所做的變更需要幾分鐘的時間才會對AWS探險者在 VS 代碼中。

## 刪除 CloudFormation 堆疊

1. 在 AWS Explorer 中，開啟您要刪除之 AWS CloudFormation 堆疊的內容功能表。



2. 選擇 Delete CloudFormation Stack (刪除 CloudFormation 堆疊)。
3. 在出現的訊息中，選擇 Yes (是)，以確認刪除。



刪除堆疊後，該堆疊不會再列於 AWS Explorer 之中。

## 使用建立 AWS CloudFormation 範本 AWS Toolkit for Visual Studio Code

AWS Toolkit for Visual Studio Code 可以幫助您編寫 AWS CloudFormation 和 SAM 模板。

### 必要條件

VS 代碼和憑據先決條件的工具包

- 在您可以從 Toolkit for VS Code 訪問 CloudFormation 服務之前，您需要滿足用戶指南中概述的要求 [安裝 Toolkit for VS Code](#)。
- 您在中建立的認證 [身分驗證與存取](#) 必須包含對 AWS CloudFormation 服務的適當讀取/寫入存取權。

#### Note

如果 CloudFormation 服務顯示載入 CloudFormation 資源時發生錯誤訊息，請檢查您已附加至這些認證的權限。另請注意，對權限所做的變更可能需要幾分鐘的時間才能在 AWS 檔案總管中更新。

CloudFormation 範本先決條

- 安裝並啟用 [Redhat 開發人員 YAML VS 程式碼](#) 擴充功能。
- 使用 Redhat 開發人員 YAML VS 代碼擴展時，您需要連接到互聯網，因為它用於在您的機器上下載和現金 JSON 模式。

## 使用 YAML 結構描述 Support 撰寫 CloudFormation 範本

此工具組使用 YAML 語言支援和 JSON 結構描述來簡化撰寫 CloudFormation 和 SAM 範本的程序。語法驗證和自動完成等功能不僅可以使過程更快，還有助於提高模板的質量。為範本選取結構描述時，建議採用下列最佳作法。

### CloudFormation 範本

- 檔案的副檔名為 .yaml 或 .yml。
- 檔案具有頂層 `AWSTemplateFormatVersion` 或資源節點。

### SAM 範本

- 已描述的所有條件 CloudFormation
- 檔案具有頂層「轉換」節點，其中包含以開頭的值 `AWS::Serverless`。

模式將在文件修改時應用。例如，將無伺服器轉換新增至範本並儲存檔案之後，將套用 SAM 範本 CloudFormation 本結構描述。

### 語法驗證

YAML 擴充功能會自動將類型驗證套用至您的範本。這會反白指定屬性具有無效類型的項目。如果您將鼠標懸停在突出顯示的條目上，擴展程序將顯示更正操作

### 自動完成

新增欄位、列舉值或其他[資源類型](#)時，您可以輸入 Ctrl + 空格來啟動 YAML 擴充功能的自動完成功能。

## 使用 CloudWatch 記錄檔使用 AWS Toolkit for Visual Studio Code

亞馬遜 CloudWatch 日誌可讓您集中管理所有系統、應用程式和 AWS 您在單一、可高度擴充的服務中使用的服務。然後，您可以輕鬆檢視日誌、在日誌中搜尋特定的錯誤碼或模式、根據特定欄位篩選日誌，或安全封存日誌以供日後分析。如需詳細資訊，請參閱[什麼是亞馬遜 CloudWatch 日誌？](#)在亞馬遜 CloudWatch 使用者指南。

下列主題說明如何使用 AWS Toolkit for Visual Studio Code 一起工作 CloudWatch 在一個登錄 AWS 帳戶。

### 主題

- [檢視 CloudWatch 日誌組和日誌串流AWS Toolkit for Visual Studio Code](#)
- [使用 CloudWatch 在 Log Stream 中使用日誌事件AWS Toolkit for Visual Studio Code](#)
- [搜尋 CloudWatch 記錄群組](#)

## 檢視 CloudWatch 日誌組和日誌串流AWS Toolkit for Visual Studio Code

日誌串流是一系列共用相同來源的日誌事件。每個單獨日誌串流是由 CloudWatch Logs 中的單獨日誌來源所組成。

日誌群組是共用相同保留、監控和存取控制設定的日誌串流群組。您可以定義日誌群組，並指定放入每個群組的串流。可以屬於一個日誌群組的日誌串流數量並沒有限制。

如需詳細資訊，請參閱 Amazon CloudWatch Logs 使用者指南中的[使用日誌群組和日誌串流](#)。

### 主題

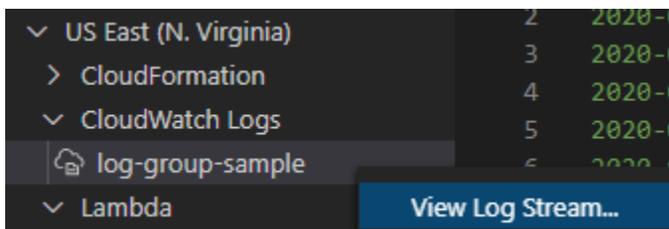
- [以 CloudWatch Logs 節點檢視日誌群組和日誌串流](#)

### 以 CloudWatch Logs 節點檢視日誌群組和日誌串流

1. 在 VS 代碼中，選擇檢視、探險者若要打開AWSExplorer
2. 按一下 CloudWatch Logs 節點展開日誌群組清單。

目前 AWS 區域的日誌群組會顯示在 CloudWatch Logs 節點下方。

3. 若要檢視日誌組中的日誌串流，請在日誌組名稱上按一下滑鼠右鍵，然後選擇檢視日誌串流。



4. 從Command Palette下，選擇要檢視的日誌串流。

#### Note

所以此Command Palette顯示每個流中最後一個事件的時間戳。

所以此[Log Stream \(日誌串流\)編輯](#)啟動以顯示流的日誌事件。

# 使用 CloudWatch 在 Log Stream 中使用日誌事件AWS Toolkit for Visual Studio Code

在您打開之後日誌串流編輯器中，您可以訪問每個串流中的日誌事件。日誌事件是由應用程式或正受到系統監控的資源所記錄的活動記錄。

## 主題

- [檢視並複製日誌串流資訊](#)
- [將日誌串流編輯器的內容儲存至本機檔案](#)

## 檢視並複製日誌串流資訊

當您開啟記錄資料流時，Log Stream (日誌串流)編輯器顯示該串流的日誌事件序列。

1. 若要找到要檢視的日誌串流，請開啟Log Stream (日誌串流)編輯器 (請參閱[檢視 CloudWatch 日誌群組和日誌串流](#))。

列出事件的每一行都會加上時間戳記，藉此顯示事件的記錄時間。

2. 您可以使用下列選項檢視和複製串流事件的相關資訊：
  - 按時間查看事件：選擇以顯示最新和較舊的記錄事件載入較新事件或者載入較舊的事件。

### Note

Log Stream (日誌串流)編輯器一開始會批次載入最近 10,000 行的日誌事件或 1 MB 的日誌資料 (以較小者為優先)。如果選擇 Load newer events (載入較新的事件)，編輯器會顯示上一批次載入後記錄的事件。如果選擇 Load older events (載入較舊的事件)，編輯器會顯示目前顯示的事件之前發生的批次事件。

- 複製日誌事件：選取要複製的事件，然後按一下滑鼠右鍵並選取複製從功能表。
- 複製記錄資料流的名稱：在標籤上按一下滑鼠右Log Stream (日誌串流)編輯器和選擇複製日誌串流名稱。

### Note

您也可以使用Command Palette請執行。AWS工具組複製日誌串流名稱。

## 將日誌串流編輯器的內容儲存至本機檔案

您可以下載 CloudWatch 日誌流編輯器到 alog 在您的本機電腦上使用本機電腦上

### Note

透過此選項，您可以僅將目前顯示在日誌事件的日誌事件儲存至檔案。舉例來說，如果日誌串流的大小總計為 5MB，而只有 2MB 載入了編輯器，則儲存的檔案也只會包含 2MB 的日誌資料。若要顯示更多要儲存的資料，請在編輯器中選擇 Load newer events (載入較新的事件) 或 Load older events (載入較舊的事件)。

1. 若要找到要複製的日誌串流，請開啟日誌串流編輯器 (請參閱[檢視 CloudWatch 日誌群組和日誌串流](#))。
2. 選擇 Save 顯示記錄資料流名稱的標籤旁邊的圖示。

### Note

您也可以使用 Command Palette 請執行。AWS 工具組儲存目前的記錄串流內容。

3. 透過對話方塊選取或建立日誌檔案的下載資料夾，然後按一下 Save (儲存)。

## 搜尋 CloudWatch 記錄群組

您可以使用「搜尋日誌群組」來搜尋日誌群組中的所有日誌資料流。

如需 Amazon CloudWatch 日誌服務的詳細資訊，請參閱 Amazon CloudWatch 使用者指南中的使用[日誌群組和日誌串流](#)主題。

### 從 VS 程式碼命令選項板搜尋記錄群組

若要從 VS Code 命令面板搜尋記錄群組，請完成下列步驟。

如需 Amazon CloudWatch 日誌篩選器和模式的詳細資訊，請參閱 Amazon CloudWatch 使用者指南中的[篩選器和模式語法](#)一節。

1. 從 VS 代碼中，按 `cmd+shift+p` ( 窗口 : `ctrl+shift+p` ) 打開命令調色板。
2. 在 [命令面板] 中，輸入指令 **AWS: Search Log Group**，然後選取它在 VS Code 中開啟 [搜尋記錄群組] 對話方塊，並依照提示繼續進行。

**Note**

在第一個提示中，您可以選擇在繼續執行後續步驟之前切換您的AWS區域。

3. 從選取記錄群組 (1/3) 提示中，選擇您要搜尋的記錄群組。
4. 從「選取時間篩選」(2/3) 提示中，選擇要套用至搜尋的時間篩選條件。
5. 從搜尋日誌群組... (3/3) 提示中，於提供的欄位中輸入您的搜尋模式，然後按**Enter**鍵繼續，或按**ESC**鍵取消搜尋。
6. 搜尋完成後，您的搜尋結果會在 VS Code 編輯器中開啟。

## 從檔AWS案總管搜尋記錄群組

若要從檔AWS Toolkit for Visual Studio Code案總管搜尋記錄群組，請完成下列步驟。

1. 從AWS Toolkit for Visual Studio Code檔案總管展開CloudWatch。
2. 開啟 (按一下滑鼠右鍵) 要搜尋之「搜尋日誌群組」的內容功能表，然後選擇「搜尋日誌群組」以開啟搜尋提示。
3. 依照提示選取時間範圍以繼續。
4. 出現提示時，在提供的欄位中輸入您的搜尋模式，然後按**Enter**鍵繼續，或按**ESC**鍵取消搜尋。
5. 搜尋完成後，您的搜尋結果會在 VS Code 編輯器中開啟。

## 使用搜尋記錄結果

成功完成 CloudWatch 記錄群組搜尋後，您的搜尋結果會在 VS Code 編輯器中開啟。下列程序說明如何使用搜尋記錄結果。

**Note**

檢視單一記錄串流時，下列功能僅限於目前作用中的記錄串流中的結果。

## 儲存搜尋記錄群組結果

若要在本機儲存搜尋記錄群組結果，請完成下列步驟。

1. 從搜尋記錄群組結果中，選擇位於 VS Code 編輯器右上角的 [將記錄儲存至檔案] 圖示按鈕。
2. 從「另存新檔」提示中，指定要儲存檔案的名稱和位置。
3. 選擇「確定」會將檔案儲存到您的本機電腦。

### 更改時間範圍的時間範圍

若要變更搜尋記錄群組結果中使用中的時間範圍，請完成下列步驟。

1. 從搜尋記錄群組結果中，選擇搜尋日期... 圖標按鈕，位於 VS 代碼編輯器的右上角。
2. 從「選取時間篩選」提示中，為搜尋記錄結果選擇新的時間範圍。
3. 關閉「選取時間篩選」提示時，會更新您的結果。

### 變更搜尋模式

若要變更搜尋記錄群組結果中使用的搜尋模式，請完成下列步驟。

1. 從搜尋記錄群組結果中，選擇依模式搜尋... 圖標按鈕，位於 VS 代碼編輯器的右上角。
2. 從搜尋日誌群組提示，在提供的欄位中輸入新的搜尋模式。
3. 按下 **Enter** 鍵可關閉提示，並使用新的搜尋模式更新結果。

## 使用 Amazon Elastic Registry

Amazon ECR 是管理的容器登錄服務 (Amazon ECR) 是 AWS 管理的容器登錄服務，具安全性和可擴展性。您可以從 Toolkit for VS Code Explorer 存取數個 Amazon ECR 服務功能。

- 建立儲存庫。
- 為儲存庫或標記影像建立 AWS App Runner 服務。
- 存取影像標籤和儲存庫 URI 或 ARN。
- 刪除映像檔標籤和儲存庫。

您也可以將 AWS CLI 和其他平台與 VS Code 存取完整的 Amazon ECR 功能。

如需有關 Amazon ECR 的詳細資訊，請參閱 Amazon Elastic Container Registry 使用者指南中的 [什麼是 Amazon ECR ?](#)。

## 先決條件

您必須完成下列步驟，才能夠完成下列步驟，才能夠從 VS 程式 Explorer 存取 Amazon ECR 服務。

### 建立 IAM 使用者

您必須先提供登入資料，才能存取AWS服務 (例如 Amazon ECR)。如此，服務能夠判斷您是否擁有使用其資源的權限。不建議您AWS直接透過根AWS帳戶的登入資料存取。而是使用AWS Identity and Access Management (IAM) 建立 IAM 使用者，然後將該使用者新增至具有管理許可的 IAM 群組。之後，您可以使用特殊的 URL 和 IAM 使用者的憑證來存取 AWS。

如果您已註冊，AWS但是尚未建立 IAM 使用者，則您可以使用 IAM 主控台建立使用者。

若要建立管理員使用者，請選擇下列其中一個選項。

選擇一種管理管理員的方式	若要	By	您也可以
在 IAM Identity Center (建議)	使用短期憑證存取 AWS。  這與安全性最佳實務一致。有關最佳實務的資訊，請參閱 IAM 使用者指南中的 <a href="#">IAM 安全最佳實務</a> 。	請遵循 AWS IAM Identity Center 使用者指南的 <a href="#">入門</a> 中的說明。	請參閱 AWS Command Line Interface 使用者指南中的 <a href="#">設定 AWS CLI 以使用 AWS IAM Identity Center</a> 設定程式設計存取。
在 IAM 中 (不建議使用)	使用長期憑證存取 AWS。	請遵循 IAM 使用者指南中 <a href="#">建立您的第一個 IAM 管理員使用者和使用者群組</a> 的說明。	請參閱 <a href="#">IAM 使用者指南</a> 中的管理 IAM 使用者的存取金鑰，設定程式設計存取。

若要以此新 IAM 使用者身分登入，請登出AWS主控台，然後使用下列 URL。在下列 URL 中，其中 `your_aws_account_id` 是您去掉連字號的AWS帳號 (例如，如果您的AWS帳號是1234-5678-9012，您的AWS帳戶 ID 就是123456789012)：

```
https://your_aws_account_id.signin.aws.amazon.com/console/
```

輸入您剛才建立的 IAM 使用者名稱和密碼。登入時導覽列會顯示「*your\_user\_name* @ *your\_aws\_account\_id*」。

如果您不希望登入頁面的 URL 包含您的 AWS 帳戶 ID，可以建立一個帳戶別名。在 IAM 儀表板中選擇 Customize (自訂)，然後輸入帳戶別名。這可以是您的公司名稱。如需詳細資訊，請參閱 [IAM 使用者和別名](#)。AWS

若要在建立帳戶別名後登入，請使用下列 URL：

```
https://your_account_alias.signin.aws.amazon.com/console/
```

若要驗證帳戶的 IAM 使用者的登入連結，請開啟 IAM 主控台，然後在儀表板的 IAM users sign-in link (IAM 使用者登入連結) 下方檢查。

如需 IAM 的詳細資訊，請參閱《[AWS Identity and Access Management 使用者指南](#)》。

## 安裝及設定泊塢視窗

您可以從「安裝 Docker [Engine](#)」[使用手冊中選取偏好的作業系統，並依照指示進行安裝和設定 Docker](#)。

## 安裝和設定AWS CLI 第 2 版

從「[安裝、更新和解除安裝AWS CLI 第 2 版使用者指南](#)」中選取您偏好的作業系統，以安裝和設定 [AWS CLI 第 2 版](#)。

## 主題

- [在 VS 代碼中使用亞馬遜彈性容器註冊表服務](#)

## 在 VS 代碼中使用亞馬遜彈性容器註冊表服務

您可以直接從 Amazon Elastic Container Registry (Amazon ECR) 服務AWSVS 代碼中的資源管理器並使用它將程序映像推送到 Amazon ECR 存儲庫。若要開始使用，您需要執行下列步驟：

1. 創建一個 Docker 文件，其中包含構建映像所需的信息。
2. 從該 Docker 文件構建圖像並標記圖像以進行處理。
3. 在您的 Amazon ECR 執行個體內建立儲存庫。

## 4. 將標記映像推送到您的儲存庫。

### 章節

- [先決條件](#)
- [1. 建立 Dockerfile](#)
- [2. 從 Dockerfile 建置映像](#)
- [3. 建立新的儲存器](#)
- [4. 推送、提取和刪除映像](#)

### 先決條件

您必須符合下列條件，才能使用 Toolkit for VS Code 的 Amazon ECR 服務功能[先決條件](#)。

#### 1. 建立 Dockerfile

Docker 使用一個名為 Dockerfile 的文件來定義可以推送和存儲在遠程儲存庫上的圖像。在您可以將圖像上傳到 ECR 儲存庫之前，您必須創建一個 Docker 文件，然後從該 Docker 文件構建圖像。

#### 建立 Dockerfile

1. 使用 VS Code Explorer 的工具組，導覽至您要儲存 Dockerfile 的目錄。
2. 建立名為的新檔案 Dockerfile。

#### Note

VS Code 可能會提示您選擇文件類型或文件擴展名。如果發生此問題，請選取純文字。Vs 代碼有一個「碼頭文件」擴展名。但是，我們建議您不要使用它。這是因為擴充功能可能會導致與特定版本的 Docker 或其他相關應用程式發生衝突。

#### 使用 VS 代碼編輯您的碼頭文件

如果 Dockerfile 有副檔名，請開啟該檔案的內容 (按一下右鍵) 選單，並移除副檔名。

從 Docker 文件中刪除文件擴展名後：

1. 直接在 VS 代碼中打開空的碼頭文件。
2. 將以下示例的內容複製到您的 Docker 文件中：

## Example Dockerfile 映像範例

```
FROM ubuntu:18.04

# Install dependencies
RUN apt-get update && \
    apt-get -y install apache2

# Install apache and write hello world message
RUN echo 'Hello World!' > /var/www/html/index.html

# Configure apache
RUN echo '. /etc/apache2/envvars' > /root/run_apache.sh && \
    echo 'mkdir -p /var/run/apache2' >> /root/run_apache.sh && \
    echo 'mkdir -p /var/lock/apache2' >> /root/run_apache.sh && \
    echo '/usr/sbin/apache2 -D FOREGROUND' >> /root/run_apache.sh && \
    chmod 755 /root/run_apache.sh

EXPOSE 80

CMD /root/run_apache.sh
```

這是使用 Ubuntu 18.04 映像的 Dockerfile。所以此跑指示會更新套件快取。安裝 Web 伺服器的軟體套件服務，然後寫入「Hello World!」內容至 Web 伺服器的根目錄。所以此揭露指令公開容器上的連接埠 80，並且命令提示字元指令啟動 Web 伺服器。

### 3. 儲存 Dockerfile。

#### Important

確保您的 Docker 文件沒有附加到該名稱的擴展名。具有擴展名的 Docker 文件可能會導致與 Docker 或其他相關應用程序的某些版本發生衝突。

## 2. 從 Dockerfile 建置映像

您創建的 Docker 文件包含構建程序圖像所需的信息。您必須先建立映像檔，才能將該映像推送到 Amazon ECR 執行個體。

## 從 Dockerfile 建置映像

1. 使用 Docker CLI 或與您的 Docker 執行個體整合的 CLI，瀏覽至包含 Docker 檔案的目錄。
2. 執行碼頭構建命令來構建 Docker 文件中定義的圖像。

```
docker build -t hello-world .
```

3. 執行 Docker 影像指令，驗證映像已正確建立。

```
docker images --filter reference=hello-world
```

Example 輸範例：

REPOSITORY SIZE	TAG	IMAGE ID	CREATED
hello-world 241MB	latest	e9ffedc8c286	4 minutes ago

- 4.

### Note

此步驟不是建立或推送映像檔的必要步驟，但是您可以看到程式映像檔在執行時的運作方式。

執行新建置的映像，請使用 Docker 執行指令。

```
docker run -t -i -p 80:80 hello-world
```

所以此-p在前面的示例中指定的選項映射公開連接埠 80在容器上連接埠 80的主機系統。如果您在本地運行 Docker，請導航到<http://localhost:80>使用您的 Web 瀏覽器。如果程序正確執行，「Hello World!」顯示語句。

如需有關的詳細資訊 Docker 執行指令，請參閱[Docker 執行參考](#)在 Docker 網站上。

### 3. 建立新的儲存器

若要將映像上傳到 Amazon ECR 執行個體，請建立可存放於其中的新儲存庫。

建立新的 Amazon ECR 儲存庫

1. 來自 VS 代碼 Docker/，選擇 AWS 工具組圖示。
2. 展開 AWS 探險者 Menu 選單。
3. 找到預設值 AWS 與您相關聯的區域 AWS 帳戶。然後，選擇它以查看通過 Toolkit for VS Code 服務列表。
4. 選擇 ECR + 選項以開始建立新的儲存器程序。
5. 遵循提示內容完成程序。
6. 完成後，您可以從 ECR 的 區段 AWSExplorer 選單。

### 4. 推送、提取和刪除映像

從 Docker 檔案建立映像並建立儲存庫之後，您可以將映像推送到 Amazon ECR 儲存庫。此外，使用 AWS 資源管理器與碼頭和 AWS CLI，您可以執行下列操作：

- 從您的儲存庫中提取映像。
- 刪除儲存在儲存庫中的映像檔。
- 刪除您的存放庫。

使用默認註冊表對 Docker 進行身份驗證

在 Amazon ECR 和 Docker 執行個體之間交換資料時，需要進行身份驗證。要使用您的註冊表驗證 Docker：

1. 開啟連接至執行個體的命令列作業系統 AWS CLI。
2. 使用 `get-login-password` 向您的私人 ECR 註冊表進行身份驗證的方法。

```
aws ecr get-login-password --region region | docker login --username AWS --password-stdin AWS_account_id.dkr.ecr.region.amazonaws.com
```

**⚠ Important**

在上述命令中，您必須同時更新`region`與`AWS_account_id`的特定於您的信息AWS帳戶。

**標記映像並推送映像至您的儲存庫**

使用您的實例對 Docker 進行身份驗證後AWS，將映像推送到您的儲存庫。

1. 使用Docker 影像指令可檢視儲存在本機的影像，並識別您要標記的影像。

```
docker images
```

Example 輸範例：

REPOSITORY SIZE	TAG	IMAGE ID	CREATED
hello-world 241MB	latest	e9ffedc8c286	4 minutes ago

2. 使用標記您的圖像Docker 標籤指令。

```
docker tag hello-world:latest AWS_account_id.dkr.ecr.region.amazonaws.com/hello-world:latest
```

3. 將已標記的映像推送到您的儲存庫中Docker 標籤指令。

```
docker push AWS_account_id.dkr.ecr.region.amazonaws.com/hello-world:latest
```

Example 輸範例：

```
The push refers to a repository [AWS_account_id.dkr.ecr.region.amazonaws.com/hello-world] (len: 1)
e9ae3c220b23: Pushed
a6785352b25c: Pushed
0998bf8fb9e9: Pushed
```

```
0a85502c06c9: Pushed
latest: digest:
sha256:215d7e4121b30157d8839e81c4e0912606fca105775bb0636b95aed25f52c89b size: 6774
```

將標記的影像成功上傳至儲存庫之後，就會顯示在AWSExplorer 選單。

從 Amazon ECR 提取映像

- 您可以將映像檔拉到您的本機執行個體Docker 標籤指令。

```
docker pull AWS_account_id.dkr.ecr.region.amazonaws.com/hello-world:latest
```

Example 輸範例：

```
The push refers to a repository [AWS_account_id.dkr.ecr.region.amazonaws.com/hello-world] (len: 1)
e9ae3c220b23: Pushed
a6785352b25c: Pushed
0998bf8fb9e9: Pushed
0a85502c06c9: Pushed
latest: digest:
sha256:215d7e4121b30157d8839e81c4e0912606fca105775bb0636b95aed25f52c89b size: 6774
```

從 Amazon ECR 儲存庫中刪除映像

從 VS Code 中刪除映像的方法有兩種。第一種方法是使用AWSExplorer。

1. 從AWS資源管理器中，展開ECR菜單
2. 展開您要從中刪除影像的儲存庫
3. 通過打開右鍵菜單，選擇與要刪除的圖像關聯的圖像標籤 ( 右鍵單擊 )
4. 選擇刪除標籤...選項刪除與該標籤關聯的所有存儲圖像

刪除映像AWSCLI

- 您也可以使用AWSecr batch-delete-image指令。

```
AWS ecr batch-delete-image \  
  --repository-name hello-world \  
  --image-ids imageTag=latest
```

Example 輸範例：

```
{  
  "failures": [],  
  "imageIds": [  
    {  
      "imageTag": "latest",  
      "imageDigest":  
      "sha256:215d7e4121b30157d8839e81c4e0912606fca105775bb0636b95aed25f52c89b"  
    }  
  ]  
}
```

從您的亞馬遜 ECR 執行個體刪除儲存庫

從 VS Code 中刪除儲存庫的方法有兩種。第一種方法是使用AWSExplorer。

1. 從AWS資源管理器中，展開ECR菜單
2. 開啟內容 (按一下滑鼠右鍵) 選單，選擇您要刪除的儲存庫
3. 選擇刪除儲存庫...選擇的儲存庫

從刪除 Amazon ECR 儲存庫AWSCLI

- 您可以使用AWSecr 刪除存儲庫指令。

#### Note

依預設，您不能刪除包含映像的儲存庫。但是，--force標誌允許這個。

```
AWS ecr delete-repository \  
--repository-name hello-world \  
--force
```

Example 輸範例：

```
{  
  "failures": [],  
  "imageIds": [  
    {  
      "imageTag": "latest",  
      "imageDigest":  
"sha256:215d7e4121b30157d8839e81c4e0912606fca105775bb0636b95aed25f52c89b"  
    }  
  ]  
}
```

## 使用 Amazon Elastic Container Service

所以此AWS Toolkit for Visual Studio Code提供了一些[Amazon Elastic Container Service \(Amazon ECS\)](#)。Toolkit for VS Code 可幫助您完成某些與 Amazon ECS 相關的工作，例如創建任務定義。

主題

- [對亞馬遜雲服務器任務定義文件使用 IntelliSense](#)
- [Amazon 彈性容器服務執行 AWS Toolkit for Visual Studio Code](#)

### 對亞馬遜雲服務器任務定義文件使用 IntelliSense

使用亞馬遜 Elastic Container Service (Amazon ECS) 時，您可能需要執行的操作之一是創建任務定義，如[建立任務定義](#)來自Amazon Elastic Container Service 開發者指南。當您安裝AWS Toolkit for Visual Studio Code時，就會安裝 Amazon ECS 任務定義檔案的 IntelliSense 功能。

先決條件

- 請確定您的系統符合[安裝 Toolkit for VS Code](#)。

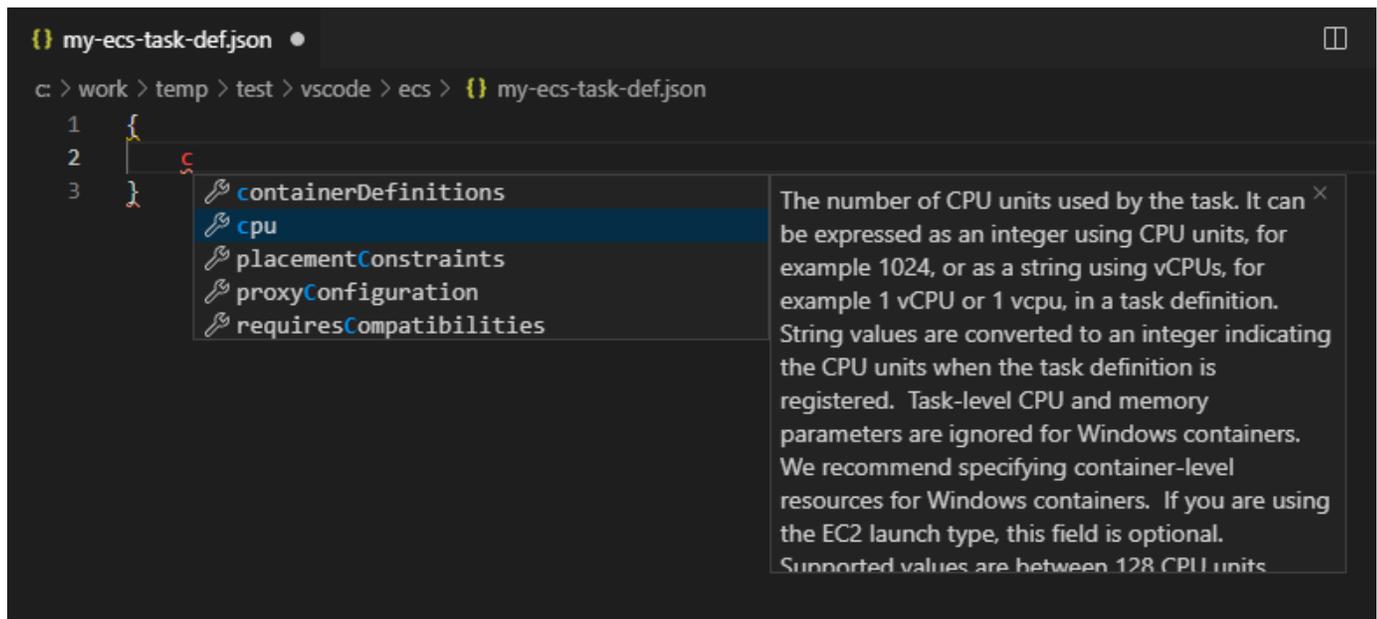
## 在 Amazon Elastic Container IntelliSense 任務定義檔案中使用 IntelliSense

下列範例示範如何在 Amazon ECS 任務定義檔案中利用 IntelliSense。

1. 為您的 Amazon ELES 任務定義建立 JSON 檔案。檔案的名稱結尾必須是 `ecs-task-def.json`，但在開頭可以有其他字元。

在此範例中，請建立名為 `my-ecs-task-def.json` 的檔案

2. 在 VS Code 編輯器中打開檔案，然後再輸入起始大括號。
3. 輸入字母“c”，表示您要將 `cpu` 新增至定義中。觀察開啟的 IntelliSense 對話方塊，類似如下。



## Amazon 彈性容器服務執行 AWS Toolkit for Visual Studio Code

您可以使用 Amazon ECS Exec 功能在 Amazon 彈性容器服務 (亞馬遜 ECS) 容器中發出單個命令。  
AWS Toolkit for Visual Studio Code

### ⚠ Important

啟用和停用 Amazon ECS Exec 會變更您帳戶中的 AWS 資源狀態。包括停止和重新啟動服務。若在啟用 Amazon ECS Exec 時變更資源狀態，可能會導致無法預期的結果。如需 Amazon ECS 的詳細資訊，請參閱[使用 Amazon ECS Exec](#) 進行偵錯的開發人員指南。

## Amazon ECS Exec 的先決條件

在您可以使用 Amazon ECS Exec 功能之前，需要滿足一些先決條件。

### Amazon ECS 要求

視您的任務是在 Amazon EC2 上託管 AWS Fargate (Fargate)，還是 Amazon ECS Exec 有不同的版本要求。

- 如果您使用的是 Amazon EC2，則必須使用在 2021 年 1 月 20 日之後發行的 Amazon ECS 最佳化 AMI，且代理程式版本為 1.50.2 或更高版本。您可以在開發人員指南 [《Amazon ECS 最佳化 AMI》](#) 中取得其他資訊。
- 如果您使用的是 AWS Fargate，則必須使用 1.4.0 或更高版本的平台。您可以在開發人員指南 [《AWS Fargate 平台版本》](#) 中獲得有關 Fargate 要求的其他資訊。

### AWS 帳戶組態和 IAM 許可

若要使用 Amazon ECS Exec 功能，您必須有一個現有的 Amazon ECS 叢集與您的帳戶相關聯。AWS Amazon ECS Exec 使用 Systems Manager 來與叢集中的容器建立連線，且需要特定的任務 IAM 角色許可才能與 SSM 服務通訊。

如需 Amazon ECS Exec 專屬的 IAM 角色和政策資訊，請參閱 [《ECS Exec 所需的 IAM 許可》](#) 開發人員指南。

## 使用 Amazon ECS Exec

您可以直接從 Toolkit for VS Code 中的 AWS 資源管理器啟用或停用 Amazon ECS Exec。啟用 Amazon ECS Exec 後，您可以從 Amazon ECS 功能表中選擇容器，然後對它們執行命令。

### 啟用 Amazon ECS Exec

1. 在 AWS 檔案總管中，找出並展開 Amazon ECS 功能表。
2. 使用您要修改的服務展開叢集。
3. 開啟服務的內容功能表 (按一下滑鼠右鍵)，然後選擇 Enable Command Execution (啟用命令執行)。

**⚠ Important**

此操作會啟動服務的新部署，可能需要幾分鐘的時間。如需詳細資訊，請參閱本節開頭的附註。

### 停用 Amazon ECS Exec

1. 在 AWS 檔案總管中，找出並展開 Amazon ECS 功能表。
2. 展開容納所需服務的叢集。
3. 開啟服務的內容功能表 (按一下滑鼠右鍵)，然後選擇 Disable Command Execution (停用命令執行)。

**⚠ Important**

此操作會啟動服務的新部署，可能需要幾分鐘的時間。如需詳細資訊，請參閱本節開頭的附註。)

### 針對容器執行命令

若要使用 AWS 檔案總管對容器執行命令，必須啟用 Amazon ECS Exec。如果未啟用，請參閱本節中的「啟用 ECS Exec」程序。

1. 在 AWS 檔案總管中，找出並展開 Amazon ECS 功能表。
2. 展開容納所需服務的叢集。
3. 展開服務以列出關聯的容器。
4. 開啟 (按一下滑鼠右鍵) 容器的內容選單，然後選擇 Run Command in Container (在容器中執行命令)。
5. 將打開一個提示，其中包含正在運行的任務列表，選擇所需的任務 ARN。

**i Note**

如果該服務只有一個工作正在執行，則會自動選取該工作，並略過此步驟。

6. 出現提示時，鍵入要運行的命令，然後按 Enter 進行處理。

# 使用 Amazon EventBridge

所以此AWS Toolkit for Visual Studio Code ( VS 代碼 ) 提供[Amazon EventBridge](#)。可讓您使 Toolkit for VS Code , 您可讓您使用 EventBridge 的某些功能 , 例如結構描述。

## 主題

- [使用 Amazon EventBridge 模式](#)

## 使用 Amazon EventBridge 模式

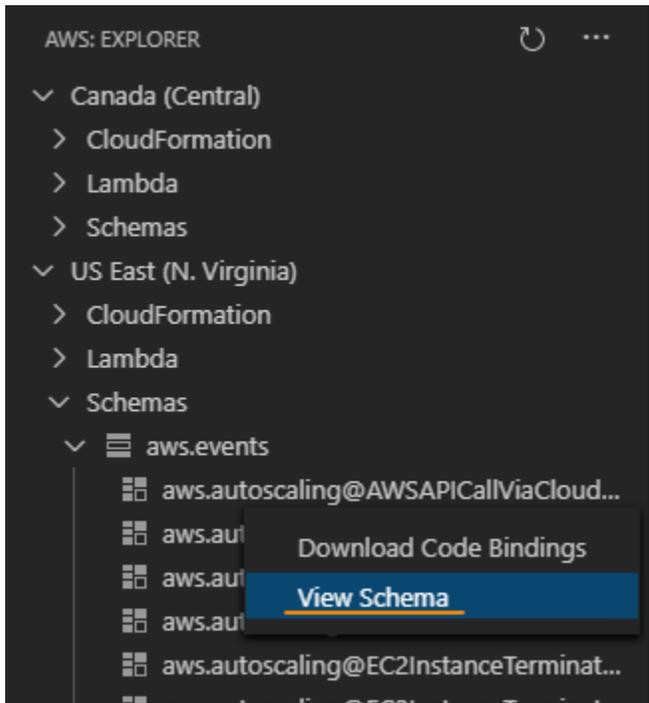
您可以使用 AWS Toolkit for Visual Studio Code ( VS 代碼 ) 在 [Amazon EventBridge 模式](#)上執行各種操作。

## 必要條件

- 請確定您的系統符合[安裝 VS Code 工具組中指定的](#)先決條件。
- 您要使用的 EventBridge 結構描述必須在您的 AWS 帳戶中可用。如果沒有 , 請建立或上傳。請參閱 [Amazon EventBridge 用戶指南中的 Amazon EventBridge 模式](#)。

## 檢視可用的結構描述

1. 在 AWS Explorer 中 , 展開 Schemas (結構描述)。
2. 展開登錄的名稱 , 內含您要檢視的結構描述。例如 , 許多提 AWS 供的結構描述都在 aws.events 登錄中。
3. 若要在編輯器中檢視結構描述 , 請開啟結構描述的內容功能表 , 然後選擇 View Schema (檢視結構描述)。

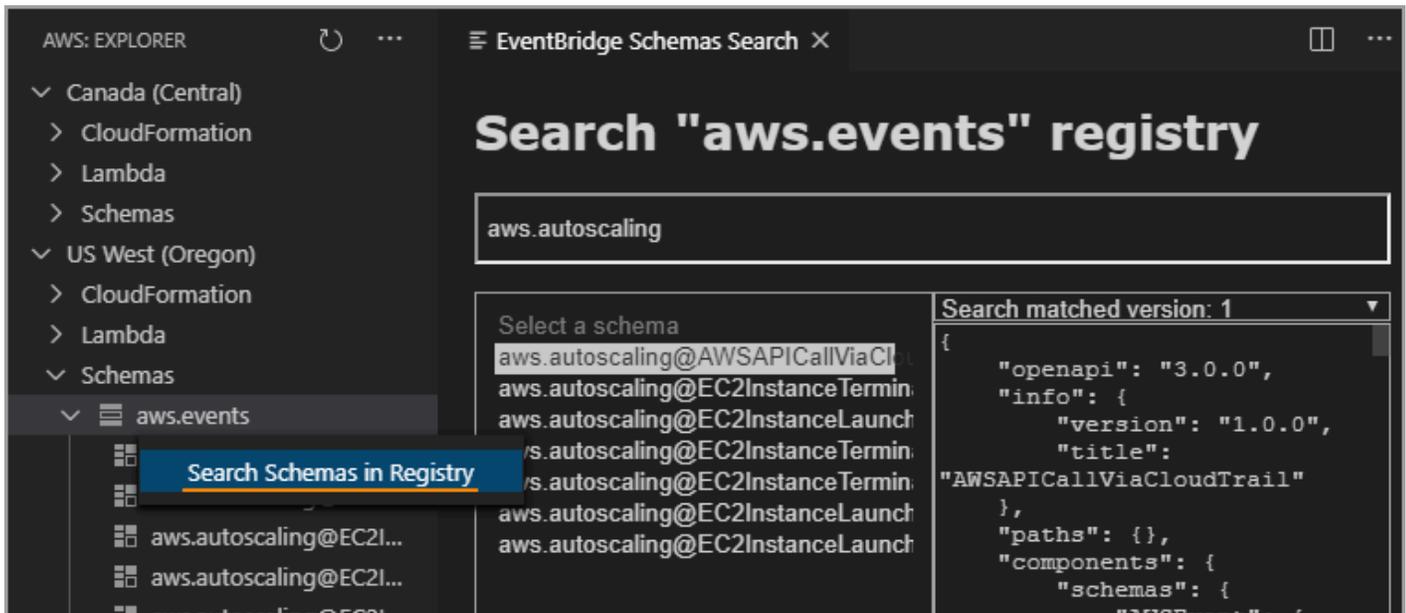


## 尋找可用的結構描述

在 AWS Explorer 中，執行下列一或多個動作：

- 開始輸入您要尋找的結構描述的標題。AWS Explorer 會反白包含相符項目的結構標題。(您必須展開登錄才能看到反白顯示的標題。)
- 開啟 Schemas (結構描述) 的內容功能表，然後選擇 Search Schemas (搜尋結構描述)。或者，展開 Schemas (結構描述)，在您要尋找的結構描述所在的登錄上，開啟內容功能表，然後選擇 Search Schemas in Registry (在登錄中搜尋結構描述)。在「結構描述搜尋」對話方塊中，開始鍵入要尋找的 EventBridge 綱要標題。此對話方塊會顯示包含相符項目的結構描述標題。

若要在對話方塊中顯示結構描述，請選取結構描述的標題。



## 為可用的結構描述產生程式碼

1. 在 AWS Explorer 中，展開 Schemas (結構描述)。
2. 展開登錄的名稱，內含您想產生程式碼的結構描述。
3. 在結構描述的標題上按一下滑鼠右鍵，然後選擇 Download code bindings (下載程式碼繫結)。
4. 在出現的精靈頁面中，選擇下列項目：
  - 結構描述的 Version (版本)
  - 程式碼繫結語言
  - 工作空間文件夾，供您在本機開發機器上存放產生的程式碼

## AWS IAM Access Analyzer

您可以使用中的 [存取分析器](#)，對在 [AWS CloudFormation 範本](#)、[Terraform 計劃](#)和原則文件中編寫的 [IAM原則執行AWS Identity and IAM Access Management \(IAM\)](#) 存取分析器JSON原則檢查。AWS Toolkit for Visual Studio Code

IAM存取分析器原則檢查包括原則驗證和自訂原則檢查。政策驗證有助於根據《AWS Identity and Access Management使用者指南》中的 [IAMJSON政策語言文法和IAM主題中的「AWS安全性最佳作法」](#) 中詳述的標準來驗證您IAM的政策。您的原則驗證發現項目包括安全性警告、錯誤、一般警告和原則建議。

您也可以根據您的安全性標準，執行自訂原則檢查以取得新的存取。每次針對新存取的自訂原則檢查都會收取費用。如需定價的詳細資訊，請參閱[AWS IAM存取分析器定價](#)網站。如需有關IAM存取分析器原則檢查的詳細資訊，請參閱AWS Identity and Access Management使用指南中的[檢查驗證原則](#)主題。

下列主題說明如何在中使用IAM存取分析器原則檢查 AWS Toolkit for Visual Studio Code。

主題

- [AWS IAM使用存取分析器](#)

## AWS IAM使用存取分析器

下列各節說明如何在中執行IAM原則驗證和自訂原則檢查 AWS Toolkit for Visual Studio Code。如需其他詳細資訊，請參閱 AWS Identity and Access Management 使用者指南中的下列主題：[IAM存取分析器原則驗證](#)和[IAM存取分析器自訂原則檢查](#)。

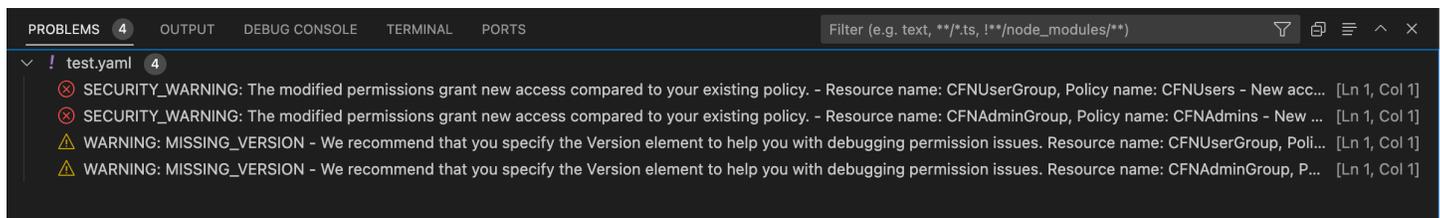
### 必要條件

您必須符合下列先決條件，才能從工具組IAM使用存取分析器原則檢查。

- 安裝 Python 或更新版本。
- 安裝 Python CLI 工具所需且在 [\[IAM原IAM則檢查\]](#) 視窗中指定的 [\[原則驗證程式\] AWS CloudFormation](#)或 [\[Terraform\]](#) 的原則驗證程式。IAM
- 設定您的 AWS 角色認證。

### IAM存取分析器原則檢查

您可以使用執行 AWS CloudFormation 範本、Terraform 計劃和JSON政策文件的原則檢查。AWS Toolkit for Visual Studio Code您可以在 VS 程式碼問題面板中檢視您的檢查發現項目。下圖顯示了 VS 代碼問題面板。



IAM訪問分析器提供 4 種類型的檢查：

- 驗證策略
- CheckAccessNotGranted
- CheckNoNewAccess
- CheckNoPublicAccess

以下各節說明如何執行每種類型的檢查。

#### Note

在執行任何類型的檢查之前，請先設定您的 AWS 角色認證。支援的檔案包括下列文件類型：  
AWS CloudFormation 範本、Terraform 計劃和JSON政策文件

檔案路徑參照通常由您的管理員或安全團隊提供，可以是系統檔案路徑或 Amazon S3 儲存貯體URI。若要使用 Amazon S3 儲存貯體URI，您目前的角色必須能夠存取 Amazon S3 儲存貯體。

每次自訂政策檢查都會收取費用。如需有關自訂原則檢查定價的詳細資訊，請參閱[AWS IAM存取分析器定價指南](#)。

## 執行驗證原則

驗證原則檢查 (也稱為原則驗證) 會根據政策文法和 AWS 最佳做法來驗證您的IAM政策。如需其他資訊，請參閱AWS Identity and Access Management使用者指南中[IAM主題中的IAMJSON原則語言文法和 AWS 安全性最佳作法](#)。

1. 在 VS Code 中，在 VS 程式碼編輯器中開啟包含 AWS IAM原則的支援檔案。
2. 若要開啟 [IAM存取分析器] 原則檢查，請按下、搜尋，然後按**CRTL+Shift+P**一下以開啟 [VS 程式碼] 編輯器中的 [IAM原則檢查] 窗格**IAM Policy Checks**，以開啟 VS 程式碼命令列。
3. 從 [IAM原則檢查] 窗格中，從下拉式功能表中選取您的文件類型。
4. 從 [驗證原則] 區段中，選擇 [執行原則驗證] 按鈕以執行驗證原則檢查。
5. 從 VS Code 的「問題」面板中，檢閱您的政策檢查結果。
6. 更新您的原則並重複此程序，重新執行驗證原則檢查，直到您的原則檢查發現項目不再顯示安全性警告或錯誤為止。

## 跑步 CheckAccessNotGranted

CheckAccessNotGranted 是一項自訂原則檢查，以確認您的政策不允許IAM執行特定動作。

**Note**

檔案路徑參照通常由您的管理員或安全團隊提供，可以是系統檔案路徑或 Amazon S3 儲存貯體URI。若要使用 Amazon S3 儲存貯體URI，您目前的角色必須能夠存取 Amazon S3 儲存貯體。至少必須指定一個動作或資源，並在下列範例之後建構檔案：

```
    {"actions": ["action1", "action2", "action3"], "resources":  
    ["resource1", "resource2", "resource3"]}
```

1. 在 VS Code 中，在 VS 程式碼編輯器中開啟包含 AWS IAM原則的支援檔案。
2. 若要開啟 [IAM存取分析器] 原則檢查，請按下、搜尋，然後按**CRTL+Shift+P**一下以開啟 [VS 程式碼] 編輯器中的 [IAM原則檢查] 窗格**IAM Policy Checks**，以開啟 VS 程式碼命令列。
3. 從 [IAM原則檢查] 窗格中，從下拉式功能表中選取您的文件類型。
4. 從 [自訂原則檢查] 區段中，選取**CheckAccessNotGranted**。
5. 在文字輸入欄位中，您可以輸入包含動作和資源的逗號分隔清單。ARNs至少必須提供一個動作或資源。
6. 選擇 [執行自訂原則檢查] 按鈕。
7. 從 VS Code 的「問題」面板中，檢閱您的政策檢查結果。自訂原則檢查會傳回PASS或FAIL結果。
8. 更新您的政策並重複此程序，重新執行檢查，直到 **CheckAccessNotGranted** 檢查返回PASS為止。

**跑步 CheckNoNewAccess**

**CheckNoNewAccess** 是一項自訂原則檢查，以驗證您的政策是否與參考原則相比授予新存取權。

1. 在 VS Code 中，在 VS 程式碼編輯器中開啟包含 AWS IAM原則的支援檔案。
2. 若要開啟 [IAM存取分析器] 原則檢查，請按下、搜尋，然後按**CRTL+Shift+P**一下以開啟 [VS 程式碼] 編輯器中的 [IAM原則檢查] 窗格**IAM Policy Checks**，以開啟 VS 程式碼命令列。
3. 從 [IAM原則檢查] 窗格中，從下拉式功能表中選取您的文件類型。
4. 從 [自訂原則檢查] 區段中，選取**CheckNoNewAccess**。

5. 輸入參考JSON政策文件。或者，您也可以提供參照JSON原則文件的檔案路徑。
6. 選取符合參考文件類型的參考原則類型。
7. 選擇 [執行自訂原則檢查] 按鈕。
8. 從 VS Code 的「問題」面板中，檢閱您的政策檢查結果。自訂原則檢查會傳回PASS或FAIL結果。
9. 更新您的政策並重複此程序，重新執行檢查，直到 CheckNoNewAccess 檢查返回PASS為止。

### 跑步 CheckNoPublicAccess

CheckNoPublicAccess 是一項自訂政策檢查，以驗證您的政策是否授與範本中支援資源類型的公用存取權。

如需有關支援資源類型的特定資訊，請參閱[cloudformation-iam-policy-validator](#)和[terraform-iam-policy-validator](#) GitHub 儲存庫。

1. 在 VS Code 中，在 VS 程式碼編輯器中開啟包含 AWS IAM原則的支援檔案。
2. 若要開啟 [IAM存取分析器] 原則檢查，請按下、搜尋，然後按**CTRL+Shift+P**一下以開啟 [VS 程式碼] 編輯器中的 [IAM原則檢查] 窗格**IAM Policy Checks**，以開啟 VS 程式碼命令列。
3. 從 [IAM原則檢查] 窗格中，從下拉式功能表中選取您的文件類型。
4. 從 [自訂原則檢查] 區段中，選取CheckNoPublicAccess。
5. 選擇 [執行自訂原則檢查] 按鈕。
6. 從 VS Code 的「問題」面板中，檢閱您的政策檢查結果。自訂原則檢查會傳回PASS或FAIL結果。
7. 更新您的政策並重複此程序，重新執行檢查，直到 CheckNoNewAccess 檢查返回PASS為止。

## 在 AWS Toolkit for Visual Studio Code 中使用 AWS IoT

AWS IoT在AWS Toolkit for Visual Studio Code允許您與AWS IoT服務，同時最大限度地減少 VS Code 中對工作流程的干擾。本使用者指南旨在協助您開始使用AWS IoT中的可用服務功能AWS Toolkit for Visual Studio Code。的其他相關資訊AWS IoT服務，請參閱開發人員指南[什麼是AWS IoT?](#)

### AWS IoT 先決條件

開始使用AWS IoT從 Toolkit for VS Code，確保你的AWS帳戶和 VS 代碼符合以下指南中的要求：

- 適用於AWS帳戶要求和AWS特定的使用者權限AWS IoT服務，請參閱[入門AWS IoT核心開發人員指南](#)。
- 如需 VS Code 特定需求的工具組，請參閱[為 VS 程式碼設定工具組](#)使用者指南。

## AWS IoT實物

AWS IoT將裝置連線至AWS雲端服務和資源。您可以將設備連接到AWS IoT通過使用稱為的對象事。物件是特定裝置或邏輯實體的代表。它可以是實體裝置或感應器 (例如燈泡或牆上的開關)。的其他相關資訊AWS IoT事情，請參閱開發人員指南[利用 管理裝置AWS IoT](#)。

### 管理AWS IoT事

Toolkit for VS Code 具有幾個功能，可以使您的AWS IoT事情管理更有效率。這些是您可以使用 VS Code 工具包來管理您的AWS IoT實物：

- [Create a thing](#)
- [Attach a certificate to a thing](#)
- [Detach a certificate from a thing](#)
- [Delete a thing](#)

### 如何建立物件

1. 來自：AWS檔案總管，展開物聯網 (IoT)服務標題，以及內容選取 (按一下滑鼠右鍵)實物。
2. 選擇建立實物從上下文菜單中打開一個對話框。
3. 依照提示，將 IoT 物件的名稱輸入實物名稱欄位。
4. 完成後，a實物圖示後面跟著您指定的名稱將在物件區段。

### 將憑證連接到物件

1. 來自：AWS檔案總管，展開物聯網 (IoT)服務部分。
2. 在下方實物子區段中，找到事情您要附加憑證的位置。
3. 內容選取 (按一下右鍵)事情並選擇連接憑證從上下文菜單中，打開帶有證書列表的輸入選擇器。
4. 從清單中選擇憑證 ID對應於您要附加到您的事物的證書。
5. 完成此操作後，您的證書可以在AWS資源管理器，作為您附加到的東西的項目。

## 從物件分離出憑證

1. 來自：AWS檔案總管，展開物聯網 (IoT)服務部分
2. 在 中實物子區段中，找到事情您要分離的憑證。
3. 內容選取 (按一下右鍵)事情並選擇分離憑證從上下文菜單。
4. 完成後，卸離的憑證將不再顯示在AWS資源管理器，但它仍然可以從憑證子部份。

## 刪除物件

1. 來自：AWS檔案總管，展開物聯網 (IoT)服務部分。
2. 在 中實物子區段中，找到事情您要刪除。
3. 內容選取 (按一下滑鼠右鍵) 物件，然後選擇刪除物件從上下文菜單中刪除它。
4. 完成後，刪除的物件將無法再從實物子部份。

### Note

請注意：您只能刪除沒有附加憑證的項目。

## AWS IoT 憑證

憑證是建立安全連線的常用方式AWS IoT服務和設備。X.509 憑證為數位憑證，其依據 X.509 公有金鑰基礎設施標準，將公有金鑰與憑證內含的身分建立關聯。的其他相關資訊AWS IoT憑證，請參閱開發人員指南[IoT 認證](#)。

## 管理憑證

VS Code 工具組提供多種管理方式AWS IoT憑證，直接從AWSExplorer。

- [Create a certificate](#)
- [Change a certificate status](#)
- [Attach a policy to a certificate](#)
- [Delete a certificate](#)

## 建立AWS IoT證書

X.509 憑證可用來連接您的執行個體AWS IoT。

1. 來自：AWS檔案總管，展開物聯網 (IoT)服務區段，以及內容選取 (按一下滑鼠右鍵)憑證。
2. 選擇建立憑證從上下文菜單中打開一個對話框。
3. 在您的本機檔案系統中選取一個目錄，以儲存您的 RSA key pair 和 X.509 憑證。

### Note

- 預設檔案名稱包含憑證 ID 做為前置詞。
- 只有 X.509 憑證會儲存在您的AWS帳戶，透過AWS IoT服務。
- 您的 RSA key pair 只能發出一次，並在出現提示時將其保存到文件系統中的安全位置。
- 如果此時無法將憑證或 key pair 儲存至檔案系統，則AWS工具包從您的證書中刪除AWS帳戶。

## 修改憑證狀態

個別憑證的狀態會顯示在其 ID 旁AWS檔案總管，可以設定為：作用中、非作用中或已撤銷。

### Note

- 您的憑證需要活躍狀態，然後才能使用它將設備連接到您的AWS IoT服務。
- 同時不活動的無論憑證先前已停用或預設為非作用中，都可以啟動憑證。
- 已獲得的證書撤銷無法重新啟用。

1. 來自：AWS檔案總管，展開物聯網 (IoT)服務部分。
  2. 在中憑證子區段中，找出您想要修改的憑證。
  3. 前後關聯選取 (按一下滑鼠右鍵) 憑證以開啟內容功能表，其中顯示該憑證可用的狀態變更選項。
- 如果憑證的狀態為不活動的，選擇啟動將狀態變更為活躍。
  - 如果憑證的狀態為活躍，選擇使無效將狀態變更為不活動的。
  - 如果憑證具有活躍或者不活動的狀態，選擇撤銷將狀態變更為撤銷。

**Note**

如果您在物件中顯示時選取附加至物件的憑證，也可以使用這些狀態變更動作中的每個動作實物子部份。

### 將 IoT 原則附加至憑證

1. 來自：AWS檔案總管，展開物聯網 (IoT)服務部分。
2. 在 中憑證子區段中，找出您想要修改的憑證。
3. 內容選取 (按一下滑鼠右鍵) 憑證，然後選擇連接政策從內容功能表中，開啟包含可用原則清單的輸入選取器。
4. 選擇您想要連接到憑證的政策。
5. 完成此操作後，您選取的策略將作為子功能表項目新增至憑證。

### 從憑證中斷連結 IoT 原則

1. 來自：AWS檔案總管，展開物聯網 (IoT)服務部分。
2. 在 中憑證子區段中，找出您想要修改的憑證。
3. 展開憑證，然後找出您想要分離的政策。
4. 內容選取 (按一下滑鼠右鍵) 政策，然後選擇分離從內容功能表。
5. 完成此操作後，策略將不再是可從您的證書訪問的項目，但可以從政策子部份。

### 刪除憑證

1. 來自：AWS檔案總管，展開物聯網 (IoT)服務標題。
2. 在 中憑證子區段中，找出您想要刪除的憑證。
3. 內容選取 (按一下滑鼠右鍵) 憑證，然後選擇刪除憑證從內容功能表。

**Note**

如果憑證已連接至物件或具有作用中狀態，則無法刪除憑證。您可以刪除已附加原則的憑證。

## AWS IoT 政策：

AWS IoT核心政策是透過 JSON 文件定義，每個政策都包含一個或多個政策陳述式。原則定義方式 AWS IoT、AWS，您的裝置可以彼此互動。如需如何建立政策文件的詳細資訊，請參閱開發人員指南 [IoT 政策](#)。

### Note

具名政策會經過版本控制，因此您可以還原。在「中」AWS資源管理器中，您的 IoT 政策列在政策小節，在 IoT 服務中。您可以透過展開原則來檢視原則版本。預設版本以星號表示。

## 管理 政策

Toolkit for VS Code 您提供了數種方式來管理您的AWS IoT服務政策。您可以透過以下方式直接管理或修改您的政策AWSVS 代碼中的資源管理器：

- [Create a policy](#)
- [Upload a new policy version](#)
- [Edit a policy version](#)
- [Change the policy version default](#)
- [Change the policy version default](#)

## 如要建立 AWS IoT 政策

### Note

您可以從建立新政策AWS資源管理器，但定義策略的 JSON 文檔必須已經存在於您的文件系統中。

1. 來自：AWS檔案總管，展開物聯網 (IoT)服務部分。
2. 內容選取 (按一下右鍵)政策子節和選擇從文件建立策略(您要自訂簡訊嗎?)政策名稱輸入欄位。
3. 輸入名稱並依照提示開啟對話方塊，要求您從檔案系統中選取 JSON 文件。
4. 選擇包含您的策略定義的 JSON 文件，該策略將在AWS資源管理器完成時。

## 上傳新AWS IoT政策版本

您可以將 JSON 文件上載至原則來建立新版本的原則。

### Note

新的 JSON 文件必須出現在您的檔案系統上，才能使用AWSExplorer。

1. 來自：AWS檔案總管，展開物聯網 (IoT)服務部分。
2. 展開政策子節以檢視您的AWS IoT政策
3. 內容-選取 (以滑鼠右鍵按一下) 您要更新的原則，然後選擇從文件建立新版本。
4. 當對話方塊開啟時，請選擇包含原則定義更新的 JSON 檔案。
5. 新版本將可從您的政策存取AWSExplorer。

## 編輯AWS IoT政策版本

您可以使用「VS 程式碼」開啟和編輯原則文件。完成編輯文件後，您可以將它儲存到您的檔案系統。然後，您可以將其上傳到您的AWS IoT服務從AWSExplorer。

1. 來自：AWS檔案總管，展開物聯網 (IoT)服務部分。
2. 展開政策分段，然後找出您想要更新的政策。從文件建立策略開啟政策名稱輸入欄位。
3. 展開您要更新的原則，然後選取 (以滑鼠右鍵按一下) 您要編輯的原則版本。
4. 選擇檢視從上下文菜單中打開 VS 代碼中的策略版本
5. 當原則文件開啟時，請進行並儲存您想要的變更。

### Note

此時，您對策略所做的變更只會儲存到您的本機檔案系統。更新版本並使用AWS檔案總管，重複[Upload a new policy version](#)程序。

## 若要選取新的原則版本預設值

1. 來自：AWS檔案總管，展開物聯網 (IoT)服務部分。

2. 展開政策分段，然後找出您想要更新的政策。
3. 展開您要更新的原則，然後進行內容選取 (以滑鼠右鍵按一下) 您要設定的原則版本，然後選擇設定為預設。
4. 完成此操作後，您選擇的新默認版本將在其旁邊顯示一個星號。

## 刪除 政策

### Note

刪除策略或策略版本之前，需要符合一些條件。

- 如果政策已連接至憑證，則無法刪除政策。
- 如果原則有任何非預設版本，則無法刪除該政策。
- 除非選取新的預設版本，或刪除整個政策，否則您無法刪除政策的預設版本。
- 刪除整個策略之前，必須先刪除該策略的所有非預設版本。

1. 來自：AWS檔案總管，展開物聯網 (IoT)服務部分。
2. 展開政策分段，然後找出您想要更新的政策。
3. 展開您要更新的原則，然後選取 (以滑鼠右鍵按一下) 您要刪除的原則版本，然後選擇刪除。
4. 刪除版本後，該版本將不再可見。
5. 當原則的唯一剩餘版本為預設值時，您可以在上下文中選取 (按一下滑鼠右鍵) 父策略，然後選擇刪除刪除它。

## 使用 AWS Lambda 函數

AWS Toolkit for Visual Studio Code 支援使用 [AWS Lambda](#) 函數。通過 Toolkit for VS Code 組，您可以為 Lambda 函數編寫程式碼，這些函數是[無伺服器應用程式](#)。此外，您可以在本機或 AWS 上叫用 Lambda 函式。

Lambda 是一種完整受管運算服務，可執程式碼以回應自訂程式碼所產生的事件，或從 Amazon Simple Storage Service (Amazon S3)、Amazon DynamoDB、Amazon Kinesis、Amazon Simple Notification Service (Amazon SNS) 和 Amazon Cognito 等各種 AWS 服務執程式碼。

## 主題

- [與遠端 Lambda 函數互動](#)

## 與遠端 Lambda 函數互動

使用 Toolkit for VS Code，您可以與[AWS Lambda](#)函數，如本主題稍後所述。

如需有關 Lambda 的詳細資訊，請參閱 [AWS Lambda 開發人員指南](#)。

### Note

如果您已使用 AWS Management Console 或其他方式建立 Lambda 函式，您可以從工具組叫用這些函式。若要建立新函數（使用 VS 程式碼），您可以將其部署到 AWS Lambda，您必須先[建立無伺服器應用程式](#)。

### 主題

- [先決條件](#)
- [呼叫 Lambda 函數](#)
- [刪除 Lambda 函數](#)
- [匯入 Lambda 函數](#)
- [上傳 Lambda 函數](#)

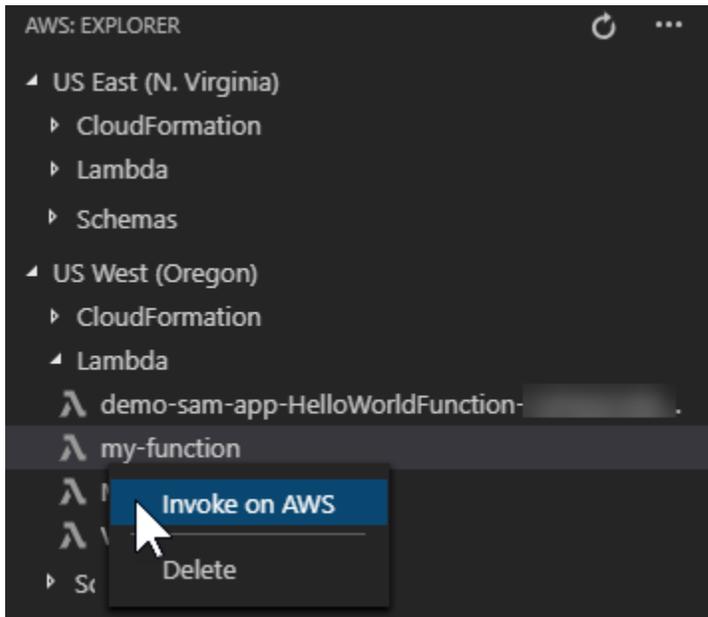
### 先決條件

- 請確定您的系統符合[安裝 VS 程 Toolkit for VS Code](#)。
- 確認您在 [身分驗證與存取](#) 中設定的登入資料包括 AWS Lambda 服務的適合讀取/寫入存取。如果在 AWS Explorer 的 Lambda 下方，您看到類似「Error loading Lambda resources」（載入 Lambda 資源時發生錯誤）的訊息，請檢查與這些憑證連接的許可。您對這些許可所做的變更需要幾分鐘的時間才會對 AWS 探險者在 VS 代碼中。

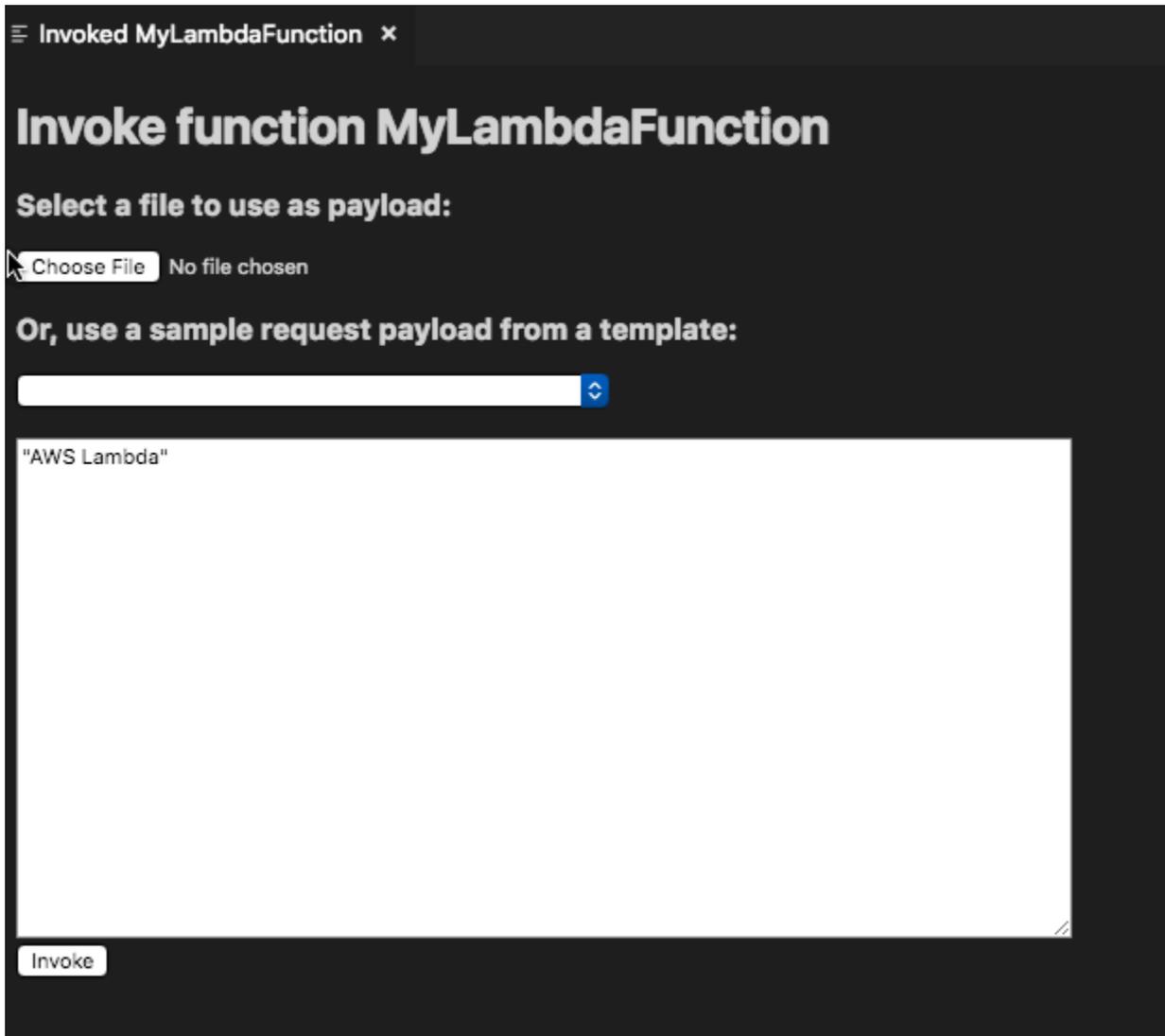
### 呼叫 Lambda 函數

您可以使用 Lambda 函數 AWS 從 Toolkit for VS Code 中。

1. 在 AWS Explorer 中，選擇您想要叫用的 Lambda 函式名稱，然後開啟其內容選單。



2. 選擇 Invoke on AWS (在 AWS 叫用)。
3. 在開啟的調用窗口中，輸入 Lambda 函數所需的輸入。例如，Lambda 函數可能需要字串作為輸入，如文字方塊中所示。



您將看到 Lambda 函數的輸出，就如對使用 VS Code 的任何其他專案一樣。

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL AWS Lambda
Loading response...
Invocation result for arn:aws:lambda:us-west-2:282147211633:function:MyLambdaFunction
Logs:
START RequestId: 04c5f09a-508f-4251-bc6d-663023153d7c Version: $LATEST
Input: AWS LambdaEND RequestId: 04c5f09a-508f-4251-bc6d-663023153d7c
REPORT RequestId: 04c5f09a-508f-4251-bc6d-663023153d7c Duration: 253.98 ms Billed Duration: 300 ms Memory Size: 128 MB
Max Memory Used: 86 MB

Payload:
"Hello, AWS Lambda!"
```

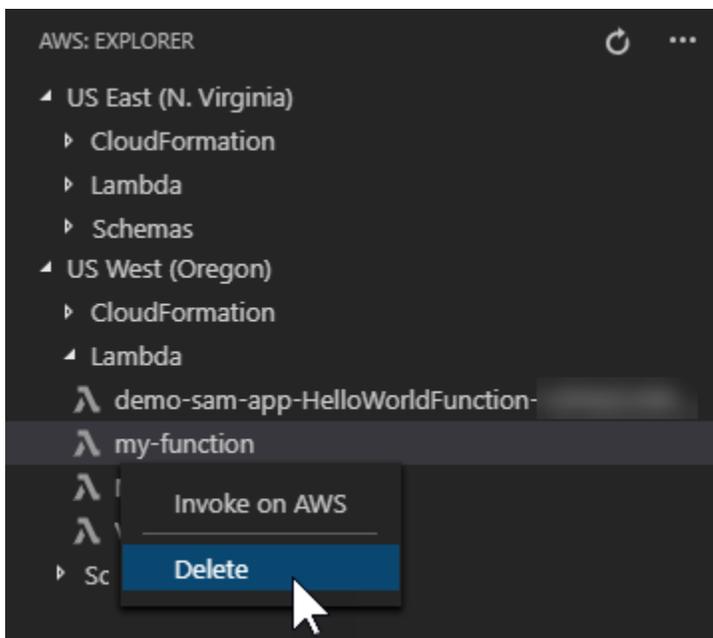
## 刪除 Lambda 函數

您也可以使用同一個內容選單刪除 Lambda 函式。

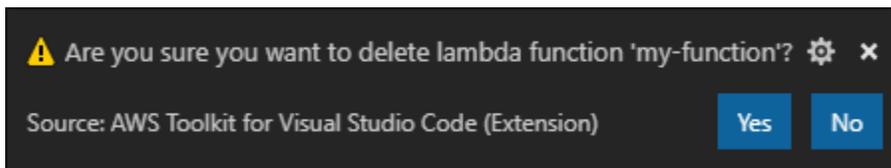
### ⚠ Warning

請勿透過此程序刪除與 [AWS CloudFormation](#) 建立關聯的 Lambda 函式(例如在本指南前述步驟中[建立無伺服器應用程式](#)時建立的 Lambda 函式)。這些函式必須透過 AWS CloudFormation 堆疊刪除。

1. 在 AWS Explorer 中，選擇您要刪除的 Lambda 函式名稱，然後開啟其內容選單。



2. 選擇 Delete (刪除)。
3. 在出現的訊息中，選擇 Yes (是)，以確認刪除。



刪除函數後，該函數不會再列於 AWS Explorer 之中。

## 匯入 Lambda 函數

您可以將遠端 Lambda 功能中的代碼導入 VS 代碼工作區以進行編輯和調試。

### Note

該工具組僅支持使用支持的 Node.js 和 Python 執行時間匯入 Lambda 函數。

1. 在 AWS 探險者中，選擇您想要匯入的 Lambda 函數名稱，然後開啟其內容選單。
2. 選擇匯入
3. 選擇要將 Lambda 代碼導入到的文件夾。當前工作空間外的文件夾將被添加至當前的工作空間。
4. 下載後，Toolkit 會將代碼添加到您的工作區，並打開包含 Lambda 處理程序代碼的文件。該工具包還創建一個啟動組態，它出現在 VS 代碼運行面板中，因此您可以使用 AWS Serverless Application Model。如需有關使用 AWS SAM 的詳細資訊，請參閱 [the section called “從範本 \(本機\) 執行和偵錯無伺服器應用程式”](#)。

## 上傳 Lambda 函數

您可以使用本機程式碼更新現有的 Lambda 函式。以這種方式更新程式碼時，並不會使用 AWS SAM CLI 進行部署，也不會建立 AWS CloudFormation 堆疊。此功能可以使用 Lambda 支援的任何執行時間來上傳 Lambda 函式。

### Warning

工具包無法檢查您的代碼是否正常工作。在更新生產 Lambda 函數之前，請確保代碼正常工作。

1. 在 AWS 探險者中，選擇您想要匯入的 Lambda 函數名稱，然後開啟其內容選單。
2. 選擇 Upload Lambda... (上傳 Lambda...)
3. 從三個選項中選擇要用於上傳 Lambda 函式的方式。選項包括：

上傳預先製作的 .zip 存檔

- 選擇 Zip 檔案從快速選取菜單中。

- 從檔案系統選擇 .zip 檔案，然後使用模式對話方塊確認上傳。這個動作會直接上傳 .zip 文件，並在部署後立即更新 Lambda。

### 直接上傳目錄

- 選擇目錄從快速選取菜單中。
- 從檔案系統選擇目錄。
- 選擇否在系統提示建置目錄時，然後使用模式對話方塊確認上傳。這個動作會直接上傳目錄，並在部署後立即更新 Lambda。

### 建置並上傳目錄

#### Note

這需要AWS SAMCLI。

- 選擇目錄從快速選取菜單中。
- 從檔案系統選擇目錄。
- 選擇是在系統提示建置目錄時，然後使用模式對話方塊確認上傳。這個動作會使用 AWS SAM CLI 的 `sam build` 命令在目錄中建置程式碼，並在部署後立即更新 Lambda。

#### Note

如果工具包在上傳之前無法檢測到匹配的處理程序，則會警告您。如果您想要更改綁定至 Lambda 函數的處理程式，您可以通過AWS Management Console或AWS CLI。

## Toolkit for VS Code 中的 Amazon Redshift

Amazon Redshift 是一種在雲端中完全受管的 PB 級資料倉儲服務。如需有關 Amazon Redshift 服務的詳細資訊，請參閱 [Amazon Redshift 使用者指南目錄](#)。

以下主題介紹了如何使用 Amazon Redshift 從 AWS Toolkit for Visual Studio Code

### 主題

- [從 VS 代碼工具包中使用 Amazon Redshift](#)

## 從 VS 代碼工具包中使用 Amazon Redshift

以下各節說明如何開始使用 Amazon Redshift 從AWS Toolkit for Visual Studio Code.

如需有關 Amazon Redshift 服務的詳細資訊，請參閱 [Amazon Redshift](#) 使用者指南主題。

### 入門

在您開始使用 Amazon Redshift 之前AWS Toolkit for Visual Studio Code，必須符合下列要求。

1. 您已從工具包連接到您的AWS帳戶。如需有關從工具組連線到AWS帳戶的其他資訊，請參閱本使用者指南中的「[連線至](#)」AWS 主題。
2. 您已建立佈建或無伺服器資料倉儲。

如果您尚未建立 Amazon Redshift 無伺服器或 Amazon Redshift 佈建的叢集，則下列程序說明如何透過主控台使用範例資料集建立資料倉儲。AWS

### 建立佈建的資料倉儲

如需有關建立 Amazon Redshift 佈建的叢集資料倉儲的其他詳細資訊，請參閱 [Amazon Redshift 入門](#) 使用者指南中的[建立範例 Amazon Redshift 叢集](#)主題。

1. 從您偏好的網際網路瀏覽器登入AWS管理主控台，然後開啟 Amazon Redshift 主控台，網址為 <https://console.aws.amazon.com/redshift/>。
2. 從 Amazon Redshift 主控台中，選擇佈建的叢集儀表板。
3. 從 [佈建的叢集] 儀表板中，選擇 [建立叢集] 按鈕以開啟 [建立叢集] 窗格。
4. 完成 [叢集配置] 區段中的必要欄位。
5. 在 [範例資料] 區段中，選取 [載入範例資料] 方塊，將範例資料集載入含**public**結構描述**Dev**的預設資料庫。
6. 在 [資料庫設定] 區段中，輸入 Admin 使用者名稱和 Admin 使用者密碼欄位的值。
7. 選擇 [建立叢集] 以建立佈建的資料倉儲。

## 建立無伺服器資料倉儲

如需建立 Amazon Redshift 無伺服器資料倉儲的其他詳細資訊，請參閱 Amazon Redshift 入門使用者指南中的〈[使用亞 Amazon Redshift 無伺服器建立資料倉儲](#)〉一節。

1. 從您偏好的網際網路瀏覽器登入AWS管理主控台，然後開啟 Amazon Redshift 主控台，網址為 <https://console.aws.amazon.com/redshift/>。
2. 在 Amazon Redshift 主控台中，選擇試用 Amazon Redshift 無伺服器按鈕以開啟 Amazon Redshift 無伺服器開始使用窗格。
3. 在「組態」區段中，選擇「使用預設設定」放射狀。
4. 在「開始使用 Amazon Redshift 無伺服器」窗格底部，選擇「儲存組態」以建立具有預設工作群組、命名空間、登入資料和加密設定的無伺服器資料倉儲。

## 從工具組連線至資料倉儲

有 3 種方法可以從工具包連接到數據庫：

- 資料庫使用者名稱和密碼
- AWS Secrets Manager
- 暫時性憑證

若要從 Toolkit 連線到位於現有佈建叢集或無伺服器資料倉儲上的資料庫，請完成以下步驟。

### Important

如果您已完成本使用者指南主題的「必要條件」一節中的步驟，且您的資料倉儲在 Toolkit 總管中看不到，請確定您在總管中使用的是正確的AWS區域。

## 使用資料庫使用者名稱和密碼方法連線至資料倉儲

1. 從工具組總管中，展開資料倉儲所在的AWS 區域位置。
2. 展開 Redshift 並選擇您的資料倉儲，以在 VS Code 中開啟 [選取連線類型] 對話方塊。
3. 從 [選取連線類型] 對話方塊中，選擇 [資料庫使用者名稱和密碼]，並提供每個提示所需的資訊。
4. 當 Toolkit 連線至資料倉儲且程序完成時，您可用的資料庫、資料表和結構描述會顯示在 Toolkit 總管中。

## 使用以下方式連線到資料倉儲 AWS Secrets Manager

### Note

此程序需要AWS祕密管理員資料庫密碼才能完成。如需有關如何設定資料庫密碼的指示，請參閱 [Sec AWSrets Manager 使用者指南](#) 中的 [建立AWS Secrets Manager資料庫密碼](#)。

1. 從工具組總管中，展開資料倉儲所在的AWS 區域位置。
2. 展開 Redshift 並選擇您的資料倉儲，以在 VS Code 中開啟 [選取連線類型] 對話方塊。
3. 從 [選取連線類型] 對話方塊中，選擇 Secrets Manager 並提供每個提示所需的資訊。
4. 當 Toolkit 連線至資料倉儲且程序完成時，您可用的資料庫、資料表和結構描述會顯示在 Toolkit 總管中。

### 使用臨時登入資料連線至資料倉儲

1. 從工具組總管中，展開資料倉儲所在的AWS區域。
2. 展開 Redshift 並選擇您的資料倉儲，以在 VS Code 中開啟 [選取連線類型] 對話方塊。
3. 從「選取連線類型」對話方塊中，選擇「暫時認證」，然後提供每個提示所需的資訊。
4. 當 Toolkit 連線至資料倉儲且程序完成時，您可用的資料庫、資料表和結構描述會顯示在 Toolkit 總管中。

### 編輯資料倉儲的連線

您可以編輯與資料倉儲的連線，以變更要連線到的資料庫。

1. 從工具組總管中，展開資料倉儲所在的AWS 區域位置。
2. 展開 Redshift，以滑鼠右鍵按一下要連線到的資料倉儲，選擇 [編輯連線]，然後提供您要連線到的資料庫名稱。
3. 當 Toolkit 連線至資料倉儲且程序完成時，您可用的資料庫、資料表和結構描述會顯示在 Toolkit 總管中。

### 刪除資料倉儲的連線

1. 從工具組總管中，展開資料倉儲所在的AWS 區域位置。

2. 展開 Redshift，以滑鼠右鍵按一下要刪除之連線的資料倉儲，然後選擇 [刪除連線]。這樣做會從 Toolkit 總管中移除可用的資料庫、資料表和結構描述。
3. 若要重新連線至資料倉儲，請選擇按一下以連線，並提供每個提示所需的資訊。根據預設，重新連線會使用先前的驗證方法來連線至資料倉儲。若要使用其他方法，請在對話方塊中選擇返回箭頭，直到出現驗證提示為止。

## 執行 SQL 敘述句

下列程序說明如何從資料庫中建立及執行 SQL 陳述式AWS Toolkit for Visual Studio Code。

### Note

若要完成下列每個程序中的步驟，您必須先完成本使用指南主題中的「從工具組連線到資料倉儲」一節。

1. 從工具組總管展開 Redshift，然後展開包含您要查詢之資料庫的資料倉儲。
2. 選擇 [建立筆記本] 以指定要在本機儲存筆記本的檔案名稱和位置，然後選擇 [確定]，在 VS Code 編輯器中開啟筆記本。
3. 在 VS 程式碼編輯器中，輸入您要儲存在此筆記本中的 SQL 陳述式。
4. 選擇「全部執行」按鈕，執行您輸入的 SQL 敘述句。
5. SQL 敘述句的輸出會顯示在您輸入的敘述句下方。

## 將降價添加到筆記本

1. 在 VS Code 編輯器的筆記本中，選擇「降價」按鈕，將「降價」儲存格新增至您的筆記本。
2. 將降價輸入到提供的單元格中。
3. 您可以使用位於「降價」儲存格右上角的編輯器工具來編輯「降價」儲存格。

## 將程式碼新增至筆記本

1. 從 VS Code 編輯器的筆記本中，選擇 [程式碼] 按鈕，將 [程式碼] 儲存格新增至您的筆記本。
2. 將您的代碼輸入到提供的單元格中。
3. 您可以選擇在「程式碼」儲存格右上角的儲存格編輯器工具中選取適當的按鈕，在「程式碼」儲存格的上方或下方執程式碼。

# 與 Amazon S3 合作

亞馬遜 Simple Storage Service (Amazon S3) 是一種可擴展的資料儲存服務。AWS Toolkit for Visual Studio Code 可讓您直接從 VS 程式碼管理您的 Amazon S3 物件和資源。

如需有關 Amazon S3 服務的詳細資訊，請參閱 [Amazon S3](#) 使用者指南。

下列主題說明如何使用來自的 Amazon S3 物件和資源AWS Toolkit for Visual Studio Code。

## 主題

- [使用 Amazon S3 資源](#)
- [使用 Amazon S3 物件](#)

## 使用 Amazon S3 資源

您可以從中使用 Amazon S3 AWS Toolkit for Visual Studio Code 來檢視、管理和編輯 Amazon S3 儲存貯體和其他資源。

以下各節說明如何使用 Amazon S3 資源，從 AWS Toolkit for Visual Studio Code 如需使用 Amazon S3 物件 (例如資料夾和檔案) 的相關資訊 AWS Toolkit for Visual Studio Code，請參閱本使用者指南中的使用 [S3 物件](#) 主題。

## 建立 Amazon S3 儲存貯體

1. 從工具包資源管理器中，打開 S3 服務的上下文 (右鍵單擊) 菜單，然後選擇創建存儲桶...。或者，選擇「建立值區」圖示以開啟「建立值區」對話方塊。
2. 在 Bucket Name (儲存貯體名稱) 欄位中，輸入有效的儲存貯體名稱。

按 Enter 以建立值區並關閉對話方塊。然後，您的新儲存貯體會顯示在工具組中的 S3 服務下。

### Note

由於 Amazon S3 允許將儲存貯體用作可公開存取URL的儲存貯體，因此您選擇的儲存貯體名稱必須是全域唯一的。如果其他帳戶已使用您要使用的名稱建立值區，則必須使用不同的名稱。

如果您無法建立新值區，請檢查 [輸出] 索引標籤中的 [AWS 工具組記錄]。如果您嘗試使用無效的值區名稱，就會發生BucketAlreadyExists錯誤。

如需詳細資訊，請參閱 Amazon Simple Storage Service 使用者指南中的 [儲存貯體限制與局限](#)。

## 在 Amazon S3 儲存貯體中新增資料夾

您可以將物件分組到資料夾中，以組織 S3 儲存貯體的內容。您也可以在資料夾內建立資料夾。

1. 從工具組總管中，展開 S3 服務以檢視 S3 資源的清單。
2. 選擇「建立資料夾」圖示以開啟「建立資料夾」對話方塊。或者，開啟值區或資料夾的內容 (按一下滑鼠右鍵) 功能表，然後選擇「建立資料夾」。
3. 在「資料夾名稱」欄位中輸入值，然後按 Enter 以建立資料夾並關閉對話方塊。您的新資料夾會顯示在工具組功能表中對應的 S3 資源下。

## 刪除 Amazon S3 儲存貯體

刪除 S3 儲存貯體時，也會刪除其中包含的資料夾和物件。因此，當您嘗試刪除值區時，系統會要求您確認是否要刪除該值區。

1. 從工具組主功能表展開 Amazon S3 服務以檢視您的 S3 資源清單。
2. 開啟儲存貯體或資料夾的內容 (按一下滑鼠右鍵) 功能表，然後選擇刪除 S3 儲存貯體。
3. 出現提示時，請在文字欄位中輸入值區的名稱，然後按 Enter 鍵刪除值區並關閉確認提示。

### Note

如果您的值區包含物件，則在刪除之前會清空該值區。如果您一次嘗試刪除大量資源或物件，可能需要一些時間才能刪除這些資源或物件。刪除後，您會收到通知，指出已成功刪除它們。

## 使用 Amazon S3 物件

存放在 S3 資源儲存貯體中的檔案、資料夾和任何其他資料稱為 S3 物件。

下列各節說明如何使用來自的 Amazon S3 物件AWS Toolkit for Visual Studio Code。如需使用 Amazon S3 資源 (例如 S3 儲存貯體) 的詳細資訊AWS Toolkit for Visual Studio Code，請參閱本使用者指南中的使用 [S3 資源](#)主題。

## 對象分頁

如果您正在處理大量 Amazon S3 物件和資料夾，分頁可讓您指定要在頁面上顯示的項目數量。

1. 導航到 VS 代碼活動欄，然後選擇擴展。
2. 從AWS工具組擴充功能中，選擇設定圖示，然後選擇擴充功能設定。
3. 在「設定」頁面上，向下捲動至「AWS> S3：每頁最多項目數」設定。
4. 在顯示「載入更多」之前，將預設值變更為要顯示的 S3 項目數目。

### Note

有效值包括 3 到 1000 之間的任何數字。此設定僅適用於一次顯示的物件或資料夾數量。系統會立即顯示您建立的所有值區。每個 AWS 帳戶預設最多可以建立 100 個儲存貯體。

5. 關閉「設定」頁面以確認您的變更。

您也可以選擇「設定」頁面右上角的「開啟設定」(JSON) 圖示，來更新 JSON 格式檔案中的設定。

## 上傳和下載 Amazon S3 對象

您可以將本機存放的檔案上傳到 Amazon S3 儲存貯體，或從 AWS Toolkit for Visual Studio Code

### 使用工具組上傳檔案

1. 從工具組總管中，展開 Amazon S3 服務以檢視 S3 資源的清單。
2. 選擇位於值區或資料夾旁邊的「上傳檔案」圖示，開啟「上傳檔案」對話方塊。或者，您可以打開上下文 (右鍵單擊) 菜單，然後選擇上傳文件。

### Note

若要將檔案上傳至物件的父資料夾或資源，請開啟任何 S3 物件的前後關聯 (按一下滑鼠右鍵) 功能表，然後選擇「上傳至父系」。

3. 使用系統的檔案管理員選取檔案，然後選擇「上傳檔案」以關閉對話方塊並上傳檔案。

### 使用命令面板上傳檔案

您可以使用 Toolkit 介面或命令調色盤將檔案上傳至值區。

1. 若要選取要上傳的檔案，請在 VS Code 中選擇該檔案的索引標籤。
2. 按 Ctrl+ 轉移 +P 以顯示「指令選項板」。
3. 在「指令選項板」中，輸入片語upload file以顯示建議指令的清單。
4. 選擇 AWS：上傳檔案命令以開啟 AWS：上傳檔案對話方塊。
5. 出現提示時，選擇您要上傳的檔案，然後選擇您要上傳該檔案的目標儲存貯體。
6. 確認您的上傳以關閉對話方塊並開始上傳程序。上載完成時，物件會顯示在工具組選單中，其中包含物件大小、上次修改日期和路徑的中繼資料。

## 下載 Amazon S3 物件

1. 從工具組總管中，展開 S3 服務。
2. 在值區或資料夾中，開啟您要下載之物件的內容 (按一下滑鼠右鍵) 選單。然後，選擇「下載為」以開啟「下載為」對話方塊。或者，您也可以選擇物件附近的「下載為」圖示。
3. 使用系統的檔案管理員，選擇目標資料夾，輸入檔案名稱，然後選擇 [下載] 以關閉對話方塊並開始下載。

## 編輯遠端物件

您可以使AWS Toolkit for Visual Studio Code用編輯存放在遠端 Amazon S3 資源中的 Amazon S3 物件。

1. 從工具組總管中，展開 S3 服務。
2. 展開包含您要編輯之檔案的 S3 資源。
3. 若要編輯檔案，請選擇鉛筆圖示 (編輯檔案)。
4. 若要編輯以唯讀模式開啟的檔案，請在 VS Code 編輯器中檢視檔案，然後選擇位於 UI 右上角的鉛筆圖示。

### Note

- 如果您重新啟動或結束 VS 程式碼，您的 IDE 會中斷與 S3 資源的連線。如果在您中斷連線時編輯任何遠端 S3 檔案，編輯會停止。您必須重新啟動 VS Code 並重新開啟編輯索引標籤，才能繼續編輯。
- [編輯檔案] 按鈕位於 UI 的右上角。只有當您在 VS Code 編輯器中主動檢視唯讀檔案時，才會顯示此功能。

- 非文字檔案無法以唯讀模式開啟。它們總是以編輯模式打開。
- 您不能從僅編輯模式切換回只讀模式，只能反過來。

## 複製 Amazon S3 物件的路徑

下列程序說明如何從中複製 Amazon S3 物件的路徑AWS Toolkit for Visual Studio Code。

1. 從工具組總管中，展開 S3 服務。
2. 展開包含您要複製路徑之物件的資源值區。
3. 開啟您要複製路徑之物件的前後關聯 (按一下滑鼠右鍵) 選單，然後選擇「複製路徑」，將物件路徑複製到本機剪貼簿。

## 為 Amazon S3 物件產生預先簽章的 URL

您可以透過預先簽署的 URL 功能授予下載時間限制許可，與其他人共用私有 Amazon S3 物件。如需詳細資訊，請參閱[使用預先簽署的 URL 共用物件](#)。

1. 從工具組總管中，展開 S3 服務。
2. 在值區或資料夾中，開啟您要共用之物件的內容 (按一下滑鼠右鍵) 選單。然後，選擇「產生預先簽署的 URL」以開啟「命令」面板。
3. 在指令調色盤中，輸入 URL 可用於存取物件的分鐘數。然後，選擇 Enter 確認並關閉對話框。
4. 產生預先簽署的 URL 之後，「VS 程式碼狀態列」會顯示已複製到本機剪貼簿之物件的預先簽署 URL。

## 刪除 Amazon S3 物件

如果物件位於未建立版本控制的值區中，您可以永久刪除該物件。對於已啟用版本控制的值區，刪除請求不會永久刪除該物件。相反地，Amazon S3 會在儲存貯體中插入刪除標記。如需詳細資訊，請參閱[刪除物件版本](#)。

1. 從工具組總管中，展開 S3 服務以檢視 S3 資源的清單。
2. 開啟您要刪除之物件的前後關聯 (按一下滑鼠右鍵) 選單，然後選擇「刪除」以開啟確認對話方塊。
3. 選擇「刪除」。以確認您要刪除 S3 物件。然後，關閉對話框。

# 使用無伺服器應用程式

提AWS Toolkit for Visual Studio Code供的支援[AWS 無伺服器應用程式](#)。下列主題說明如何從中開始建立和使用AWS Serverless Application Model (AWS SAM) 應用程式AWS Toolkit for Visual Studio Code。

## 主題

- [開始使用無伺服器應用程式](#)
- [執行和偵錯 Lambda 函數，同時排除 AWS SAM 範本資源](#)
- [執行並除錯本機 Amazon API Gateway 資源](#)
- [除錯無伺服器應用程式的組態選項](#)
- [排除無伺服器應用程式](#)

## 開始使用無伺服器應用程式

以下各節說明如何開始AWS 無伺服器應用程式從AWS Toolkit for Visual Studio Code、using AWS Serverless Application Model (AWS SAM) 和AWS CloudFormation堆疊建立。

## 必要條件

您必須先完成下列先決條件AWS 無伺服器應用程式，才能建立或使用。

### Note

下列作業可能會要求您在變更完成之前結束或重新啟動 VS Code。

- 安裝命AWS SAM令列介面 (CLI)。如需有關如何安裝 AWS SAM CLI 的其他資訊和指示，請參閱本AWS Serverless Application Model使用手冊中的 [〈安裝 AWS SAM CLI〉](#) 主題。
- 從您的AWS配置文件中，識別您的默認AWS區域。如需有關組態檔的詳細資訊，請參閱「AWS Command Line Interface使用者指南」中的 [「組態與認證檔案設定」](#) 主題。
- 安裝您的語言 SDK 並配置您的工具鏈。有關如何配置工具鏈的其他資訊，AWS Toolkit for Visual Studio Code請參閱本使用者指南中的[配置工具鏈](#)主題。
- 從 VS 程式碼市集安裝 [YAML 語言支援延伸模組](#)。這是必需的AWS SAM模板文件的 CodeLens 功能是可訪問的。如需其他相關資訊 CodeLens，請參[CodeLens](#)閱 VS Code 文件中的章節

## 無伺服器應用程式的 IAM 許可

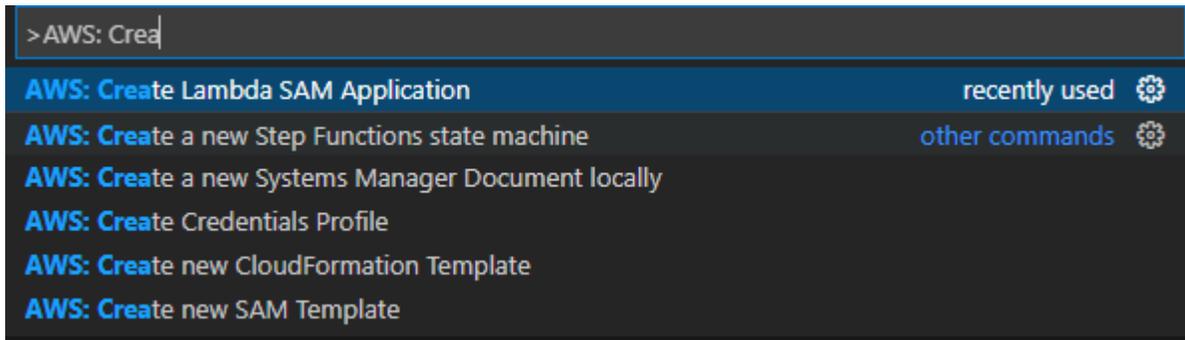
在 Toolkit for VS Code 中，您必須擁有包含部署和執行無伺服器應用程式所需的 AWS Identity and Access Management (IAM) 許可的登入資料設定檔。您必須擁有下列服務的適當讀取/寫入存取權：IAM AWS CloudFormation、Lambda、Amazon API Gateway、亞馬遜簡單儲存服務 (Amazon S3) 和亞馬遜彈性容器登錄 (Amazon ECR)。

如需有關設定部署和執行無伺服器應用程式所需驗證的其他資訊，請參閱AWS Serverless Application Model開發人員指南中的[管理資源存取和權限](#)。如需有關如何設定認證的資訊，請參閱本使用指南[AWS IAM認證](#)中的。

## 建立新的無伺服器應用程式 (本機)

此程序示範如何使用 VS Code 的工具組建立無伺服器應用AWS SAM程式。此程序的輸出是開發主機上的本機目錄，其中包含無伺服器應用程式範例，您可以建置、在本機測試、修改及部署到AWS雲端。

1. 若要開啟「指令選項板」，請選擇「檢視」、「指令選項板」，然後輸入AWS。
2. 選擇AWS工具組建立 Lambda SAM 應用程



### Note

如果未安裝 AWS SAM CLI，您會在 VS 程式碼編輯器的右下角收到錯誤訊息。如果發生這種情況，請確認您已符合所有[假設和先決條件](#)。

3. 選擇AWS SAM應用程式的執行階段。

### Note

如果您選取了包含「(Image)」的執行時間，您的應用程式就是 Image 套件類型。如果您選取了不包含「(Image)」的執行時間，您的應用程式就是 Zip 類型。如需詳細了解

Image 和 Zip 套件類型的差異，請參閱 AWS Lambda 開發人員指南中的 [Lambda 部署套件](#)。

- 視您選取的執行階段而定，系統可能會要求您為 SAM 應用程式選取相依性管理員和執行階段架構。

## Dependency Manager

搖籃或 Maven 之間進行選擇。

### Note

此組建自動化工具的選擇僅適用於 Java 執行階段。

## Architecture

請在 64 或 64 軸之間進行選擇。

在 ARM64 模擬環境中執行無伺服器應用程式，而不是預設的 x86\_64 型環境中執行無伺服器應用程式的選項適用於下列執行階段：

- 節點 12.x ( 郵編和圖像 )
- 節點 14.x ( 郵編和圖像 )
- python3.8 ( 郵編和圖像 )
- python3.9 ( 郵編和圖像 )
- 蟒蛇 3.10 ( 郵編和圖像 )
- 蟒蛇 3.11 ( 郵編和圖像 )
- 蟒蛇 3.12 ( 郵編和圖像 )
- 與搖籃 ( 郵編和圖像 ) 的 java8.al2
- 使用 Maven 的 java8.al2 ( 僅限郵編 )
- 具有搖籃的 java11 ( 郵編和圖像 )
- 使用 Maven 的 java11 ( 僅限郵遞區號 )

**⚠ Important**

您必須安裝 1.33.0 或更新AWS CLI版本，才能允許應用程式在基於 ARM64 的環境中執行。如需詳細資訊，請參閱[必要條件](#)。

5. 您可以選擇新專案的位置。如果現有工作空間已開啟，您可以使用此工作空間。選擇已存在的不同資料夾，或建立新資料夾並選取之。在此範例中，請選擇 There are no workspace folders open (無工作空間資料夾開啟)，以建立名為 MY-SAM-APP 的資料夾。
6. 輸入新專案的名稱。在此範例中，使用 my-sam-app-nodejs。按 Enter 鍵之後，Toolkit for VS Code 需要一些時間來創建項目。

建立專案時，隨即將您的應用程式新增到目前的工作空間。您應會在 Explorer 視窗中看到它。

### 開啟無伺服器應用程式 (本機)

若要在本機開發主機上開啟無伺服器應用程式，請開啟包含應用程式範本檔案的資料夾。

1. 從檔案中，選擇開啟資料夾...。
2. 在「開啟資料夾」對話方塊中，導覽至您要開啟的無伺服器應用程式資料夾。
3. 選擇「選取檔案夾」按鈕。

當您開啟應用程式的資料夾時，它會新增至檔案總管視窗。

### 從範本 (本機) 執行和偵錯無伺服器應用程式

您可以使用 Toolkit for VS Code 來設定如何偵錯無伺服器應用程式，並在開發環境中在本機執行這些應用程式。

您可以使用 VS 程式碼 [CodeLens](#) 功能來識別合格的 Lambda 函數，開始設定偵錯行為。CodeLens 啟用與原始程式碼的內容感知互動。如需確保您可以存取 CodeLens 功能的相關資訊，請檢[必要條件](#)閱本主題稍早的章節。

**i Note**

在此範例中，您會偵錯使用 JavaScript。不過，您可以使用 Toolkit for VS Code 偵錯功能，搭配下列語言和執行階段：

- C# - .NET Core 2.1, 3.1; .NET 5.0
- JavaScript/TypeScript — Node.js 12. 十、十四. x
- Python — 三、三、三、三、三
- 爪哇 — 八、八、十一
- 去 — 1. 倍

您的語言選擇也會影響 CodeLens 偵測合格 Lambda 處理常式的方 如需詳細資訊，請參閱[執行和偵錯 Lambda 函數](#)，同時排除 [AWS SAM 範本資源](#)。

在此程序中，您會使用本主題稍早一[建立新的無伺服器應用程式 \(本機\)](#)節中建立的範例應用程式。

1. 若要在 VS Code 的 [檔案總管] 中檢視您的應用程式檔案，請選擇 [檢視]、[檔
2. 從應用程式資料夾 (例如，my-sample-app) 開啟 template.yaml 檔案。

#### Note

如果您使用的範本名稱與名稱不同 template.yaml，則 YAML 檔案中不會自動提供該 CodeLens 指標。這意味著您必須手動添加調試配置。

3. 在的編輯器中 template.yaml，移至定義無伺服器資源的範本 Resources 區段。在這種情況下，這是類型的 HelloWorldFunction 資源 AWS::Serverless::Function。

在此資源的 CodeLens 指示器中，選擇 [新增偵錯組態]。

4. 在命令調色盤中，選取 AWS SAM 應用程式將在其中執行的執行階段。
5. 在 launch.json 檔案的編輯器中編輯或確認下列組態屬性的值：
  - "name" – 輸入方便識讀的名稱，讓它顯示在 Run (執行) 檢視的 Configuration (組態) 下拉式欄位中。
  - "target" — 確定該值 "template" 使 AWS SAM 範本成為除錯工作階段的進入點。
  - "templatePath" – 輸入 template.yaml 檔案的相對或絕對路徑。
  - "logicalId" — 確定名稱符合 AWS SAM 範本 [資源] 區段中指定的名稱。在此例中，此名稱為 AWS::Serverless::Function 類型的 HelloWorldFunction。

如需詳細了解 `launch.json` 檔案中的上述與其他項目，請參閱 [除錯無伺服器應用程式的組態選項](#)。

- 如果您認為除錯組態沒有問題，請儲存 `launch.json`。然後，要開始調試，請在 RUN 視圖中選擇綠色的「播放」按鈕。

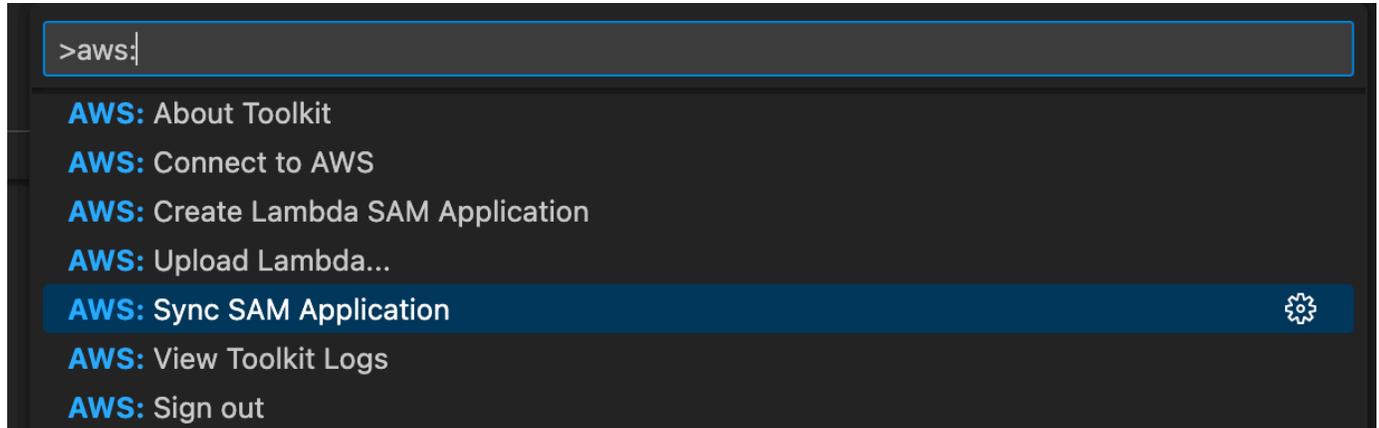
當除錯工作階段啟動時，DEBUG CONSOLE (除錯主控台) 面板會顯示除錯輸出，並顯示 Lambda 函式傳回的任何值。除錯AWS SAM應用程式時，會在「輸出」面板中選取「AWS工具組」作為「輸出」通道。)

## 同步AWS SAM應用

會AWS Toolkit for Visual Studio Code執行 AWS SAM CLI 命令，`sam sync`將無伺服器應用程式部署到AWS 雲端。如需有關AWS SAM同步的其他資訊，請參閱AWS Serverless Application Model開發人員指南中的 [AWS SAMCLI 命令參考](#)主題

下列程序說明如何將無伺服器應用程式`sam sync`從 VS Code 的工具組部署到AWS 雲端與。

- 從 VS Code 的主功能表中，展開 [檢視] 並選擇 [命令面板]，以開啟 [命令面板]。
- 從命令面板中搜尋AWS並選擇同步 SAM 應用程式以開始設定同步處理。



- 選擇要同步無伺服器應用程式的目標AWS區域。
- 選擇要用於部署的 `template.yaml` 檔案。
- 選取現有的 Amazon S3 儲存貯體，或輸入要部署應用程式的新 Amazon S3 儲存貯體名稱。

### Important

您的 Amazon S3 儲存貯體必須符合下列要求：

- 值區必須位於您要同步的地區中。
- 在現有的所有 Amazon S3 儲存貯體名稱中，該 Amazon S3 儲存貯體名稱必須為全域唯一。

6. 如果您的無伺服器應用程式含有套件類型為 Image 的函式，請輸入此部署可以使用的 Amazon ECR 儲存庫名稱。儲存庫必須位於您要部署檔案的區域。
7. 從先前的部署清單中選取部署堆疊，或建立新的部署堆疊並輸入新堆疊名稱。然後，繼續開始同步過程。

#### Note

先前部署中使用的堆疊會依據工作區和區域重新叫用。

8. 在同步處理期間，部署的狀態會擷取到 VS Code 的 [終端機] 索引標籤中。從終端機索引標籤確認您的同步成功，如果發生錯誤，您會收到通知。



#### Note

如需有關同步的其他詳細資訊，可從「指令選項板」存取AWS Toolkit for Visual Studio Code記錄。

若要從「指令面板」存取AWS Toolkit for Visual Studio Code記錄，請展開「檢視」，選擇「指令調色盤」**AWS: View AWS Toolkits Logs**，然後搜尋並在其填入清單中時加以選取。

部署完成後，您會在AWS檔案總管中看到您的應用程式。如需有關如何呼叫做為應用程式一部分建立的 Lambda 函數的詳細資訊，請參閱本使用者指南中的[與遠端 Lambda 函數互動](#)主題。

## 從刪除無伺服器應用程式 AWS 雲端

刪除無伺服器應用程式時，您會刪除之前部署至 AWS 雲端的 AWS CloudFormation 堆疊。請注意，此程序不會從本機主機刪除應用程式目錄。

1. 開啟 [AWS 瀏覽器](#)。

2. 在 [AWS工具組總管] 視窗中，展開包含您要刪除之已部署應用程式的 [區域]，然後展開AWS CloudFormation。
3. 開啟與您要刪除之無伺服器應用程式對應之AWS CloudFormation堆疊名稱的內容 (按一下滑鼠右鍵) 功能表，然後選擇「刪除AWS CloudFormation堆疊」。
4. 若要確認您要刪除選取的堆疊，請選擇「是」。

如果堆疊刪除成功，Toolkit for VS Code 會從AWS檔案總管的AWS CloudFormation清單中移除堆疊名稱。

## 執行和偵錯 Lambda 函數，同時排除 AWS SAM 範本資源

測試 AWS SAM 應用程式時，您可以選擇只執行和偵錯 Lambda 函數，並排除 AWS SAM 範本定義的其他資源。此方法涉及使用此[CodeLens](#)功能，在您可以直接叫用的原始程式碼中識別 Lambda 函數處理常式。

偵測到的 Lambda 處理常式 CodeLens 取決於您用於應用程式的語言和執行階段。

語言/執行時間	由指標識別 Lambda 函數的 CodeLens標準
C# ( 點網核 2.1 , 3.1;. NET5.0)	<p>函式具備以下特性：</p> <ul style="list-style-type: none"> <li>• 這是一個公共類的公共功能。</li> <li>• 它有一個或兩個參數。使用兩個參數，第二個參數必須實現接ILambdaContext 口。</li> <li>• 它在 VS Code 工作區資料夾內的父資料夾中有一個*.csproj檔案。</li> </ul> <p>已安裝並啟用 <a href="#">ms-dotnettools.csharp</a> 副檔名 (或任何提供 C# 語言符號的擴充功能)。</p>
JavaScript/TypeScript (Node.js 12. x, 14 倍)	<p>函式具備以下特性：</p> <ul style="list-style-type: none"> <li>• 它是經過匯出的函式，最多有三個參數。</li> <li>• 它在 VS Code 工作區資料夾內的父資料夾中有一個package.json 檔案。</li> </ul>
Python (3.7, 3.8, 3.9, 三十一, 三)	<p>函式具備以下特性：</p>

語言/執行時間	由指標識別 Lambda 函數的 CodeLens標準
	<ul style="list-style-type: none"> <li>• 它是最上層函數。</li> <li>• 它在 VS Code 工作區資料夾內的父資料夾中有一個requirements.txt 檔案。</li> </ul> <p>已安裝並啟用 <a href="#">ms-pyth.python</a> 擴充功能 (或任何為 Python 提供語言符號的擴充功能)。</p>
<p>爪哇文 (8, 八月二, 11)</p>	<p>函式具備以下特性：</p> <ul style="list-style-type: none"> <li>• 這是一個公共的，非抽象類的公共功能。</li> <li>• 它有一個，兩個或三個參數： <ul style="list-style-type: none"> <li>• 一個參數：參數可以是任何東西。</li> <li>• 兩個參數：參數必須是 a java.io.InputStream 和 java.io.OutputStream OR，最後一個參數必須是com.amazonaws.services.lambda.runtime.Context。</li> <li>• 三個參數：參數必須是 a, java.io.InputStream 而java.io.OutputStream AND最後一個參數必須是com.amazonaws.services.lambda.runtime.Context。</li> </ul> </li> <li>• 它在 VS 代碼工作區文件夾中的父文件夾中有一build.gradle 個pom.xml ( Gradle ) 或 ( Maven ) 文件。</li> </ul> <p>已安裝並啟用 <a href="#">redhat.java</a> 擴充功能 (或為 Java 提供語言符號的任何擴充功能)。無論您使用哪個 Java 運行時，此擴展都需要 Java 11。</p> <p>該 <a href="#">vscjava</a>。 <a href="#">vscode-java-debug</a>已安裝並啟用擴充功能 (或任何提供 Java 偵錯工具的擴充功能)。</p>

語言/執行時間	由指標識別 Lambda 函數的 CodeLens 標準
圍棋 (1. 倍)	<p>函式具備以下特性：</p> <ul style="list-style-type: none"> <li>• 它是最上層函數。</li> <li>• 它需要 0 和 2 個參數之間。如果有兩個引數，第一個引數必須實作 <code>context.Context</code>。</li> <li>• 它返回 0 和 2 個參數之間。如果有超過 0 個參數，則必須實現最後一個參數 <code>error</code>。</li> <li>• 它在 VS 代碼工作區 <code>go.mod</code> 文件夾中有一個文件。</li> </ul> <p>隨即安裝、設定和啟用了 <a href="#">golang.go 擴充功能</a>。</p>

## 如何從應用程式程式碼直接執行與除錯無伺服器應用程式

1. 若要在 VS 程式碼檔案總管中檢視應用程式檔案，請選擇 [檢視] > [檔案總管]
2. 從應用程式資料夾 (例如，`my-sample-app`)，展開函式資料夾 (在本例中為 `hello-world`)，然後 `app.js` 開啟檔案。
3. 在識別合格 Lambda 函數處理常式的 CodeLens 指標中，選擇 `Add Debug Configuration`。
4. 在命令調色盤中，選取 AWS SAM 應用程式將在其中執行的執行階段。
5. 在 `launch.json` 檔案的編輯器中編輯或確認下列組態屬性的值：
  - `"name"` – 輸入方便識讀的名稱，讓它顯示在 Run (執行) 檢視的 Configuration (組態) 下拉式欄位中。
  - `"target"` – 確認值為 `"code"`，以便直接叫用 Lambda 函式處理常式。
  - `"lambdaHandler"` – 輸入程式碼中所用方法的名稱，Lambda 會呼叫此程式碼來叫用您的函數。例如，對於中的應用程式 JavaScript，預設值為 `app.lambdaHandler`。
  - `"projectRoot"` – 為包含 Lambda 函式的應用程式輸入檔案路徑。
  - `"runtime"` – 輸入或確認 Lambda 執行環境的有效執行時間，例如 `"nodejs.12x"`。
  - `"payload"` – 選擇下列其中一個選項來定義您想要提供給 Lambda 函式作為輸入的事件酬載：
    - `"json"`: 定義事件有效負載的 JSON 格式化鍵值對。

- "path"：作為事件酬載的檔案的路徑。

在下面的範例中，選"json"項定義有效負載。

如需詳細了解 launch.json 檔案中的上述與其他項目，請參閱 [除錯無伺服器應用程式的組態選項](#)。

6. 如果您對調試配置感到滿意，要開始調試，請選擇旁邊的綠色播放箭頭RUN。

當偵錯工作階段開始時，面DEBUGCONSOLE板會顯示除錯輸出，並顯示 Lambda 函數傳回的任何值。除錯 AWS SAM 應用程式時，會在「輸出」面板中選取「AWS 工具組」作為「輸出」通道。)

## 執行並除錯本機 Amazon API Gateway 資源

您可以運行或調試AWS SAM API Gateway 本機資源，在template.yaml，通過運行 VS 代碼啟動配置type=aws-sam與invokeTarget.target=api。

### Note

API Gateway 支援 REST 和 HTTP 這兩種類型的 API。不過，API Gateway 功能AWS Toolkit for Visual Studio Code僅支援 REST API。有時 HTTP API 可稱為「API Gateway V2 API」。

### 如何執行並除錯本機 API Gateway 資源

1. 選擇以下其中一種方法來建立 AWS SAM API Gateway 資源的啟動組態：
  - 選項 1：前往處理常式來源程式碼 (.js、.cs 或 .py 檔案)AWS SAM專案中，將滑鼠游標暫留在 Lambda 處理常式上，然後選擇新增除錯組態 CodeLens。然後，在選單中選擇標示為標示為標示為標示的項目API Gateway 事件。
  - 選項 2：Edit (編輯)launch.json並使用以下語法建立新的啟動組態。

```
{
  "type": "aws-sam",
  "request": "direct-invoke",
  "name": "myConfig",
```

```
"invokeTarget": {
  "target": "api",
  "templatePath": "n12/template.yaml",
  "logicalId": "HelloWorldFunction"
},
"api": {
  "path": "/hello",
  "httpMethod": "post",
  "payload": {
    "json": {}
  }
},
"sam": {},
"aws": {}
}
```

2. 在 VS 代碼中執行「面板中，選擇啟動配置 (命名 myConfig 在上面的例子中)。
3. (選用) 在 Lambda 專案程式碼中新增中斷點。
4. 類型 F5 或選擇 Play (播放) 中的執行資料表面板。
5. 在輸出窗格中檢視結果。

## 組態

如果您使用 `invokeTarget.target` 屬性值 `api`，工具組會變更啟動組態驗證和行為，以便支援 `api` 欄位。

```
{
  "type": "aws-sam",
  "request": "direct-invoke",
  "name": "myConfig",
  "invokeTarget": {
    "target": "api",
    "templatePath": "n12/template.yaml",
    "logicalId": "HelloWorldFunction"
  },
  "api": {
    "path": "/hello",
    "httpMethod": "post",
    "payload": {
```

```
    "json": {}
  },
  "queryString": "abc=def&qrs=tuv",
  "headers": {
    "cookie": "name=value; name2=value2; name3=value3"
  }
},
"sam": {},
"aws": {}
}
```

將範例中的值更換為如下所示：

`invokeTarget.logicalId`

API 資源。

路徑

啟動組態請求的 API 路徑，例如 `"path": "/hello"`。

必須為經由 `invokeTarget.templatePath` 指定，從 `template.yaml` 解析的有效 API 路徑。

`httpMethod`

以下其中一個動詞：「delete」、「get」、「head」、「options」、「patch」、「post」、「put」。

`payload`

要在請求中傳送的 JSON 酬載 (HTTP 主體)，結構和規則與 [lambda.payload](#) 欄位相同。

`payload.path` 指向包含 JSON 酬載的檔案。

`payload.json` 指定 JSON 酬載內嵌。

標頭

選用的名稱/值組對應，用來指定要包含在請求中的 HTTP 標頭，如以下範例所示。

```
"headers": {
  "accept-encoding": "deflate, gzip;q=1.0, *;q=0.5",
  "accept-language": "fr-CH, fr;q=0.9, en;q=0.8, de;q=0.7, *;q=0.5",
  "cookie": "name=value; name2=value2; name3=value3",
```

```
"user-agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.198 Safari/537.36",
}
```

## querystring

選用的字串，用於設定請求中的 querystring，例如 "querystring": "abc=def&ghi=jkl"。

## AWS

AWS 連線資訊的提供方式。如需詳細資訊，請參閱 [AWS連線 \(「aws」\) 屬性資料表除錯無伺服器應用程式的組態選項](#) 區段。

## sam

AWS SAM CLI 建置應用程式的方式。如需詳細資訊，請參閱 [AWS山姆 CLI \(「山姆」\) 屬性資料表除錯無伺服器應用程式的組態選項](#) 區段。

## 除錯無伺服器應用程式的組態選項

當您打開 launch.json 文件編輯調試配置，您可以使用 VS 代碼 [IntelliSense](#) 功能查看和自動完成有效的屬性。觸發條件 IntelliSense 在編輯器中，按控制鍵+空格鍵。

```
"lambda": {
  "runtime": "nodejs12.x",
  "event": {
    "json": {}
  }
}
```

IntelliSense 讓您以直接方式或使用 Lambda 函數來尋找並定義叫用 Lambda 函數的屬性AWS SAMTemplate Template 您也可以定義以下屬性："lambda" (函數的執行方式)、"sam" (AWS SAM CLI 建置應用程式的方式)，以及"aws" (AWS 連線資訊的提供方式)。

AWS SAM：直接的 Lambda 處理式叫用/範本為基礎的 Lambda 叫

屬性	描述
type	指定用於管理啟動組態的擴充功能。一律設為 aws-sam，以便使用 AWS SAM CLI 進行本機建置與除錯。

屬性	描述
name	指定方便識讀的名稱，讓它顯示在 Debug launch configuration (除錯啟動組態清單) 中。
request	指定組態的類型由指定的擴展來執行 (aws-sam)。一律設為 <code>direct-invoke</code> 來啟動 Lambda 函數。
invokeTarget	<p>指定叫用資源的進入點。</p> <p>如要直接叫用 Lambda 函數，請設定以下 <code>invokeTarget</code> 欄位的值：</p> <ul style="list-style-type: none"> <li>• <code>target</code> – 設為 <code>code</code>。</li> <li>• <code>lambdaHandler</code> – 要叫用的 Lambda 函數處理常式名稱。</li> <li>• <code>projectRoot</code> — 包含 Lambda 函數處理常式的應用程式檔案路徑。</li> <li>• <code>architecture</code> — 執行本機 SAM Lambda 應用程式之模擬環境的處理器架構。對於某些運行時，您可以選擇 <code>arm64</code> 而不是預設 <code>x86_64</code> 架構。如需詳細資訊，請參閱 <a href="#">建立新的無伺服器應用程式 (本機)</a>。</li> </ul> <p>用於使用調用 Lambda 資源 AWS SAM 範本，設定下列項目的值 <code>invokeTarget</code> 欄位：</p> <ul style="list-style-type: none"> <li>• <code>target</code> – 設為 <code>template</code>。</li> <li>• <code>templatePath</code> — 的路徑 AWS SAM 範本檔案。</li> <li>• <code>logicalId</code> – 要叫用的 <code>AWS::Lambda::Function</code> 或 <code>AWS::Serverless::Function</code> 資源名稱。您可以在 YAML 格式中找到資源名稱 AWS SAM Template Template。請注意，AWS 工具組隱式識別定義的函數 <code>PackageType: Image</code> 中的 AWS SAM Template Template <a href="#">以映像為基礎</a> Lambda 函數。如需詳細資訊，請參閱「<a href="#">Lambda 部署套件</a>」中的 AWS Lambda 開發人員指南。</li> </ul>

Lambda ("**lambda**") 屬性

屬性	描述
environmentVariables	<p>傳送操作參數至 Lambda 函數。例如，如果您正在寫入到 Amazon S3 儲存貯體，而不是在硬編碼您寫入的儲存貯體名稱，請設定儲存貯體名稱為環境變數。</p> <div data-bbox="591 472 1508 1520" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"> <p><b>Note</b></p> <p>為無伺服器應用程式指定環境變數時，您必須將組態新增至AWS SAMTemplate Template(template.yaml ) 和launch.json file.</p> <p>中環境變數的格式化範例AWS SAMTemplate Template</p> <pre data-bbox="672 785 1474 1220">Resources:   HelloWorldFunction:     Type: AWS::Serverless::Function   Properties:     CodeUri: hello-world/     Handler: app.lambdaHandlerN10     Runtime: nodejs10.x     Environment:       Variables:         SAMPLE1: Default Sample 1 Value</pre> <p>中環境變數的格式化範例launch.json 檔案:</p> <pre data-bbox="672 1325 1474 1486">"environmentVariables": {   "SAMPLE1": "My sample 1 value" }</pre> </div>
payload	<p>針對您提供給 Lambda 函數作為輸入的事件酬載提供兩個選項。</p> <ul style="list-style-type: none"> <li>• "json" : 定義事件酬載的 JSON 格式金鑰數值組。</li> <li>• "path" : 作為事件酬載的檔案的路徑。</li> </ul>
memoryMB	指定為執行叫用的 Lambda 函數而提供的記憶體 (MB)。

屬性	描述
runtime	指定 Lambda 函數使用的執行時間。如需詳細資訊，請參閱 <a href="#">AWS Lambda 執行時間</a> 。
timeoutSec	設定除錯工作階段逾時之前的可用時間 (以秒為單位)。
pathMappings	<p>指定本機程式碼相對於其在容器中執行的位置。</p> <p>默認情況下，VS 代碼集的工具包localRoot 連至本機工作區中 Lambda 函數的程式碼根目錄，以及remoteRoot 至/var/task，這是在 Lambda 中執行之程式碼的預設工作目錄。如果工作目錄在 Docker 檔案中變更，或使用WorkingDirectory 中的參數AWS CloudFormation範本檔，至少有一個範本pathMapping 必須指定項目，以便偵錯工具可以成功將本機設定的中斷點對應至 Lambda 容器中執行的程式碼。</p> <p>格式範例pathMappings 中的launch.json 檔案:</p> <pre> "pathMappings": [   {     "localRoot": " \${workspaceFolder}/sam-app/HelloWorldFunction ",     "remoteRoot": " /var/task "   } ] </pre> <p><b>警告：</b></p> <ul style="list-style-type: none"> <li>對於以 .NET 為基礎的 Lambda 函數，remoteRoot 條目必須是構建目錄。</li> <li>對於以節點 .js 為基礎的 Lambda 函數，您只能指定單一路徑對應項目。</li> </ul>

Toolkit for VS Code 使用AWS SAMCLI 在本機建置並除錯無伺服器應用程式。您可以使用 launch.json 檔案中的 "sam" 組態屬性設定 AWS SAM CLI 命令的行為。

## AWS SAM CLI ("sam") 屬性

屬性	描述	預設值
buildArguments	設定 sam build 命令建置 Lambda 來源程式碼的方式。若要檢視建置選項，請參閱 AWS Serverless Application Model 開發人員指南中的 <a href="#">SAM 建置</a> 。	空字串
containerBuild	指出是否在 Lambda 式的碼頭容器中建置函數。	false
dockerNetwork	指定 Lambda Docker 容器應連線的現有 Docker 網路名稱或 ID，以及預設橋接網路。如果沒有指定，則 Lambda 容器只會連線到預設的橋接 Docker 網路。	空字串
localArguments	指定其他本機叫用引數。	空字串
skipNewImageCheck	指定命令是否應略過將 Lambda 執行階段的最新 Docker 映像提取下來的動作。	false
template	自訂您的AWS SAM使用參數輸入客戶值的範本。如需更多詳細資訊，請參閱 <a href="https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/parameters-section-structure.html">https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/parameters-section-structure.html</a> 使用者指南中的AWS CloudFormation參數部分。	"parameters":{}

## AWS 連線 ("aws") 屬性

屬性	描述	預設值
credentials	從憑證檔案選取特定設定檔 (例如 profile:default )，藉此取得 AWS 憑證。	所以此AWS您現有的憑據 <a href="#">共享AWS組態檔或共用AWS登入資料檔</a> 提供給 Toolkit for VS Code。
region	設定服務的 AWS 區域 (例如 us-east-1)。	與作用中的憑證設定檔相關聯的預設 AWS 區域。

### 範例：範本啟動設定

以下為範例啟動組態檔AWS SAM模板目標：

```
{
  "configurations": [
    {
      "type": "aws-sam",
      "request": "direct-invoke",
      "name": "my-example:HelloWorldFunction",
      "invokeTarget": {
        "target": "template",
        "templatePath": "template.yaml",
        "logicalId": "HelloWorldFunction"
      },
      "lambda": {
        "payload": {},
        "environmentVariables": {}
      }
    }
  ]
}
```

### 範例：程式碼啟動組態

以下是 Lambda 函數目標的啟動設定檔範例：

```
{
```

```
"configurations": [  
  {  
    "type": "aws-sam",  
    "request": "direct-invoke",  
    "name": "my-example:app.lambda_handler (python3.7)",  
    "invokeTarget": {  
      "target": "code",  
      "projectRoot": "hello_world",  
      "lambdaHandler": "app.lambda_handler"  
    },  
    "lambda": {  
      "runtime": "python3.7",  
      "payload": {},  
      "environmentVariables": {}  
    }  
  }  
]
```

## 排除無伺服器應用程式

本主題詳細介紹了使用 Toolkit to VS Code 創建無伺服器應用程式時可能會遇到的常見錯誤，以及如何解決問題。

### 主題

- [我怎樣才能使用 SAM 啟動配置的 SAM 配置？](#)
- [錯誤：「運行時錯誤：容器不存在」](#)
- [錯誤：「碼頭。錯誤。API 錯誤：500 伺服器錯誤... 您已達到拉取率限制。」](#)
- [錯誤：「500 服務器錯誤：正在裝入 C:\Users\...」](#)
- [使用 WSL，網絡視圖（例如，「調用AWS「表單」被破壞](#)
- [調試 TypeScript 應用程序，但斷點不起作用](#)

### 我怎樣才能使用 SAM 啟動配置的 SAM 配置？

指定您的 SAM CLI 的位置 [示例配置](#) 通過配置 `--config-file` 參數中的 `sam.localArguments` 啟動組態的屬性。例如，如果 `samconfig.toml` 文件位於工作區的頂層：

```
"sam": {
```

```
"localArguments": ["--config-file", "${workspaceFolder}/samconfig.toml"],  
}
```

錯誤：「運行時錯誤：容器不存在」

所以此sam build命令可以顯示此錯誤，如果您的系統沒有足夠的磁盤空間用於 Docker 容器。如果您的系統存儲只有 1-2 GB 的可用空間，sam build可能會在處理過程中失敗，即使系統存儲在構建開始之前未完全滿。如需詳細資訊，請參閱「[GitHub 問題](#)」。

錯誤：「碼頭。錯誤。API 錯誤：500 伺服器錯誤... 您已達到拉取率限制。」

Docker Hub 限制匿名用戶可以發出的請求。如果您的系統達到限制，Docker 將失敗，並且此錯誤出現在 VS 代碼的輸出視圖中：

```
docker.errors.APIError: 500 Server Error: Internal Server Error ("toomanyrequests: You  
have  
reached your pull rate limit. You may increase the limit by authenticating and  
upgrading:  
https://www.docker.com/increase-rate-limit")
```

確保您的系統碼頭伺服已通過您的 Docker Hub 登入資料進行身份驗證。

錯誤：「500 服務器錯誤：正在裝入 C:\Users\...」

Windows 用戶可能會在調試時看到此 Docker 掛載錯誤AWS SAM應用程序：

```
Fetching lambci/lambda:nodejs10.x Docker container image.....  
2019-07-12 13:36:58 Mounting C:\Users\\AppData\Local\Temp\ ... as /var/  
task:ro,delegated inside runtime container  
Traceback (most recent call last):  
...  
requests.exceptions.HTTPError: 500 Server Error: Internal Server Error ...
```

嘗試刷新共享驅動器的憑據（在 Docker 設置中）。

使用 WSL，網絡視圖（例如，「調用AWS「表單」被破壞

這是 Cisco VPN 用戶的已知 VS 代碼問題。如需詳細資訊，請參閱「[GitHub 問題](#)」。

建議使用解決方法[此 WSL 追蹤問題](#)。

## 調試 TypeScript 應用程式，但斷點不起作用

如果沒有源映射將編譯後的 JavaScript 文件鏈接到源 TypeScript 文件，則會發生這種情況。要糾正此問題，請打開 `tsconfig.json` 文件並確保設置了以下選項和值：`"inlineSourceMap": true`。

## 使用 Systems Manager 自動化文件

AWS Systems Manager 可讓您檢視及控制 AWS 上的基礎設施。Systems Manager 提供統一的使用者界面，以便您檢視來自多項 AWS 服務的操作資料，以及自動執行跨 AWS 資源的操作任務。

[Systems Manager 文件](#) 定義 Systems Manager 在受管執行個體上執行的動作。自動化文件是一種 Systems Manager 文件類型，使用其執行常見的維護和部署和部署，例如建立或更新 Amazon Machine Image (AMI)。本主題概述如何使用建立、編輯、發佈和刪除自動化文件 AWS Toolkit for Visual Studio Code。

### 主題

- [假設和先決條件](#)
- [Systems Manager 自動化文件的 IAM 許可](#)
- [建立新的 Systems Manager 自動化文件](#)
- [開啟現有的 Systems Manager 自動化文件](#)
- [編輯 Systems Manager 自動化文件](#)
- [發佈 Systems Manager 自動化文件](#)
- [刪除 Systems Manager 自動化文件](#)
- [執行 Systems Manager 自動化文件](#)
- [疑難排解 VS 代碼工具包中的 Systems Manager 自動化文檔](#)

## 假設和先決條件

在開始之前，請確認：

- 您已經安裝了視覺工作室代碼和最新版本的 AWS Toolkit for Visual Studio Code。如需詳細資訊，請參閱 [安裝 AWS Toolkit for Visual Studio Code](#)。
- 您熟悉 Systems Manager。如需詳細資訊，請參閱 [AWS Systems Manager 使用者指南](#)。

- 您熟悉 Systems Manager 自動化使用案例。如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的 [AWS Systems Manager 自動化](#)。

## Systems Manager 自動化文件的 IAM 許可

在 Toolkit for VS Code 中，您必須具有包含 AWS Identity and Access Management (IAM) 建立、編輯、發佈和刪除 Systems Manager 自動化文件所需的 (IAM) 許可。以下政策文件定義了可在主體政策中使用的必要 IAM 許可：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:ListDocuments",
        "ssm:ListDocumentVersions",
        "ssm:DescribeDocument",
        "ssm:GetDocument",
        "ssm:CreateDocument",
        "ssm:UpdateDocument",
        "ssm:UpdateDocumentDefaultVersion",
        "ssm>DeleteDocument"
      ],
      "Resource": "*"
    }
  ]
}
```

如需如何更新 IAM 政策的詳細資訊，請參閱 IAM 使用者指南中的 [建立 IAM 政策](#)。有關設定憑證描述檔的詳細資訊，請參閱 [AWS IAM 認證](#)。

## 建立新的 Systems Manager 自動化文件

您可以在中建立新的自動化文件 JSON 或者 YAML 使用 Visual Studio Code。建立新的自動化文件之後，它將顯示在未命名檔案中。您可以命名檔案，然後將其儲存在 VS Code 中，但該檔案名稱不可見 AWS。

### 要新建自動化文件

1. 打開 VS 代碼。

2. 在「」檢視功能表中，選擇Command Palette開啟 Command Palette。
3. 在 Command Palette 中，輸入AWS工具組在本機建立新的 Systems Manager 文件。
4. 為 Hello World 範例選擇其中一個入門範本。
5. 選擇 JSON 或 YAML。

隨即建立新的自動化文件。

#### Note

您在 VS Code 中的新自動化文件不會自動顯示在AWS。您必須將其發佈到 AWS，然後才能運行它。

## 開啟現有的 Systems Manager 自動化文件

您使用AWSExrer 搜尋現有的 Systems Manager 自動化文件。開啟現有的自動化文件時，它會顯示在 VS Code 中未命名檔案。

開啟您的自動化文件

1. 打開 VS 代碼。
2. 從左側導覽中，選擇AWS開啟AWSExrer。
3. 在中AWSExrerSystems Manager，在您要開啟的文件上選擇下載圖示，然後選擇文件版本。檔案將會以該版本的格式開啟。否則，請選擇以 JSON 格式下載或者下載為 YAML。

#### Note

在 VS Code 中將自動化文檔本地保存為文件並不會使其出現在AWS。它需要發佈到AWS在執行之前。

## 編輯 Systems Manager 自動化文件

如果您擁有任何「自動化」文件，這些文件會顯示在我擁有系統管理員文件的類別AWSExrer。您可以擁有已存在於中的自動化文件AWS，您可以擁有先前發佈至的新文件或更新的文件AWS來自 VS 代碼。

當您開啟要在 VS Code 中進行編輯的自動化文件時，您可以執行的動作比在 AWS Management Console。例如：

- 兩者都有模式驗證 JSON 和 YAML 格式。
- 您可以在文件編輯器中使用片段來建立任何自動化步驟類型。
- 有對各種選項的自動完成支持 JSON 和 YAML。

## 處理版本

Systems Manager 自動化文件使用版本進行變更管理。您可以在 VS Code 中為自動化文件選擇預設版本。

### 設定預設版本

- 在 AWS Explorer 中，導覽至您要設定預設版本的文件，開啟文件的內容 (按一下滑鼠右鍵) 選單，然後選擇 Set default version (設定預設版本)。

#### Note

如果選定文件只有一個版本，則將無法變更預設版本。

## 發佈 Systems Manager 自動化文件

在 VS Code 中編輯自動化文件之後，您就可以將其發佈到 AWS。

### 發佈自動化文件

1. 使用中概述的程序，開啟要發佈的自動化文件 [開啟現有的 Systems Manager 自動化文件](#)。
2. 進行您想要發佈的變更。如需詳細資訊，請參閱 [編輯 Systems Manager 自動化文件](#)。
3. 在開啟的檔案右上角，選擇上傳圖示。
4. 在發佈工作流程對話方塊中，選擇 AWS 您想要將自動化文件發佈至其中的區域。
5. 如果您要發佈新文件，請選擇快速建立。否則，請選擇快速更新更新現有的自動化文檔 AWS 區域。
6. 輸入此「自動化」文件的名稱。

當您將更新發佈至現有的自動化文件時 AWS，則會將新版本新增至文件中。

## 刪除 Systems Manager 自動化文件

您可以在 VS 代碼中刪除自動化文檔。刪除自動化文件會同時刪除該文件及其所有版本。

### Important

- 刪除是一種破壞性動作，而且無法復原。
- 刪除已執行的自動化文件並不會刪除AWS啟動時建立或修改的資源。

### 刪除自動化文件

1. 打開 VS 代碼。
2. 從左側導覽中，選擇AWS開啟AWSExrer。
3. 在中AWSExrerSystems Manager開啟要刪除文件的操作功能表 (按一下滑鼠右鍵)，然後選擇刪除文件。

## 執行 Systems Manager 自動化文件

一旦您的自動化文件發佈至AWS，您可以運行它以代替您執行任務AWS帳戶。要運行自動化文件，請使用 AWS Management Console、Systems Manager API、AWS CLI 或 AWS Tools for PowerShell。如需執行自動化文件的說明，請參閱[執行簡易自動化](#)中的AWS Systems Manager使用者指南。

或者，如果您想搭配使用一個 AWS 開發套件與 Systems Manager API 執行自動化文件，請參閱 [AWS 開發套件參考](#)。

### Note

執行自動化文件可以在AWS並可能產生帳單費用。我們強烈建議您在啟動之前了解自動化文件將在您帳戶中建立的資源。

## 疑難排解 VS 代碼工具包中的 Systems Manager 自動化文檔

我將自己的自動化文件儲存在 VS Code 中，但沒有在AWS Management Console。

在 VS 代碼中保存自動化文檔不會將自動化文檔發佈到AWS。如需發佈自動化文件的詳細資訊，請參閱 [發佈 Systems Manager 自動化文件](#)。

發佈我的自動化文件失敗，出現許可錯誤。

請確認您的 AWS 憑證描述檔具有發佈自動化文件所需的許可。如需許可政策範例，請參閱 [Systems Manager 自動化文件的 IAM 許可](#)。

我將我的自動化文件發佈到AWS，但我沒有看到它AWS Management Console。

請確保您已將文件發佈到AWS您正在瀏覽的地區AWS Management Console。

我已經刪除自己的自動化文件，但仍然需要為它建立的資源支付費用。

刪除自動化文件不會同時刪除其建立或修改的資源。您可以從 [AWS 計費管理主控台](#) 識別您所建立的 AWS 資源，瞭解您的費用，然後選擇要從中刪除的資源。

## 使用 AWS Step Functions

AWS Toolkit for Visual Studio Code ( VS 代碼 ) 提供了對 [AWS Step Functions](#)。使用 Toolkit for VS Code，您可以創建，更新和執行 Step Functions 狀態機器。

主題

- [使用 AWS Step Functions](#)

## 使用 AWS Step Functions

您可以使用 AWS Toolkit for Visual Studio Code ( VS 代碼 ) 與[狀態機](#)執行各種操作。

主題

- [必要條件](#)
- [在 VS 代碼中使用狀態機](#)
- [國家機模板](#)
- [狀態機圖形可視化](#)
- [程式碼片段](#)
- [程式碼完成與驗證](#)

### 必要條件

- 請確定您的系統符合[安裝 VS Code 工具組中指定的](#)先決條件，然後安裝工具組。

- 開啟AWS 檔案總管之前，請確定您已設定認證。

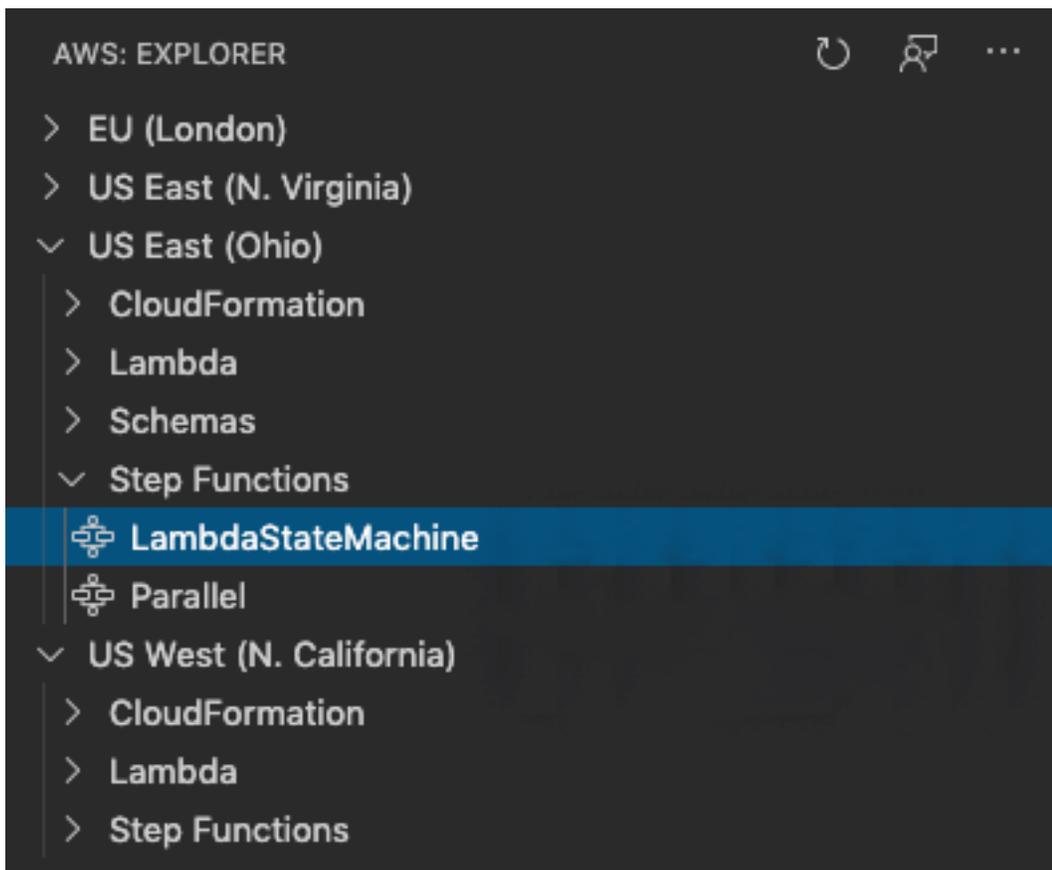
## 在 VS 代碼中使用狀態機

您可以使用 VS Code 與遠端狀態機器互動，並以 JSON 或 YAML 格式在本機開發狀態機器。您可以建立或更新狀態機器、列出現有的狀態機器、執行並下載它們。VS Code 還可讓您從範本建立新狀態機器、檢視狀態機器的視覺效果，並提供程式碼片段、程式碼完成和程式碼驗證。

### 列出現有狀態機

如果您已建立狀態機器，則可以檢視狀態機器列表：

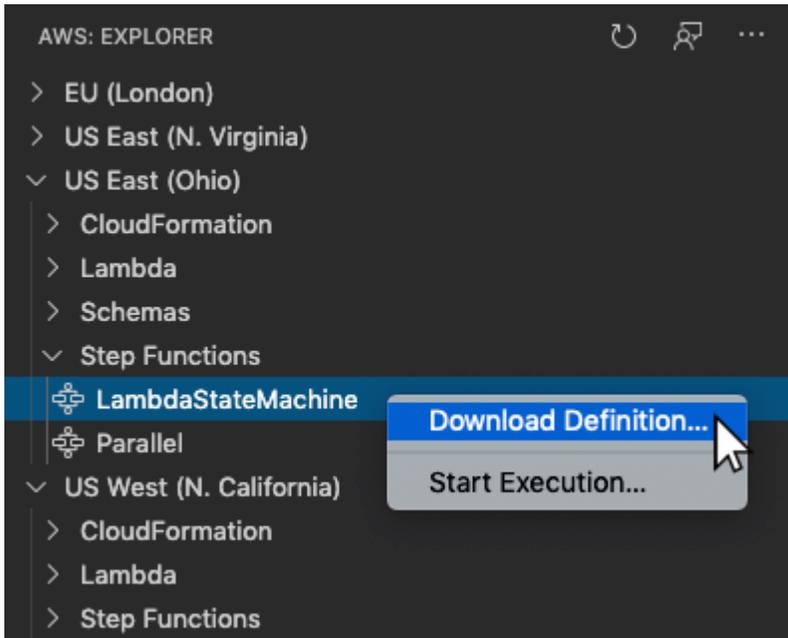
1. 開啟 AWS Explorer。
2. 選擇 Step Functions
3. 驗證列在您帳戶中的所有狀態機器。



### 下載狀態機

如何下載狀態機器：

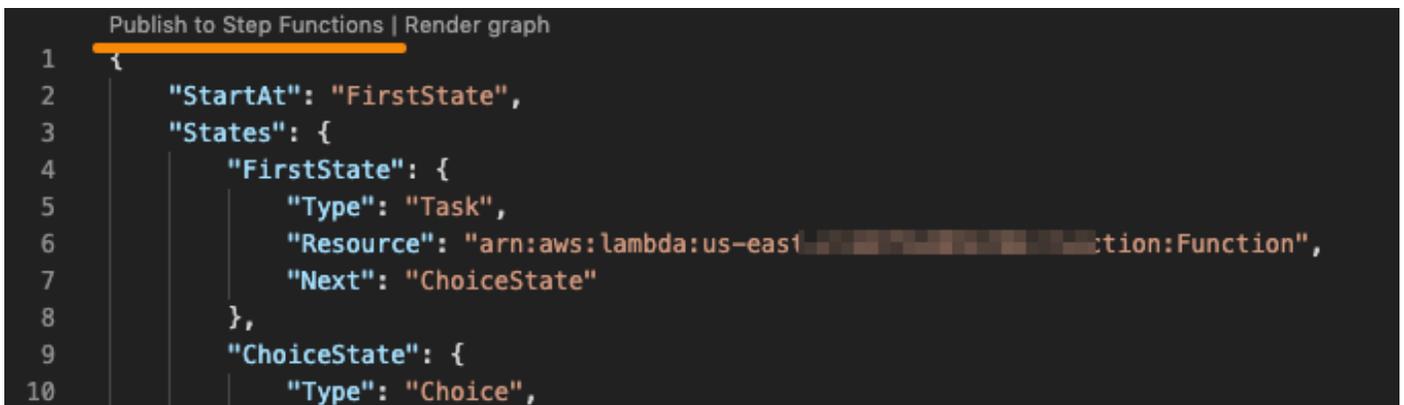
1. 在AWS 檔案總管中，以滑鼠右鍵按一下您要下載的狀態機器。
2. 選取 Download (下載)，然後選取您要下載狀態機器的位置。
3. 請確認狀態機器已正確下載。



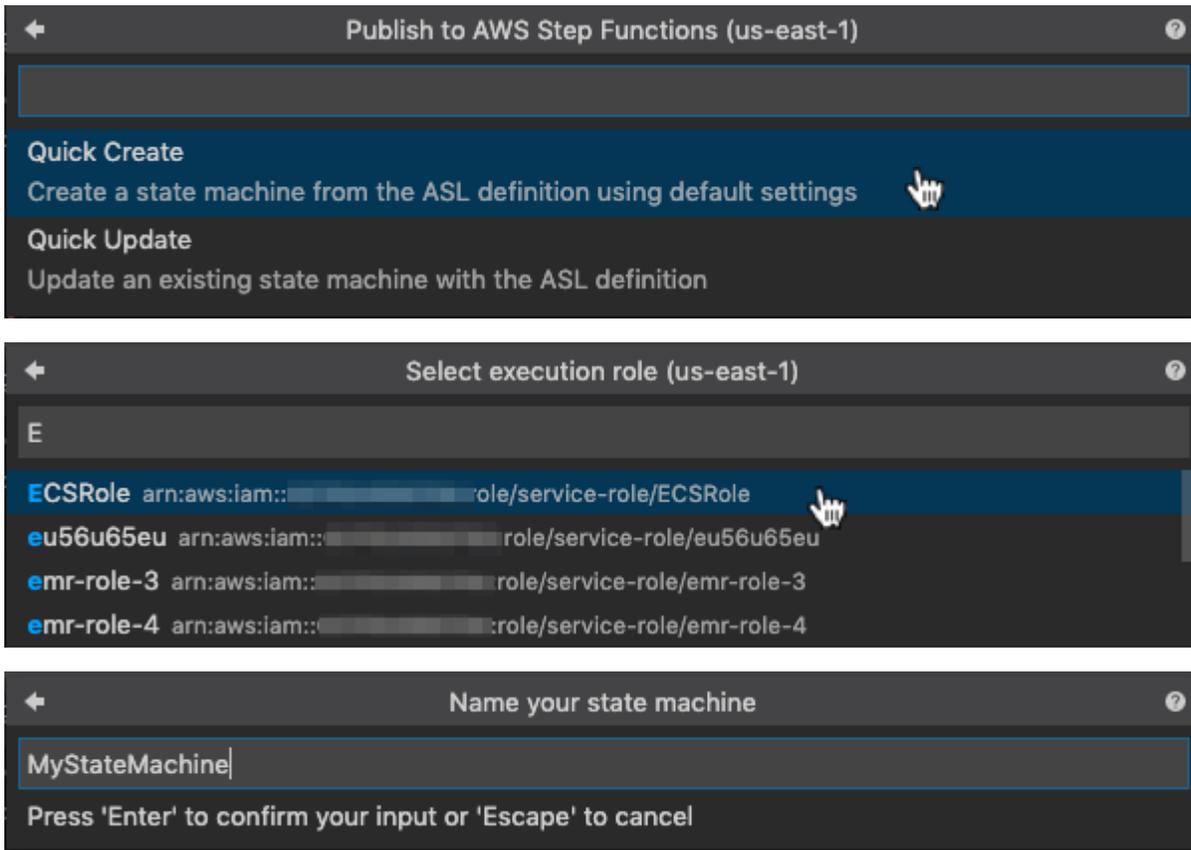
## 建立狀態機器

您可以自行建立新的狀態機器，也可以使用範本。如需從範本建立狀態機器的詳細資訊，請參閱狀態機器範本一節。如何建立新的狀態機器：

1. 使用狀態機器定義，建立新的 [Amazon 狀態語言](#) (ASL) 檔案。使用右下角的功能表，將其設置為 Amazon 狀態語言。
2. 選取發佈至 Step Functions。



3. 選取快速建立、選擇角色，然後命名您的狀態機器。



## 更新狀態機器

如何更新狀態機器：

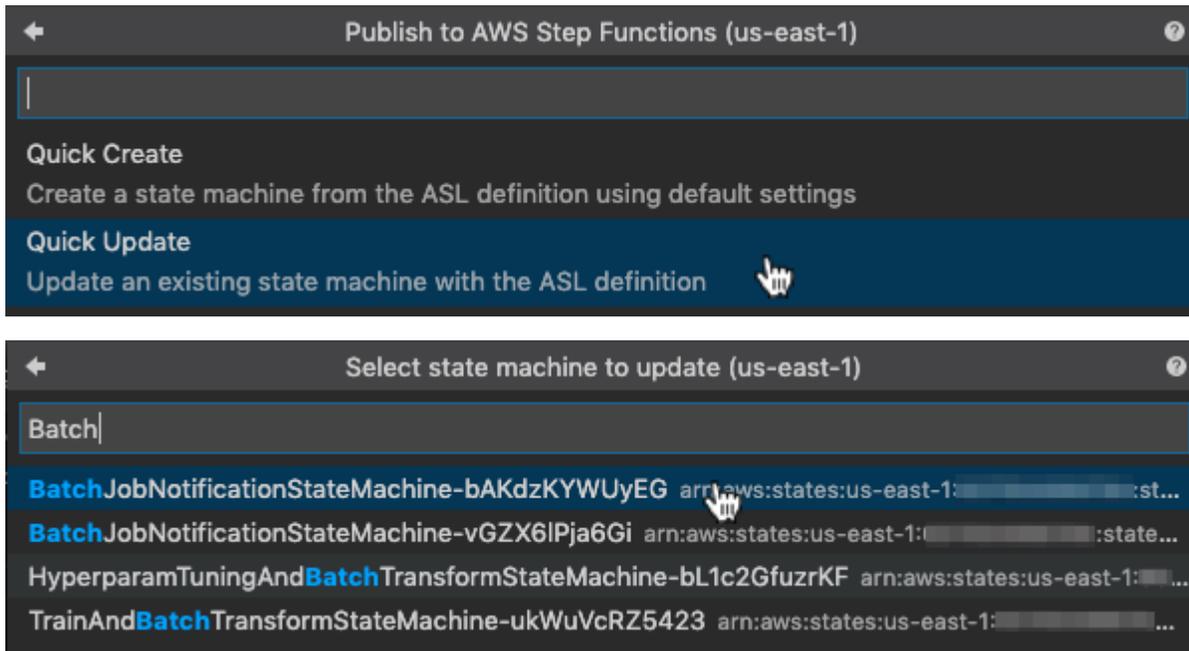
1. 使用狀態機器定義編輯 ASL 檔案。
2. 選取發佈至 Step Functions。

```

Publish to Step Functions | Render graph
1  {
2    "StartAt": "FirstState",
3    "States": {
4      "FirstState": {
5        "Type": "Task",
6        "Resource": "arn:aws:lambda:us-east-1:123456789012:func:Function",
7        "Next": "ChoiceState"
8      },
9      "ChoiceState": {
10     "Type": "Choice",

```

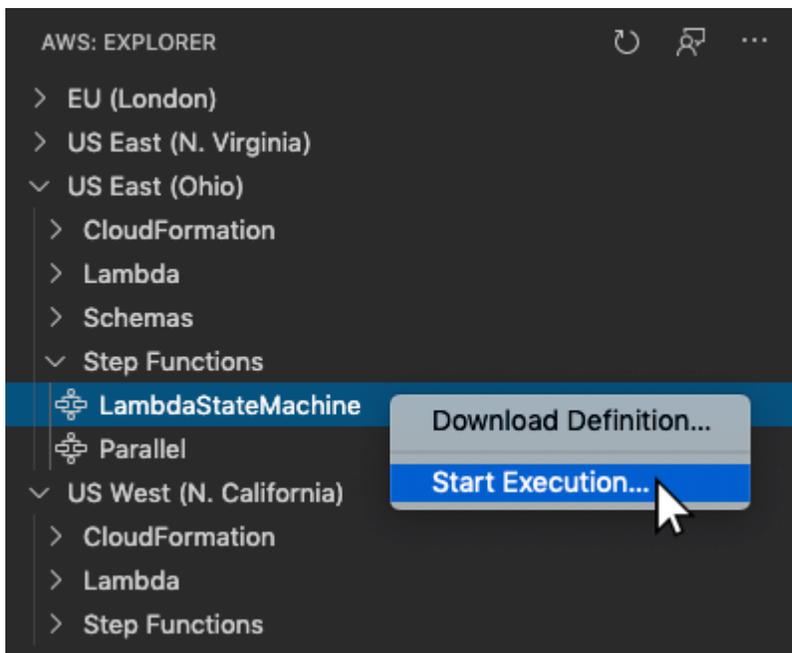
3. 選取快速更新，然後選取您要更新的狀態機器。

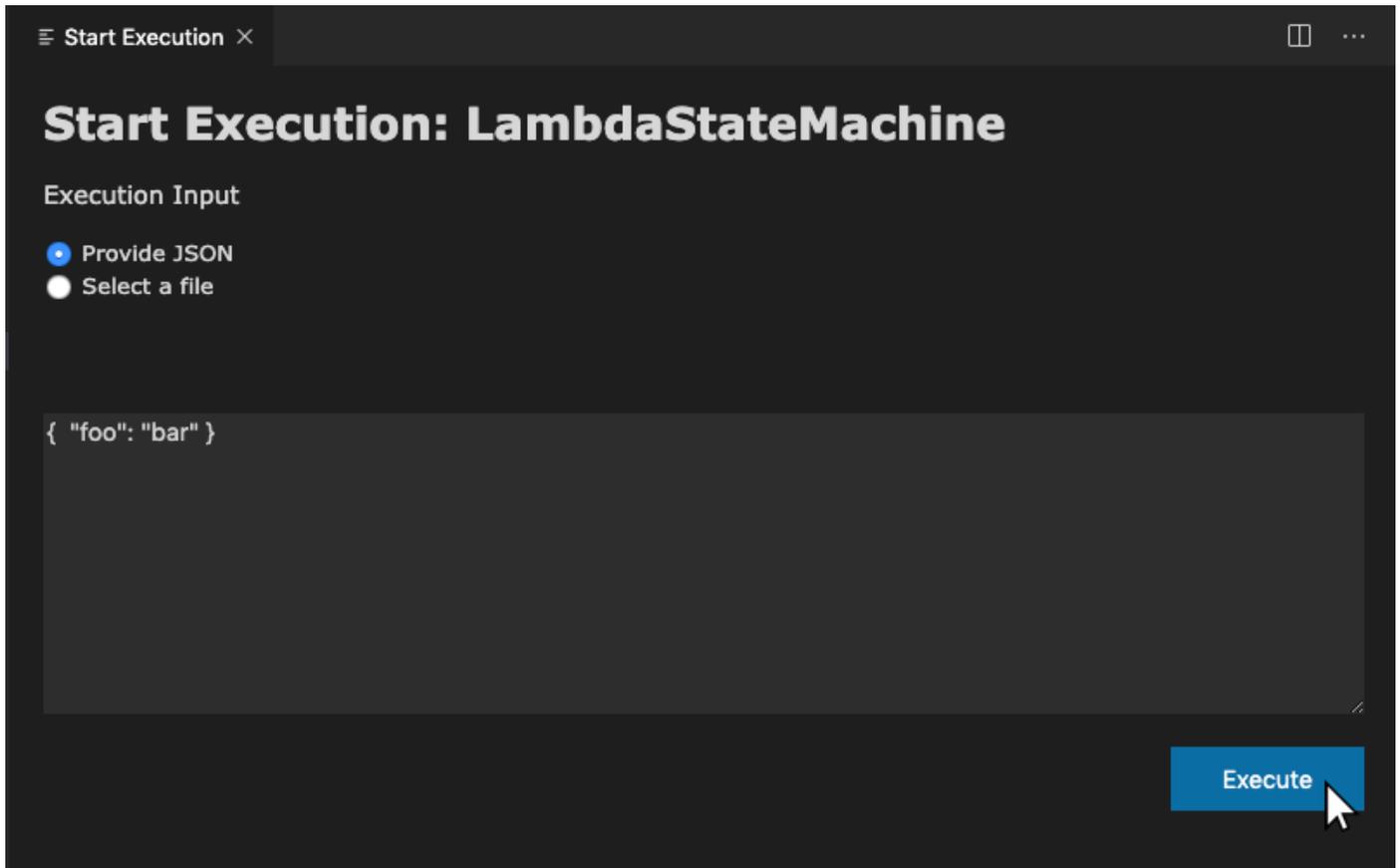


## 運行狀態機

要運行狀態機：

1. 在資AWS 源管理器中，右鍵單擊要運行的狀態機器。
2. 為狀態機提供輸入。您可以嘗試從文件輸入，以及在文字方塊中輸入。
3. 啟動狀態機並驗證它是否成功運行。

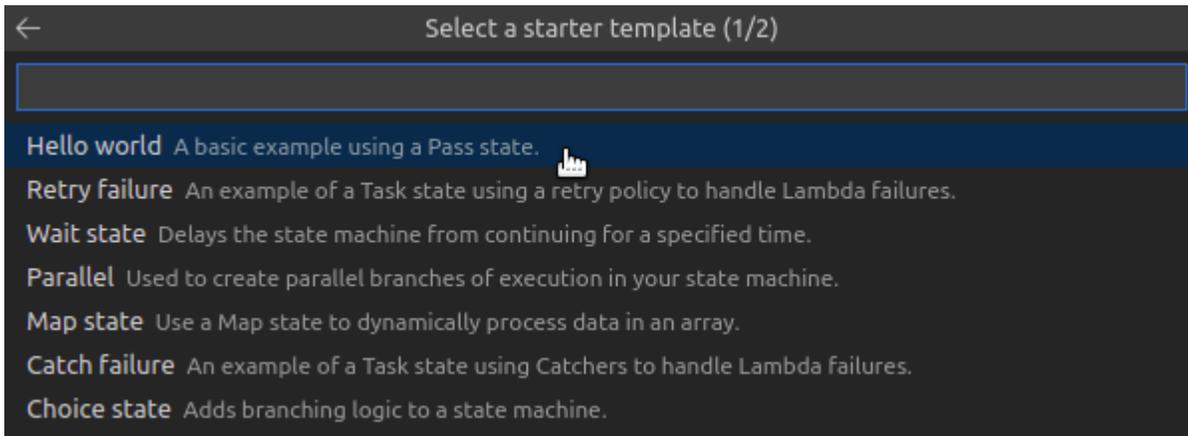




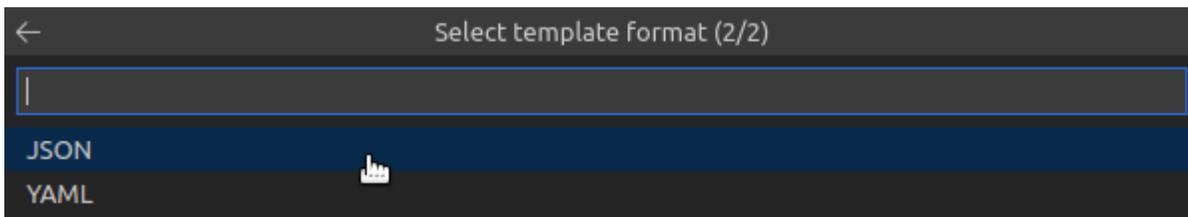
## 国家机模板

建立狀態機器時，可以從範本選擇建立狀態機器。範本包含具有數種常用狀態的範例狀態機器定義，並提供您一個起點。如何使用狀態機器範本：

1. 在 VS 代碼中打開命令調色板。
2. 選取AWS 工具組建立新的 Step Functions 狀態機器。
3. 選擇您要使用的範本。



#### 4. 選擇您要使用 JSON 或 YAML 範本格式。



## 狀態機圖形可視化

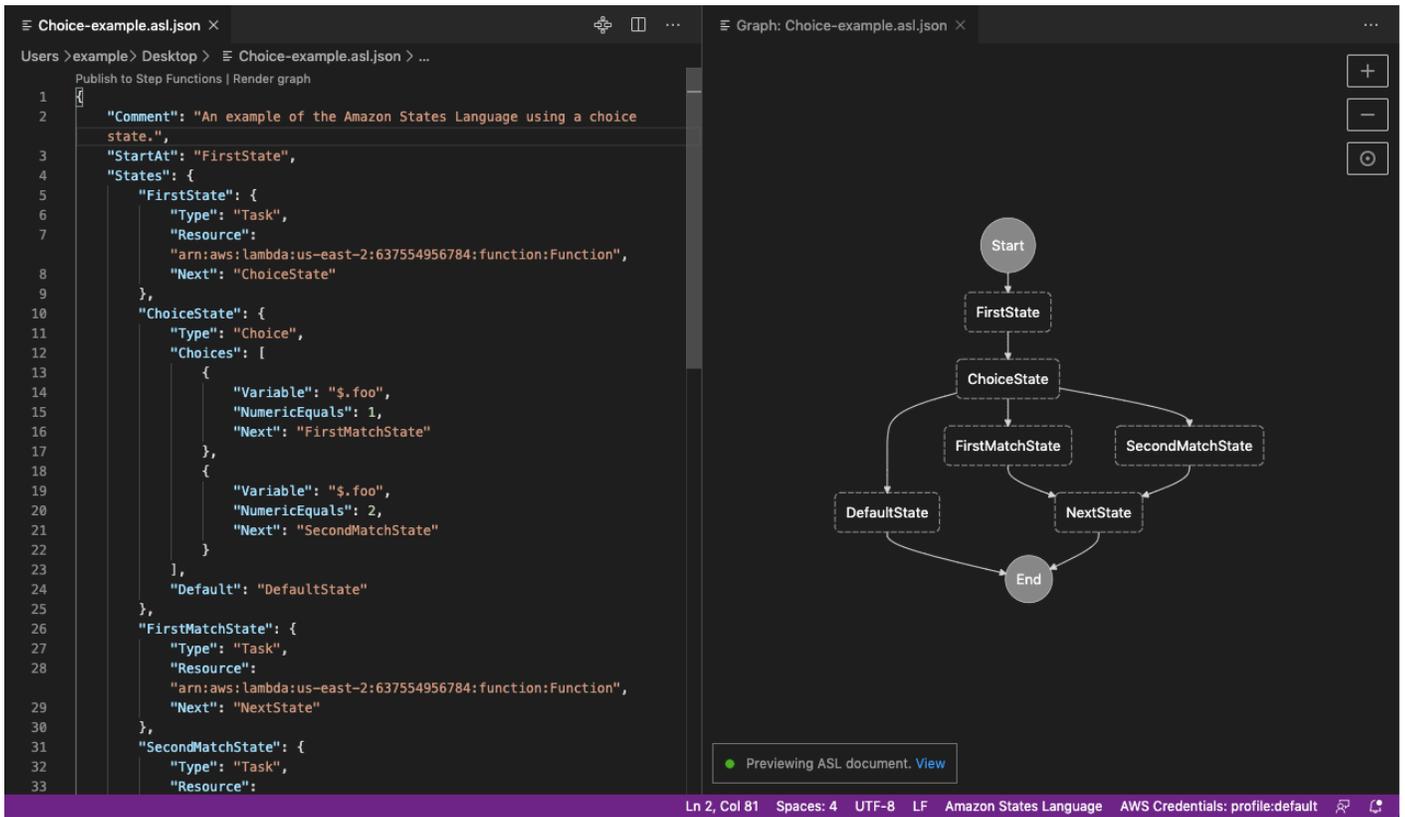
圖形視覺化可讓您以圖形格式檢視狀態機器的外觀。當您創建圖形可視化時，另一個選項卡將打開並顯示狀態機 JSON 或 YAML 的可視化。然後，您可以將您同時正在寫入的狀態機器定義及其視覺效果進行比較。變更狀態機器定義時，將更新視覺化。

### Note

若要建立狀態機器定義的視覺化，必須在作用中的編輯器中開啟定義。若您關閉或重新命名定義檔，則將關閉視覺化。

如何建立狀態機器圖形視覺化：

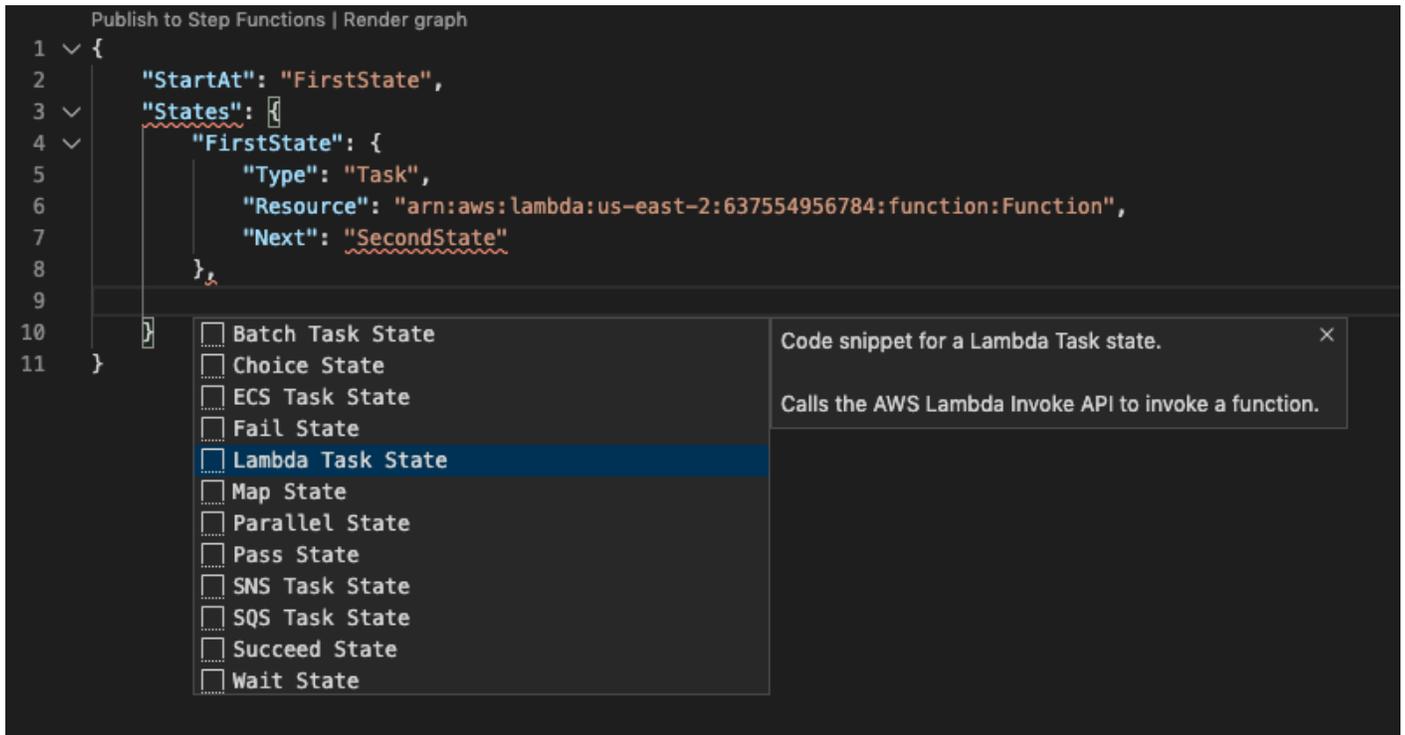
1. 定義您的狀態機器。
2. 在 VS 代碼中打開命令調色板。
3. 若要建立視覺效果，請使用右上角的視覺效果按鈕，或選擇「AWS 彩現圖表」。



## 程式碼片段

程式碼片段可讓您插入程式碼的一小段。如何使用程式碼片段：

1. 開啟檔案並以 JSON 格式或 `.asl.yaml` YAML 格式 `.asl.json` 的副檔名儲存檔案。
2. 使用狀態屬性，建立新的狀態機器。
3. 將游標放置在狀態內。
4. 使用按鍵組合 `Control + Space`，然後選取您偏好的程式碼片段。
5. 用 `Tab` 於遍歷程式碼片段中的變數和參數。
6. 將游標放置在相關狀態內，以測試 `Retry` 和 `Catch` 片段。



## 程式碼完成與驗證

如何檢視程式碼完成的運作方式：

1. 建立數個狀態。
2. 將游標置於「下一個」StartAt、或「預設」性質之後。
3. 使用按鍵組合 Control + Space，列出可用的完成項目。額外屬性均可以用 Control + Space 再次存取，並將以 State 的 Type 為基礎。
4. 工作時，程式碼驗證將檢視：
  - 缺少屬性
  - 不正確的值
  - 無終端狀態
  - 指向不存在的狀態

```

"FirstMatchState": {
  "Type": "Task",
  "Resource": "arn:aws:lambda:us-east-2:637554956784:function:Function",
  "Next": ""
},
"SecondMatchState": {
  "Type": "Task",
  "Resource": "arn:aws:lambda:us-east-2:637554956784:function:Function",
  "Next": ""
},
"DefaultState": {
  "Type": "Fail",
  "Error": "DefaultStateError",
  "Cause": "No Matches!"
},
"NextState": {
  "Type": "Task",
  "Resource": "arn:aws:lambda:us-east-2:637554956784:function:Function",
  "End": true
}

```

```

"FirstMatchState": {
  "Type": "Task",
  "Resource": "arn:aws:lambda:us-east-2:637554956784:function:Function",
  "Catch": [
    {
      "Error": "DefaultStateError",
      "Cause": "No Matches!"
    }
  ],
  "End": true
}

```

An array of objects, called Catchers, that define a fallback state. This state is executed if the state encounters runtime errors and its retry policy is exhausted or isn't defined.

## 使用威脅撰寫器

您可以使用 AWS Toolkit for Visual Studio Code 搭配安全威脅撰寫器工具。威脅撰寫器是一種威脅建模工具，可簡化您的威脅建模程序。

如需有關安全威脅撰寫器工具的詳細資訊，請參閱[威脅撰寫器 GitHub 存放庫](#)。

下列主題說明如何使用中的安全威脅撰寫器 AWS Toolkit for Visual Studio Code。

### 主題

- [使用工具組中的威脅撰寫器](#)

## 使用工具組中的威脅撰寫器

透過威脅撰寫器，您可以直接在 VS Code 中建立、檢視及編輯威脅撰寫器威脅模型。如需有關安全威脅撰寫器工具的詳細資訊，請參閱[威脅撰寫器 GitHub 存放庫](#)。

下列各節說明如何存取中的安全威脅撰寫器工具 AWS Toolkit for Visual Studio Code。

### 從工具組存取威脅撰寫器

您可以從工具包訪問威脅撰寫器的 3 種主要方法。

#### 透過現有的威脅模型存取威脅撰寫器

若要開啟威脅撰寫器，請在 VS Code 中開啟現有的威脅模型檔案 ( 副檔名 `.tc.json` )。威脅撰寫器會在 VS 程式碼編輯器視窗中自動開啟並呈現威脅模型檔案的視覺效果。

#### 建立新的威脅撰寫器威脅模型

1. 在「VS 程式碼」主功能表中，展開「檔案」，然後選擇「新增檔案」。
2. 在「新增檔案」對話方塊中，選擇「威脅撰寫器檔案」。
3. 出現提示時，輸入 a file name，然後按下 **enter** 鍵開啟威脅撰寫器，並在新的 VS Code 編輯器視窗中建立空白威脅模型檔案的視覺效果。

#### 從命令調色盤建立新的威脅撰寫器威脅模型

1. 在 VS 程式碼中，按下 **Cmd + Shift + P** 或 **Ctrl + Shift + P** (視窗) 開啟命令選項板。
2. 在搜尋欄位中，輸入 **Threat Composer** 並在結果中填入新的安全威脅撰寫器檔案時選擇「建立新的威脅撰寫器檔案」
3. 出現提示時，輸入 a file name，然後按下 **enter** 鍵開啟威脅撰寫器，並在新的 VS Code 編輯器視窗中建立空白威脅模型檔案的視覺效果。

## 使用資源

除了存取資源 AWS 管理器中預設列出的 AWS 服務之外，您還可以移至 [資源]，並從數百個資源中選擇要新增至介面的資源。在中 AWS，資源是您可以使用的實體。一些可以新增的資源包括 Amazon AppFlow、Amazon Kinesis 資料串流、AWS IAM 角色 VPC、Amazon 和亞馬遜 CloudFront 分發。

完成選擇後，您可以前往資源並展開資源類型，列出該類型的可用資源。例如：若您選取 `AWS Toolkit:Lambda::Function` 資源類型，您可以存取定義不同功能、其屬性及其屬性的資源。

將資源類型新增到 Resources (資源) 後，您便可使用下列方式來與資源互動：

- 檢視此資源類型目前「AWS 區域」中可用的現有資源清單。
- 檢視描述資源的JSON檔案的唯讀版本。
- 複製資源的資源識別符。
- 檢視說明資源類型用途的 AWS 文件，以及用於建立資源模型的結構描述 (以JSON及YAML格式)。
- 透過編輯和儲存符合結構描述的JSON格式範本來建立新資源。 \*
- 更新或刪除現有的資源。 \*

#### Important

\* 在當前版本中 AWS Toolkit for Visual Studio Code 創建，編輯和刪除資源的選項是一項實驗性功能。實驗功能會持續進行測試和更新，因此可能存在可用性問題。並且實驗功能可能會從中刪除，恕 AWS Toolkit for Visual Studio Code 不另行通知。

若要允許對資源使用實驗性功能，請開啟 VS Code 中的 [設定] 窗格，然後展開 [擴充功能]IDE，然後選擇 [AWS 工具組]。

在「AWS 工具組實驗」下，選取 `jsonResourceModification` 以允許您建立、更新及刪除資源。如需詳細資訊，請參閱 [使用實驗功能](#)。

## IAM存取資源的權限

您需要特定的 AWS Identity and Access Management 權限才能存取與 AWS 服務相關聯的資源。例如，IAM實體 (例如使用者或角色) 需要 `Lambda` 許可才能存取 `AWS Toolkit:Lambda::Function` 資源。

除了服務資源的權限之外，IAM實體還需要權限，才能允許 Toolkit for VS Code 代表呼叫 AWS Cloud Control API 作業。雲端控制API作業可讓IAM使用者或角色存取和更新遠端資源。

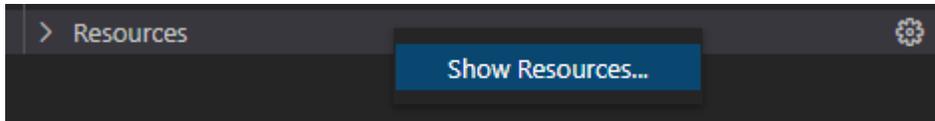
授與權限最簡單的方法是將 AWS 受管理的原則 `PowerUserAccess` 附加至使用 Toolkit 介面呼叫這些 API 作業的 IAM 實體。此 [受管理原則](#) 會授與執行應用程式開發工作 (包括呼叫 API 作業) 的一系列權限。

如需定義遠端資源允許 API 作業的特定權限，請參閱 [AWS Cloud Control API 使用者指南](#)。

## 新增並與現有資源互動

1. 在AWS 檔案總管中，在資源上按一下滑鼠右鍵，然後選擇

窗格會顯示可供選取的資源類型清單。



2. 在選取窗格中，選取要新增至AWS 檔案總管的資源類型，然後按 Return 鍵或選擇 [確定] 以確認。

您選取的資源類型會列在資源底下。

### Note

如果您已將資源類型新增至 AWS Explorer，然後清除了該類型的核取方塊，在您選擇之後確定之後，該類型就不會再列於資源底下。只有目前選取的資源類型會顯示在 AWS Explorer 中。

3. 若要檢視資源類型已存在的資源，請展開該類型的項目。

可用資源清單會顯示在資源類型下方。

4. 若要與特定資源互動，請在其名稱上按一下滑鼠右鍵，然後選擇以下其中一個選項：

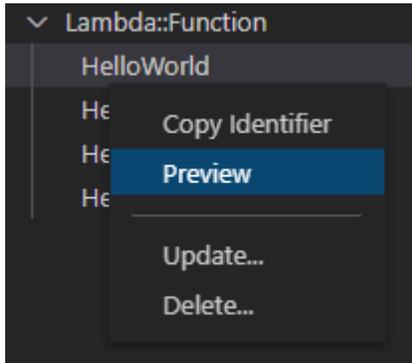
- 複製資源識別碼：將特定資源的識別碼複製到剪貼簿。(例如：AWS Toolkit:DynamoDB::Table 資源可以使用 TableName 屬性進行識別。)
- 預覽：檢視描述資源之JSON已格式化範本的唯一讀版本。

資源範本顯示後，您可以選擇編輯器標籤右側的「更新」圖示來修改該範本。若要更新資源，您必須`???`啟用必要項目。

- 更新：在 VS 代碼編輯器中編輯資源的JSON格式化模板。如需詳細資訊，請參閱[建立和編輯資源](#)。
- 刪除：在顯示的對話方塊中確認刪除，以刪除資源。(目前在這個版本的`???`中刪除資源 AWS Toolkit for Visual Studio Code。)

**⚠ Warning**

如果刪除資源，則使用該資源的任何 AWS CloudFormation 堆疊將無法更新。若要修正此更新失敗，您必須重新建立資源，或移除堆疊 AWS CloudFormation 範本中的資源參考。如需詳細資訊，請參閱這篇[知識中心文章](#)。



## 建立和編輯資源

**⚠ Important**

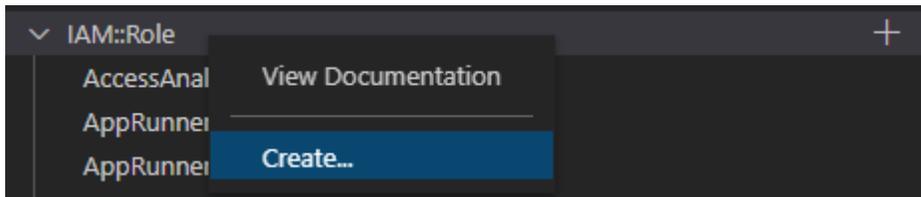
建立和更新資源的功能目前在此版本的 AWS Toolkit for Visual Studio Code 中為[???](#)。

建立新資源涉及將資源類型新增至 [資源] 清單，然後編輯定義資源、其屬性及其屬性的 JSON 已格式化範本。

例如，屬於該資源類型的 `AWS Toolkit:SageMaker::UserProfile` 資源使用可為 Amazon SageMaker Studio 建立使用者設定檔的範本定義。定義此使用者描述檔資源的範本必須符合 `AWS Toolkit:SageMaker::UserProfile` 的資源類型結構描述。例如，如果範本因為缺少或不正確的屬性而不符合結構描述，就無法建立或更新資源。

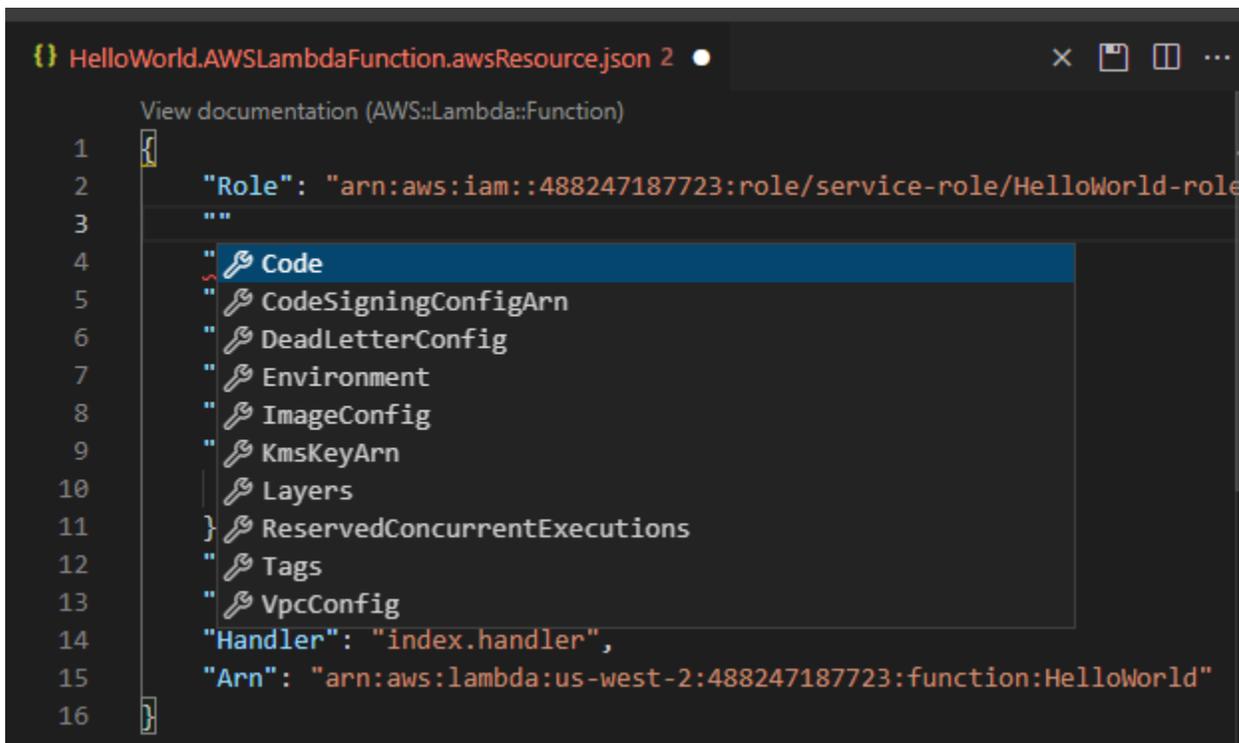
1. 在資源上按一下滑鼠右鍵，然後選擇顯示資源，為您要建立的資源新增資源類型。
2. 在資源下新增資源類型後，選擇加號 (「+」) 圖示以在新編輯器中開啟範本檔案。

或者，您可以用滑鼠右鍵按一下資源類型的名稱，然後選擇 [建立] 您也可以選擇檢視文件，存取有關如何建立資源模型的資訊。



3. 在編輯器中，開始定義組成資源範本的屬性。自動完成功能會建議符合範本結構描述的屬性名稱。當您將游標暫留在某個屬性類型上時，窗格會顯示其用途的描述。如需結構描述的詳細資訊，請選擇檢視文件。

任何不符合資源結構描述的文字都會以紅色波浪底線表示。



4. 完成資源宣告後，請選擇 [儲存] 圖示以驗證範本，並將資源儲存至遠端 AWS 雲端。

如果您的範本是根據其結構描述定義資源，系統會顯示一則訊息，以確認資源已建立。(如果資源已存在，訊息會確認資源已更新。)

資源建立完成後，會新增至資源類型標題下的清單中。

5. 如果您的檔案包含錯誤，系統會顯示一則訊息，說明無法建立或更新資源。選擇「檢視記錄檔」以識別您需要修正的範本元素。

# AWS Toolkit for Visual Studio Code 的安全

## 主題

- [資料保護 AWS Toolkit for Visual Studio Code](#)

## 資料保護 AWS Toolkit for Visual Studio Code

AWS [共用責任模型](#)適用於 Visual Studio 程式碼之 AWS 工具組中的資料保護。如此模型中所述，AWS 負責保護執行所有 AWS 雲端。您負責維護在此基礎設施上託管內容的控制權。您也同時負責所使用 AWS 服務的安全組態和管理任務。如需有關資料隱私權的詳細資訊，請參閱[資料隱私權FAQ](#)。如需歐洲資料保護的相關資訊，請參閱AWS 安全性GDPR部落格上的[AWS 共同責任模型和](#)部落格文章。

基於資料保護目的，我們建議您使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 保護 AWS 帳戶認證並設定個別使用者。如此一來，每個使用者都只會獲得授與完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 對每個帳戶使用多重要素驗證 (MFA)。
- 使用SSL/TLS與 AWS 資源溝通。我們需要 TLS 1.2 並推薦 TLS 1.3。
- 使用設定API和使用者活動記錄 AWS CloudTrail。
- 使用 AWS 加密解決方案以及其中的所有默認安全控制 AWS 服務。
- 使用進階的受管安全服務 (例如 Amazon Macie)，協助探索和保護儲存在 Amazon S3 的敏感資料。
- 如果 AWS 透過命令列介面或存取時需要 FIPS 140-3 驗證的密碼編譯模組API，請使用端點。FIPS 如需有關可用FIPS端點的詳細資訊，請參閱[聯邦資訊處理標準 \(FIPS\) 140-3](#)。

我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊，放在標籤或自由格式的文字欄位中，例如名稱欄位。這包括當您使用工 AWS 具組適用於 Visual Studio 程式碼或其他 AWS 服務使用主控台API AWS CLI、或 AWS SDKs。您在標籤或自由格式文字欄位中輸入的任何資料都可能用於計費或診斷日誌。如果您提供URL給外部伺服器，我們強烈建議您不要在中包含認證資訊，URL以驗證您對該伺服器的要求。

# AWS Toolkit for Visual Studio Code 使用者指南的文件歷史記錄

下表說明每個 AWS Toolkit for Visual Studio Code 版本的重要變更。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
<a href="#">AWS Identity and Access Management (IAM) 存取分析器更新</a>	更新了IAM訪問分析器內容以包含新的API引用。	2024年7月10日
<a href="#">AWS Identity and Access Management (IAM) 存取分析器</a>	為IAM訪問分析器添加了新的用戶指南主題。	2024年5月23日
<a href="#">Connect 到 AWS 授權流程已更新</a>	授權流程已更新，以反映驗證程序的變更，以及 Amazon Q 與 AWS Toolkit for Visual Studio Code.	2024 年 4 月 30 日
<a href="#">Amazon Q 擴展 VS 代碼</a>	截至 2024 年 4 月 30 日，現在 CodeWhisperer 是 Amazon Q 和 Amazon Q 的一部分可作為 VS 代碼的擴展。	2024 年 4 月 30 日
<a href="#">開發環境中的虛擬私有雲 Support 援</a>	已更新內容涵蓋 UI 變更，以支援VPC開發環境。	2024年1月21日
<a href="#">應用程式編</a>	已將新的應用程式撰寫器主題新增至 AWS Toolkit for Visual Studio Code 使用指南。	2023 年 11 月 28 日
<a href="#">SSO支援 CodeCatalyst</a>	已更新內容，以涵蓋IAM身分識別中心支援 CodeCatalyst 和開發環境。	2023 年 11 月 17 日

<a href="#">添加 VS 代碼和工具包下載鏈接</a>	更新內容與 VS 代碼的下載鏈接和 AWS Toolkit for Visual Studio Code.	2023 年 11 月 1 日
<a href="#">Amazon Redshift 主題</a>	在 AWS Toolkit for Visual Studio Code 用戶指南中添加了新的 Amazon Redshift 主題。	2023 年 10 月 17 日
<a href="#">Connect 到 AWS 授權流程已更新</a>	授權流程已更新，以著重於服務特定的驗證方法。	2023 年 9 月 29 日
<a href="#">建立的使用者指南：建立範本 CloudFormation</a>	創建了一個新的用戶指南，描述如何使用 Toolkit for VS Code 創建 CloudFormation 模板	2021 年 12 月 17 日
<a href="#">次要 UI 更新</a>	將「預覽機器狀態」的現有文字更新為「轉譯圖形」，以便更符合 UI。	2021 年 12 月 14 日
<a href="#">建立了 Amazon Elastic Container Service Exec 的使用者指南</a>	這是 Amazon ECS Exec 的概述。	2021 年 12 月 13 日
<a href="#">創建 VS 代碼服務工 AWS IoT 工具包的用戶指南</a>	本使用者指南旨在協助您開始使用 VS Code 工具組的 AWS IoT 服務。	2021 年 11 月 22 日
<a href="#">實驗功能支援</a>	已新增對開啟服務實驗性功能的支 AWS 援。	2021 年 10 月 14 日
<a href="#">AWS 資源 Support</a>	新增對資源類型的支援，以及建立、編輯和刪除資源的介面選項。	2021 年 10 月 14 日
<a href="#">Amazon 服 ECR 務概述 AWS Toolkit for Visual Studio Code</a>	新增了可在 VS Code 中存取的 Amazon ECR 服務功能的概觀和逐步解說	2021 年 10 月 14 日

<a href="#">ARM64環境 Support</a>	您現在可以在模擬環境以及 ARM64基於 x86_64 的環境中執行無伺服器應用程式。	2021 年 10 月 1 日
<a href="#">AWS 無伺服器應用程式</a>	增加了對在 ARM64 平台上運行 AWS SAM 應用程式的	2021 年 9 月 30 日
<a href="#">格式更新 Node.js 區段</a>	根據客戶的反饋，更新了 Node TypeScript .js/ 的格式。	2021 年 8 月 12 日
<a href="#">App Runner 支援</a>	增加了 AWS App Runner 對 AWS Toolkit for Visual Studio Code.	2021 年 8 月 11 日
<a href="#">調試 Go 功能</a>	增加了對調試本地 Go 功能的支持。	2021 年 5 月 10 日
<a href="#">調試 Java 函數</a>	增加了對調試本地 Java 函數的支持。	2021 年 4 月 22 日
<a href="#">YAML支援 AWS Step Functions</a>	增YAML加了對 AWS Step Functions.	2021 年 3 月 4 日
<a href="#">偵錯 Amazon API 閘道資源</a>	已新增對本機 Amazon API 閘道資源除錯的支援。	2020 年 12 月 1 日
<a href="#">Amazon API 网关</a>	增加了對 Amazon API 網關的支持。	2020 年 12 月 1 日
<a href="#">AWS 無伺服器應用程式</a>	使用無伺服器應用程式新增 Lambda 容器映像的支援。	2020 年 12 月 1 日
<a href="#">AWS Systems Manager 支持</a>	新增對 Systems Manager 自動化文件的支援。	2020 年 9 月 30 日
<a href="#">CloudWatch 日誌</a>	增加了對 CloudWatch 日誌的支持。	2020 年 8 月 24 日
<a href="#">Amazon Simple Storage Service (Amazon S3)</a>	增加了對 Amazon S3 的支持。	2020 年 7 月 30 日

<a href="#">AWS Step Functions 支持</a>	增加了對 AWS Step Functions .	2020 年 3 月 31 日
<a href="#">安全性內容</a>	已新增安全內容。	2020 年 2 月 6 日
<a href="#">使用 Amazon EventBridge 模式</a>	增加了對 Amazon EventBridge 模式支持	2019 年 12 月 1 日
<a href="#">AWS CDK</a>	服務的預覽版 AWS CDK 本。	2019 年 11 月 25 日
<a href="#">使用外部認證程序</a>	已新增有關使用外部認證程序取得認 AWS 證的資訊。	二零一九年九月二十
<a href="#">用 IntelliSense 於工作定義檔</a>	IntelliSense 添加了使用 Amazon ECS 任務定義文件的支持。	2019 年 9 月 24 日
<a href="#">使用者指南 AWS Toolkit for Visual Studio Code</a>	一般可用性版本。	2019 年 7 月 11 日
<a href="#">使用者指南 AWS Toolkit for Visual Studio Code</a>	更新文件結構，以清楚呈現、易於使用。	2019 年 7 月 3 日
<a href="#">正在安裝 AWS Toolkit for Visual Studio Code</a>	已新增有關安裝語言 SDKs 以支援各種工具鏈的資訊。	2019 年 6 月 12 日
<a href="#">設定您的工具鏈</a>	已新增有關設定各種工具鏈的資訊。	2019 年 6 月 12 日
<a href="#">初始版本</a>	AWS Toolkit for Visual Studio Code 使用者指南的初始版本。	2019 年 3 月 28 日

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。