使用者指南

AWS 適用於 VS 程式碼的工具組

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS 適用於 VS 程式碼的工具組: 使用者指南

Copyright © 2026 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 Developer 和 Amazon CodeWhisperer 2
下載 工具組 3
從 VS Code Marketplace 下載 Toolkit 3
來自 的其他 IDE 工具組 AWS 3
開始使用 4
安裝 Toolkit for VS 程式碼 4
先決條件 4
下載並安裝 AWS Toolkit for Visual Studio Code 4
選用先決條件 5
連線至 AWS 6
註冊 AWS 帳戶 6
先決條件 6
開啟登入面板 6
AWS 從 Toolkit 連線至 7
Amazon CodeCatalyst 的身分驗證 8
變更 AWS 區域 8
將區域新增至 AWS Explorer 9
從 AWS Explorer 隱藏區域 9
設定您的工具鏈 9
設定 .NET Core 的工具鏈 9
設定 Node.js 的工具鏈 10
設定 Python 的工具鏈 10
設定 Java 的工具鏈 10
設定 Go 的工具鏈 11
使用您的工具鏈 11
身分驗證與存取 12
IAM Identity Center 12
IAM 憑證 12
建立 IAM 使用者 13
從 建立共用登入資料檔案 AWS Toolkit for Visual Studio Code 13
新增其他登入資料設定檔 14

AWS 建置器 ID 15
使用外部登入資料程序 15
更新防火牆和閘道 15
AWS Toolkit for Visual Studio Code 端點 15
Amazon Q 外掛程式端點 16
Amazon Q Developer 端點 17
Amazon Q 程式碼轉換端點 17
身分驗證端點 17
身分端點 17
遙測 18
參考 18
使用 AWS 20
實驗功能 21
AWS Explorer 21
AWS Documents 22
AWS 文件入門 23
在 VS 程式碼中檢視文件、自動完成和驗證 23
Amazon CodeCatalyst 24
什麼是 Amazon CodeCatalyst？ 24
開始使用 CodeCatalyst 24
使用 CodeCatalyst 資源 25
使用開發環境 28
疑難排解 30
Amazon API Gateway 31
AWS App Runner 31
先決條件 32
定價 35
建立 App Runner 服務 35
管理 App Runner 服務 37
AWS 應用程式建置器 40
使用 AWS 應用程式建置器 40
AWS 基礎設施編寫器 43
使用 AWS 基礎設施編寫器 44
AWS CDK 45
AWS CDK 應用程式 45
CloudFormation 堆疊 47

刪除 CloudFormation 堆疊 47
建立 CloudFormation 範本 48
Amazon CloudWatch Logs 49
檢視 CloudWatch 日誌群組和日誌串流 50
使用 CloudWatch 日誌事件 51
搜尋日誌群組 52
CloudWatch Logs Live Tail 54
Amazon DocumentDB 56
使用 Amazon DocumentDB 56
Amazon EC2 61
使用 Amazon EC2 62
Amazon EC2 故障診斷 69
Amazon ECR 71
使用 Amazon ECR 72
建立 App Runner 服務 81
Amazon ECS 82
使用 IntelliSense 處理任務定義檔案 82
Amazon ECS Exec 83
Amazon EventBridge 85
使用 Amazon EventBridge 結構描述 86
AWS IAM Access Analyzer 88
使用 AWS IAM Access Analyzer 88
AWS IoT 92
AWS IoT 先決條件 92
AWS IoT 實物 92
AWS IoT 憑證 94
AWS IoT 政策 96
AWS Lambda 函數 99
使用 AWS Lambda 函數 99
AWS Lambda console 至 IDE 105
AWS Lambda LocalStack 支援 106
Lambda 遠端偵錯 111
Amazon Redshift 120
使用 Amazon Redshift 120
Amazon S3 124
使用 S3 資源 124

使用 S3 物件 126
Amazon SageMaker Unified Studio 129
AWS 無伺服器應用程式 129
開始使用 130
使用無伺服器土地 137
直接從程式碼執行和偵錯 Lambda 函數 139
執行並除錯本機 Amazon API Gateway 資源 142
除錯無伺服器應用程式的組態選項 145
疑難排解 151
AWS Systems Manager 153
假設和先決條件 153
Systems Manager 自動化文件的 IAM 許可 154
建立新的 Systems Manager 自動化文件 154
開啟現有的 Systems Manager 自動化文件 155
編輯 Systems Manager 自動化文件 155
發佈 Systems Manager 自動化文件 156
刪除 Systems Manager 自動化文件 157
執行 Systems Manager 自動化文件 157
疑難排解 157
AWS Step Functions 158
使用步驟函數 158
使用 Workflow Studio 161
威脅編寫器 165
使用威脅編寫器 166
Resources 166
存取資源的 IAM 許可 167
新增並與現有資源互動 167
建立和編輯資源 169
疑難排解 171
疑難排解最佳實務 171
設定檔 ... 在組態檔案中找不到 172
SAM json 結構描述：無法在 template.yaml 檔案中變更結構描述 173
安全 174
資料保護 174
文件歷史紀錄 176
..... clxxxii

AWS Toolkit for Visual Studio Code

這是 AWS Toolkit for VS 程式碼的使用者指南。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

Toolkit for VS Code 是 Visual Studio Code (VS Code) 編輯器的開放原始碼延伸。此擴充功能可讓開發人員更輕鬆地在本機開發、偵錯和部署使用 Amazon Web Services () 的無伺服器應用程式AWS。

主題

- 入門 AWS Toolkit for Visual Studio Code
- 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 Visual Studio 程式碼編輯器，請造訪 https：//https://code.visualstudio.com/。

Amazon Q Developer 和 Amazon CodeWhisperer

自 2024 年 4 月 30 日起，Amazon CodeWhisperer 現在是 Amazon Q Developer 的一部分，其中包括內嵌程式碼建議和 Amazon Q Developer 安全性掃描。從 VS Code Marketplace 下載 Amazon Q Developer IDE 延伸模組以開始使用。

如需 Amazon Q Developer 服務的詳細資訊，請參閱《Amazon Q Developer 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 程式碼

您可以透過 IDE AWS Toolkit for Visual Studio Code 中的 VS Code Marketplace 下載、安裝和設定 。如需詳細說明，請參閱本使用者指南入門主題中的下載和安裝一節。

從 VS Code Marketplace 下載 Toolkit

或者，您可以從 Web 瀏覽器導覽至 VS Code Marketplace 來下載 AWS Toolkit for Visual Studio Code 安裝檔案。

來自 的其他 IDE 工具組 AWS

除了 之外 AWS Toolkit for Visual Studio Code， AWS 還提供適用於 JetBrains 和 Visual Studio 的 IDE Toolkit。

AWS Toolkit for JetBrains 連結

- 依照此連結，從 JetBrains Marketplace 下載 AWS Toolkit for JetBrains 。
- 若要進一步了解 AWS Toolkit for JetBrains，請參閱 AWS Toolkit for JetBrains 使用者指南。

Toolkit for Visual Studio 連結

- 請遵循此連結，從 Visual Studio Marketplace 下載 Toolkit for Visual Studio。
- 若要進一步了解 Toolkit for Visual Studio，請參閱 Toolkit for Visual Studio 使用者指南。

入門 AWS Toolkit for Visual Studio Code

AWS Toolkit for Visual Studio Code 可讓您的服務 AWS 和資源直接從 VS Code 整合開發環境 (IDE) 取得。

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
- 連線至 AWS
- 變更 AWS 區域
- 設定您的工具鏈

安裝 AWS Toolkit for Visual Studio Code

先決條件

若要從 AWS Toolkit for Visual Studio Code VS Code 開始使用 ，必須符合下列權限。若要進一步了解如何從 存取所有可用的 AWS 服務和資源 AWS Toolkit for Visual Studio Code，請參閱本指南的 the section called “選用先決條件”一節。

- VS 程式碼需要 Windows、macOS 或 Linux 作業系統。
- AWS Toolkit for Visual Studio Code 需要您從 VS Code 1.73.0 版或更新版本使用 。

如需 VS 程式碼或下載最新版 VS 程式碼的詳細資訊，請參閱 VS 程式碼下載網站。

下載並安裝 AWS Toolkit for Visual Studio Code

您可以透過 IDE AWS Toolkit for Visual Studio Code 中的 VS Code Marketplace 下載、安裝和設定 。或者，您可以從 Web 瀏覽器導覽至 VS Code Marketplace 來下載 AWS Toolkit for Visual Studio Code 安裝檔案。

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

1. 使用以下連結在 VS Code IDE 中開啟 AWS Toolkit for Visual Studio Code 延伸模組：開啟 VS Code Marketplace。

> **Note**
>
> 如果您的機器上尚未執行 VS 程式碼，則載入 VS 程式碼時，此操作可能需要一些時間。

2. 從 VS Code Marketplace 中的 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 程式碼以完成安裝程序。

選用先決條件

您必須先擁有下列項目 AWS Toolkit for Visual Studio Code，才能使用 的特定功能：

- Amazon Web Services (AWS) 帳戶： AWS 帳戶並非使用 的必要條件 AWS Toolkit for Visual Studio Code，但功能在沒有帳戶的情況下會受到大幅限制。若要取得 AWS 帳戶，請前往 AWS 首頁。選擇建立 AWS 帳戶或完成註冊（如果您之前已造訪過網站）。
- 程式碼開發 – 您要使用之語言的相關 SDK。您可以從以下連結下載，或使用您最愛的套件管理工具：
  - .NET 開發套件：https://dotnet.microsoft.com/download
  - Node.js 軟體開發套件：https://nodejs.org/en/download
  - Python 軟體開發套件：https://www.python.org/downloads
  - Java 開發套件： https://aws.amazon.com/corretto/
  - Go 開發套件： https://golang.org/doc/install
- AWS SAM CLI – 這是一個 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。不過，我們建議您安裝它（和 Docker，如下所述），因為它是任何 AWS Serverless Application Model (AWS SAM) 功能的必要項目，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》中的安裝 AWS SAM CLI。
- Docker – CLI AWS SAM 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。
- Package Manager – 套件管理員，可讓您下載和共用應用程式碼。
  - .NET：NuGet
  - Node.js：npm

- Python：pip
- Java：Gradle 或 Maven

連線至 AWS

大多數 Amazon Web Services (AWS) 資源都是透過 AWS 帳戶管理。使用 AWS 不需要帳戶 AWS Toolkit for Visual Studio Code，但 Toolkit 函數在沒有連線的情況下受到限制。

如果您先前已透過其他服務 AWS （例如 AWS Command Line Interface) 設定 AWS 帳戶和身分驗證，則 AWS Toolkit for Visual Studio Code 會自動偵測您的登入資料。

註冊 AWS 帳戶

若要開始使用 AWS，您需要 AWS 帳戶。有兩種註冊方式 AWS。如果您想要完全控制帳戶組態和內容的存放位置，或者如果您有受管制的工作負載，例如 HIPAA 或 FedRAMP，請使用註冊 AWS （進階）。如果您是新使用者，並想要使用您已擁有的登入來登入， AWS 並在選取 中建立具有預先設定預設值 AWS 的環境AWS 區域，請使用註冊 AWS。

先決條件

如果您是初次使用 AWS 或尚未建立 帳戶，則有 2 個主要步驟可將 AWS Toolkit for Visual Studio Code 與 AWS 您的帳戶連線：

1. 設定身分驗證：有 3 種主要方法可從 驗證 AWS 您的帳戶 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身分驗證和存取主題。
2. AWS 從 Toolkit 向 驗證：您可以從 Toolkit 中與 AWS 您的帳戶連線，方法是完成本使用者指南以下章節中的程序。

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 Explorer AWS 開啟工具組登入面板：

1. 從 中 AWS Toolkit for Visual Studio Code展開 EXPLORER。
2. 選取 ... 圖示，展開更多動作...選單。
3. 從更多動作...功能表中，選擇連線至 AWS以開啟AWS 工具組登入面板。

若要使用 VS Code 命令面板開啟 AWS Toolkit Sign In 面板：

1. 按 **Shift+Command+P**(**Ctrl+Shift+P** Windows) 開啟命令貨盤。
2. 在搜尋欄位中輸入 **AWS: Add a New Connection** 。
3. 選取 **AWS: Add a New Connection**以開啟AWS 工具組登入面板。

AWS 從 Toolkit 連線至

驗證並與 SSO 連線

若要 AWS 使用 驗證和連線 AWS IAM Identity Center，請完成下列程序。

> **Note**
>
> 使用 AWS Builder ID 或 IAM Identity Center 進行身分驗證會在您的預設 Web 瀏覽器中啟動 AWS 授權入口網站。每次您的登入資料過期時，都必須重複此程序，才能續約 AWS 帳戶與之間的連線 AWS Toolkit for Visual Studio Code。

驗證並與 IAM Identity Center AWS 連線

1. 從AWS 工具組登入面板中，選擇人力索引標籤，然後選取繼續按鈕以繼續。
2. 從使用 IAM Identity Center 登入面板中，輸入組織的開始 URL。此 URL 是由您公司的管理員或服務台提供給您。
3. 從下拉式選單中選取 your AWS Region。這是託管您的身分目錄 AWS 的區域。
4. 選擇繼續按鈕，並確認您想要在預設 Web 瀏覽器中開啟AWS 授權請求網站。
5. 遵循預設 Web 瀏覽器中的提示，授權程序完成時會通知您，關閉瀏覽器是安全的，並返回 VS 程式碼。

使用 IAM 登入資料進行驗證和連線

若要 AWS 使用 IAM 登入資料來驗證和連線 ，請完成下列程序。

使用 IAM 登入資料進行驗證和連線

1. 從AWS 工具組登入面板中，選擇 IAM 登入資料，然後選取繼續按鈕以繼續。

2. 在提供的欄位中輸入**Secret Key**您 AWS 帳戶的 **Profile NameAccess Key**、 和 ，然後選擇繼續按鈕，將設定檔新增至您的組態檔案，並將 Toolkit 與您的 AWS 帳戶連線。
3. 當身分驗證完成且已建立連線時， Toolkit AWS Explorer 會更新以顯示您的 AWS 服務和資源。

Amazon CodeCatalyst 的身分驗證

若要從 Toolkit 開始使用 CodeCatalyst，請驗證您的 AWS 建置器 ID 或 IAM Identity Center 憑證並與其連線。

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 Builder ID 驗證和連線

1. 從AWS 工具組登入面板中，選擇人力索引標籤，然後選取繼續按鈕以繼續。
2. 在使用 SSO 登入面板頂端，選擇略過以登入連結。
3. 遵循預設 Web 瀏覽器中的提示，授權程序完成時會通知您，關閉瀏覽器是安全的，並返回 VS 程式碼。

驗證並與 IAM Identity Center 連線

1. 從AWS 工具組登入面板中，選擇人力索引標籤，然後選取繼續按鈕以繼續。
2. 從使用 IAM Identity Center 登入面板中，輸入組織的開始 URL。此 URL 是由您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 區域。這是託管您的身分目錄 AWS 的區域。
4. 選擇繼續按鈕，並確認您想要在預設 Web 瀏覽器中開啟AWS 授權請求網站。
5. 遵循預設 Web 瀏覽器中的提示，授權程序完成時會通知您，關閉瀏覽器是安全的，並返回 VS 程式碼。

變更 AWS 區域

AWS 區域指定 AWS 資源的管理位置。當您從 連線到 AWS 您的帳戶時 AWS Toolkit for Visual Studio Code，系統會自動在 AWS Explorer 中顯示您的預設 AWS 區域。

下列各節說明如何從 AWS Explorer 新增或隱藏區域。

將區域新增至 AWS Explorer

完成下列程序，將區域新增至 AWS Explorer。

1. 從 VS 程式碼中，展開主選單上的檢視，然後選擇命令調色盤，以開啟命令調色盤。 或使用下列快速鍵：
   - Windows 和 Linux – 按 **Ctrl+Shift+P**。
   - macOS – 按 **Shift+Command+P**。
2. 從命令面板中，搜尋**AWS: Show or Hide Regions**並選擇 AWS：顯示或隱藏區域，以顯示可用區域的清單。
3. 從清單中，選取您要新增至 AWS Explorer AWS 的區域。
4. 選擇確定按鈕以確認您的選擇並更新 AWS Explorer。

從 AWS Explorer 隱藏區域

若要從 AWS Explorer 檢視隱藏區域，請完成下列程序。

1. 從 AWS Explorer 找到您要隱藏 AWS 的區域。
2. 開啟您要隱藏之區域的內容選單 （ 按一下滑鼠右鍵 ）。
3. 選擇顯示或隱藏區域以開啟 AWS：顯示或隱藏 VS 程式碼中的區域選項。
4. 取消選取要在 AWS Explorer 檢視中隱藏的區域。

設定您的工具鏈

AWS Toolkit for Visual Studio Code 支援所有 AWS 服務的多種語言。下列各節說明如何設定不同語言的工具鏈。

設定 .NET Core 的工具鏈

1. 確保您已安裝 AWS Toolkit for VS Code。
2. 安裝 C# 延伸模組。此擴充功能可讓 VS 程式碼偵錯 .NET Core 應用程式。
3. 開啟 AWS Serverless Application Model (AWS SAM) 應用程式，或建立一個。
4. 開啟包含 的資料夾`template.yaml`。

設定 Node.js 的工具鏈

1. 請確定您已安裝 AWS Toolkit for VS Code。
2. 開啟 AWS SAM 應用程式，或建立一個應用程式。
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 編譯器必須全域安裝，或安裝在專案的 中`package.json`。

設定 Python 的工具鏈

1. 請確定您已安裝 AWS Toolkit for VS Code。
2. 安裝適用於 Visual Studio Code 的 Python 延伸模組。此延伸模組可讓 VS 程式碼偵錯 Python 應用程式。
3. 開啟 AWS SAM 應用程式，或建立一個應用程式。
4. 開啟包含 的資料夾`template.yaml`。
5. 在應用程式的根目錄開啟終端機，並透過執行 `virtualenv`來設定 `python -m venv ./.venv`。

> **Note**
>
> 您只需要為每個系統設定`virtualenv`一次。

6. 執行下列其中一項`virtualenv`來啟用：
   - Linux 或 macOS (Bash)： `source ./.venv/bin/activate`
   - Windows (PowerShell)： `./.venv/Scripts/Activate.ps1`
   - Windows（命令提示）： `.venv\Scripts\activate.bat`

設定 Java 的工具鏈

1. 請確定您已安裝 AWS Toolkit for VS Code。

2. 安裝 Java 延伸模組和 Java 11。此擴充功能可讓 VS 程式碼辨識 Java 函數。
3. 安裝 Java 除錯器延伸模組。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，或建立一個應用程式。
5. 開啟包含 的資料夾`template.yaml`。

設定 Go 的工具鏈

1. 確保您已安裝 AWS Toolkit for VS Code。
2. 偵錯 Go Lambda 函數需要 Go 1.14 或更高版本。
3. 安裝 Go 延伸模組。

> **Note**
>
> 偵錯 Go1.15+ 執行時間需要 0.25.0 版或更新版本。

4. 使用命令調色盤安裝 Go 工具：
   a. 從命令調色盤中，選擇 `Go: Install/Update Tools`。
   b. 從一組核取方塊中，選取 `dlv`和 `gopls`。
5. 開啟 AWS SAM 應用程式，或建立一個應用程式。
6. 開啟包含 的資料夾`template.yaml`。

使用您的工具鏈

設定工具鏈後，您可以使用它來執行或偵錯 AWS SAM 應用程式。

的身分驗證和存取 AWS Toolkit for Visual Studio Code

您不需要向 進行身分驗證 AWS ，即可開始使用 AWS Toolkit for Visual Studio Code。不過，大多數 AWS 資源是透過 AWS 帳戶管理。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 登入資料進行身分驗證。

下列主題包含每個登入資料類型的其他詳細資訊。

如需如何使用 AWS Toolkit for Visual Studio Code 現有登入資料在 AWS 中連線至 的詳細資訊，請參閱《 使用者指南》中的連線至 AWS主題。

主題

- AWS IAM Identity Center
- AWS IAM 登入資料
- AWS 開發人員的建置器 ID
- 使用外部登入資料程序
- 更新防火牆和閘道以允許存取

AWS IAM Identity Center

AWS IAM Identity Center 是管理 AWS 帳戶身分驗證的建議最佳實務。

如需如何設定軟體開發套件 (SDKs) 的 IAM Identity Center 的詳細說明，請參閱 AWS SDKs和工具參考指南中的 IAM Identity Center 身分驗證一節。

如需如何驗證 AWS 工具組並將其與您現有的 IAM Identity Center 登入資料連線的詳細資訊，請參閱《 使用者指南》中的連線至 AWS主題。

AWS IAM 登入資料

AWS 透過本機儲存的存取金鑰，向 AWS 您的帳戶進行 IAM 登入資料驗證。

如需如何驗證 AWS 工具組並將其與您現有的 IAM AWS 登入資料連線的詳細資訊，請參閱本使用者指南中的連線至 AWS主題。

下列各節說明如何設定 IAM 登入資料，以從 驗證 AWS 您的帳戶 AWS Toolkit for Visual Studio Code。

> **Important**
>
> 設定 IAM 登入資料以向 AWS 您的帳戶進行身分驗證之前，請注意：
>
> - 如果您已透過其他服務 AWS （例如 AWS CLI) 設定 IAM 登入資料，則 AWS Toolkit for Visual Studio Code 會自動偵測這些登入資料，並在 VS 程式碼中提供這些登入資料。
> - AWS 建議使用 IAM Identity Center 身分驗證。如需 IAM AWS 最佳實務的其他資訊，請參閱 AWS 《Identity and Access Management 使用者指南》中 IAM 的安全最佳實務一節。
> - 為避免安全風險，在開發專用軟體或使用真實資料時，請勿使用 IAM 使用者進行身分驗證。而是改為搭配使用聯合功能和身分提供者，例如《AWS IAM Identity Center 使用者指南》中的什麼是 IAM Identity Center？。

建立 IAM 使用者

您必須先完成步驟 1：建立 IAM 使用者和步驟 2：使用 AWS SDKs 和工具參考指南中的長期憑證主題進行身分驗證，才能設定 AWS Toolkit for Visual Studio Code 來驗證 AWS 您的帳戶。

> **Note**
>
> 步驟 3：更新 SDK 和工具參考指南中的共用登入資料檔案AWS SDKs是選用的。
> 如果您完成步驟 3， 會在下列 the section called “從 建立共用登入資料檔案 AWS Toolkit for Visual Studio Code” 期間 AWS Toolkit for Visual Studio Code 自動偵測您的登入資料。
> 如果您尚未完成步驟 3， 會 AWS Toolkit for Visual Studio Code 逐步引導您建立 的程序`credentials file`，如the section called “從 建立共用登入資料檔案 AWS Toolkit for Visual Studio Code”下方所述。

從 建立共用登入資料檔案 AWS Toolkit for Visual Studio Code

您帳戶的共用組態檔案和共用登入資料檔案存放區組態和登入資料資訊 AWS 。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 建立共用登入資料檔案 AWS Toolkit for Visual Studio Code

1. 按 **Shift+Command+P**(**Ctrl+Shift+P** Windows) 開啟命令貨盤。
2. 在搜尋欄位中輸入 **AWS: Add a New Connection** 。
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 登入資料，然後選取繼續按鈕以繼續。
5. 在提供的欄位中輸入**Secret Key**您 AWS 帳戶的 **Profile NameAccess Key**、 和 ，然後選擇繼續按鈕，將設定檔新增至您的組態檔案，並將 Toolkit 與您的 AWS 帳戶連線。
6. 當身分驗證完成且已建立連線時， 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，並導致身分驗證失敗。
>
> ```
> [Profile_Name]
> xaws_access_key_id= AKIAI44QH8DHBEXAMPLE
> xaws_secret_access_key= wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
> ```
>
> 以下是為回應失敗的身分驗證嘗試而產生的日誌訊息範例。
>
> ```
> 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.
> ```

新增其他登入資料設定檔

您可以將多個登入資料新增至您的組態檔案。若要這樣做，請開啟命令面板，然後選擇AWS 工具組建立登入資料設定檔。這會開啟登入資料檔案。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 開發人員的建置器 ID

AWS Builder ID 是某些 AWS 服務為選用或必要的額外 AWS 帳戶。如需 AWS 建置器 ID 身分驗證方法的詳細資訊，請參閱《 登入使用者指南》中的使用 AWS 建置器 ID AWS 登入主題。

如需如何驗證 AWS 工具組並將其與您現有的 AWS 建置器 ID 連線的詳細資訊，請參閱本使用者指南中的連線至 AWS主題。

使用外部登入資料程序

您可以修改 AWS， AWS Toolkit for Visual Studio Code 為 未直接支援的登入資料程序設定 `shared config file`。

對於 AWS Toolkit for Visual Studio Code 和 ，修改`shared config file`登入資料程序的 是相同的 AWS Command Line Interface。如需如何設定外部登入資料的詳細資訊，請參閱AWS Command Line Interface 《 使用者指南》中的具有外部程序主題的採購登入資料。

更新防火牆和閘道以允許存取

如果您使用 Web 內容篩選解決方案篩選特定 AWS 網域或 URL 端點的存取權，則必須允許列出下列端點，才能存取透過 和 Amazon Q AWS Toolkit for Visual Studio Code 提供的所有服務和功能。

AWS Toolkit for Visual Studio Code 端點

以下是需要允許的特定 AWS Toolkit for Visual Studio Code 端點和參考清單。

Endpoint

```
https://idetoolkits.amazonwebservices.com/endpoints.json
```

託管檔案

```
https://idetoolkits-hostedfiles.amazonaws.com/Notifications/VSCode/startup/1.x.json
https://idetoolkits-hostedfiles.amazonaws.com/Notifications/VSCode/emergency/1.x.json
```

結構描述支援

```
https://raw.githubusercontent.com/aws/serverless-application-model/main/samtranslator/
schema/schema.json
https://api.github.com/repos/devfile/api/releases/latest
https://raw.githubusercontent.com/devfile/api/${devfileSchemaVersion}/schemas/latest/
devfile.json
```

cSharpSamDebug 安裝指令碼

```
https://aka.ms/getvsdbgps1
https://aka.ms/getvsdbgsh
```

Amazon Q 外掛程式端點

以下是需要允許列出的 Amazon Q 外掛程式特定端點和參考清單。

```
https://idetoolkits-hostedfiles.amazonaws.com/*    (Plugin for configs)
https://idetoolkits.amazonwebservices.com/*   (Plugin for endpoints)
https://aws-toolkit-language-servers.amazonaws.com/*  (Language Server Process)
https://client-telemetry.us-east-1.amazonaws.com/ (Telemetry)
https://cognito-identity.us-east-1.amazonaws.com    (Telemetry)
```

```
https://aws-language-servers.us-east-1.amazonaws.com (Language Server Process)
```

Amazon Q Developer 端點

以下是需要允許列出的 Amazon Q Developer 特定端點和參考清單。

```
https://codewhisperer.us-east-1.amazonaws.com (Inline,Chat, QSDA,...)
https://q.us-east-1.amazonaws.com (Inline,Chat, QSDA....)
https://desktop-release.codewhisperer.us-east-1.amazonaws.com/ (Download url for CLI.)
https://specs.q.us-east-1.amazonaws.com (Url for autocomplete specs used by CLI)
* aws-language-servers.us-east-1.amazonaws.com (Local Workspace context)
```

Amazon Q 程式碼轉換端點

以下是需要允許列出的 Amazon Q Code Transform 特定端點和參考清單。

```
https://docs.aws.amazon.com/amazonq/latest/qdeveloper-ug/security_iam_manage-access-
with-policies.html
```

身分驗證端點

以下是需要允許列出的身分驗證端點和參考清單。

```
[Directory ID or alias].awsapps.com
* oidc.[Region].amazonaws.com
*.sso.[Region].amazonaws.com
*.sso-portal.[Region].amazonaws.com
*.aws.dev
*.awsstatic.com
*.console.aws.a2z.com
*.sso.amazonaws.com
```

身分端點

下列清單包含特定於身分的端點，例如 AWS IAM Identity Center 和 AWS Builder ID。

AWS IAM Identity Center

如需 IAM Identity Center 所需端點的詳細資訊，請參閱AWS IAM Identity Center《 使用者指南》中的啟用 IAM Identity Center 主題。

企業 IAM Identity Center

```
https://[Center director id].awsapps.com/start (should be permitted to initiate auth)
https://us-east-1.signin.aws (for facilitating authentication, assuming IAM Identity
 Center is in IAD)
https://oidc.(us-east-1).amazonaws.com
https://log.sso-portal.eu-west-1.amazonaws.com.
https://portal.sso.eu-west-1.amazonaws.com
```

AWS 建置器 ID

```
https://view.awsapps.com/start (must be blocked to disable individual tier)
https://codewhisperer.us-east-1.amazonaws.com and q.us-east-1.amazonaws.com (should be
 permitted)
```

遙測

以下是需要允許列出的遙測特定端點。

```
https://telemetry.aws-language-servers.us-east-1.amazonaws.com/
https://client-telemetry.us-east-1.amazonaws.com
```

參考

以下是端點參考的清單。

```
idetoolkits-hostedfiles.amazonaws.com.
cognito-identity.us-east-1.amazonaws.com.
```

```
amazonwebservices.gallery.vsassets.io.
eu-west-1.prod.pr.analytics.console.aws.a2z.com.
prod.pa.cdn.uis.awsstatic.com.
portal.sso.eu-west-1.amazonaws.com.
log.sso-portal.eu-west-1.amazonaws.com.
prod.assets.shortbread.aws.dev.
prod.tools.shortbread.aws.dev.
prod.log.shortbread.aws.dev.
a.b.cdn.console.awsstatic.com.
assets.sso-portal.eu-west-1.amazonaws.com.
oidc.eu-west-1.amazonaws.com.
aws-toolkit-language-servers.amazonaws.com.
aws-language-servers.us-east-1.amazonaws.com.
idetoolkits.amazonwebservices.com.
```

使用 AWS 服務和工具

AWS Toolkit for Visual Studio Code 可讓您直接在 VS Code 中使用 AWS 服務、工具和資源。以下是涵蓋每個 Toolkit for VS Code 服務及其功能的指南主題清單。選擇服務或工具，以取得其功能、設定方式，以及使用基本功能的詳細資訊。

主題

- 使用實驗功能
- 在 AWS Explorer 中使用 AWS 服務
- AWS Documents
- Amazon CodeCatalyst for VS 程式碼
- 使用 Amazon API Gateway
- AWS App Runner 搭配 使用 AWS Toolkit for Visual Studio Code
- AWS 應用程式建置器
- AWS 基礎設施編寫器
- AWS CDK 適用於 VS 程式碼的
- 使用 AWS CloudFormation 堆疊
- 使用 使用 CloudWatch Logs AWS Toolkit for Visual Studio Code
- Amazon DocumentDB
- Amazon Elastic Compute Cloud
- 使用 Amazon Elastic Container Registry
- 使用 Amazon Elastic Container Service
- 使用 Amazon EventBridge
- AWS IAM Access Analyzer
- 在 AWS IoT 中使用 AWS Toolkit for Visual Studio Code
- AWS Lambda 函數
- Toolkit for VS 程式碼中的 Amazon Redshift
- 使用 Amazon S3
- Amazon SageMaker Unified Studio for VS 程式碼

- 使用無伺服器應用程式
- 使用 Systems Manager 自動化文件
- AWS Step Functions
- 使用威脅編寫器
- 使用 資源

使用實驗功能

實驗功能可在正式發佈 AWS Toolkit for Visual Studio Code 之前，提早存取 中的功能。

> **Warning**
>
> 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。實驗功能可能會從 中移除，AWS Toolkit for Visual Studio Code 恕不另行通知。

您可以在 VS Code IDE 的設定窗格AWS 的工具組區段中，為特定 AWS 服務啟用實驗性功能。

1. 若要編輯 VS 程式碼中的 AWS 設定，請選擇檔案、偏好設定、設定。
2. 在設定窗格中，展開延伸模組，然後選擇AWS 工具組。
3. 在 AWS：Experiments 下，選取您要在發行之前存取之實驗性功能的核取方塊。如要關閉實驗功能，請清除相關核取方塊。

在 AWS Explorer 中使用 AWS 服務

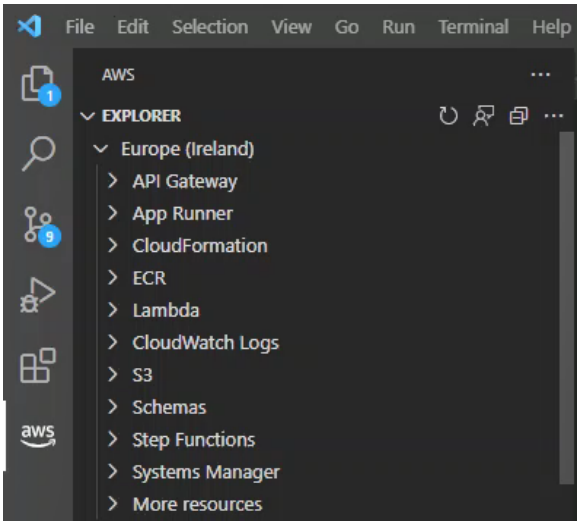
AWS Explorer 可讓您檢視使用 時可以使用的一些 AWS 服務 AWS Toolkit for Visual Studio Code。

本節提供如何在 VS 程式碼中存取和使用 AWS Explorer 的相關資訊。其假設您已在系統上安裝並設定 Toolkit for VS 程式碼。

部分重點提示：

- 如果工具組已安裝並正確設定，您應該會在 AWS Explorer 中看到項目。若要查看 AWS Explorer，請選擇活動列中的 AWS 圖示。

  例如：



- 某些功能需要特定 AWS 許可。例如，若要查看 AWS 帳戶中的 AWS Lambda 函數，您在 中設定的登入資料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 AWS Explorer 中未立即顯示 AWS 的服務互動，您可以前往更多資源，並從數百個可新增至界面的資源中進行選擇。

  例如，您可以從可用的資源類型中選擇 AWS Toolkit：CodeArtifact：：Repository。將此資源類型新增至更多資源後，您可以展開 項目，以檢視資源清單，這些資源清單會建立具有自己的屬性和屬性的不同 CodeArtifact 儲存庫。此外，您可以在 JSON 格式範本中描述資源的屬性和屬性，這些範本可以儲存以在 AWS 雲端中建立新資源。

AWS Documents

AWS Toolkit for Visual Studio Code 支援 AWS Serverless Application Model `JSON Schema`適用於 的 `AWS SAM templates`，透過直接在 VS Code 中啟用定義、自動完成和驗證來增強範本撰寫體驗。AWS 文件支援所有 AWS SAM 和 CloudFormation 資源。如需其他詳細資訊，請參閱下列資源：

- 如需 JSON 結構描述的特定資訊，請參閱 JSON 結構描述 JSON-Schema.org 網站。
- 如需 AWS SAM 範本的其他資訊，請參閱《 AWS Serverless Application Model開發人員指南》中的AWS SAM 範本結構主題。
- 如需 AWS 資源和屬性類型的其他資訊，請參閱CloudFormation《 使用者指南》中的AWS 資源和屬性類型參考主題。
- 如需 AWS Toolkit AWS SAM 使用之結構描述的詳細資訊，請參閱 AWS GitHub AWS Serverless Application Model 儲存庫中的結構描述。

AWS 文件入門

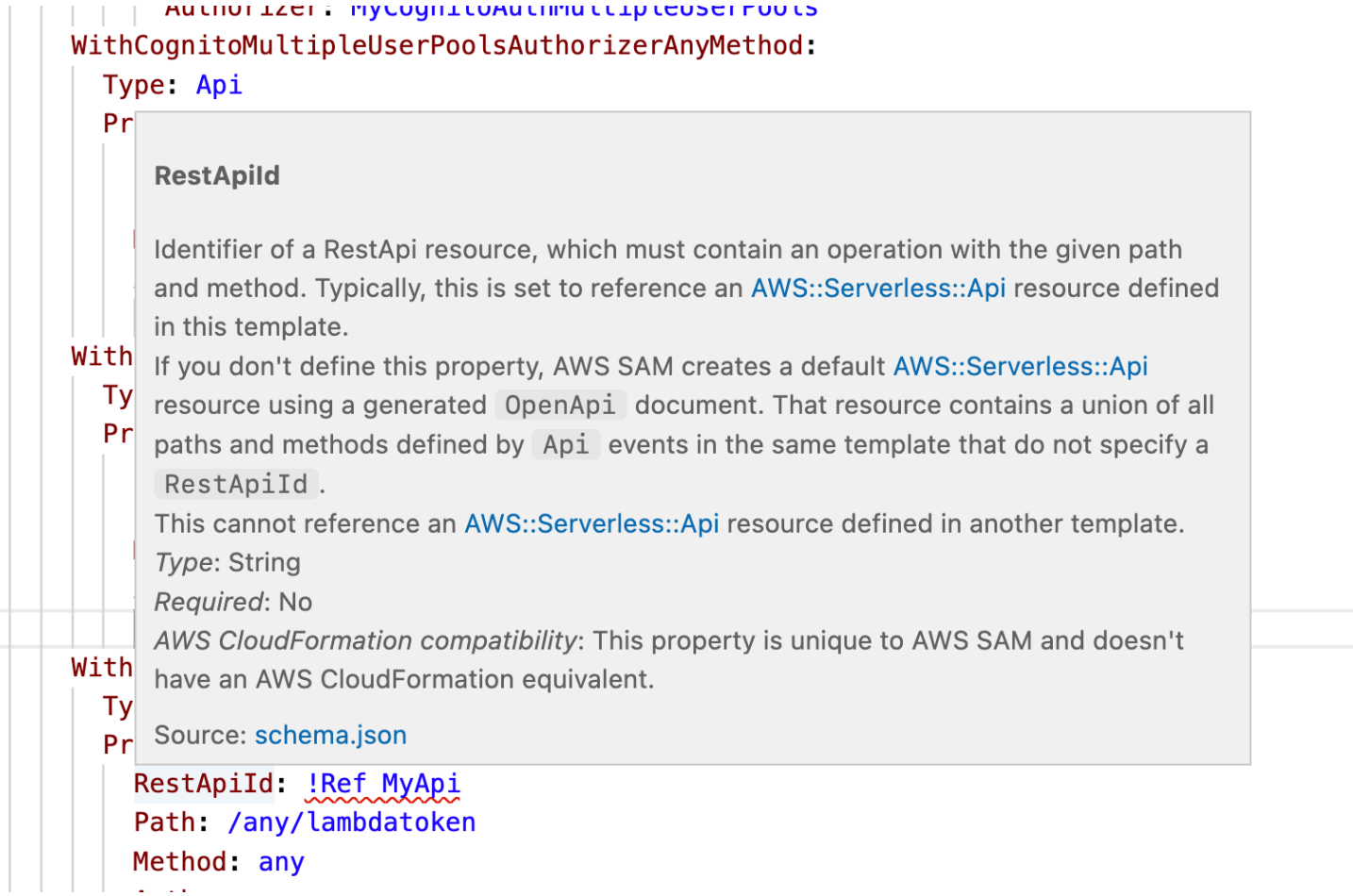
若要開始使用 VS Code 中的 AWS 文件，請從 IDE 或 VS Code Marketplace 安裝 AWS Toolkit for Visual Studio Code 擴充功能，然後開啟任何 AWS SAM 範本。

在 VS 程式碼中檢視文件、自動完成和驗證

檢視文件、自動完成和驗證是 AWS Toolkit 隨附的功能。如需這些功能在 VS 程式碼中的外觀範例，請參閱下圖。

- 若要檢視開啟 AWS SAM 範本的文件，請將指標停留在文件中的行輸入上。
- 對於自動完成，請開始在 AWS SAM 範本中輸入 ，以根據您的輸入啟用具有建議的快顯視窗。
- 您的 AWS SAM 範本會自動掃描以進行驗證，而錯誤會以燈泡圖示反白顯示，您可以選取該圖示以取得其他建議。

如需這些功能在 VS 程式碼中的外觀範例，請參閱下圖。



Amazon CodeCatalyst for VS 程式碼

什麼是 Amazon CodeCatalyst？

Amazon CodeCatalyst 是提供給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 CodeCatalyst 資源。您也可以使用 AWS Toolkit 啟動執行 VS Code 的開發環境虛擬運算環境，直接在雲端中運作。如需 CodeCatalyst 服務的詳細資訊，請參閱《Amazon CodeCatalyst 使用者指南》。

下列主題說明如何使用 CodeCatalyst 連接 VS Code，以及如何從 Toolkit for VS Code 使用 CodeCatalyst。

主題

- CodeCatalyst 和 Toolkit for VS Code 入門
- 在 VS Code 中使用 Amazon CodeCatalyst 資源
- 在開發環境中使用 Toolkit
- 對 Amazon CodeCatalyst 和 VS Code 進行故障診斷

CodeCatalyst 和 Toolkit for VS Code 入門

若要開始使用 VS Code 中的 CodeCatalyst，請遵循下列程序。

主題

- 建立 CodeCatalyst 帳戶
- 將 AWS Toolkit 與 CodeCatalyst 連線

建立 CodeCatalyst 帳戶

您必須有作用中的 AWS 建置器 ID 或 AWS IAM Identity Center 登入資料，才能從 Toolkit for VS Code 連線至 CodeCatalyst。若要進一步了解 AWS 建置器 ID、IAM Identity Center 和 CodeCatalyst 登入資料，請參閱 CodeCatalyst 使用者指南中的使用 CodeCatalyst 設定一節。 CodeCatalyst

將 AWS Toolkit 與 CodeCatalyst 連線

若要將 AWS Toolkit 與您的 CodeCatalyst 帳戶連線，請參閱本使用者指南的 AWS連線至主題中的 Amazon CodeCatalyst 身分驗證一節。

在 VS Code 中使用 Amazon CodeCatalyst 資源

下列各節提供可從 Toolkit for VS Code 取得的 Amazon CodeCatalyst 資源管理功能的概觀。

如需有關開發環境以及如何從 CodeCatalyst 存取這些環境的詳細資訊，請參閱《Amazon CodeCatalyst 使用者指南》中的開發環境一節。

下列各節說明如何從 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 CodeCatalyst 開發環境
- 從第三方儲存庫建立開發環境
- VS Code 中的 CodeCatalyst 命令

複製儲存庫

CodeCatalyst 是以雲端為基礎的服務，需要您連線到雲端才能處理 CodeCatalyst 專案。如果您偏好在本機處理專案，您可以將 CodeCatalyst 儲存庫複製到本機機器，並在下次連線至雲端時與線上 CodeCatalyst 專案同步。

若要使用 AWS Toolkit 將儲存庫從 CodeCatalyst 帳戶複製到 VS 程式碼，請完成下列步驟：

> **Note**
>
> 如果您要從第三方服務複製儲存庫，系統可能會提示您使用該服務的登入資料進行身分驗證。複製儲存庫時，VS 程式碼會在複製儲存庫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？ 訊息隨即出現。

1. 從 Toolkit for VS Code 中，展開 DEVELOPER TOOLS 瀏覽器。
2. 展開 CodeCatalyst，選擇複製儲存庫。
3. 從選取 CodeCatalyst 儲存庫對話方塊中，搜尋您要複製的儲存庫，然後選取它以開啟選擇要複製的資料夾對話方塊。
4. 選擇選取儲存庫位置以關閉提示，並開始複製儲存庫。

5. 從對話方塊視窗中，選擇下列其中一項以完成複製程序：

   - 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 程式碼視窗中開啟您的儲存庫，請選擇在新視窗中開啟。
   - 若要在不開啟儲存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 程式碼中開啟現有的開發環境，請完成下列步驟。

> **Note**
>
> 選取開發環境會開啟開發環境，以開始將 VS 程式碼與 CodeCatalyst 連線的程序。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，狀態會更新為 程序失敗原因的相關資訊，以及開啟程序日誌的連結。
> - 如果程序成功，您的開發環境會從 VS 程式碼在新視窗中開啟。

1. 從 Toolkit for VS Code 中，展開 DEVELOPER TOOLS 總管。
2. 展開 CodeCatalyst 並選擇開啟開發環境，以在 VS 程式碼中開啟選取 CodeCatalyst 開發環境對話方塊。
3. 從選取 CodeCatalyst 開發環境對話方塊中，選擇您要開啟的開發環境。

建立 CodeCatalyst 開發環境

若要建立新的開發環境，請完成下列步驟：

> **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。
> - 開發環境會持續儲存您的工作。這表示您的開發環境可以停止，而不會遺失您的工作。停止開發環境可降低讓開發環境保持正常運作所需的成本。

- 儲存是建立開發環境後無法變更的唯一設定。
- VS 程式碼會顯示您的開發環境在狀態視窗中建立的進度。建立開發環境後，VS 程式碼會在新視窗中開啟開發環境，以及信任此資料夾中檔案的作者嗎？ 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 DEVELOPER TOOLS 瀏覽器。
2. 展開 CodeCatalyst，然後選擇建立開發環境選項，以在 VS 程式碼中開啟建立 CodeCatalyst 開發環境選單。
3. 從來源碼區段中，選擇下列其中一個選項：
   - 使用現有的 CodeCatalyst 儲存庫：從現有的 CodeCatalyst 儲存庫建立開發環境。您必須選取 CodeCatalyst 專案和分支。
   - 建立空的開發環境：建立空的開發環境。
4. （選用）從別名區段中，輸入開發環境的備用名稱。
5. （選用）從開發環境組態區段中，變更下列設定以符合您的特定需求。
   - 運算：選擇編輯運算以變更指派給您系統的處理能力和 RAM 數量。
   - 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：選擇編輯儲存體大小，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。

連結到第三方存儲庫做為來源的作業，是在 CodeCatalyst 中的專案層級處理。如需如何將第三方存儲庫連線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。

VS Code 中的 CodeCatalyst 命令

還有其他 VS Code 命令會指派給 CodeCatalyst 相關功能，這些功能不會直接顯示在 AWS Toolkit 中。

若要從命令調色盤檢視指派給 CodeCatalyst 的命令清單，請完成下列步驟：

1. 從 Toolkit for VS Code 中，展開 DEVELOPER TOOLS 探索器。
2. 選擇顯示 CodeCatalyst 命令以開啟命令調色盤，其中包含 的預先填入搜尋`CodeCatalyst`。
3. 從清單中選擇 CodeCatalyst 命令以啟用它。

在開發環境中使用 Toolkit

開發環境是 Amazon CodeCatalyst 的虛擬運算環境。下列各節說明如何使用 從開發環境建立、啟動和運作 AWS Toolkit for Visual Studio Code。

如需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 devfile 設定開發環境

`devfile` 規格是 YAML 的開放標準格式，可用於定義開發環境的組態。每個開發環境都有一個 devfile。如果您在沒有儲存庫或從不包含 devfile 的儲存庫建立開發環境，預設會自動套用到來源。您可以從 CodeCatalyst 或您的 IDE 更新 Devfile。在 VS Code 的本機或遠端執行個體中更新 devfile 的程序是相同的，但如果您在本機更新 devfile，您必須在更新生效之前將更新推送到您的來源儲存庫。

如需使用 devfile 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您的 devfile。

> **Important**
>
> 如果您`Devfile`從 VS 程式碼編輯 ，請注意下列事項：
>
> - 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。
> - 如果您在根資料夾中建立沒有 devfile 的開發環境，或未與來源儲存庫建立關聯的開發環境，則當您建立開發環境時，會產生具有預設組態設定的 devfile。
> - 如需如何定義和設定 的說明`Devfile`，請參閱 https：// 網站上的新增命令文件。 devfile.io

1. 從 Toolkit for VS Code 中，展開 DEVELOPER TOOLS 瀏覽器。
2. 展開 CodeCatalyst，然後在您目前的開發環境中，選擇開啟開發檔案以在`devfile.yaml`新的編輯器視窗中開啟。

3. 從 VS 程式碼編輯器中，更新您的 devfile，然後儲存您的變更。
4. 下次啟動開發環境時，系統會更新組態，以符合 中定義的規格`Devfile`。

AWS 從開發環境驗證和連線至

若要從開發環境存取您的所有 AWS 資源，您必須驗證 Toolkit 的遠端執行個體並將其與 AWS 您的帳戶連線。啟動開發環境時， Toolkit 的遠端執行個體會自動使用從 Toolkit 本機執行個體繼承的憑證進行身分驗證。

更新 Toolkit 遠端執行個體登入資料的程序與 Toolkit 本機執行個體中的身分驗證體驗相同。如需如何 AWS 從 Toolkit 更新登入資料、驗證和連線至 的詳細指示，請參閱本使用者指南入門主題中的連線至 AWS 一節。

如需與 相容之每個 AWS 身分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身分驗證和存取主題。

在開發環境中使用 Toolkit for VS 程式碼

在 VS Code 中開啟或建立開發環境後，您可以從 Toolkit for VS Code 工作，類似於從 VS Code 的本機執行個體工作。執行 VS Code 的開發環境已設定為自動安裝 Toolkit， AWS 並與您的 AWS 建置器 ID 連線。

停止開發環境

若要停止目前的開發環境：

1. 從 Toolkit for VS Code 中，展開 DEVELOPER TOOLS 瀏覽器。
2. 展開 CodeCatalyst，然後選擇停止開發環境。
3. 當 VS 程式碼提示時，請確認您想要停止開發環境。
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，請完成下列步驟：

> **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 DEVELOPER TOOLS 瀏覽器。
2. 展開 CodeCatalyst 並選擇開啟設定，以開啟目前開發環境的開發環境設定檢視。
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、指派給它的專案，並停止您的環境。
   - Devfile：檢視`Devfile`開發環境的 名稱和位置。選擇在編輯器中開啟按鈕`Devfile`來開啟您的 。
   - 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。

對 Amazon CodeCatalyst 和 VS Code 進行故障診斷

下列主題解決使用 Amazon CodeCatalyst 和 VS Code 時的潛在技術問題。

主題

- VS 程式碼版本
- Amazon CodeCatalyst 的許可
- 從 Toolkit for VS 程式碼連線至開發環境

VS 程式碼版本

您的 VS 程式碼版本預期會在您的系統上設定 `vscode://` URIs的處理常式。如果沒有此處理常式，您將無法從 AWS Toolkit 存取所有 CodeCatalyst 功能。例如，從 VS Code Insiders 啟動開發環境時發生錯誤。這是因為 VS Code Insiders 會處理 `vscode-insiders://` URIs 且不處理 `vscode://` URIs。

Amazon CodeCatalyst 的許可

以下是從 使用 CodeCatalyst 的檔案許可要求 AWS Toolkit for Visual Studio Code：

- 將`~/.ssh/config`檔案的自有存取許可設定為 `read`和 `write`。限制所有其他使用者的`write`許可。
- 將 `~/.ssh/id_dsa` 和 `~/.ssh/id_rsa` 檔案的存取許可設定為 `read` 。限制所有其他使用者的`readwrite`和 `execute`許可。
- 您的`globals.context.globalStorageUri.fsPath`檔案必須位於可寫入的位置。

從 Toolkit for VS 程式碼連線至開發環境

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

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

- 選擇開啟組態。 . . 按鈕，可在 VS 程式碼編輯器中開啟您的`~/.ssh/config`檔案。
- 從編輯器中，選取並刪除`Host aws-devenv-*`區段的內容。
- 儲存您對 `Host aws-devenv-*`的 所做的變更`~/.ssh/config`。然後，關閉 檔案。
- 重新嘗試從 Toolkit for VS 程式碼連線至開發環境。

使用 Amazon API Gateway

您可以使用 在連線帳戶中瀏覽和執行遠端 API Gateway 資源 AWS AWS Toolkit for Visual Studio Code。

> **Note**
>
> 此功能不支援偵錯。

瀏覽和執行遠端 API Gateway 資源

1. 在 AWS Explorer 中，選擇 API Gateway 以展開選單。會列出遠端 API Gateway 資源。
2. 找到您要叫用的 API Gateway 資源，開啟其內容 （ 按一下滑鼠右鍵 ） 選單，然後選擇叫用 AWS。
3. 在參數表單中，指定叫用參數。
4. 若要執行遠端 API Gateway 資源，請選擇叫用。結果會在 VS 程式碼輸出檢視中顯示。

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，從原始碼或容器映像直接部署到 AWS 雲端中可擴展且安全的 Web 應用程式。使用它，您不需要學習新技術、決定要使用的運算服務，或知道如何佈建和設定 AWS 資源。

您可以使用 根據來源映像或來源碼 AWS App Runner 來建立和管理服務。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)：將私有映像儲存在您的帳戶中 AWS 。
- 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 支援 GitHub 作為原始碼儲存庫提供者。

先決條件

若要使用 與 App Runner 互動， AWS Toolkit for Visual Studio Code 需要下列項目：

- 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來源的特定資訊，例如：容器映像 URI 或與 GitHub 儲存庫的連線。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 所需許可的最簡單方法是將現有的 AWS 受管政策連接到相關 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。

您可以在使用 VS Code 的命令面板設定服務執行個體時自動建立此角色。

> **Note**
>
> AWSServiceRoleForAppRunner 服務連結角色允許 AWS App Runner 完成下列任務：

- 將日誌推送至 Amazon CloudWatch Logs 日誌群組。
- 建立 Amazon CloudWatch Events 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 管理主控台 中或使用 呼叫的 API 操作建立 時 AWS Toolkit for Visual Studio Code， 會為您 AWS App Runner 建立此服務連結角色。

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 從來源映像或來源碼部署服務。

Source image

如果您是從來源映像部署，您可以從私有或公有映像登錄檔取得該 AWS 映像儲存庫的連結。

- 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 Toolkit for VS Code 中的 AWS Explorer 取得私有 Amazon ECR 儲存庫的 URI：
>
> - 開啟 AWS Explorer 並展開 ECR 節點，以檢視該 AWS 區域的儲存庫清單。
> - 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 的命令面板設定服務執行個體時，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

若要將原始碼部署到 AWS App Runner 服務，該程式碼必須存放在由支援的儲存庫提供者維護的 Git 儲存庫中。App Runner 支援 GitHub 作為原始碼儲存庫提供者。

如需設定 GitHub 儲存庫的相關資訊，請參閱 GitHub 上的入門文件。

若要從 GitHub 儲存庫將原始碼部署至 App Runner 服務，App Runner 會建立至 GitHub 的連線。如果您的儲存庫是私有的 (也就是說，它不能在 GitHub 上公開存取)，則您必須提供連線的詳細資訊給 App Runner。

> **Important**
>
> 若要建立 GitHub 連線，您必須使用 App Runner 主控台 (https://console.aws.amazon.com/apprunner) 來建立一個將 GitHub 連結到 AWS的連線。使用 VS Code 的命令面板設定服務執行個體時，您可以選擇 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 服務執行個體提供受管執行期，可讓您的程式碼建置和執行。 AWS App Runner 目前支援下列執行期：

- 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 AWS Explorer 和 VS Code 的命令面板，在 Toolkit for VS Code 中建立 App Runner 服務。在您選擇在特定 AWS 區域中建立服務之後， Command Palette 提供的編號步驟會引導您完成設定應用程式執行之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. Open AWS Explorer，如果尚未開啟。
2. 在 App Runner 節點上按一下滑鼠右鍵並選擇 Create Service (建立服務)。

   命令調色盤隨即顯示。
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 支援 GitHub 作為原始碼儲存庫提供者。

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。
> 根據預設，自動部署會關閉，當 Command Palette 標頭上的 圖示具有穿過它的對角線時，就會顯示此選項。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 Command Palette 步驟回報找不到標籤，您需要返回步驟以選取包含已標記容器映像的儲存庫。
3. 如果您使用的是 Amazon ECR 私有登錄，則需要 ECR 存取角色 AppRunnerECRAccessRole，允許 App Runner 在您帳戶中存取 Amazon Elastic Container Registry (Amazon ECR) 映像。選擇 Command Palette 標頭上的「+」圖示，以自動建立此角色。(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 為單位的記憶體的組合。

   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. 針對選取連線，選擇連結 GitHub 的連線 AWS。可供選取的連線清單列於 App Runner 主控台上的 GitHub connections (GitHub 連線) 頁面。
2. 對於 Select a remote GitHub repository (選取遠端 GitHub 儲存庫)，請選擇或輸入遠端儲存庫的 URL。

   已使用 Visual Studio Code 的來源控制管理 (SCM) 設定的遠端儲存庫可供選取。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。

   如果您選擇在此處設定所有設定，請使用命令面板來指定下列項目：

   - 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 為單位的記憶體的組合。

   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 Explorer 窗格來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

如果您需要暫時停用 Web 應用程式並停止程式碼執行，您可以暫停 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. Open AWS Explorer，如果尚未開啟。
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. Open AWS Explorer，如果尚未開啟。
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. Open AWS Explorer，如果尚未開啟。
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 Logs 來監控、存放和存取您 App Runner 等服務的日誌串流。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. 從命令調色盤中，從群組中選擇日誌串流。

   VS 程式碼編輯器會顯示組成串流的日誌事件清單。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. Open AWS Explorer，如果尚未開啟。
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. 在命令面板中，輸入 delete，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 應用程式建置器

AWS 的 Application Builder AWS Toolkit for Visual Studio Code 是視覺化建置專案、在本機迭代專案，以及將應用程式部署到其中的指南 AWS。

下列主題說明如何從 使用 AWS Application Builder AWS Toolkit for Visual Studio Code。

主題

- 使用 AWS 應用程式建置器

使用 AWS 應用程式建置器

下列各節說明如何在 中存取 AWS Application Builder AWS Toolkit for Visual Studio Code。使用 Application Builder，您可以在本機以視覺化方式建置專案、迭代專案，並將其部署到其中 AWS。如需 Application Builder 的功能和潛在使用案例以及您本機 AWS Lambda 體驗的概觀，請參閱 AWS 開發人員 YouTube 影片 *New* AWS Lambda Local IDE Experience！。

使用 AWS Application Builder Explorer

若要存取 AWS Toolkit 中的 Application Builder，請在 VS Code 中開啟 AWS Toolkit，然後展開 AWS Application Builder Explorer。 AWS Application Builder Explorer 包含在 VS 程式碼編輯器標籤中開啟 Application Builder 逐步解說的連結，並顯示目前 VS 程式碼工作區中包含 AWS Application Builder 相關資源的資料夾。

從 AWS Toolkit 中的 Application Builder Explorer 中，有 4 個project-folder-level動作可從專案資料夾旁的按鈕圖示存取，或開啟專案資料夾的內容選單（按一下滑鼠右鍵）：

- 開啟範本檔案：在 VS Code Explorer 中開啟範本檔案。
- 使用 Infrastructure Composer 開啟：在 VS 程式碼編輯器中使用 AWS Infrastructure Composer 開啟範本檔案。如需使用 AWS Infrastructure Composer 的詳細資訊，請參閱 AWS Infrastructure Composer 開發人員指南中的什麼是 AWS Infrastructure Composer 主題。
- 組建 SAM 範本：開啟 AWS Toolkit 中的指定組建參數對話方塊。您可以選擇指定建置的建置旗標，或從 samconfig 使用預設值。如需 AWS SAM 範本的詳細資訊，請參閱《 AWS Serverless Application Model開發人員指南》中的範本結構主題。
- 部署 SAM 應用程式：在 VS 程式碼中開啟選取部署命令對話方塊，您可以在其中選擇部署應用程式或同步，以更新已部署的應用程式。如需部署 AWS SAM 應用程式的詳細資訊，請參閱《 AWS Serverless Application Model開發人員指南》中的部署您的應用程式和資源主題。

您可以從專案資料夾中 AWS Lambda 函數旁的按鈕圖示，或在 AWS Lambda 函數上按一下滑鼠右鍵，存取 2 個動作：

- 本機調用和偵錯組態：在您的 VS 程式碼編輯器中開啟本機調用和偵錯組態表單。使用此表單，您可以建立、編輯和執行 類型的 launch-configs：`aws-sam`。如需有關 SAM Debug 組態的其他資訊，請參閱本使用者指南中的用於偵錯無伺服器應用程式的組態選項主題。

  > **Note**
  >
  > VS Code 目前不支援在 ARM64 架構上偵錯 .NET Core 應用程式。如果您嘗試偵錯 .NET Core 應用程式，則會顯示下列錯誤：
  > `The vsdbg debugger does not currently support the arm64 architecture. Function will run locally without debug.`
  > 如需此問題的其他詳細資訊，請參閱 DotNet GitHub 儲存庫中的此 VSCode-csharp 問題。

- 開啟函數處理常式：開啟包含函數處理常式的專案檔案。

部署的 AWS Lambda 函數有 2 個其他動作可用。

- 遠端調用：在 VS 程式碼編輯器中開啟遠端調用組態功能表。
- 搜尋日誌：在 VS 程式碼中開啟搜尋日誌對話方塊。

應用程式建置器的逐步解說

應用程式建置器的逐步解說是step-by-step互動式指南，引導您完成使用應用程式建置器建置新 AWS 應用程式的程序。您可以從兩個位置存取應用程式建置器的逐步解說： 中的應用程式建置器瀏覽器 AWS Toolkit for Visual Studio Code 和 VS 程式碼歡迎索引標籤。當您從 AWS 工具組中的應用程式建置器瀏覽器選取應用程式建置器的逐步解說時，它會在 VS 程式碼編輯器視窗中的 VS 程式碼歡迎索引標籤中開啟應用程式建置器的逐步解說。

應用程式建置器的逐步解說包含 5 個主要部分：

1. 安裝

安裝區段會檢查您是否已安裝 Application Builder 和其他選用 AWS CLI 工具所需的工具。如果您沒有所需的工具或您的工具已過時，系統會引導您完成安裝正確版本的程序。

若要查看是否已安裝正確 AWS CLI 和選用的工具，請選取您要測試之 AWS CLI 或其他工具的 按鈕。選取按鈕後，您的 AWS Toolkit Logs 會更新，而 VS 程式碼會顯示警示訊息，其中包含工具的狀態。如果您需要安裝或更新工具，應用程式建置器的逐步解說會更新您需要繼續執行的指示和資源。

如需安裝 的詳細資訊 AWS CLI，請參閱《 AWS CLI開發人員指南》中的安裝或更新至最新版本的主題 AWS CLI。如需安裝 AWS SAM CLI 的詳細資訊，請參閱《CLI AWS SAM 開發人員指南》中的安裝 AWS SAM CLI 主題。

2. 選擇您的應用程式範本

選擇您的應用程式範本區段會引導您完成從範本建置新應用程式的程序。

若要選擇範本並初始化您的應用程式，請完成下列步驟。

1. 從應用程式建置器的逐步解說中，選取選擇您的應用程式範本區段，以在畫面上顯示範本選項清單。
2. 從清單中選擇範本，然後選擇初始化專案按鈕以開啟 VS 程式碼對話方塊。
3. 完成 VS 程式碼對話方塊中的步驟，以初始化您的新應用程式。
4. Toolkit AWS 會記錄初始化程序期間應用程式狀態的更新。
5. 若要在 Application Builder Explorer 中檢視您的應用程式，請選擇重新整理 Application Builder Explorer 圖示，以使用您的變更來更新瀏覽器。

3. 在本機反覆運算

本機迭代區段包含範例映像，示範如何使用 VS Code 和 AWS Toolkit Explorer 中提供的應用程式建置器功能進行迭代。

如需 VS 程式碼和 AWS 工具組瀏覽器中所有可用應用程式建置器功能的詳細資訊，請參閱本使用者指南主題中的使用應用程式建置器瀏覽器一節。

4. 部署至 AWS

部署至 AWS 區段包含如何設定登入資料以與 連線的相關資訊 AWS ，以便部署您的應用程式，以及如何使用 Application Builder 部署應用程式的範例。

若要透過應用程式建置器演練 AWS 中的現有登入資料連線至 ，請完成下列其中一個程序。

人力： AWS 使用單一登入來登入 。

1. 從應用程式建置器逐步解說中的部署至 AWS區段中，選擇設定登入資料按鈕，以在 AWS Toolkit Explorer 中開啟 AWS： LOGIN 選單。

2. 從 AWS：LOGIN 功能表中，選擇人力，然後選擇繼續按鈕以繼續。
3. 在提供的欄位中輸入您的開始 URL，從下拉式選單中選擇您的AWS 區域，然後選擇繼續按鈕以繼續。
4. 從 VS 程式碼快顯視窗中，確認您想要在預設瀏覽器中開啟 AWS 身分驗證網站。
5. 在預設瀏覽器中，完成身分驗證步驟後，您會在身分驗證完成時收到通知，而且可以安全地關閉瀏覽器視窗。

IAM 登入資料：存放用於 AWS CLI 工具的金鑰。

1. 從應用程式建置器逐步解說中的部署至 AWS區段中，選擇設定登入資料按鈕，以在 AWS Toolkit Explorer 中開啟 AWS： LOGIN 選單。
2. 從 AWS：LOGIN 功能表中，選擇 IAM 登入資料，然後選擇繼續按鈕以繼續。
3. 在提供的欄位中輸入設定檔名稱，然後輸入您的 **`Access Key`**和 **`Secret Key`**，然後選擇繼續按鈕以繼續。
4. VS 程式碼會顯示身分驗證的狀態，並在身分驗證完成或登入資料無效時通知您。

如需使用 設定登入資料以進行部署的詳細資訊 AWS CLI，請參閱《 AWS CLI開發人員指南》中的設定 AWS CLI主題。如需有關使用現有登入資料 AWS 從 AWS Toolkit 連線至 的其他資訊，請參閱《 使用者指南》中的連線至 AWS主題。

AWS 基礎設施編寫器

您可以使用 AWS Toolkit for Visual Studio Code 來使用 AWS Infrastructure Composer。 AWS Infrastructure Composer 是 AWS 應用程式的視覺化建置器，可協助設計應用程式架構和視覺化您的 CloudFormation 基礎設施。

如需 AWS Infrastructure Composer 的詳細資訊，請參閱 AWS Infrastructure Composer 使用者指南。

下列主題說明如何從 使用 AWS Infrastructure Composer AWS Toolkit for Visual Studio Code。

主題

- 在 Toolkit 中使用 AWS Infrastructure Composer

在 Toolkit 中使用 AWS Infrastructure Composer

AWS 的 Infrastructure Composer 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 Infrastructure Composer 來視覺化和修改 CloudFormation 和 AWS Serverless Application Model (AWS SAM) 範本。使用 Infrastructure Composer 時，您的變更會持續存放，讓您可以直接在 VS 程式碼編輯器中或使用互動式畫布在編輯檔案之間無縫切換。

如需 AWS Infrastructure Composer、入門資訊和教學課程的詳細資訊，請參閱 AWS Infrastructure Composer 使用者指南。

下列各節說明如何從 存取 AWS Infrastructure Composer AWS Toolkit for Visual Studio Code。

從工具組存取 AWS 基礎設施編寫器

您可以從 Toolkit 存取 AWS Infrastructure Composer 有 3 種主要方式。

從現有範本存取 AWS Infrastructure Composer

1. 從 VS 程式碼中，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，按一下編輯器視窗右上角的 AWS 基礎設施編寫器按鈕。
3. AWS Infrastructure Composer 會在 VS Code 編輯器視窗中開啟並視覺化您的範本檔案。

從內容選單存取 AWS 基礎設施編寫器 （ 按一下滑鼠右鍵 ）

1. 在 VS 程式碼中，用滑鼠右鍵按一下您要使用 AWS Infrastructure Composer 開啟的範本檔案。
2. 在內容功能表中，選擇使用應用程式編寫器開啟選項。
3. AWS Infrastructure Composer 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。

從命令面板存取 AWS 基礎設施編寫器

1. 從 VS 程式碼按 **`Cmd + Shift + P`**或 **`Ctrl + Shift + P`**(Windows) 開啟命令面板
2. 在搜尋欄位中，輸入 **`AWS Infrastructure Composer`**，然後在填入結果時選擇 AWS Infrastructure Composer。
3. 選擇您要開啟的範本檔案， AWS Infrastructure Composer 會開啟並在新的 VS 程式碼編輯器視窗中視覺化您的範本檔案。

AWS CDK 適用於 VS 程式碼的

這是預覽版本中功能的發行前文件。內容可能變動。

AWS CDK Explorer 可讓您操作 AWS Cloud Development Kit (AWS CDK)應用程式或應用程式。您可以在 AWS Cloud Development Kit (AWS CDK) 開發人員指南 AWS CDK 中找到 的詳細資訊。

AWS CDK 應用程式由稱為建構的建置區塊組成，其中包括 CloudFormation 堆疊的定義及其內的 AWS 資源。使用 AWS CDK Explorer，您可以視覺化 AWS CDK 建構中定義的堆疊和資源。此視覺效果是在 Visual Studio Code (VS Code) 編輯器開發人員工具窗格中的樹狀檢視中提供。

本節提供如何在 VS 程式碼編輯器AWS CDK 中存取和使用 的相關資訊。其假設您已安裝並設定本機 IDE 的 Toolkit for VS 程式碼。

主題

- 使用 AWS CDK 應用程式

使用 AWS CDK 應用程式

這是預覽版本中功能的發行前文件。內容可能變動。

使用 AWS Toolkit for VS Code 中的 AWS CDK Explorer 來視覺化和使用 AWS CDK 應用程式。

先決條件

- 請確定您的系統符合安裝 Toolkit for VS 程式碼中指定的先決條件。
- 安裝 AWS CDK 命令列界面，如《 AWS Cloud Development Kit (AWS CDK) 開發人員指南》中的 AWS CDK 入門的前幾個章節所述。

> **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 AWS Toolkit for VS Code AWS CDK Explorer，您可以管理存放在應用程式 CDK 建構中的堆疊和資源。 AWS CDK Explorer 會使用 `tree.json` 檔案中定義的資訊，在樹狀檢視中顯示您的資源，這些資訊會在您執行 **`cdk synth`**命令時建立。根據預設， `tree.json` 檔案位於應用程式的 `cdk.out` 目錄中。

若要開始使用 Toolkit AWS CDK Explorer，您需要建立 CDK 應用程式。

1. 完成《 AWS CDK 開發人員指南》中 Hello World 教學課程的前幾個步驟。

   > **Important**
   >
   > 當您到達部署堆疊的教學步驟時，請停止並返回本指南。

   > **Note**
   >
   > 您可以在 VS Code 編輯器內的作業系統命令列或終端機視窗中執行教學課程中提供的命令，例如 **`cdk initmkdir`**和 。

2. 完成 CDK 教學課程的必要步驟後，開啟您在 VS 程式碼編輯器中建立的 CDK 內容。
3. 從 AWS 導覽窗格中，展開 CDK （預覽） 標題。您的 CDK 應用程式及其相關資源現在會顯示在 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 程式碼編輯器時，您可以一次載入多個資料夾。每個資料夾可以包含多個 CDK 應用程式，如上圖所示。 AWS CDK Explorer 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。如果您變更 CDK 應用程式的各個層面，例如新增更多資源，則需要再次執行該命令，才能查看樹狀檢視中反映的變更。

在 AWS CDK 應用程式上執行其他操作

您可以使用 VS 程式碼編輯器在 CDK 應用程式上執行其他操作，就像使用作業系統的命令列或其他工具一樣。例如，您可以在編輯器中更新程式碼檔案，並使用 VS Code Terminal 視窗部署應用程式。

若要嘗試這些類型的動作，請使用 VS 程式碼編輯器繼續《 AWS CDK 開發人員指南》中的 Hello World 教學課程。請務必執行最後一個步驟，銷毀應用程式的資源，以免 AWS 您的帳戶產生非預期的成本。

使用 AWS CloudFormation 堆疊

AWS Toolkit for Visual Studio Code 支援 AWS CloudFormation堆疊。使用 Toolkit for VS 程式碼，您可以使用 AWS CloudFormation 堆疊執行特定任務，例如刪除它們。

主題

- 刪除 CloudFormation 堆疊
- 使用 建立 AWS CloudFormation 範本 AWS Toolkit for Visual Studio Code

刪除 CloudFormation 堆疊

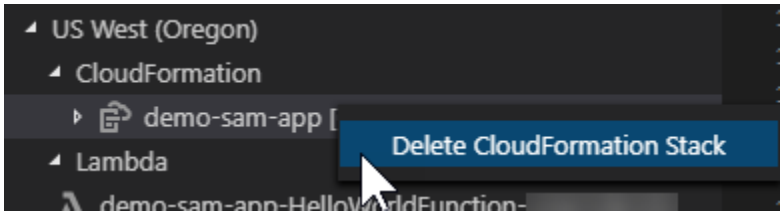
您可以使用 AWS Toolkit for Visual Studio Code 來刪除 CloudFormation 堆疊。

先決條件

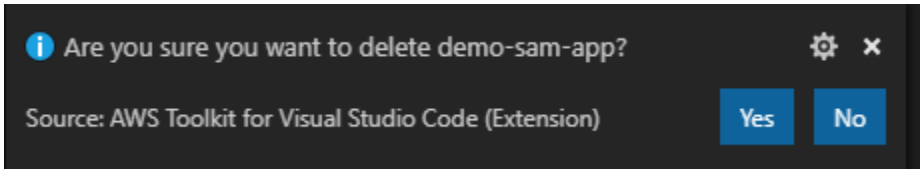
- 請確定您的系統符合安裝 Toolkit for VS 程式碼中指定的先決條件。
- 請確定您在 中設定的登入資料身分驗證與存取包含對 CloudFormation 服務的適當讀取/寫入存取權。如果在 AWS Explorer 的 CloudFormation 下方，您看到類似「Error loading CloudFormation resources」(載入 CloudFormation 資源時發生錯誤) 的訊息，請檢查與這些憑證連接的許可。您對許可所做的變更將需要幾分鐘的時間，才能影響 VS 程式碼中的 AWS Explorer。

刪除 CloudFormation 堆疊

1. 在 AWS Explorer 中，開啟您要刪除之 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 可協助您撰寫 AWS CloudFormation 和 SAM 範本。

先決條件

適用於 VS 程式碼和登入資料先決條件的 Toolkit

- 您必須先符合使用者指南安裝 Toolkit for VS Code 中概述的需求，才能從 Toolkit for VS Code 存取 CloudFormation 服務。
- 您在 中建立的登入資料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 資源訊息時發生錯誤，請檢查您已連接到這些登入資料的許可。另請注意，對許可所做的變更可能需要幾分鐘的時間才能在 AWS Explorer 中更新。

CloudFormation 範本先決條件

- 安裝並啟用 Redhat 開發人員 YAML VS 程式碼延伸。

- 使用 Redhat 開發人員 YAML VS 程式碼延伸時，您需要連線到網際網路，因為它用於在您的機器上下載和快取 JSON 結構描述。

使用 YAML 結構描述支援撰寫 CloudFormation 範本

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讓程序更快，也有助於改善範本的品質。選取範本的結構描述時，建議使用下列最佳實務。

CloudFormation 範本

- 檔案具有 .yaml 或 .yml 副檔名。
- 檔案具有最上層`AWSTemplateFormatVersion`或資源節點。

SAM 範本

- 已針對 CloudFormation 說明的所有條件
- 檔案具有最上層的轉換節點，其中包含開頭為 的值`AWS::Serverless`。

結構描述將在檔案修改時套用。例如，SAM 範本結構描述會在將無伺服器轉換新增至 CloudFormation 範本並儲存檔案後套用。

語法驗證

YAML 延伸模組會自動將類型驗證套用至您的範本。這會反白顯示指定屬性的無效類型項目。如果您將滑鼠暫留在反白顯示的項目上，延伸項目會顯示修正動作。

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 延伸模組的自動完成功能。

使用 使用 CloudWatch Logs AWS Toolkit for Visual Studio Code

Amazon CloudWatch Logs 可將您使用的所有系統、應用程式和 AWS 服務的日誌，集中在可高度擴展的單一服務中。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。如需更多詳細資訊，請參閱 Amazon CloudWatch 使用者指南中的什麼是 Amazon CloudWatch Logs？。

下列主題說明如何使用 AWS Toolkit for Visual Studio Code 來使用 AWS 帳戶中的 CloudWatch Logs。

主題

- 使用 檢視 CloudWatch 日誌群組和日誌串流 AWS Toolkit for Visual Studio Code
- 使用 在日誌串流中使用 CloudWatch 日誌事件 AWS Toolkit for Visual Studio Code
- 搜尋 CloudWatch 日誌群組
- Amazon CloudWatch Logs Live Tail

使用 檢視 CloudWatch 日誌群組和日誌串流 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。

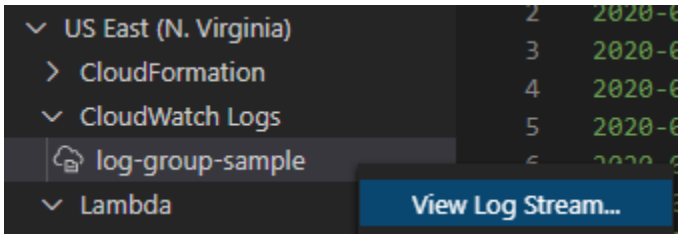
主題

- 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 程式碼中，選擇檢視、 Explorer 以開啟 AWS Explorer。
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。
3. 若要檢視日誌群組中的日誌串流，請在日誌群組名稱上按一下滑鼠右鍵，然後選擇檢視日誌串流。



4. 從命令調色盤中，從要檢視的群組中選取日誌串流。

> **Note**
>
> 命令面板會顯示每個串流中最後一個事件的時間戳記。

Log Stream 編輯器會啟動以顯示串流的日誌事件。

使用 在日誌串流中使用 CloudWatch 日誌事件 AWS Toolkit for Visual Studio Code

開啟 Log Steam 編輯器之後，您可以存取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

當您開啟日誌串流時，日誌串流編輯器會顯示該串流的日誌事件序列。

1. 若要尋找要檢視的日誌串流，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)。

   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：

   - 依時間檢視事件：選擇 Load newer events (載入較新的事件) 或 Load older events (載入較舊的事件)，可顯示最新與較舊的日誌事件。

     > **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，從選單中選取 Copy (複製)。
- 複製日誌串流的名稱：在日誌串流編輯器的索引標籤上按一下滑鼠右鍵，然後選擇複製日誌串流名稱。

> **Note**
>
> 您也可以使用命令面板來執行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

您可以將 CloudWatch 日誌串流編輯器的內容下載到本機中的 `log` 檔案。

> **Note**
>
> 使用此選項，您可以儲存為僅存檔目前顯示在日誌串流編輯器中的日誌事件。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. 若要尋找要複製的日誌串流，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. 選擇顯示日誌串流名稱的索引標籤旁的儲存圖示。

> **Note**
>
> 您也可以使用 命令面板來執行 AWS Toolkit Save Current Log Stream Content。

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 日誌群組

您可以使用搜尋日誌群組來搜尋日誌群組中的所有日誌串流。

如需 Amazon CloudWatch Logs 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 Code Command Palette 搜尋日誌群組

若要從 VS Code Command Palette 搜尋日誌群組，請完成下列步驟。

如需 Amazon CloudWatch Logs 篩選條件和模式的詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的篩選條件和模式語法一節。

1. 從 VS 程式碼中，按 **cmd+shift+p**（視窗：）開啟命令調色盤**ctrl+shift+p**。
2. 從命令調色盤中，輸入命令 **AWS: Search Log Group**，然後選取該命令以開啟 VS 程式碼中的搜尋日誌群組對話方塊，然後依照提示繼續。

   > **Note**
   >
   > 從第一個提示中，您可以選擇先切換您的 AWS 區域，再繼續進行後續步驟。

3. 從選取日誌群組 (1/3) 提示中，選擇您要搜尋的日誌群組。
4. 從選取時間篩選條件 (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，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。
6. 當搜尋完成時，您的搜尋結果會在 VS 程式碼編輯器中開啟。

從 AWS Explorer 搜尋日誌群組

若要從 AWS Toolkit for Visual Studio Code Explorer 搜尋日誌群組，請完成下列步驟。

1. 從 AWS Toolkit for Visual Studio Code Explorer 展開 CloudWatch。
2. 開啟您要搜尋之搜尋日誌群組的內容選單（按一下滑鼠右鍵），然後選擇搜尋日誌群組以開啟搜尋提示。
3. 選取時間範圍以繼續，依照提示操作。
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. 當搜尋完成時，您的搜尋結果會在 VS 程式碼編輯器中開啟。

使用搜尋日誌結果

成功完成 CloudWatch 日誌群組搜尋後，您的搜尋結果會在 VS 程式碼編輯器中開啟。下列程序說明如何使用搜尋日誌結果。

> **Note**
>
> 檢視單一日誌串流時，下列功能僅限於目前作用中日誌串流中的結果。

儲存您的搜尋日誌群組結果

若要在本機儲存您的搜尋日誌群組結果，請完成下列步驟。

1. 從搜尋日誌群組結果中，選擇 VS 程式碼編輯器右上角的儲存日誌至檔案圖示按鈕。
2. 從另存新檔提示中，指定您要儲存檔案的名稱和位置。
3. 選擇確定會將檔案儲存至本機電腦。

變更時間範圍

若要變更搜尋日誌群組結果中作用中的時間範圍，請完成下列步驟。

1. 從搜尋日誌群組結果中，選擇 VS 程式碼編輯器右上角的依日期搜尋...圖示按鈕。
2. 從選取時間篩選條件提示中，為您的搜尋日誌結果選擇新的時間範圍。
3. 您的結果會在選取時間篩選條件提示關閉時更新。

變更搜尋模式

若要變更搜尋日誌群組結果中作用中的搜尋模式，請完成下列步驟。

1. 從搜尋日誌群組結果中，選擇 VS Code 編輯器右上角的依模式搜尋...圖示按鈕。
2. 從搜尋日誌群組提示中，在提供的欄位中輸入新的搜尋模式。
3. 按下 **`Enter`**鍵以關閉提示，並使用新的搜尋模式更新您的結果。

Amazon CloudWatch Logs Live Tail

使用 Amazon CloudWatch Logs Live Tail 將 CloudWatch 日誌事件擷取至特定日誌群組時，即時串流這些事件。 Amazon CloudWatch

如需 Live Tail 功能的詳細資訊，請參閱《Amazon CloudWatch Logs 使用者指南》中的使用 CloudWatch Logs Live Tail 進行故障診斷主題。 Amazon CloudWatch

Live Tail 工作階段會按工作階段使用時間 (每分鐘) 產生費用。如需定價的相關資訊，請參閱《Amazon CloudWatch 定價指南》的付費方案一節中的日誌索引標籤。

從 VS Code Command Palette 啟動 Live Tail 工作階段

若要從 VS Code Command Palette 啟動 Live Tail 工作階段，請完成下列步驟。

如需 Amazon CloudWatch Logs 篩選條件和模式的詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的篩選條件和模式語法一節。

從命令面板啟動結尾工作階段

1. 從 VS 程式碼中，按 **cmd+shift+p**(Windows：) 開啟命令面板**ctrl+shift+p**。
2. 從命令面板中，輸入命令 `AWS: Tail Log Group`，然後選取它以開啟 VS 程式碼中的尾端日誌群組對話方塊，然後依照提示繼續。

> **Note**
>
> 在第一個提示中，您可以選擇先切換您的 AWS 區域，再繼續進行後續步驟。

3. 從 Tail Log Group (1/3) 提示中，選擇您要結尾的日誌群組。
4. 從包含來自... (2/3) 的日誌事件提示中，選擇要套用至結尾工作階段的日誌串流篩選條件。
5. 從提供日誌事件篩選條件模式... (3/3) 提示中，在提供的欄位中輸入篩選條件模式，然後按 **Enter** 鍵繼續，或按 **ESC**鍵取消搜尋。
6. 完成後，您的結果會串流到 VS 程式碼編輯器

> **Note**
>
> 如果在 VS 程式碼視窗中執行的 Live Tail 工作階段符合新提交的 Tail Log Group 命令的組態，則不會啟動新的工作階段。反之，您現有的工作階段會成為作用中的文字編輯器。

從 AWS Explorer 啟動 Live Tail 工作階段

若要從 AWS Toolkit Explorer 啟動 Live Tail 工作階段，請完成下列步驟。

從 AWS Explorer 啟動結尾工作階段

1. 從 AWS Toolkit Explorer 展開 CloudWatch。

2. 開啟您要結尾之日誌群組的內容選單（按一下滑鼠右鍵），然後選擇結尾日誌群組以開啟結尾提示。
3. 按照提示繼續。
4. 您的結果將串流至 VS 程式碼編輯器。

停止 Live Tail 工作階段

有兩種方式可以停止執行中的尾隨工作階段。

停止結尾工作階段

1. 按一下結尾工作階段文字文件底部的停止結尾 CodeLens。
2. 關閉包含結尾工作階段文字文件的所有編輯器。

Amazon DocumentDB

您可以使用 直接在 VS 程式碼中管理您的 Amazon DocumentDB 叢集和執行個體 AWS Toolkit for Visual Studio Code。Amazon DocumentDB（與 MongoDB 相容）是一種快速、可靠且全受管的資料庫服務，可簡化雲端中 MongoDB 相容資料庫的設定、操作和擴展。如需 Amazon DocumentDB 服務的詳細資訊，請參閱 Amazon DocumentDB 開發人員指南。

下列主題說明如何搭配 使用 Amazon DocumentDB AWS Toolkit for Visual Studio Code。

主題

- 在 Toolkit 中使用 Amazon DocumentDB

在 Toolkit 中使用 Amazon DocumentDB

Amazon DocumentDB（與 MongoDB 相容）是一種快速、可靠且全受管的資料庫服務，可簡化雲端中 MongoDB 相容資料庫的設定、操作和擴展。

如需 Amazon DocumentDB、入門資訊和教學課程的詳細資訊，請參閱 Amazon DocumentDB 開發人員指南。

下列各節說明如何搭配 使用 Amazon DocumentDB AWS Toolkit for Visual Studio Code。

從 AWS Toolkit 存取 Amazon DocumentDB

若要使用 AWS Toolkit 存取 Amazon DocumentDB，請完成下列程序。

在 AWS 工具組中存取 Amazon DocumentDB

1. 從 VS 程式碼開啟 AWS Toolkit for Visual Studio Code。
2. 從 AWS Toolkit 中，展開 Explorer。
3. 從 Explorer 展開 Amazon DocumentDB，以顯示您現有的 Amazon DocumentDB 資源。

建立執行個體型叢集。

若要開始使用 Amazon DocumentDB，請完成下列程序來建立叢集。

建立執行個體型叢集

1. 從 AWS Toolkit for Visual Studio Code開啟（按一下滑鼠右鍵）Amazon DocumentDB 的內容選單，然後選取建立叢集以在 VS 程式碼中開啟建立 Amazon DocumentDB 叢集對話方塊。
2. 從叢集類型畫面中，選擇執行個體型叢集。
3. 從叢集名稱畫面中，指定新叢集的名稱。
4. 從選取引擎版本畫面中，選擇您偏好的 Amazon DocumentDB 引擎版本。
5. 在管理員使用者名稱和密碼畫面中，指定管理員使用者名稱和密碼來保護您的叢集。
6. 從指定儲存體加密畫面中，選擇是否要加密叢集。
7. 在執行個體數量畫面中，設定您偏好的執行個體數量。
8. 從選取執行個體類別畫面中，選擇您偏好的執行個體類別，然後繼續建立新的叢集。

> **Note**
>
> 啟動叢集可能需要幾分鐘的時間。

複製叢集端點

若要複製 Amazon DocumentDB 叢集端點，請完成下列程序。

複製叢集端點

1. 從 AWS Toolkit for Visual Studio Code展開 Amazon DocumentDB 以顯示您的 Amazon DocumentDB 叢集。
2. 在您要從中複製連線詳細資訊的叢集上按一下滑鼠右鍵，然後選擇複製端點，將叢集端點資訊複製到剪貼簿。
3. 您的叢集端點現在可以貼到您的文件中。

在瀏覽器中開啟

在 AWS 主控台中開啟 Amazon DocumentDB 叢集，以取得更多叢集管理功能。若要在預設 Web 瀏覽器中開啟 Amazon DocumentDB 叢集的 AWS 主控台，請完成下列程序。

在 AWS 主控台中開啟叢集

1. 從 AWS Toolkit for Visual Studio Code展開 Amazon DocumentDB 以顯示您的 Amazon DocumentDB 叢集。
2. 在您要在 AWS 主控台中檢視的叢集上按一下滑鼠右鍵，然後選擇在瀏覽器中開啟。
3. AWS 主控台會在您的預設 Web 瀏覽器中開啟 Amazon DocumentDB 叢集。

展開現有的叢集

若要透過新增執行個體來擴展 Amazon DocumentDB 叢集，請完成下列程序。

新增執行個體以擴展您的叢集

1. 從 AWS Toolkit for Visual Studio Code展開 Amazon DocumentDB 以顯示您的 Amazon DocumentDB 叢集。
2. 在您要展開的叢集上按一下滑鼠右鍵，然後選擇新增執行個體，以在 VS 程式碼中開啟新增執行個體對話方塊。
3. 出現提示時，請在文字欄位中輸入新執行個體的名稱，然後按 **Enter**鍵繼續。
4. 出現提示時，請從清單中選擇執行個體類別以繼續。
5. AWS Explorer 會在新執行個體就緒時顯示建立狀態和更新。

停止叢集

完成下列程序以停止您的 Amazon DocumentDB 叢集。

> **Note**
>
> 當您的叢集停止時，大多數叢集管理功能將無法使用。

停止 Amazon DocumentDB 叢集

1. 從 AWS Toolkit for Visual Studio Code展開 Amazon DocumentDB 以顯示您的 Amazon DocumentDB 叢集。
2. 選擇您要停止之叢集旁邊的停止叢集按鈕，或在叢集上按一下滑鼠右鍵，然後選擇停止叢集。
3. 出現提示時，請選擇是以停止叢集，或選擇取消以取消停止程序並讓叢集保持執行狀態。
4. AWS Explorer 會顯示叢集的狀態，並在叢集停止時更新。

重新啟動執行個體

重新啟動執行個體對於疑難排解和進行次要變更非常有用，而不會影響整個叢集。若要重新啟動 Amazon DocumentDB 執行個體，請完成下列程序。

重新啟動叢集執行個體

1. 從 AWS Toolkit for Visual Studio Code展開 Amazon DocumentDB 以顯示您的 Amazon DocumentDB 叢集。
2. 在您要重新啟動的叢集執行個體上按一下滑鼠右鍵，然後選擇重新啟動執行個體。
3. 出現提示時，請選擇是以重新啟動執行個體，或選擇取消以取消重新啟動程序，並讓執行個體停止。
4. AWS Explorer 會顯示叢集的狀態，並在執行個體重新啟動時更新。

刪除執行個體

若要刪除 Amazon DocumentDB 叢集執行個體，請完成下列程序。

> **Note**
>
> 刪除執行個體不會影響您叢集中的資料。如果您刪除主要執行個體，其中一個複本執行個體會接管為可寫入的執行個體。

刪除叢集執行個體

1. 從 AWS Toolkit for Visual Studio Code展開 Amazon DocumentDB 以顯示您的 Amazon DocumentDB 叢集。
2. 在要刪除的叢集執行個體上按一下滑鼠右鍵，然後選擇刪除以在 VS 程式碼中開啟 delete-cluster-instance 確認對話方塊。
3. 遵循確認提示，然後按 **Enter**鍵刪除叢集執行個體。
4. AWS Explorer 會顯示叢集執行個體的狀態，並在執行個體刪除時更新。

檢視、新增和移除標籤

標籤用於組織和追蹤您環境中的資源。若要檢視或編輯與您的 Amazon DocumentDB 叢集相關聯的標籤，請完成下列其中一個程序。

檢視叢集標籤

1. 從 AWS Toolkit for Visual Studio Code展開 Amazon DocumentDB 以顯示您的 Amazon DocumentDB 叢集。
2. 在您要檢視標籤的叢集上按一下滑鼠右鍵，然後選擇標籤...以開啟對話方塊的標籤**your cluster name**。
3. 您的標籤會顯示在對話方塊中，如果沒有標籤與您的叢集相關聯，則會顯示訊息 【未指派標籤】。

將標籤新增至叢集

1. 從 AWS Toolkit for Visual Studio Code展開 Amazon DocumentDB 以顯示您的 Amazon DocumentDB 叢集。
2. 在您要新增標籤的叢集上按一下滑鼠右鍵，然後選擇標籤...以開啟對話方塊的標籤**your cluster name**。
3. 選擇新增標籤...按鈕，在 VS 程式碼中開啟新增標籤對話方塊。

4. 在文字欄位中輸入新標籤，然後按 Enter 鍵繼續。
5. 在文字欄位中輸入值，然後按 Enter 將鍵/值對新增至叢集。

從叢集移除標籤

1. 從 AWS Toolkit for Visual Studio Code展開 Amazon DocumentDB 以顯示您的 Amazon DocumentDB 叢集。
2. 在您要從中移除標籤的叢集上按一下滑鼠右鍵，然後選擇標籤...以開啟對話方塊的標籤**`your cluster name`**。
3. 選擇移除標籤...按鈕以開啟 VS 程式碼中的從對話方塊移除標籤**`your cluster name`**。
4. 從提供的清單中選擇您要移除的標籤，以從叢集中移除標籤。

修改執行個體類別

若要修改 Amazon DocumentDB 叢集執行個體的類別，請完成下列程序。

修改執行個體類別

1. 從 AWS Toolkit for Visual Studio Code展開 Amazon DocumentDB 以顯示您的 Amazon DocumentDB 叢集。
2. 在您要修改的叢集執行個體上按一下滑鼠右鍵，然後選擇修改類別...，以在 VS 程式碼中開啟選取執行個體類別對話方塊。
3. 從清單中選擇執行個體的新類別，以更新類別。
4. AWS Explorer 會顯示叢集執行個體的狀態，並在執行個體的類別更新時更新。

Amazon Elastic Compute Cloud

適用於 的 Amazon Elastic Compute Cloud AWS Toolkit for Visual Studio Code 可讓您從 VS 程式碼啟動並連線至 Amazon EC2 執行個體。如需 Amazon EC2 的詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的什麼是 Amazon EC2？ 主題。

下列主題說明如何從 使用 AWS Application Builder AWS Toolkit for Visual Studio Code。

主題

- 使用 Amazon Elastic Compute Cloud
- Amazon Elastic Compute Cloud 故障診斷

使用 Amazon Elastic Compute Cloud

下列各節說明如何在 中使用 Amazon Elastic Compute Cloud AWS Toolkit for Visual Studio Code。

先決條件

本使用者指南主題中所述的功能已在具有下列作業系統的 Amazon EC2 執行個體上進行測試：

- Windows 2016+

> **Note**
>
> 此作業系統僅適用於連接 VS 程式碼終端機。它在連接完整的 VS Code 遠端執行個體時無法運作。如需 VS Code 終端機和遠端執行個體的詳細資訊，請參閱 VS Code 文件中的終端機入門和 VS Code Remote Development 主題。 https://code.visualstudio.com/docs/remote/remote-overview

- Amazon Linux 2023
- Ubuntu，22.04

需要本機安裝的 SSH 才能開啟 Amazon EC2 執行個體的遠端連線，但不需要開啟 Amazon EC2 執行個體的終端機。

您的 Amazon EC2 執行個體描述檔必須包含下列 AWS Identity and Access Management (IAM) 許可。

```
"ssmmessages:CreateControlChannel",
"ssmmessages:CreateDataChannel",
"ssmmessages:OpenControlChannel",
"ssmmessages:OpenDataChannel",
"ssm:DescribeAssociation",
"ssm:ListAssociations",
"ssm:UpdateInstanceInformation
```

> **Note**
>
> 必要的許可包含在下列 AWS 受管政策中。

- AmazonSSMManagedInstanceCore
- AmazonSSMManagedEC2InstanceDefaultPolicy

檢視現有的 Amazon EC2 執行個體

若要從 AWS Toolkit 檢視現有的 Amazon EC2 執行個體，請完成下列步驟。

1. 從 AWS Toolkit 中，展開 AWS Toolkit Explorer。
2. 展開包含您要檢視之 Amazon EC2 執行個體的區域。
3. 展開 EC2 標題以顯示現有的 Amazon EC2 執行個體。

啟動新的 Amazon EC2 執行個體

使用 AWS Toolkit 建立新的 Amazon EC2 執行個體有 3 種方式。

每個工作流程都會在主控台中 AWS 開啟啟動執行個體精靈。如需從啟動執行個體精靈啟動新 Amazon EC2 執行個體的詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的使用主控台主題中的啟動執行個體精靈啟動 EC2 執行個體。若要啟動新的 Amazon EC2 執行個體，請完成下列其中一個程序。

從 VS Code Command Palette 啟動新的 Amazon EC2 執行個體

1. 從 VS 程式碼中，按 **command + shift + P (Windows: ctrl + shift + P)**
2. 從 VS Code Command Palette 中，搜尋命令並在填入清單中時選取該**AWS: Launch EC2**命令，以在 VS Code 中開啟啟動 EC2 執行個體選取區域提示。
3. 從啟動 EC2 執行個體選取區域提示中，選擇您要在其中啟動新執行個體的區域，然後確認您想要在預設 Web 瀏覽器中開啟 AWS 主控台。
4. 在預設 Web 瀏覽器的 AWS 主控台中，完成身分驗證程序以繼續啟動執行個體精靈。
5. 從啟動執行個體精靈中，完成必要的區段，然後選擇啟動執行個體按鈕以啟動新的 Amazon EC2 執行個體。
6. AWS Explorer 會更新以顯示新的 Amazon EC2 執行個體。

從 AWS Explorer 啟動新的 Amazon EC2 執行個體

1. 展開 AWS Toolkit Explorer，然後展開您要在其中建立新的 Amazon EC2 執行個體的區域。

2. 展開或暫留在 EC2 標題上，然後選擇 +（啟動 EC2 執行個體）圖示。
3. 出現提示時，請確認您想要在預設 Web 瀏覽器中開啟 AWS 主控台。
4. 從 Web 瀏覽器的 AWS 主控台中，完成身分驗證程序以繼續啟動執行個體精靈。
5. 從啟動執行個體精靈中，完成必要的區段，然後選擇啟動執行個體按鈕以啟動新的 Amazon EC2 執行個體。
6. AWS Explorer 會更新以顯示新的 Amazon EC2 執行個體。

從內容選單啟動新的 Amazon EC2 執行個體（按一下滑鼠右鍵）

1. 展開 AWS Toolkit Explorer，然後展開您要在其中建立新的 Amazon EC2 執行個體的區域。
2. 在 EC2 標題上按一下滑鼠右鍵，然後選擇啟動 EC2 執行個體。
3. 出現提示時，請確認您想要在預設 Web 瀏覽器中開啟 AWS 主控台。
4. 從 Web 瀏覽器的 AWS 主控台中，完成身分驗證程序以繼續啟動執行個體精靈。
5. 從啟動執行個體精靈中，完成必要的區段，然後選擇啟動執行個體按鈕以啟動新的 Amazon EC2 執行個體。
6. AWS Explorer 會更新以顯示新的 Amazon EC2 執行個體。

將 VS 程式碼連線至 Amazon EC2 執行個體

有 3 種方式可從 VS Code 連線至 Amazon EC2 執行個體。若要將 VS 程式碼連線至 EC2 執行個體，請完成下列其中一個程序。

從命令面板將 VS 程式碼連線至 Amazon EC2 執行個體

1. 從 VS 程式碼中，按 **command + shift + P (Windows: ctrl + shift + P)**
2. 從 VS Code Command Palette 中搜尋**AWS: Connect VS Code to EC2 instance...**命令，並在命令填入清單中時選取命令，以在 VS Code 中開啟選取 EC2 執行個體提示。
3. 從選取 EC2 執行個體提示中，選擇包含您要連線之執行個體的區域，然後選擇您要連線的執行個體。
4. VS 程式碼會在建立連線時顯示狀態。
5. 新視窗隨即開啟，以在連線完成時顯示您的 Amazon EC2 執行個體。

從 AWS Explorer 將 VS 程式碼連線至 Amazon EC2 執行個體。

1. 展開 AWS Toolkit Explorer，然後展開包含您要連線之 Amazon EC2 執行個體的區域。

2. 將滑鼠游標暫留在 Amazon EC2 執行個體上，然後選擇 （將 VS 程式碼連接至 EC2 執行個體） 圖示。

> **Note**
>
> 您也可以從 AWS Explorer 的 EC2 服務標題中選擇 （將 VS 程式碼連接至 EC2 執行個體） 圖示。 EC2

3. VS 程式碼會在建立連線時顯示狀態。
4. 新視窗隨即開啟，以在連線完成時顯示您的 Amazon EC2 執行個體。

從右鍵選單將 VS 程式碼連線至 Amazon EC2 執行個體

1. 展開 AWS Toolkit Explorer，然後展開包含您要連線之 Amazon EC2 執行個體的區域。
2. 在您要連線的 Amazon EC2 執行個體上按一下滑鼠右鍵，然後選擇將 VS 程式碼連線至 EC2 執行個體。

> **Note**
>
> 您也可以在 AWS Explorer 中的 EC2 服務標題上按一下滑鼠右鍵，然後選擇將 VS 程式碼連線至 EC2 執行個體。

3. VS 程式碼會在建立連線時顯示狀態。
4. 新視窗隨即開啟，以在連線完成時顯示您的 Amazon EC2 執行個體。

開啟 Amazon EC2 執行個體的終端機。

有 3 種方式可從 VS 程式碼終端機連線至 Amazon EC2 執行個體。

從命令面板將 VS 程式碼連線至 Amazon EC2 執行個體

1. 從 VS 程式碼中，按 **`command + shift + P (Windows: ctrl + shift + P)`**
2. 從 VS Code Command Palette 中搜尋**`AWS:Open terminal to EC2 instance...`**命令，並在命令填入清單中時選取命令，以在 VS Code 中開啟選取 EC2 執行個體提示。
3. 從選取 EC2 執行個體提示中，選擇包含您要在終端機中開啟之執行個體的區域，然後選擇執行個體。
4. VS 程式碼會在建立連線時顯示狀態。

5. VS Code Terminal 會在連線完成時開啟以顯示您的新工作階段。

從 AWS Explorer 在 VS 程式碼終端機中開啟 Amazon EC2 執行個體。

1. 展開 AWS Toolkit Explorer，然後展開包含您要連線之 Amazon EC2 執行個體的區域。
2. 將滑鼠游標暫留在 Amazon EC2 執行個體上，然後選擇 （開啟終端機至 EC2 執行個體...) 圖示。

> **Note**
>
> 您也可以從 AWS Explorer 的 EC2 服務標題中選擇 （開啟終端機至 EC2 執行個體...) 圖示。 EC2

3. VS 程式碼會在建立連線時顯示狀態。
4. VS Code Terminal 會在連線完成時開啟以顯示您的新工作階段。

從右鍵選單在 VS 程式碼終端機中開啟 Amazon EC2 執行個體

1. 展開 AWS Toolkit Explorer，然後展開包含您要在 VS 程式碼終端機中開啟之 Amazon EC2 執行個體的區域。
2. 在您要在終端機中開啟的 Amazon EC2 執行個體上按一下滑鼠右鍵，然後選擇開啟終端機至 EC2 執行個體...。

> **Note**
>
> 您也可以在 AWS Explorer 中的 EC2 服務標題上按一下滑鼠右鍵，然後選擇開啟終端機至 EC2 執行個體...。

3. VS 程式碼會在建立連線時顯示狀態。
4. VS Code Terminal 會在連線完成時開啟以顯示您的新工作階段。

啟動或重新啟動 Amazon EC2 執行個體

有 3 種方式可以啟動或重新啟動 Amazon EC2 執行個體。

從命令面板重新啟動 Amazon EC2 執行個體

1. 從 VS 程式碼中，按 **command + shift + P (Windows: ctrl + shift + P)**

2. 從 VS Code Command Palette 中搜尋**AWS: Reboot EC2 instance**命令，並在命令填入清單中時選取命令，以在 VS Code 中開啟選取 EC2 執行個體提示。

> **Note**
>
> 若要啟動未執行的執行個體，您必須選擇 **AWS: Start EC2 instance**命令。**AWS: Reboot EC2 instance** 命令只會重新啟動目前正在執行的執行個體。

3. 從選取 EC2 執行個體提示中，選擇包含您要啟動或重新啟動之執行個體的區域。
4. VS 程式碼會在執行個體重新啟動時顯示狀態。
5. AWS Explorer 會更新以顯示您的執行個體在完成重新啟動時正在執行。

從 AWS Explorer 啟動或重新啟動 Amazon EC2 執行個體

1. 展開 AWS Toolkit Explorer，然後展開包含您要啟動或重新啟動之 Amazon EC2 執行個體的區域。
2. 將滑鼠暫留在 Amazon EC2 執行個體上，然後選擇 （ 重新啟動 EC2 執行個體 ） 圖示。

> **Note**
>
> 如果執行個體已停止，則唯一的選項是 （ 啟動 EC2 執行個體 ） 圖示

3. VS 程式碼會在執行個體重新啟動時顯示狀態。
4. AWS Explorer 會更新以顯示您的執行個體在完成重新啟動時正在執行。

從右鍵選單啟動或重新啟動 Amazon EC2 執行個體

1. 展開 AWS Toolkit Explorer，然後展開包含您要啟動或重新啟動之 Amazon EC2 執行個體的區域。
2. 在您要連線的 Amazon EC2 執行個體上按一下滑鼠右鍵，然後選擇重新啟動 EC2 執行個體。

> **Note**
>
> 如果執行個體停止，則唯一的選項是啟動 EC2 執行個體。

3. VS 程式碼會在執行個體重新啟動時顯示狀態。
4. AWS Explorer 會更新以顯示您的執行個體在完成重新啟動時正在執行。

停止 Amazon EC2 執行個體

有 3 種方法可以停止 Amazon EC2 執行個體。

從命令面板停止 Amazon EC2 執行個體

1. 從 VS 程式碼中，按 **command + shift + P (Windows: ctrl + shift + P)**
2. 從 VS Code Command Palette 中搜尋**AWS: Stop EC2 instance**命令，並在命令填入清單中時選取命令，以在 VS Code 中開啟選取 EC2 執行個體提示。
3. 從選取 EC2 執行個體提示中，選擇包含您要停止之執行個體的區域。
4. VS 程式碼會在執行個體停止時顯示狀態。
5. AWS Explorer 會更新以顯示您的執行個體已停止。

從 AWS Explorer 停止 Amazon EC2 執行個體

1. 展開 AWS Toolkit Explorer，然後展開包含您要停止之 Amazon EC2 執行個體的區域。
2. 將滑鼠游標暫留在 Amazon EC2 執行個體上，然後選擇 （停止 EC2 執行個體） 圖示。
3. VS 程式碼會在執行個體停止時顯示狀態。
4. AWS Explorer 會更新以顯示您的執行個體已停止。

從右鍵功能表停止 Amazon EC2 執行個體

1. 展開 AWS Toolkit Explorer，然後展開包含您要停止之 Amazon EC2 執行個體的區域。
2. 在您要連線的 Amazon EC2 執行個體上按一下滑鼠右鍵，然後選擇重新啟動 EC2 執行個體。
3. VS 程式碼會在執行個體停止時顯示狀態。
4. AWS Explorer 會更新以顯示您的執行個體已停止。

複製執行個體 ID

若要複製執行個體 ID，請完成下列步驟。

1. 在您想要從中複製 ID 的執行個體上按一下滑鼠右鍵。
2. 選擇複製執行個體 ID。
3. 執行個體 ID 會複製到本機剪貼簿。

複製名稱

若要複製執行個體名稱，請完成下列步驟。

1. 在您想要從中複製名稱的執行個體上按一下滑鼠右鍵。
2. 選擇複製執行個體名稱。
3. 執行個體名稱會複製到本機剪貼簿。

複製 ARN

若要複製執行個體 ARN，請完成下列步驟。

1. 在您想要從中複製 ARN 的執行個體上按一下滑鼠右鍵。
2. 選擇複製執行個體 ARN。
3. 執行個體 ARN 會複製到您的本機剪貼簿。

Amazon Elastic Compute Cloud 故障診斷

下列各節說明如何疑難排解在 中使用 Amazon Elastic Compute Cloud 時可能發生的已知問題 AWS Toolkit for Visual Studio Code。如需疑難排解 Amazon EC2 服務特定問題的詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的疑難排解 Amazon EC2 執行個體問題主題。

一般偵錯

如果您因任何原因遇到遠端連線問題，請先檢查是否可以從 AWS 主控台建立 AWS Systems Manager 連線。

若要從主控台透過 Systems Manager 連線至 Amazon EC2 執行個體 AWS ，請完成下列步驟。

1. 從 Web 瀏覽器導覽至 AWS 主控台。
2. 完成身分驗證以繼續主控台 AWS EC2 登陸。
3. 從 Amazon EC2 導覽窗格中，選擇執行個體。
4. 選取您要連線之執行個體旁的方塊。
5. 選擇連線按鈕，在新的瀏覽器索引標籤中開啟連線至執行個體畫面。

> **Note**
>
> 您只能連線到執行中的執行個體。如果您無法選取連線按鈕，請檢查您的執行個體是否正在執行。

6. 在連線至執行個體畫面中，選擇工作階段管理員索引標籤，然後選擇連線按鈕，以在目前的瀏覽器索引標籤中開啟 Systems Manager 連線。

> **Note**
>
> 如果您最近啟動執行個體，但無法使用 選項來連接 Systems Manager，您可能需要再等待幾分鐘，該選項才會變成可用。

目標執行個體未執行

若要從終端機或遠端連線連線至 Amazon EC2 執行個體，執行個體必須正在執行。嘗試從 AWS Toolkit 連線至執行個體之前，請從 AWS Explorer AWS 管理主控台或 啟動執行個體 AWS Command Line Interface。

目標執行個體沒有 IAM 角色或具有不當許可的 IAM 角色

若要連線至 Amazon EC2 執行個體，它必須具有已連接正確許可的 IAM 角色。如果您嘗試連線到未連接 IAM 角色的執行個體，您會收到 VS 程式碼的通知。

如果您嘗試連線到具有 IAM 角色但缺少必要許可的執行個體，系統會提示您新增最低必要動作做為現有 IAM 角色的內嵌政策。更新內嵌政策後，您會連線到執行個體。如需 IAM 角色、許可和將角色連接至執行個體的詳細資訊，請參閱《Amazon Elastic Compute Cloud 使用者指南》中的 Amazon EC2 主題的 IAM 角色，以及AWS 《Systems Manager 使用者指南》中的步驟 2：驗證或新增工作階段管理員主題的執行個體許可。

下列範例包含最低必要動作。

```
"ssmmessages:CreateControlChannel",
"ssmmessages:CreateDataChannel",
"ssmmessages:OpenControlChannel",
"ssmmessages:OpenDataChannel",
"ssm:DescribeAssociation",
```

```
"ssm:ListAssociations",
"ssm:UpdateInstanceInformation
```

> **Note**
>
> 必要的許可包含在下列 AWS 受管政策中。
>
> - `AmazonSSMManagedEC2InstanceDefaultPolicy`
> - `AmazonSSMManagedInstanceCore`

目標執行個體沒有執行中的 Systems Manager 代理程式

您可能會因為許多不同的原因而遇到此問題。若要修正此問題，請先重新啟動執行個體並再次嘗試連線。或者，透過非 SMS 連線方法手動啟動初始連線。如需 Systems Manager 的詳細資訊，請參閱 Systems Manager 中的使用 Systems Manager 代理程式主題。 AWS

啟動時，Amazon EC2 狀態表示其正在執行，但連線未通過

如果您最近開始或建立了執行個體的新 IAM 角色，但無法建立連線，請再等待幾分鐘，然後再嘗試建立連線。

使用 Amazon Elastic Container Registry

Amazon Elastic Container Registry (Amazon ECR) 是安全且可擴展的 AWS 受管容器登錄服務。您可以從 Toolkit for VS Code Explorer 存取數個 Amazon ECR 服務函數。

- 建立儲存庫。
- 為您的儲存庫或標記的映像建立 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。

您也可以透過 VS Code 主控台，將 CLI 和其他平台與 VS Code AWS 整合，以存取 Amazon ECR 函數的完整範圍。

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？。

主題

- 使用 Amazon Elastic Container Registry
- 透過 Amazon ECR 建立 App Runner 服務

使用 Amazon Elastic Container Registry

您可以直接從 VS Code 中的 AWS Explorer 存取 Amazon Elastic Container Registry (Amazon ECR) 服務，並使用它將程式映像推送到 Amazon ECR 儲存庫。若要開始使用，您需要執行下列步驟：

1. 建立一個 Dockerfile，其中包含建置映像檔所需的資訊。
2. 從該 Dockerfile 建置映像檔並標記映像檔以進行處理。
3. 在 Amazon ECR 執行個體內建立儲存庫。
4. 將映像檔推送到您的儲存庫。

先決條件

您需要完成這些步驟，才能從 VS Code 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。這是為了讓服務可以判斷您是否具有存取其資源的許可。我們不建議您透過根 AWS 帳戶的登入資料 AWS 直接存取 。反之，請使用 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。然後，您可以使用 AWS 特殊 URL 和 IAM 使用者的登入資料來存取 。

如果您註冊 ， AWS 但未自行建立 IAM 使用者，您可以使用 IAM 主控台建立一個使用者。

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	到	根據	您也可以
在 IAM Identity Center	使用短期憑證存取 AWS。	請遵循 AWS IAM Identity Center 使用者指南的入門中的說明。	透過在 AWS Command Line Interface 使用者指南中設定 AWS CLI 以使用 來設定 AWS

選擇一種管理管理員的方式	到	根據	您也可以
(建議)	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 IAM 安全最佳實務。		IAM Identity Center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》中建立 IAM 使用者以進行緊急存取 的指示。	請依照《IAM 使用者指南》中的管理 IAM 使用者的存取金鑰設定以程式設計方式存取。

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 是不含連字號的 AWS 帳號（例如，如果 AWS 您的帳戶號碼是 1234-5678-9012，則 AWS 您的帳戶 ID 為 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 AWS 您的帳戶 ID，您可以建立帳戶別名。從 IAM 儀表板中，選擇自訂並輸入帳戶別名。這可以是您的公司名稱。如需詳細資訊，請參閱《IAM 使用者指南》中的AWS 您的帳戶 ID 及其別名。

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》。

安裝和設定 Docker

您可以從安裝 Docker 引擎使用者指南中選取您偏好的作業系統，並遵循指示來安裝和設定 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 CLI 第 2 版。 AWS

1. 建立 Dockerfile。

Docker 使用稱為 Dockerfile 的檔案來定義可在遠端儲存庫上推送和存放的映像。您必須先建立 Dockerfile，然後從該 Dockerfile 建置映像，才能將映像上傳至 ECR 儲存庫。

建立 Dockerfile。

1. 使用 Toolkit for VS Code Explorer 導覽至您要存放 Dockerfile 的目錄。
2. 建立稱為 Dockerfile 的新檔案。

> **Note**
>
> VS 程式碼可能會提示您選取檔案類型或副檔名。如果發生這種情況，請選取純文字。Vs Code 具有「dockerfile」副檔名。不過，我們建議您不要使用它。這是因為副檔名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 程式碼編輯您的 Dockerfile

如果 Dockerfile 有副檔名，請開啟該檔案的內容 (按一下滑鼠右鍵) 選單，然後移除副檔名。

從 Dockerfile 中刪除副檔名後：

1. 直接在 VS 程式碼中開啟空的 Dockerfile。
2. 將下列範例的內容複製到 Dockerfile：

   Example Dockerfile 映像檔範本

```
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 映像檔的 Dockerfile。RUN 指令會更新套件快取。安裝 Web 伺服器的軟體套件服務，然後寫入「Hello World!」 內容至 Web 伺服器的文件根目錄。EXPOSE 指令會公開容器上的連接埠 80，而 CMD 指令會啟動 Web 伺服器。

3. 儲存 Dockerfile。

> **Important**
>
> 請確定您的 Dockerfile 沒有連接至名稱的副檔名。具有副檔名的 Dockerfile 可能會導致與 Docker 或其他相關應用程式的某些版本發生衝突。

2. 從 Dockerfile 建置映像

您建立的 Dockerfile 包含建置程式映像所需的資訊。您必須先建置映像，才能將該映像推送至 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，導覽至包含 Dockerfile 的目錄。
2. 執行 Docker build 命令來建置 Dockerfile 中定義的映像。

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

3. 執行 Docker 映像命令來驗證映像是否正確建立。

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

Example輸出範例：

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

4.

> **Note**
>
> 此步驟並非建立或推送映像的必要步驟，但您可以查看程式映像在執行時的運作方式。

若要執行新建置的映像，請使用 Docker run 命令。

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

上述範例中指定的 -p 選項會將容器上的公開連接埠 80 映射至主機系統的連接埠 80。如果您在本機執行 Docker，請使用 Web 瀏覽器導覽至 http：//localhost：80。如果程式正確執行，則為「Hello World！」 隨即顯示 陳述式。

如需 Docker run 命令的詳細資訊，請參閱 Docker 網站上的 Docker run 參考資料。

3. 建立新的儲存庫

若要將映像檔上傳到 Amazon ECR 執行個體，請建立可存放它的新儲存庫。

建立新的 Amazon ECR 儲存庫

1. 從 VS 程式碼活動列中，選擇 AWS Toolkit 圖示。
2. 展開 AWS Explorer 功能表。

3. 找到與您 AWS 帳戶相關聯的預設 AWS 區域。然後，選取它以查看透過 Toolkit for VS 程式碼的服務清單。
4. 選擇 ECR + 選項以開始建立新儲存庫程序。
5. 依照提示完成程序。
6. 完成後，您可以從 AWS Explorer 功能表的 ECR 區段存取您的新儲存庫。

4. 推送、提取和刪除映像

從 Dockerfile 建立映像檔並建立儲存庫之後，您可以將映像檔推送到 Amazon ECR 儲存庫。此外，使用 AWS Explorer 搭配 Docker 和 AWS CLI，您可以執行下列動作：

- 從儲存庫中提取映像檔。
- 刪除儲存庫中儲存的映像檔。
- 刪除您的儲存庫。

使用預設登錄檔驗證 Docker

在 Amazon ECR 和 Docker 執行個體之間交換資料時，需要進行身分驗證。若要向您的登錄檔驗證 Docker：

1. 開啟連接到 CLI AWS 執行個體的命令列作業系統。
2. 使用 get-login-password 方法來驗證您的私人 ECR 登錄檔。

```
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，您必須同時將 **region**和 更新**AWS_account_id**為 AWS 帳戶特有的資訊。

標記映像並將其推送到您的儲存庫

使用 執行個體驗證 Docker 之後 AWS，請將映像推送到您的儲存庫。

1. 使用 Docker 映像命令來檢視您儲存在本機的映像，並識別您要標記的映像。

```
docker images
```

Example輸出範例：

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

2. 使用 Docker tag 命令標記 Docker 映像檔。

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

3. 使用 Docker 標籤命令將標記的映像推送到您的儲存庫。

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

Example輸出範例：

```
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
```

標記的映像成功上傳至儲存庫後，它就會顯示在 AWS Explorer 功能表中。

從 Amazon ECR 提取映像

- 您可以將映像檔提取到 Docker tag 命令的本機執行個體。

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

Example輸出範例：

```
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 儲存庫刪除映像

有兩種方法可從 VS 程式碼刪除映像。第一個方法是使用 AWS Explorer。

1. 從 AWS Explorer 展開 ECR 功能表
2. 展開您要從中刪除映像的儲存庫
3. 開啟內容功能表（按一下滑鼠右鍵），選擇與要刪除之影像相關聯的影像標籤
4. 選擇刪除標籤...選項，以刪除與該標籤相關聯的所有預存映像

使用 CLI AWS 刪除映像

- 您也可以使用 AWS ecr batch-delete-image 命令刪除儲存庫中的映像檔。

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

Example輸出範例：

```
{
    "failures": [],
```

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

從 Amazon ECR 執行個體刪除儲存庫

有兩種方法可從 VS 程式碼刪除儲存庫。第一個方法是使用 AWS Explorer。

1. 從 AWS Explorer 展開 ECR 功能表
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. 選擇所選儲存庫的刪除儲存庫...選項

從 CLI 刪除 Amazon ECR AWS 儲存庫

- 您可以使用 AWS ecr delete-repository 命令刪除儲存庫。

> **Note**
>
> 根據預設，您無法刪除包含映像的儲存庫。不過， --force 旗標允許此操作。

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

Example輸出範例：

```
{
    "failures": [],
    "imageIds": [
        {
```

```
            "imageTag": "latest",
            "imageDigest":
 "sha256:215d7e4121b30157d8839e81c4e0912606fca105775bb0636b95aed25f52c89b"
        }
    ]
}
```

透過 Amazon ECR 建立 App Runner 服務

下列主題說明如何從 中的 Amazon Elastic Container Registry (Amazon ECR) 節點建立和啟動 AWS App Runner 服務 AWS Toolkit for Visual Studio Code。如需 AWS App Runner 和 Amazon ECR 服務的詳細資訊，請參閱 AWS App Runner和 Amazon ECR 使用者指南。

先決條件

您必須先完成下列操作，才能在 AWS Toolkit 中 AWS App Runner 從 Amazon ECR 建立和啟動 。如需如何完成這些程序的詳細指南，請參閱本使用者指南中的使用 Amazon Elastic Container Registry 主題。

1. 建立 `dockerfile`。
2. 從 建置映像`dockerfile`。
3. 建立新的儲存庫。
4. 標記並推送映像到您的儲存庫。

從現有的 Amazon ECR 儲存庫建立 AWS App Runner 服務

下列程序說明如何在 AWS Toolkit 中從現有的 Amazon ECR 儲存庫建立 AWS App Runner 服務。

1. 從 AWS Explorer 中，展開包含您要從中建立 AWS App Runner 服務的 Amazon ECR 儲存庫的區域。
2. 展開 Amazon ECR 服務節點以檢視您的 Amazon ECR 儲存庫。
3. 開啟您要從中建立 AWS App Runner 服務的 Amazon ECR 儲存庫或儲存庫映像的內容選單 （按一下滑鼠右鍵）。
4. 從內容功能表中，選擇建立 App Runner Service，以在 VS 程式碼中開啟 AWS App Runner 建立精靈

5. 從輸入新服務的連接埠 (1/5) 中，輸入您要使用的連接埠號碼，然後按 **Enter** 繼續。
6. 從設定環境變數 (2/5) 中，選擇使用檔案... 瀏覽以選擇瀏覽您的本機檔案，或選擇略過以略過此步驟。
7. 從選取要從 ECR 提取的角色 (3/5)，從清單中選擇現有的 IAM 角色。

> **Note**
>
> 需要 AppRunnerECRAccessRole 存取角色，才能從 Amazon ECR 私有登錄檔建立 AWS App Runner 服務。如果清單中沒有可用的有效角色，請選擇 + （建立角色...) 圖示，以自動建立 AppRunnerECRAccessRole 並將其指派給您的登錄檔。

8. 從為您的服務命名 (4/5)，輸入新服務的名稱，然後按 **Enter** 繼續。
9. 從選取執行個體組態 (5/5) 從清單中選擇 **vCPU**和 **Memory**組態，以建立新的服務。
10. 從 AWS Explorer 展開 App Runner 服務節點以檢視您的 AWS App Runner 資源。當您的新服務建立成功時，狀態會自動更新為執行中。

使用 Amazon Elastic Container Service

AWS Toolkit for Visual Studio Code 提供 Amazon Elastic Container Service (Amazon ECS) 的一些支援。Toolkit for VS Code 可協助您進行特定 Amazon ECS 相關工作，例如建立任務定義。

主題

- 使用 IntelliSense for Amazon ECS 任務定義檔案
- 中的 Amazon Elastic Container Service Exec AWS Toolkit for Visual Studio Code

使用 IntelliSense for Amazon ECS 任務定義檔案

使用 Amazon Elastic Container Service (Amazon ECS) 時，您可以執行的其中一項操作是建立任務定義，如 Amazon Elastic Container Service 開發人員指南中的建立任務定義所述。安裝 時 AWS Toolkit for Visual Studio Code，安裝包含適用於 Amazon ECS 任務定義檔案的 IntelliSense 功能。

先決條件

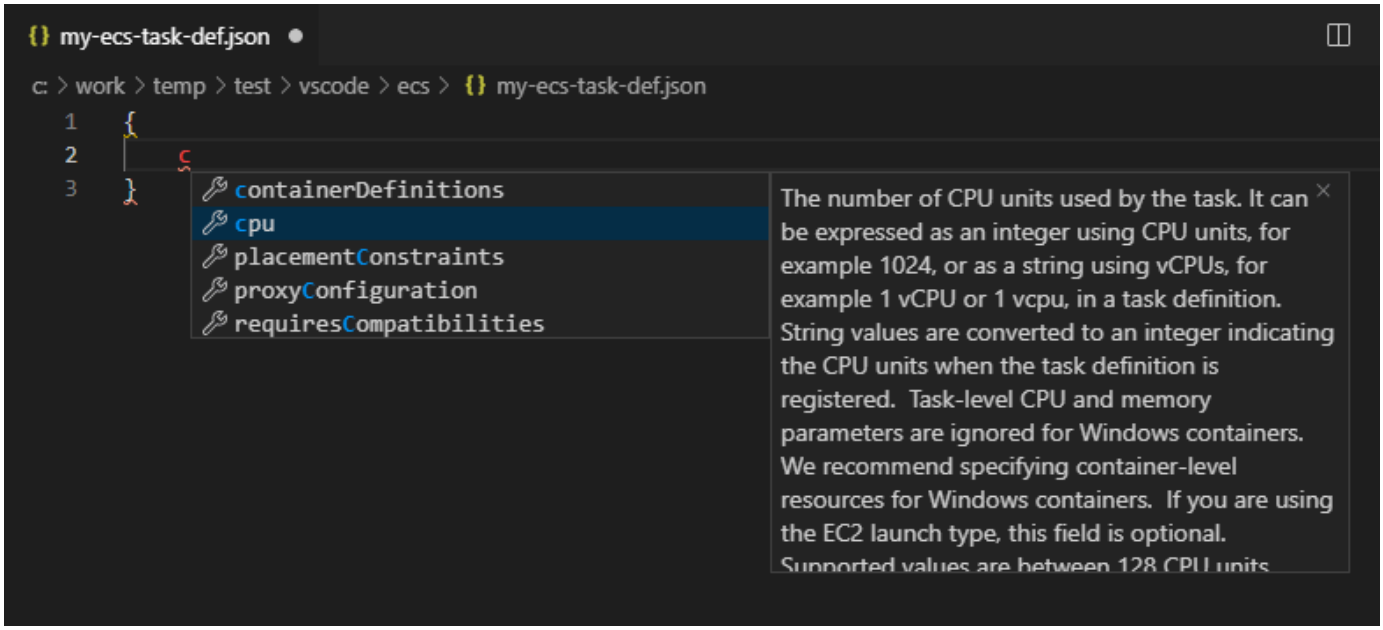
- 請確定您的系統符合安裝 Toolkit for VS 程式碼中指定的先決條件。

在 Amazon ECS 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 ECS 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 程式碼編輯器中開啟 檔案，然後輸入初始大括號。
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。



中的 Amazon Elastic Container Service Exec AWS Toolkit for Visual Studio Code

您可以使用 Amazon ECS Exec 功能 AWS Toolkit for Visual Studio Code，透過 在 Amazon Elastic Container Service (Amazon ECS) 容器中發出單一命令。

> **Important**
>
> 啟用和停用 Amazon ECS Exec 會變更您 AWS 帳戶中資源的狀態。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，請參閱開發人員指南使用 Amazon ECS Exec 進行偵錯。

Amazon ECS Exec 的先決條件

您必須先符合一些先決條件，才能使用 Amazon ECS Exec 功能。

Amazon ECS 要求

視您的任務是託管在 Amazon EC2 還是 而定 AWS Fargate，Amazon ECS Exec 有不同的版本需求。

- 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，則必須使用平台版本 1.4.0 或更新版本。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，您需要有與 AWS 帳戶相關聯的現有 Amazon ECS 叢集。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

您可以在 Toolkit for VS Code 中直接從 AWS Explorer 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，然後對其執行命令。

啟用 Amazon ECS Exec

1. 從 AWS Explorer 找到並展開 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. 從 AWS Explorer 找到並展開 Amazon ECS 功能表。
2. 展開存放所需服務的叢集。
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 Explorer 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. 從 AWS Explorer 找到並展開 Amazon ECS 功能表。
2. 展開存放所需服務的叢集。
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. 將開啟一個提示，其中包含執行中的任務清單，選擇您想要的任務 ARN。

   > **Note**
   >
   > 如果該服務只執行一個任務，則會自動選取該任務，並略過此步驟。

6. 出現提示時，輸入您要執行的命令，然後按 Enter 來處理。

使用 Amazon EventBridge

AWS Toolkit for Visual Studio Code (VS 程式碼） 支援 Amazon EventBridge。使用 Toolkit for VS Code，您可以處理 EventBridge 的某些層面，例如結構描述。

主題

- 使用 Amazon EventBridge 結構描述

使用 Amazon EventBridge 結構描述

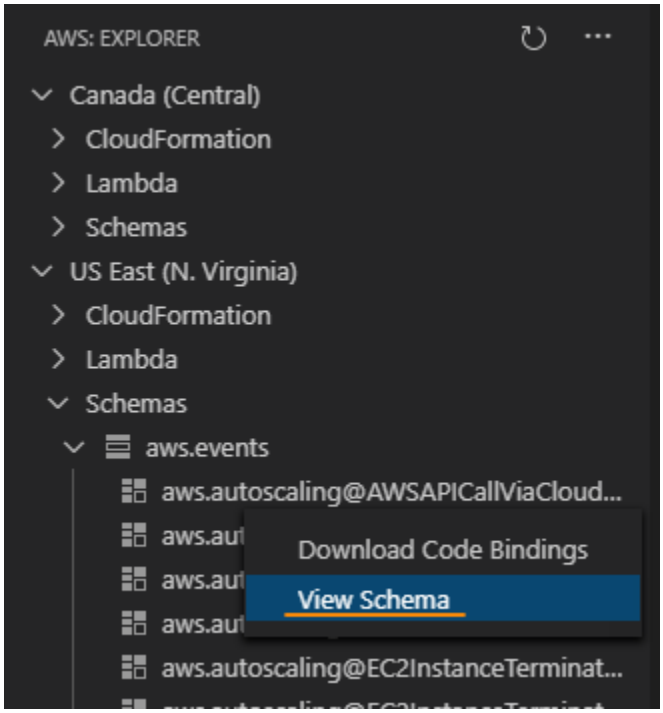
您可以使用 AWS Toolkit for Visual Studio Code (VS 程式碼） 在 Amazon EventBridge 結構描述上執行各種操作。

先決條件

- 請確定您的系統符合安裝 Toolkit for VS 程式碼中指定的先決條件。
- 您想要使用的 EventBridge 結構描述必須在 AWS 您的帳戶中使用。如果沒有，請建立或上傳。請參閱《Amazon EventBridge 使用者指南》中的 Amazon EventBridge 結構描述。 EventBridge

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 AWS 提供的結構描述都在 aws.events 登錄檔中。
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)。



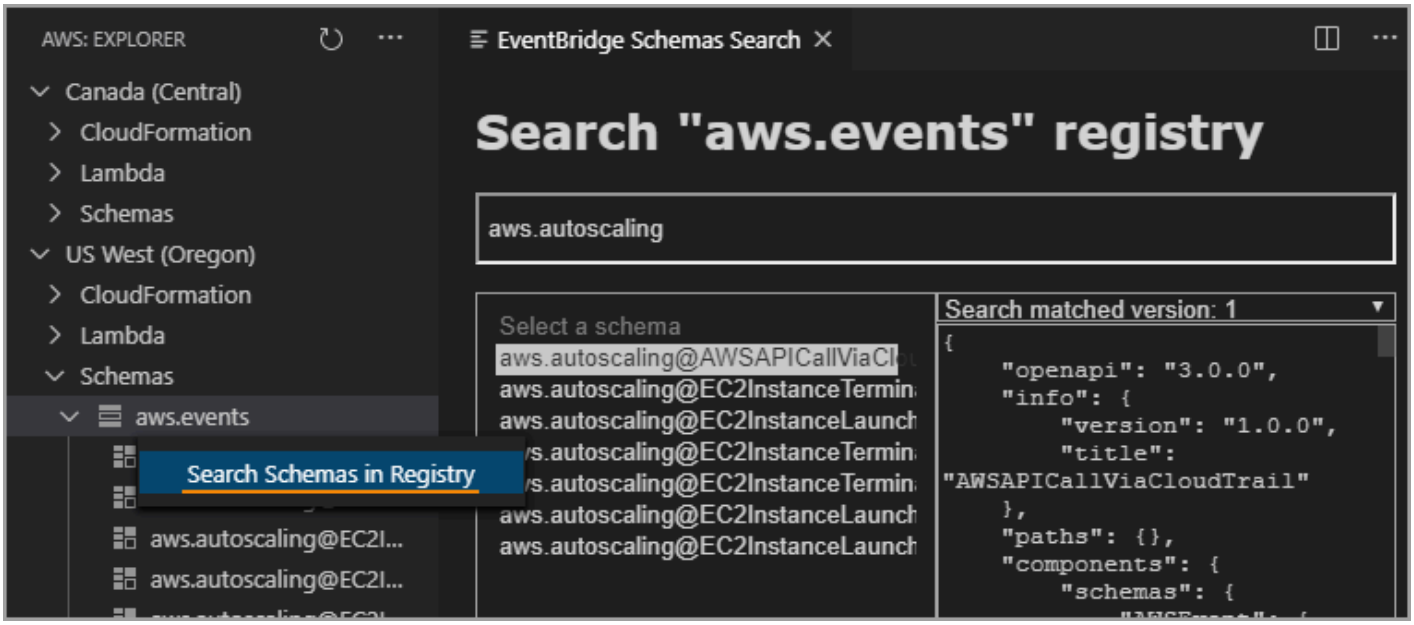
尋找可用的結構描述

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 EventBridge 結構描述搜尋對話方塊中，開始輸入您要尋找的結構描述標題。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 AWS IAM Access Analyzer，對範本、Terraform 計劃和 JSON 政策文件中撰寫的 IAM 政策執行 Identity and Access Management (IAM) Access Analyzer 政策檢查 AWS Toolkit for Visual Studio Code。 CloudFormation

IAM Access Analyzer 政策檢查包括政策驗證和自訂政策檢查。政策驗證有助於根據《 使用者指南》中 IAM JSON 政策語言語法和 IAM 主題中安全最佳實務中詳述的標準來驗證 IAM 政策AWS Identity and Access Management。 AWS https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html您的政策驗證調查結果包括安全警告、錯誤、一般警告和政策建議。

您也可以根據您的安全標準，針對新存取執行自訂政策檢查。每個自訂政策檢查是否有新存取都會產生費用。如需定價的詳細資訊，請參閱 AWS IAM Access Analyzer 定價網站。如需 IAM Access Analyzer 政策檢查的詳細資訊，請參閱AWS Identity and Access Management《 使用者指南》中的驗證政策的檢查主題。

下列主題說明如何在 中使用 IAM Access Analyzer 政策檢查 AWS Toolkit for Visual Studio Code。

主題

- 使用 AWS IAM Access Analyzer

使用 AWS IAM Access Analyzer

下列各節說明如何在 中執行 IAM 政策驗證和自訂政策檢查 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 Access Analyzer 政策驗證和 IAM Access Analyzer 自訂政策檢查。

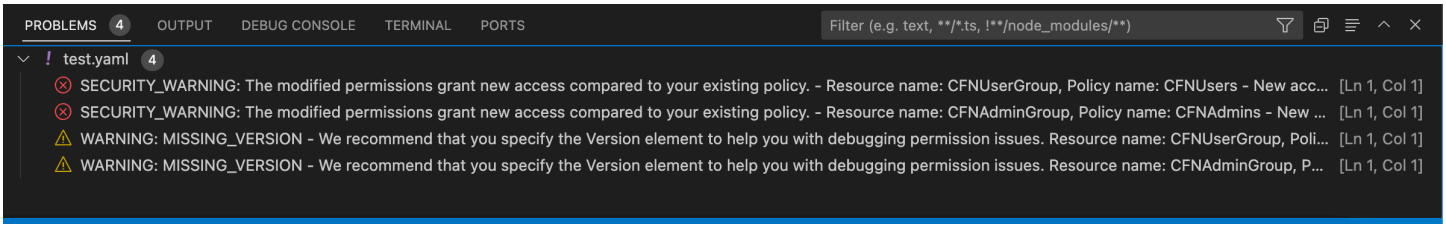
先決條件

您必須先符合下列先決條件，才能從 Toolkit 使用 IAM Access Analyzer 政策檢查。

- 安裝 Python 3.6 版或更新版本。
- 安裝 Python CLI 工具所需的適用於 的 IAM 政策驗證器 CloudFormation或適用於 Terraform 的 IAM 政策驗證器，並在 IAM 政策檢查視窗中指定。
- 設定您的 AWS 角色登入資料。

IAM Access Analyzer 政策檢查

您可以使用 對 CloudFormation 範本、Terraform 計劃和 JSON 政策文件執行政策檢查 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問題清單。下圖顯示 VS 程式碼問題面板。



IAM Access Analyzer 提供 4 種類型的檢查：

- 驗證政策
- CheckAccessNotGranted
- CheckNoNewAccess
- CheckNoPublicAccess

下列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 角色登入資料。支援的檔案包括下列文件類型： CloudFormation 範本、Terraform 計劃和 JSON 政策文件
> 檔案路徑參考通常由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。如需自訂政策檢查定價的詳細資訊，請參閱 AWS IAM Access Analyzer 定價指南。

執行驗證政策

驗證政策檢查也稱為政策驗證，會根據 IAM 政策文法和 AWS 最佳實務來驗證您的政策。如需詳細資訊，請參閱《 AWS Identity and Access Management使用者指南》中的 IAM JSON 政策語言的語法和 AWS IAM 主題中的安全最佳實務。

1. 從 VS 程式碼中，在 VS 程式碼編輯器中開啟包含 AWS IAM 政策的支援檔案。

2. 若要開啟 IAM Access Analyzer 政策檢查，請按 開啟 VS Code Command Palette**CRTL+Shift+P**、搜尋 **IAM Policy Checks**，然後按一下 以在 VS Code 編輯器中開啟 IAM 政策檢查窗格。
3. 從 IAM 政策檢查窗格中，從下拉式選單中選取您的文件類型。
4. 從驗證政策區段中，選擇執行政策驗證按鈕來執行驗證政策檢查。
5. 從 VS Code 中的問題面板中，檢閱您的政策檢查問題清單。
6. 更新您的政策並重複此程序，重新執行驗證政策檢查，直到政策檢查結果不再顯示安全警告或錯誤為止。

執行 CheckAccessNotGranted

CheckAccessNotGranted 是自訂政策檢查，確認您的政策不允許特定 IAM 動作。

> **Note**
>
> 檔案路徑參考通常由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，且檔案應在下列範例之後建構：
>
> ```
> {"actions": ["action1", "action2", "action3"], "resources":
> ["resource1", "resource2", "resource3"]}
> ```

1. 從 VS 程式碼中，在 VS 程式碼編輯器中開啟包含 AWS IAM 政策的支援檔案。
2. 若要開啟 IAM Access Analyzer 政策檢查，請按 開啟 VS Code Command Palette**CRTL+Shift+P**，搜尋 **IAM Policy Checks**，然後按一下 以在 VS Code 編輯器中開啟 IAM 政策檢查窗格。
3. 從 IAM 政策檢查窗格中，從下拉式選單中選取您的文件類型。
4. 從自訂政策檢查區段中，選取 CheckAccessNotGranted。
5. 在文字輸入欄位中，您可以輸入逗號分隔清單，其中包含動作和資源 ARNs。至少必須提供一個動作或資源。
6. 選擇執行自訂政策檢查按鈕。
7. 從 VS Code 中的問題面板中，檢閱您的政策檢查問題清單。自訂政策檢查會傳回 PASS或 FAIL結果。

8. 更新您的政策並重複此程序，重新執行 CheckAccessNotGranted 檢查，直到傳回 為止PASS。

執行 CheckNoNewAccess

CheckNoNewAccess 是一種自訂政策檢查，用於驗證您的政策是否授予與參考政策相比的新存取權。

1. 從 VS 程式碼中，在 VS 程式碼編輯器中開啟包含 AWS IAM 政策的支援檔案。
2. 若要開啟 IAM Access Analyzer 政策檢查，請按 開啟 VS Code Command Palette**CRTL+Shift+P**，搜尋 **IAM Policy Checks**，然後按一下 以在 VS Code 編輯器中開啟 IAM 政策檢查窗格。
3. 從 IAM 政策檢查窗格中，從下拉式選單中選取您的文件類型。
4. 從自訂政策檢查區段中，選取 CheckNoNewAccess。
5. 輸入參考 JSON 政策文件。或者，您可以提供參考 JSON 政策文件的檔案路徑。
6. 選取符合您參考文件類型的參考政策類型。
7. 選擇執行自訂政策檢查按鈕。
8. 從 VS Code 中的問題面板中，檢閱您的政策檢查問題清單。自訂政策檢查會傳回 PASS或 FAIL結果。
9. 更新您的政策並重複此程序，重新執行 CheckNoNewAccess 檢查，直到傳回 為止PASS。

執行 CheckNoPublicAccess

CheckNoPublicAccess 是一種自訂政策檢查，用於驗證您的政策是否授予範本中支援資源類型的公開存取權。

如需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. 從 VS 程式碼中，在 VS 程式碼編輯器中開啟包含 AWS IAM 政策的支援檔案。
2. 若要開啟 IAM Access Analyzer 政策檢查，請按 開啟 VS Code Command Palette**CRTL+Shift+P**，搜尋 **IAM Policy Checks**，然後按一下 以在 VS Code 編輯器中開啟 IAM 政策檢查窗格。
3. 從 IAM 政策檢查窗格中，從下拉式選單中選取您的文件類型。
4. 從自訂政策檢查區段中，選取 CheckNoPublicAccess。
5. 選擇執行自訂政策檢查按鈕。
6. 從 VS Code 中的問題面板中，檢閱您的政策檢查問題清單。自訂政策檢查會傳回 PASS或 FAIL結果。
7. 更新您的政策並重複此程序，重新執行 CheckNoNewAccess 檢查，直到傳回 為止PASS。

在 AWS IoT 中使用 AWS Toolkit for Visual Studio Code

AWS IoT 中的 AWS Toolkit for Visual Studio Code 可讓您與服務互動 AWS IoT ，同時將 VS Code 中工作流程的中斷降至最低。此使用者指南旨在協助您開始使用 中可用的 AWS IoT 服務功能 AWS Toolkit for Visual Studio Code。如需 AWS IoT 服務的其他資訊，請參閱開發人員指南什麼是 AWS IoT？

AWS IoT 先決條件

若要 AWS IoT 從 Toolkit for VS Code 開始使用 ，請確定 AWS 您的帳戶和 VS Code 符合下列指南中的要求：

- 如需 AWS IoT 服務特定的 AWS 帳戶需求和 AWS 使用者許可，請參閱 AWS IoT 核心開發人員入門指南。
- 如需 Toolkit for VS Code 特定需求，請參閱設定 Toolkit for VS Code 使用者指南。

AWS IoT 實物

AWS IoT 會將裝置連線至 AWS 雲端服務和資源。您可以使用稱為 物件的物件， AWS IoT 將裝置連線至 。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)。如需有關 AWS IoT 實物的其他資訊，請參閱開發人員指南使用 管理裝置 AWS IoT。

管理 AWS IoT 物件

Toolkit for VS 程式碼有多項功能，可讓您的 AWS IoT 物件管理更有效率。以下是您可以使用 VS Code 工具組來管理 AWS IoT 物件的方式：

- 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. 從 AWS Explorer 展開 IoT 服務標題，然後內容選擇 （按一下滑鼠右鍵） 事物。
2. 從內容功能表中選擇建立物件以開啟對話方塊。

3. 在物件名稱欄位中輸入 IoT 物件的名稱，以遵循提示。
4. 完成此操作後，物件區段中會顯示一個物件圖示，後面接著您指定的名稱。

將憑證連接到物件

1. 從 AWS Explorer 展開 IoT 服務區段。
2. 在實物子區段下，找到您要連接憑證的實物。
3. 內容選擇（按一下滑鼠右鍵）物件，然後從內容功能表選擇連接憑證，以開啟包含憑證清單的輸入選擇器。
4. 從清單中，選擇與您要連接至物件之憑證對應的憑證 ID。
5. 完成此操作後，即可在 AWS 瀏覽器中存取您的憑證，做為您連接憑證的物件項目。

從物件中分離憑證

1. 從 AWS Explorer 展開 IoT 服務區段
2. 在 Things (物件) 子區段中，找出您要分離憑證的物件。
3. Context-select（右鍵）物件，然後選擇從 context-menu 分離憑證。
4. 完成後，分離的憑證將不再顯示在 AWS Explorer 中的該物件下，但仍可從憑證子區段存取。

刪除物件

1. 從 AWS Explorer 展開 IoT 服務區段。
2. 在實物子區段中，找到您要刪除的實物。
3. Context-select（右鍵）物件，然後從內容功能表選擇刪除物件來刪除它。
4. 完成後，刪除的物件將無法再從 Things 子區段中使用。

> **Note**
>
> 注意：您只能刪除未連接憑證的物件。

AWS IoT 憑證

憑證是在您的 AWS IoT 服務和裝置之間建立安全連線的常見方式。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。如需 AWS IoT 憑證的其他資訊，請參閱開發人員指南身分驗證 (IoT)。

管理憑證

VS Code 工具組提供多種方式，可讓您直接從 AWS Explorer 管理 AWS IoT 憑證。

- Create a certificate
- Change a certificate status
- Attach a policy to a certificate
- Delete a certificate

建立 AWS IoT 憑證

X.509 憑證可用來與 的執行個體連線 AWS IoT。

1. 從 AWS Explorer 中，展開 IoT 服務區段，然後內容選擇 （按一下滑鼠右鍵） 憑證。
2. 從內容功能表選擇建立憑證以開啟對話方塊。
3. 在本機檔案系統中選取目錄，以儲存 RSA 金鑰對和 X.509 憑證。

> **Note**
>
> - 預設檔案名稱會包含憑證 ID 做為字首。
> - 只有 X.509 憑證會透過 AWS IoT 服務儲存在 AWS 您的帳戶中。
> - 您的 RSA 金鑰對只能核發一次，請在出現提示時將其儲存到檔案系統的安全位置。
> - 如果憑證或金鑰對目前無法儲存至您的檔案系統，則 AWS Toolkit 會從 AWS 您的帳戶刪除憑證。

修改憑證狀態

個別憑證的狀態會顯示在 AWS Explorer 中的 ID 旁，並可設定為：作用中、非作用中或已撤銷。

> **Note**
>
> - 您的憑證需要有作用中狀態，才能使用它將裝置連接到您的 AWS IoT 服務。
> - 非作用中的憑證可以啟用，無論是先前已停用，還是預設為非作用中。
> - 已撤銷的憑證則無法重新啟用。

1. 從 AWS Explorer 中，展開 IoT 服務區段。
2. 在憑證子區段中，找到您要修改的憑證。
3. Context-select （按一下滑鼠右鍵） 憑證以開啟內容功能表，顯示該憑證可用的狀態變更選項。

- 若憑證狀態為 inactive (非作用中)，請選擇 activate (啟用)，將狀態變更為 active (作用中)。
- 若憑證狀態為 active (作用中)，請選擇 deactivate (停用)，將狀態變更為 inactive非作用中 (作用中)。
- 若憑證狀態為 active (作用中) 或 inactive (非作用中)，請選擇 revoke (撤銷)，將狀態變更為 revoked (已撤銷)。

> **Note**
>
> 如果您在物件子區段中顯示時選取連接至物件的憑證，則這些狀態變更動作也可以使用。

將 IoT 政策連接至憑證

1. 從 AWS Explorer 展開 IoT 服務區段。
2. 在憑證子區段中，找到您要修改的憑證。
3. Context-select （按一下滑鼠右鍵） 憑證，然後從內容功能表中選擇連接政策，以開啟包含可用政策清單的輸入選擇器。
4. 選擇您要連接到憑證的政策。
5. 完成此操作後，您選取的政策會新增至憑證做為子選單項目。

將 IoT 政策從憑證分離

1. 從 AWS Explorer 展開 IoT 服務區段。
2. 在憑證子區段中，找到您要修改的憑證。
3. 展開憑證並找到您要分離的政策。
4. Context-select （右鍵） 政策，然後選擇從內容功能表分離。
5. 完成後，政策將不再是可從憑證存取的項目，但可從政策子區段取得。

刪除憑證

1. 從 AWS Explorer 展開 IoT 服務標題。
2. 在憑證子區段中，找到您要刪除的憑證。
3. Context-select （右鍵） 憑證，然後從內容功能表中選擇刪除憑證。

> **Note**
>
> 若憑證已連接至物件或處於作用中狀態，則無法刪除。您可以刪除已連接政策的憑證。

AWS IoT 政策

AWS IoT 核心政策是透過 JSON 文件定義，每個文件都包含一或多個政策陳述式。政策定義 AWS IoT AWS和您的裝置可以如何互相互動。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**
>
> 具名政策是版本化的，因此您可以轉返它們。在 AWS Explorer 中，您的 IoT 政策列在 IoT 服務的政策子區段下。您可以展開政策來檢視政策版本。預設版本以星號表示。

管理政策

Toolkit for VS Code 提供多種方式，可讓您管理 AWS IoT 服務政策。以下是您可以直接從 VS 程式碼中的 AWS Explorer 管理或修改政策的方式：

- Create a policy
- Upload a new policy version
- Edit a policy version
- Change the policy version defualt
- Change the policy version defualt

建立 AWS IoT 政策

> **Note**
>
> 您可以從 AWS Explorer 建立新政策，但定義政策的 JSON 文件必須已存在於您的檔案系統中。

1. 從 AWS Explorer 中，展開 IoT 服務區段。
2. Context-select （右鍵） 政策子區段，然後選擇從文件建立政策，以開啟政策名稱輸入欄位。
3. 輸入名稱並依照提示開啟對話方塊，從檔案系統中選取 JSON 文件。
4. 選擇包含您的政策定義的 JSON 檔案，政策將在完成時在 AWS 瀏覽器中提供。

上傳新的 AWS IoT 政策版本

將 JSON 文件上傳至政策，即可建立新的政策版本。

> **Note**
>
> 新的 JSON 文件必須存在於您的檔案系統上，才能使用 AWS Explorer 建立新版本。

1. 從 AWS Explorer 中，展開 IoT 服務區段。
2. 展開政策子區段以檢視您的 AWS IoT 政策
3. Context-select （右鍵） 您要更新的政策，然後選擇從文件建立新版本。
4. 對話方塊開啟時，請選擇包含政策定義更新的 JSON 檔案。

5. 新版本可從 Explorer 中的政策存取 AWS 。

編輯 AWS IoT 政策版本

您可以使用 VS 程式碼來開啟和編輯政策文件。完成文件編輯後，您可以將文件儲存到您的檔案系統。然後，您可以從 AWS Explorer 將其上傳至您的 AWS IoT 服務。

1. 從 AWS Explorer 展開 IoT 服務區段。
2. 展開政策子區段，並找到您要更新的政策。從文件中建立政策以開啟政策名稱輸入欄位。
3. 展開您要更新的政策，然後選取您要編輯的政策版本內容 （按一下滑鼠右鍵）。
4. 從內容選單中選擇檢視，以在 VS 程式碼中開啟政策版本
5. 開啟政策文件時，請進行並儲存您想要的變更。

> **Note**
>
> 此時，您對政策所做的變更只會儲存至本機檔案系統。若要更新版本並使用 AWS Explorer 追蹤版本，請重複Upload a new policy version此程序中所述的步驟。

選取新的政策版本預設值

1. 從 AWS Explorer 中，展開 IoT 服務區段。
2. 展開 Policies (政策) 子區段，找出您要更新的政策。
3. 展開您要更新的政策，然後選取您要設定的政策版本內容 （按一下滑鼠右鍵），然後選擇設定為預設。
4. 完成後，您選擇的新預設版本旁邊會有一個星號。

刪除 政策

> **Note**
>
> 您必須先符合條件，才能刪除政策或政策版本。
>
> - 如果政策已連接至憑證，則無法刪除該政策。

- 如果政策具有任何非預設版本，則無法刪除該政策。
- 除非選取新的預設版本，或整個政策已刪除，否則您無法刪除政策的預設版本。
- 您必須先刪除該政策的所有非預設版本，才能刪除整個政策。

1. 從 AWS Explorer 中，展開 IoT 服務區段。
2. 展開 Policies (政策) 子區段，找出您要更新的政策。
3. 展開您要更新的政策，然後選取要刪除的政策版本（按一下滑鼠右鍵），然後選擇刪除。
4. 刪除版本時，將無法再從 Explorer 中看到該版本。
5. 當政策唯一剩餘的版本為預設值時，您可以選取（按一下滑鼠右鍵）父政策，然後選擇刪除來刪除它。

AWS Lambda 函數

AWS Toolkit for Visual Studio Code 提供 AWS Lambda 函數的全方位支援，可讓您直接從 VS Code 建置、測試和部署 。

Lambda 是一種全受管、事件驅動的運算服務，可自動執行您的程式碼，以回應來自 200 多個 AWS 服務和software-as-a-service(SaaS) 應用程式的事件。如需 AWS Lambda 服務的詳細資訊，請參閱 AWS Lambda開發人員指南。

下列主題說明如何在 AWS Lambda 中使用 AWS Toolkit for Visual Studio Code。

主題

- 使用 AWS Lambda 函數
- AWS Lambda console 至 IDE
- AWS Lambda 透過 LocalStack 支援
- AWS Lambda 遠端偵錯

使用 AWS Lambda 函數

AWS Toolkit for Visual Studio Code 可讓您在本機 VS Code 環境中使用 AWS Lambda 函數。使用 AWS Toolkit，您可以建立、編輯、測試、偵錯和部署 Lambda 函數，而無需離開 IDE。如需 AWS Lambda 服務的詳細資訊，請參閱 AWS Lambda開發人員指南。

下列各節說明如何開始使用 中的 Lambda 函數 AWS Toolkit for Visual Studio Code。

> **Note**
>
> 如果您已經使用 建立 Lambda 函數 AWS 管理主控台，則可以從 Toolkit 叫用它們。此外，您可以從 將 Lambda 函數開啟為 VS 程式碼 AWS Lambda console，如需其他資訊，請參閱本使用者指南中的 AWS Lambda console 至 IDE主題。若要在 VS 程式碼中建立新的 Lambda 函數，請遵循本使用者指南中建立新的無伺服器應用程式（本機）主題中概述的步驟。

先決條件

必須符合下列條件，才能在 AWS Toolkit 中使用 AWS Lambda 服務。

- 已安裝最新版本的 AWS Toolkit for Visual Studio Code ，並使用 AWS 登入資料進行設定。
- 您的 AWS Identity and Access Management (IAM) 受管許可和政策已設定為使用 AWS Lambda 服務。如需如何設定許可和建立相容 AWS 受管政策的詳細資訊，請參閱《 AWS Lambda 開發人員指南》中的 AWS Identity and Access ManagementAWS Lambda主題的 。
- 您有現有的 AWS Lambda 函數，或熟悉如何建立一個函數。如需如何建立 Lambda 函數的指示，請參閱《 AWS Lambda 開發人員指南》中的建立您的第一個 Lambda 函數主題。

叫用 Lambda 函數

若要從 AWS 您的帳戶叫用 Lambda 函數到 VS 程式碼，請完成下列步驟。

1. 從 AWS Toolkit for Visual Studio Code中，展開 AWS 瀏覽器。
2. 從 AWS 瀏覽器展開 Lambda 以檢視您的 Lambda 資源。
3. 開啟您要叫用之 Lambda 函數的內容選單（按一下滑鼠右鍵），然後選擇在雲端叫用，或在雲端圖示中選擇叫用，以在 VS 程式碼中開啟遠端叫用組態選單。
4. 從遠端調用組態功能表中，指定您的承載設定，並新增事件所需的任何其他資訊。

   > **Note**
   >
   > 當您在 AWS 瀏覽器中選擇在雲端叫用時，第一個叫用程序可能會立即開始執行。輸出會顯示在 VS 程式碼終端機的 OUTPUT 索引標籤中。

5. 選擇遠端調用按鈕來調用您的函數，輸出會顯示在 VS 程式碼終端機的輸出索引標籤中。

刪除 Lambda 函式

若要刪除 Lambda 函數，請完成下列程序。

> **Warning**
>
> 請勿使用此程序刪除與 CloudFormation 相關聯的 Lambda 函數。這些函數必須透過您的 CloudFormation 堆疊刪除。

1. 從 AWS Toolkit for Visual Studio Code中，展開 AWS 瀏覽器。
2. 從 AWS 瀏覽器展開 Lambda 以檢視您的 Lambda 資源。
3. 在要刪除的 Lambda 函數上按一下滑鼠右鍵，然後選擇刪除。
4. 出現提示時，請確認您想要刪除函數。

刪除函數之後，它不會再列在 AWS 瀏覽器中。

下載 Lambda 函式

您可以從遠端 Lambda 函數下載程式碼到 VS 程式碼工作區，以進行編輯和偵錯。

> **Note**
>
> 若要下載 Lambda 函數，您必須在具有可存取資料夾的 VS 程式碼工作區中工作，且 AWS Toolkit 僅支援使用 Node.js 和 Python 執行時間搭配 Lambda 函數的此功能。

1. 從 AWS Toolkit for Visual Studio Code中，展開 AWS 瀏覽器。
2. 從 AWS 瀏覽器展開 Lambda 以檢視您的 Lambda 資源。
3. 在您要下載的 Lambda 函數上按一下滑鼠右鍵，然後選擇下載。
4. 您的 Lambda 函數會在 VS 程式碼編輯器中開啟，並在下載完成時顯示在 AWS 瀏覽器中。 AWS Toolkit 也會在 VS Code 執行面板中建立啟動組態，可讓您使用 在本機執行和偵錯 Lambda 函數 AWS Serverless Application Model。如需使用 的詳細資訊 AWS SAM，請參閱 the section called “從範本執行和偵錯無伺服器應用程式 （本機）”。

部署新 Lambda 函數的更新

您可以從本機機器上未指定的暫時位置部署更新至新的 Lambda 函數。

> **Note**
>
> 當您的 lambda 檔案有未部署的變更時，您會收到位於 VS 程式碼編輯器和 AWS 瀏覽器中修改檔案旁邊的 M 圖示的通知。

從 VS 程式碼編輯器部署

1. 在 VS 程式碼編輯器中從 Lambda 函數開啟檔案，然後變更檔案。
2. 從 VS 程式碼主功能表手動儲存或按 **`option+s`**(Mac) **`ctrl+s`** (Windows)。
3. VS Code 會自動提示您將變更部署到雲端，選擇部署按鈕以確認部署。
4. VS Code 會更新您部署的狀態，並在程序完成時通知您。

從 AWS Explorer 部署

1. 在 VS 程式碼編輯器中從 Lambda 函數開啟檔案，然後變更檔案。
2. 從 AWS Toolkit 中，展開 AWS 瀏覽器。
3. 從 AWS 瀏覽器中，使用您要部署變更的 Lambda 函數展開 AWS 區域。
4. 從 AWS 區域展開 Lambda 並導覽您要部署變更的函數。
5. 從函數旁的快速功能表中，選擇儲存並部署程式碼圖示。
6. VS Code 會更新您部署的狀態，並在程序完成時通知您。

上傳現有 Lambda 函數的更新

下列程序說明如何上傳對現有 Lambda 函數所做的本機變更。此功能支援使用任何 Lambda 支援的執行時間上傳。

> **Warning**
>
> 上傳 lambda 函數之前，請注意下列事項：
>
> - 以這種方式更新程式碼不會使用 AWS SAM CLI 進行部署或建立 CloudFormation 堆疊

- AWS Toolkit 不會驗證程式碼。在上傳任何變更至雲端之前，驗證您的程式碼並測試您的函數 (s)。

上傳 Zip 封存檔

1. 從 AWS Toolkit for Visual Studio Code中，展開 AWS 瀏覽器。
2. 從 AWS 瀏覽器展開 Lambda 以檢視您的 Lambda 資源。
3. 在您要上傳變更的 Lambda 函數上按一下滑鼠右鍵，然後選擇上傳 Lambda... 以開啟選取上傳類型功能表。
4. 選擇 ZIP Archive 以在`ZIP Archive`本機目錄中尋找 。
5. 出現提示時，請確認上傳以開始上傳選取的 `ZIP Archive`。
6. 您的上傳狀態會顯示在 VS 程式碼中，上傳程序完成時會通知您。

上傳目錄而不建置

1. 從 AWS Toolkit for Visual Studio Code中，展開 AWS 瀏覽器。
2. 從 AWS 瀏覽器展開 Lambda 以檢視您的 Lambda 資源。
3. 在您要上傳變更的 Lambda 函數上按一下滑鼠右鍵，然後選擇上傳 Lambda... 以開啟選取上傳類型功能表。
4. 選擇目錄以繼續到建置目錄畫面。
5. 從建置目錄畫面中，選擇否以選擇要上傳的本機目錄。
6. 出現提示時，請確認上傳以上傳選取的目錄。
7. 您的上傳狀態會顯示在 VS 程式碼中，上傳程序完成時會通知您。

使用組建上傳目錄

> **Note**
>
> 請注意以下事項：
>
> - 此程序需要 AWS Serverless Application Model CLI。
> - Toolkit 會在上傳之前， AWS 通知您無法偵測到相符的處理常式。

- 若要變更連接至 Lambda 函數的處理常式，請使用 AWS Lambda console 或 AWS Command Line Interface。

1. 從 AWS Toolkit for Visual Studio Code中，展開 AWS 瀏覽器。
2. 從 AWS 瀏覽器展開 Lambda 以檢視您的 Lambda 資源。
3. 在您要上傳變更的 Lambda 函數上按一下滑鼠右鍵，然後選擇上傳 Lambda... 以開啟選取上傳類型功能表。
4. 選擇目錄以繼續到建置目錄畫面。
5. 從建置目錄畫面中，選擇是，然後選擇要上傳的本機目錄。
6. 出現提示時，請確認上傳以開始建置和上傳選取的目錄。
7. 您的上傳狀態會顯示在 VS 程式碼中，上傳程序完成時會通知您。

將您的 Lambda 函數轉換為 AWS SAM 專案

若要將您的 Lambda 函數轉換為 AWS SAM 堆疊，請完成下列步驟。

> **Warning**
>
> 目前，將 Lambda 函數轉換為 AWS SAM 專案時，僅支援一部分的資源。若要在轉換後尋找任何遺失的資源，請檢查 Lambda 主控台，並手動將其新增至您的 AWS SAM 範本。如需支援和不支援資源的其他詳細資訊，請參閱《 AWS CloudFormation 開發人員指南》中的資源類型支援主題。

1. 從 AWS Toolkit 中，展開 AWS 瀏覽器。
2. 從 AWS 瀏覽器中，使用您要轉換為 AWS SAM 專案的 Lambda 函數展開 AWS 區域。
3. 從 AWS 區域中，展開 Lambda 並導覽您要轉換為 AWS SAM 堆疊的函數。
4. 從 Lambda 函數旁的快速選單中，選擇轉換為 SAM 應用程式圖示來瀏覽本機檔案系統，並指定新 AWS SAM 專案的位置。
5. 指定位置後， AWS Toolkit 會開始將您的 Lambda 函數轉換為 AWS SAM 專案，VS Code 會提供程序狀態的更新。

> **Note**
>
> 此程序可能需要幾分鐘的時間。

6. 當 VS 程式碼提示時，輸入堆疊名稱，然後按 **Enter** 鍵繼續。
7. VS Code 會繼續更新您專案的狀態，然後在程序完成時通知您，並以 VS Code 工作區的形式開啟您的新 AWS SAM 專案。

AWS Lambda console 至 IDE

AWS Lambda console 至 IDE 功能可讓您將 AWS Lambda 函數從 下載 AWS Lambda console 至 VS 程式碼。在 VS Code 中使用 Lambda 函數可讓您存取其他本機開發選項，例如 AWS Serverless Application Model (AWS SAM) 和 AWS Cloud Development Kit (AWS CDK)。

如需 的詳細資訊 AWS Lambda，請參閱 AWS Lambda開發人員指南。若要開始使用 AWS Toolkit 中的 Lambda 函數，請參閱本使用者指南中的使用 AWS Lambda 函數主題。下列各節說明如何將工作流程從 Lambda 主控台移至 VS 程式碼。如需將 Lambda 函數從 Lambda 主控台移至 VS 程式碼的詳細資訊，包括如何開始使用 Lambda 主控台，請參閱《 開發人員指南》中的使用 VS 程式碼在本機開發 Lambda 函數主題。 AWS Lambda

從主控台遷移至本機開發

若要從 VS Code 中的 Lambda 主控台開啟 Lambda 函數，請完成下列步驟：

1. 從 Web 瀏覽器開啟 Lambda 主控台。
2. 從 Lambda 主控台中，選擇要在 VS 程式碼中開啟的函數。
3. 從函數檢視中，導覽至程式碼來源索引標籤。
4. 從程式碼來源索引標籤中，選擇在 VS 程式碼中開啟。

在 VS 程式碼中使用 Lambda 函數

當您的 Lambda 函數透過 Lambda 主控台在 VS 程式碼中開啟時：

- VS Code 會自動在您的本機機器上啟動。
- 您的 Lambda 函數會開啟為 VS 程式碼工作區。
- 您的 Lambda 會在 VS 程式碼編輯器中`handler file`開啟。

> **Note**
>
> 如果工作區`handler file`中沒有正確設定 ，則 VS 程式碼編輯器中不會開啟任何檔案。

透過 Lambda 主控台在 VS 程式碼中開啟 Lambda 函數可讓您存取所有現有的 AWS Toolkit Lambda 功能，包括使用完整語言支援編輯函數程式碼的能力、本機測試、遠端偵錯、部署支援和相依性管理。如需 AWS Toolkit 中支援之 Lambda 功能的詳細資訊，請參閱本使用者指南中的 AWS Lambda服務目錄。

AWS Lambda 透過 LocalStack 支援

在 中使用 LocalStack 支援建置、測試和偵錯無伺服器應用程式 AWS Toolkit for Visual Studio Code。LocalStack 是一種 AWS 雲端模擬器，允許對無伺服器應用程式進行本機測試。

如需 的詳細資訊 AWS Lambda，請參閱 AWS Lambda開發人員指南。若要進一步了解 LocalStack，請造訪其網站 LocalStack。

先決條件

以下是在 VS 程式碼中使用 LocalStack 的先決條件。

> **Note**
>
> LocalStack CLI 是在設定程序期間安裝，但如果您偏好不同版本的 LocalStack CLI，則所需的最低版本為 4.8.0。

- 需要 LocalStack Web 應用程式帳戶才能存取免費和付費 LocalStack 方案可用的所有功能。LocalStack 社群版本可在沒有 帳戶的情況下使用。
- 在 VS 程式碼中使用 LocalStack 需要 Docker。如需 Docker LocalStack 需求的詳細資訊，請參閱 LocalStack 文件中的 LocalStack Docker 映像主題。
- 建議：The AWS Command Line Interface (AWS CLI) 可協助您在模擬雲端環境中使用 服務。

安裝 LocalStack

若要安裝 LocalStack 免費和付費分層版本，請完成下列步驟。

> **Note**
>
> 如需如何設定 LocalStack Community Edition 的說明，請參閱本主題之設定 LocalStack 一節中的 LocalStack Community 內容。 LocalStack

1. 從 AWS Toolkit 中，展開 APPLICATION BUILDER Explorer。
2. 選擇開啟逐步解說按鈕，在 VS 程式碼編輯器中開啟開始建置應用程式逐步解說索引標籤。
3. 從逐步解說中，選擇安裝 LocalStack 以啟動 VS Code 中的 LocalStack 安裝程序。

設定 LocalStack

安裝 VS Code 的 LocalStack 擴充功能後，您可能會在需要設定時看到下列其中一個指標：

- 在預設位於 IDE 左下角的 VS 程式碼狀態列中，LocalStack 狀態為紅色。
- VS 程式碼會提示您設定 LocalStack。

LocalStack 的設定和組態有兩種類型，取決於您使用的 LocalStack 版本。下列標籤區段說明每個 LocalStack 設定程序。

> **Note**
>
> LocalStack 免費和付費方案版本需要 LocalStack 驗證權杖。如需 LocalStack 定價的特定資訊，請參閱其選擇您的計劃定價指南。

LocalStack 免費和付費方案

設定 LocalStack 的方式有兩種。

- 從 VS 程式碼設定 LocalStack 開始提示，選擇設定按鈕。
- 從 VS 程式碼狀態列中，選擇 LocalStack 狀態圖示以開啟設定 LocalStack 以開始使用提示，然後選擇設定按鈕。

在設定期間，系統會執行下列步驟：

1. 安裝 LocalStack CLI。

2. 檢查您是否擁有 LocalStack 帳戶。
3. 如果您有 LocalStack 帳戶，系統會引導您完成預設 Web 瀏覽器中的身分驗證程序。同樣地，如果您沒有 LocalStack 帳戶，系統會在身分驗證程序之前引導您完成帳戶設定。

設定 LocalStack 後，VS Code 狀態列中的 LocalStack 狀態會更新。

> **Note**
>
> 如果您尚未建立 LocalStack 的 AWS 設定檔，系統會自動為您建立新的設定檔，做為 LocalStack 設定程序的一部分。

LocalStack 社群

LocalStack 的 Community Edition 是免費的，不需要您註冊 帳戶，它從不需要授權的 Docker 映像執行。如需 LocalStack Community Edition 的其他詳細資訊，請參閱 LocalStack Community 映像文件。下列各節說明在 VS Code 中使用 LocalStack 社群版本所需的先決條件和基本設定。

啟動新的執行個體

若要啟動 LocalStack Community 的新執行個體，請完成下列程序。

> **Note**
>
> 下列範例會在連接埠 4566 上啟動 LocalStack 的容器執行個體。如果您指定不同的連接埠值，則必須更新設定 AWS CLI 和 AWS Toolkit 區段中程序中指定的連接埠值。

1. 從 VS 程式碼中，按 開啟 VS 程式碼終端機**ctrl + `(backtick)**。
2. 在終端機中輸入以下內容。

   Mac：

   ```
   docker run -d --name localstack_main \
   >> -p 4566:4566 \
   >> -v /var/run/docker.sock:/var/run/docker.sock \
   >> localstack/localstack
   ```

   Windows：

```
docker run -d --name localstack_main `
>> -p 4566:4566 `
>> -v /var/run/docker.sock:/var/run/docker.sock `
>> localstack/localstack
```

3. 程序完成時，終端機會更新 Docker 執行個體的狀態。

LocalStack 的這個容器化執行個體可讓您存取您在下載程序期間指定的 AWS 服務。

設定 LocalStack 和 Docker 的 CLI。

若要設定 AWS CLI 和 AWS Toolkit 在 Docker 中使用 LocalStack，請完成下列步驟來設定新的設定檔：

1. 從 VS 程式碼中，按 開啟 VS 程式碼終端機**ctrl + `(backtick)**。
2. 在終端機中輸入以下內容。

```
~/.aws/credentials
[localstack]
aws_access_key_id = test
aws_secret_access_key = test
~/.aws/config
[profile localstack]
region = us-east-1
output = json
endpoint_url = http://localhost:4566 [default localstack endpoint]
```

3. Toolkit 會 AWS 偵測 LocalStack 設定檔並更新連線狀態選單。

設定後，從狀態列的設定檔區段中選擇 LocalStack AWS 設定檔，即可在 AWS 瀏覽器中顯示 LocalStack 資源。此外，您可以在 VS 程式碼終端機的輸出索引標籤中檢視 LocalStack 日誌。

在 VS 程式碼中啟動 LocalStack

您可以使用下列任一方法啟動 LocalStack：

從 VS 程式碼狀態列啟動 LocalStack

1. 從 VS 程式碼導覽至狀態列，然後選擇啟動 LocalStack 按鈕以啟動 LocalStack。
2. LocalStack 成功啟動時，VS 程式碼狀態列會更新。

從 VS Code Command Palette 啟動 LocalStack

1. 從 VS 程式碼中，按 **Cmd + Shift + P**(Mac) 或 **Control + Shift + P**(Windows) 開啟命令調色盤。
2. 從命令面板中，**Start LocalStack**在搜尋列中輸入 ，並在填入結果時從清單中選擇它。
3. LocalStack 成功啟動時，VS 程式碼狀態列會更新。

從 VS 程式碼終端機啟動 LocalStack

1. 從 VS 程式碼中，按 開啟 VS 程式碼終端機**ctrl + `(backtick)**。
2. 從 VS 程式碼終端機，輸入 **localstack start** CLI 命令。
3. LocalStack 成功啟動時，VS 程式碼狀態列會更新。

建置範例無伺服器應用程式

若要開始在 VS Code 中使用 LocalStack，您需要範例無伺服器應用程式。如果您的 AWS 帳戶中已有現有的應用程式，您可以使用 LocalStack 將其部署到本機，或使用 AWS Serverless Land 建立新的應用程式。

如需有關在 Toolkit AWS 中使用無伺服器土地建立應用程式的詳細資訊，請參閱《 使用者指南》中的使用無 AWS 伺服器土地主題。如需 Serverless Land 的詳細資訊，請參閱 Serverless Land Web 應用程式主要登陸頁面。

使用 LocalStack 測試和偵錯 Lambda 函數

在 LocalStack VS Code 延伸模組中測試和偵錯 Lambda 函數類似於使用部署到 AWS 雲端的函數。主要差別是您的 AWS Toolkit 執行個體必須使用 LocalStack 帳戶進行身分驗證，才能使用 LocalStack 部署和偵錯函數。

> **Note**
>
> 本節所述的測試和偵錯功能不適用於 LocalStack Community Edition。
> 若要在 VS 程式碼中使用 LocalStack，請連線至 Toolkit 中的 LocalStack AWS 設定檔。當您的 LocalStack 設定檔處於作用中狀態時，VS 程式碼狀態列會顯示 AWS： profile：localstack（自訂端點）和核取記號。

如需有關在 AWS Toolkit 中使用 Lambda 函數的詳細資訊，請參閱本使用者指南中的使用 AWS Lambda 函數主題。

AWS Lambda 遠端偵錯

AWS Toolkit for Visual Studio Code 可讓您直接在 VS Code 中偵錯在雲端執行的 AWS Lambda 函數。透過 AWS Lambda 遠端偵錯，您可以檢查執行中的函數、設定中斷點、檢查變數和逐步偵錯，而無需修改其現有的開發工作流程。

下列各節說明如何在 中使用 Lambda 遠端偵錯 AWS Toolkit for Visual Studio Code。

Lambda 遠端偵錯的運作方式

Toolkit 會使用額外的 Lambda 偵錯層暫時修改 Lambda 函數，並將 Lambda 調用逾時限制延長至 900 秒，以 AWS 啟用遠端偵錯。使用 Secure Tunneling 在本機偵錯工具與 Lambda 執行期環境 AWS IoT 之間建立安全連線。此連線可讓您在遠端執行時，使用本機程式碼中斷點逐步完成函數。偵錯工作階段完成後，所有暫時修改都會自動還原為其原始設定。

開始使用

支援的執行時期

Lambda 遠端偵錯支援下列執行時間。

- Python (Amazon Linux 2023)
- Java
- Typescript/JavaScript/Node.js (Amazon Linux 2023)

> **Note**
>
> Lambda 遠端偵錯不支援 Lambda 受管執行個體和 OCI 映像函數類型。

先決條件

在開始之前，必須符合下列先決條件。

- 您必須在 AWS Toolkit 中設定有效的 AWS 登入資料。如需安裝 AWS Toolkit 和設定登入資料的其他詳細資訊，請參閱本使用者指南中的入門主題。

- Lambda 函數已部署至 AWS 您的帳戶。如需部署 Lambda 函數的詳細資訊，請參閱《 AWS Lambda開發人員指南》中的建立您的第一個 Lambda 函數主題。
- 您必須擁有適當的 AWS Identity and Access Management (IAM) 政策和許可，才能偵錯函數。如需 Lambda 許可的其他詳細資訊，請參閱《 AWS Lambda開發人員指南》中的主題AWS 的 受管政策 AWS Lambda。以下是政策的範例，其中包含在 Toolkit 中使用 Lambda AWS 遠端偵錯所需的最低許可。

> **Note**
>
> 透過 AWS AWS IoT 安全通道啟用遠端偵錯。這可讓您的本機偵錯工具建立與 Lambda 執行期環境的安全連線。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lambda:ListFunctions",
        "lambda:GetFunction",
        "lambda:GetFunctionConfiguration",
        "lambda:GetLayerVersion",
        "lambda:UpdateFunctionConfiguration",
        "lambda:InvokeFunction",
        "lambda:PublishVersion",
        "lambda:DeleteFunction",
        "iot:OpenTunnel",
        "iot:RotateTunnelAccessToken",
        "iot:ListTunnels"
      ],
      "Resource": "*"
    }
  ]
}
```

存取 Lambda 遠端偵錯

有兩種主要路徑可存取 AWS Toolkit 中的 Lambda 遠端偵錯： AWS Explorer 或 Application Builder Explorer。從 AWS 瀏覽器，您可以透過節點 AWS Lambda 存取 Lambda 遠端偵錯。從 Application Builder Explorer，您可以透過本機 AWS SAM 專案存取 Lambda 遠端偵錯。

從 AWS 瀏覽器存取 Lambda 遠端偵錯

1. 從 VS 程式碼中，開啟 AWS Toolkit 延伸模組。
2. 從 AWS Toolkit 中展開 AWS 瀏覽器。
3. 從瀏覽器展開 Lambda 節點。
4. 導覽至您要偵錯的函數，然後從內容功能表中選擇遠端叫用圖示，以開啟遠端叫用組態畫面。

從 Application Builder Explorer 存取 Lambda 遠端偵錯。

1. 從 VS 程式碼中，開啟 AWS Toolkit 延伸模組。
2. 從 AWS Toolkit 中，展開應用程式建置器瀏覽器。
3. 從瀏覽器展開包含您要偵錯之 Lambda AWS SAM專案的專案。
4. 展開您要偵錯的已部署Lambda函數。
5. 導覽至函數遠端，然後從內容功能表中選擇遠端叫用圖示，以開啟遠端叫用組態畫面。

使用 Lambda 遠端偵錯

下列各節說明如何在 中使用 Lambda 遠端偵錯 AWS Toolkit for Visual Studio Code。

> **Note**
>
> Lambda 函數有 5 層限制，函數程式碼和所有連接層的合併限制為 250MB。Lambda 遠端偵錯需要至少 1 個可用層才能執行。

設定偵錯工作階段

開始之前，請先完成下列程序來設定偵錯工作階段。

1. 完成從 AWS 瀏覽器存取 Lambda 遠端偵錯，或從上一節的應用程式建置器瀏覽器程序存取 Lambda 遠端偵錯，以開啟遠端調用組態功能表。

2. 從遠端調用組態功能表中，選取遠端偵錯核取方塊以顯示遠端偵錯屬性。
3. 指定本機處理常式檔案的本機根路徑。

> **Note**
>
> 本機根路徑是您原始碼的位置，其符合已部署的 Lambda 函數。如果您使用 Application Builder Explorer 中部署的函數，則會自動偵測您的本機根路徑。
> 如果您沒有儲存在本機的原始程式碼，請選擇下載遠端程式碼按鈕來擷取 Lambda 函數原始程式碼。這將在 VS 程式碼編輯器`handler file`中開啟您的 。

4. 從承載區段中，指定取得測試事件資料的位置。

設定中斷點和偵錯

完成下列程序，設定中斷點並開始偵錯。

1. 在 VS 程式碼編輯器`handler file`的 中，按一下裝訂邊，在您要暫停偵錯的行號上設定中斷點。
2. 當您對中斷點感到滿意時，請返回遠端調用組態功能表，確認您的設定已正確設定，然後選擇遠端調用按鈕以開始偵錯。
3. Toolkit 會使用偵錯功能 AWS 更新您的 Lambda 函數、為偵錯工作階段建立安全通道、使用指定的承載叫用函數，然後在到達中斷點時暫停程序。
4. 在中斷點暫停時，使用 RUN AND DEBUG 窗格來檢視您的 VARIABLES、CALL STACK 和 BREAKPOINTS。

更新和測試您的函數

若要使用快速部署修改程式碼和測試變更，請完成下列程序。

1. 啟用偵錯工作階段後，請在 VS 程式碼編輯器`handler file`中變更您的 。
2. 儲存您的變更 (**`Command+S on macOS`**、**`Ctrl+S on Windows`**)
3. 出現提示時，請確認您希望繼續部署變更。Toolkit AWS 將使用修改後的程式碼更新您的 Lambda 函數。
4. 透過設定新的中斷點並再次選取遠端調用按鈕，繼續偵錯和測試您的變更。

> **Note**
>
> 或者，您可以在 VS 程式碼偵錯控制項中取消選取連接偵錯工具選項，然後選擇遠端調用按鈕來執行函數，而無需偵錯。

結束偵錯工作階段

下列每個選項都會結束遠端偵錯工作階段，並從專案中移除偵錯層。

- 從遠端調用組態畫面選擇移除偵錯設定選項。
- 從 VS 程式碼偵錯控制項中選擇中斷連線圖示。
- 在 VS 程式碼編輯器`handler file`中關閉 。

> **Note**
>
> 謹記下列事項：
>
> - Lambda 除錯層會在閒置 60 秒後自動移除。計數會在您上次調用完成時開始。
> - 如果您在偵錯過程中對infrastructure-as-code(IaC) 受管 (AWS SAM AWS CDK、Terraform) 函數進行程式碼變更，請將它們儲存至本機專案，並考慮更新您的來源控制儲存庫。重新部署 IaC 函數時，會覆寫未儲存的變更。
> - 如果您僅為偵錯目的進行暫時性變更，建議您從來源控制重新部署函數，以確保其符合您的生產程式碼。

使用來源映射偵錯 TypeScript Lambda 函數

下列各節說明如何使用來源映射偵錯 TypeScript Lambda 函數。

先決條件

若要偵錯 TypeScript Lambda 函數，必須符合下列先決條件。

- 您必須在啟用來源映射選項的情況下編譯 TypeScript。如需詳細資訊，請參閱 VS 程式碼文件中的 JavaScript 來源映射支援主題。
- 不支援內嵌來源映射。您必須使用個別`.js.map`的檔案來存放來源映射。

組態

若要在 AWS Toolkit 中為 TypeScript Lambda 函數設定 Lambda 遠端偵錯，請完成下列步驟。

1. 從 AWS Toolkit 中展開 AWS 瀏覽器。
2. 從瀏覽器展開 Lambda 節點。
3. 導覽至您要為 TypeScript 設定的函數，然後從內容功能表中選擇遠端叫用圖示，以開啟遠端叫用組態畫面。
4. 選取遠端偵錯核取方塊以啟用遠端偵錯。
5. 指向包含您 的目錄，以設定您的本機根路徑`TypeScript handler file`。

   > **Note**
   >
   > 是您設定偵錯中斷點`TypeScript handler file`的位置。

6. 展開遠端偵錯其他組態設定。
7. 選取來源映射核取方塊以啟用來源映射。
8. 將輸出檔案欄位設定為 Lambda 函數複本的本機目錄。

   Example

   如果 `app.js`和 `app.map` 位於 中`.aws-sam/build/HelloWorldFunction`，請將 Out 檔案位置設為 `/Users/`**`user`**`/`**`project`**`/aws-sam/build/`**`HelloWorldFunction`**`/*`。

   > **Note**
   >
   > Out 檔案路徑應該是絕對路徑。
   > 對於 AWS SAM 和 AWS CDK 專案， AWS Toolkit 支援自動來源映射偵測。如果這些專案的輸出檔案欄位保持空白，工具組會自動嘗試偵測來源映射位置。

9. 當您對設定感到滿意時，請選擇遠端調用按鈕以開始偵錯 TypeScript 函數。

故障診斷和進階使用案例

如果您的偵錯工作階段失敗，請完成這些步驟以開始疑難排解程序。

1. 將 AWS Toolkit 更新至最新版本。
2. 關閉遠端調用組態 Web 檢視並重新開啟，以重新整理 Web 檢視。

3. 完全關閉並重新開啟 VS 程式碼，以重新啟動 VS 程式碼。
4. 開啟 VS Code Command Palette 並輸入命令 **`AWS: Reset Lambda Remote Debugging Snapshot`**，並在填入結果時選取它，以重設 Lambda 遠端偵錯快照。
5. 如果您無法對問題進行故障診斷，請將問題提交至 AWS Toolkit for Visual Studio Code GitHub 問題。

進階使用案例：程式碼簽署組態

遠端偵錯需要將偵錯層連接至 Lambda 函數。如果您的函數已啟用並強制執行程式碼簽署組態， AWS Toolkit 就無法自動將偵錯層連接至函數。

有兩種選項可解決程式碼簽署組態問題。

- 暫時移除程式碼簽署。
- 使用已簽署的偵錯層。

暫時移除程式碼簽署

透過設定 來更新程式碼簽署組態`UntrustedArtifactOnDeployment : Warn`，然後在`Enforced`偵錯程序完成後重新啟用 。

如需詳細資訊，請參閱 API 參考中的 UpdateCodeSigningConfig AWS Lambda 參考。

使用已簽章的偵錯層

1. 從 AWS Toolkit 中的 Lambda 遠端偵錯，展開遠端偵錯其他組態區段。
2. 從遠端偵錯其他組態區段中，從層覆寫欄位複製區域層 ARN。
3. 從 中 AWS CLI，使用以下命令下載 layer 版本 `aws lambda get-layer-version-by-arn --arn` **`layer-arn`**，將 layer-arn 取代為您的 layer ARN。如需如何下載已簽署偵錯層的詳細說明，請參閱《 命令參考》中的 get-layer-version-by-arn 參考。 AWS CLI
4. 使用程式碼簽署組態簽署 layer，並將其發佈至您的帳戶。如需簽署和發佈指引，請參閱《 AWS Serverless Application Model 開發人員指南》中的為您的 AWS SAM 應用程式主題設定程式碼簽署。
5. 在層簽署並發佈至您的帳戶之後，請返回 Lambda 遠端偵錯的遠端偵錯其他組態區段，然後在層覆寫欄位中輸入新的層 ARN。程序完成時，Lambda 遠端偵錯會使用您簽署的層，而不是預設層。

進階使用案例：使用 SnapStart 或佈建並行對函數進行偵錯

對於使用 SnapStart 或佈建並行設定的 Lambda 函數，發佈新版本需要更多時間。若要加速偵錯工作流程，您可以將 Lambda 遠端偵錯設定為僅更新函數的$LATEST版本，而不是發佈新版本。

1. 從遠端調用組態設定畫面中，展開遠端偵錯其他組態設定。
2. 取消選取發佈版本選項。
3. Toolkit AWS 現在只會更新函數的$LATEST版本，並使用它進行偵錯。

> **Note**
>
> 作為使用 $LATEST版本偵錯的副作用，您應該避免可能叫用 $LATEST版本的其他流量，以確保不受干擾的偵錯環境。

支援的 區域

當區域不支援遠端偵錯時，會發生下列錯誤。

```
Region ${region} doesn't support remote debugging yet
```

以下是支援的 區域清單。

- ap-east-1
- ap-northeast-1
- ap-northeast-2
- ap-south-1
- ap-southeast-1
- ap-southeast-2
- ca-central-1
- eu-central-1
- eu-north-1
- eu-west-1
- eu-west-2
- eu-west-3

- me-central-1
- me-south-1
- sa-east-1
- us-east-1
- us-east-2
- us-west-1
- us-west-2

Lambda RequestEntityTooLargeException

Lambda 函數有 5 層限制，函數程式碼和所有連接層的合併限制為 250MB。遠端偵錯層約為 40MB，如果您有大型函數套件或多個層，可能會導致函數超過此限制。如需其他詳細資訊，請參閱《 AWS Lambda 開發人員指南》中的 Lambda： InvalidParameterValueException 或 RequestEntityTooLargeException 主題一節。

下列清單說明疑難排解和修正此錯誤的方法。

- 減少函數大小：最佳化函數程式碼並移除不必要的相依性。
- 移除未使用的圖層：在偵錯期間暫時移除非必要的圖層。
- 使用外部相依性：將大型相依性移至外部儲存體，例如 Amazon S3，並在執行時間將其載入。

故障診斷 Java 偵錯

若要偵錯 Java Lambda 函數，您必須在本機安裝與 Lambda 函數執行時間版本相同的 Java 版本。

例如，偵錯 Java 25 函數時，您必須在 AWS Toolkit 正在執行的本機環境中安裝 Java 25。如果您嘗試使用本機安裝的 Java 21 或舊版對 Java 25 函數進行偵錯，遠端偵錯將無法在您設定的中斷點停止。

在開始偵錯工作階段之前，請確定您的本機 Java 版本符合您 Lambda 函數的執行時間版本。

超過 IoT 安全通道配額

以下是在 Lambda 遠端偵錯中達到 AWS IoT 安全通道連線的每日限制時，發生通道配額超過錯誤的範例。

```
Error creating/reusing tunnel: LimitExceededException: Exceeded quota of Lambda
 debugging tunnels
```

AWS IoT 安全通道連線具有下列配額：

- 自由層 IoT 安全通道每天分配 10 個連線。
- 每個通道最多支援一個 VS 程式碼執行個體 12 小時。
- 配額適用於每個 AWS 帳戶、每天。

如果您遇到 AWS IoT 安全通道錯誤，請等待每日配額重設或聯絡 AWS 支援以請求提高配額限制。如需 AWS 支援聯絡資訊，請參閱AWS 支援聯絡入口網站。如需 AWS IoT 安全通道的詳細資訊，請參閱《AWS IoT 開發人員指南》中的AWS IoT 安全通道主題。

Toolkit for VS 程式碼中的 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 Amazon Redshift 服務的詳細資訊，請參閱 Amazon Redshift 使用者指南資料表的內容。

下列主題說明如何從 使用 Amazon Redshift AWS Toolkit for Visual Studio Code。

主題

- 從 Toolkit for VS 程式碼使用 Amazon Redshift

從 Toolkit for VS 程式碼使用 Amazon Redshift

下列各節說明如何從 開始使用 Amazon Redshift AWS Toolkit for Visual Studio Code。

如需 Amazon Redshift 服務的詳細資訊，請參閱 Amazon Redshift 使用者指南主題。

開始使用

從 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 Toolkit （工具組） 連線至 AWS 您的帳戶。如需從 Toolkit 連線至您 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 Serverless 或 Amazon Redshift 佈建叢集，下列程序說明如何從 AWS 主控台使用範例資料集建立資料倉儲。

建立佈建的資料倉儲

如需建立 Amazon Redshift 佈建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，並在 Amazon Redshift 主控台上開啟 https：//https://console.aws.amazon.com/redshift/。
2. 從 Amazon Redshift 主控台中，選擇佈建叢集儀表板。
3. 從佈建叢集儀表板中，選擇建立叢集按鈕以開啟建立叢集窗格。
4. 完成叢集組態區段中的必要欄位。
5. 在範例資料區段中，選取載入範例資料方塊，以**Dev**使用**public**結構描述**Tickit**將範例資料集載入預設資料庫。
6. 在資料庫組態區段中，管理員使用者名稱和管理員使用者密碼欄位的輸入值。
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 Serverless 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 Amazon Redshift Serverless 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，並在 Amazon Redshift 主控台上開啟 Amazon Redshift 主控台，網址為 https：//https://console.aws.amazon.com/redshift/。
2. 從 Amazon Redshift 主控台中，選擇嘗試 Amazon Redshift Serverless 按鈕以開啟開始使用 Amazon Redshift Serverless 窗格。
3. 在組態區段中，選擇使用預設設定放射。
4. 在開始使用 Amazon Redshift Serverless 窗格底部，選擇儲存組態，以使用預設工作群組、命名空間、憑證和加密設定建立無伺服器資料倉儲。

從 Toolkit 連線至資料倉儲

有 3 種方法可從 Toolkit 連線至資料庫：

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

若要從 Toolkit 連線至位於現有佈建叢集或無伺服器資料倉儲上的資料庫，請完成下列步驟。

> **Important**
>
> 如果您已完成本使用者指南主題的先決條件一節中的步驟，且您的資料倉儲在 Toolkit Explorer 中不可見，請確定您是從 Explorer 中的正確 AWS 區域工作。

使用資料庫使用者名稱和密碼方法連線至您的資料倉儲

1. 從 Toolkit Explorer 中，展開 AWS 區域 資料倉儲所在的 。
2. 展開 Redshift 並選擇您的資料倉儲，以在 VS 程式碼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 連線到您的資料倉儲且程序完成時，您的可用資料庫、資料表和結構描述會顯示在 Toolkit Explorer 中。

使用 連線至您的資料倉儲 AWS Secrets Manager

> **Note**
>
> 此程序需要 AWS 秘密管理員資料庫秘密才能完成。如需如何設定資料庫秘密的指示，請參閱《AWS Secrets Manager 使用者指南》中的建立 AWS Secrets Manager 資料庫秘密。

1. 從 Toolkit Explorer 中，展開 AWS 區域 資料倉儲所在的 。
2. 展開 Redshift 並選擇您的資料倉儲，以在 VS 程式碼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 連線到您的資料倉儲且程序完成時，您的可用資料庫、資料表和結構描述會顯示在 Toolkit Explorer 中。

使用暫時登入資料連線至您的資料倉儲

1. 從 Toolkit Explorer 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 程式碼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，選擇暫時登入資料，並提供每個提示所需的資訊。

4. 當 Toolkit 連線到您的資料倉儲且程序完成時，您的可用資料庫、資料表和結構描述會顯示在 Toolkit Explorer 中。

編輯資料倉儲的連線

您可以編輯資料倉儲的連線，以變更要連線的資料庫。

1. 從 Toolkit Explorer 中，展開 AWS 區域 資料倉儲所在的 。
2. 展開 Redshift，在您要連線的資料倉儲上按一下滑鼠右鍵，選擇編輯連線，並提供您要連線的資料庫名稱。
3. 當 Toolkit 連線到您的資料倉儲且程序完成時，您的可用資料庫、資料表和結構描述會顯示在 Toolkit Explorer 中。

刪除資料倉儲的連線

1. 從 Toolkit Explorer 中，展開 AWS 區域 資料倉儲所在的 。
2. 展開 Redshift，在資料倉儲上按一下滑鼠右鍵，並按一下要刪除的連線，然後選擇刪除連線。這樣做會從 Toolkit Explorer 中移除可用的資料庫、資料表和結構描述。
3. 若要重新連線至您的資料倉儲，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身分驗證方法來連線至資料倉儲。若要使用不同的方法，請選擇對話方塊中的向後箭頭，直到您到達身分驗證提示為止。

執行 SQL 陳述式

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

> **Note**
>
> 若要完成下列每個程序的步驟，您必須先完成從 Toolkit 連線至資料倉儲一節，位於本使用者指南主題中。

1. 從 Toolkit Explorer 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檔案名稱和位置，將筆記本存放在本機，然後選擇確定，以在 VS 程式碼編輯器中開啟筆記本。
3. 從 VS 程式碼編輯器中，輸入您想要在此筆記本中存放的 SQL 陳述式。

4. 選擇全部執行按鈕來執行您輸入的 SQL 陳述式。
5. SQL 陳述式的輸出會顯示在您輸入的陳述式下方。

將 Markdown 新增至筆記本

1. 在 VS 程式碼編輯器的筆記本中，選擇 Markdown 按鈕，將 Markdown 儲存格新增至筆記本。
2. 在提供的儲存格中輸入 Markdown。
3. 您可以使用 Markdown 儲存格右上角的編輯器工具來編輯 Markdown 儲存格。

將程式碼新增至筆記本

1. 在 VS 程式碼編輯器的筆記本中，選擇程式碼按鈕，將程式碼儲存格新增至筆記本。
2. 在提供的儲存格中輸入程式碼。
3. 您可以選擇在程式碼儲存格上方或下方執行程式碼，方法是從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。

使用 Amazon S3

Amazon 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 Code 管理 Amazon S3 物件和資源。

如需 Amazon S3 服務的詳細資訊，請參閱《Amazon S3 使用者指南》。

下列主題說明如何從 使用 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 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。

下列各節說明如何從 使用 Amazon S3 資源 AWS Toolkit for Visual Studio Code。如需使用 Amazon S3 物件的詳細資訊，例如資料夾和檔案 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 Toolkit Explorer 開啟 S3 服務的內容（按一下滑鼠右鍵）選單，然後選擇建立儲存貯體...。或者，選擇建立儲存貯體圖示以開啟建立儲存貯體對話方塊。
2. 在 Bucket Name (儲存貯體名稱) 欄位中，輸入有效的儲存貯體名稱。

   按下 Enter 以建立儲存貯體並關閉對話方塊。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 允許將您的儲存貯體用作可公開存取的 URL，您選擇的儲存貯體名稱必須是全域唯一的。如果另一個帳戶已使用您要使用的名稱建立儲存貯體，您必須使用不同的名稱。
> 如果您無法建立新的儲存貯體，請檢查輸出索引標籤中的AWS 工具組日誌。如果您嘗試使用無效的儲存貯體名稱，則會發生`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，以組織 S3 儲存貯體的內容。您也可以在資料夾中建立資料夾。

1. 從 Toolkit Explorer 中，展開 S3 服務以檢視您的 S3 資源清單。
2. 選擇建立資料夾圖示以開啟建立資料夾對話方塊。或者，開啟儲存貯體或資料夾的內容（按一下滑鼠右鍵）選單，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 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

當您刪除 S3 儲存貯體時，也會刪除其包含的資料夾和物件。因此，當您嘗試刪除儲存貯體時，系統會要求您確認是否要刪除它。

1. 從工具組主功能表中，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）選單，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儲存貯體的名稱，然後按 Enter 鍵刪除儲存貯體並關閉確認提示。

> **Note**
>
> 如果您的儲存貯體包含物件，則會在刪除之前將其清空。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將其刪除。刪除後，您會收到通知，指出已成功刪除。

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。

下列各節說明如何從 使用 Amazon S3 物件 AWS Toolkit for Visual Studio Code。如需使用 Amazon S3 資源的詳細資訊，例如 S3 儲存貯體， AWS Toolkit for Visual Studio Code請參閱本使用者指南中的使用 S3 資源主題。

物件分頁

如果您使用大量的 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. 導覽至 VS 程式碼活動列，然後選擇延伸。
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：每頁項目上限設定。
4. 將預設值變更為您希望在顯示「載入更多」之前顯示的 S3 項目數量。

> **Note**
>
> 有效值包含 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。您建立的所有儲存貯體都會一次顯示。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，以更新 JSON 格式檔案中的設定。

上傳和下載 Amazon S3 物件

您可以將本機儲存的檔案上傳至 Amazon S3 儲存貯體，或從 下載遠端 Amazon S3 物件到您的本機系統 AWS Toolkit for Visual Studio Code。

使用 Toolkit 上傳檔案

1. 從 Toolkit Explorer 展開 Amazon S3 服務，以檢視您的 S3 資源清單。
2. 選擇位於儲存貯體或資料夾旁的上傳檔案圖示，以開啟上傳檔案對話方塊。或者，您可以開啟內容（按一下滑鼠右鍵）選單，然後選擇上傳檔案。

   > **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 物件的內容（按一下滑鼠右鍵）選單，然後選擇上傳至父系。

3. 使用您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 界面或 Command Palette 將檔案上傳至儲存貯體。

1. 若要選取要上傳的檔案，請在 VS 程式碼中選擇該檔案的標籤。
2. 按下 Ctrl+Shift+P 以顯示命令面板。
3. 在命令面板中，輸入 片語`upload file`以顯示建議的命令清單。
4. 選擇 AWS： Upload File 命令以開啟 AWS： Upload File 對話方塊。
5. 出現提示時，請選擇您要上傳的檔案，然後選擇您要上傳該檔案的儲存貯體。
6. 確認您的上傳以關閉對話方塊，並開始上傳程序。上傳完成時，物件會顯示在工具組功能表中，其中包含包含物件大小、上次修改日期和路徑的中繼資料。

下載 Amazon S3 物件

1. 從 Toolkit Explorer 中，展開 S3 服務。
2. 從儲存貯體或資料夾，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，選擇物件附近的下載為圖示。
3. 使用系統的檔案管理員，選擇目的地資料夾，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

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

1. 從 Toolkit Explorer 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 程式碼編輯器中檢視檔案，然後選擇 UI 右上角的鉛筆圖示。

> **Note**
>
> - 如果您重新啟動或結束 VS 程式碼，IDE 會中斷與 S3 資源的連線。如果中斷連線時正在編輯任何遠端 S3 檔案，則編輯會停止。您必須重新啟動 VS 程式碼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。只有在您主動在 VS 程式碼編輯器中檢視唯讀檔案時，才會顯示。
> - 無法以唯讀模式開啟非文字檔案。它們一律以編輯模式開啟。
> - 您無法從編輯唯讀模式切換回唯讀模式，只能切換到其他方向。

複製 Amazon S3 物件的路徑

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

1. 從 Toolkit Explorer 中，展開 S3 服務。
2. 展開資源儲存貯體，其中包含您要複製路徑的物件。
3. 開啟您要複製路徑之物件的內容（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

您可以透過預先簽章的 URL 功能授予有時間限制的下載許可，與他人共用私有 Amazon S3 物件。如需詳細資訊，請參閱使用預先簽章的 URL 共用物件。

1. 從 Toolkit Explorer 中，展開 S3 服務。
2. 從儲存貯體或資料夾，開啟您要共用之物件的內容（按一下滑鼠右鍵）選單。然後，選擇產生預先簽章的 URL 以開啟命令面板。
3. 從命令面板中，輸入 URL 可用來存取物件的分鐘數。然後，選擇 Enter 以確認並關閉對話方塊。
4. 產生預先簽章的 URL 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章 URL。

刪除 Amazon S3 物件

如果物件位於非版本控制的儲存貯體中，您可以永久刪除它。對於已啟用版本控制的儲存貯體，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 Toolkit Explorer 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內容（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。. . 以確認您想要刪除 S3 物件。然後，關閉對話方塊。

Amazon SageMaker Unified Studio for VS 程式碼

Amazon SageMaker Unified Studio 是新一代 Amazon SageMaker 的一部分，是整合 AWS 資料、分析、人工智慧 (AI) 和機器學習 (ML) 服務的統一開發體驗。它提供從單一界面建置、部署、執行和監控工作流程的位置。如需設定 Amazon SageMaker Unifed Studio 與 VS Code IDE 整合的詳細資訊，請參閱《Amazon SageMaker Unified Studio 使用者指南》中的在 VS Code 中設定 Amazon SageMaker Unified Studio 整合。

使用無伺服器應用程式

AWS Toolkit for Visual Studio Code 支援 AWS 無伺服器應用程式。下列主題說明如何從 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

- 無伺服器應用程式入門
- 使用無 AWS 伺服器土地
- 直接從程式碼執行和偵錯 Lambda 函數
- 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
- 對無伺服器應用程式進行故障診斷

無伺服器應用程式入門

下列各節說明如何使用 AWS Serverless Application Model (AWS SAM) 和 CloudFormation 堆疊 AWS Toolkit for Visual Studio Code，從 AWS 無伺服器應用程式 開始建立 。

先決條件

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**
>
> 下列操作可能需要您在變更完成之前結束或重新啟動 VS 程式碼。

- 安裝 AWS SAM 命令列界面 (CLI)。如需如何安裝 CLI AWS SAM 的其他資訊和指示，請參閱本AWS Serverless Application Model 使用者指南中的安裝 AWS SAM CLI 主題。
- 從您的 AWS 組態檔案中，識別您的預設 AWS 區域。如需組態檔案的詳細資訊，請參閱AWS Command Line Interface 《 使用者指南》中的組態和登入資料檔案設定主題。
- 安裝您的語言 SDK 並設定您的工具鏈。如需如何從 設定工具鏈的其他資訊， AWS Toolkit for Visual Studio Code 請參閱《 使用者指南》中的設定工具鏈主題。
- 從 VS Code Marketplace 安裝 YAML 語言支援擴充功能。這是 AWS SAM 範本檔案的 CodeLens 功能可以存取的必要項目。如需 CodeLens 的其他資訊，請參閱 VS Code 文件中的 CodeLens 一節

無伺服器應用程式的 IAM 許可

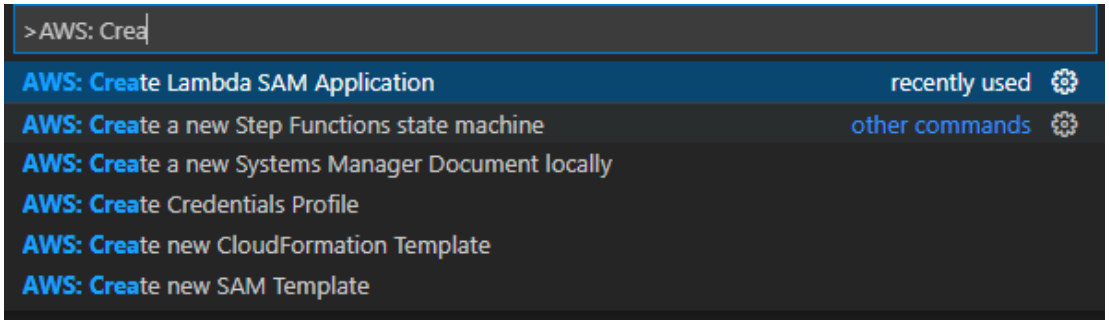
在 Toolkit for VS Code 中，您必須擁有憑證設定檔，其中包含部署和執行無伺服器應用程式所需的 AWS Identity and Access Management (IAM) 許可。您必須擁有下列服務的適當讀取/寫入存取權：IAM CloudFormation、Lambda、Amazon API Gateway、Amazon Simple Storage Service (Amazon S3) 和 Amazon Elastic Container Registry (Amazon ECR)。

如需設定部署和執行無伺服器應用程式所需的身分驗證的詳細資訊，請參閱《 AWS Serverless Application Model 開發人員指南》中的管理資源存取和許可。如需如何設定登入資料的資訊，請參閱本使用者指南AWS IAM 登入資料中的 。

建立新的無伺服器應用程式（本機）

此程序說明如何使用 Toolkit for VS Code 建立無伺服器應用程式 AWS SAM。此程序的輸出是開發主機上的本機目錄，其中包含範例無伺服器應用程式，您可以建置、本機測試、修改和部署至 AWS 雲端。

1. 若要開啟命令調色盤，請選擇檢視、命令調色盤，然後輸入 AWS。
2. 選擇 AWS Toolkit Create Lambda SAM 應用程式。



> **Note**
>
> 如果未安裝 AWS SAM CLI，您會在 VS Code 編輯器的右下角收到錯誤。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 應用程式的執行時間。

> **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。

4. 根據您選取的執行時間，您可能需要為 SAM 應用程式選取相依性管理員和執行時間架構。

   Dependency Manager

   選擇 Gradle 或 Maven。

> **Note**
>
> 此建置自動化工具選擇僅適用於 Java 執行期。

Architecture

選擇 x86_64 或 arm64。

在 ARM64-based 型模擬環境中執行無伺服器應用程式的選項，而非預設的 x86_64 型環境，可用於下列執行時間：

- nodejs12.x (ZIP 和映像）
- nodejs14.x (ZIP 和映像）
- python3.8 (ZIP 和映像）
- python3.9 (ZIP 和映像）
- python3.10 (ZIP 和映像）
- python3.11 (ZIP 和映像）
- python3.12 (ZIP 和映像）
- java8.al2 搭配 Gradle (ZIP 和映像）
- java8.al2 搭配 Maven（僅限 ZIP)
- java11 搭配 Gradle (ZIP 和映像）
- java11 搭配 Maven（僅限 ZIP)

> **Important**
>
> 您必須安裝 1 AWS CLI .33.0 版或更新版本，才能允許應用程式在ARM64-based環境中執行。如需詳細資訊，請參閱先決條件。

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 後， Toolkit for VS Code 需要一些時間來建立專案。

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. 選擇選取資料夾按鈕。

當您開啟應用程式的資料夾時，它會新增至 Explorer 視窗。

從範本執行和偵錯無伺服器應用程式（本機）

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

您可以使用 VS CodeLens 功能來識別合格的 Lambda 函數，以開始設定偵錯行為。CodeLens 可讓您與原始程式碼進行內容感知互動。如需確保您可存取 CodeLens 功能的相關資訊，請參閱本主題稍早的先決條件一節。

> **Note**
>
> 在此範例中，您會偵錯使用 JavaScript 的應用程式。不過，您可以使用 Toolkit for VS 程式碼偵錯功能搭配下列語言和執行時間：
>
> - C# – .NET Core 2.1、3.1；.NET 5.0
> - JavaScript/TypeScript – Node.js 12.x、14.x
> - Python – 3.6、3.7、3.8、3.9、3.10、3.11、3.12
> - Java – 8、8.al2、11
> - Go – 1.x
>
> 您的語言選擇也會影響 CodeLens 偵測合格 Lambda 處理常式的方式。如需詳細資訊，請參閱直接從程式碼執行和偵錯 Lambda 函數。

在此程序中，您會使用本主題先前建立新的無伺服器應用程式（本機）章節中建立的範例應用程式。

1. 若要在 VS Code 的檔案總管中檢視應用程式檔案，請選擇檢視、總管。
2. 從應用程式資料夾 (例如 my-sample-app) 中開啟 `template.yaml` 檔案。

> **Note**
>
> 如果您使用的名稱與 不同的範本`template.yaml`，則不會自動在 YAML 檔案中使用 CodeLens 指標。這表示您必須手動新增偵錯組態。

3. 在 的編輯器中`template.yaml`，前往定義無伺服器資源之範本的 `Resources`區段。在此情況下，這是類型 `HelloWorldFunction`的資源`AWS::Serverless::Function`。

   在此資源的 CodeLens 指示器中，選擇新增偵錯組態。
   使用 `template.yaml` 檔案中的 CodeLens 指標來新增偵錯組態。
4. 在命令面板中，選取應用程式 AWS SAM 將在其中執行的執行時間。
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"` – 確定值為 ，`"template"`讓 AWS SAM 範本成為偵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 範本的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`。

   設定 `launch.json` 檔案以進行以範本為基礎的偵錯。

   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. 如果您認為除錯組態沒有問題，請儲存 `launch.json`。然後，若要開始偵錯，請在 RUN 檢視中選擇綠色「播放」按鈕。

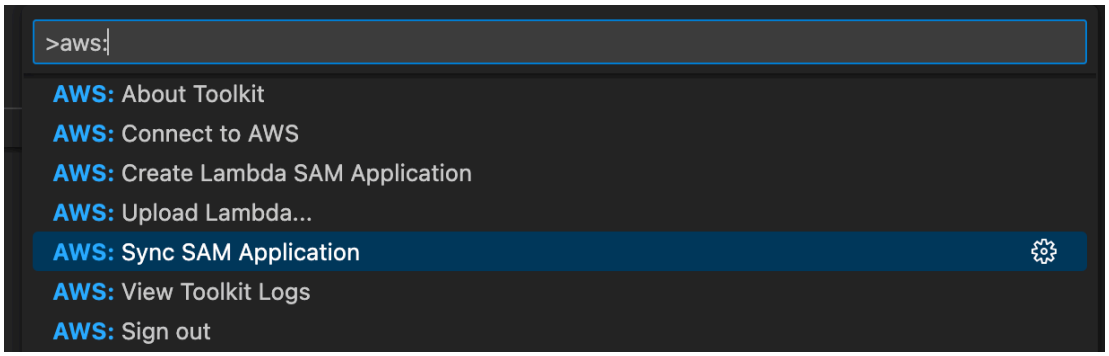
   當除錯工作階段啟動時，DEBUG CONSOLE (除錯主控台) 面板會顯示除錯輸出，並顯示 Lambda 函式傳回的任何值。（偵錯 AWS SAM 應用程式時，會在輸出面板中選取 AWS Toolkit 做為輸出通道。）

同步 AWS SAM 應用程式

會 AWS Toolkit for Visual Studio Code 執行 AWS SAM CLI 命令`sam sync`，將您的無伺服器應用程式部署到 AWS 雲端。如需 AWS SAM 同步的詳細資訊，請參閱《 AWS Serverless Application Model 開發人員指南》中的 AWS SAM CLI 命令參考主題

下列程序說明如何`sam sync`從 Toolkit for VS Code AWS 雲端 使用 將無伺服器應用程式部署至 。

1. 從 VS Code 的主選單中，展開檢視並選擇命令調色盤，以開啟命令調色盤。
2. 從命令調色盤搜尋 AWS ，然後選擇同步 SAM 應用程式以開始設定同步。



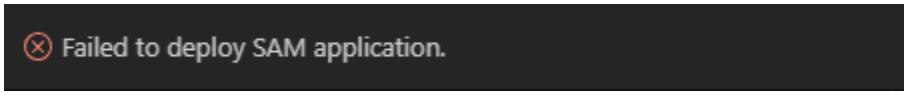
3. 選擇要同步無伺服器應用程式的 AWS 區域。
4. 選擇要用於部署的 `template.yaml` 檔案。
5. 選取現有的 Amazon S3 儲存貯體，或輸入要部署應用程式的新 Amazon S3 儲存貯體名稱。

> **Important**
>
> 您的 Amazon S3 儲存貯體必須符合下列要求：
>
> - 儲存貯體必須位於您要同步的區域。
> - 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擇部署堆疊，或建立新的部署堆疊以輸入新的堆疊名稱。然後，繼續開始同步程序。

> **Note**
>
> 每個工作區和區域會回收先前部署中使用的堆疊。

8. 在同步程序期間，您的部署狀態會在 VS 程式碼的終端機索引標籤中擷取。確認您的同步已從終端機索引標籤成功，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 命令面板存取 AWS Toolkit for Visual Studio Code 日誌。

若要從命令調色盤存取 AWS Toolkit for Visual Studio Code 日誌，請展開檢視，選擇命令調色盤，然後搜尋 **`AWS: View AWS Toolkits Logs`**，然後在填入清單中時選取它。

部署完成時，您會在 AWS Explorer 中看到您的應用程式。如需如何叫用在應用程式中建立的 Lambda 函數的詳細資訊，請參閱本使用者指南中的 使用 AWS Lambda 函數主題。

從 刪除無伺服器應用程式 AWS 雲端

刪除無伺服器應用程式涉及刪除您先前部署到 AWS 雲端的 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 Explorer。
2. 在 AWS Toolkit Explorer 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 CloudFormation。
3. 開啟對應至您要刪除之無伺服器應用程式之 CloudFormation 堆疊名稱的內容 （按一下滑鼠右鍵） 選單，然後選擇刪除 CloudFormation 堆疊。
4. 若要確認要刪除選取的堆疊，請選擇是。

如果堆疊刪除成功， Toolkit for VS Code 會從 AWS Explorer 的 CloudFormation 清單中移除堆疊名稱。

使用無 AWS 伺服器土地

AWS 中的 Serverless Land AWS Toolkit for Visual Studio Code 是一系列功能，可協助您建置事件驅動型架構。下列主題章節說明如何在 AWS Toolkit 中使用 Serverless Land。如需 Serverless Land 的詳細資訊，請參閱 Serverless Land Web 應用程式。

存取無伺服器土地

工具 AWS 組中有 3 個主要進入點可存取無伺服器土地：

- VS 程式碼命令面板
- AWS Toolkit Explorer
- AWS 工具組應用程式建置器瀏覽器

從 VS Code Command Palette 開啟無伺服器土地

若要從 VS Code Command Palette 開啟 Serverless Land，請完成下列步驟。

1. 從 VS 程式碼中，按 **option+shift+p**(Mac) 或 **control+shift+p**(Windows) 開啟命令調色盤。
2. 從 VS Code Command Palette，在搜尋列**AWS Create application with Serverless template**中輸入 。
3. 選擇 AWS：在填入清單中時，使用無伺服器範本建立應用程式。
4. 當程序完成時，Serverless Land 精靈會開啟 VS 程式碼中的為您選取模式應用程式 (1/5) 畫面。

從 AWS Toolkit Explorer 開啟無伺服器土地。

若要從 AWS Toolkit Explorer 開啟 Serverless Land，請完成下列步驟。

1. 從 AWS Toolkit Explorer 中，展開您要開啟無伺服器土地的區域。
2. 開啟 Lambda 節點的內容功能表 （按一下滑鼠右鍵）。
3. 從內容功能表中選擇使用無伺服器範本建立應用程式。
4. 當程序完成時，Serverless Land 精靈會開啟 VS 程式碼中的為您選取模式應用程式 (1/5) 畫面。

從 Application Builder Explorer 開啟無伺服器土地

若要從 AWS Toolkit Application Builder Explorer 開啟 Serverless Land，請完成下列步驟。

1. 從 AWS Toolkit Explorer 導覽至 Application Builder Explorer。
2. 在 Application Builder Explorer 上按一下滑鼠右鍵，然後從內容功能表中選擇使用無伺服器範本建立應用程式。
3. 當程序完成時，Serverless Land 精靈會開啟 VS 程式碼中的為您選取模式應用程式 (1/5) 畫面。

使用 Serverless 範本建立應用程式

若要使用 Serverless 範本建立應用程式，請完成下列步驟。

1. 從無伺服器登陸精靈為您的應用程式選取模式 (1/5) 畫面，為您的應用程式基礎選擇模式。

> **Note**
>
> 若要檢視特定模式的預覽和更多詳細資訊，請選擇位於您要檢視之模式旁邊的無伺服器登陸中開啟圖示。無伺服器登陸模式會在您的預設 Web 瀏覽器中開啟。

2. 在選取執行時間 (2/5) 畫面中，為您的專案選擇執行時間。
3. 在選取 IaC (3/5) 畫面中，為您的專案選擇 IaC 選項。
4. 在選取專案位置 (4/5) 畫面中，選擇要存放專案的位置。
5. 在輸入專案名稱 (5/5) 畫面中，輸入新應用程式的名稱。
6. 當程序完成時，您的新應用程式會顯示在 VS Code Explorer 中，而您的專案會在 VS Code 編輯器中`readme.md`開啟。

> **Note**
>
> 建立新應用程式之後，您可以在 `readme.md` 檔案中找到應用程式類型特有的其他動作。此外，您可以使用 AWS Application Builder 開啟您的 AWS Serverless Application Model (AWS SAM) 應用程式，以進行本機測試、偵錯等。
> 如需有關在 AWS Toolkit 中使用 Application Builder 的詳細資訊，請參閱《 使用者指南》中的使用 AWS Application Builder Explorer 主題。

直接從程式碼執行和偵錯 Lambda 函數

測試 AWS SAM 應用程式時，您可以選擇僅執行 Lambda 函數並進行偵錯，並排除 AWS SAM 範本定義的其他資源。此方法涉及使用 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。

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

語言/執行時間	CodeLens 指標要識別的 Lambda 函數條件
C# (dotnetcore2.1、3.1；.NET 5.0)	函式具備以下特性： • 這是公有類別的公有函數。 • 它有一個或兩個參數。使用兩個參數時，第二個參數必須實作`ILambdaContext` 界面。 • 它在 VS Code 工作區資料夾中的父資料夾中有一個`*.csproj`檔案。 安裝並啟用 ms-dotnettools.csharp 延伸模組（或任何提供 C# 語言符號的延伸模組）。
JavaScript/TypeScript (Node.js 12.x、14.x)	函式具備以下特性： • 它是經過匯出的函式，最多有三個參數。 • 它在 VS Code 工作區資料夾中的父資料夾中有一個`package.json` 檔案。
Python (3.7、3.8、3.9、3.10、3.11、3.12)	函式具備以下特性： • 它是最上層函數。 • 它在 VS Code 工作區資料夾中的父資料夾中有一個`requirements.txt` 檔案。 ms-python.python 延伸模組（或任何提供 Python 語言符號的延伸模組）已安裝並啟用。
Java (8、8.al2、11)	函式具備以下特性：

語言/執行時間	CodeLens 指標要識別的 Lambda 函數條件

	• 這是公有、非抽象類別的公有函數。
	• 它有一個、兩個或三個參數：
	  • 一個參數：參數可以是任何項目。
	  • 兩個參數：參數必須是 ，`java.io.InputStream` 而 `java.io.OutputStream` OR 最後一個參數必須是 `com.amazonaws.services.lambda.runtime.Context` 。
	  • 三個參數：參數必須是 `java.io.InputStream` ，`java.io.OutputStream` 而最後一個參數必須是 `com.amazonaws.services.lambda.runtime.Context` 。
	• 它在 VS Code 工作區資料夾中的父資料夾中具有 `build.gradle` (Gradle) 或 `pom.xml`(Maven) 檔案。
	
	已安裝並啟用 redhat.java 延伸模組 （或任何提供 Java 語言符號的延伸模組）。無論您使用哪個 Java 執行時間，此延伸模組都需要 Java 11。
	
	已安裝並啟用 vscjava.vscode-java-debug 延伸模組 （或任何提供 Java 除錯器的延伸模組）。

語言/執行時間	CodeLens 指標要識別的 Lambda 函數條件
Go (1.x)	函式具備以下特性： • 它是最上層函數。 • 它需要 0 到 2 個引數。如果有兩個引數，第一個引數必須實作 `context.Context` 。 • 它傳回 0 到 2 個引數。如果有 0 個以上的引數，最後一個引數必須實作 `error`。 • 它在 VS Code 工作區資料夾中有一個`go.mod`檔案。 golang.go 延伸模組已安裝、設定和啟用。

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、總管。
2. 從應用程式資料夾（例如 my-sample-app)，展開函數資料夾（在此情況下為 hello-world)，然後開啟 `app.js` 檔案。
3. 在識別合格 Lambda 函數處理常式的 CodeLens 指示器中，選擇 `Add Debug Configuration`。
   存取 Lambda 函數處理常式 CodeLens 指示器中的新增偵錯組態選項。
4. 在命令面板中，選取 AWS SAM 應用程式將在其中執行的執行時間。
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。舉例來說，JavaScript 應用程式的預設值為 `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 `"nodejs12.x"`。
   - `"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"：定義事件酬載的 JSON 格式金鑰數值組。
- "path"：作為事件酬載的檔案的路徑。

在下列範例中，"json"選項會定義承載。

設定 launch.json 檔案以直接叫用 Lambda 函數。

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. 如果您對偵錯組態感到滿意，若要開始偵錯，請選擇 RUN 旁的綠色播放箭頭。

當偵錯工作階段開始時，DEBUG CONSOLE 面板會顯示偵錯輸出，並顯示 Lambda 函數傳回的任何值。（偵錯 AWS SAM 應用程式時，會在輸出面板中選取 AWS Toolkit 作為輸出通道。）

執行並除錯本機 Amazon API Gateway 資源

您可以使用 執行 的 VS 程式碼啟動組態template.yaml，以執行或偵錯 中指定的 AWS SAM API Gateway type=aws-sam 本機資源invokeTarget.target=api。

> **Note**
>
> API Gateway 支援 REST 和 HTTP 這兩種類型的 API。不過，具有 的 API Gateway 功能 AWS Toolkit for Visual Studio Code 僅支援 REST APIs。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：造訪 AWS SAM 專案中的處理常式原始碼 (.js、.cs 或 .py 檔案），將滑鼠游標移至 Lambda 處理常式上，然後選擇新增偵錯組態 CodeLens。然後，在功能表中，選擇標示為 API 事件的項目。
   - 選項 2：使用以下語法編輯launch.json和建立新的啟動組態。

```
{
```

```
  "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 Code Run 面板中，選擇啟動組態 (myConfig在上述範例中命名）。
3. (選用) 在 Lambda 專案程式碼中新增中斷點。
4. 輸入 F5 或在執行面板中選擇播放。
5. 在輸出窗格中檢視結果。

組態

如果您使用 invokeTarget.target 屬性值 api，Toolkit 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

如何提供 AWS 連線資訊。如需詳細資訊，請參閱 除錯無伺服器應用程式的組態選項區段中的AWS 連線 ("aws") 屬性表。

sam

CLI AWS SAM 如何建置應用程式。如需詳細資訊，請參閱 除錯無伺服器應用程式的組態選項區段中的 AWS SAM CLI ("sam") 屬性表。

除錯無伺服器應用程式的組態選項

當您開啟 `launch.json` 檔案來編輯偵錯組態時，您可以使用 VS Code IntelliSense 功能來檢視和自動完成有效的屬性。若要在編輯器中觸發 IntelliSense，請按 Ctrl+Spacebar。

IntelliSense 可讓您尋找和定義直接叫用 Lambda 函數或使用 AWS SAM 範本的屬性。您也可以定義 `"lambda"`（函數的執行方式）、 `"sam"`（CLI AWS SAM 如何建置應用程式） 和 `"aws"`（AWS 連線資訊的提供方式） 的屬性。

AWS SAM：直接 Lambda 處理常式調用 / 範本型 Lambda 調用

屬性	Description
type	指定用於管理啟動組態的延伸模組。一律將 aws-sam 設定為使用 AWS SAM CLI 在本機建置和偵錯。
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	指定要經由指定延伸模組執行的組態類型 (aws-sam)。一律設為 direct-invoke 來啟動 Lambda 函數。
invokeTarget	指定叫用資源的進入點。 如要直接叫用 Lambda 函數，請設定以下 invokeTarget 欄位的值： • target – 設為 code。 • lambdaHandler – 要叫用的 Lambda 函數處理常式名稱。 • projectRoot – 包含 Lambda 函數處理常式之應用程式檔案的路徑。 • architecture – 您本機 SAM Lambda 應用程式執行所在之模擬環境的處理器架構。對於某些執行時間，您可以選擇 arm64 而非預設x86_64架構。如需詳細資訊，請參閱建立新的無伺服器應用程式（本機）。 若要使用 AWS SAM 範本叫用 Lambda 資源，請為下列invokeTarget 欄位設定值： • target – 設為 template。 • templatePath – 範本檔案的 AWS SAM 路徑。 • logicalId – 要叫用的 AWS::Lambda::Function 或 AWS::Serverless::Function 資源名稱。您可以在 YAML 格式 AWS SAM 範本中找到資源名稱。請注意， 會隱含地將 AWS SAM 範本PackageType: Image 中以 AWS 工具組 定義的函數識別為影像型 Lambda 函數。如需詳細資訊，

屬性	Description
	請參閱《 AWS Lambda 開發人員指南》中的 Lambda 部署套件。

Lambda (**"lambda"**) 屬性

屬性	Description

environmentVariables	將操作參數傳遞至 Lambda 函數。例如，如果您要寫入 Amazon S3 儲存貯體，而不是硬式編碼寫入的儲存貯體名稱，請將儲存貯體名稱設定為環境變數。
	
	Note
	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 SAM 範本 (`template.yaml`) 和 `launch.json` 檔案。
	範本中 AWS SAM 環境變數的格式範例：
	`Resources:`
	` HelloWorldFunction:`
	` Type: AWS::Serverless::Function`
	` Properties:`
	`  CodeUri: hello-world/`
	`  Handler: app.lambdaHandlerN10`
	`  Runtime: nodejs10.x`
	`  Environment:`
	`  Variables:`
	`  SAMPLE1: Default Sample 1 Value`
	
	`launch.json` 檔案中環境變數的格式範例：
	`"environmentVariables": {`
	`  "SAMPLE1": "My sample 1 value"`
	` }`

payload	針對您提供給 Lambda 函數作為輸入的事件酬載提供兩個選項。

屬性	Description
	• "json"：定義事件酬載的 JSON 格式金鑰數值組。 • "path"：作為事件酬載的檔案的路徑。
memoryMB	指定為執行調用 Lambda 函數而提供的記憶體 MB (MB)。
runtime	指定 Lambda 函數使用的執行時間。如需詳細資訊，請參閱 AWS Lambda 執行時間。
timeoutSec	設定除錯工作階段逾時之前的可用時間 (以秒為單位)。
pathMappings	指定本機程式碼與其在容器中執行位置的相對位置。 根據預設， Toolkit for VS 程式碼會localRoot 設定為本機工作區中的 Lambda 函數程式碼根目錄，以及remoteRoot 設定為 /var/task ，這是在 Lambda 中執行程式碼的預設工作目錄。如果在 Dockerfile 中或使用 CloudFormation 範本檔案中的 WorkingDirectory 參數變更工作目錄，必須至少指定一個pathMapping 項目，以便偵錯工具能夠成功將本機設定的中斷點映射到 Lambda 容器中執行的程式碼。 pathMappings launch.json 檔案中的 格式化範例： `"pathMappings": [ { "localRoot": " ${workspaceFolder}/sam-app/HelloWorldFunction ", "remoteRoot": " /var/task " } ]` 警告： • 對於以 .NET 映像為基礎的 Lambda 函數，remoteRoot 項目必須是建置目錄。 • 對於以 Node.js 為基礎的 Lambda 函數，您只能指定單一路徑映射項目。

Toolkit for VS 程式碼使用 AWS SAM CLI 在本機建置和偵錯無伺服器應用程式。您可以使用 `launch.json` 檔案中`"sam"`組態的屬性來設定 AWS SAM CLI 命令的行為。

AWS SAM CLI (**`"sam"`**) 屬性

屬性	Description	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 SAM 建置。	空字串
`containerBuild`	指出是否要在類似 Lambda 的 Docker 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`	指定其他本機調用引數。	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`	使用參數來輸入客戶值來自訂您的 AWS SAM 範本。如需更多詳細資訊，請參閱 https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/parameters-section-structure.html 使用者指南中的AWS CloudFormation 參數部分。	`"parameters":{}`

AWS 連線 (**"aws"**) 屬性

屬性	Description	預設值
`credentials`	從登入資料檔案選取特定設定檔（例如 `profile:default`）以取得 AWS 登入資料。	現有共用組態檔案或共用 AWS 登入資料檔案提供給 Toolkit for VS Code 的登入資料。AWSAWS
`region`	設定服務的 AWS 區域（例如 us-east-1)。	與作用中登入資料設定檔相關聯的預設 AWS 區域。

範例：範本啟動組態

以下是 AWS SAM 範本目標的啟動組態檔案範例：

```
{
    "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 函數目標的啟動組態檔案範例：

```
{
```

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

對無伺服器應用程式進行故障診斷

本主題詳細說明使用 Toolkit for VS 程式碼建立無伺服器應用程式時可能遇到的常見錯誤，以及如何解決這些錯誤。

主題

- 如何搭配 SAM 啟動組態使用 samconfig.toml？
- 錯誤：「RuntimeError：容器不存在」
- 錯誤：「docker.errors.APIError：500 伺服器錯誤... 您已達到提取率限制。」
- 錯誤：「500 伺服器錯誤：掛載 C：\Users\...」
- 使用 WSL 時，Webview（例如，「叫用 AWS」表單）會中斷
- 偵錯 TypeScript 應用程式，但中斷點無法運作

如何搭配 SAM 啟動組態使用 samconfig.toml？

在啟動組態的 `sam.localArguments` 屬性中設定 `--config-file`引數，以指定 SAM CLI samconfig.toml 的位置。例如，如果 samconfig.toml 檔案位於工作區的最上層：

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

```
}
```

錯誤：「RuntimeError：容器不存在」

如果您的系統沒有足夠的磁碟空間用於 Docker 容器，則 `sam build`命令可以顯示此錯誤。如果您的系統儲存體只有 1-2 GB 的可用空間，`sam build`可能會在處理期間失敗，即使系統儲存體在建置開始之前並未完全填滿。如需詳細資訊，請參閱此 GitHub 問題。

錯誤：「docker.errors.APIError：500 伺服器錯誤... 您已達到提取率限制。」

Docker Hub 會限制匿名使用者可以提出的請求。如果您的系統達到限制，Docker 失敗，且此錯誤會出現在 VS 程式碼的 OUTPUT 檢視中：

```
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 Docker 服務已使用您的 Docker Hub 憑證進行身分驗證。

錯誤：「500 伺服器錯誤：掛載 C：\Users\...」

Windows 使用者在偵錯 AWS SAM 應用程式時可能會看到此 Docker 掛載錯誤：

```
Fetching lambci/lambda:nodejs10.x Docker container image......
2019-07-12 13:36:58 Mounting C:\Users\<username>\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 ...
```

嘗試重新整理共用磁碟機的登入資料（在 Docker 設定中）。

使用 WSL 時，Webview（例如，「叫用 AWS」表單）會中斷

這是 Cisco VPN 使用者的已知 VS 程式碼問題。如需詳細資訊，請參閱此 GitHub 問題。

此 WSL 追蹤問題建議解決方法。

偵錯 TypeScript 應用程式，但中斷點無法運作

如果沒有將編譯的 JavaScript 檔案連結至來源 TypeScript 檔案的來源映射，就會發生這種情況。若要更正此問題，請開啟 `tsconfig.json` 檔案，並確保已設定下列選項和值：`"inlineSourceMap": true`。

使用 Systems Manager 自動化文件

AWS Systems Manager 可讓您了解和控制 上的基礎設施 AWS。Systems Manager 提供統一的使用者介面，讓您可以檢視來自多個 AWS 服務的操作資料，並自動化跨 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，可用來執行常見的維護和部署任務，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
- 對 Toolkit for VS Code 中的 Systems Manager Automation 文件進行故障診斷

假設和先決條件

在開始之前，請確認：

- 您已安裝 Visual Studio Code 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 Automation 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，您必須擁有憑證設定檔，其中包含建立、編輯、發佈和刪除 Systems Manager Automation 文件所需的 AWS Identity and Access Management (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：

JSON

```
{
    "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。如需如何設定登入資料設定檔的資訊，請參閱 AWS IAM 登入資料。

建立新的 Systems Manager 自動化文件

您可以在 JSONYAML中或使用 Visual Studio Code 建立新的自動化文件。當您建立新的自動化文件時，它會以無標題的檔案顯示。您可以命名您的檔案並將其儲存在 VS 程式碼中，但檔案的名稱不可見 AWS。

建立新的自動化文件

1. 開啟 VS Code。

2. 在檢視功能表上，選擇命令調色盤以開啟命令調色盤。
3. 在命令面板中，輸入 AWS Toolkit Create a new Systems Manager Document Locally。
4. 為 Hello World 範例選擇其中一個入門範本。
5. 選擇 JSON 或 YAML。

   建立新的自動化文件。

> **Note**
>
> VS Code 中的新自動化文件不會自動出現在 中 AWS。您必須先將其發佈至 ， AWS 才能執行它。

開啟現有的 Systems Manager 自動化文件

您可以使用 AWS Explorer 尋找現有的 Systems Manager Automation 文件。當您開啟現有的自動化文件時，它會在 VS 程式碼中顯示為無標題檔案。

開啟自動化文件

1. 開啟 VS Code。
2. 從左側導覽中 AWS ，選擇 AWS以開啟 Explorer。
3. 在 AWS Explorer 中，針對 Systems Manager，選擇您要開啟的文件上的下載圖示，然後選擇文件版本。檔案會以該版本的格式開啟。否則，請選擇下載為 JSON 或下載為 YAML。

> **Note**
>
> 在本機將自動化文件儲存為 VS 程式碼中的檔案，並不會使其出現在 中 AWS。在執行 AWS 之前，它需要發佈到 。

編輯 Systems Manager 自動化文件

如果您擁有任何自動化文件，它們會出現在 AWS Explorer 中 Systems Manager 文件的「由我擁有」類別中。您可以擁有已存在於 中的自動化文件 AWS，也可以擁有先前 AWS 從 VS 程式碼發佈至 的新文件或更新的文件。

當您開啟要在 VS 程式碼中編輯的自動化文件時，您可以比在 中執行更多操作 AWS 管理主控台。例如：

- JSON 和 YAML 格式都有結構描述驗證。
- 文件編輯器中有可供您建立任何自動化步驟類型的程式碼片段。
- JSON 和 中的各種選項都有自動完成支援YAML。

處理版本

Systems Manager Automation 文件使用版本進行變更管理。您可以在 VS 程式碼中選擇自動化文件的預設版本。

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 中編輯自動化文件後，您可以將文件發佈至 AWS。

發佈自動化文件

1. 使用 中概述的程序，開啟您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. 在開啟檔案的右上角，選擇上傳圖示。
4. 在發佈工作流程對話方塊中，選擇您要發佈自動化文件 AWS 的區域。
5. 如果您要發佈新文件，請選擇快速建立。否則，請選擇快速更新以更新該 AWS 區域中現有的自動化文件。
6. 輸入此自動化文件的名稱。

當您發佈現有自動化文件的更新至 時 AWS，文件會新增新版本。

刪除 Systems Manager 自動化文件

您可以在 VS 程式碼中刪除自動化文件。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不會刪除 AWS 啟動時建立或修改的資源。

刪除您的自動化文件

1. 開啟 VS Code。
2. 從左側導覽中，選擇 AWS以開啟 AWS Explorer。
3. 在 AWS Explorer 中，針對 Systems Manager，開啟您要刪除之文件的內容 （按一下滑鼠右鍵） 選單，然後選擇刪除文件。

執行 Systems Manager 自動化文件

發佈自動化文件到 後 AWS，您就可以執行它，以代表您在 AWS 帳戶中執行任務。若要執行自動化文件，您可以使用 AWS 管理主控台、APIs、 AWS CLI或 AWS Tools for PowerShell。如需如何執行自動化文件的說明，請參閱AWS Systems Manager 《 使用者指南》中的執行簡單的自動化。

或者，如果您想要搭配 Systems Manager APIs AWS SDKs 來執行自動化文件，請參閱 AWS SDK 參考。

> **Note**
>
> 執行自動化文件可以在 中建立新的資源， AWS 並可能產生帳單成本。強烈建議您在啟動自動化文件之前，先了解該文件會在帳戶中建立的內容。

對 Toolkit for VS Code 中的 Systems Manager Automation 文件進行故障診斷

我在 VS Code 中儲存了自動化文件，但我在 中看不到它 AWS 管理主控台。

在 VS 程式碼中儲存自動化文件並不會將自動化文件發佈至 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，並出現許可錯誤。

請確定您的 AWS 登入資料設定檔具有發佈自動化文件的必要許可。如需許可政策範例，請參閱 Systems Manager 自動化文件的 IAM 許可。

我發佈了自動化文件到 AWS，但我在 中看不到它 AWS 管理主控台。

請確定您已將文件發佈至您在 中瀏覽的相同 AWS 區域 AWS 管理主控台。

我已刪除自動化文件，但仍然需要為其建立的資源支付費用。

刪除自動化文件並不會刪除其建立或修改的資源。您可以從AWS 帳單管理主控台識別您建立 AWS 的資源、探索您的費用，然後選擇要從該處刪除的資源。

AWS Step Functions

您可以使用 AWS Step Functions建立工作流程（也稱為狀態機器），以建置分散式應用程式、自動化程序、協調微服務，以及建立資料和機器學習管道。下列主題說明如何在 AWS Step Functions 中使用 AWS Toolkit for Visual Studio Code。如需 AWS Step Functions 服務的詳細資訊，請參閱 AWS Step Functions開發人員指南。

主題

- 使用 AWS Step Functions
- 使用 AWS Step Functions Workflow Studio

使用 AWS Step Functions

下列各節說明如何使用 AWS Step Functions `Amazon State Language (ASL)` AWS Toolkit 中包含狀態機器定義的檔案。如需 AWS Step Functions 狀態機器的詳細資訊，請參閱《 AWS Step Functions開發人員指南》中的 Step Functions 中的了解狀態機器主題。

檢視 Step Functions 狀態機器

若要在 AWS Toolkit Explorer 中檢視包含狀態機器定義的現有ASL檔案，請完成下列步驟。

1. 從 AWS Toolkit Explorer 中，展開包含您要檢視之ASL檔案的區域。

2. 展開 Step Functions 標題。
3. 您的ASL檔案會顯示在 AWS Explorer 中。

建立 Step Functions 狀態機器

在 AWS Toolkit 中，您可以從 檔案建立新的 Step Functions 狀態機器，也可以使用 範本。下列程序說明如何從 檔案建立 Step Functions 狀態機器。如需從範本建立 SFN； 狀態機器的詳細資訊，請參閱本使用者指南主題中的下列狀態機器範本區段。

> **Note**
>
> 若要在 VS 程式碼中使用 Step Functions，包含狀態機器定義的 Amazon State Language(ASL) 檔案副檔名必須以 `asl.json`、 `asl.yml`或 結尾`.asl.yaml`。
> 根據預設，相關的 Step Functions 檔案會在 Workflow Studio 中開啟。如需透過 AWS Toolkit 在 Workflow Studio 中運作的詳細資訊，請參閱本使用者指南中的使用 Workflow Studio 主題。

1. 在 VS 程式碼的工作區中，建立新的檔案。
2. 為您的檔案命名，並將副檔名指定為 `asl.json`、 `asl.yml`或 `.asl.yaml`。
3. 建立時， AWS Toolkit 會在 AWS Step Functions Workflow Studio 中開啟新檔案。
4. 從 Workflow Studio 選擇公用程式功能表中的儲存按鈕，以儲存您的新ASL檔案。

從範本建立 Step Functions 狀態機器

在 AWS Toolkit 中，您可以從範本建立 Step Functions 狀態機器。範本程序會建立包含狀態機器定義的ASL檔案，為您的專案提供起點。下列程序說明如何從 AWS Toolkit 中的範本建立 Step Functions 狀態機器。

1. 從 AWS Toolkit Explorer 中，展開您要在其中建立 Step Functions 狀態機器的區域。
2. 開啟（按一下滑鼠右鍵） Step Functions 的內容選單，然後選擇建立新的 Step Functions 狀態機器，以在 VS 程式碼中開啟選取入門範本 (1/2) 精靈。
3. 從選取入門範本 (1/2) 精靈中，選擇 Step Functions 狀態機器的範本類型以繼續。
4. 在選取範本格式 (2/2) 畫面中，為您的範本格式選擇 YAML 或 JSON。
5. 包含狀態機器定義的新ASL檔案會在 VS 程式碼編輯器中開啟。

下載 Step Functions 狀態機器

若要將遠端儲存的 Step Functions 狀態機器下載至 VS Code 的本機執行個體，請完成下列步驟。

1. 從 AWS Toolkit Explorer 中，展開包含您要下載之 Step Functions 狀態機器的區域。
2. 展開 Step Functions，然後在您要下載的 Step Functions 狀態機器上按一下滑鼠右鍵，然後選擇下載定義...。
3. 指定在本機存放 Step Functions 狀態機器的位置以繼續。
4. 當程序完成時，Step Functions 狀態機器會在 Workflow Studio 中開啟。

將變更儲存至 Step Functions 狀態機器

下列程序說明如何儲存對 Step Functions 狀態機器所做的變更。

> **Note**
>
> 在 Workflow Studio 同步至本機檔案時所做的編輯，但在 VS Code 編輯器或 Workflow Studio 中儲存您的工作之前，不會儲存。如果您在開啟 Workflow Studio 時修改並儲存本機檔案，且您的ASL檔案中未偵測到任何錯誤，則在儲存完成時，您會在 Workflow Studio 中收到成功通知。不過，如果您的本機檔案包含無效 JSON或 YAML且您嘗試儲存，則本機檔案無法同步，且您在 Workflow Studio 中收到警告通知。

1. 從 Workflow Studio 中包含狀態機器定義的開啟ASL檔案，導覽至公用程式按鈕。
2. 選擇 Save (儲存) 按鈕。
3. VS 程式碼會在檔案儲存時通知您。

執行 Step Functions 狀態機器

下列程序說明如何在 AWS Toolkit 中執行 Step Functions 狀態機器。

1. 從 AWS Toolkit Explorer 中，展開包含您要執行之 Step Functions 狀態機器的區域。
2. 展開 Step Functions，然後在您要執行的 Step Functions 狀態機器上按一下滑鼠右鍵。
3. 從內容功能表中，選擇開始執行以啟動啟動程序。
4. 啟動狀態會顯示在 VS 程式碼的 AWS Toolkit Output 視窗中。

使用程式碼片段

程式碼片段是根據您正在處理的程式碼產生的自動化建議。若要在工具組中搭配 Step Functions 使用程式碼片段，請完成下列步驟。

> **Note**
>
> 若要在 VS 程式碼中使用 Step Functions 程式碼片段，包含狀態機器定義的ASL檔案副檔名必須以 `.asl.json`、 `.asl.yml`或 結尾`.asl.yaml`。
> 根據預設，相關的 Step Functions 檔案會在 Workflow Studio 中開啟。

1. 從 VS 程式碼中，開啟包含您要修改或建立新ASL檔案之狀態機器定義的ASL檔案。
2. 如果您處於設計模式，請從 Workflow Studio 切換至程式碼模式。
3. 從 Workflow Studio 程式碼編輯器，將游標放在 `"States"` 屬性中。
4. 按 **`control + space`**開啟程式碼片段功能表，可按 **`control + space`**以根據 `"State"` 存取其他屬性`"Type"`。
5. 從清單中選擇您想要的程式碼片段。

程式碼驗證

當您在 Workflow Studio 中處理 Step Functions 時，程式碼驗證會主動識別錯誤，並針對下列項目提出建議：

- 缺少屬性
- 不正確的值
- 非終端機狀態
- 指向不存在的狀態

使用 AWS Step Functions Workflow Studio

下列各節說明如何在 中使用 AWS Step Functions Workflow Studio AWS Toolkit for Visual Studio Code。如需 AWS Step Functions Workflow Studio 的詳細資訊，請參閱《 AWS Step Functions開發人員指南》中的開發工作流程主題

開啟工作流程 Studio

以下清單說明您可以在 VS 程式碼中開啟 Workflow Studio 的不同路徑。

> **Note**
>
> 若要在 VS Code 中使用 Workflow Studio，包含狀態機器定義的 Amazon State Language(ASL) 檔案副檔名必須以 asl.json、 asl.yml或 結尾asl.yaml。如需在 AWS Toolkit 中下載或建立新的狀態機器定義的詳細資訊，請參閱《 使用者指南AWS Step Functions》中的下載狀態機器和建立狀態機器一節。

- 從 AWS Explorer 開啟包含狀態機器定義的ASL檔案的內容選單 （ 按一下滑鼠右鍵 ），然後選擇在 Workflow Studio 中開啟。
- 從包含狀態機器定義的開啟ASL檔案中，選擇 VS Code 編輯器視窗中標籤旁邊的 Open with Workflow Studio 圖示。
- 從包含狀態機器定義的開啟ASL檔案中，選擇位於檔案頂端的 CodeLens 命令 Open with Workflow Studio。
- 關閉並重新開啟包含狀態機器定義的ASL檔案會自動在 Workflow Studio 中重新開啟檔案，除非手動停用預設的 Workflow Studio。

設計模式和程式碼模式

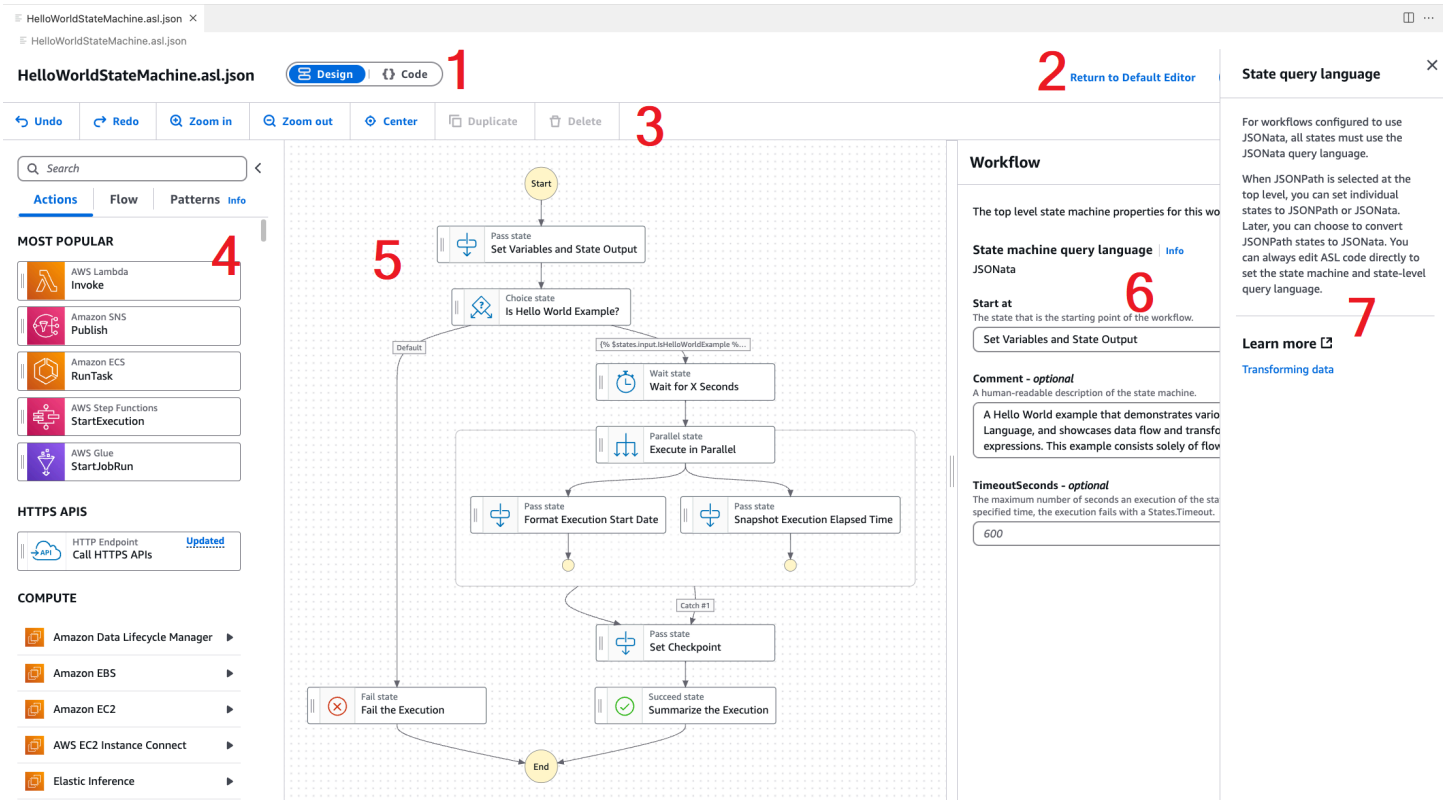
Workflow Studio 有兩種模式可讓您使用包含狀態機器定義的ASL檔案：設計模式和程式碼模式。設計模式提供圖形界面，可讓您在建置原型時將工作流程視覺化。程式碼模式具有整合式程式碼編輯器，您可以在其中檢視、寫入和編輯工作流程中的ASL定義。

> **Note**
>
> 如需設計與程式碼模式中每個 UI 區段的詳細資訊，請參閱《 AWS Step Functions開發人員指南》中的使用 Workflow Studio 主題。並非所有 Workflow Studio 功能都可在 AWS Toolkit 中使用，例如 Config 模式。

設計模式 UI 有 7 個主要區段，如下圖所示和說明。

1. 模式按鈕：在設計和程式碼模式之間切換的按鈕。

2. 公用程式按鈕：一組用於執行任務的按鈕，例如結束 Workflow Studio、儲存工作流程，或在 JSON 或 YAML 檔案中匯出ASL定義。
3. 設計工具列：工具列包含一組執行常見動作的按鈕，例如復原、刪除和縮放控制。
4. 狀態瀏覽器：包含工作流程畫布drag-and-drop狀態的瀏覽器。狀態會組織成索引標籤，並定義為動作、流程和模式。
5. Canvas 和工作流程圖表：工作流程的視覺化轉譯，您可以在其中捨棄、重組和選取組態狀態。
6. Inspector Panel：檢視和編輯畫布上選取之任何狀態的屬性。根據畫布工作流程圖表中選取的狀態，標籤會填入組態、輸入/輸出、變數和錯誤處理的狀態特定選項。
7. 資訊連結：當您需要協助時，開啟包含內容資訊的面板。這些面板也包含 AWS Step Functions開發人員指南中相關主題的連結。



在設計期間使用單一狀態測試

從 Workflow Studio 測試狀態 UI，您可以測試狀態機器的個別狀態。這包括提供狀態輸入、設定變數以及進行 AWS SAM 和 CloudFormation 定義替換的能力。

若要進一步了解基礎設施即程式碼 (IaC)、資源定義和轉換資料，請參閱《 AWS Step Functions開發人員指南》中的使用 AWS SAM 建置 Step Functions 工作流程和使用 JSONata 轉換資料。

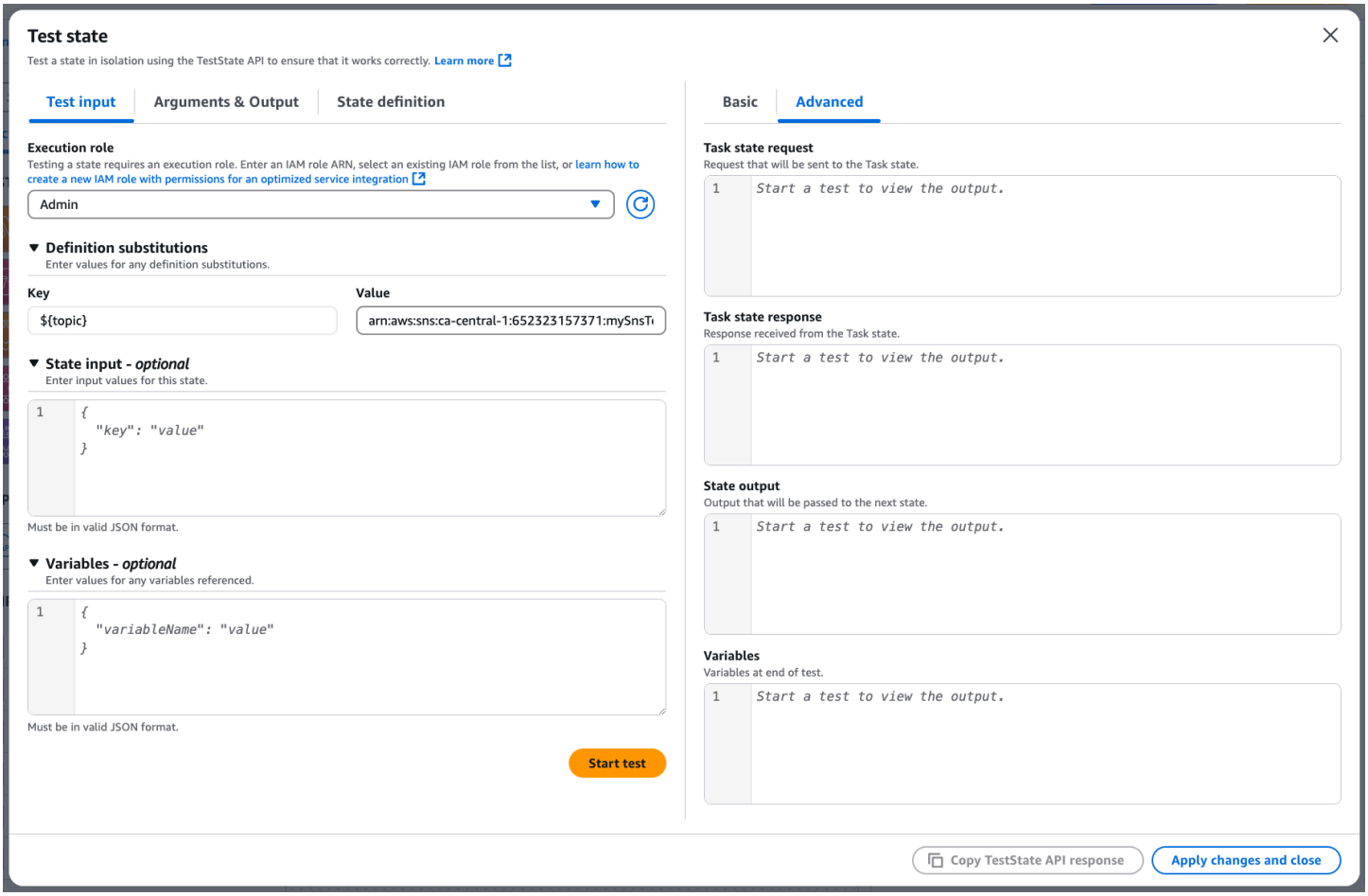
下列程序說明如何在 Workflow Studio 中開啟 test-state UI。

開啟測試狀態 UI

1. 從 Workflow Studio 的設計模式索引標籤中，導覽至畫布，然後在檢查器面板中選擇狀態以開啟畫布。
2. 從檢查器面板中，選擇測試狀態按鈕。
3. 測試狀態 UI 會在 VS 程式碼中開啟。

測試狀態 UI 有 3 個主要標籤：測試輸入、引數和輸出、狀態定義。測試輸入索引標籤有 3 個額外的欄位，可讓您提供狀態輸入、設定變數，以及從 AWS SAM 或 CloudFormation 範本指定定義替換。在狀態定義索引標籤中，您可以調整工作流程並重新測試。完成測試後，您可以套用和儲存狀態機器定義的變更。

下列螢幕擷取畫面顯示測試狀態 UI，其中包含主題資源定義。



預設停用 Workflow Studio

根據預設，Workflow Studio 是包含狀態機器定義ASL之檔案的預設編輯器。您可以透過修改本機.vscode目錄中settings.json的檔案來停用預設設定。如果您預設停用 Workflow Studio，仍然可以透過位於本主題的開啟工作流程 Studio 區段中列出的方法來存取它。

若要從 VS 程式碼編輯settings.json檔案，請完成下列步驟。

1. 從 VS 程式碼中，按 **option+shift+p**(Mac) 或 **ctrl+shift+p**(Windows) 開啟命令調色盤。
2. 從 VS Code Command Palette 中，**Open User Settings (JSON)**在搜尋欄位中輸入 ，然後在填入清單中時選擇 選項。
3. 在編輯器settings.json中的 中，將下列修改新增至您的 檔案。

```
                {
                    "workbench.editorAssociations": {
                    // Use all the following overrides or a specific one for a
certain file type
                    "*.asl.json": "default",
                    "*.asl.yaml": "default",
                    "*.asl.yml": "default"
                    }
                }
```

4. 將變更儲存至 settings.json，然後重新整理或重新啟動 VS 程式碼。

使用威脅編寫器

您可以使用 AWS Toolkit for Visual Studio Code 來使用 Threat Composer 工具。Threat Composer 是一種威脅模型工具，可簡化您的威脅模型建構程序。

如需 Threat Composer 工具的詳細資訊，請參閱 Threat Composer GitHub 儲存庫。

下列主題說明如何在 中使用 Threat Composer AWS Toolkit for Visual Studio Code。

主題

- 從 工具組使用威脅編寫器

從 工具組使用威脅編寫器

透過 Threat Composer，您可以直接在 VS Code 中建立、檢視和編輯 Threat Composer 威脅模型。如需 Threat Composer 工具的詳細資訊，請參閱 Threat Composer GitHub 儲存庫。

下列各節說明如何存取 中的 Threat Composer 工具 AWS Toolkit for Visual Studio Code。

從工具組存取威脅編寫器

您可以從 Toolkit 存取 Threat Composer 有 3 種主要方式。

透過現有的威脅模型存取威脅編寫器

若要開啟威脅編寫器，請在 VS 程式碼中開啟現有的威脅模型檔案 （延伸 `.tc.json`)。Threat Composer 會在 VS Code 編輯器視窗中自動開啟並轉譯威脅模型檔案的視覺化效果。

建立新的威脅編寫器威脅模型

1. 從 VS 程式碼主功能表中，展開檔案，然後選擇新檔案。
2. 從新檔案對話方塊中，選擇威脅編寫器檔案...。
3. 出現提示時，輸入 `file name`，然後按 **enter**鍵開啟威脅編寫器，並在新的 VS 程式碼編輯器視窗中建立空威脅模型檔案的視覺化效果。

從命令面板建立新的威脅編譯器威脅模型

1. 從 VS 程式碼中，按 **Cmd + Shift + P**或 **Ctrl + Shift + P**(Windows) 開啟命令面板。
2. 在搜尋欄位中，輸入 **Threat Composer**，然後在填入結果時選擇建立新的威脅編寫器檔案。
3. 出現提示時，輸入 `file name`，然後按 **enter**鍵開啟威脅編寫器，並在新的 VS 程式碼編輯器視窗中建立空威脅模型檔案的視覺化效果。

使用 資源

除了存取預設在 AWS Explorer 中列出的 AWS 服務之外，您也可以前往 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。可以新增的一些資源包括 Amazon AppFlow、Amazon Kinesis Data Streams、 AWS IAM 角色、Amazon VPC 和 Amazon CloudFront 分佈。

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：

- 檢視目前 AWS 區域中可用於此資源類型的現有資源清單。
- 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 AWS 的文件，以及建立資源模型的結構描述 (JSON 和 YAML 格式）。
- 透過編輯並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**
>
> *在目前版本的建立、編輯和刪除資源 AWS Toolkit for Visual Studio Code 選項中， 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。實驗功能可能會從 中移除， AWS Toolkit for Visual Studio Code 恕不另行通知。
> 若要允許將實驗性功能用於 資源，請開啟 VS Code IDE 中的設定窗格，然後展開延伸模組，然後選擇 AWS Toolkit。
> 在AWS 工具組實驗下，選取 jsonResourceModification，以允許您建立、更新和刪除資源。
> 如需詳細資訊，請參閱使用實驗功能。

存取資源的 IAM 許可

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

除了服務資源的許可之外，IAM 實體還需要許可，以允許 Toolkit for VS Code 代表其呼叫 AWS Cloud Control API 操作。雲端控制 API 作業允許 IAM 使用者或角色存取和更新遠端資源。

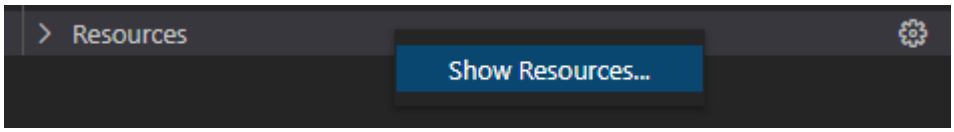
授予許可的最簡單方法是將 AWS 受管政策 PowerUserAccess 連接至使用 Toolkit 介面呼叫這些 API 操作的 IAM 實體。這個受管政策會授與一系列許可，以執行應用程式開發任務，包括呼叫 API 作業。

如需定義遠端資源上允許 API 作業的特定許可，請參閱《AWS 雲端控制 API 使用者指南》。

新增並與現有資源互動

1. 在 AWS Explorer 中，以滑鼠右鍵按一下資源，然後選擇顯示資源。

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 Explorer 的資源類型，然後按傳回或選擇確定以確認。

   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。

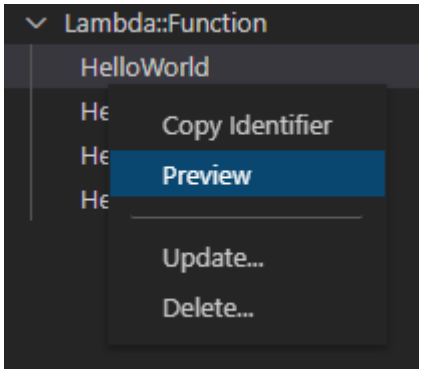
   可用資源清單會顯示在資源類型下方。

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：

   - 複製資源識別符：將特定資源的識別符複製到剪貼簿。(例如：AWS `Toolkit:DynamoDB::Table` 資源可以使用 `TableName` 屬性進行識別。)
   - Preview (預覽)：檢視描述資源的 JSON 格式範本的唯讀版本。

     顯示資源範本之後，您可以選擇編輯器索引標籤右側的更新圖示來修改它。若要更新資源，您必須???啟用必要的 。
   - 更新：在 VS 程式碼編輯器中編輯資源的 JSON 格式範本。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。（在此版本的 ???中，刪除資源目前是 AWS Toolkit for Visual Studio Code。）

     > **Warning**
     >
     > 如果您刪除資源，使用該資源的任何 AWS CloudFormation 堆疊都將無法更新。若要修正此更新失敗，您需要重新建立資源，或移除堆疊 CloudFormation 範本中對此資源的參考。如需詳細資訊，請參閱這篇知識中心文章。



建立和編輯資源

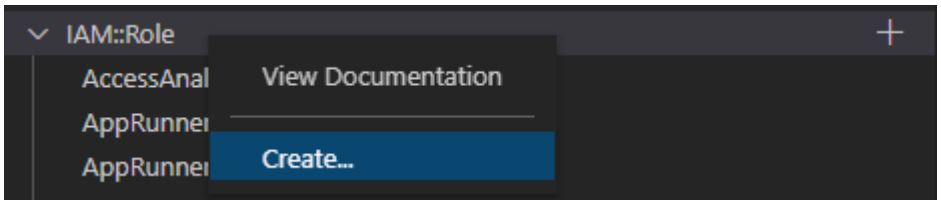
> **Important**
>
> 建立和更新資源的功能目前在此版本的 AWS Toolkit for Visual Studio Code中為???。

建立新資源的程序包含將資源類型新增至資源清單，然後編輯定義資源、其內容及其屬性的 JSON 格式範本。

例如，屬於資源類型的`AWS Toolkit:SageMaker::UserProfile`資源是以範本定義，該範本會為 Amazon SageMaker AI Studio 建立使用者設定檔。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。

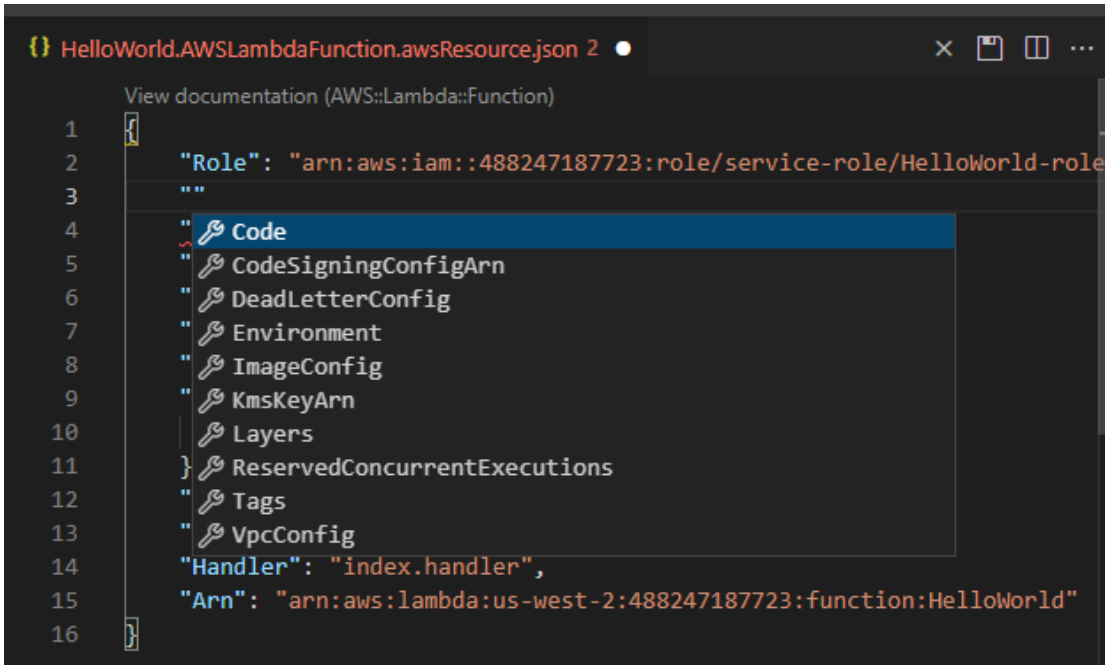
1. 透過在資源上按一下滑鼠右鍵並選擇顯示資源，為您要建立的資源新增資源類型。
2. 在資源下新增資源類型之後，請選擇加號 ("+") 圖示，在新的編輯器中開啟範本檔案。

   或者，您可以在資源類型的名稱上按一下滑鼠右鍵，然後選擇建立。您也可以選擇檢視文件，以存取如何建立資源模型的相關資訊。



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。當您將滑鼠暫留在屬性類型上時，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波紋紅色底線表示。



4. 完成宣告資源後，請選擇儲存圖示以驗證您的範本，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。選擇檢視日誌以識別您需要修正的範本元素。

故障診斷 AWS Toolkit for Visual Studio Code

下列各節包含有關 AWS Toolkit for Visual Studio Code 和從 工具組使用 AWS 服務的一般疑難排解資訊。如需 AWS 工具組中與疑難排解 SAM 問題特別相關的問題，請參閱《 使用者指南》中的對無伺服器應用程式進行疑難排解主題。

主題

- 疑難排解最佳實務
- 設定檔 ... 在組態檔案中找不到
- SAM json 結構描述：無法在 template.yaml 檔案中變更結構描述

疑難排解最佳實務

以下是疑難排解 AWS Toolkit for Visual Studio Code 問題時的建議最佳實務。如需如何貢獻 的詳細資訊 AWS Toolkit for Visual Studio Code，請參閱 AWS Toolkit for Visual Studio Code GitHub 儲存庫中的貢獻主題 AWS Toolkit for Visual Studio Code。

- 嘗試在傳送報告之前重新建立您的問題或錯誤。
- 記下重新建立過程中每個步驟、設定和錯誤訊息的詳細記錄。
- 收集您的 AWS Toolkit Debug Logs。如需如何尋找 AWS Toolkit Debug 日誌的詳細說明，請參閱本使用者指南主題中的如何尋找 AWS 日誌程序。
- 檢查是否有開啟的請求、已知的解決方案，或在 AWS Toolkit for Visual Studio Code GitHub 儲存庫的問題區段中報告您未解決AWS Toolkit for Visual Studio Code 的問題。

> **Note**
>
> 下列程序說明如何檢視 AWS Toolkit Debug 日誌。檢視 Amazon Q Debug 日誌的程序完全相同，除非您從 VS Code Command Palette 選擇 Amazon Q： 檢視日誌。

如何尋找 AWS Toolkit for Visual Studio Code 偵錯日誌

1. 從 VS 程式碼中，按 **Cmd + Shift + P**或 **Ctrl + Shift + P**(Windows) 開啟命令調色盤**AWS View Logs**，然後在搜尋欄位中輸入 。
2. 選擇AWS 檢視日誌以在 VS 程式碼終端機輸出視窗中開啟您的 AWS 工具組日誌。

3. 從 VS 程式碼終端機輸出視窗中，展開齒輪圖示選單，然後選擇偵錯。
4. 再次展開齒輪圖示選單，然後選擇設為預設。
5. 按 **Cmd + Shift + P**或 **Ctrl + Shift + P**(Windows) 重新開啟命令調色盤，搜尋 **Reload Window**，然後選擇開發人員：重新載入視窗。
6. VS 程式碼會重新載入，而 VS 程式碼終端機輸出視窗會顯示您更新的 AWS Toolkit Debug 日誌。

設定檔 ... 在組態檔案中找不到

問題

> **Note**
>
> 此問題僅適用於 `~/.aws/config` 檔案，不適用於 `~/.aws/credentials` 檔案。如需 AWS 組態和 AWS 登入資料檔案的詳細資訊，請參閱 SDK 和工具參考指南中的共用組態和登入資料檔案主題。 AWS

選擇登入資料時 AWS Toolkit 日誌會顯示具有此結構的訊息：`Profile` *`name`* `could not be found in shared credentials file`。

以下是 AWS Toolkit 日誌中此錯誤的外觀範例：

```
        2023-08-08 18:20:45 [ERROR]: _aws.auth.reauthenticate: Error: Unable to
authenticate connection
        -> CredentialsProviderError: Profile vscode-prod-readonly could not be found
in shared credentials file.
```

解決方案

如果您的設定檔已存在於 中`~/.aws/config`，請檢查其開頭是否為 `[profile` 。以下是正確建構的使用者設定檔範例：

```
        [profile example]
        region=us-west-2
        credential_process=...
```

以下是不正確建構的使用者設定檔範例：

```
        [example]
        region=us-west-2
        credential_process=...
```

SAM json 結構描述：無法在 template.yaml 檔案中變更結構描述

問題

您無法在 SAM template.yaml 中手動選取不同的 json 結構描述

解決方案

更新至 vscode-yaml 1.11+ 版後，您可以在 YAML 檔案頂端新增**`yaml-language-server`**管制線，以強制 URI 使用結構描述。如需有關在 Redhat 開發人員 GitHub 儲存庫的 yaml 語言伺服器主題中使用內嵌結構描述一節的詳細資訊。以下是**`yaml-language-server`**管制線的範例。

```
        # yaml-language-server: $schema=https://raw.githubusercontent.com/aws/
serverless-application-model/main/samtranslator/schema/schema.json
```

AWS Toolkit for VS 程式碼的安全性

的雲端安全性 AWS 是最高優先順序。身為 AWS 客戶，您可以受益於資料中心和網路架構，這些架構是為了滿足最安全敏感組織的需求而建置。

安全性是 AWS 與您之間的共同責任。共同責任模式將其描述為雲端的安全性，和雲端中的安全性：

- 雲端的安全性 – AWS 負責保護在 中執行 AWS 服務的基礎設施 AWS 雲端。 AWS 也提供您可以安全使用的服務。作為AWS 合規計畫的一部分，第三方稽核人員會定期測試和驗證我們安全的有效性。若要了解適用於 AWS Toolkit for VS Code 的合規計劃，請參閱AWS 合規計劃的 服務範圍。
- 雲端的安全性 – 您的責任取決於您使用 AWS 的服務。您也必須對其他因素負責，包括資料的機密性、您公司的要求和適用法律和法規。

本文件可協助您了解如何在使用 Toolkit for VS Code 時套用共同責任模型。下列主題說明如何設定 Toolkit for VS Code 以符合您的安全與合規目標。您也會了解如何使用其他 AWS 服務來協助您監控和保護 Toolkit for VS Code 資源。

主題

- AWS Toolkit for VS 程式碼中的資料保護

AWS Toolkit for VS 程式碼中的資料保護

AWS 共同責任模型適用於 AWS Toolkit for VS Code 中的資料保護。如此模型所述， AWS 負責保護執行所有 的全域基礎設施 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。如需資料隱私權的詳細資訊，請參閱資料隱私權常見問答集。 如需歐洲資料保護的詳細資訊，請參閱一般資料保護規則 (GDPR) 中心。

基於資料保護目的，我們建議您保護 AWS 帳戶 登入資料，並使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 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：

- 每個帳戶均要使用多重要素驗證 (MFA)。
- 使用 SSL/TLS 與 AWS 資源通訊。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 使用 設定 API 和使用者活動記錄 AWS CloudTrail。如需有關使用 CloudTrail 追蹤擷取 AWS 活動的資訊，請參閱AWS CloudTrail 《 使用者指南》中的使用 CloudTrail 追蹤。
- 使用 AWS 加密解決方案，以及其中的所有預設安全控制 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。
- 如果您在 AWS 透過命令列界面或 API 存取 時需要 FIPS 140-3 驗證的密碼編譯模組，請使用 FIPS 端點。如需有關 FIPS 和 FIPS 端點的更多相關資訊，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 Toolkit for VS Code 或使用 AWS 服務 主控台、API AWS CLI或其他 AWS SDKs 時。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。如果您提供外部伺服器的 URL，我們強烈建議請勿在驗證您對該伺服器請求的 URL 中包含憑證資訊。

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。如需獲得此文件更新的通知，您可以訂閱 RSS 摘要。

變更	描述	日期
Amazon SageMaker Unified Studio	與 Amazon SageMaker Unified Studio 服務整合。	2025 年 9 月 18 日
LocalStack	新增了使用者指南主題，以支援 LocalStack 的啟動。	2025 年 9 月 11 日
使用 AWS Lambda 函數	更新使用者指南主題，以在工具組中包含更新的 Lambda 功能。	2025 年 7 月 17 日
AWS Lambda 遠端偵錯	將新的 AWS Lambda 遠端偵錯主題新增至 AWS Toolkit for Visual Studio Code 使用者指南。	2025 年 7 月 17 日
AWS Lambda 主控台到 IDE	AWS Lambda console 將 IDE 主題的新 新增至 AWS Toolkit for Visual Studio Code 使用者指南。	2025 年 7 月 17 日
更新 AWS Step Functions 內容並新增對 Workflow Studio 的支援	新增了 的現有內容 AWS Step Functions 和 AWS Step Functions Workflow Studio 的使用者指南主題的更新，以支援功能啟動。	2025 年 3 月 6 日
AWS 無伺服器土地	已將新的 AWS Serverless Land 主題新增至 AWS Application Builder TOC。	2025 年 3 月 6 日

更新防火牆和閘道以允許存取	必須允許列出的端點和資源清單，才能存取 和 AWS Toolkit for Visual Studio Code Amazon Q for VS Code 延伸模組中的所有服務和功能。	2025 年 2 月 28 日
支援 Amazon ECR App Runner	在 AWS Toolkit 中新增從 Amazon Elastic Container Registry 節點啟動 AWS App Runner 服務的文件支援。	2025 年 2 月 6 日
Amazon DocumentDB	將新的 Amazon DocumentDB 主題新增至 AWS Toolkit for Visual Studio Code 使用者指南。	2025 年 2 月 6 日
EC2 支援	Amazon Elastic Compute Cloud 服務的支援已新增至工具組。	2025 年 1 月 31 日
AWS 文件	新增 AWS 文件的新使用者指南主題。	2025 年 1 月 20 日
Amazon CloudWatch Logs Live Tail	新增子主題以支援 中的 Amazon CloudWatch Logs Live Tail 功能 AWS Toolkit for Visual Studio Code。	2024 年 12 月 15 日
AWS 應用程式建置器	將新的 AWS Application Builder 主題新增至 AWS Toolkit for Visual Studio Code 使用者指南。	2024 年 10 月 30 日
基礎設施編寫器	AWS Application Composer 現在是 AWS Infrastructure Composer。	2024 年 10 月 3 日

AWS Identity and Access Management (IAM) Access Analyzer 更新	更新 IAM Access Analyzer 內容以包含新的 API 參考。	2024 年 7 月 10 日
AWS Identity and Access Management (IAM) Access Analyzer	新增了 IAM Access Analyzer 的新使用者指南主題。	2024 年 5 月 23 日
連接至 AWS 授權流程已更新	已更新授權流程，以反映驗證程序的變更，以及 Amazon Q 與 的區隔 AWS Toolkit for Visual Studio Code。	2024 年 4 月 30 日
VS 程式碼的 Amazon Q 延伸模組	自 2024 年 4 月 30 日起，CodeWhisperer 現已成為 Amazon Q 的一部分，且 Amazon Q 可作為 VS Code 的延伸。	2024 年 4 月 30 日
在開發環境中支援虛擬私有雲端	更新涵蓋 UI 變更的內容，以支援開發環境中的 VPC。	2024 年 1 月 21 日
基礎設施編寫器	將新的 Infrastructure Composer 主題新增至 AWS Toolkit for Visual Studio Code 使用者指南。	2023 年 11 月 28 日
CodeCatalyst 的 SSO 支援	更新內容以涵蓋 CodeCatalyst 和開發環境的 IAM Identity Center 支援。	2023 年 11 月 17 日
新增 VS 程式碼和工具組下載連結	使用 VS Code 和 的下載連結更新內容 AWS Toolkit for Visual Studio Code。	2023 年 11 月 1 日
Amazon Redshift 主題	將新的 Amazon Redshift 主題新增至 AWS Toolkit for Visual Studio Code 使用者指南。	2023 年 10 月 17 日

連接至 AWS 授權流程已更新	授權流程已更新，以專注於服務特定的身分驗證方法。	2023 年 9 月 29 日
建立的使用者指南：建立 CloudFormation 範本	建立新的使用者指南，說明如何使用 Toolkit for VS 程式碼建立 CloudFormation 範本	2021 年 12 月 17 日
次要 UI 更新	將 "Preview Machine State" 的現有文字更新為 "Render graph"，以便更符合 UI。	2021 年 12 月 14 日
建立了 Amazon Elastic Container Service Exec 的使用者指南	這是 Amazon ECS Exec 的概觀。	2021 年 12 月 13 日
建立 AWS IoT Toolkit for VS Code 服務的使用者指南	此使用者指南旨在協助您開始使用適用於 Toolkit for VS Code 的 AWS IoT 服務。	2021 年 11 月 22 日
實驗功能支援	新增啟用 AWS 服務的實驗性功能的支援。	2021 年 10 月 14 日
支援 AWS 資源	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
的 Amazon ECR 服務概觀 AWS Toolkit for Visual Studio Code	新增可在 VS Code 中存取的 Amazon ECR 服務功能概觀和逐步解說	2021 年 10 月 14 日
支援 ARM64 環境	您現在可以在 ARM64-based模擬環境以及 x86_64 型環境中執行無伺服器應用程式。	2021 年 10 月 1 日
AWS 無伺服器應用程式	新增支援在 ARM64 平台上執行 AWS SAM 應用程式	2021 年 9 月 30 日
格式化更新 Node.js 區段	根據客戶意見回饋，已更新 Node.js/ TypeScript 的格式。	2021 年 8 月 12 日

App Runner 支援	新增對 AWS App Runner 的支援 AWS Toolkit for Visual Studio Code。	2021 年 8 月 11 日
偵錯 Go 函數	新增對本機 Go 函數偵錯的支援。	2021 年 5 月 10 日
偵錯 Java 函數	新增對本機 Java 函數偵錯的支援。	2021 年 4 月 22 日
的 YAML 支援 AWS Step Functions	新增 的 YAML 支援 AWS Step Functions。	2021 年 3 月 4 日
偵錯 Amazon API Gateway 資源	新增對本機 Amazon API Gateway 資源偵錯的支援。	2020 年 12 月 1 日
Amazon API Gateway	新增對 Amazon API Gateway 的支援。	2020 年 12 月 1 日
AWS 無伺服器應用程式	新增對具有無伺服器應用程式的 Lambda 容器映像的支援。	2020 年 12 月 1 日
AWS Systems Manager 支援	新增對 Systems Manager Automation 文件的支援。	2020 年 9 月 30 日
CloudWatch Logs	新增適用於 CloudWatch Logs 的 支援。	2020 年 8 月 24 日
Amazon S3	新增對 Amazon S3 的支援。	2020 年 7 月 30 日
AWS Step Functions 支援	新增 的支援 AWS Step Functions。	2020 年 3 月 31 日
使用 Amazon EventBridge 結構描述	新增對 Amazon EventBridge 結構描述的支援	2019 年 12 月 1 日
AWS CDK	AWS CDK 服務的預覽版本。	2019 年 11 月 25 日

使用外部登入資料程序	新增使用外部登入資料程序來取得 AWS 登入資料的相關資訊。	2019 年 9 月 25 日
使用 IntelliSense 處理任務定義檔案	增加支援以 IntelliSense 來處理 Amazon ECS 任務定義檔案。	2019 年 9 月 24 日
的使用者指南 AWS Toolkit for Visual Studio Code	一般可用性版本。	2019 年 7 月 11 日
的使用者指南 AWS Toolkit for Visual Studio Code	更新文件結構，以清楚呈現、易於使用。	2019 年 7 月 3 日
安裝 AWS Toolkit for Visual Studio Code	新增安裝語言開發套件，以支援各種工具鏈。	2019 年 6 月 12 日
設定您的工具鏈	新增有關設定各種工具鏈的資訊。	2019 年 6 月 12 日
初始版本	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。