



使用者指南

AWS 使用 Amazon Q 的工具組



AWS 使用 Amazon Q 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、任從何貶低或使 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 使用 Amazon Q 的工具組	1
什麼是 Toolkit AWS for Visual Studio with Amazon Q	1
AWS Explorer	1
Amazon Q	1
相關資訊	2
Amazon Q	3
什麼是 Amazon Q	3
下載 工具組	4
從 Visual Studio Marketplace 下載 Toolkit	4
來自 的其他 IDE 工具組 AWS	4
入門	5
安裝和設定	5
先決條件	5
安裝 AWS 工具組	6
解除安裝 AWS Toolkit	7
連線至 AWS	8
先決條件	9
AWS 從 Toolkit 連線至	9
Amazon Q Developer	10
AWS 工具組	1
文件和教學課程	13
對安裝問題進行故障診斷	13
Visual Studio 的管理員許可	14
取得安裝日誌	14
安裝不同的 Visual Studio 延伸模組	15
聯絡支援	15
設定檔和視窗繫結	16
Toolkit for Visual Studio 的設定檔和視窗繫結	16
身分驗證和存取	17
IAM Identity Center	17
從 驗證 IAM Identity Center AWS Toolkit for Visual Studio	17
IAM 憑證	19
建立 IAM 使用者	19
建立登入資料檔案	20

從工具組編輯 IAM 使用者憑證	20
從文字編輯器編輯 IAM 使用者憑證	21
從 AWS Command Line Interface (AWS CLI) 建立 IAM 使用者	21
AWS 建置器 ID	22
多重要素驗證 (MFA)	22
步驟 1：建立 IAM 角色以將存取權委派給 IAM 使用者	22
步驟 2：建立擔任角色許可的 IAM 使用者	23
步驟 3：新增政策以允許 IAM 使用者擔任角色	23
步驟 4：管理 IAM 使用者的虛擬 MFA 裝置	24
步驟 5：建立設定檔以允許 MFA	24
外部憑證	25
更新防火牆和閘道	26
AWS Toolkit for Visual Studio 端點	26
Amazon Q 外掛程式端點	26
Amazon Q Developer 端點	27
Amazon Q 程式碼轉換端點	27
身分驗證端點	27
身分端點	27
遙測	28
參考	28
使用 AWS 服務	30
Amazon CodeCatalyst	30
什麼是 Amazon CodeCatalyst？	30
開始使用 CodeCatalyst	31
使用 CodeCatalyst	32
疑難排解	33
CloudWatch Logs 整合	34
設定 CloudWatch Logs	34
使用 CloudWatch Logs	35
管理 Amazon EC2 執行個體	41
Amazon Machine Image 和 Amazon EC2 執行個體檢視	41
啟動 Amazon EC2 執行個體	43
連線至 Amazon EC2 執行個體	45
結束 Amazon EC2 執行個體	48
管理 Amazon ECS 執行個體	51
修改服務屬性	51

停止任務	51
刪除服務	52
刪除叢集	52
建立儲存庫	52
刪除儲存庫	53
從 AWS Explorer 管理安全群組	53
建立安全群組	53
為安全群組新增許可	54
從 Amazon EC2 執行個體建立 AMI	55
在 Amazon Machine Image 上設定啟動許可	56
Amazon Virtual Private Cloud (VPC)	57
使用 建立用於部署的公有私有 VPC AWS Elastic Beanstalk	57
使用 Visual Studio 的 CloudFormation 範本編輯器	62
在 Visual Studio 中建立 CloudFormation 範本專案	62
在 Visual Studio 中部署 CloudFormation 範本	65
在 Visual Studio 中格式化 CloudFormation 範本	67
從 AWS Explorer 使用 Amazon S3	69
建立 Amazon S3 儲存貯體	69
從 AWS Explorer 管理 Amazon S3 儲存貯體	69
將檔案和資料夾上傳至 Amazon S3	71
來自 AWS Toolkit for Visual Studio 的 Amazon S3 檔案操作	72
從 AWS Explorer 使用 DynamoDB	76
建立 DynamoDB 資料表	77
將 DynamoDB 資料表檢視為網格	78
編輯和新增屬性和值	79
掃描 DynamoDB 資料表	81
AWS CodeCommit 搭配 Visual Studio Team Explorer 使用	82
的登入資料類型 AWS CodeCommit	82
連線至 AWS CodeCommit	82
建立儲存庫	84
設定 Git 登入資料	84
複製儲存庫	86
使用 儲存庫	87
在 Visual Studio 中使用 CodeArtifact	88
將 CodeArtifact 儲存庫新增為 NuGet 套件來源	88
AWS Explorer 的 Amazon RDS	89

啟動 Amazon RDS 資料庫執行個體	89
在 RDS 執行個體中建立 Microsoft SQL Server 資料庫	96
Amazon RDS 安全群組	98
從 AWS Explorer 使用 Amazon SimpleDB	101
從 AWS Explorer 使用 Amazon SQS	103
建立佇列	104
刪除佇列	104
管理佇列屬性	104
傳送訊息至佇列	105
身分和存取權管理	106
建立和設定 IAM 使用者	107
建立 IAM 群組	108
將 IAM 使用者新增至 IAM 群組	108
產生 IAM 使用者的登入資料	110
建立 IAM 角色	112
建立 IAM 政策	113
AWS Lambda	115
基本 AWS Lambda 專案	115
基本 AWS Lambda 專案建立 Docker 影像	121
教學課程：使用 建置和測試無伺服器應用程式 AWS Lambda	128
教學課程：建立 Amazon Rekognition Lambda 應用程式	134
教學課程：搭配 使用 Amazon Logging Framework AWS Lambda 來建立應用程式日誌	142
部署至 AWS	145
發佈至 AWS	145
先決條件	146
支援的應用程式類型	146
將應用程式發佈至 AWS 目標	147
AWS Lambda	148
先決條件	149
相關主題	149
列出透過 .NET Core CLI 可用的 Lambda 命令	149
從 .NET Core CLI 發佈 .NET Core Lambda 專案	150
部署至 AWS Elastic Beanstalk	152
部署 ASP.NET 應用程式 (傳統)	152
部署 ASP.NET 應用程式 (.NET Core) (舊版)	163
指定 AWS 登入資料	165

重新發佈至 Elastic Beanstalk (舊版)	165
自訂部署 (傳統)	167
自訂部署 (.NET Core)	169
多個應用程式支援	172
部署至 Amazon EC2 Container Service	175
指定 AWS 登入資料	176
部署 ASP.NET Core 2.0 應用程式 (遠) (舊版)	177
部署 ASP.NET Core 2.0 應用程式 (EC2)	184
疑難排解	188
疑難排解最佳實務	188
檢視和篩選 Amazon Q 安全性掃描	189
工具 AWS 組未正確安裝	189
防火牆和代理設定	190
防火牆和代理設定故障診斷	190
自訂憑證	191
允許列出和其他步驟	192
安全	193
資料保護	193
身分和存取權管理	194
目標對象	194
使用身分驗證	195
使用政策管理存取權	196
AWS 服務 如何使用 IAM	197
對 AWS 身分和存取進行故障診斷	198
合規驗證	199
恢復能力	199
基礎設施安全性	200
組態與漏洞分析	200
文件歷史紀錄	202
文件歷史紀錄	202
.....	ccviii

AWS 使用 Amazon Q 的工具組

這是 AWS Toolkit for Visual Studio 搭配 Amazon Q 的使用者指南。如果您要尋找 AWS Toolkit for VS Code，請參閱 [《》的使用者指南 AWS Toolkit for Visual Studio Code](#)。

什麼是 Toolkit AWS for Visual Studio with Amazon Q

Toolkit AWS for Visual Studio with Amazon Q 是 Visual Studio IDE 的延伸，可讓您更輕鬆地開發、偵錯和部署使用 Amazon Web Services 的 .NET 應用程式。Visual Studio 2022 版及更新版本支援 AWS Toolkit with Amazon Q。如需如何下載和安裝套件的詳細資訊，請參閱本使用者指南中的 [安裝和設定](#) 主題。

Note

Toolkit for Visual Studio 也針對 Visual Studio 2008、2010、2012、2013、2015、2017 和 2019 版本發行。不過，不再支援這些版本。如需詳細資訊，請參閱本使用者指南中的 [安裝和設定](#) 主題。

AWS Toolkit with Amazon Q 包含下列功能，可增強您的開發體驗。

AWS Explorer

AWS Explorer 工具視窗可在 IDE 的檢視選單中存取，並可讓您與 Visual Studio 中的 AWS 服務互動。如需支援的 AWS 服務和功能清單，請參閱《使用者指南》中的 [使用 AWS 服務](#) 主題。

Amazon Q

在 Visual Studio 中與 Amazon Q Developer 聊天，詢問有關在 建置 AWS 以及協助軟體開發的問題。Amazon Q 可以解釋編碼概念和程式碼片段、產生程式碼和單元測試，並透過偵錯或重構來改善程式碼。

若要安裝和設定 Toolkit for Visual Studio 的 Amazon Q，請參閱本使用者指南中的 [入門](#) 主題。若要進一步了解如何使用 Amazon Q Developer，請參閱 [《Amazon Q Developer 使用者指南》](#) 中的 [Amazon Q Developer in IDEs](#) 主題。如需 Amazon Q 計劃和定價的詳細資訊，請參閱 [Amazon Q 定價](#) 指南。

相關資訊

若要開啟問題或檢視目前開啟的問題，請造訪 <https://github.com/aws/aws-toolkit-visual-studio/issues>。

若要進一步了解 Visual Studio，請造訪 <https://visualstudio.microsoft.com/vs/>。

Amazon Q

什麼是 Amazon Q

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安全掃描。

若要進一步了解如何在 中使用 Amazon Q Developer AWS Toolkit for Visual Studio，請參閱 [《Amazon Q Developer 使用者指南》](#) 中的 [Amazon Q Developer in IDEs](#) 主題。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 Visual Studio

您可以透過 IDE 中的 Visual Studio Marketplace 下載、安裝和設定 Toolkit for Visual Studio。如需詳細說明，請參閱本使用者指南入門主題中的[安裝 AWS Toolkit for Visual Studio](#) 一節。

從 Visual Studio Marketplace 下載 Toolkit

導覽至 Web 瀏覽器中的 Visual Studio [AWS 下載網站](#)，以下載 [Toolkit for Visual Studio](#) 安裝檔案。

來自的其他 IDE 工具組 AWS

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

AWS Toolkit for Visual Studio Code 連結

- 依照此連結，從 VS Code Marketplace [下載 AWS Toolkit for Visual Studio Code](#)。
- 若要進一步了解 AWS Toolkit for Visual Studio Code，請參閱[AWS Toolkit for Visual Studio Code](#) 《使用者指南》。

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。

入門

AWS Toolkit for Visual Studio 可讓您的 AWS 服務和資源從 Visual Studio 整合開發環境 (IDE) 取得。

為了協助您開始使用，下列主題說明如何安裝、設定和設定 AWS Toolkit for Visual Studio。

主題

- [安裝和設定 AWS Toolkit for Visual Studio](#)
- [連線至 AWS](#)
- [針對的安裝問題進行故障診斷 AWS Toolkit for Visual Studio](#)
- [設定檔和視窗繫結](#)

安裝和設定 AWS Toolkit for Visual Studio

下列主題說明如何下載、安裝、設定和解除安裝 AWS Toolkit for Visual Studio。

主題

- [先決條件](#)
- [安裝 AWS Toolkit for Visual Studio](#)
- [解除安裝 AWS Toolkit for Visual Studio](#)

先決條件

以下是設定 支援版本的先決條件 AWS Toolkit for Visual Studio。

- Visual Studio 19 或更新版本
- Windows 10 或更新版本的 Windows 版本
- 管理員存取 Windows 和 Visual Studio
- 作用中 IAM AWS 登入資料

Note

不支援的版本 AWS Toolkit for Visual Studio 適用於 Visual Studio 2008、2010、2012、2013、2015 和 2017。若要下載不支援的版本，請導覽至 [AWS Toolkit for Visual Studio](#) 登陸頁面，然後從下載連結清單中選擇您想要的版本。若要進一步了解 IAM 登入資料或註冊帳戶，請造訪 [AWS 主控台](#) 闡道。

安裝 AWS Toolkit for Visual Studio

若要安裝 AWS Toolkit for Visual Studio，請從下列程序尋找您的 Visual Studio 版本，並完成必要的步驟。您可以在 [AWS Toolkit for Visual Studio](#) 登陸頁面 AWS Toolkit for Visual Studio 找到所有版本的下載連結。

Note

如果您在安裝時遇到問題 AWS Toolkit for Visual Studio，請參閱本指南中的 [疑難排解安裝問題](#) 主題。

安裝 AWS Toolkit for Visual Studio for Visual Studio 2022

若要從 Visual Studio 安裝 AWS Toolkit for Visual Studio 2022，請完成下列步驟：

1. 從主功能表中，導覽至延伸模組，然後選擇管理延伸模組。
2. 從搜尋方塊中，搜尋 AWS。
3. 選擇 Visual Studio 2022 相關版本的下載按鈕，並遵循安裝提示。

Note


您可能需要手動關閉並重新啟動 Visual Studio 以完成安裝程序。

4. 當下載和安裝完成時，您可以從檢視功能表中選擇 AWS Explorer AWS Toolkit for Visual Studio 來開啟。

安裝 AWS Toolkit for Visual Studio for Visual Studio 2019

若要從 Visual Studio 安裝 AWS Toolkit for Visual Studio 2019，請完成下列步驟：

1. 從主功能表中，導覽至延伸模組，然後選擇管理延伸模組。
2. 從搜尋方塊中，搜尋 AWS。
3. 選擇 Visual Studio 2017 和 2019 的下載按鈕，然後依照提示操作。

 Note

您可能需要手動關閉並重新啟動 Visual Studio 以完成安裝程序。

4. 當下載和安裝完成時，您可以從檢視功能表中選擇 AWS Explorer AWS Toolkit for Visual Studio 來開啟。


解除安裝 AWS Toolkit for Visual Studio

若要解除安裝 AWS Toolkit for Visual Studio，請從下列程序尋找您的 Visual Studio 版本，並完成必要的步驟。

解除安裝 AWS Toolkit for Visual Studio for Visual Studio 2022

若要從 Visual Studio 解除安裝 AWS Toolkit for Visual Studio 2022，請完成下列步驟：

1. 從主功能表中，導覽至延伸模組，然後選擇管理延伸模組。
2. 從管理延伸模組導覽功能表中，展開已安裝標題。
3. 找到 AWS Toolkit for Visual Studio 2022 擴充功能，然後選擇解除安裝按鈕。

 Note

如果導覽功能表的已安裝區段 AWS Toolkit for Visual Studio 中看不到，您可能需要重新啟動 Visual Studio。

4. 依照畫面上的提示完成解除安裝程序。

解除安裝 AWS Toolkit for Visual Studio for Visual Studio 2019

若要從 Visual Studio 解除安裝 AWS Toolkit for Visual Studio 2019，請完成下列步驟：

1. 從主功能表中，導覽至工具，然後選擇管理延伸模組。
2. 從管理延伸項目導覽功能表中，展開已安裝標題。

3. 找到 AWS Toolkit for Visual Studio 2019 延伸模組，然後選擇解除安裝按鈕。
4. 依照畫面上的提示完成解除安裝程序。

解除安裝 AWS Toolkit for Visual Studio for Visual Studio 2017

若要在 Visual Studio 中解除安裝 AWS Toolkit for Visual Studio 2017，請完成下列步驟：

1. 從主功能表中，導覽至工具，然後選擇延伸模組和更新。
2. 從延伸項目與更新導覽功能表中，展開已安裝標題。
3. 找到 AWS Toolkit for Visual Studio 2017 擴充功能，然後選擇解除安裝按鈕。
4. 依照畫面上的提示完成解除安裝程序。

解除安裝 AWS Toolkit for Visual Studio for Visual Studio 2013 或 2015

若要解除安裝 AWS Toolkit for Visual Studio 2013 或 2015，請完成下列步驟：

1. 從 Windows 控制面板開啟程式和功能。

Note

您可以從 `appwiz.cpl` Windows 命令提示字元或 Windows Run 對話方塊執行，立即開啟程式和功能。

2. 從已安裝的程式清單中，開啟適用於 AWS Windows 的工具的內容選單（按一下滑鼠右鍵）。
3. 選擇解除安裝，然後依照提示完成解除安裝程序。

Note

在解除安裝程序期間，不會刪除您的範例目錄。如果您已修改範例，則會保留此目錄。必須手動移除此目錄。

連線至 AWS

下列各節說明如何開始使用 AWS Toolkit for Visual Studio 搭配 Amazon Q。當您第一次在安裝擴充功能後啟動 Visual Studio 時，編輯器視窗中會顯示入門。從入門索引標籤中，您可以完成下列動作。

- 啟用或停用 Amazon Q 和 AWS Toolkit。
- 使用新登入資料新增和驗證。
- 使用現有的登入資料進行驗證。
- 存取文件和教學課程，協助您開始使用 Amazon Q 和 AWS Toolkit。

先決條件

若要開始使用 Amazon Q 和 AWS Toolkit，您需要使用 AWS 登入資料進行身分驗證。如果您先前已透過其他 AWS 工具或服務（例如 AWS Command Line Interface）設定 AWS 帳戶和身分驗證，則 AWS Toolkit 會自動偵測您的登入資料。如果您是初次使用 AWS 或尚未建立帳戶，您可以從註冊[AWS 入口網站註冊](#) AWS 帳戶。如需設定新 AWS 帳戶的詳細資訊，請參閱 AWS 設定使用者指南中的[概觀](#)主題。

AWS 從 Toolkit 連線至

若要從 AWS Toolkit 連線至 AWS 您的帳戶，請完成以下操作，隨時開啟入門索引標籤。

在 Visual Studio 中開啟入門索引標籤

1. 從 Visual Studio 的主要功能表展開延伸模組，然後展開 AWS Toolkit 子功能表。
2. 選擇 Getting started (入門)。
3. 開始使用索引標籤會在 Visual Studio 編輯器視窗中開啟。

從入門索引標籤中，有 2 個主要區段：

- 功能：在本節中，您可以啟用或停用 Amazon Q 和 AWS Toolkit 等功能。
- 文件和教學課程：您已啟用功能的參考集合。

Note

只有在啟用一或多個功能時，才會顯示文件和教學課程區段。

Amazon Q Developer

在入門索引標籤的 Amazon Q 區段中，您可以啟用或停用 Amazon Q、新增連線，或切換到不同的 AWS 連線。您必須先啟用 Amazon Q，才能檢視或存取任何這些動作。若要啟用 Amazon Q，請按一下啟用按鈕。

停用 Amazon Q 時，所有 Amazon Q 功能都會從 Visual Studio 中完全移除。啟用 Amazon Q 會在入門索引標籤中自動開啟 Amazon Q 的設定身分驗證。若要繼續，您必須使用 AWS IAM Identity Center 登入資料進行身分驗證，才能存取 Professional 方案或 AWS 您的 Builder ID 來存取免費方案。如需每個方案選項的詳細資訊，請參閱 [《Amazon Q 開發人員使用者指南》](#) 中的 [了解 Amazon Q 開發人員服務方案](#) 主題。

若要繼續，請完成下列其中一個程序。

使用 IAM Identity Center 進行專業層身分驗證

Note

使用 Professional 層進行身分驗證所需的設定檔名稱、開始 URL、設定檔區域或 SSO 區域欄位通常由您公司或組織的管理員提供。如需 IAM Identity Center 登入資料的詳細資訊，請參閱 [《IAM Identity Center 使用者指南》](#) 中的 [什麼是 AWS IAM Identity Center](#) 主題。

1. 從 Amazon Q 入門：AWS 工具組畫面，選擇 Amazon Q 圖磚中的登入按鈕，以導覽至 Amazon Q 的設定身分驗證畫面。
2. 從 Amazon Q 的設定身分驗證畫面中，導覽至 Professional 方案區段，填寫必要欄位，然後選擇連線按鈕。
3. 確認您想要在預設 Web 瀏覽器中開啟 AWS 授權請求入口網站。
4. 完成 AWS 授權請求入口網站所需的步驟，您會在安全關閉瀏覽器並返回 Visual Studio 時收到通知
5. 在入門索引標籤中，Amazon Q 會更新，以顯示您在程序完成時已與 IAM Identity Center 連線。

使用 AWS Builder ID 的免費方案身分驗證

Note

如需 AWS 建置器 ID 的其他詳細資訊，請參閱 [《登入使用者指南》中的使用 AWS 建置器 ID AWS 登入主題](#)。

1. 從 Amazon Q 入門：AWS 工具組畫面，選擇 Amazon Q 圖磚中的登入按鈕，以導覽至 Amazon Q 的設定身分驗證畫面。
2. 從 Amazon Q 的設定身分驗證畫面中，導覽至免費方案區段，然後選擇註冊或登入按鈕。
3. 確認您想要在預設 Web 瀏覽器中開啟 AWS 授權請求入口網站。
4. 完成 AWS 授權請求入口網站所需的步驟，您會在安全關閉瀏覽器並返回 Visual Studio 時收到通知。
5. 在入門索引標籤中，Amazon Q 會更新，以顯示您在程序完成時已使用您的 AWS 建置器 ID 連線。

使用 IAM Identity Center 或 AWS Builder ID 登入資料進行驗證後，您可以存取 Visual Studio 中的 Amazon Q。此外，您可以在入門索引標籤中執行下列動作：

- 登出：會中斷目前登入資料與所有 Amazon Q 函數的連線。Amazon Q 保持啟用狀態，但大多數功能無法運作。
- 停用 Amazon Q：完全停用 Visual Studio 中的所有 Amazon Q 功能。

AWS 工具組

從 AWS 工具組入門 AWS 索引標籤中的工具組區段中，您可以啟用或停用 AWS 工具組、新增連線或切換到不同的 AWS 連線。您必須先啟用 AWS Toolkit，才能檢視或存取任何這些動作。若要啟用 AWS Toolkit，請按一下啟用按鈕。

啟用 AWS Toolkit 時，Toolkit AWS 的設定身分驗證會在 AWS Toolkit 入門索引標籤中自動載入。若要繼續，您必須使用您的 AWS IAM Identity Center 登入資料或 IAM 使用者角色登入資料進行身分驗證。

Note

如需 IAM Identity Center 登入資料的詳細資訊，請參閱《[IAM Identity Center 使用者指南](#)》中的[什麼是 AWS IAM Identity Center](#) 主題。如需 IAM 使用者角色登入資料的詳細資訊，請參閱 AWS SDKs 和工具參考指南中的[AWS 存取金鑰：長期登入](#) 資料主題。

驗證並與 IAM Identity Center 連線

1. 從入門：AWS 工具組與 Amazon Q 畫面中，選擇 AWS 工具組圖磚中的登入按鈕，以導覽至工具 AWS 組設定身分驗證畫面。
2. 從設定 AWS Toolkit 身分驗證畫面，從設定檔類型下拉式選單中選擇 IAM Identity Center（單一登入的後續者）。
3. 從從現有的設定檔中選擇或新增新的下拉式功能表中，選擇現有的設定檔或選取新增設定檔以新增新的設定檔資訊。

Note

如果您選擇現有的設定檔，請前往步驟 7。

4. 在設定檔名稱欄位中，輸入與您要進行身分驗證的 IAM Identity Center 帳戶 **profile name** 相關聯的。
5. 在啟動 URL 文字欄位中，輸入連接至 IAM Identity Center 憑證 **Start URL** 的。
6. 從設定檔區域（預設為 us-east-1）下拉式功能表中，選擇您要驗證之 IAM Identity Center 使用者設定檔定義的設定檔區域。
7. 從 SSO 區域（預設為 us-east-1）下拉式功能表中，選擇 IAM Identity Center 憑證定義的 SSO 區域。
8. 選擇連線按鈕，在預設 Web 瀏覽器中開啟 AWS 授權請求網站。
9. 請遵循預設 Web 瀏覽器中的提示，在授權程序完成時通知您，關閉瀏覽器並返回 Visual Studio 是安全的。
10. 在入門索引標籤中 AWS，工具組區段會更新，以顯示您在程序完成時已與 IAM Identity Center 連線。

使用 IAM 使用者角色登入資料進行驗證和連線

1. 從入門：AWS 工具組與 Amazon Q 畫面中，選擇 AWS 工具組圖磚中的登入按鈕，以導覽至工具 AWS 組設定身分驗證畫面。
2. 從工具 AWS 組設定身分驗證畫面中，從設定檔類型下拉式選單中選擇 IAM 使用者角色。
3. 在從現有的設定檔中選擇或新增新的下拉式功能表中，選擇 **Add new profile**。

Note

如果您要從清單中選擇現有的設定檔名稱，請跳到步驟 8。

4. 在設定檔名稱文字欄位中，輸入新設定檔的名稱。
5. 在存取金鑰 ID 文字欄位中，輸入您要進行身分驗證之設定檔 **Access Key ID** 的。
6. 在私密金鑰文字欄位中，輸入您要進行身分驗證之設定檔 **Secret Key** 的。
7. 從儲存位置（預設為共用登入資料檔案）下拉式功能表中，指定您要使用共用登入資料檔案或 .NET 加密存放區來存放登入資料。
8. 從設定檔區域（預設為 us-east-1）下拉式功能表中，選擇連接至您要驗證之設定檔的分割區和設定檔區域。
9. 選擇連線按鈕，將此設定檔新增至您的 AWS 儲存位置和/或進行驗證 AWS。
10. 在入門索引標籤中 AWS，工具組區段會更新，以顯示您在程序完成時已與您的 IAM 使用者角色登入資料連線。

使用 IAM Identity Center 或 IAM 使用者角色登入資料進行身分驗證後，您可以在 Toolkit for Visual Studio 中存取 AWS Explorer。此外，您可以從入門索引標籤登出和停用 AWS Toolkit for Visual Studio with Amazon Q。

文件和教學課程

文件和教學課程區段會根據您的 AWS 服務和功能偏好設定，自動更新文件和教學課程建議。至少啟用一項功能時，才會顯示這些參考。

針對的安裝問題進行故障診斷 AWS Toolkit for Visual Studio

下列資訊已知可在設定時解決常見的安裝問題 AWS Toolkit for Visual Studio。

如果您在安裝時發生錯誤，AWS Toolkit for Visual Studio 或不確定安裝是否已完成，請檢閱下列各節中的資訊。

Visual Studio 的管理員許可

AWS Toolkit for Visual Studio 擴充功能需要管理員許可，以確保可存取所有 AWS 服務和功能。

如果您有本機管理員許可，您的管理員許可可能不會直接延伸至您的 Visual Studio 執行個體。

若要在本機啟動具有管理員許可的 Visual Studio：

1. 從 Windows 找到 Visual Studio 應用程式啟動器（圖示）。
2. 開啟 Visual Studio 圖示的內容選單（按一下滑鼠右鍵）以開啟內容選單。
3. 從內容功能表中選取以管理員身分執行。

若要從遠端啟動具有管理員許可的 Visual Studio：

1. 在 Windows 中，尋找您用來連線至 Visual Studio 遠端執行個體之應用程式的應用程式啟動器。
2. 開啟應用程式的內容選單（按一下滑鼠右鍵）以開啟內容選單。
3. 從內容功能表中選取以管理員身分執行。

Note

無論您是在本機啟動程式，還是從遠端連線，Windows 可能會提示您確認管理登入資料。

取得安裝日誌

如果您已完成上述管理員許可區段中的步驟，並確認您正在執行或使用管理員許可連線至 Visual Studio，則取得安裝日誌檔案有助於診斷其他問題。

若要 AWS Toolkit for Visual Studio 從 .vsix 檔案手動安裝並產生安裝日誌檔案，請完成下列步驟。

1. 在[AWS Toolkit for Visual Studio](#)登陸頁面中，遵循下載連結並儲存您要安裝的 AWS Toolkit for Visual Studio 版本 .vsix 檔案。
2. 從 Visual Studio 主功能表中，展開工具標頭，展開命令列子選單，然後選擇 Visual Studio 開發人員命令提示字元。
3. 從 Visual Studio 開發人員命令提示字元輸入下列格式的 vsixinstaller 命令：

```
vsixinstaller /logFile:[file path to log file] [file path to Toolkit installation file]
```

- 將取代[file path to log file]為您要建立安裝日誌之目錄的檔案名稱和完整檔案路徑。vsixinstaller 命令的範例與指定的檔案路徑和檔案名稱類似如下：

```
vsixinstaller /logfile:C:\Users\Documents\install-log.txt [file path to AWSToolkitPackage.vsix]
```

- [file path to Toolkit installation file] 將取代為 AWSToolkitPackage.vsix 所在目錄的完整檔案路徑。

具有 Toolkit 安裝檔案完整檔案路徑的 vsixinstaller 命令範例應類似下列：

```
vsixinstaller /logfile:[file path to log file] C:\Users\Downloads\AWSToolkitPackage.vsix
```

- 檢查您的檔案名稱和路徑是否正確，然後執行 vsixinstaller 命令。

完整 vsixinstaller 命令的範例如下所示：

```
vsixinstaller /logfile:C:\Users\Documents\install-log.txt C:\Users\Downloads\AWSToolkitPackage.vsix
```

安裝不同的 Visual Studio 延伸模組

如果您已經取得安裝日誌檔案，但仍無法判斷安裝程序失敗的原因，請檢查您是否能夠安裝其他 Visual Studio 延伸模組。安裝不同的 Visual Studio 延伸模組可以為您的安裝問題提供額外的洞見。如果您無法安裝任何 Visual Studio 延伸模組，則可能需要對 Visual Studio 的問題進行疑難排解，而不是進行疑難排解 AWS Toolkit for Visual Studio。

聯絡支援

如果您已檢閱本指南中包含的所有區段，並需要額外的資源或支援，則可以檢視過去的問題，或從 [AWS Toolkit for Visual Studio Github Issues](#) 網站開啟新問題。

為了協助加速問題的解決方案：

- 檢查過去和目前的問題，以查看其他人是否遇到類似的情況。
- 記下您為了解決問題所採取的每個步驟的詳細記錄。
- 儲存您從安裝 AWS Toolkit for Visual Studio 或其他擴充功能取得的任何日誌檔案。
- 將 AWS Toolkit for Visual Studio 安裝日誌檔連接至新問題。

設定檔和視窗繫結

Toolkit for Visual Studio 的設定檔和視窗繫結

使用 Toolkit for Visual Studio 的發佈工具、精靈和其他功能時，請注意下列事項：

- AWS Explorer 視窗會一次繫結至單一設定檔和區域。從 AWS Explorer 開啟的 Windows 預設為該繫結設定檔和區域。
- 開啟新視窗後，您可以使用 Explorer AWS 的執行個體切換到不同的設定檔或區域。
- Toolkit for Visual Studio 發佈工具和功能會自動預設為 AWS Explorer 中設定的設定檔和區域。
- 如果在發佈工具、精靈或功能中指定了新的設定檔或區域：之後建立的所有資源都會繼續使用新的設定檔和區域設定。
- 如果您有多個開啟的 Visual Studio 執行個體，則每個執行個體都可以繫結至不同的設定檔和區域。
- AWS Explorer 會儲存指定的最後一個設定檔和區域，且最後關閉的 Visual Studio 執行個體會保留其值。

身分驗證和存取

您不需要向進行身分驗證 AWS，即可開始使用 AWS Toolkit for Visual Studio 搭配 Amazon Q。不過，大多數 AWS 資源是透過 AWS 帳戶管理。若要使用 Amazon Q 服務和功能存取所有 AWS Toolkit for Visual Studio，您將需要至少 2 種類型的帳戶身分驗證：

1. 您 AWS 帳戶的 AWS Identity and Access Management (IAM) 或 AWS IAM Identity Center 身分驗證。大多數 AWS 的服務和資源是透過 IAM 和 IAM Identity Center 管理。
2. 對於某些 AWS 其他服務，AWS 建置器 ID 是選用的。

下列主題包含每種憑證類型和驗證方法的其他詳細資訊及設定指示。

主題

- [AWS 中的 IAM Identity Center 登入資料 AWS Toolkit for Visual Studio](#)
- [AWS IAM 登入資料](#)
- [AWS 建置器 ID](#)
- [Toolkit for Visual Studio 中的多重要素驗證 \(MFA\)](#)
- [設定外部登入資料](#)
- [更新防火牆和閘道以允許存取](#)

AWS 中的 IAM Identity Center 登入資料 AWS Toolkit for Visual Studio

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 和的詳細說明 AWS Toolkit for Visual Studio，請參閱 AWS SDKs和工具參考指南中的 [IAM Identity Center 身分驗證](#) 一節。

從 驗證 IAM Identity Center AWS Toolkit for Visual Studio

若要透過將 IAM Identity Center 設定檔新增至您的 `credentials`或 `config` 檔案 AWS Toolkit for Visual Studio，從 驗證 IAM Identity Center，請完成下列步驟。

1. 從您偏好的文字編輯器中，開啟存放在 `<home-directory>\.aws\credentials` 檔案中的 AWS 登入資料資訊。

- 從區段 `credentials file` 下的 `[default]`，新增名為 IAM Identity Center 設定檔的範本。以下是範例範本：

Important

在 `credential` 檔案中建立項目時，請勿使用字詞描述檔，因為會與 `credential` 檔案命名慣例產生衝突。

`profile_` 只有在 `config` 檔案中設定具名設定檔時，才包含字首。

```
[sso-user-1]
sso_start_url = https://example.com/start
sso_region = us-east-2
sso_account_id = 123456789011
sso_role_name = readOnly
region = us-west-2
```

- **sso_start_url**：指向組織 IAM Identity Center 使用者入口網站的 URL。
- **sso_region**：包含 IAM Identity Center 入口網站主機 AWS 的區域。這可以與稍後在預設 `region` 參數中指定的 AWS 區域不同。
- **sso_account_id**：包含 IAM 角色 AWS 的帳戶 ID，具有您要授予此 IAM Identity Center 使用者的許可。
- **sso_role_name**：IAM 角色的名稱，定義使用者使用此設定檔透過 IAM Identity Center 取得登入資料的許可。
- **region**：此 IAM Identity Center 使用者登入的預設 AWS 區域。

Note

您也可以 AWS CLI 執行 `aws configure sso` 命令，將啟用 IAM Identity Center 的設定檔新增至您的。執行此命令後，您會為 IAM Identity Center 啟動 URL (`sso_start_url`) 和託管 IAM Identity Center 目錄 AWS 的區域 (`region`) 提供值。

如需詳細資訊，請參閱 AWS Command Line Interface 《使用者指南》中的 [設定 AWS CLI 以使用 AWS 單一登入](#)。

使用 IAM Identity Center 登入

使用 IAM Identity Center 設定檔登入時，預設瀏覽器會啟動至 `sso_start_url` 指定的 `credential file`。您必須先驗證 IAM Identity Center 登入，才能存取中的 AWS 資源 AWS Toolkit for Visual Studio。如果您的登入資料過期，您必須重複連線程序，才能取得新的臨時登入資料。

AWS IAM 登入資料

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

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。

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（例如 AWS CLI）設定 IAM 登入 AWS 資料，則 AWS Toolkit for Visual Studio 會自動偵測這些登入資料。
- AWS 建議使用 AWS 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。反之，請搭配 等身分提供者使用聯合 AWS IAM Identity Center。如需詳細資訊，請參閱 [什麼是 IAM Identity Center？](#) AWS IAM Identity Center 《使用者指南》中的。

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 來驗證 AWS 您的帳戶。

Note

步驟 3：更新共用憑證是選用的。

如果您完成步驟 3，AWS Toolkit for Visual Studio 會自動從 偵測您的登入資料 `credentials file`。

如果您尚未完成步驟 3，會 AWS Toolkit for Visual Studio 逐步引導您建立的程序，`credentials file` 如從下方 建立 [登入資料檔案 AWS Toolkit for Visual Studio](#) 一節所述。

建立登入資料檔案

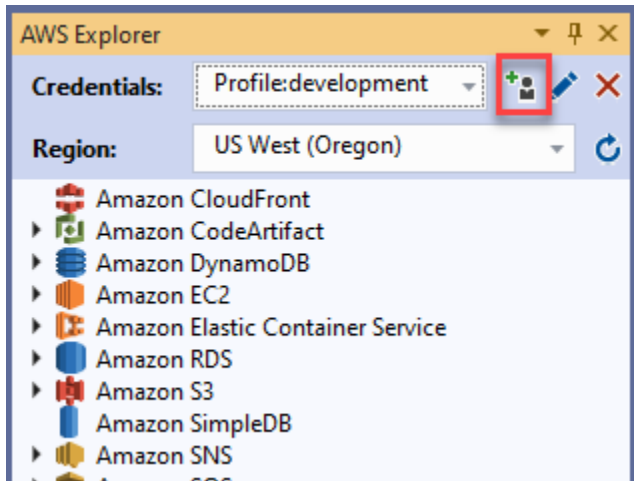
若要將使用者新增至 或從 建立 credentials file AWS Toolkit for Visual Studio :

Note

從工具組新增使用者設定檔時：

- 如果 credentials file 已存在，新的使用者資訊會新增至現有的 檔案。
- 如果 credentials file 不存在，則會建立新的檔案。

1. 從 AWS Explorer 選擇新帳戶設定檔圖示以開啟新帳戶設定檔對話方塊。



2. 完成新帳戶設定檔對話方塊中的必要欄位，然後選擇確定按鈕以建立 IAM 使用者。

從工具組編輯 IAM 使用者憑證

若要從工具組編輯 IAM 使用者憑證，請完成下列步驟：

1. 從 AWS Explorer 中的登入資料下拉式清單中，選擇您要編輯的 IAM 使用者登入資料。
2. 選擇編輯設定檔圖示以開啟編輯設定檔對話方塊。
3. 從編輯設定檔對話方塊中完成您的更新，然後選擇確定按鈕以儲存您的變更。

若要從工具組刪除 IAM 使用者憑證，請完成下列步驟：

1. 從 AWS Explorer 中的登入資料下拉式清單中，選擇您要刪除的 IAM 使用者登入資料。

2. 選擇刪除設定檔圖示以開啟刪除設定檔提示。
3. 確認您想要刪除設定檔，以將其從 中移除Credentials file。

Important

無法在編輯設定檔對話方塊中編輯支援進階存取功能的設定檔，例如 IAM Identity Center 或多重要素驗證 (MFA) AWS Toolkit for Visual Studio。若要變更這些類型的設定檔，您必須credentials file使用文字編輯器編輯。

從文字編輯器編輯 IAM 使用者憑證

除了使用 管理 IAM 使用者之外 AWS Toolkit for Visual Studio，您還可以credential files從偏好的文字編輯器進行編輯。Windows credential file中的預設位置為 C:\Users *USERNAME*\.aws\credentials。

如需 位置和結構的詳細資訊credential files，請參閱 AWS SDKs [和工具參考指南中的共用組態和登入資料檔案](#)一節。

從 AWS Command Line Interface (AWS CLI) 建立 IAM 使用者

AWS CLI 是您可以使用 命令 的另一個工具credentials file，在 中建立 IAM 使用者aws configure。

如需從 建立 IAM 使用者的詳細資訊，AWS CLI 請參閱AWS CLI 《使用者指南》中的[設定 AWS CLI](#)主題。

Toolkit for Visual Studio 支援下列組態屬性：

```
aws_access_key_id
aws_secret_access_key
aws_session_token
credential_process
credential_source
external_id
mfa_serial
role_arn
role_session_name
source_profile
sso_account_id
```

```
sso_region  
sso_role_name  
sso_start_url
```

AWS 建置器 ID

AWS Builder ID 是使用特定服務或功能時可能需要的額外 AWS 身分驗證方法，例如使用 Amazon CodeCatalyst 複製第三方儲存庫。

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。

如需從中複製 CodeCatalyst 儲存庫的詳細資訊 AWS Toolkit for Visual Studio，請參閱本使用者指南中的 [使用 Amazon CodeCatalyst](#) 主題。

Toolkit for Visual Studio 中的多重要素驗證 (MFA)

多重要素驗證 (MFA) 是 AWS 您的帳戶的額外安全性。MFA 要求使用者在存取 AWS 網站或服務時，從 AWS 支援的 MFA 機制提供登入憑證和唯一身分驗證。

AWS 支援 MFA 身分驗證的虛擬和硬體裝置範圍。以下是透過智慧型手機應用程式啟用的虛擬 MFA 裝置範例。如需 MFA 裝置選項的詳細資訊，請參閱 [《IAM 使用者指南》](#) 中的 [在中使用多重要素驗證 \(MFA\) AWS](#)。

步驟 1：建立 IAM 角色以將存取權委派給 IAM 使用者

下列程序說明如何設定將許可指派給 IAM 使用者的角色除錯。如需角色刪除的詳細資訊，請參閱 AWS Identity and Access Management [《使用者指南》](#) 中的 [建立角色以將許可委派給 IAM 使用者](#) 主題。

1. 前往 IAM 主控台，網址為 <https://console.aws.amazon.com/iam>。
2. 在導覽列中選擇角色，然後選擇建立角色。
3. 在建立角色頁面中，選擇另一個 AWS 帳戶。
4. 輸入所需的帳戶 ID 並標記需要 MFA 核取方塊。

Note

若要尋找 12 位數的帳號 (ID)，請前往主控台中的導覽列，然後選擇支援、支援中心。

5. 選擇下一步：許可。

6. 將現有政策連接至您的角色，或為其建立新政策。您在此頁面上選擇的政策決定 IAM 使用者可以使用 Toolkit 存取哪些 AWS 服務。
7. 連接政策後，選擇下一步：將 IAM 標籤新增至角色的選項標籤。然後選擇下一步：檢閱以繼續。
8. 在檢閱頁面中，輸入必要的角色名稱 (例如工具組角色)。您也可以新增選用的角色描述。
9. 選擇建立角色。
10. 當確認訊息顯示時 (例如「已建立角色工具組角色」)，請在訊息中選擇角色的名稱。
11. 在摘要頁面中，選擇複製圖示以複製角色 ARN，並將其貼到檔案中。(設定 IAM 使用者擔任角色時，您需要此 ARN。)

步驟 2：建立擔任角色許可的 IAM 使用者

此步驟會建立沒有許可的 IAM 使用者，以便新增內嵌政策。

1. 前往 IAM 主控台，網址為 <https://console.aws.amazon.com/iam>。
2. 在導覽列中選擇使用者，然後選擇新增使用者。
3. 在新增使用者頁面中，輸入所需的使用者名稱 (例如工具組使用者)，並標記程式設計存取核取方塊。
4. 選擇下一步：許可、下一步：標籤和下一步：檢閱以瀏覽下一頁。您在此階段不會新增許可，因為使用者將擔任角色的許可。
5. 在檢閱頁面中，您會收到此使用者沒有許可的通知。選擇 Create user (建立使用者)。
6. 在成功頁面中，選擇下載 .csv 以下載包含存取金鑰 ID 和私密存取金鑰的檔案。(在登入資料檔案中定義使用者的設定檔時，您需要兩者。)
7. 選擇關閉。

步驟 3：新增政策以允許 IAM 使用者擔任角色

下列程序會建立內嵌政策，允許使用者擔任角色 (以及該角色的許可)。

1. 在 IAM 主控台的使用者頁面中，選擇您剛建立的 IAM 使用者 (例如工具組使用者)。
2. 在摘要頁面的許可索引標籤中，選擇新增內嵌政策。
3. 在建立政策頁面中，選擇選擇服務，在尋找服務中輸入 STS，然後從結果中選擇 STS。
4. 針對動作，開始輸入 AssumeRole 一詞。出現時標記 AssumeRole 核取方塊。
5. 在資源區段中，確定已選取特定，然後按一下新增 ARN 以限制存取。
6. 在新增 ARN(s) 對話方塊中，針對為角色指定 ARN 新增您在步驟 1 中建立的角色 ARN。

新增角色的 ARN 之後，與該角色相關聯的受信任帳戶和角色名稱會顯示在具有路徑的帳戶和角色名稱中。

7. 選擇新增。
8. 返回建立政策頁面，選擇指定請求條件（選用）、標記 MFA 必要核取方塊，然後選擇關閉以確認。
9. 選擇 Review policy (檢閱政策)
10. 在檢閱政策頁面中，輸入政策的名稱，然後選擇建立政策。

許可索引標籤會顯示直接連接到 IAM 使用者的新內嵌政策。

步驟 4：管理 IAM 使用者的虛擬 MFA 裝置

1. 下載虛擬 MFA 應用程式並安裝到您的智慧型手機。

如需支援的應用程式清單，請參閱[多重要素驗證](#)資源頁面。

2. 在 IAM 主控台中，從導覽列中選擇使用者，然後選擇擔任角色的使用者（在此情況下為工具組使用者）。
3. 在摘要頁面中，選擇安全登入資料索引標籤，然後針對指派的 MFA 裝置選擇管理。
4. 在管理 MFA 裝置窗格中，選擇虛擬 MFA 裝置，然後選擇繼續。
5. 在設定虛擬 MFA 裝置窗格中，選擇顯示 QR 碼，然後使用您安裝在智慧型手機上的虛擬 MFA 應用程式掃描程式碼。
6. 掃描 QR 代碼後，虛擬 MFA 應用程式會產生一次性 MFA 代碼。在 MFA 代碼 1 和 MFA 代碼 2 中輸入兩個連續的 MFA 代碼。
7. 選擇 Assign MFA (指派 MFA)。
8. 返回使用者的安全登入資料索引標籤，複製新指派 MFA 裝置的 ARN。

ARN 包含您的 12 位數帳戶 ID，格式如下：`arn:aws:iam::123456789012:mfa/toolkit-user`。在下一個步驟中定義 MFA 設定檔時，您需要此 ARN。

步驟 5：建立設定檔以允許 MFA

下列程序會建立設定檔，允許 MFA 從 Toolkit for Visual Studio 存取 AWS 服務。

您建立的設定檔包含三個您在先前步驟中複製和儲存的資訊：

- IAM 使用者的存取金鑰（存取金鑰 ID 和私密存取金鑰）
- 將許可委派給 IAM 使用者之角色的 ARN
- 指派給 IAM 使用者的虛擬 MFA 裝置的 ARN

在包含您登入資料的 AWS 共用 AWS 登入資料檔案或 SDK 存放區中，新增下列項目：

```
[toolkit-user]
aws_access_key_id = AKIAIOSFODNN7EXAMPLE
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY

[mfa]
source_profile = toolkit-user
role_arn = arn:aws:iam::111111111111:role/toolkit-role
mfa_serial = arn:aws:iam::111111111111:mfa/toolkit-user
```

提供的範例中定義了兩個設定檔：

- [toolkit-user] 描述檔包含存取金鑰和私密存取金鑰，這些金鑰會在您在步驟 2 中建立 IAM 使用者時產生並儲存。
- [mfa] 描述檔會定義如何支援多重要素驗證。有三個項目：
 - `source_profile`：指定其登入資料用於擔任此設定檔中此 `role_arn` 設定所指定角色的設定檔。在這種情況下，它是 `toolkit-user` 描述檔。
 - `role_arn`：指定您要用來執行使用此設定檔請求之操作的 IAM 角色的 Amazon Resource Name (ARN)。在此情況下，這是您在步驟 1 中建立之角色的 ARN。
 - `mfa_serial`：指定使用者擔任角色時必須使用之 MFA 裝置的識別或序號。在這種情況下，這是您在步驟 3 中設定的虛擬裝置的 ARN。

設定外部登入資料

如果您有方法可以產生或查詢 未直接支援的登入資料 AWS，您可以將包含 `credential_process` 設定的設定檔新增至共用登入資料檔案。此設定指定執行的外部命令，以產生或擷取要使用的身分驗證憑證。例如，您可以在 `config` 檔案中包含類似下列的項目：

```
[profile developer]
credential_process = /opt/bin/awscreds-custom --username helen
```

如需使用外部登入資料和相關安全風險的詳細資訊，請參閱AWS Command Line Interface 《使用者指南》中的[使用外部程序採購登入](#)資料。

更新防火牆和閘道以允許存取

如果您使用 Web 內容篩選解決方案來篩選對特定 AWS 網域或 URL 端點的存取，則必須允許列出下列端點，才能存取和 Amazon Q AWS Toolkit for Visual Studio 提供的所有服務和功能。如需如何針對 AWS Toolkit with Amazon Q 的防火牆和代理設定進行故障診斷的詳細步驟，請參閱本使用者指南中的故障診斷主題中的[防火牆和代理設定](#)一節。如需為 Amazon Q 設定公司代理的詳細資訊，請參閱《Amazon Q 開發人員使用者指南》中的[在 Amazon Q 主題中設定公司代理](#)。

AWS Toolkit for Visual Studio 端點

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

端點

```
https://idetoolkits-hostedfiles.amazonaws.com/*
https://idetoolkits.amazonwebservices.com/*
http://vstoolkit.amazonwebservices.com/*
https://aws-vs-toolkit.s3.amazonaws.com/*
https://raw.githubusercontent.com/aws/aws-toolkit-visual-studio/main/version.json
https://aws-toolkit-language-servers.amazonaws.com/*
```

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 auto-complete 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
```

```
amazonwebservicess.gallery.vassets.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.amazonwebservicess.com
```

使用 AWS 服務

下列主題說明如何開始使用 AWS Toolkit for Visual Studio 搭配 Amazon Q AWS 的服務。

主題

- [Amazon CodeCatalyst for AWS Toolkit for Visual Studio with Amazon Q](#)
- [Visual Studio 的 Amazon CloudWatch Logs 整合](#)
- [管理 Amazon EC2 執行個體](#)
- [管理 Amazon ECS 執行個體](#)
- [從 AWS Explorer 管理安全群組](#)
- [從 Amazon EC2 執行個體建立 AMI](#)
- [在 Amazon Machine Image 上設定啟動許可](#)
- [Amazon Virtual Private Cloud \(VPC\)](#)
- [使用 Visual Studio 的 CloudFormation 範本編輯器](#)
- [從 AWS Explorer 使用 Amazon S3](#)
- [從 AWS Explorer 使用 DynamoDB](#)
- [AWS CodeCommit 搭配 Visual Studio Team Explorer 使用](#)
- [在 Visual Studio 中使用 CodeArtifact](#)
- [AWS Explorer 的 Amazon RDS](#)
- [從 AWS Explorer 使用 Amazon SimpleDB](#)
- [從 AWS Explorer 使用 Amazon SQS](#)
- [身分和存取權管理](#)
- [AWS Lambda](#)

Amazon CodeCatalyst for AWS Toolkit for Visual Studio with Amazon Q

什麼是 Amazon CodeCatalyst ?

Amazon CodeCatalyst 是提供給軟體開發團隊的雲端協作空間。使用 AWS Toolkit for Visual Studio 搭配 Amazon Q，您可以直接從 AWS Toolkit for Visual Studio 搭配 Amazon Q 檢視和管理 CodeCatalyst 資源。如需 CodeCatalyst 的詳細資訊，請參閱 [Amazon CodeCatalyst](#) 使用者指南。

下列主題說明如何透過 Amazon Q with CodeCatalyst 連接 AWS Toolkit for Visual Studio，以及如何透過 Toolkit for Visual Studio with Amazon Q 使用 AWS CodeCatalyst。

主題

- [Amazon CodeCatalyst 和 AWS Toolkit for Visual Studio with Amazon Q 入門](#)
- [使用 Toolkit AWS for Visual Studio 中的 Amazon CodeCatalyst 資源搭配 Amazon Q](#)
- [疑難排解](#)

Amazon CodeCatalyst 和 AWS Toolkit for Visual Studio with Amazon Q 入門

若要從 AWS Toolkit for Visual Studio 搭配 Amazon Q 開始使用 Amazon CodeCatalyst，請完成下列步驟。

主題

- [使用 Amazon Q 安裝 AWS Toolkit for Visual Studio](#)
- [建立 CodeCatalyst 帳戶和 AWS 建置器 ID](#)
- [使用 CodeCatalyst 將 AWS Toolkit for Visual Studio 與 Amazon Q 連線](#)

使用 Amazon Q 安裝 AWS Toolkit for Visual Studio

將 AWS Toolkit for Visual Studio 與 CodeCatalyst 帳戶整合之前，請確定您使用的是最新版本的 AWS Toolkit for Visual Studio 與 Amazon Q。如需如何安裝和設定最新版本的 AWS Toolkit for Visual Studio 與 Amazon Q 的詳細資訊，請參閱本使用者指南中的[設定 AWS Toolkit for Visual Studio 與 Amazon Q](#) 一節。

建立 CodeCatalyst 帳戶和 AWS 建置器 ID

除了安裝最新版本的 AWS Toolkit for Visual Studio 搭配 Amazon Q 之外，您還必須擁有作用中的 AWS 建置器 ID 和 CodeCatalyst 帳戶，才能透過 Amazon Q 與 AWS Toolkit for Visual Studio 連線。如果您沒有作用中的 AWS 建置器 ID 或 CodeCatalyst 帳戶，請參閱 [CodeCatalyst 使用者指南中的使用 CodeCatalyst 設定](#) 一節。CodeCatalyst

Note

AWS Builder ID 與您的 AWS 登入資料不同。如需如何使用 AWS 建置器 ID 註冊和驗證的說明，請參閱本使用者指南中的[身分驗證和存取：AWS 建置器 ID](#) 主題。

如需 AWS 建置器 IDs 的詳細資訊，請參閱《AWS 一般參考使用者指南》中的 [AWS 建置器 ID](#) 主題。

使用 CodeCatalyst 將 AWS Toolkit for Visual Studio 與 Amazon Q 連線

若要將 AWS Toolkit for Visual Studio 與 Amazon Q 與您的 CodeCatalyst 帳戶連線，請完成下列步驟。

1. 從 Visual Studio 中的 Git 功能表項目中，選擇複製儲存庫...
2. 在瀏覽儲存庫區段中，選取 Amazon CodeCatalyst 做為供應商。
3. 從連線區段中，選擇使用 AWS 建置器 ID 連線，在偏好的 Web 瀏覽器中開啟 CodeCatalyst 主控台。
4. 從您的瀏覽器，在提供的欄位中輸入您的 AWS 建置器 ID，然後依照指示繼續。
5. 出現提示時，選擇允許以確認 AWS Toolkit for Visual Studio 與 Amazon Q 和 CodeCatalyst 帳戶之間的連線。連線程序完成後，CodeCatalyst 會顯示確認訊息，說明您可以安全關閉瀏覽器。

使用 Toolkit AWS for Visual Studio 中的 Amazon CodeCatalyst 資源搭配 Amazon Q

下列各節提供適用於 AWS Toolkit for Visual Studio with Amazon Q 的 Amazon CodeCatalyst 資源管理功能的概觀。

主題

- [複製儲存庫](#)

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。若要在本機處理專案，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 CodeCatalyst 專案同步。

若要將儲存庫複製到本機電腦，請完成下列步驟。

1. 從 Visual Studio 中的 Git 功能表項目中，選擇複製儲存庫...
2. 在瀏覽儲存庫區段中，選取 Amazon CodeCatalyst 做為供應商。

Note

如果連線區段顯示 Not Connected 訊息，請完成本使用者指南的 [身分驗證和存取：AWS Builder ID](#) 區段中的步驟，然後再繼續。

3. 選擇您要從中複製儲存庫的空間和專案。
4. 從儲存庫區段中，選擇您要複製的儲存庫。
5. 從路徑區段中，選擇您要複製儲存庫的資料夾。

Note

此資料夾一開始必須是空的，才能成功複製。

6. 選取複製以開始複製儲存庫。
7. 複製儲存庫之後，Visual Studio 會載入您複製的解決方案

Note

如果 Visual Studio 未在複製的儲存庫中開啟解決方案，您可以從來源控制選單的 Git 全域設定中開啟 Git 儲存庫設定時自動載入解決方案來調整 Visual Studio 選項。

疑難排解

以下是使用 AWS Toolkit for Visual Studio 搭配 Amazon Q 來處理 Amazon CodeCatalyst 時已知問題的疑難排解主題。

主題

- [憑證](#)

憑證

如果您在嘗試從 CodeCatalyst 複製 git 型儲存庫時遇到要求登入資料的對話方塊，您的 AWS CodeCommit 登入資料協助程式可能會全域設定，進而干擾 CodeCatalyst。如需 AWS CodeCommit 登入資料協助程式的詳細資訊，請參閱《[AWS CodeCommit 使用者指南](#)》中的 [在 Windows 上使用 AWS CLI 登入資料協助程式設定 HTTPS 連線的 CodeCommit 儲存庫步驟](#) 一節。AWS CodeCommit

若要限制 AWS CodeCommit 登入資料協助程式僅處理 CodeCommit URLs，請完成下列步驟。

1. 在中開啟全域 git 組態檔案：`%userprofile%\.gitconfig`
2. 在檔案中找到下列區段：

```
[credential]
  helper = !aws codecommit credential-helper $@
  UseHttpPath = true
```

3. 將該區段變更為以下內容：

```
[credential "https://git-codecommit.*.amazonaws.com"]
  helper = !aws codecommit credential-helper $@
  UseHttpPath = true
```

4. 儲存變更，然後完成複製儲存庫的步驟。

Visual Studio 的 Amazon CloudWatch Logs 整合

來自 AWS Toolkit for Visual Studio 與 Amazon Q 的 Amazon CloudWatch Logs 整合可讓您監控、存放和存取 CloudWatch Logs 資源，而不必離開 IDE。若要進一步了解如何設定 CloudWatch 服務以及如何使用 CloudWatch Logs 功能，請從下列主題中選擇。

主題

- [設定 Visual Studio 的 CloudWatch Logs 整合](#)
- [在 Visual Studio 中使用 CloudWatch Logs](#)

設定 Visual Studio 的 CloudWatch Logs 整合

您需要有 AWS 帳戶，才能使用 Amazon CloudWatch Logs 與 AWS Toolkit with Amazon Q 整合。您可以從[AWS 登入](#)網站建立新 AWS 帳戶。大多數可從 AWS Toolkit with Amazon Q 取得的 CloudWatch Logs 功能都可以使用作用中的 AWS 登入資料存取。如果特定功能需要額外的組態，這些要求會包含在[使用 CloudWatch Logs](#) 指南的相關區段中。

如需設定 CloudWatch Logs 的其他資訊和選項，請參閱《Amazon CloudWatch Logs 指南》中的[開始設定](#)一節。

在 Visual Studio 中使用 CloudWatch Logs

Amazon CloudWatch Logs 整合可讓您透過 Toolkit for Visual Studio 搭配 Amazon Q 監控、存放和存取 CloudWatch Logs。無需離開 IDE 即可存取 CloudWatch Logs 功能，透過簡化 CloudWatch Logs 開發程序並減少對工作流程的中斷來提高效率。AWS 下列主題說明如何使用 CloudWatch Logs 整合的基本功能。

主題

- [CloudWatch 日誌群組](#)
- [CloudWatch Log Streams](#)
- [CloudWatch Log Events](#)
- [對 CloudWatch Logs 的其他存取](#)

CloudWatch 日誌群組

log group 是 log streams 共用相同保留、監控和存取控制設定的 群組。可以屬於一個日誌群組的日誌串流數量並沒有限制。

檢視日誌群組

View Log Groups 此功能會顯示 CloudWatch Log Groups Explorer 中的日誌群組清單。

若要存取檢視日誌群組功能並開啟 CloudWatch Log Groups Explorer，請完成下列步驟。

1. 從 AWS Explorer 展開 Amazon CloudWatch。
2. 按兩下日誌群組或開啟內容功能表（按一下滑鼠右鍵），然後選取檢視，以開啟 CloudWatch Log Groups Explorer。

Note

CloudWatch Log Groups Explorer 會在與 Solutions Explorer 相同的視窗位置開啟。

篩選日誌群組

您的個別帳戶可以包含數千個不同的日誌群組。若要簡化特定群組的搜尋，請使用下列filtering功能。

1. 從 CloudWatch Log Groups Explorer，在視窗頂端的搜尋列中設定游標。
2. 開始輸入與您要尋找的日誌群組相關的字首。
3. CloudWatch Log Groups Explorer 會自動更新，以顯示符合您在上一個步驟中指定的搜尋詞彙的結果。

刪除日誌群組

若要刪除特定日誌群組，請參閱下列程序。

1. 從 CloudWatch Log Groups Explorer，在要刪除的日誌群組上按一下滑鼠右鍵。
2. 出現提示時，請確認您想要刪除目前選取的日誌群組。
3. 選擇是按鈕會刪除選取的日誌群組，然後重新整理 CloudWatch Log Groups Explorer。

重新整理日誌群組

若要重新整理 CloudWatch Log Groups Explorer 中顯示的目前日誌群組清單，請選擇工具列中的重新整理圖示按鈕。

複製日誌群組 ARN

若要複製特定日誌群組的 ARN，請完成下列步驟。

1. 從 CloudWatch Log Groups Explorer，在您要從中複製 ARN 的日誌群組上按一下滑鼠右鍵。
2. 從功能表中選擇複製 ARN 選項。
3. ARN 現在會複製到本機剪貼簿並準備好貼上。

CloudWatch Log Streams

日誌串流是共享相同來源的一系列日誌事件。

Note

檢視日誌串流時，請注意下列屬性：

- 根據預設，日誌串流會依最新的事件時間戳記排序。
- 與日誌串流相關聯的資料欄可以依遞增或遞減順序排序，方法是切換位於資料欄標頭中的插入。
- 篩選的項目只能依日誌串流名稱排序。

檢視日誌串流

1. 從 CloudWatch Log Groups Explorer 按兩下日誌群組，或在日誌群組上按一下滑鼠右鍵，然後從內容功能表中選取檢視日誌串流。
2. 新的索引標籤會在文件視窗中開啟，其中包含與您的日誌群組相關聯的日誌串流清單。

篩選日誌串流

1. 從日誌串流索引標籤的文件視窗中，在搜尋列中設定游標。
2. 開始輸入與您要尋找的日誌串流相關的字首。
3. 當您輸入時，目前的顯示會自動更新，以依您的輸入篩選日誌串流。

重新整理日誌串流

若要重新整理文件視窗中顯示的目前日誌串流清單，請選擇搜尋列旁邊的工具列中的重新整理圖示按鈕。

複製日誌串流 ARN

若要複製特定日誌串流的 ARN，請完成下列步驟。

1. 從日誌串流索引標籤的文件視窗中，在您要從中複製 ARN 的日誌串流上按一下滑鼠右鍵。
2. 從功能表中選擇複製 ARN 選項。
3. ARN 現在會複製到本機剪貼簿並準備好貼上。

下載日誌串流

匯出日誌串流功能會在本機下載和存放選取的日誌串流，以供自訂工具和軟體存取以進行其他處理。

1. 從日誌串流索引標籤的文件視窗中，在您要下載的日誌串流上按一下滑鼠右鍵。

2. 選擇匯出日誌串流以開啟匯出至文字檔案對話方塊。
3. 選擇您要在本機存放檔案的位置，並在提供的文字欄位中指定名稱。
4. 選取確定以確認下載。下載的狀態會顯示在 Visual Studio 任務狀態中心

CloudWatch Log Events

日誌事件是由 CloudWatch 監控的應用程式或資源記錄的活動記錄。

記錄事件動作

日誌事件會顯示為資料表。根據預設，事件會從最舊的事件排序為最新的事件。

下列動作與 Visual Studio 中的日誌事件相關聯：

- 包裝文字模式：您可以按一下事件來切換包裝文字。
- 文字包裝按鈕：位於 `document window toolbar`，此按鈕會切換所有項目的文字包裝開啟和關閉。
- 將訊息複製到剪貼簿：選取您要複製的訊息，然後在選取項目上按一下滑鼠右鍵，然後選擇複製（鍵盤快速鍵 `Ctrl + C`）。

檢視日誌事件

1. 從文件視窗中，選擇包含日誌串流清單的標籤。
2. 按兩下日誌串流，或在日誌串流上按一下滑鼠右鍵，然後從功能表中選取檢視日誌串流。
3. 新的日誌事件索引標籤會在文件視窗中開啟，其中包含與所選日誌串流相關聯的日誌事件資料表。

篩選日誌事件

您可以透過三種方式篩選日誌事件：依內容、時間範圍或兩者。若要依內容和時間範圍篩選日誌事件，請先依內容或時間範圍篩選您的訊息，然後依其他方法篩選這些結果。

若要依內容篩選日誌事件：

1. 從日誌事件索引標籤的文件視窗中，在視窗頂端的搜尋列中設定游標。
2. 開始輸入與您正在搜尋的日誌事件相關的詞彙或片語。
3. 當您輸入時，目前的顯示會自動開始篩選您的日誌事件。

Note

篩選條件模式區分大小寫。您可以用雙引號 (****) 括住具有非英數字元的確切詞彙和片語，以改善搜尋結果。如需篩選模式的詳細資訊，請參閱《Amazon CloudWatch 指南》中的[篩選和模式語法](#)主題。

若要檢視在特定時間範圍內產生的日誌事件：

1. 從日誌事件索引標籤的文件視窗中，選擇工具列中的行事曆圖示按鈕。
2. 使用提供的欄位，指定您要搜尋的時間範圍。
3. 當您指定日期和時間限制時，篩選的結果會自動更新。

Note

清除篩選條件選項會清除目前所有的date-and-time篩選條件選擇。

重新整理日誌事件

若要重新整理日誌事件索引標籤中顯示的目前日誌事件清單，請選擇工具列中的重新整理圖示按鈕。

對 CloudWatch Logs 的其他存取

您可以直接從 Visual Studio 中的 AWS Toolkit 存取與其他 AWS 服務和資源相關聯的 CloudWatch Logs。

Lambda

若要檢視與 Lambda 函數相關聯的日誌串流：

Note

您的 Lambda 執行角色必須具有適當的許可，才能將日誌傳送至 CloudWatch Logs。如需 CloudWatch Logs 所需 Lambda 許可的詳細資訊，請參閱 <https://docs.aws.amazon.com/lambda/latest/dg/monitoring-cloudwatchlogs.html#monitoring-cloudwatchlogs-prereqs>

1. 從 AWS Toolkit Explorer 中，展開 Lambda。

2. 在您要檢視的函數上按一下滑鼠右鍵，然後選擇檢視日誌以在文件視窗中開啟相關聯的日誌串流。

若要使用 Lambda 整合 檢視日誌串流 `function view`：

1. 從 AWS Toolkit Explorer 中，展開 Lambda。
2. 在您要檢視的函數上按一下滑鼠右鍵，然後選擇檢視函數以在文件視窗中開啟函數檢視。
3. 從 `function view` 切換到日誌索引標籤，隨即顯示與所選 Lambda 函數相關聯的日誌串流。

ECS

若要檢視與 ECS 任務容器相關聯的日誌資源，請完成下列程序。

Note

為了讓 Amazon ECS 服務將日誌傳送至 CloudWatch，特定 Amazon ECS 任務的每個容器必須符合必要的組態。如需所需設定和組態的詳細資訊，請參閱[使用 AWS 日誌日誌驅動程式指南](#)。

1. 從 AWS Toolkit Explorer 中，展開 Amazon ECS。
2. 選擇您要檢視的 Amazon ECS 叢集，以在文件視窗中開啟新的 ECS 叢集索引標籤。
3. 從位於 ECS 叢集標籤左側的導覽選單中，選擇任務以列出與叢集相關聯的所有任務。
4. 從任務顯示中，選取任務，然後選擇位於左下角的檢視日誌連結。

Note

此顯示會列出叢集中包含的所有任務，只有符合所需日誌組態的每個任務才會顯示 View Logs 連結。

- 如果任務僅與單一容器相關聯，則檢視日誌連結會開啟該容器的日誌串流。
- 如果任務與多個容器相關聯，檢視日誌連結會開啟檢視 ECS 任務的 CloudWatch Logs 對話方塊，使用容器：下拉式功能表選擇您要檢視日誌的容器，然後選擇確定。

5. 新的索引標籤會在文件視窗中開啟，顯示與您容器選擇相關聯的日誌串流。

管理 Amazon EC2 執行個體

AWS Explorer 提供 Amazon Machine Image (AMI) 和 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體的詳細檢視。您可以從這些檢視中，從 AMI 啟動 Amazon EC2 執行個體、連線至該執行個體，以及停止或終止執行個體，全都從 Visual Studio 開發環境內部執行。您可以使用執行個體檢視從執行個體建立 AMIs。如需詳細資訊，請參閱 [從 Amazon EC2 執行個體建立 AMI](#)。

Amazon Machine Image 和 Amazon EC2 執行個體檢視

從 AWS Explorer 中，您可以顯示 Amazon Machine Image (AMIs) 和 Amazon EC2 執行個體的檢視。在 AWS Explorer 中，展開 Amazon EC2 節點。

若要顯示 AMIs 檢視，請在第一個子節點 AMIs 上開啟內容（按一下滑鼠右鍵）選單，然後選擇檢視。

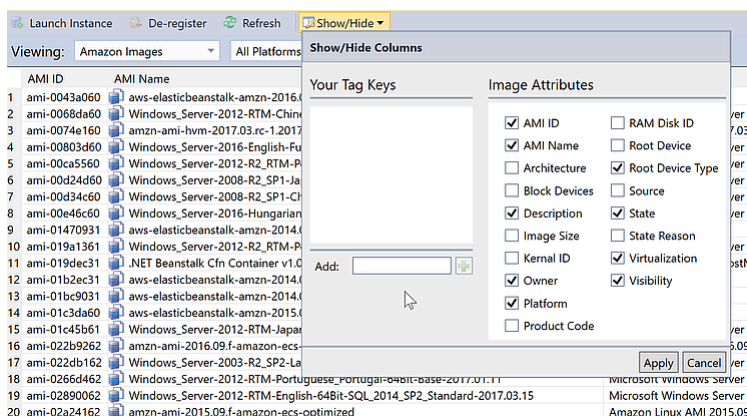
若要顯示 Amazon EC2 執行個體檢視，請在執行個體節點上開啟內容（按一下滑鼠右鍵）選單，然後選擇檢視。

您也可以按兩下適當的節點來顯示任一檢視。

- 檢視的範圍是 Explorer 中指定的區域 AWS（例如，美國西部（加利佛尼亞北部）區域）。
- 您可以按一下並拖曳來重新排列資料欄。若要排序資料欄中的值，請按一下資料欄標題。
- 您可以使用檢視中的下拉式清單和篩選條件方塊來設定檢視。初始檢視會顯示 AWS Explorer 中指定帳戶所擁有之任何平台類型的 AMIs (Windows 或 Linux)。

顯示/隱藏資料欄

您也可以選擇檢視頂端的顯示/隱藏下拉式清單，以設定要顯示哪些資料欄。如果您關閉檢視並重新開啟，您選擇的資料欄將會持續存在。



顯示/隱藏 AMI 和執行個體檢視的資料欄 UI

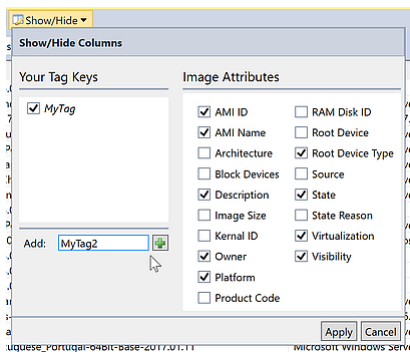
標記 AMIs、執行個體和磁碟區

您也可以使用顯示/隱藏下拉式清單，為您擁有 AMIs、Amazon EC2 執行個體或磁碟區新增標籤。標籤是名稱值對，可讓您將中繼資料連接至 AMIs、執行個體和磁碟區。標籤名稱的範圍涵蓋在您的帳戶，以及您的 AMIs 和執行個體。例如，如果您為 AMIs 和執行個體使用相同的標籤名稱，則不會發生衝突。標籤名稱不區分大小寫。

如需標籤的詳細資訊，請參閱《Amazon EC2 Linux 執行個體使用者指南》中的[使用標籤](#)。

新增標籤

1. 在新增方塊中，輸入標籤的名稱。選擇綠色按鈕與加號 (+)，然後選擇套用。



將標籤新增至 AMI 或 Amazon EC2 執行個體

新的標籤會以斜體顯示，這表示尚未有與該標籤相關聯的值。

在清單檢視中，標籤名稱會顯示為新資料欄。當至少有一個值與標籤相關聯時，標籤會顯示在 [AWS 管理主控台](#)。

2. 若要新增標籤的值，請在該標籤的欄中按兩下儲存格，然後輸入值。若要刪除標籤值，請按兩下儲存格並刪除文字。

如果您清除顯示/隱藏下拉式清單中的標籤，對應的欄會從檢視中消失。標籤會保留，以及與 AMIs、執行個體或磁碟區相關聯的任何標籤值。

Note

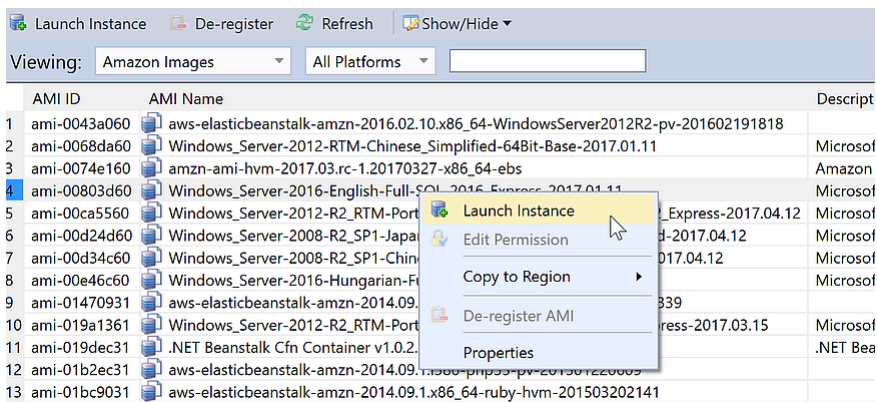
如果您在顯示/隱藏下拉式清單中清除沒有關聯值的標籤，AWS 工具組會完全刪除該標籤。它不會再出現在清單檢視或顯示/隱藏下拉式清單中。若要再次使用該標籤，請使用顯示/隱藏對話方塊來重新建立該標籤。

啟動 Amazon EC2 執行個體

AWS Explorer 提供啟動 Amazon EC2 執行個體所需的所有功能。在本節中，我們將選取 Amazon Machine Image (AMI)、進行設定，然後將其啟動為 Amazon EC2 執行個體。

啟動 Windows Server Amazon EC2 執行個體

1. 在 AMIs 頂端的下拉式清單中，選擇 Amazon Images。在右側的下拉式清單中，選擇 Windows。在篩選方塊中，輸入 Elastic Block Storage ebs 的。檢視重新整理可能需要一些時間。
2. 在清單中選擇 AMI，開啟內容（按一下滑鼠右鍵）選單，然後選擇啟動執行個體。



AMI 清單

3. 在啟動新的 Amazon EC2 執行個體對話方塊中，為您的應用程式設定 AMI。

執行個體類型

選擇要啟動的 EC2 執行個體類型。如需執行個體類型清單和定價資訊，請參閱 [EC2 定價](#) 頁面。

名稱

輸入執行個體的名稱。此名稱不能超過 256 個字元。

金鑰對

金鑰對用於取得您用來使用遠端桌面通訊協定 (RDP) 登入 EC2 執行個體的 Windows 密碼。選擇您可以存取私有金鑰的金鑰對，或選擇建立金鑰對的選項。如果您在 Toolkit 中建立金鑰對，Toolkit 可以為您存放私有金鑰。

存放在 Toolkit 中的金鑰對會加密。您可以在 %LOCALAPPDATA%\AWSToolkit\keypairs (通常為 : C:\Users\\AppData\Local\AWSToolkit\keypairs) 找到這些金鑰對。您可以將加密的金鑰對匯出至 .pem 檔案。

- a. 在 Visual Studio 中，選取檢視並按一下 AWS Explorer。

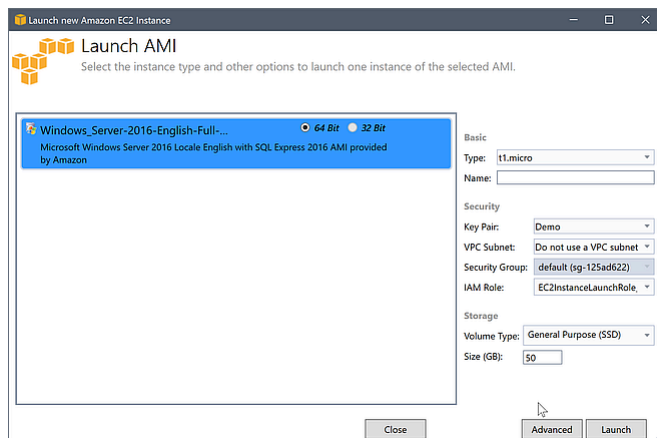
- b. 按一下 Amazon EC2，然後選取金鑰對。
- c. 系統會列出金鑰對，以及由標記為儲存在 AWSToolkit 中的 Toolkit 建立/管理的金鑰對。
- d. 用滑鼠右鍵按一下您建立的金鑰對，然後選取匯出私有金鑰。私有金鑰將未加密並存放在您指定的位置。

安全群組

安全群組控制 EC2 執行個體將接受的網路流量類型。選擇將允許連接埠 3389 上傳入流量的安全群組，這是 RDP 使用的連接埠，因此您可以連線至 EC2 執行個體。如需有關如何使用 Toolkit 建立安全群組的資訊，請參閱[從 AWS Explorer 管理安全群組](#)。

執行個體描述檔

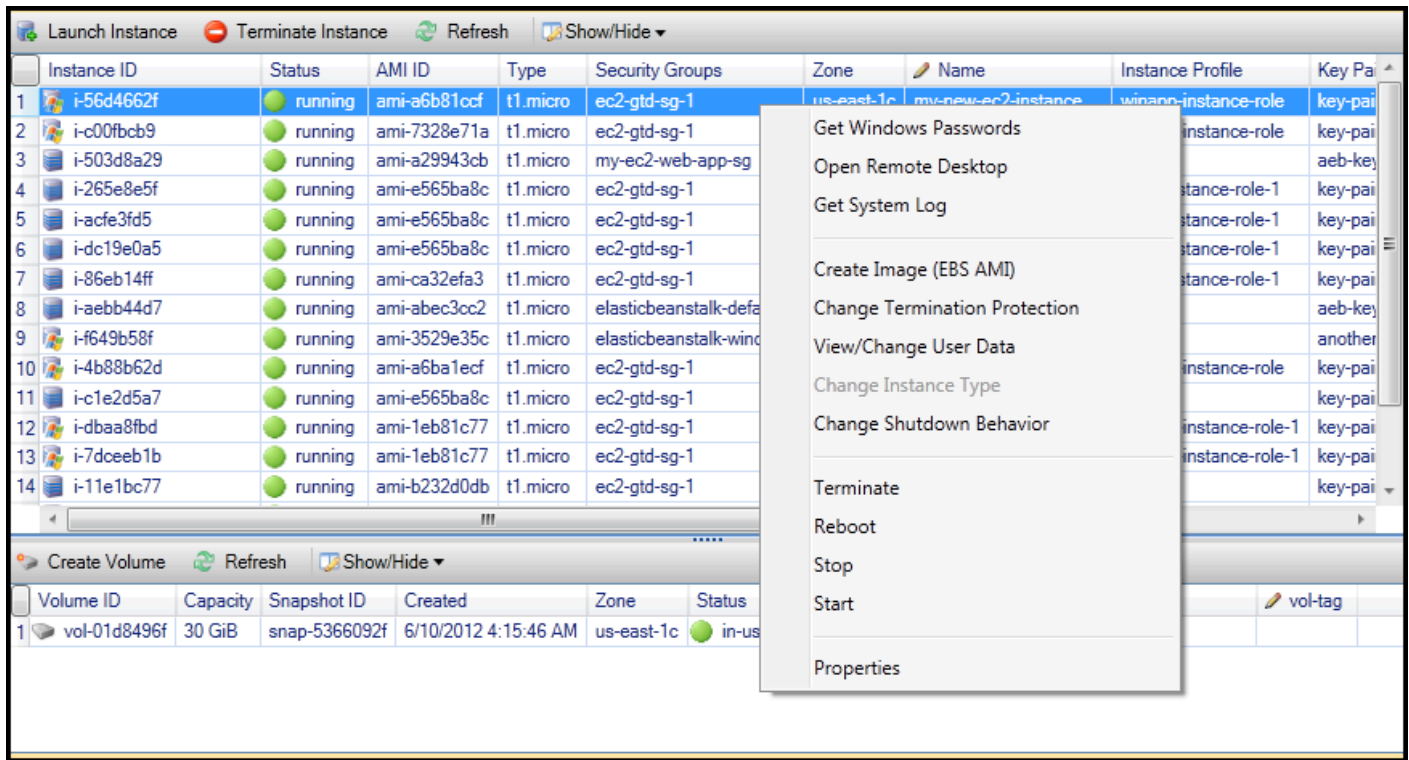
執行個體描述檔是 IAM 角色的邏輯容器。當您選擇執行個體描述檔時，您可以將對應的 IAM 角色與 EC2 執行個體建立關聯。IAM 角色設定的政策會指定對 Amazon Web Services 和帳戶資源的存取。一個 EC2 執行個體與 IAM 角色關聯時，在執行個體上執行的應用程式軟體將以 IAM 角色指定的許可執行。這可讓應用程式軟體無需指定自己的任何 AWS 登入資料即可執行，這可讓軟體更安全。如需 IAM 角色的詳細資訊，請參閱[IAM 使用者指南](#)。



EC2 啟動 AMI 對話方塊

4. 選擇啟動。

在 AWS Explorer 中，在 Amazon EC2 的執行個體子節點上，開啟內容（按一下滑鼠右鍵）選單，然後選擇檢視。Amazon EC2 Toolkit AWS 會顯示與作用中帳戶相關聯的 Amazon EC2 執行個體清單。您可能需要選擇重新整理，才能查看您的新執行個體。當執行個體第一次出現時，它可能處於待定狀態，但在幾分鐘後，它就會轉換為執行中狀態。



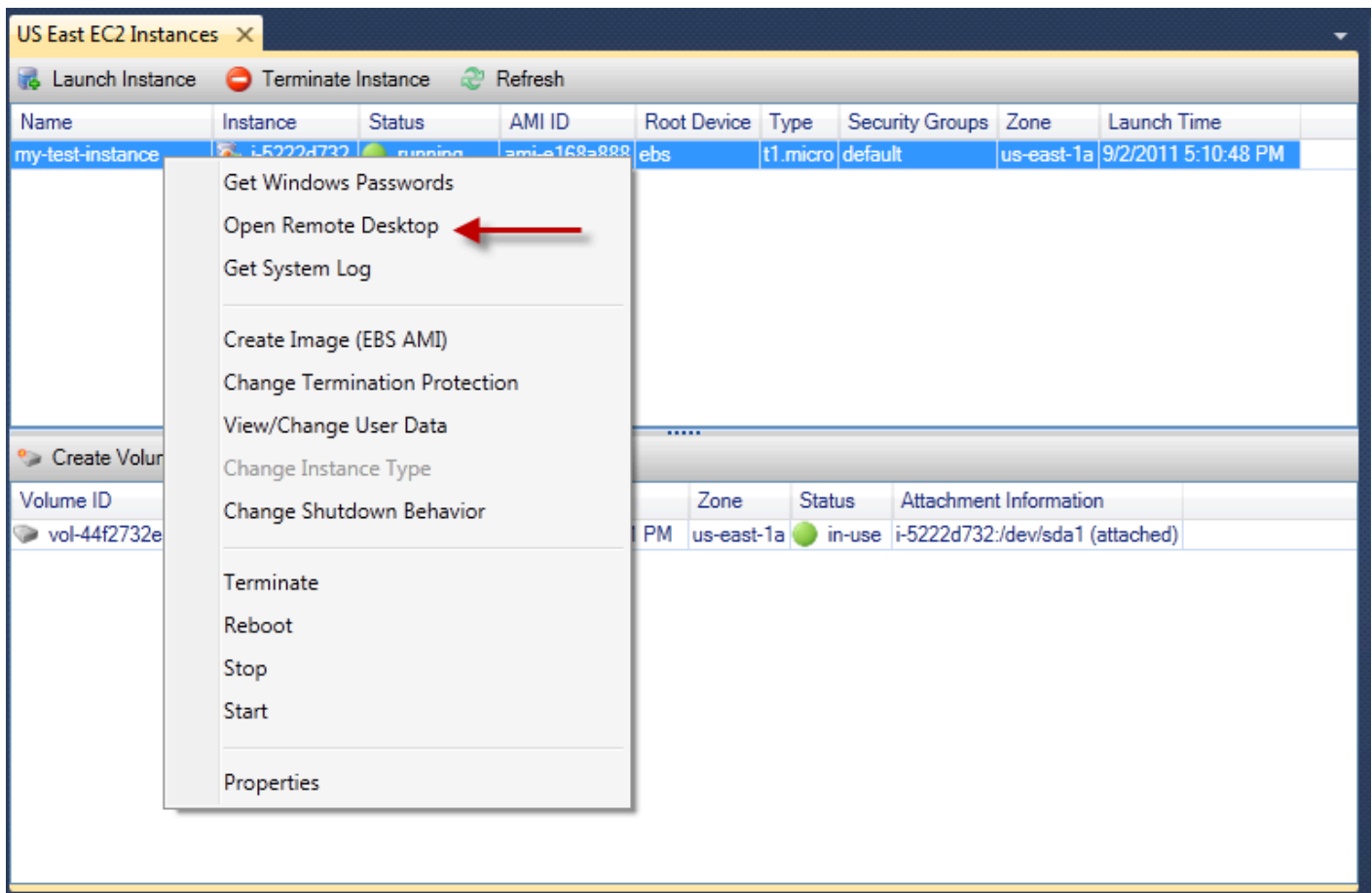
連線至 Amazon EC2 執行個體

您可以使用 Windows 遠端桌面連線至 Windows Server 執行個體。對於身分驗證，AWS Toolkit 可讓您擷取執行個體的管理員密碼，或者您可以直接使用與執行個體相關聯的已儲存金鑰對。在下列程序中，我們將使用儲存的金鑰對。

使用 Windows 遠端桌面連線至 Windows Server 執行個體

1. 在 EC2 執行個體清單中，在您要連線的 Windows Server 執行個體上按一下滑鼠右鍵。從內容功能表中，選擇開啟遠端桌面。

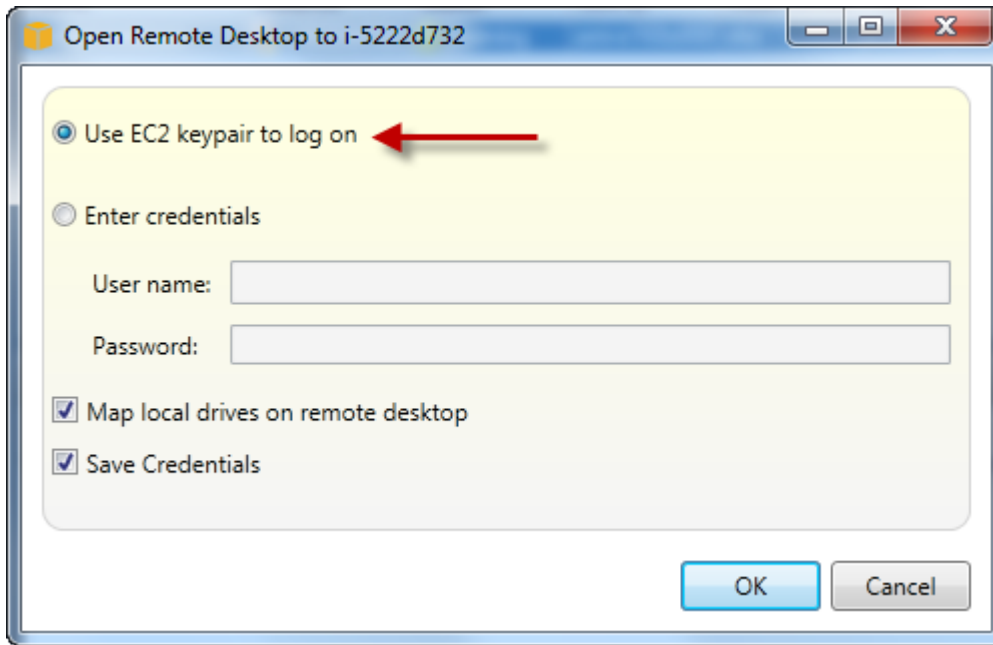
如果您想要使用管理員密碼進行身分驗證，您可以選擇取得 Windows 密碼。



EC2 執行個體內容功能表

2. 在開啟遠端桌面對話方塊中，選擇使用 EC2 金鑰對登入，然後選擇確定。

如果您未將金鑰對與 AWS Toolkit 一起存放，請指定包含私有金鑰的 PEM 檔案。

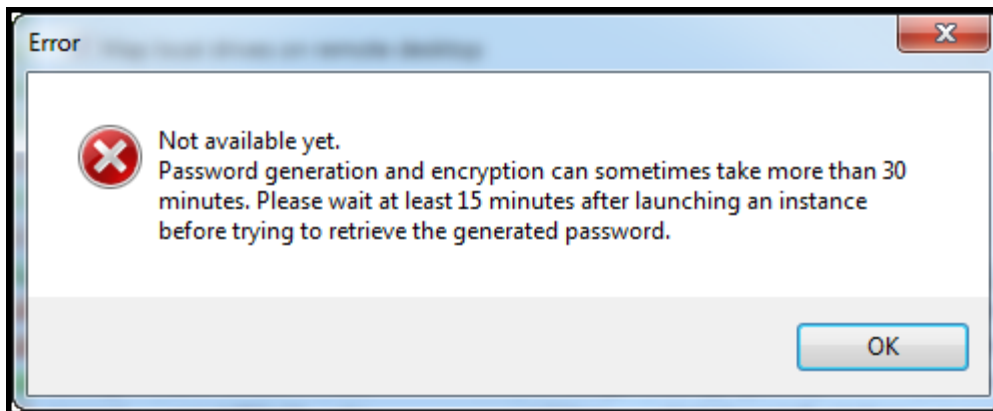


開啟遠端桌面對話方塊

3. 遠端桌面視窗隨即開啟。您不需要登入，因為使用金鑰對進行身分驗證。您將以 Amazon EC2 執行個體上的管理員身分執行。

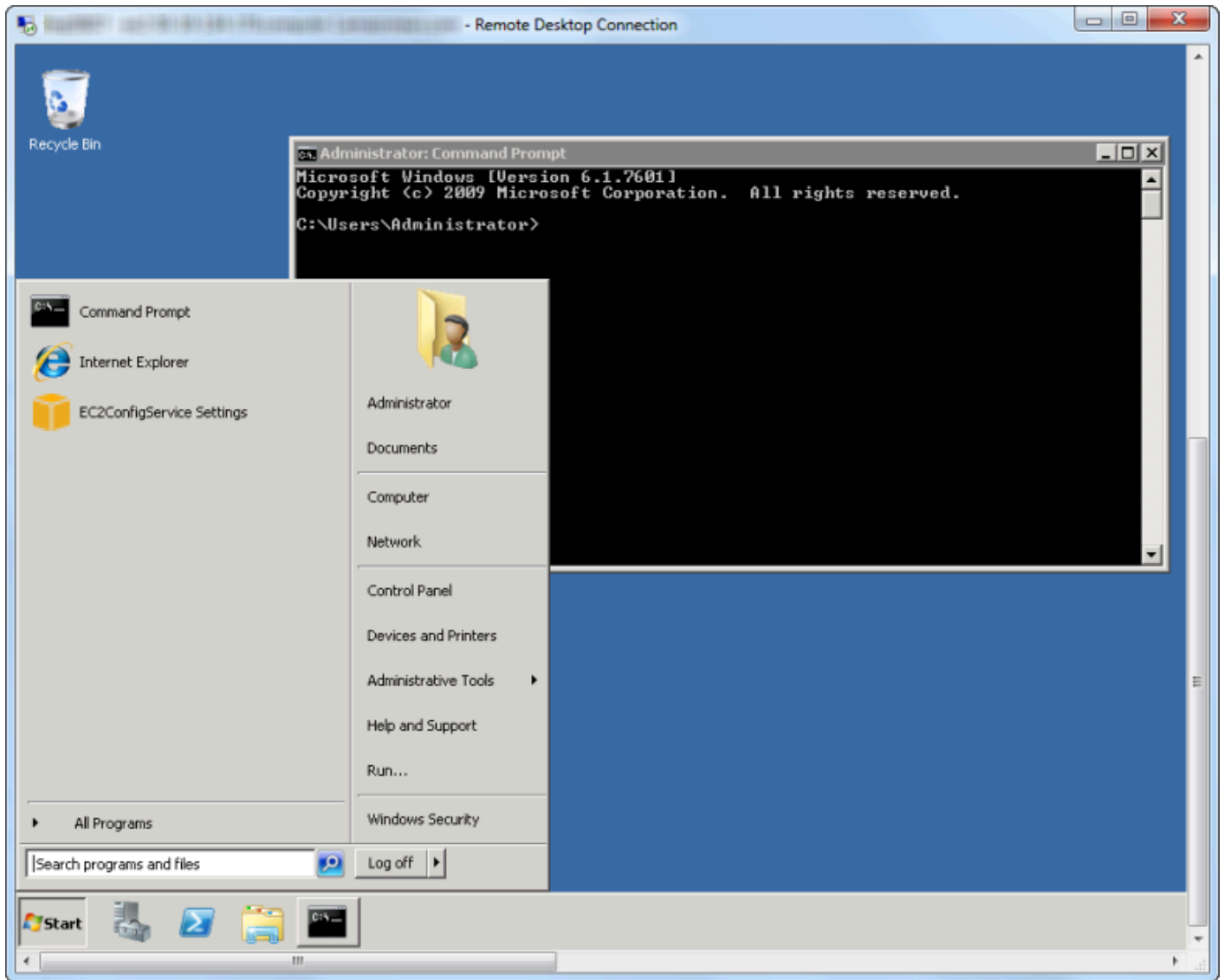
如果 EC2 執行個體最近才啟動，您可能無法連線的原因有兩個：

- 遠端桌面服務可能尚未啟動並執行。請等待幾分鐘後再試一次。
- 密碼資訊可能尚未傳輸至執行個體。在這種情況下，您會看到類似以下內容的訊息方塊。



尚未提供密碼

下列螢幕擷取畫面顯示透過遠端桌面以管理員身分連線的使用者。



遠端桌面

結束 Amazon EC2 執行個體

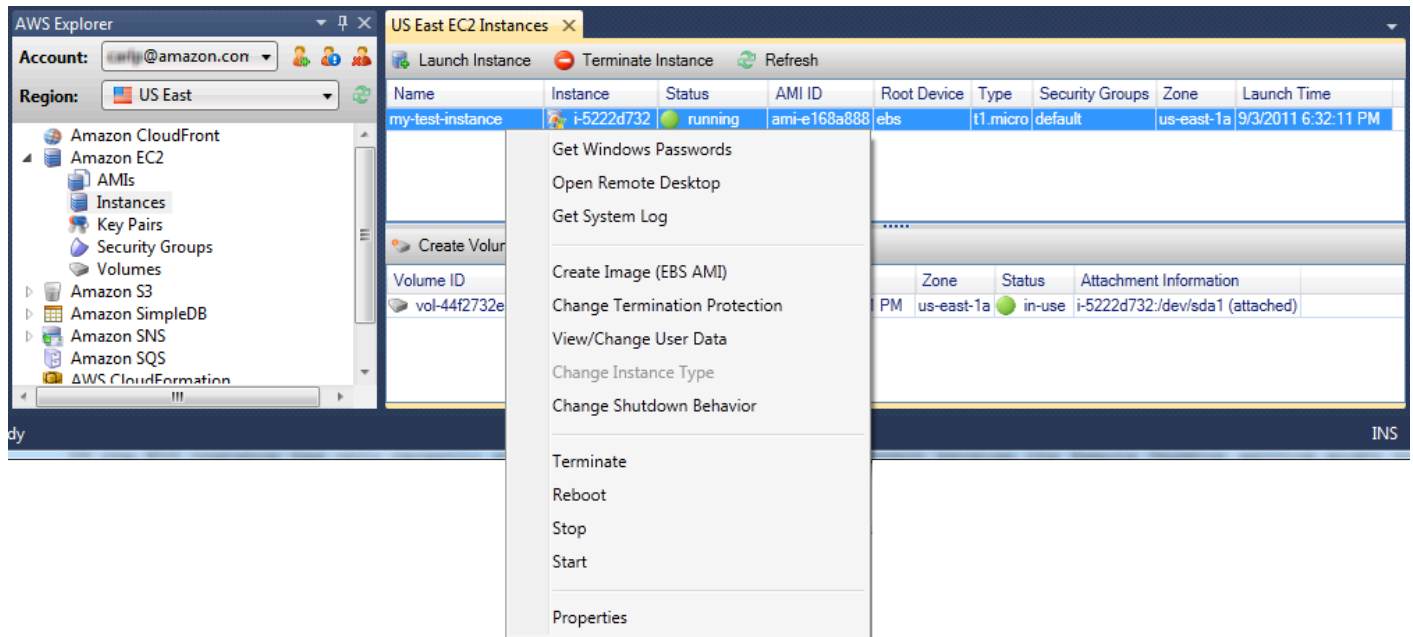
使用 AWS Toolkit，您可以從 Visual Studio 停止或終止執行中的 Amazon EC2 執行個體。若要停止執行個體，EC2 執行個體必須使用 Amazon EBS 磁碟區。如果 EC2 執行個體未使用 Amazon EBS 磁碟區，則您唯一的選項是終止執行個體。

如果您停止執行個體，儲存在 EBS 磁碟區上的資料會保留。如果您終止執行個體，儲存在執行個體本機儲存裝置上的所有資料都會遺失。在任何一種情況下，停止或終止，您都不會繼續支付 EC2 執行個體的費用。不過，如果您停止執行個體，則會繼續向您收取執行個體停止後持續存在的 EBS 儲存體費用。

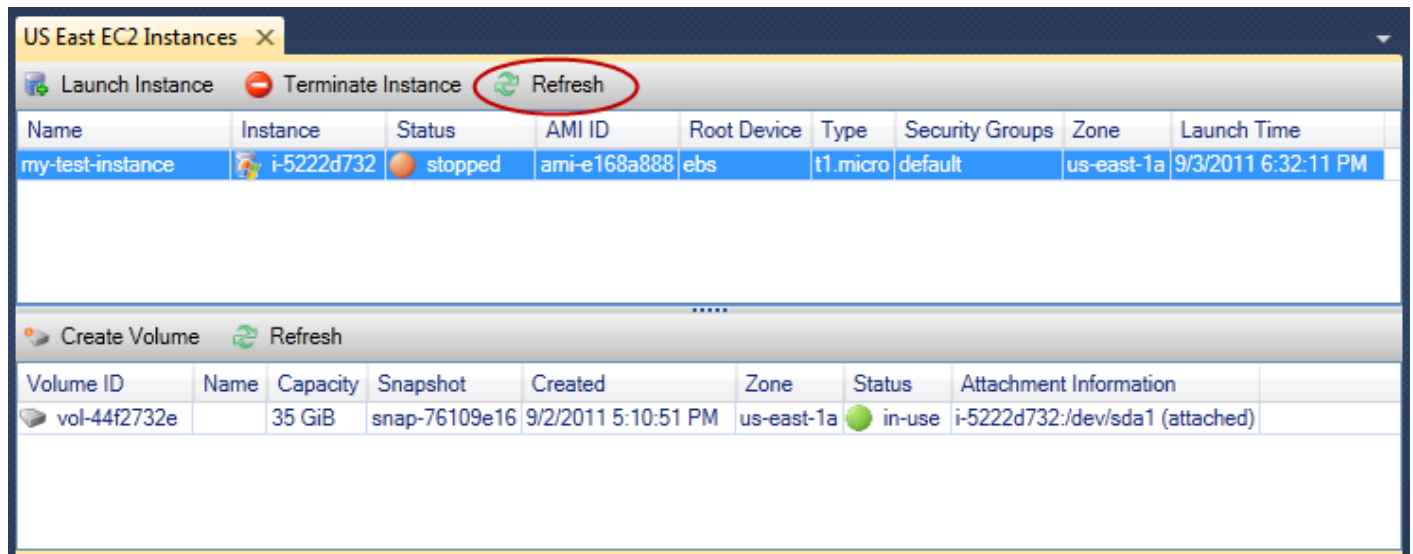
結束執行個體的另一種可能方法是使用遠端桌面連線至執行個體，然後從 Windows 開始功能表使用關機。在這種情況下，您可以將執行個體設定為停止或終止。

停止 Amazon EC2 執行個體

1. 在 AWS Explorer 中，展開 Amazon EC2 節點，開啟執行個體的內容（按一下滑鼠右鍵）選單，然後選擇檢視。在執行個體清單中，在您要停止的執行個體上按一下滑鼠右鍵，然後從內容功能表中選擇停止。選擇是以確認您想要停止執行個體。

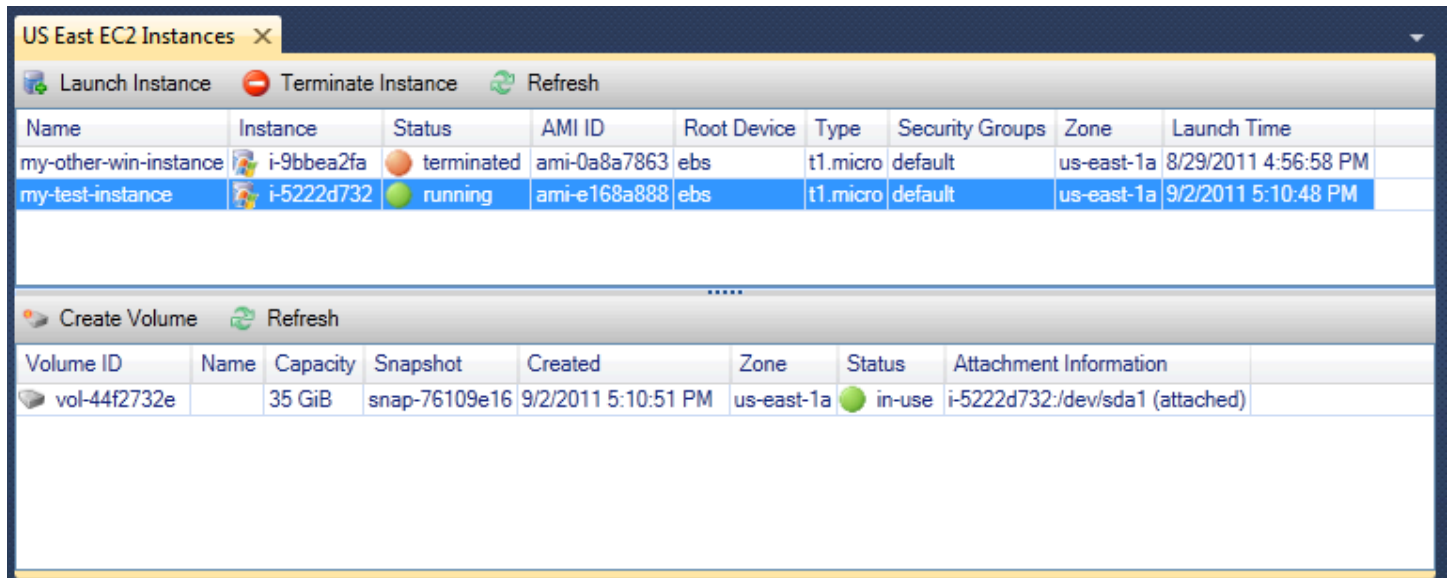


2. 在執行個體清單頂端，選擇重新整理以查看 Amazon EC2 執行個體狀態的變更。由於我們停止而不是終止執行個體，因此與執行個體相關聯的 EBS 磁碟區仍處於作用中狀態。



已終止的執行個體仍可見

如果您終止執行個體，它將繼續與執行中或已停止的執行個體一起出現在執行個體清單中。最後，會 AWS 回收這些執行個體，它們會從清單中消失。您不需要為處於終止狀態的執行個體付費。



The screenshot displays the AWS Management Console interface for EC2 instances in the US East region. It shows two instances: 'my-other-win-instance' (terminated) and 'my-test-instance' (running). Below the instances, an EBS volume 'vol-44f2732e' is shown as attached to the 'my-test-instance'.

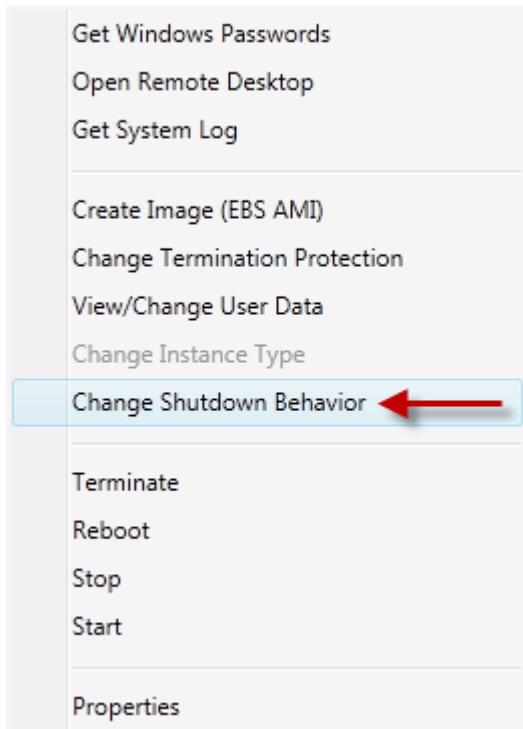
Name	Instance	Status	AMI ID	Root Device	Type	Security Groups	Zone	Launch Time
my-other-win-instance	i-9bbea2fa	terminated	ami-0a8a7863	ebs	t1.micro	default	us-east-1a	8/29/2011 4:56:58 PM
my-test-instance	i-5222d732	running	ami-e168a888	ebs	t1.micro	default	us-east-1a	9/2/2011 5:10:48 PM

Volume ID	Name	Capacity	Snapshot	Created	Zone	Status	Attachment Information
vol-44f2732e		35 GiB	snap-76109e16	9/2/2011 5:10:51 PM	us-east-1a	in-use	i-5222d732:/dev/sda1 (attached)

在關閉時指定 EC2 執行個體的行為

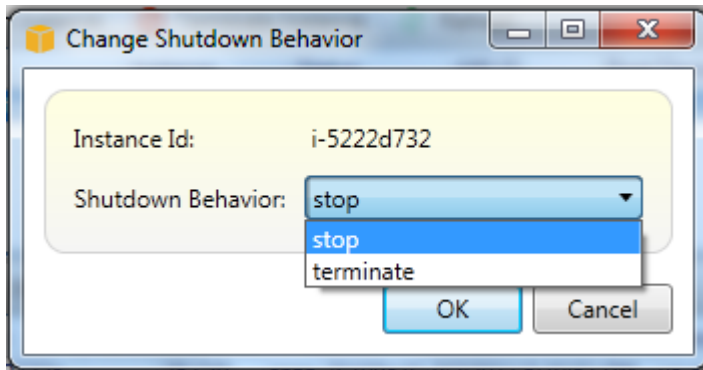
AWS 工具組可讓您指定如果從開始功能表中選取關機，Amazon EC2 執行個體是否會停止或終止。

1. 在執行個體清單中，在 Amazon EC2 執行個體上按一下滑鼠右鍵，然後選擇變更關閉行為。



變更關閉行為選單項目

2. 在變更關閉行為對話方塊中，從關閉行為下拉式清單中，選擇停止或終止。



管理 Amazon ECS 執行個體

AWS Explorer 提供 Amazon Elastic Container Service (Amazon ECS) 叢集和容器儲存庫的詳細檢視。您可以從 Visual Studio 開發環境中建立、刪除和管理叢集和容器詳細資訊。

修改服務屬性

您可以從叢集檢視檢視服務詳細資訊、服務事件和服務屬性。

1. 在 AWS Explorer 中，開啟要管理之叢集的內容（按一下滑鼠右鍵）選單，然後選擇檢視。
2. 在 ECS 叢集檢視中，按一下左側的服務，然後按一下詳細資訊檢視中的詳細資訊索引標籤。您可以按一下事件來查看事件訊息和部署狀態。
3. 按一下 Edit (編輯)。您可以變更所需的任務計數，以及運作狀態百分比下限和上限。
4. 按一下儲存以接受變更，或按一下取消以還原現有值。

停止任務

您可以在叢集檢視中查看任務的目前狀態，並停止一或多個任務。

停止任務

1. 在 AWS Explorer 中，開啟叢集的內容（按一下滑鼠右鍵）選單，其中包含您要停止的任務，然後選擇檢視。
2. 在 ECS 叢集檢視中，按一下左側的任務。

3. 確定所需的任務狀態設定為 **Running**。選擇要停止的個別任務，然後按一下停止或按一下全部停止以選取並停止所有執行中的任務。
4. 在停止任務對話方塊中，選擇是。

刪除服務

您可以從叢集檢視中刪除叢集中的服務。

刪除叢集服務

1. 在 AWS Explorer 中，使用您要刪除的服務開啟叢集的內容（按一下滑鼠右鍵）選單，然後選擇檢視。
2. 在 ECS 叢集檢視中，按一下左側的服務，然後按一下刪除。
3. 在刪除叢集對話方塊中，如果叢集中有負載平衡器和目標群組，您可以選擇使用叢集刪除它們。刪除服務時不會使用它們。
4. 在刪除叢集對話方塊中，選擇確定。刪除叢集時，會從 AWS Explorer 中移除叢集。

刪除叢集

您可以從 AWS Explorer 刪除 Amazon Elastic Container Service 叢集。

刪除叢集

1. 在 AWS Explorer 中，開啟您要在 Amazon ECS 叢集節點下刪除之叢集的內容（按一下滑鼠右鍵）選單，然後選擇刪除。
2. 在刪除叢集對話方塊中，選擇確定。刪除叢集時，會從 AWS Explorer 中移除叢集。

建立儲存庫

您可以從 AWS Explorer 建立 Amazon Elastic Container Registry 儲存庫。

建立儲存庫

1. 在 AWS Explorer 中，開啟 Amazon ECS 下儲存庫節點的內容（按一下滑鼠右鍵）選單，然後選擇建立儲存庫。
2. 在建立儲存庫對話方塊中，提供儲存庫名稱，然後選擇確定。

刪除儲存庫

您可以從 AWS Explorer 刪除 Amazon Elastic Container Registry 儲存庫。

刪除儲存庫

1. 在 AWS Explorer 中，開啟 Amazon ECS 下儲存庫節點的內容（按一下滑鼠右鍵）選單，然後選擇刪除儲存庫。
2. 在刪除儲存庫對話方塊中，您可以選擇刪除儲存庫，即使其包含映像。否則，只有在空白時才會將其刪除。按一下是。

從 AWS Explorer 管理安全群組

Toolkit for Visual Studio 可讓您建立和設定安全群組，以搭配 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體和 使用 CloudFormation。當您啟動 Amazon EC2 執行個體或部署應用程式時 CloudFormation，您可以指定要與 Amazon EC2 執行個體建立關聯的安全群組。（部署以 CloudFormation 建立 Amazon EC2 執行個體。）

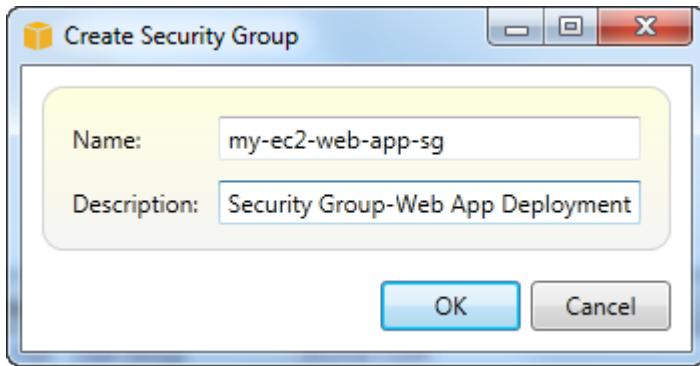
安全群組對於傳入網路流量的功能，就像防火牆一樣。安全群組會指定 Amazon EC2 執行個體上允許的網路流量類型。它也可以指定只接受來自特定 IP 地址的傳入流量，或只接受來自指定使用者或其他安全群組的傳入流量。

建立安全群組

在本節中，我們將建立安全群組。建立之後，安全群組將不會設定任何許可。以額外操作處理設定許可。

建立安全群組

1. 在 AWS Explorer 的 Amazon EC2 節點下，開啟安全群組節點上的內容（按一下滑鼠右鍵）選單，然後選擇檢視。
2. 在 EC2 安全群組索引標籤上，選擇建立安全群組。
3. 在建立安全群組對話方塊中，輸入安全群組的名稱和描述，然後選擇確定。

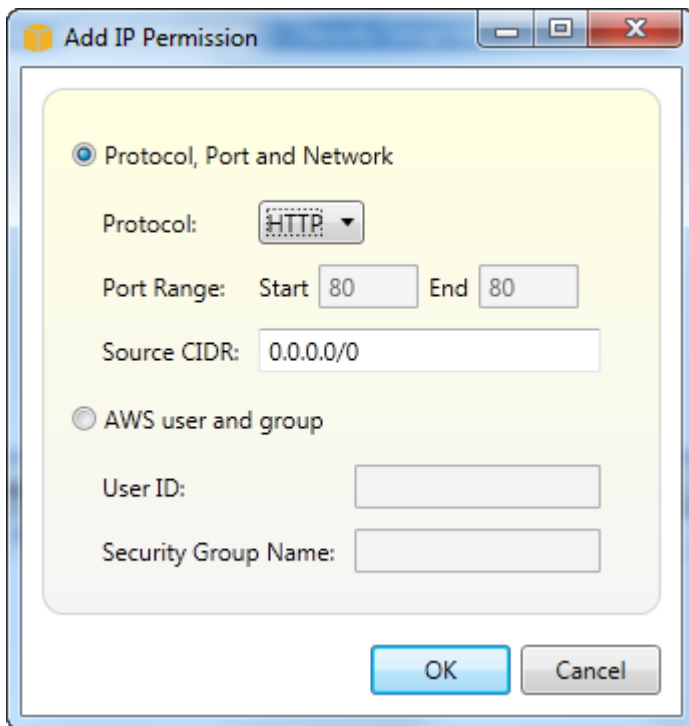


為安全群組新增許可

在本節中，我們會將許可新增至安全群組，以允許透過 HTTP 和 HTTPS 通訊協定的 Web 流量。我們也會允許其他電腦使用 Windows 遠端桌面通訊協定 (RDP) 進行連線。

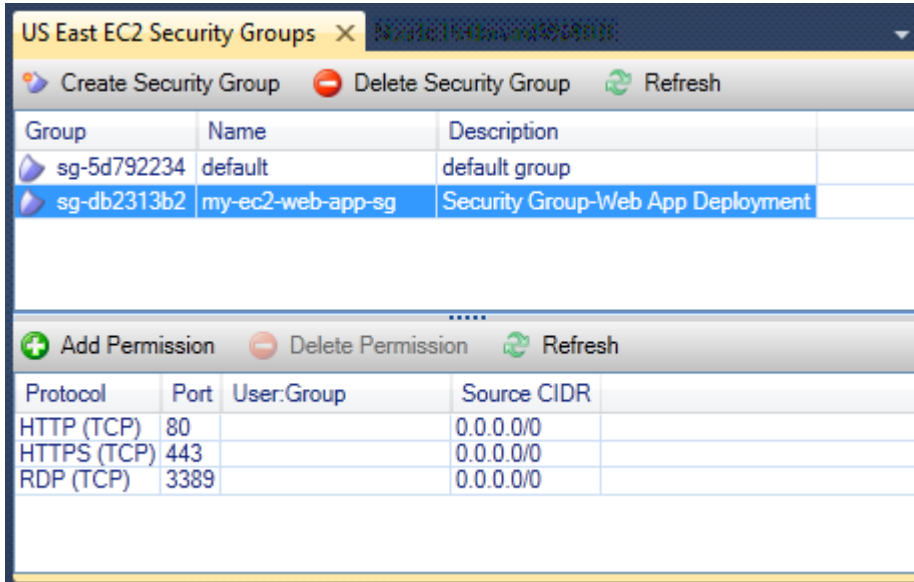
將許可新增至安全群組

1. 在 EC2 安全群組索引標籤上，選擇安全群組，然後選擇新增許可按鈕。
2. 在新增 IP 許可對話方塊中，選擇通訊協定、連接埠和網路選項按鈕，然後從通訊協定下拉式清單中選擇 HTTP。連接埠範圍會自動調整為連接埠 80，這是 HTTP 的預設連接埠。來源 CIDR 欄位預設為 0.0.0.0/0，指定將從任何外部 IP 地址接受 HTTP 網路流量。選擇確定。



此安全群組的開放連接埠 80 (HTTP)

3. 針對 HTTPS 和 RDP 重複此程序。您的安全群組許可現在看起來應該如下所示。



您也可以指定使用者 ID 和安全群組名稱，在安全群組中設定許可。在此情況下，此安全群組中的 Amazon EC2 執行個體將接受來自指定安全群組中 Amazon EC2 執行個體的所有傳入網路流量。您也必須指定使用者 ID 做為取消混淆安全群組名稱的方式；安全群組名稱在所有中都不需要是唯一的 AWS。如需安全群組的詳細資訊，請前往 [EC2 文件](#)。

從 Amazon EC2 執行個體建立 AMI

您可以使用 [建立 Amazon Machine Image \(AMI\) AWS Toolkit for Visual Studio](#)。如需 AMIs 的詳細資訊，請參閱《[Amazon Elastic Compute Cloud for Windows Instances 使用者指南](#)》中的 [Amazon Machine Image \(AMI\)](#) 主題。

若要從結束的 Amazon EC2 執行個體建立 AMI，請完成下列程序。

從現有的 Amazon EC2 執行個體建立 AMI

1. 從 AWS Toolkit Explorer 展開 Amazon EC2，然後選擇執行個體以檢視現有執行個體的清單。
2. 在您想要用作 AMI 基礎的執行個體上按一下滑鼠右鍵，然後選擇建立映像 (ABS AMI) 以開啟建立映像對話方塊視窗。
3. 從建立映像對話方塊中，將映像的名稱和描述新增至提供的欄位中，然後選擇確定按鈕以繼續。
4. 建立映像時，映像建立確認視窗會在 Visual Studio 中開啟，選擇確定按鈕以繼續。

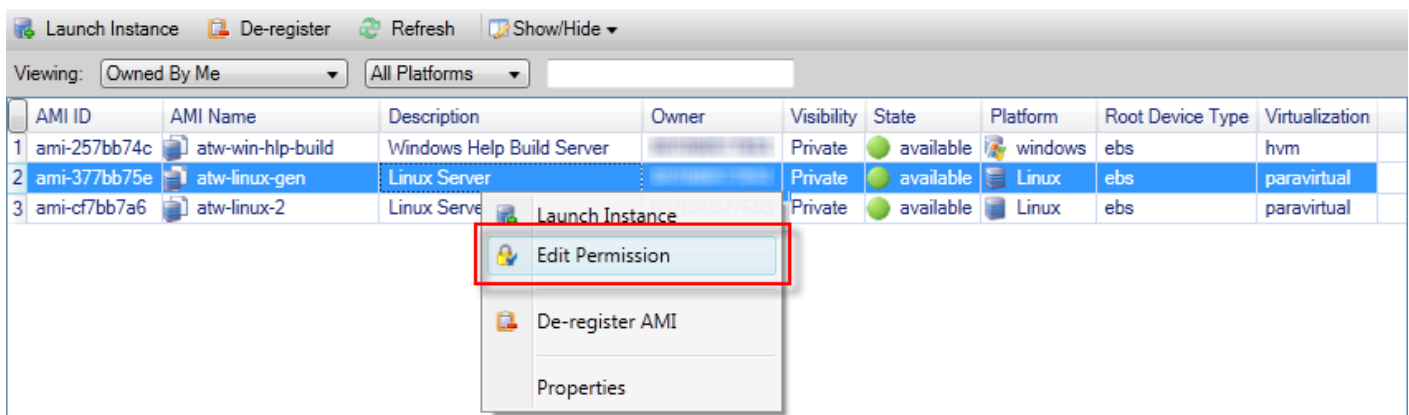
若要使用 AWS Toolkit 檢視您的新 AMI，請展開 Amazon EC2 並按兩下 AMIs，以在 Visual Studio Editor payne 中開啟視窗，顯示現有 AMIs 的清單。如果您在清單中看不到新的 AMI，請選擇 AMI 視窗頂端的重新整理按鈕。

在 Amazon Machine Image 上設定啟動許可

您可以從 AWS Explorer 中的 AMIs 設定 AMIs 啟動許可。您可以使用設定 AMI 許可對話方塊從 AMIs 複製許可。

在 AMI 上設定許可

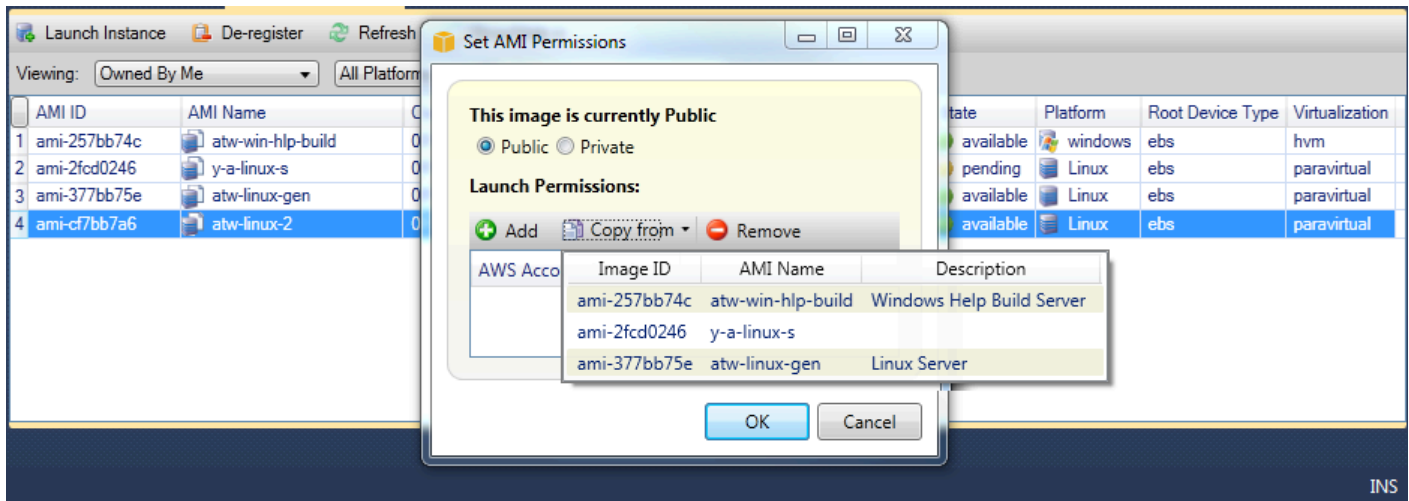
1. 在 AWS Explorer 的 AMIs 中，開啟 AMI 上的內容（按一下滑鼠右鍵）選單，然後選擇編輯許可。



2. 設定 AMI 許可對話方塊中有三個可用選項：

- 若要提供啟動許可，請選擇新增，然後輸入您要為其授予啟動許可之 AWS 使用者的帳號。
- 若要移除啟動許可，請選擇您要從中移除啟動許可之 AWS 使用者的帳戶號碼，然後選擇移除。
- 若要將許可從一個 AMI 複製到另一個 AMI，請從清單中選擇 AMI，然後選擇複製來源。在您選擇的 AMI 上具有啟動許可的使用者，將會獲得目前 AMI 的啟動許可。您可以使用複製來源清單中的其他 AMIs 重複此程序，將多個 AMIs 的許可複製到目標 AMI。

從 Explorer 顯示 AMI 檢視時，複製者清單僅包含帳戶所擁有且處於作用中狀態 AMIs AWS。AMIs 因此，如果作用中帳戶沒有其他 AMIs 擁有，複製清單可能不會顯示任何 AMIs。



複製 AMI 許可對話方塊

Amazon Virtual Private Cloud (VPC)

Amazon Virtual Private Cloud (Amazon VPC) 可讓您在定義的虛擬網路中啟動 Amazon Web Services 資源。此虛擬網路與您在自己的資料中心中操作的傳統網路相似，且具備使用 AWS 可擴展基礎設施的優勢。如需詳細資訊，請前往 [Amazon VPC 使用者指南](#)。

Toolkit for Visual Studio 可讓開發人員存取類似但 [AWS 管理主控台](#) 來自 Visual Studio 開發環境的 VPC 功能。AWS Explorer 的 Amazon VPC 節點包含下列區域的子節點。

- [VPC](#)
- [子網路](#)
- [彈性 IP](#)
- [網際網路閘道 \(Internet Gateway\)](#)
- [網路 ACL](#)
- [路由表](#)
- [安全群組](#)

使用 建立用於部署的公有私有 VPC AWS Elastic Beanstalk

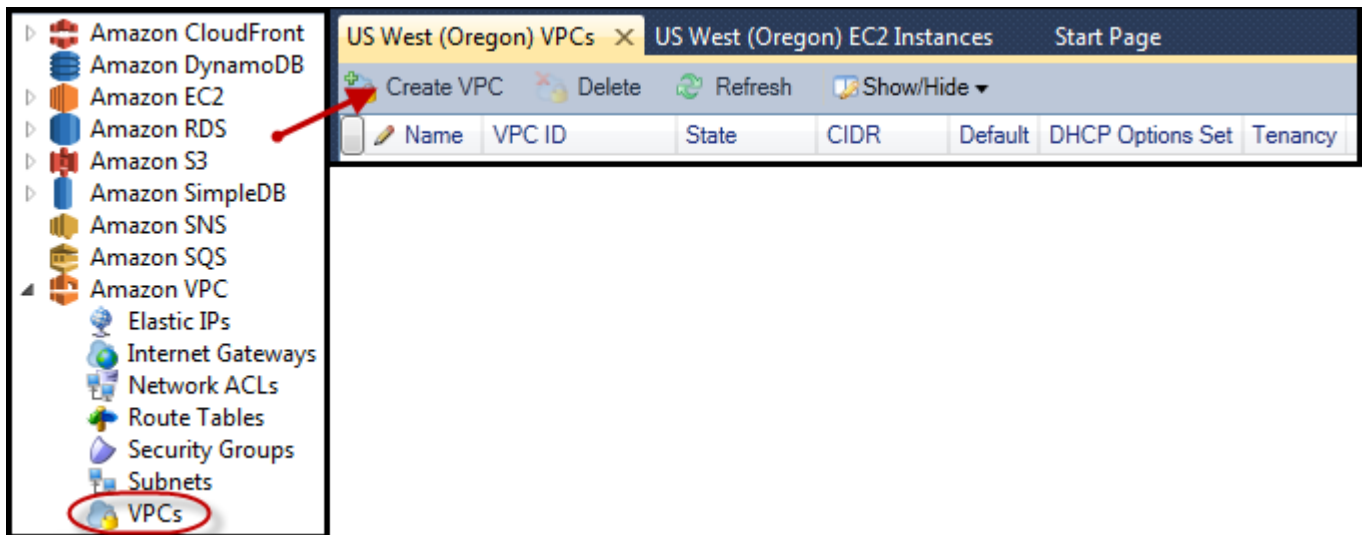
本節說明如何建立同時包含公有和私有子網路的 Amazon VPC。公有子網路包含執行網路位址轉譯 (NAT) 的 Amazon EC2 執行個體，可讓私有子網路中的執行個體與公有網際網路通訊。兩個子網路必須位於相同的可用區域 (AZ)。

這是在 VPC 中部署 AWS Elastic Beanstalk 環境所需的最低 VPC 組態。在此案例中，託管應用程式的 Amazon EC2 執行個體位於私有子網路中；將傳入流量路由至應用程式的 Elastic Load Balancing 負載平衡器位於公有子網路中。

如需網路位址轉譯 (NAT) 的詳細資訊，請參閱《Amazon Virtual Private Cloud 使用者指南》中的 [NAT 執行個體](#)。如需如何設定部署以使用 VPC 的範例，請參閱 [部署至 Elastic Beanstalk](#)。

建立公有/私有子網路 VPC

1. 在 AWS Explorer 的 Amazon VPC 節點中，開啟 VPCs 子節點，然後選擇建立 VPC。



2. 設定 VPC，如下所示：

- 輸入 VPC 的名稱。
- 選取具有公有子網路和具有私有子網路核取方塊。
- 從每個子網路的可用區域下拉式清單方塊中，選擇可用區域。請務必對兩個子網路使用相同的可用區域。
- 對於私有子網路，在 NAT 金鑰對名稱中，提供金鑰對。此金鑰對用於執行從私有子網路到公有網際網路之網路位址轉譯的 Amazon EC2 執行個體。
- 選取設定預設安全群組以允許 NAT 流量核取方塊。

輸入 VPC 的名稱。選取具有公有子網路和具有私有子網路核取方塊。從每個子網路的可用區域下拉式清單方塊中，選擇可用區域。請務必對兩個子網路使用相同的可用區域。對於私有子網路，在 NAT 金鑰對名稱中，提供金鑰對。此金鑰對用於執行從私有子網路到公有網際網路之網路位址轉譯的 Amazon EC2 執行個體。選取設定預設安全群組以允許 NAT 流量核取方塊。

選擇確定。

Create VPC

Name:

CIDR Block*:

Tenancy:

With Public Subnet

Public Subnet: Availability Zone:

A subnet will be added to the VPC with an internet gateway associated to it. This will allow instances in this subnet access to the internet.

With Private Subnet

Private Subnet: Availability Zone:

NAT Instance Type: NAT Key Pair Name:

Configure default security group to allow traffic to NAT

Instances in the private subnet can establish outbound connections to the Internet via the public subnet using Network Address Translation. (Hourly charges for NAT instances apply)

Creation of public or private subnets will be performed in the background. To check the status view the output window.

您可以在 AWS Explorer 的 VPC 索引標籤中檢視新的 VPCs。

Name	VPC ID	State	CIDR	Default	DHCP Options Set	Tenancy
1 myDeploymentVPC	vpc-da0013b3	available	10.0.0.0/16	False	dopt-80cddae9	default

NAT 執行個體可能需要幾分鐘的時間才能啟動。當可用時，您可以在 AWS Explorer 中展開 Amazon EC2 節點，然後開啟執行個體子節點來檢視它。

Amazon Elastic Block Store (Amazon EBS) 磁碟區會自動為 NAT 執行個體建立。如需 Amazon EBS 的詳細資訊，請參閱《Amazon Amazon EC2 執行個體使用者指南》中的 [Amazon Elastic Block Store \(EBS\)](#) 主題。

The screenshot shows the AWS Management Console interface. At the top, there are tabs for 'Env: myPBEEnv', 'US West (Oregon) VPCs', 'US West (Oregon) EC2 Instances', and 'SimpleDbMembershipProvider.cs'. Below the tabs, there are buttons for 'Launch Instance', 'Terminate Instance', 'Refresh', and 'Show/Hide'. The main content area contains two tables:

Instance ID	Status	AMI ID	Type	Security Groups	Zone	Name	Instance Profile	Key Pair Name	Launch Time	Public DNS
i-709d9342	running	ami-52ff7262	m1.small	default	us-west-2b	NAT		key-pair-vs-1ip	4/5/2013 9:26:57 AM	

Volume ID	Capacity	Snapshot ID	Created	Zone	Status	Attachment Information	vol-tag
vol-da5a91e2	8 GiB	snap-4301d52b	4/5/2013 9:27:00 AM	us-west-2b	in-use	i-709d9342:/dev/sda1 (attached)	

如果您將應用程式部署到 [AWS Elastic Beanstalk 環境](#)，並選擇在 VPC 中啟動環境，Toolkit 會將 VPC 的組態資訊填入發佈至 Amazon Web Services 對話方塊。

Toolkit 只會從 Toolkit 中建立 VPCs 填入資訊，而不是從使用 建立 VPCs 填入資訊 AWS 管理主控台。這是因為當 Toolkit 建立 VPC 時，它會標記 VPC 的元件，以便其可以存取其資訊。

下列來自 部署精靈的螢幕擷取畫面顯示一個對話方塊範例，其中填入了 Toolkit 中建立之 VPC 的值。

The screenshot shows the 'Publish to AWS' dialog box. The title bar says 'Publish to AWS'. The main content area is titled 'AWS Options' and contains the following fields:

- AWS Options**: Set Amazon EC2 options for the deployed application.
- Amazon EC2**:
 - Container type *: 64bit Windows Server 2012 running IIS 8 CFN
 - Use custom AMI: [Empty text box]
 - Instance type *: Micro
 - Key pair *: key-pair-vs-1ip
 - Launch into VPC
 - VPC *: myDeploymentVPC - vpc-da00
 - ELB Scheme *: Public
 - Security Group *: NATGroup (sg-374a535b)
 - ELB Subnet *: Public - subnet-de0013b7 (10.0.0.0/24 - us-west-2b)
 - Instances Subnet *: Private - subnet-d60013bf (10.0.1.0/24 - us-west-2b)

Below the fields, there is a note: *To run AWS Elastic Beanstalk applications inside a VPC, you will need to configure at least the following: Create two subnets: one for your EC2 instances and one for your Elastic Load Balancer. Traffic must be able to be routed from your Elastic Load Balancer to your EC2 instances. Your EC2 instances must be able to connect to the Internet and AWS endpoints. For more information visit [AWS Elastic Beanstalk User Guide](#)*

At the bottom, there are four buttons: Cancel, Back, Next, and Finish.

刪除 VPC

若要刪除 VPC，您必須先終止 VPC 中的任何 Amazon EC2 執行個體。

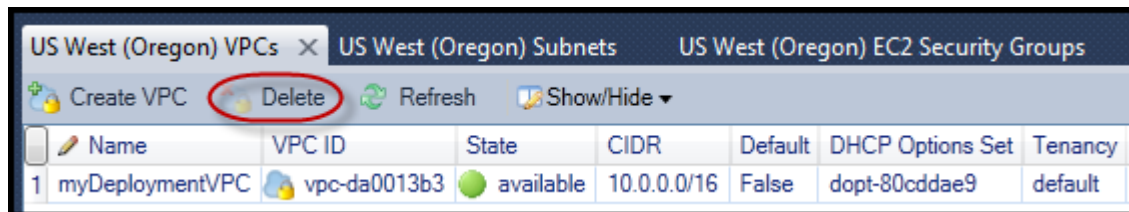
1. 如果您已將應用程式部署到 VPC 中的 AWS Elastic Beanstalk 環境，請刪除環境。這將終止託管您應用程式的任何 Amazon EC2 執行個體以及 Elastic Load Balancing 負載平衡器。

如果您嘗試直接終止託管應用程式的執行個體而不刪除環境，Auto Scaling 服務會自動建立新的執行個體以取代已刪除的執行個體。如需詳細資訊，請前往 [Auto Scaling 開發人員指南](#)。

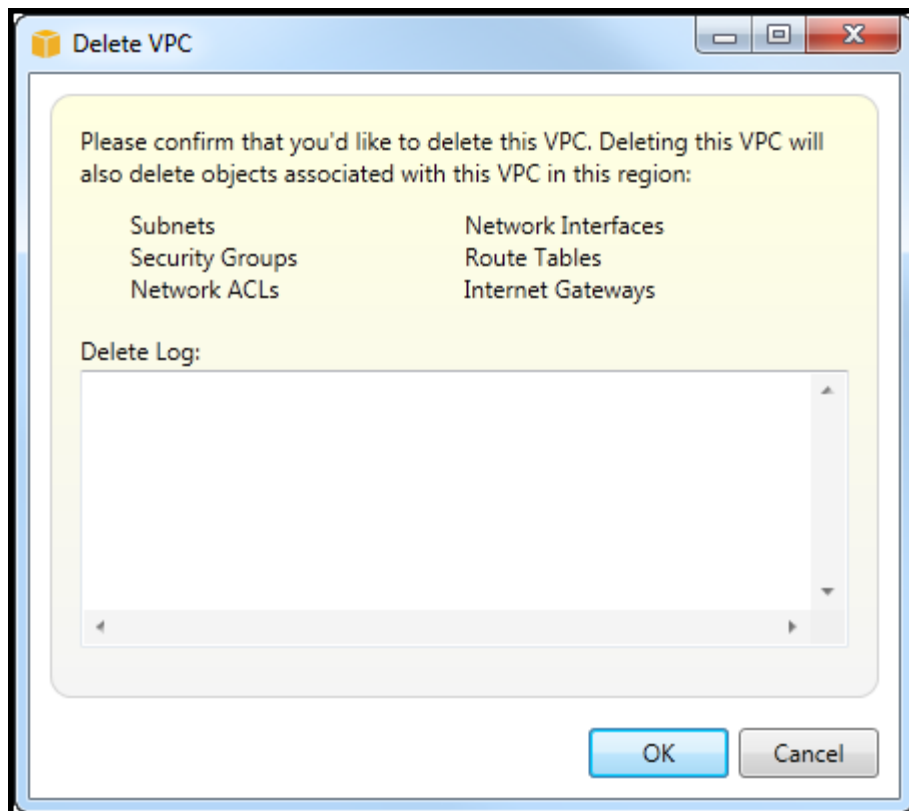
2. 刪除 VPC 的 NAT 執行個體。

您不需要刪除與 NAT 執行個體相關聯的 Amazon EBS 磁碟區，即可刪除 VPC。不過，如果您未刪除磁碟區，即使您刪除 NAT 執行個體和 VPC，仍需支付該磁碟區的費用。

3. 在 VPC 索引標籤上，選擇刪除連結以刪除 VPC。



4. 在刪除 VPC 對話方塊中，選擇確定。



使用 Visual Studio 的 CloudFormation 範本編輯器

Toolkit for Visual Studio 包含 Visual Studio 的 CloudFormation 範本編輯器和 CloudFormation 範本專案。支援的功能包括：

- 使用提供的範本專案類型建立新的 CloudFormation 範本（空白或從現有堆疊或範本複製）。
- 使用自動 JSON 驗證、自動完成、程式碼摺疊和語法反白來編輯範本。
- 自動建議範本中欄位值的內部函數和資源參考參數。
- 從 Visual Studio 為範本執行常見動作的選單項目。

主題

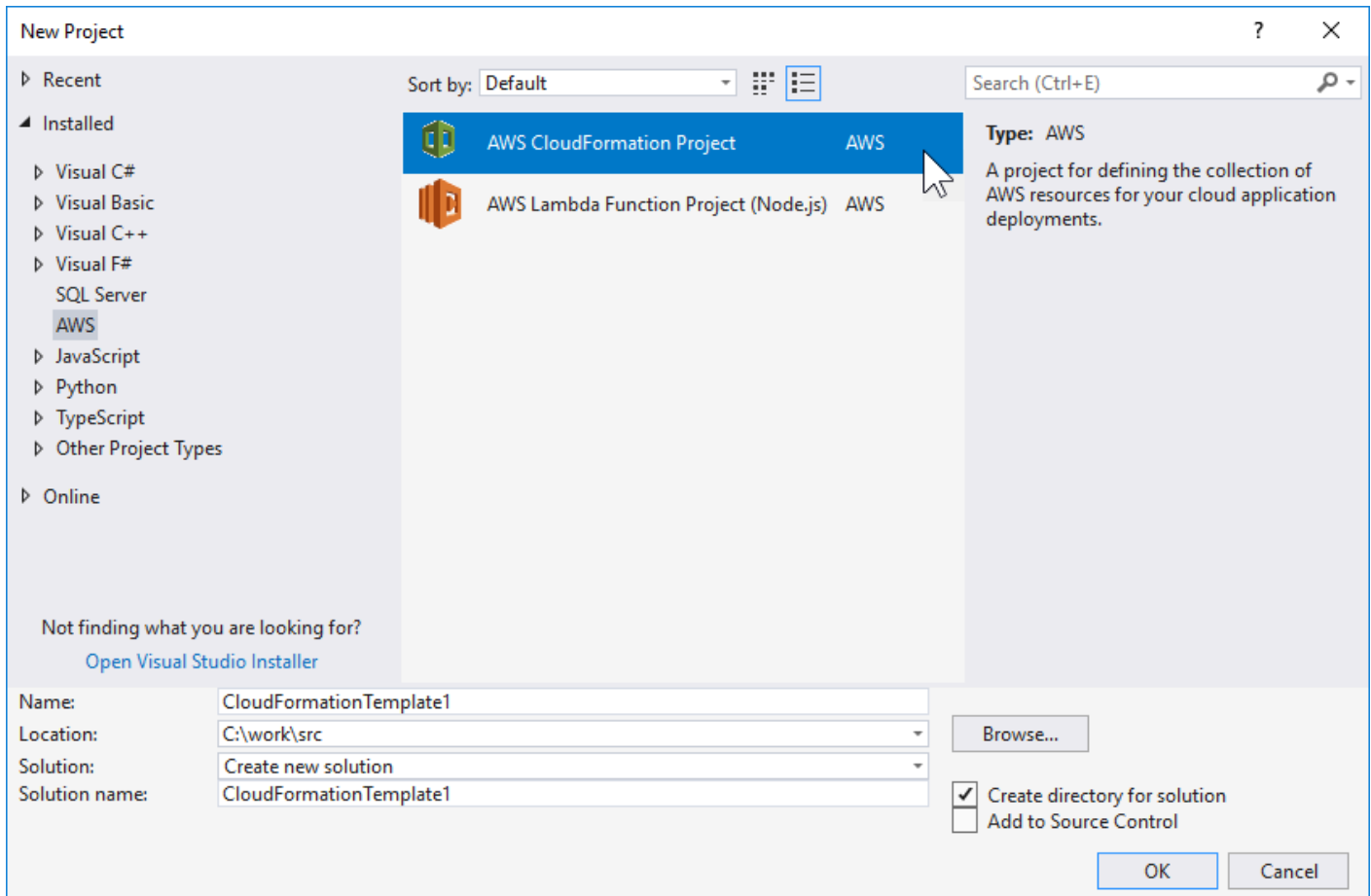
- [在 Visual Studio 中建立 CloudFormation 範本專案](#)
- [在 Visual Studio 中部署 CloudFormation 範本](#)
- [在 Visual Studio 中格式化 CloudFormation 範本](#)

在 Visual Studio 中建立 CloudFormation 範本專案

建立範本專案

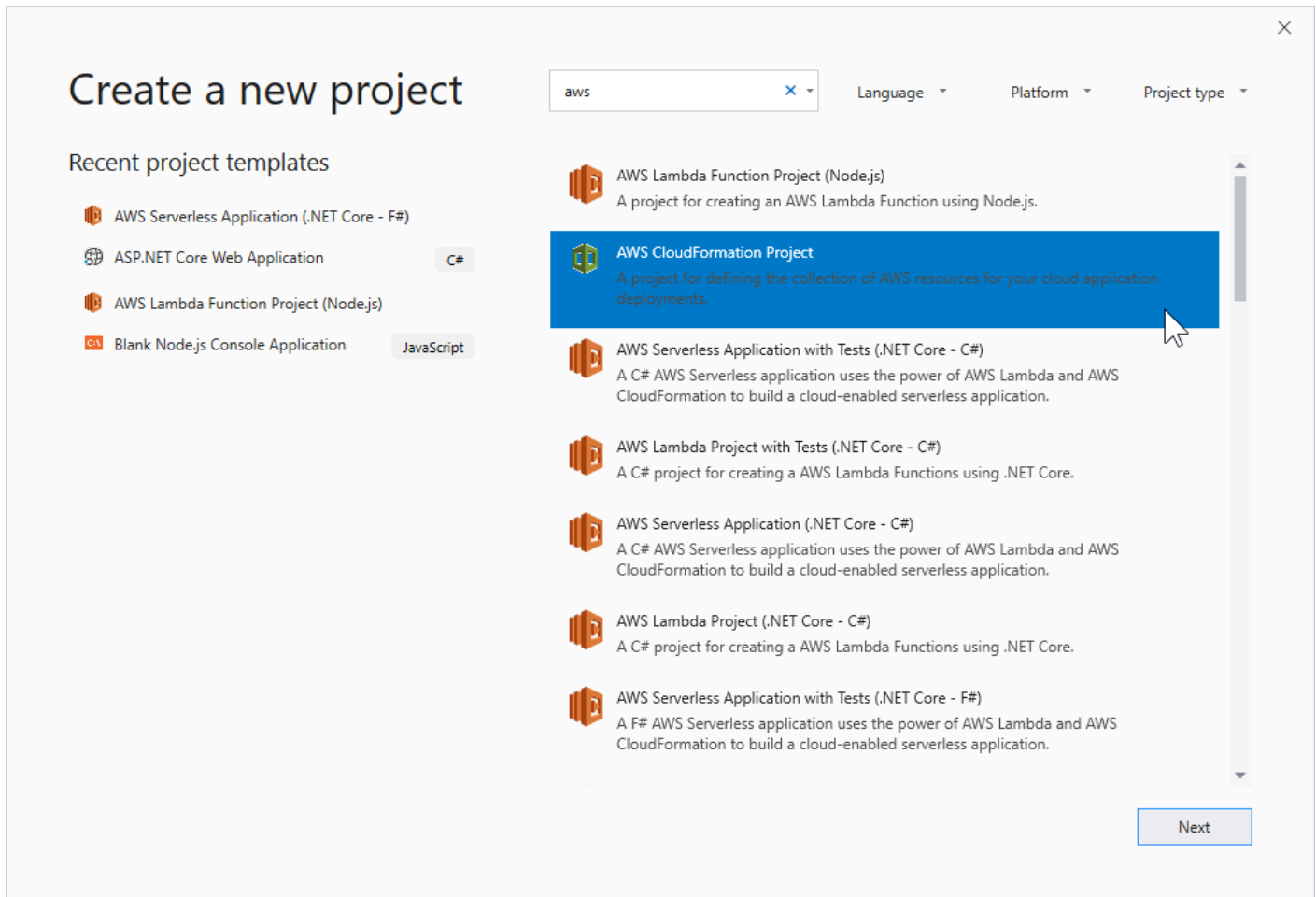
1. 在 Visual Studio 中，選擇檔案，選擇新增，然後選擇專案。
2. 針對 Visual Studio 2017：

在新增專案對話方塊中，展開已安裝，然後選取 AWS。



針對 Visual Studio 2019 :

在新增專案對話方塊中，確保語言、平台和專案類型下拉式方塊設定為「全部...」，並在搜尋欄位中輸入 aws。



3. 選取 AWS CloudFormation 專案範本。

4. 針對 Visual Studio 2017：

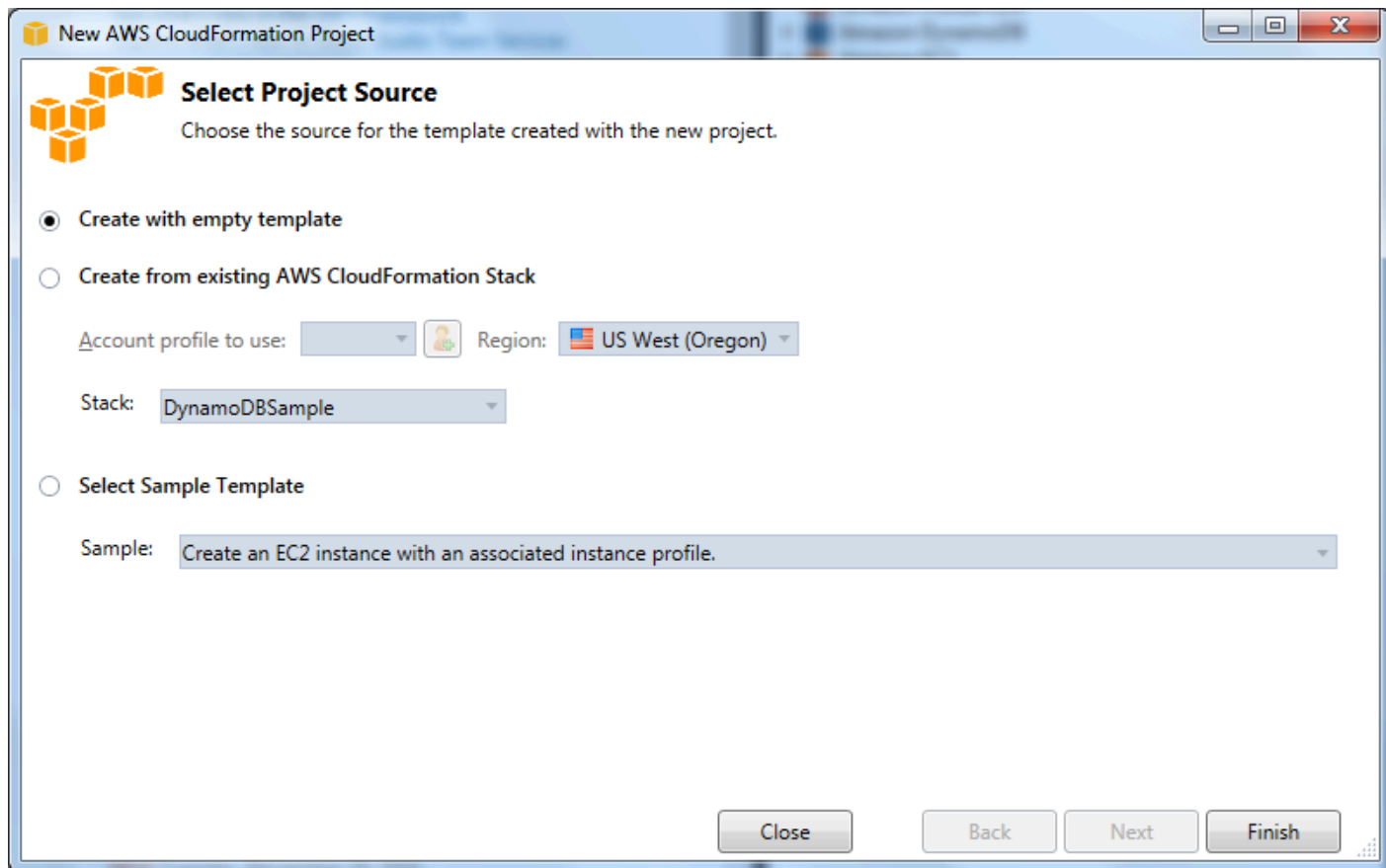
輸入範本專案所需的名稱、位置等，然後按一下確定。

針對 Visual Studio 2019：

按一下 Next (下一步)。在下一個對話方塊中，輸入範本專案所需的名稱、位置等，然後按一下建立。

5. 在選取專案來源頁面上，選擇您將建立的範本來源：

- 使用空白範本建立 會產生新的空白 CloudFormation 範本。
- 從現有 AWS |CFN| 堆疊建立 會從您 AWS 帳戶中的現有堆疊產生範本。(堆疊不需要狀態為 CREATE_COMPLETE。)
- 選取範例範本會從其中一個 CloudFormation 範例範本產生範本。

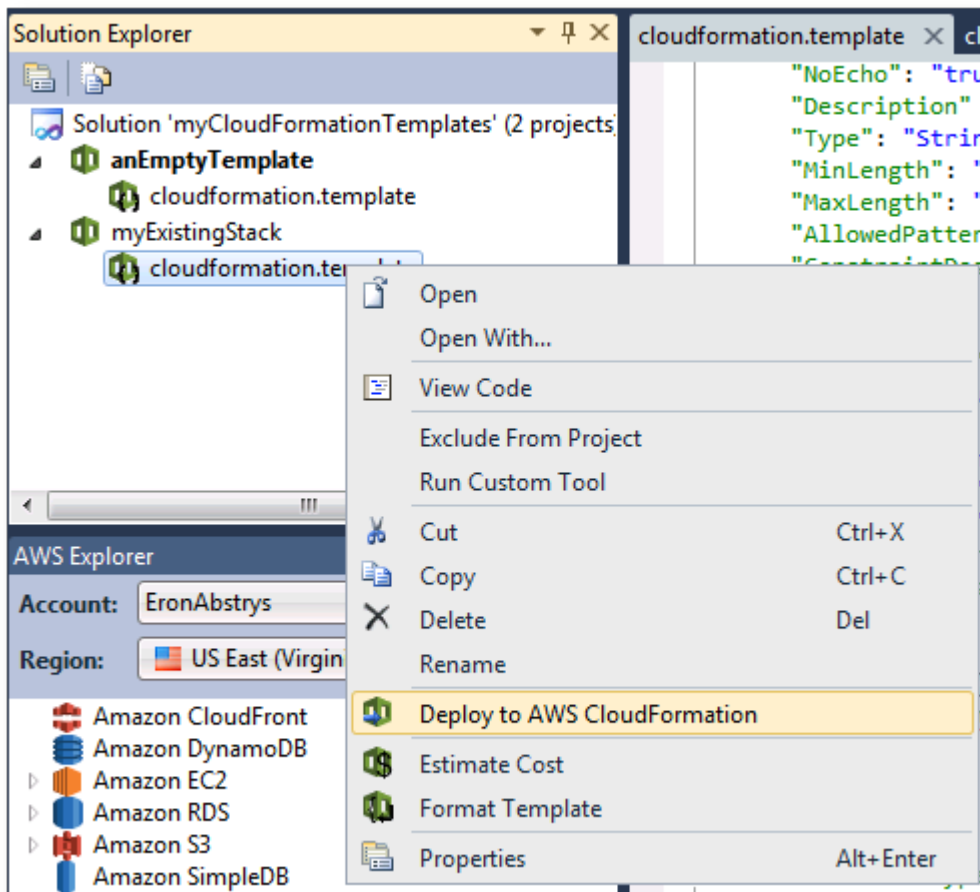


6. 若要完成 CloudFormation 範本專案的建立，請選擇完成。

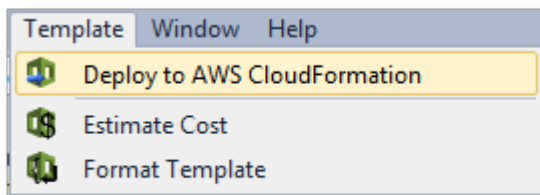
在 Visual Studio 中部署 CloudFormation 範本

部署 CFN 範本

1. 在 Solution Explorer 中，開啟您要部署之範本的內容（按一下滑鼠右鍵）選單，然後選擇部署。
AWS CloudFormation



或者，若要部署目前正在編輯的範本，請從範本功能表中選擇部署至 AWS CloudFormation。



2. 在部署範本頁面上，選擇要 AWS 帳戶 用來啟動堆疊的 ，以及要啟動堆疊的區域。

Deploy Template

Select Template

To create a stack, fill in the name for your stack and select a template. You may choose one of the sample templates to get started quickly or on your local hard drive.

Account to use: Region:

Create New Stack

SNS Topic (Optional):

Creation Timeout:

Rollback on failure

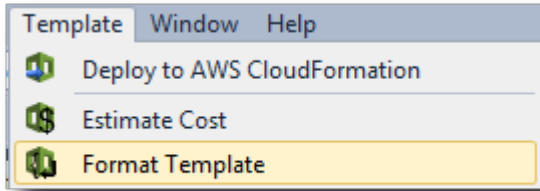
Update Existing Stack

3. 選擇建立新堆疊，然後輸入堆疊的名稱。
4. 選擇下列其中任何一個選項 (或不選擇)：
 - 若要接收有關堆疊進度的通知，請從 SNS 主題下拉式清單中選擇 SNS 主題。您也可以選擇建立新主題並在方塊中輸入電子郵件地址，以建立 SNS 主題。
 - 使用建立逾時指定在宣告失敗之前，CloudFormation 應允許建立堆疊的時間長度（並復原，除非已清除轉返失敗選項）。
 - 如果您希望堆疊在失敗時轉返（即刪除本身），請在失敗時使用轉返。如果您希望堆疊保持作用中狀態以進行偵錯，即使堆疊無法完成啟動，也請保持清除此選項。
5. 選擇完成以啟動堆疊。

在 Visual Studio 中格式化 CloudFormation 範本

- 在 Solution Explorer 中，開啟範本的內容（按一下滑鼠右鍵）選單，然後選擇格式化範本。

或者，若要格式化您目前正在編輯的範本，請從範本功能表中選擇格式化範本。



您的 JSON 程式碼將會格式化，以清楚顯示其結構。

```
"Properties" : {
  "SecurityGroups" : [ { "Ref" : "InstanceSecurityGroup" } ],
  "KeyName" : { "Ref" : "KeyName" },
  "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" : "AWSRegion" }, { "Fn::FindInMap" : [ "AWSInstanceType2Arch", { "Ref" : "InstanceType" }, "Arch" ] } ] },
  "UserData" : { "Fn::Base64" : { "Fn::Join" : [ "", [
    "#!/bin/bash\n",
    "yum update -y aws-cfn-bootstrap\n",
    "\n",
    "/opt/aws/bin/cfn-init -s ", { "Ref" : "AWS::StackName" }, " -r Ec2Instance ",
    "--access-key ", { "Ref" : "HostKeys" },
    "--secret-key ", { "Fn::GetAtt" : [ "HostKeys", "SecretAccessKey" ] },
    "--region ", { "Ref" : "AWS::Region" }, "\n",
    "/opt/aws/bin/cfn-signal -e $? ", { "Ref" : "WaitHandle" }, "\n"
  ] ] } }
  ]
}
},
```

```
"Properties" : {
  "SecurityGroups" : [
    {
      "Ref" : "InstanceSecurityGroup"
    }
  ],
  "KeyName" : {
    "Ref" : "KeyName"
  },
  "ImageId" : {
    "Fn::FindInMap" : [
      "AWSRegionArch2AMI",
      {
        "Ref" : "AWS::Region"
      },
      {
        "Fn::FindInMap" : [
          "AWSInstanceType2Arch",
          {
            "Ref" : "InstanceType"
          },
          "Arch"
        ]
      }
    ]
  },
  "UserData" : {
    "Fn::Base64" : {
      "Fn::Join" : [
        "",
        [
          "#!/bin/bash\n",
          "yum update -y aws-cfn-bootstrap\n",
          "/opt/aws/bin/cfn-init -s ",
          {
            "Ref" : "AWS::StackName"
          },
          " -r Ec2Instance ",
          "--access-key ",
          {
            "Ref" : "HostKeys"
          },
          "\n"
        ]
      ]
    }
  }
}
```

從 AWS Explorer 使用 Amazon S3

Amazon Simple Storage Service (Amazon S3) 可讓您從網際網路的任何連線存放和擷取資料。您存放在 Amazon S3 上的所有資料都與您的帳戶相關聯，預設情況下，只能由您存取。Toolkit for Visual Studio 可讓您將資料存放在 Amazon S3 上，並檢視、管理、擷取和分發該資料。

Amazon S3 使用儲存貯體的概念，您可以將其視為類似於檔案系統或邏輯磁碟機。儲存貯體可包含類似於目錄的資料夾，以及類似於檔案的物件。在本節中，我們將使用這些概念，逐步解說 Toolkit for Visual Studio 公開的 Amazon S3 功能。

Note

若要使用此工具，您的 IAM 政策必須授予 `s3:GetBucketAcl`、`s3:GetBucket` 和 `s3:ListBucket` 動作的許可。如需詳細資訊，請參閱 [IAM AWS 政策概觀](#)。

建立 Amazon S3 儲存貯體

儲存貯體是 Amazon S3 中最基本的儲存單位。

建立 S3 儲存貯體

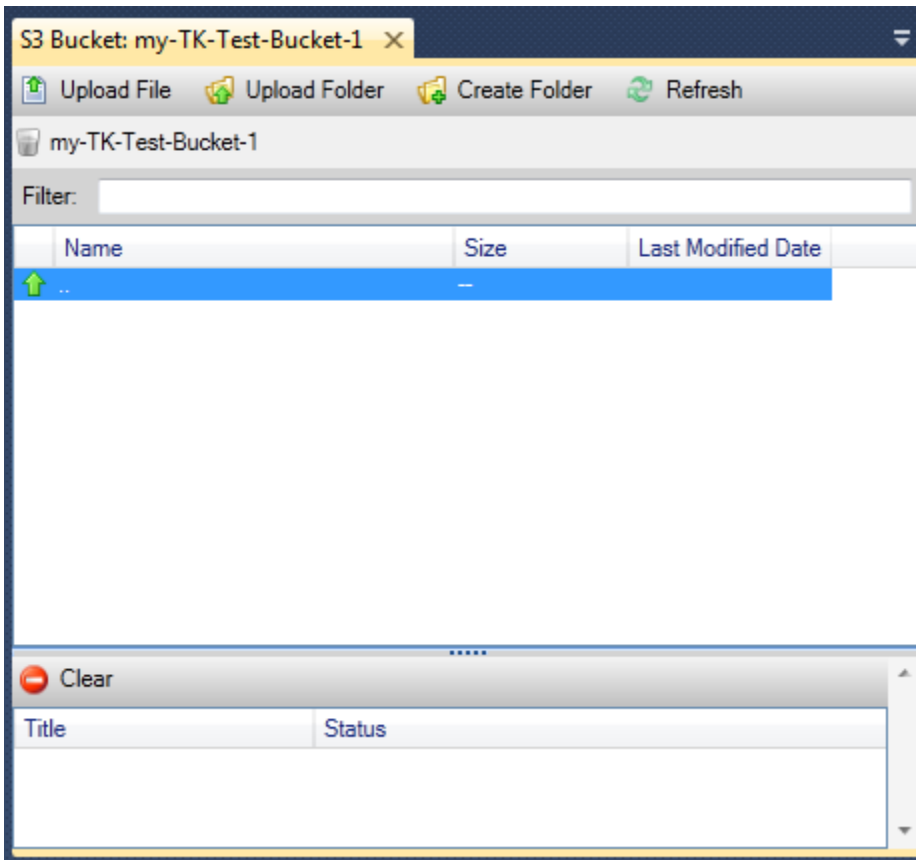
1. 在 AWS Explorer 中，開啟 Amazon S3 節點的內容（按一下滑鼠右鍵）選單，然後選擇建立儲存貯體。
2. 在建立儲存貯體對話方塊中，輸入儲存貯體的名稱。儲存貯體名稱必須是唯一的 AWS。如需其他限制條件的資訊，請前往 [Amazon S3 文件](#)。
3. 選擇確定。

從 AWS Explorer 管理 Amazon S3 儲存貯體

在 AWS Explorer 中，當您開啟 Amazon S3 儲存貯體的內容（按一下滑鼠右鍵）選單時，可以使用下列操作。

瀏覽

顯示儲存貯體中包含的物件檢視。從這裡，您可以從本機電腦建立資料夾或上傳檔案或整個目錄和資料夾。下方窗格會顯示有關上傳程序的狀態訊息。若要清除這些訊息，請選擇清除圖示。您也可以直接在 AWS Explorer 中按兩下儲存貯體名稱來存取儲存貯體的此檢視。



屬性

顯示對話方塊，您可以在其中執行下列動作：

- 將範圍為的 Amazon S3 許可設定為：
 - 您作為儲存貯體擁有者。
 - 所有已通過身分驗證的使用者 AWS。
 - 具有網際網路存取的每個人。
- 開啟儲存貯體的記錄。
- 使用 Amazon Simple Notification Service (Amazon SNS) 設定通知，以便在您使用低冗餘儲存 (RRS) 時，發生資料遺失時會收到通知。RRS 是一種 Amazon S3 儲存選項，提供比標準儲存更低的耐用性，但成本較低。如需詳細資訊，請參閱 [S3 FAQs](#)。
- 使用儲存貯體中的資料建立靜態網站。

政策

可讓您為儲存貯體設定 AWS Identity and Access Management (IAM) 政策。如需詳細資訊，請前往 [IAM 文件](#) 和 [IAM](#) 和 [S3](#) 的使用案例。

建立預先簽章的 URL

可讓您產生可分發的限時 URL，以提供儲存貯體內容的存取權。如需詳細資訊，請參閱 [如何建立預先簽章的 URL](#)。

檢視分段上傳

可讓您檢視分段上傳。Amazon S3 支援將大型物件上傳分成數個部分，讓上傳程序更有效率。如需詳細資訊，請前往 [S3 文件中的分段上傳討論](#)。

刪除

可讓您刪除儲存貯體。您僅能刪除空的儲存貯體。

將檔案和資料夾上傳至 Amazon S3

您可以使用 AWS Explorer 將檔案或整個資料夾從本機電腦傳輸到任何儲存貯體。

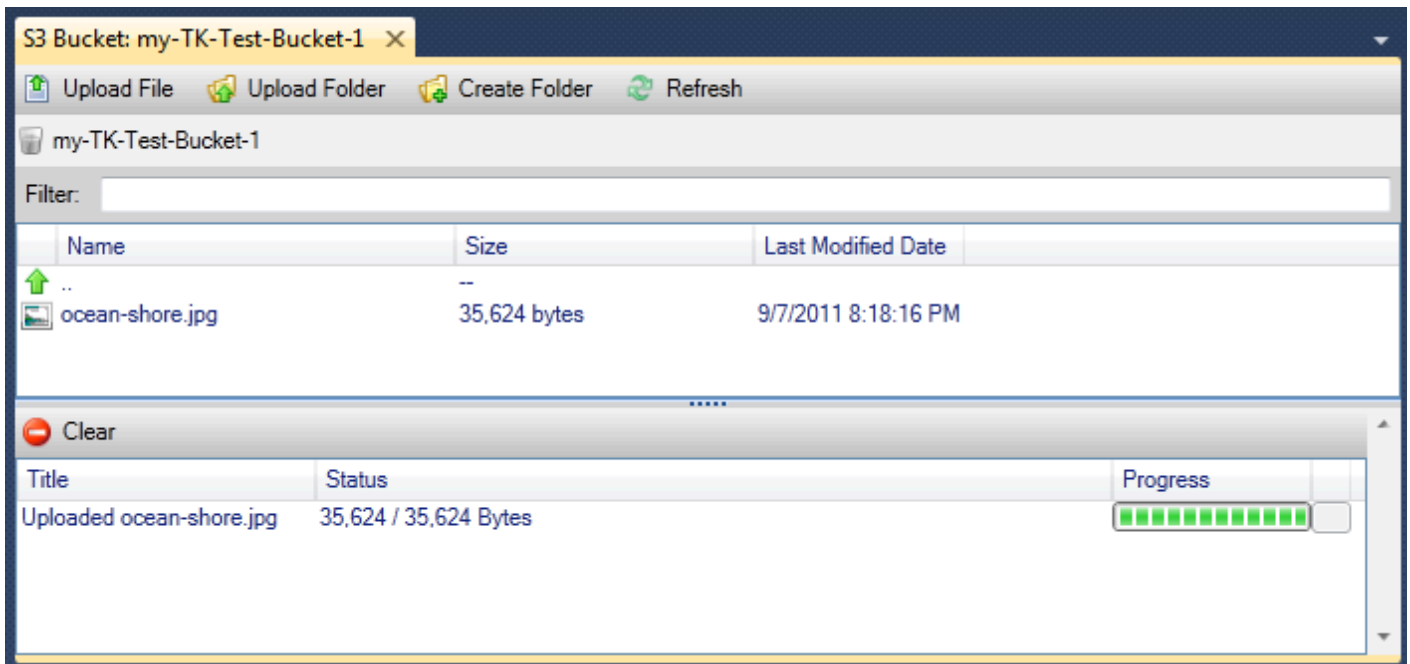
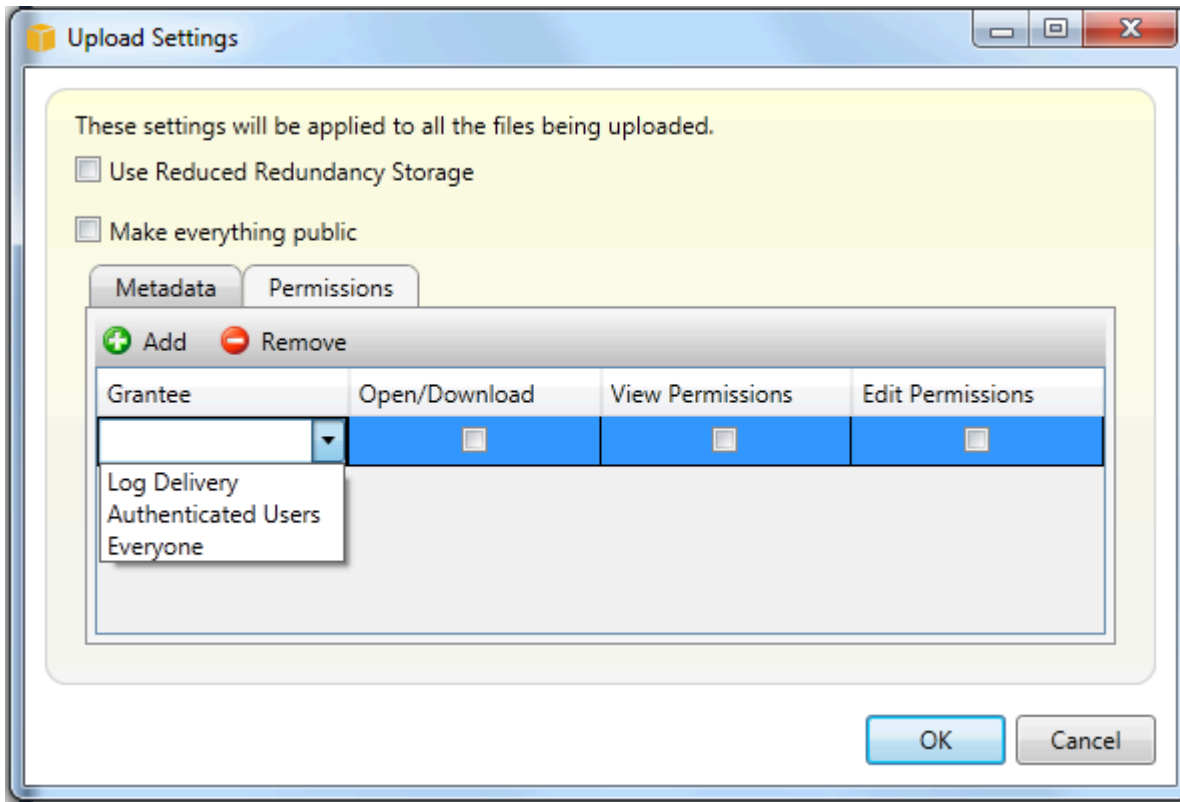
Note

如果您上傳與 Amazon S3 儲存貯體中已存在之檔案或資料夾名稱相同的檔案或資料夾，則上傳的檔案會覆寫現有的檔案，而不會出現警告。

將檔案上傳至 S3

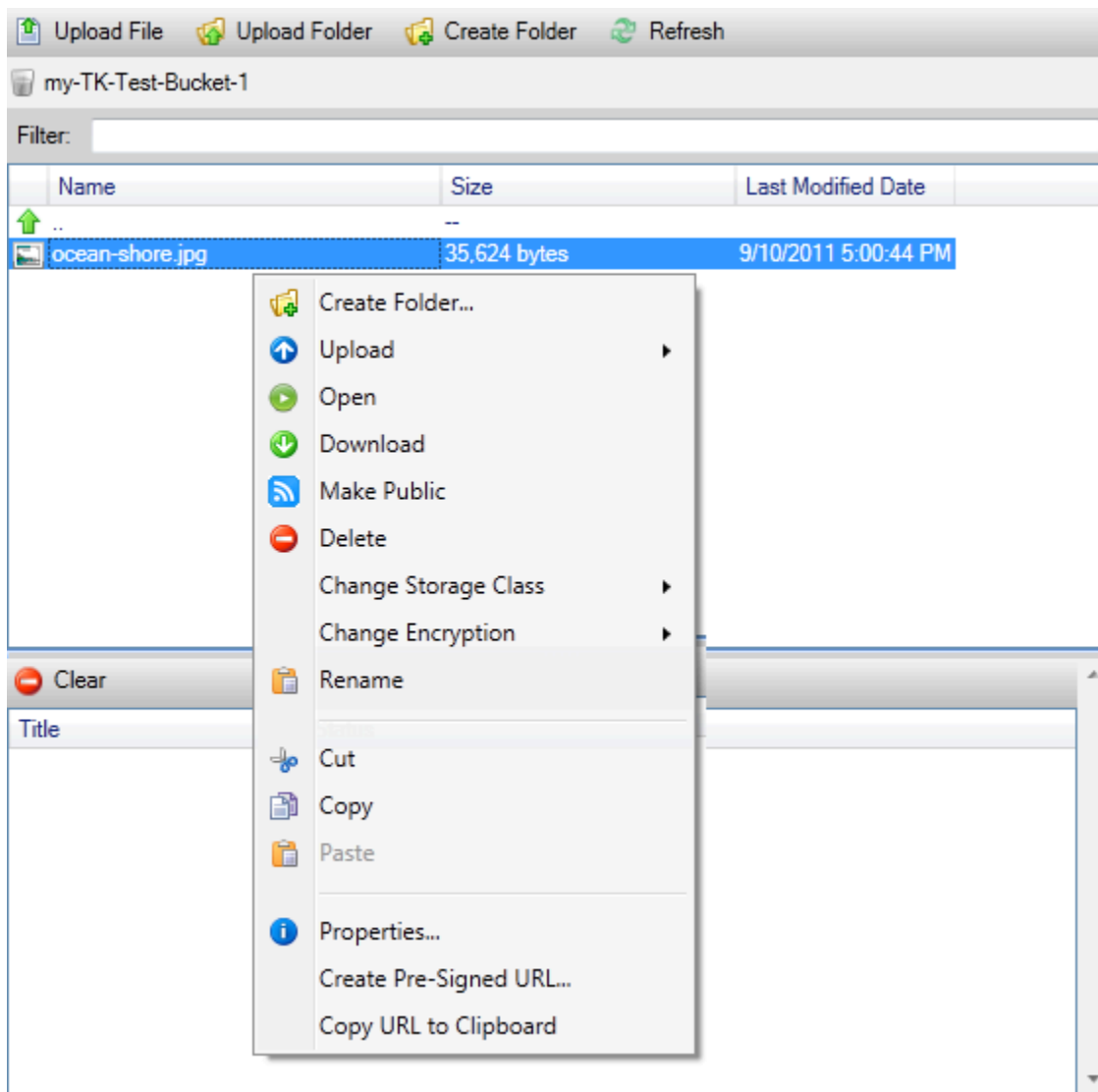
1. 在 AWS Explorer 中，展開 Amazon S3 節點，按兩下儲存貯體或開啟儲存貯體的內容（按一下滑鼠右鍵）選單，然後選擇瀏覽。
2. 在儲存貯體的瀏覽檢視中，選擇上傳檔案或上傳資料夾。
3. 在檔案開啟對話方塊中，導覽至要上傳的檔案，選擇它們，然後選擇開啟。如果您要上傳資料夾，請導覽至 並選擇該資料夾，然後選擇開啟。

上傳設定對話方塊可讓您設定所上傳檔案或資料夾的中繼資料和許可。選取全部公開核取方塊等同於將開啟/下載許可設定為每個人。您可以選取 [選項](#)，以針對上傳的檔案使用 [降低備援儲存體](#)。



來自 AWS Toolkit for Visual Studio 的 Amazon S3 檔案操作

如果您在 Amazon S3 檢視中選擇檔案並開啟內容（按一下滑鼠右鍵）選單，您可以對檔案執行各種操作。



建立資料夾

可讓您在目前的儲存貯體中建立資料夾。(等同於選擇建立資料夾連結。)

上傳

可讓您上傳檔案或資料夾。(等同於選擇上傳檔案或上傳資料夾連結。)

Open

嘗試在預設瀏覽器中開啟選取的檔案。根據檔案類型和預設瀏覽器的功能，可能不會顯示檔案。它可能只是由您的瀏覽器下載。

下載

開啟 Folder-Tree 對話方塊，讓您下載選取的檔案。

公開

將所選檔案的許可設定為開啟/下載和所有人。(等同於選取「上傳設定」對話方塊上的「全部設為公有」核取方塊。)

刪除

刪除選取的檔案或資料夾。您也可以選擇檔案或資料夾並按 來刪除這些檔案或資料夾Delete。

變更儲存體方案

將儲存類別設定為標準或降低備援儲存 (RRS)。若要檢視目前的儲存類別設定，請選擇屬性。

變更加密

可讓您在 檔案上設定伺服器端加密。若要檢視目前的加密設定，請選擇屬性。

Rename (重新命名)

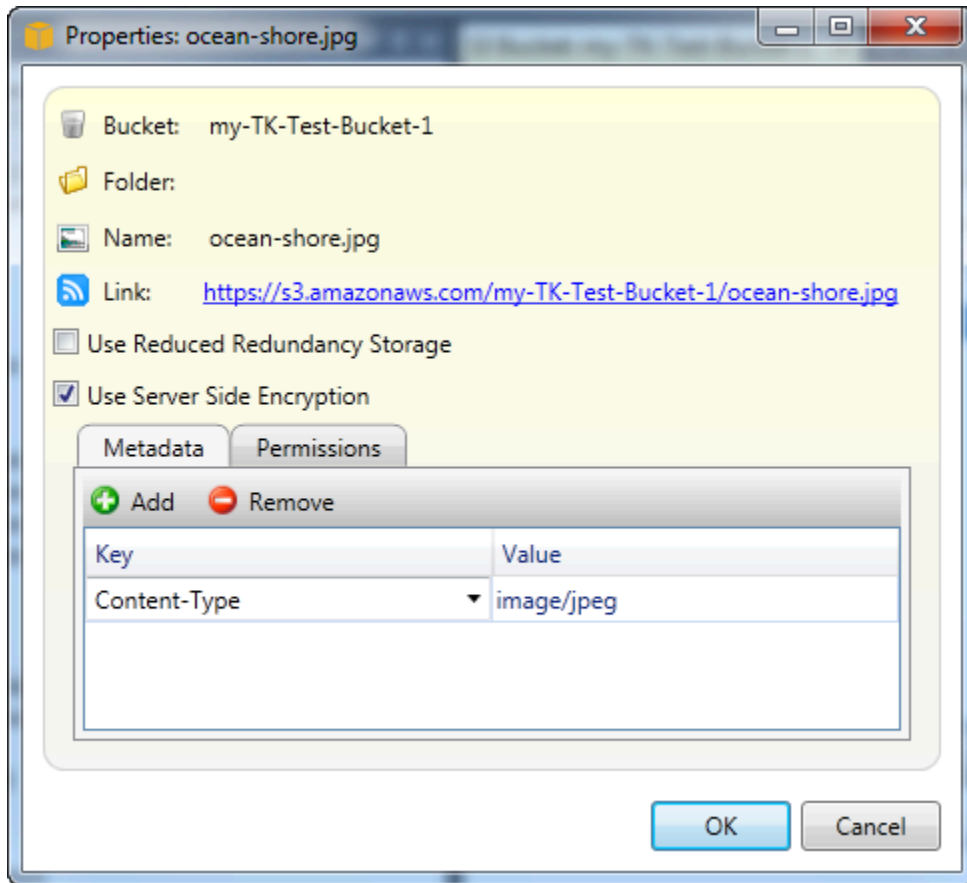
可讓您重新命名檔案。您無法重新命名資料夾。

剪下 | 複製 | 貼上

可讓您在資料夾之間或儲存貯體之間剪下、複製和貼上檔案或資料夾。

屬性

顯示對話方塊，可讓您設定檔案的中繼資料和許可，以及在降低冗餘儲存 (RRS) 和標準之間切換檔案的儲存，以及設定檔案的伺服器端加密。此對話方塊也會顯示 檔案的 https 連結。如果您選擇此連結，Toolkit for Visual Studio 會在您的預設瀏覽器中開啟 檔案。如果您擁有將檔案設定為 Open/Download 和 Everyone 的許可，其他人將可以透過此連結存取檔案。我們建議您建立和分發預先簽章URLs，而不是分發此連結。



建立預先簽章的 URL

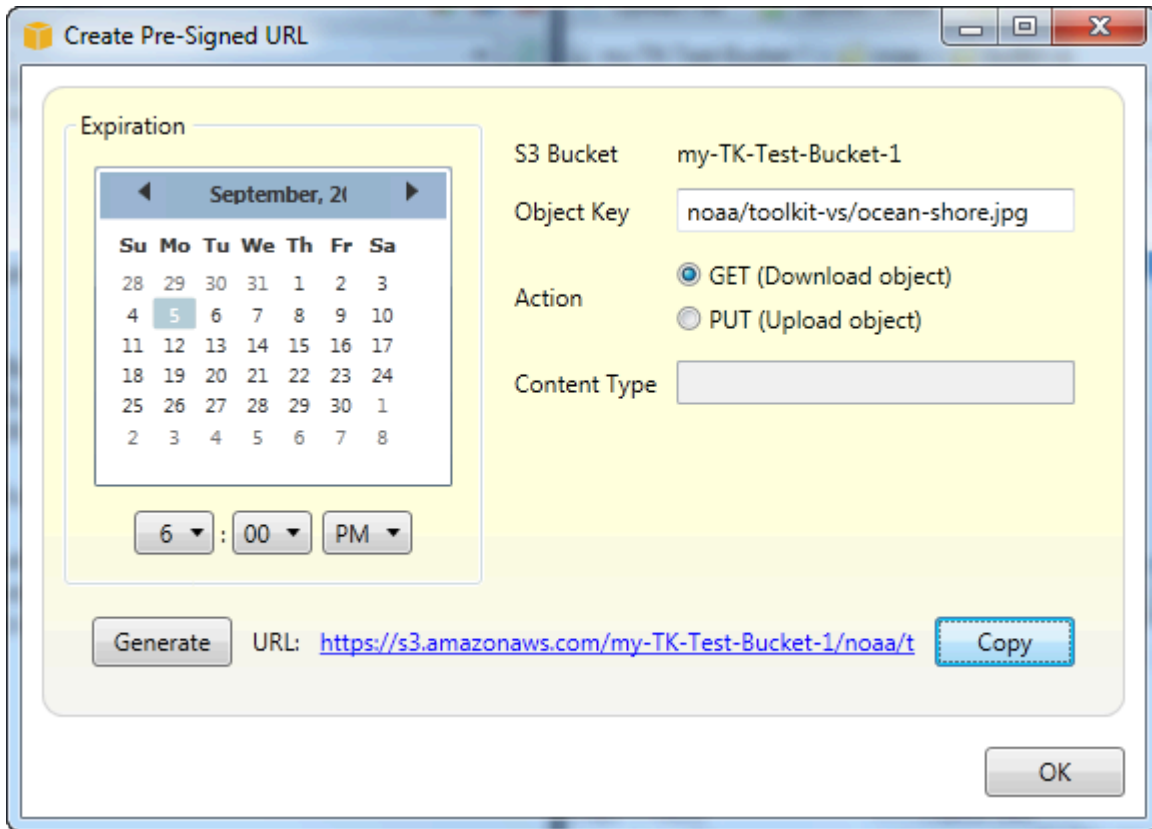
可讓您建立限時預先簽章的 URL，以便其他人存取您在 Amazon S3 上存放的內容。

如何建立預先簽章的 URL

您可以為儲存貯體或儲存貯體中的檔案建立預先簽章的 URL。其他人接著可以使用此 URL 來存取儲存貯體或檔案。URL 會在您建立 URL 時指定的一段時間後過期。

建立預先簽章的 URL

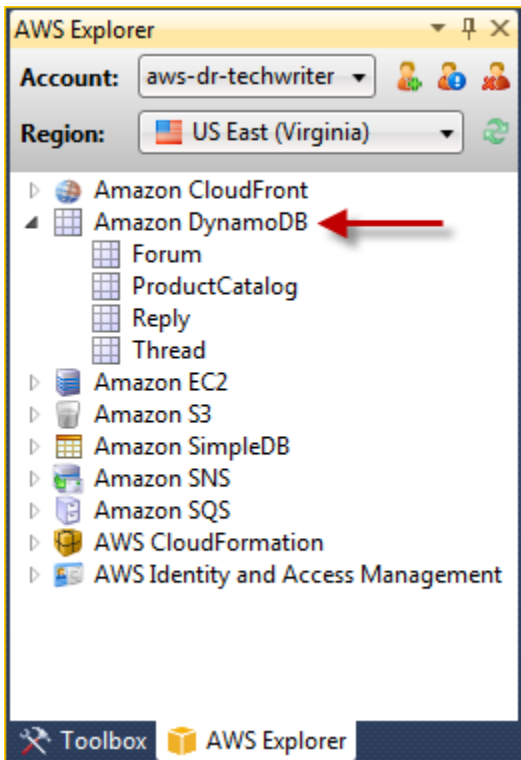
1. 在建立預先簽章的 URL 對話方塊中，設定 URL 的過期日期和時間。預設設定是從目前時間起算一小時。
2. 選擇產生按鈕。
3. 若要複製剪貼簿連結，請選擇複製。



從 AWS Explorer 使用 DynamoDB

Amazon DynamoDB 是一種快速、可輕鬆擴展、高度可用、經濟實惠、非關聯式資料庫服務。DynamoDB 會移除資料儲存體的傳統可擴展性限制，同時維持低延遲和可預測的效能。Toolkit for Visual Studio 提供在開發環境中使用 DynamoDB 的功能。如需 DynamoDB 的詳細資訊，請參閱 Amazon Web Services 網站上的 [DynamoDB](#)。

在 Toolkit for Visual Studio 中，AWS Explorer 會顯示與作用中相關聯的所有 DynamoDB 資料表 AWS 帳戶。



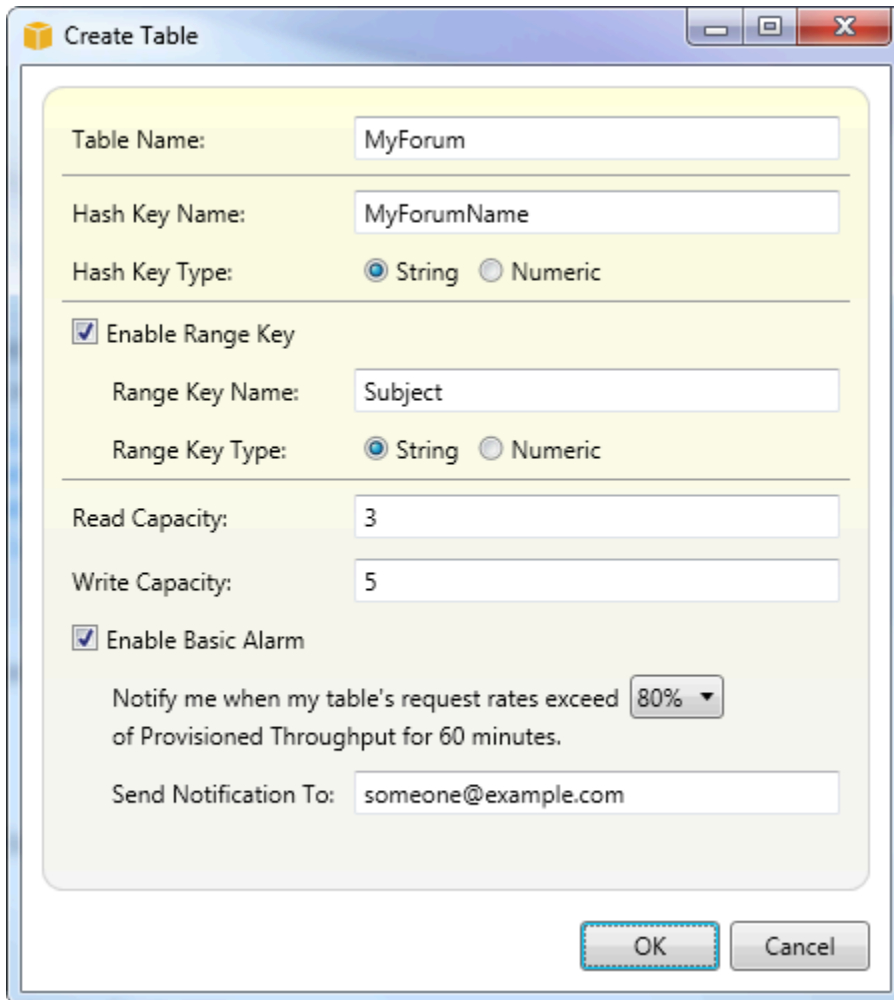
建立 DynamoDB 資料表

您可以使用 Toolkit for Visual Studio 來建立 DynamoDB 資料表。

在 AWS Explorer 中建立資料表

1. 在 AWS Explorer 中，開啟 Amazon DynamoDB 的內容（按一下滑鼠右鍵）選單，然後選擇建立資料表。
2. 在建立資料表精靈的資料表名稱中，輸入資料表的名稱。
3. 在雜湊金鑰名稱欄位中，輸入主要雜湊金鑰屬性，然後從雜湊金鑰類型按鈕中選擇雜湊金鑰類型。DynamoDB 使用主索引鍵屬性建置未排序的雜湊索引，並使用範圍主索引鍵屬性建置選用的排序範圍索引。如需主要雜湊金鑰屬性的詳細資訊，請前往《Amazon DynamoDB 開發人員指南》中的[主要金鑰](#)一節。
4. （選用）選取啟用範圍金鑰。在範圍索引鍵名稱欄位中，輸入範圍索引鍵屬性，然後從範圍索引鍵類型按鈕中選擇範圍索引鍵類型。
5. 在讀取容量欄位中，輸入讀取容量單位的數量。在寫入容量欄位中，輸入寫入容量單位的數量。您必須指定至少三個讀取容量單位和五個寫入容量單位。如需讀取和寫入容量單位的詳細資訊，請前往 [DynamoDB 中的佈建輸送量](#)。
6. （選用）選取啟用基本警示，以便在資料表的請求率過高時提醒您。選擇傳送提醒之前，每 60 分鐘必須超過的佈建輸送量百分比。在傳送通知至 中，輸入電子郵件地址。

7. 按一下確定以建立資料表。



The screenshot shows a 'Create Table' dialog box with the following configuration:

- Table Name: MyForum
- Hash Key Name: MyForumName
- Hash Key Type: String (selected)
- Enable Range Key
- Range Key Name: Subject
- Range Key Type: String (selected)
- Read Capacity: 3
- Write Capacity: 5
- Enable Basic Alarm
- Notify me when my table's request rates exceed 80% of Provisioned Throughput for 60 minutes.
- Send Notification To: someone@example.com

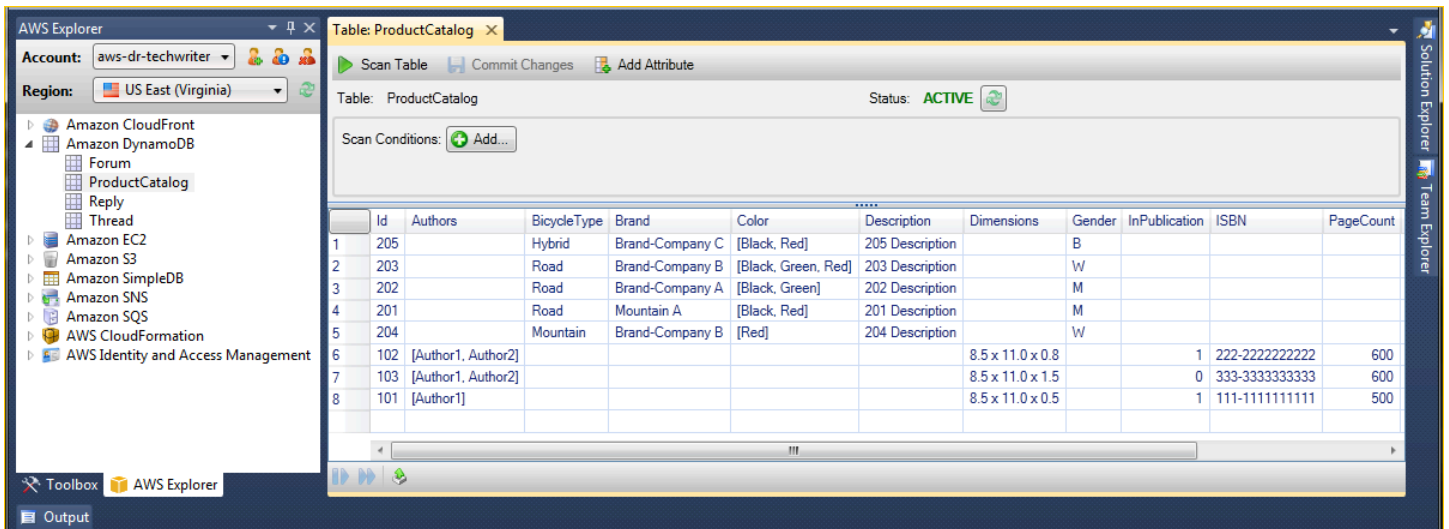
Buttons: OK, Cancel

如需 DynamoDB 資料表的詳細資訊，請前往[資料模型概念 - 資料表、項目和屬性](#)。

將 DynamoDB 資料表檢視為網格

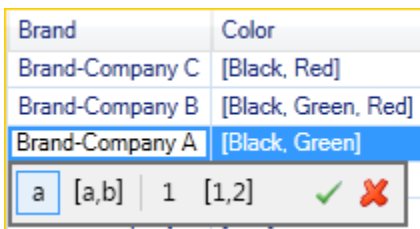
若要開啟其中一個 DynamoDB 資料表的網格檢視，請在 AWS Explorer 中按兩下對應至資料表的子節點。從網格檢視中，您可以查看存放在該資料表中的項目、屬性和值。每一列都會對應到資料表中的某個項目。資料表欄會對應到屬性。資料表中的每個儲存格都會保存與該項目之屬性相關聯的值。

屬性可以擁有屬於字串或數字的值。有些屬性的值包含字串或數字的「集合」。集合值會顯示為逗號分隔的清單，並且用方括號括住。

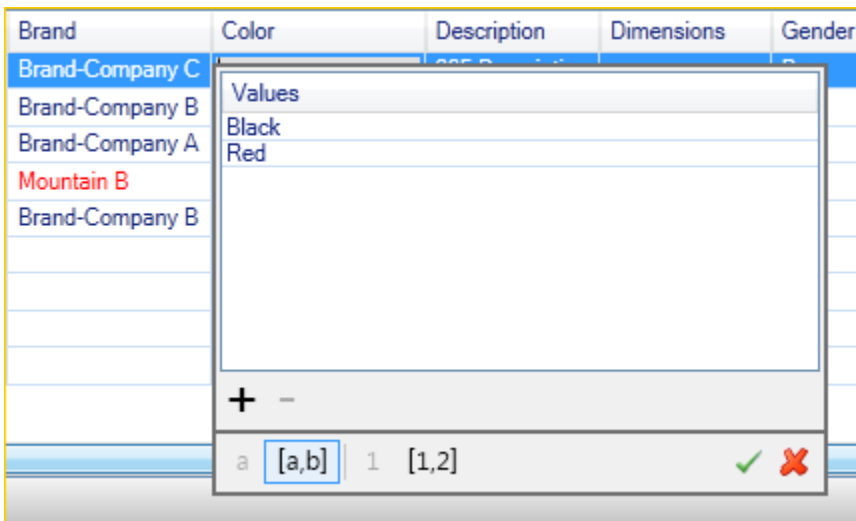


編輯和新增屬性和值

透過按兩下儲存格，您可以編輯項目對應屬性的值。對於集合值屬性，您也可以新增或刪除其中的個別值。



除了變更屬性的值之外，您還可以變更屬性的值格式，但有一些限制。例如，任何數值都能轉換為字串值。如果您有字串值，其內容為數字，例如 125，儲存格編輯器可讓您將值的格式從字串轉換為數字。您也可以將單一值轉換為集合值。但是，通常您不能將集合值轉換成為單一值；唯一的例外情況，就是當該集合值其實僅有一個元素。

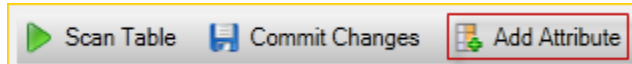


編輯屬性值後，請選擇綠色核取記號以確認您的變更。如果您想要捨棄變更，請選擇紅色 X。

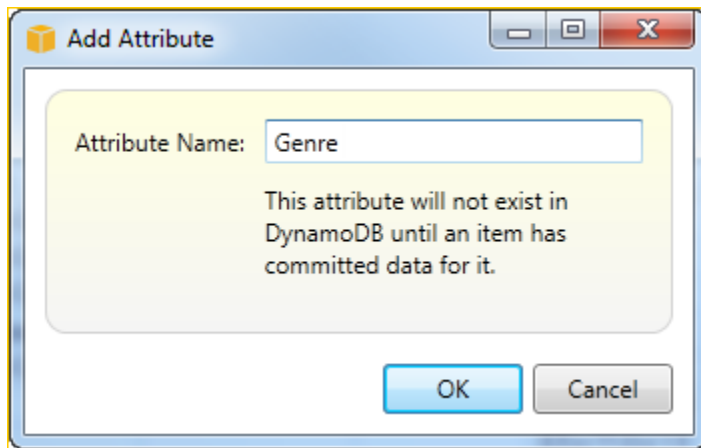
在您確認變更後，屬性值會以紅色顯示。這表示屬性已更新，但新值尚未寫回 DynamoDB 資料庫。若要將變更寫回 DynamoDB，請選擇遞交變更。若要捨棄變更，請選擇掃描資料表，並在工具組詢問您是否要在掃描前遞交變更時，選擇否。

新增 屬性

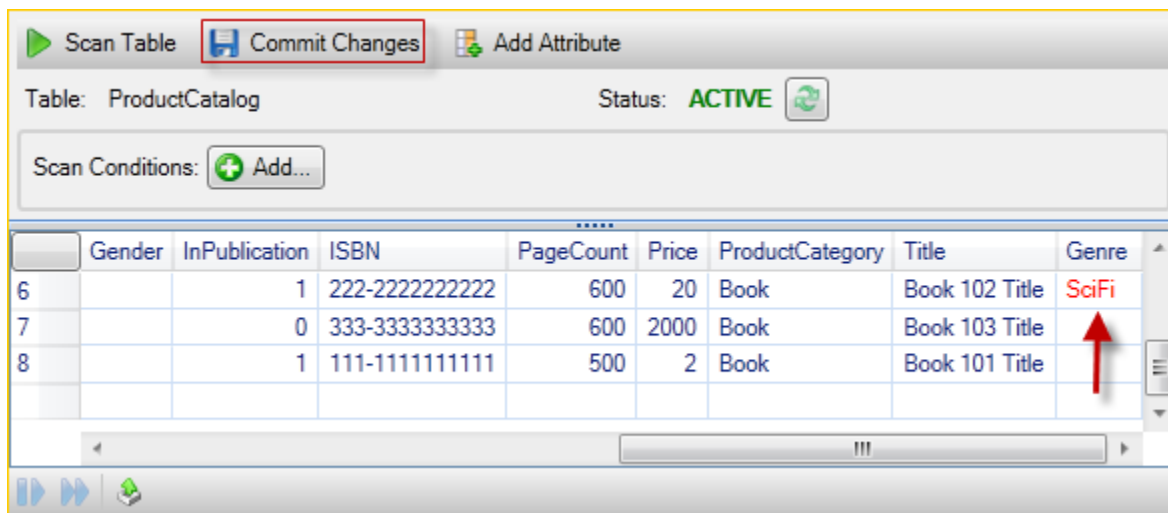
從網格檢視中，您也可以將屬性新增至資料表。若要新增屬性，請選擇新增屬性。



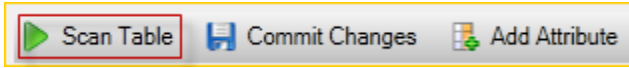
在新增屬性對話方塊中，輸入屬性的名稱，然後選擇確定。



若要讓新屬性成為資料表的一部分，您必須為至少一個項目新增值，然後選擇遞交變更按鈕。若要捨棄新的屬性，只需關閉資料表的網格檢視，而不選擇遞交變更。



掃描 DynamoDB 資料表



您可以從 Toolkit 對 DynamoDB 資料表執行掃描。進行掃描時，您可以定義一組條件，而掃描會從資料表中傳回符合您所設定條件的所有項目。掃描是昂貴的操作，應謹慎使用，以避免中斷資料表上較高優先順序的生產流量。如需使用掃描操作的詳細資訊，請前往 Amazon DynamoDB 開發人員指南。

從 AWS Explorer 對 DynamoDB 資料表執行掃描

1. 在網格檢視中，選擇掃描條件：新增按鈕。
2. 在掃描子句編輯器中，選擇要比對的屬性、應如何解譯屬性的值（字串、數字、設定值）、應如何比對（例如以或 Contains 開頭），以及應比對的常值。
3. 視需要為您的搜尋新增更多掃描子句。掃描只會傳回符合您所有掃描子句之條件的項目。比對字串值時，掃描會執行區分大小寫的比較。
4. 在網格檢視頂端的按鈕列上，選擇掃描資料表。

若要移除掃描子句，請選擇帶有每個子句右側的白線的紅色按鈕。

	Id	BicycleType	Brand	Color	Description	Gender	Price	ProductCategory	Title
1	202	Road	Brand-Company A	[Black, Green]	202 Description	M	200	Bicycle	21-Bike-202
2	201	Road	Mountain A	[Black, Red]	201 Description	M	100	Bicycle	18-Bike-201

若要返回包含所有項目的資料表檢視，請移除所有掃描子句，然後再次選擇掃描資料表。

為掃描結果編製分頁

檢視底部有三個按鈕。



前兩個藍色按鈕提供掃描結果的分頁。第一個按鈕會顯示額外的結果頁面。第二個按鈕會顯示額外的 10 頁結果。在此內容中，頁面等於 1 MB 的內容。

將掃描結果匯出至 CSV

第三個按鈕會將結果從目前的掃描匯出至 CSV 檔案。

AWS CodeCommit 搭配 Visual Studio Team Explorer 使用

您可以使用 AWS Identity and Access Management (IAM) 使用者帳戶來建立 Git 登入資料，並使用它們從 Team Explorer 中建立和複製儲存庫。

的登入資料類型 AWS CodeCommit

大多數 AWS Toolkit for Visual Studio 使用者都知道設定包含其存取和私密金鑰的 AWS 登入資料設定檔。這些登入資料設定檔用於 Toolkit for Visual Studio，以啟用對服務 APIs 呼叫，例如，列出 AWS Explorer 中的 Amazon S3 儲存貯體或啟動 Amazon EC2 執行個體。AWS CodeCommit 與 Team Explorer 的整合也會使用這些登入資料設定檔。不過，若要使用 Git 本身，您需要額外的登入資料，特別是 HTTPS 連線的 Git 登入資料。您可以在使用者指南中的[使用 Git 登入資料設定 HTTPS 使用者（使用者名稱和密碼）](#)中閱讀這些登入 AWS CodeCommit 資料。

您只能為 IAM 使用者帳戶建立 AWS CodeCommit 的 Git 登入資料。您無法為根帳戶建立它們。您最多可以為服務建立兩組這些登入資料，雖然您可以將一組登入資料標記為非作用中，但非作用中組仍會計入兩組的限制。請注意，您可以隨時刪除並重新建立登入資料。當您 AWS CodeCommit 在 Visual Studio 中使用時，您的傳統 AWS 登入資料會用於使用服務本身，例如當您建立和列出儲存庫時。使用中託管的實際 Git 儲存庫時 AWS CodeCommit，您可以使用 Git 登入資料。

作為支援的一部分 AWS CodeCommit，Toolkit for Visual Studio 會自動為您建立和管理這些 Git 登入資料，並將其與您的 AWS 登入資料設定檔建立關聯。您不需要擔心擁有正確的登入資料集，即可在 Team Explorer 中執行 Git 操作。使用 AWS 登入資料描述檔連線至 Team Explorer 後，每當您使用 Git 遠端時，會自動使用相關聯的 Git 登入資料。

連線至 AWS CodeCommit

當您在 Visual Studio 2015 或更新版本中開啟 Team Explorer 視窗時，您會在管理連線的託管服務供應商區段中看到 AWS CodeCommit 項目。



AWS CodeCommit
Amazon, Inc.

AWS CodeCommit is a fully-managed source control service that makes it easy for companies to host secure and highly scalable private Git repositories.

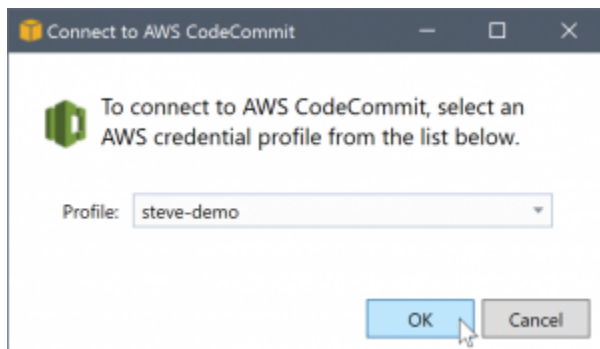
[Connect...](#)

[Sign up](#)

選擇註冊會在瀏覽器視窗中開啟 Amazon Web Services 首頁。當您選擇 Connect 時，會發生什麼情況取決於 Toolkit for Visual Studio 是否可以找到具有 AWS 存取和私密金鑰的登入資料設定檔，讓它 AWS 代表您呼叫。當 Toolkit for Visual Studio 找不到任何本機儲存的登入資料時，您可能已使用 IDE 中顯示的新入門頁面來設定登入資料設定檔。或者，您可能已經使用 Toolkit for Visual Studio、或 AWS Tools for Windows PowerShell，AWS CLI 並且已有 AWS 登入資料設定檔可供 Toolkit for Visual Studio 使用。

當您選擇連線時，Toolkit for Visual Studio 會開始尋找要在連線中使用的登入資料設定檔的程序。如果 Toolkit for Visual Studio 找不到登入資料設定檔，它會開啟對話方塊，邀請您輸入的存取和私密金鑰 AWS 帳戶。我們強烈建議您使用 IAM 使用者帳戶，而不是您的根登入資料。此外，如前所述，您最終需要的 Git 登入資料只能為 IAM 使用者建立。提供存取和私密金鑰並建立登入資料設定檔後，即可使用 Team Explorer 和 AWS CodeCommit 之間的連線。

如果 Toolkit for Visual Studio 找到多個 AWS 登入資料設定檔，系統會提示您選取要在 Team Explorer 中使用的帳戶。



如果您只有一個登入資料設定檔，Toolkit for Visual Studio 會略過設定檔選擇對話方塊，並立即連線：

當 Team Explorer 和 AWS CodeCommit 透過您的登入資料設定檔建立連線時，邀請對話方塊會關閉，並顯示連線面板。

[Manage Connections](#) ▾

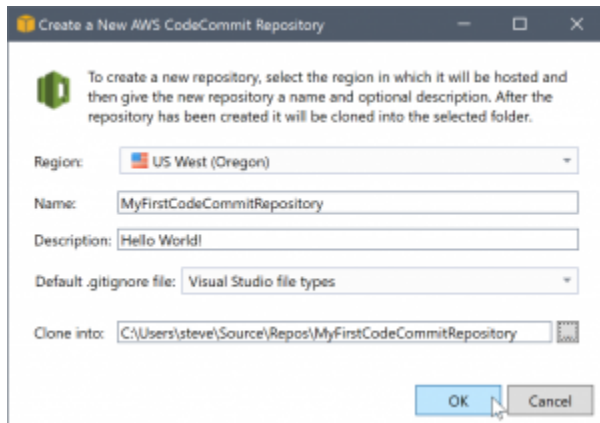
▲ AWS CodeCommit
[Clone](#) | [Create](#) | [Sign out steve-demo](#)

由於您沒有在本機複製的儲存庫，因此面板只會顯示您可以執行的操作：複製、建立和登出。如同其他供應商，AWS CodeCommit 在 Team Explorer 中，在任何指定時間只能繫結至單一 AWS 登入資料設定檔。若要切換帳戶，您可以使用登出來移除連線，以便使用不同的帳戶啟動新的連線。

現在您已建立連線，您可以按一下建立連結來建立儲存庫。

建立儲存庫

當您按一下建立連結時，會開啟建立新 AWS CodeCommit 儲存庫對話方塊。



AWS CodeCommit 儲存庫是依區域組織，因此您可以在區域中選取要託管儲存庫的區域。清單包含 AWS CodeCommit 支援的所有區域。您提供新儲存庫的名稱（必要）和描述（選用）。

對話方塊的預設行為是使用儲存庫名稱（當您輸入名稱時，資料夾位置也會更新）來為新儲存庫加上資料夾位置。若要使用不同的資料夾名稱，請在輸入儲存庫名稱完成後，編輯複製到資料夾路徑。

您也可以選擇自動建立儲存庫的初始 .gitignore 檔案。AWS Toolkit for Visual Studio 提供 Visual Studio 檔案類型的內建預設值。您也可以選擇沒有檔案，或使用您想要在儲存庫之間重複使用的自訂現有檔案。只要在清單中選取使用自訂，然後導覽至要使用的自訂檔案。

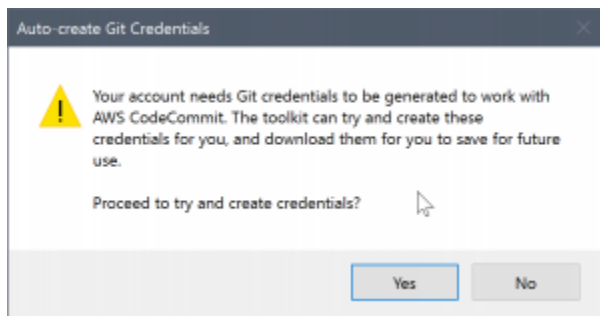
擁有儲存庫名稱和位置後，您就可以按一下確定並開始建立儲存庫。Toolkit for Visual Studio 會請求服務建立儲存庫，然後在本機複製新儲存庫，如果您使用的是 .gitignore 檔案，請新增初始遞交。此時您開始使用 Git 遠端，因此 Toolkit for Visual Studio 現在需要存取先前所述的 Git 登入資料。

設定 Git 登入資料

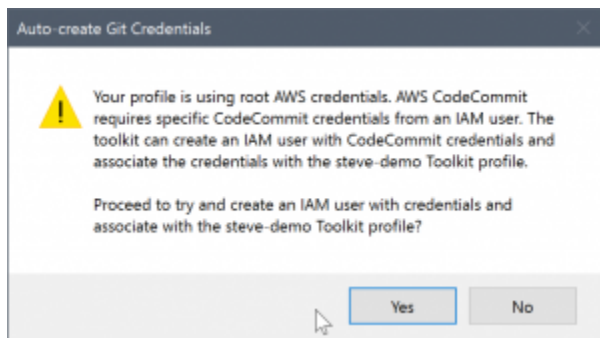
此時，您一直使用 AWS 存取和私密金鑰來請求服務建立您的儲存庫。現在您需要使用 Git 本身來執行實際的複製操作，Git 不了解 AWS 存取和私密金鑰。反之，您需要提供使用者名稱和密碼登入資料給 Git，以便在與遠端的 HTTPS 連線上使用。

如設定 Git 登入資料所述，您要使用的 Git 登入資料必須與 IAM 使用者相關聯。您無法為根登入資料產生它們。您應該一律將 AWS 登入資料設定檔設定為包含 IAM 使用者存取和私密金鑰，而不是根金鑰。Toolkit for Visual Studio 可以嘗試 AWS CodeCommit 為您設定 Git 登入資料，並將其與您先前在 Team Explorer 中用來連接的 AWS 登入資料設定檔建立關聯。

當您在建立新 AWS CodeCommit 儲存庫對話方塊中選擇確定並成功建立儲存庫時，Toolkit for Visual Studio 會檢查在 Team Explorer 中連接的 AWS 登入資料設定檔，以判斷的 Git 登入資料是否存在，AWS CodeCommit 並在本機與設定檔相關聯。若是如此，Toolkit for Visual Studio 會指示 Team Explorer 在新的儲存庫上開始複製操作。如果 Git 登入資料無法在本機使用，Toolkit for Visual Studio 會檢查 Team Explorer 中用於連線的帳戶登入資料的類型。如果登入資料適用於 IAM 使用者，如我們建議，會顯示下列訊息。

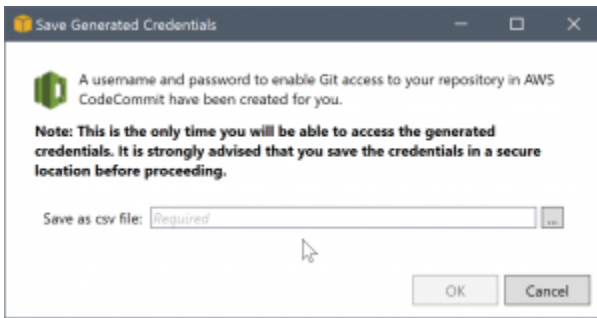


如果登入資料是根登入資料，則會改為顯示下列訊息。



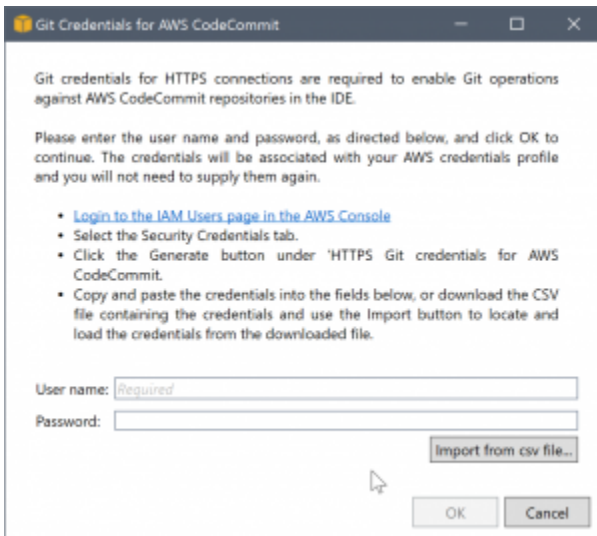
在這兩種情況下，Toolkit for Visual Studio 都會提供來嘗試執行工作，為您建立必要的 Git 登入資料。在第一個案例中，只需要為 IAM 使用者建立一組 Git 登入資料。使用根帳戶時，Toolkit for Visual Studio 會先嘗試建立 IAM 使用者，然後繼續為該新使用者建立 Git 登入資料。如果 Toolkit for Visual Studio 必須建立新的使用者，它會將 AWS CodeCommit 進階使用者受管政策套用至該新使用者帳戶。此政策僅允許存取，AWS CodeCommit 並允許使用執行所有操作，但刪除儲存庫 AWS CodeCommit 除外。

建立登入資料時，您只能檢視一次。因此，Toolkit for Visual Studio 會提示您將新建立的登入資料儲存為 .csv 檔案，然後再繼續。



這也是我們強烈建議的內容，請務必將它們儲存到安全的位置！

在某些情況下，Toolkit for Visual Studio 可能無法自動建立登入資料。例如，您可能已建立 AWS CodeCommit (二) 的 Git 登入資料集數目上限，或者您可能沒有足夠的程式設計權限，讓 Toolkit for Visual Studio 為您執行工作 (如果您以 IAM 使用者身分登入)。在這些情況下，您可以登入 AWS 管理主控台來管理登入資料，或從管理員取得登入資料。然後，您可以在 Toolkit for Visual Studio 顯示的 Git 登入 AWS CodeCommit 資料對話方塊中輸入它們。

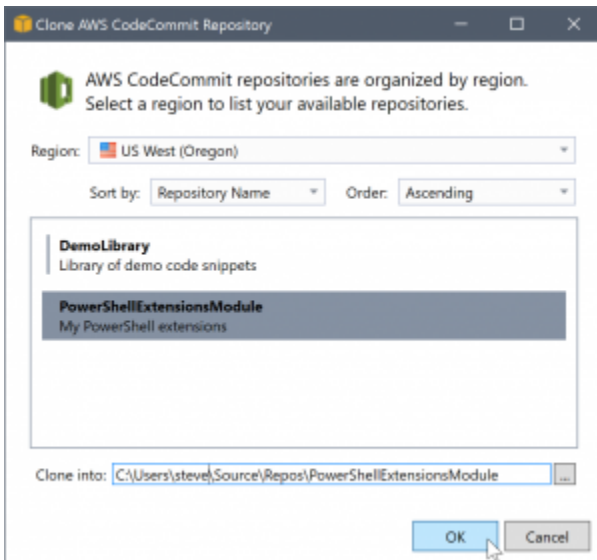


現在 Git 的登入資料已可用，新儲存庫的複製操作會繼續進行 (請參閱 Team Explorer 中操作的進度指示)。如果您選擇套用預設 .gitignore 檔案，則會將其遞交至具有「初始遞交」註解的儲存庫。

這就是設定登入資料以及在 Team Explorer 中建立儲存庫。備妥必要的登入資料後，您未來建立新儲存庫時看到的就是建立新 AWS CodeCommit 儲存庫對話方塊本身。

複製儲存庫

若要複製現有的儲存庫，請返回 Team Explorer AWS CodeCommit 中的連線面板。按一下複製連結以開啟複製 AWS CodeCommit 儲存庫對話方塊，然後選取要複製的儲存庫，以及您要放置在磁碟上的位置。



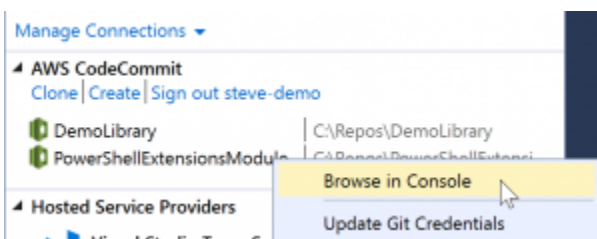
選擇區域後，Toolkit for Visual Studio 會查詢服務，以探索該區域中可用的儲存庫，並將它們顯示在對話方塊的中央清單部分。也會顯示每個儲存庫的名稱和選用描述。您可以重新排序清單，依儲存庫名稱或上次修改日期排序，並依遞增或遞減順序排序。

選取儲存庫後，您可以選擇要複製的位置。這會預設為與其他外掛程式中使用的儲存庫位置相同的 Team Explorer，但您可以瀏覽或輸入任何其他位置。根據預設，儲存庫名稱會加上所選路徑的尾碼。不過，如果您想要特定路徑，只要在選取資料夾之後編輯文字方塊即可。當您按一下 OK 時，方塊中的文字都會是您會找到複製儲存庫的資料夾。

選取儲存庫和資料夾位置後，請按一下確定以繼續複製操作。如同建立儲存庫一樣，您可以在 Team Explorer 中看到複製操作的進度。

使用 儲存庫

當您複製或建立儲存庫時，請注意連線的本機儲存庫會列在操作連結下的 Team Explorer 連線面板中。這些項目可讓您方便地存取儲存庫以瀏覽內容。只要在儲存庫上按一下滑鼠右鍵，然後選擇在主控台中瀏覽。



您也可以使用更新 Git 登入資料來更新與登入資料描述檔相關聯的預存 Git 登入資料。如果您已輪換登入資料，這會很有用。命令會開啟 Git 登入資料 AWS CodeCommit對話方塊，您可以在其中輸入或匯入新的登入資料。

儲存庫上的 Git 操作會如預期般運作。您可以進行本機遞交，並在準備好共用時，使用 Team Explorer 中的同步選項。由於 Git 登入資料已存放在本機，並與我們連線的 AWS 登入資料設定檔相關聯，因此系統不會提示我們再次提供這些登入資料以進行 AWS CodeCommit 遠端操作。

在 Visual Studio 中使用 CodeArtifact

AWS CodeArtifact 是一種全受管成品儲存庫服務，可讓組織輕鬆安全地存放和共用用於應用程式開發的軟體套件。您可以將 CodeArtifact 與熱門建置工具和套件管理員搭配使用，例如 NuGet 和 .NET Core CLI 和 Visual Studio。您也可以設定 CodeArtifact 從外部公有儲存庫提取套件，例如 [NuGet.org](https://www.nuget.org)

在 CodeArtifact 中，您的套件會儲存在儲存庫中，然後存放在網域中。使用 CodeArtifact 儲存庫 AWS Toolkit for Visual Studio 簡化 Visual Studio 的組態，讓您輕鬆地從 CodeArtifact 和 NuGet.org。

將 CodeArtifact 儲存庫新增為 NuGet 套件來源

若要使用 CodeArtifact 中的套件，您需要在 Visual Studio 的 NuGet Package Manager 中將儲存庫新增為 package 來源

將儲存庫新增為套件來源

1. 在 AWS Explorer 中，導覽至 AWS CodeArtifact 節點中的儲存庫。
2. 開啟您要新增之儲存庫的內容（按一下滑鼠右鍵）選單，然後選擇複製 NuGet 來源端點。
3. 在工具 > 選項選單中，導覽至 NuGet NuGet Package Manager 節點下方的套件來源。
4. 在套件來源中，選取加號 (+)、編輯名稱，然後貼上您先前在來源欄位中複製的 NuGet 來源端點 URL。
5. 選取新新增的套件來源旁的核取方塊以啟用它。

Note

建議您將 NuGet.org 的外部連線新增至 CodeArtifact，並在 Visual Studio 中停用 nuget.org 套件來源。使用外部連線時，從 NuGet.org 提取的所有相依性都會存放在 CodeArtifact 中。如果 NuGet.org 因任何原因而停機，您所需的套件仍然可用。如需外部連線的詳細資訊，請參閱 AWS CodeArtifact 《使用者指南》中的 [新增外部連線](#)。

6. 選擇確定以關閉選單。

如需搭配 Visual Studio 使用 CodeArtifact 的詳細資訊，請參閱AWS CodeArtifact 《使用者指南》中的[搭配 Visual Studio 使用 CodeArtifact](#)。

AWS Explorer 的 Amazon RDS

Amazon Relational Database Service (Amazon RDS) 是一種服務，可讓您在雲端中佈建和管理 SQL 關聯式資料庫系統。Amazon RDS 支援三種類型的資料庫系統：

- MySQL Community Edition
- Oracle Database Enterprise Edition
- Microsoft SQL Server (Express、Standard 或 Web Edition)

如需詳細資訊，請參閱 [Amazon RDS 使用者指南](#)。

這裡討論的許多功能也可以透過 Amazon RDS 的 [AWS 管理主控台](#) 使用。

主題

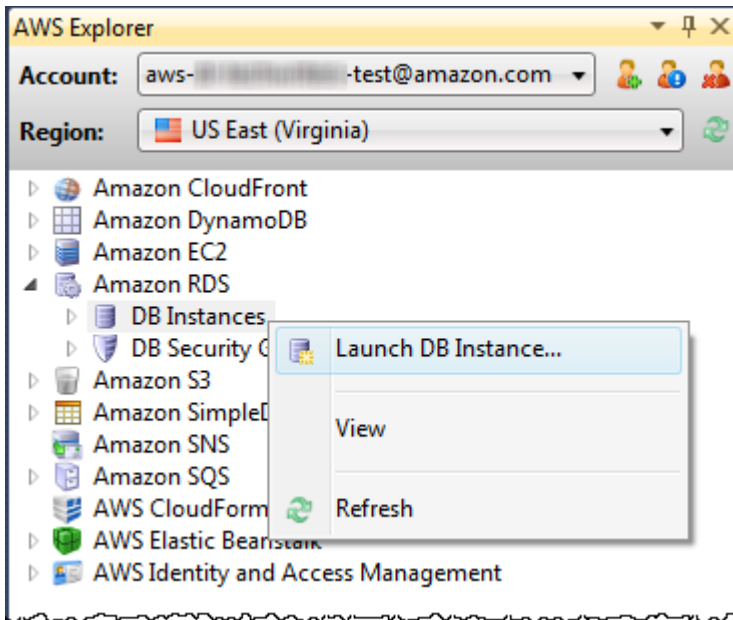
- [啟動 Amazon RDS 資料庫執行個體](#)
- [在 RDS 執行個體中建立 Microsoft SQL Server 資料庫](#)
- [Amazon RDS 安全群組](#)

啟動 Amazon RDS 資料庫執行個體

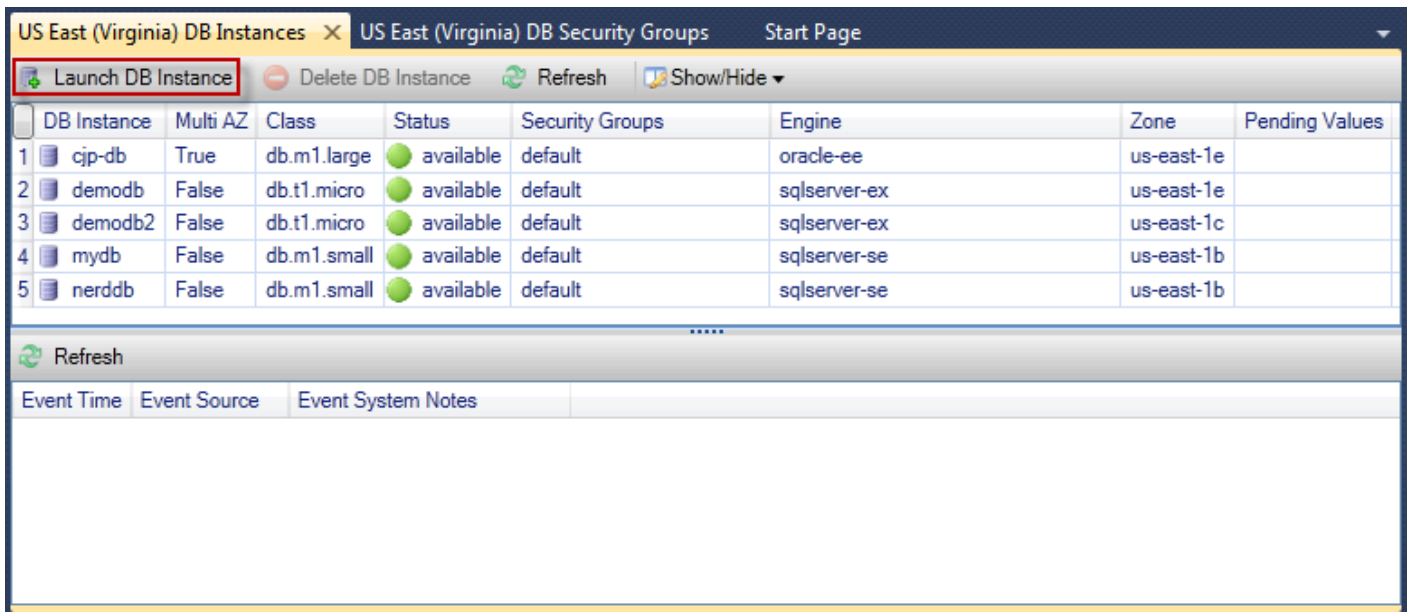
使用 AWS Explorer，您可以啟動 Amazon RDS 支援的任何資料庫引擎的執行個體。下列逐步解說顯示啟動 Microsoft SQL Server Standard Edition 執行個體的使用者體驗，但使用者體驗與所有支援的引擎相似。

啟動 Amazon RDS 執行個體

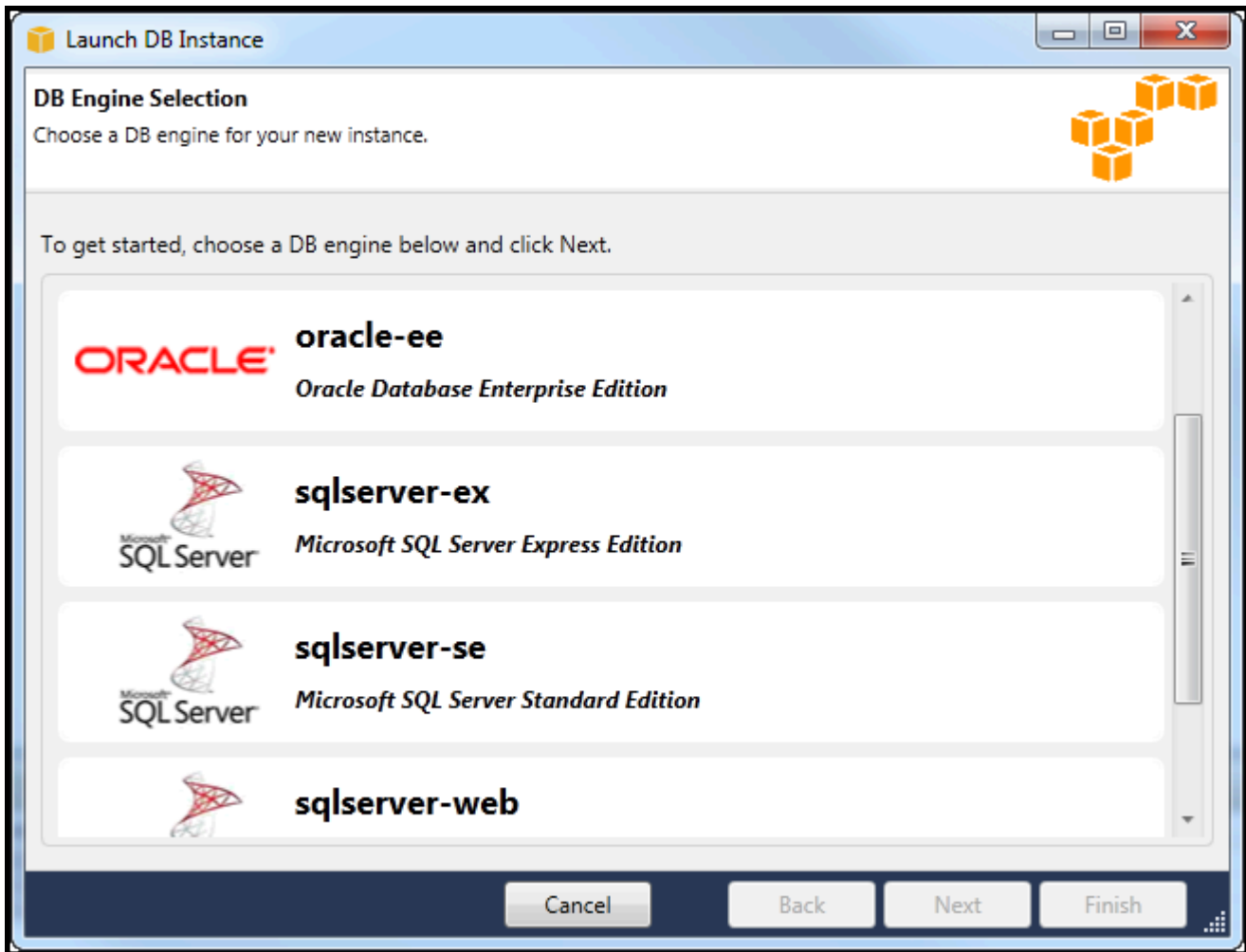
1. 在 AWS Explorer 中，開啟 Amazon RDS 節點的內容（按一下滑鼠右鍵）選單，然後選擇啟動資料庫執行個體。



或者，在資料庫執行個體索引標籤上，選擇啟動資料庫執行個體。



2. 在資料庫引擎選擇對話方塊中，選擇要啟動的資料庫引擎類型。在此演練中，選擇 Microsoft SQL Server Standard Edition (sqlserver-se)，然後選擇下一步。



- 在資料庫引擎執行個體選項對話方塊中，選擇組態選項。

在資料庫引擎執行個體選項和類別區段中，您可以指定下列設定。

License Model (授權模式)

引擎類型	授權
Microsoft SQL Server	包含授權
MySQL	general-public-license
Oracle	bring-your-own-license

授權模型會根據資料庫引擎的類型而有所不同。引擎類型授權 Microsoft SQL Server 授權包含的 MySQL general-public-license Oracle bring-your-own-license

資料庫執行個體版本

選擇您要使用的資料庫引擎版本。如果只支援一個版本，則會為您選取該版本。

資料庫執行個體類別

選擇資料庫引擎的執行個體類別。執行個體類別的定價會有所不同。如需詳細資訊，請參閱 [Amazon RDS 定價](#)。

執行多可用區部署

選取此選項可建立多可用區域部署，以提高資料耐用性和可用性。Amazon RDS 會在不同的可用區域中佈建和維護資料庫的待命複本，以便在排程或非計劃停機時自動容錯移轉。如需多可用區域部署定價的相關資訊，請參閱 [Amazon RDS](#) 詳細資訊頁面的定價區段。Microsoft SQL Server 不支援此選項。

自動升級次要版本

選取此選項，讓 AWS 自動為您在 RDS 執行個體上執行次要版本更新。

在 RDS 資料庫執行個體區段中，您可以指定下列設定。

Allocated Storage (配置的儲存體)

引擎	最小 (GB)	上限 (GB)
MySQL	5	1024
Oracle Enterprise Edition	10	1024
Microsoft SQL Server Express Edition	30	1024
Microsoft SQL Server Standard Edition	250	1024
Microsoft SQL Server Web Edition	30	1024

配置儲存體的最小值和最大值取決於資料庫引擎的類型。引擎最低 (GB) 最高 (GB) MySQL 5 1024
Oracle Enterprise Edition 10 1024 Microsoft SQL Server Express Edition 30 1024 Microsoft SQL
Server Standard Edition 250 1024 Microsoft SQL Server Web Edition 30 1024

DB Instance Identifier (資料庫執行個體識別符)

指定資料庫執行個體的名稱。此名稱不區分大小寫。它將在 AWS Explorer 中以小寫形式顯示。

Master User Name (主要使用者名稱)

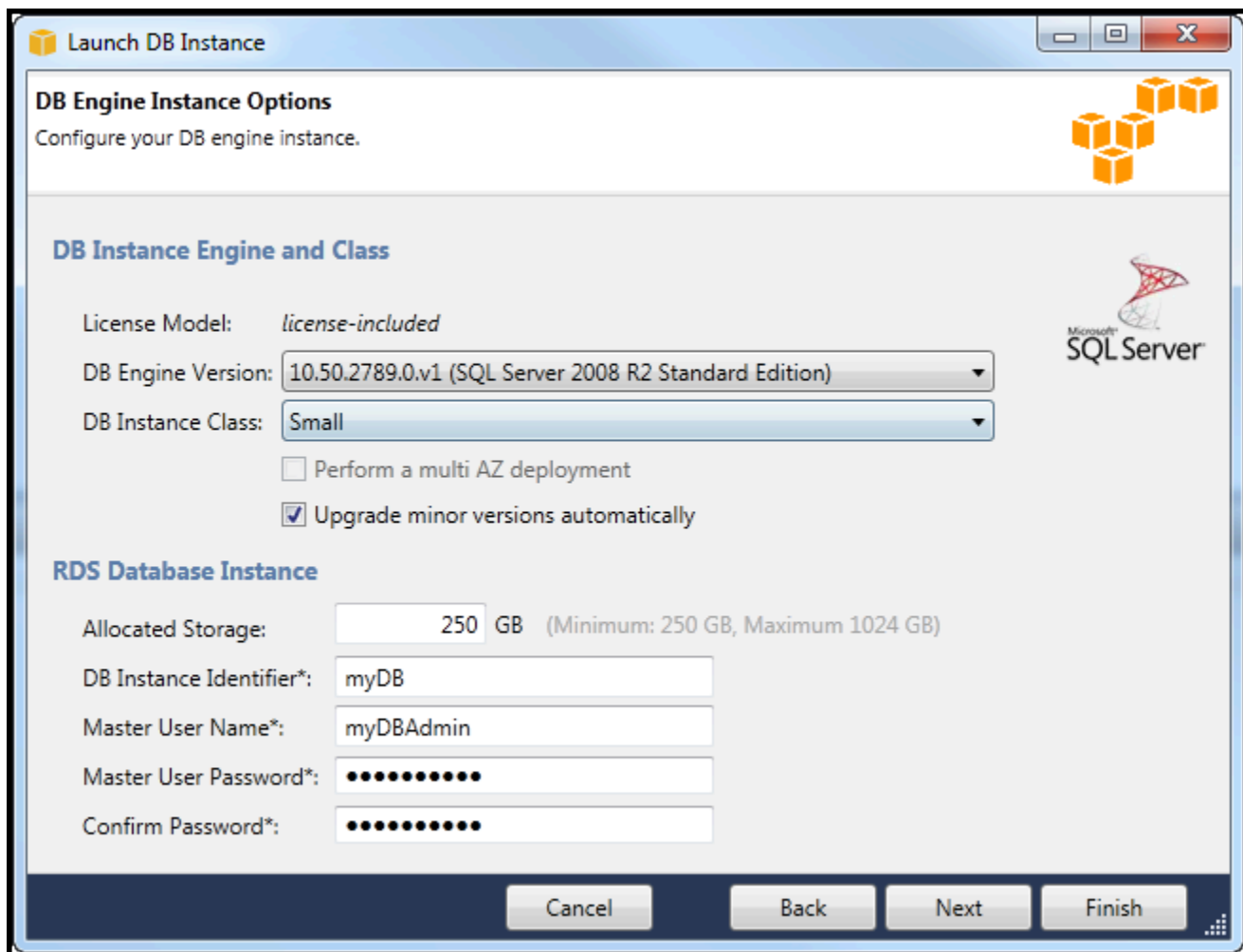
輸入資料庫執行個體管理員的名稱。

Master User Password (主要使用者密碼)

輸入資料庫執行個體管理員的密碼。

確認密碼

再次輸入密碼以驗證密碼是否正確。



The screenshot shows the 'Launch DB Instance' wizard in the AWS Management Console. The window title is 'Launch DB Instance'. The main heading is 'DB Engine Instance Options' with the subtitle 'Configure your DB engine instance.' There is an AWS logo in the top right corner. The 'DB Instance Engine and Class' section includes: License Model: license-included; DB Engine Version: 10.50.2789.0.v1 (SQL Server 2008 R2 Standard Edition); DB Instance Class: Small; checkboxes for 'Perform a multi AZ deployment' (unchecked) and 'Upgrade minor versions automatically' (checked). The 'RDS Database Instance' section includes: Allocated Storage: 250 GB (Minimum: 250 GB, Maximum 1024 GB); DB Instance Identifier*: myDB; Master User Name*: myDBAdmin; Master User Password*: masked with dots; Confirm Password*: masked with dots. At the bottom are buttons for 'Cancel', 'Back', 'Next', and 'Finish'.

1. 在其他選項對話方塊中，您可以指定下列設定。

Database Port (資料庫連接埠)

這是執行個體用來在網路上通訊的 TCP 連接埠。如果您的電腦透過防火牆存取網際網路，請將此值設定為防火牆允許流量的連接埠。

可用區域

如果您想要在區域中的特定可用區域中啟動執行個體，請使用此選項。您指定的資料庫執行個體可能無法在指定區域的所有可用區域中使用。

RDS 安全群組

選取要與您的執行個體建立關聯的 RDS 安全群組（或群組）。RDS 安全群組會指定允許存取執行個體的 IP 地址、Amazon EC2 AWS 帳戶執行個體和。如需 RDS 安全群組的詳細資訊，請參閱 [Amazon RDS 安全群組](#)。Toolkit for Visual Studio 會嘗試判斷您目前的 IP 地址，並提供將此地址新增至與執行個體相關聯之安全群組的選項。不過，如果您的電腦透過防火牆存取網際網路，則 Toolkit 為您的電腦產生的 IP 地址可能不準確。若要判斷要使用的 IP 地址，請洽詢您的系統管理員。

資料庫參數群組

（選用）從此下拉式清單中，選擇要與執行個體建立關聯的資料庫參數群組。資料庫參數群組可讓您變更執行個體的預設組態。如需詳細資訊，請參閱 [Amazon Relational Database Service 使用者指南](#) 和 [本文](#)。

當您在此對話方塊上指定設定時，請選擇下一步。

Launch DB Instance

Additional Options
Set additional configuration options for your instance.

Database Port: 1150-65535

Availability Zone:

If you have custom security or parameter groups you would like to associate with this instance, select them below otherwise proceed with default settings.

DB Security Groups:

- default

DB Parameter Group:

Add current CIDR (best estimate 72.21.198.68/32) to the selected security group(s)

Cancel Back Next Finish

2. Backup and Maintenance 對話方塊可讓您指定 Amazon RDS 是否應備份執行個體，如果是，備份應保留多久。您也可以指定備份應該發生的時段。

此對話方塊也可讓您指定您是否希望 Amazon RDS 在執行個體上執行系統維護。維護包括例行修補程式和次要版本升級。

您為系統維護指定的時段不能與為備份指定的時段重疊。

選擇下一步。

Launch DB Instance

Backup and Maintenance
Set backup and maintenance options for your instance

Automatic Backups

No automatic backups Backup and retain for: 1 day

Use a custom backup window:

Start time: 00 : 00 (UTC)
Duration: 0.5 hours

System Maintenance

Use a custom maintenance window:

On: Monday
Start: 00 : 00 (UTC)
Duration: 0.5 hours

Cancel Back Next Finish

3. 精靈中的最終對話方塊可讓您檢閱執行個體的設定。如果您需要修改設定，請使用返回按鈕。如果所有設定都正確，請選擇啟動。

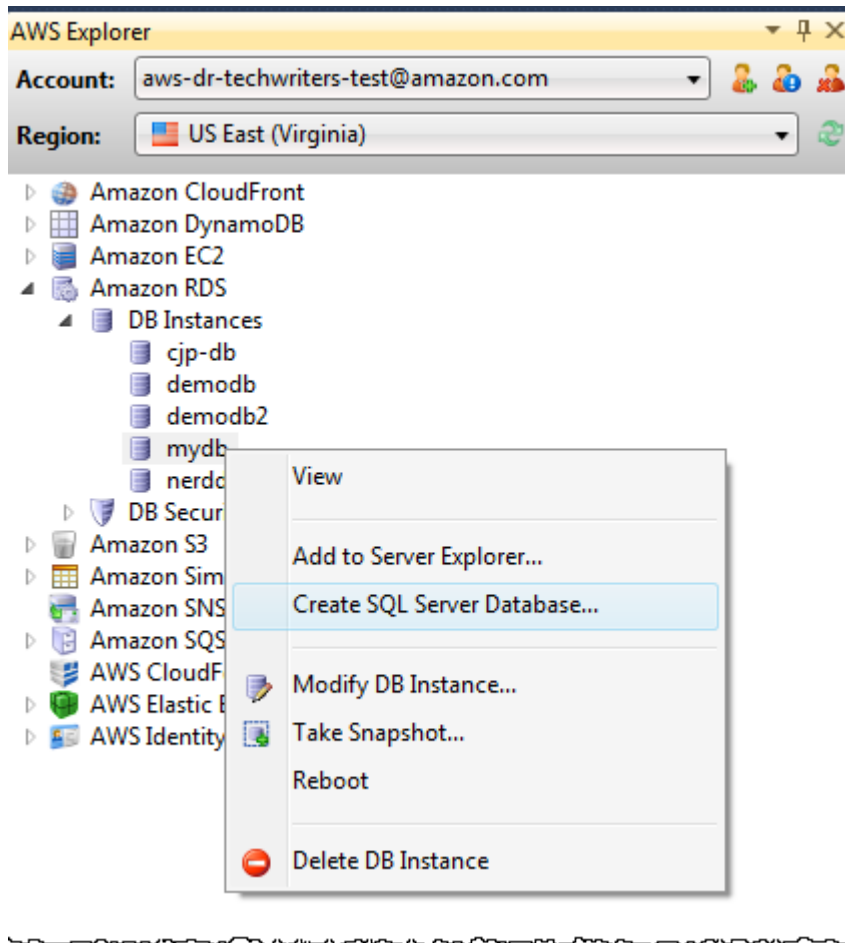
在 RDS 執行個體中建立 Microsoft SQL Server 資料庫

Microsoft SQL Server 的設計方式是在啟動 Amazon RDS 執行個體之後，您需要在 RDS 執行個體中建立 SQL Server 資料庫。

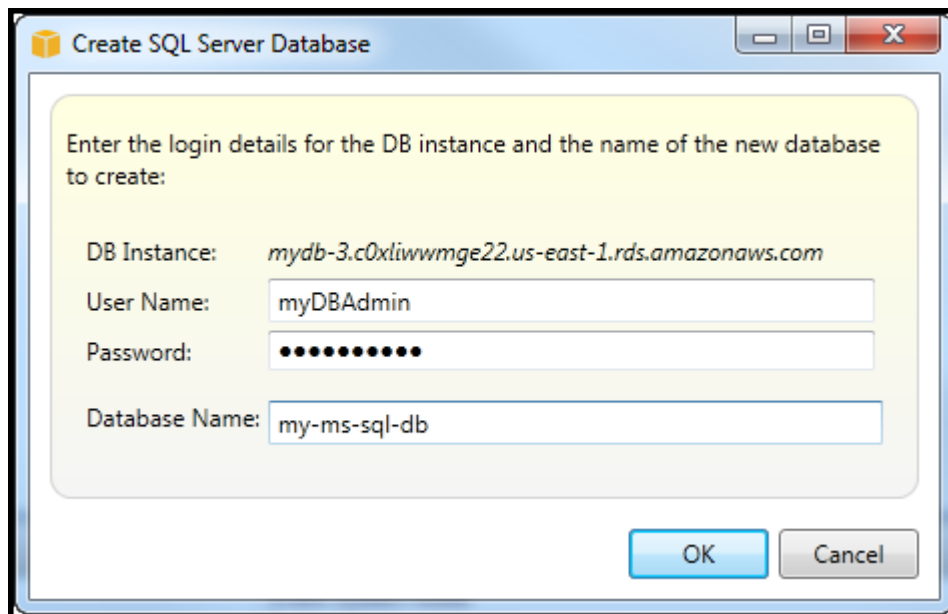
如需如何建立 Amazon RDS 執行個體的資訊，請參閱[啟動 Amazon RDS 資料庫執行個體](#)。

建立 Microsoft SQL Server 資料庫

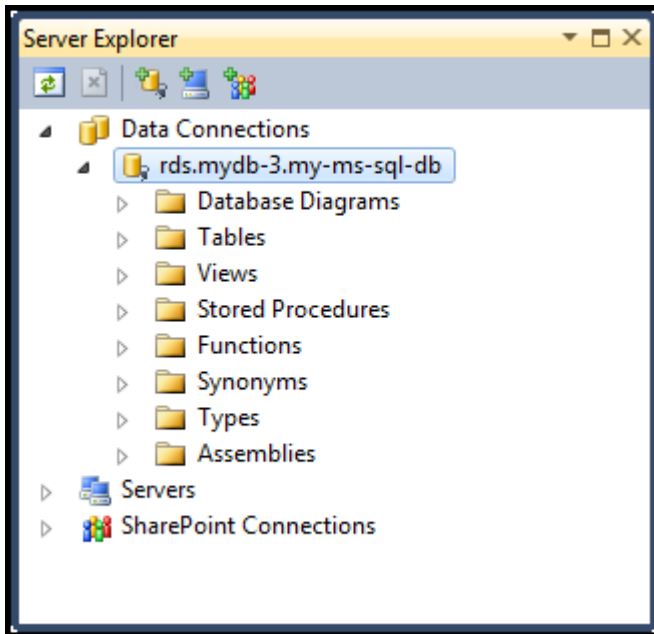
1. 在 AWS Explorer 中，開啟對應至 Microsoft SQL Server RDS 執行個體之節點的內容（按一下滑鼠右鍵）選單，然後選擇建立 SQL Server 資料庫。



2. 在建立 SQL Server 資料庫對話方塊中，輸入您在建立 RDS 執行個體時指定的密碼，輸入 Microsoft SQL Server 資料庫的名稱，然後選擇確定。



3. Toolkit for Visual Studio 會建立 Microsoft SQL Server 資料庫，並將其新增至 Visual Studio Server Explorer。



Amazon RDS 安全群組

Amazon RDS 安全群組可讓您管理 Amazon RDS 執行個體的網路存取。使用安全群組時，您可以使用 CIDR 表示法指定一組 IP 地址，而且 Amazon RDS 執行個體只會辨識來自這些地址的網路流量。

雖然它們的運作方式類似，但 Amazon RDS 安全群組與 Amazon EC2 安全群組不同。您可以將 EC2 安全群組新增至 RDS 安全群組。然後，任何屬於 EC2 安全群組成員的 EC2 執行個體都可以存取屬於 RDS 安全群組成員的 RDS 執行個體。

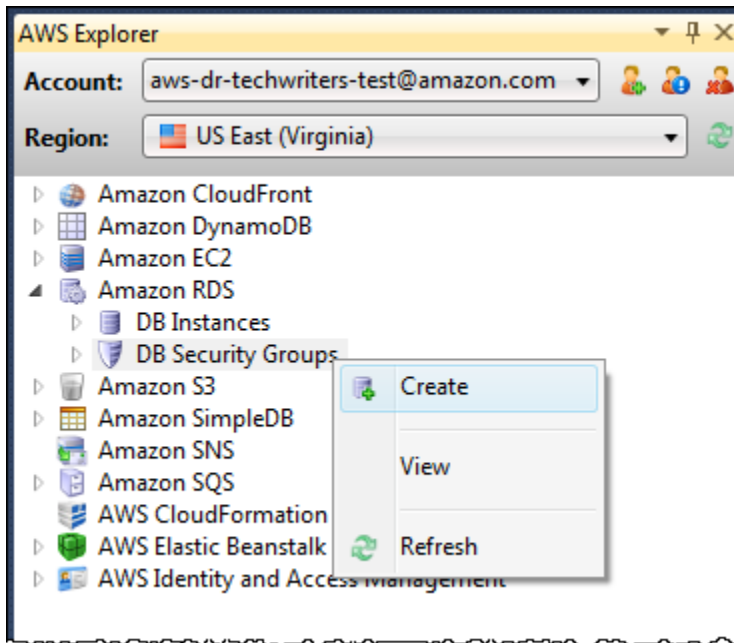
如需 Amazon RDS 安全群組的詳細資訊，請前往 [RDS 安全群組](#)。如需 Amazon EC2 安全群組的詳細資訊，請參閱 [EC2 使用者指南](#)。

建立 Amazon RDS 安全群組

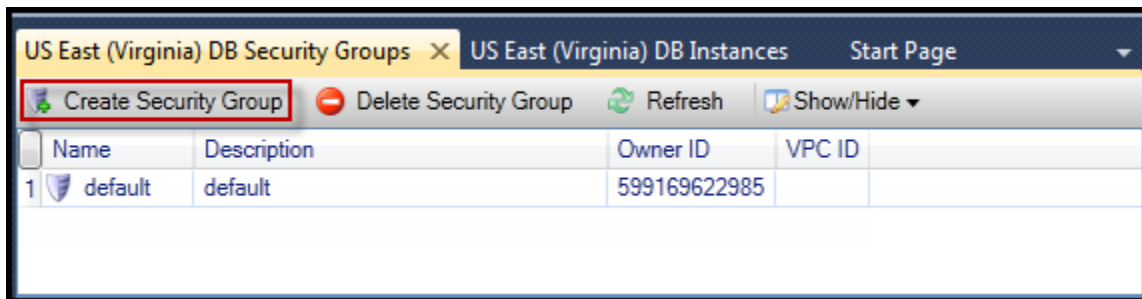
您可以使用 Toolkit for Visual Studio 來建立 RDS 安全群組。如果您使用 AWS Toolkit 啟動 RDS 執行個體，精靈將允許您指定要與執行個體搭配使用的 RDS 安全群組。您可以在啟動精靈之前，使用下列程序來建立該安全群組。

建立 Amazon RDS 安全群組

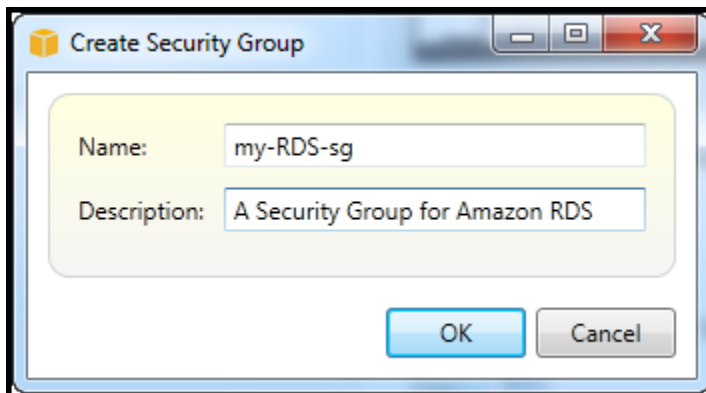
1. 在 AWS Explorer 中，展開 Amazon RDS 節點，開啟資料庫安全群組子節點的內容（按一下滑鼠右鍵）選單，然後選擇建立。



或者，在安全群組索引標籤上，選擇建立安全群組。如果未顯示此索引標籤，請開啟資料庫安全群組子節點的內容（按一下滑鼠右鍵）選單，然後選擇檢視。



2. 在建立安全群組對話方塊中，輸入安全群組的名稱和描述，然後選擇確定。



設定 Amazon RDS 安全群組的存取許可

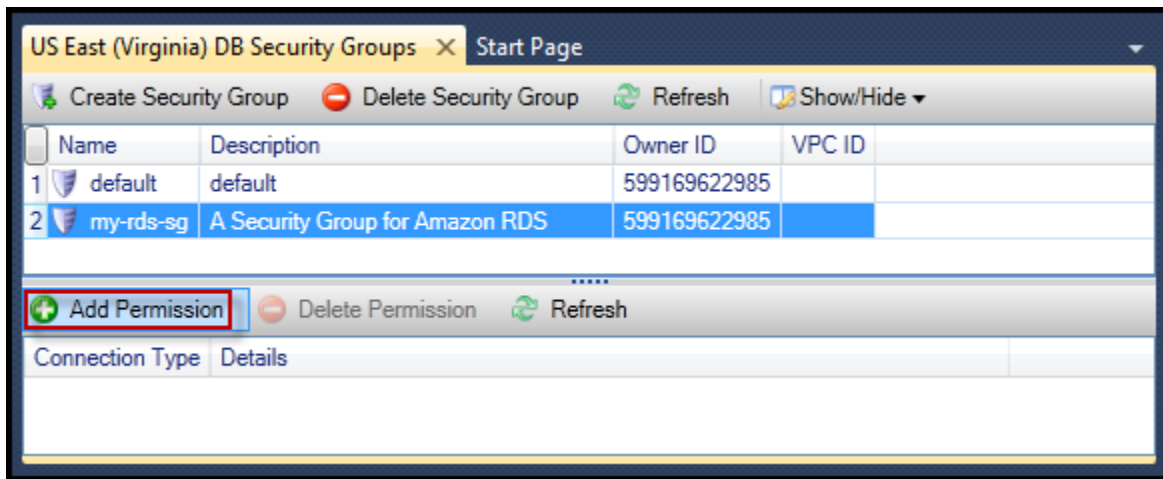
根據預設，新的 Amazon RDS 安全群組不提供網路存取。若要啟用存取使用安全群組的 Amazon RDS 執行個體，請使用下列程序來設定其存取許可。

設定 Amazon RDS 安全群組的存取權

1. 在安全群組索引標籤上，從清單檢視中選擇安全群組。如果您的安全群組未出現在清單中，請選擇重新整理。如果您的安全群組仍未出現在清單中，請確認您正在檢視正確 AWS 區域的清單。AWS Toolkit 中的安全群組索引標籤是區域特定的。

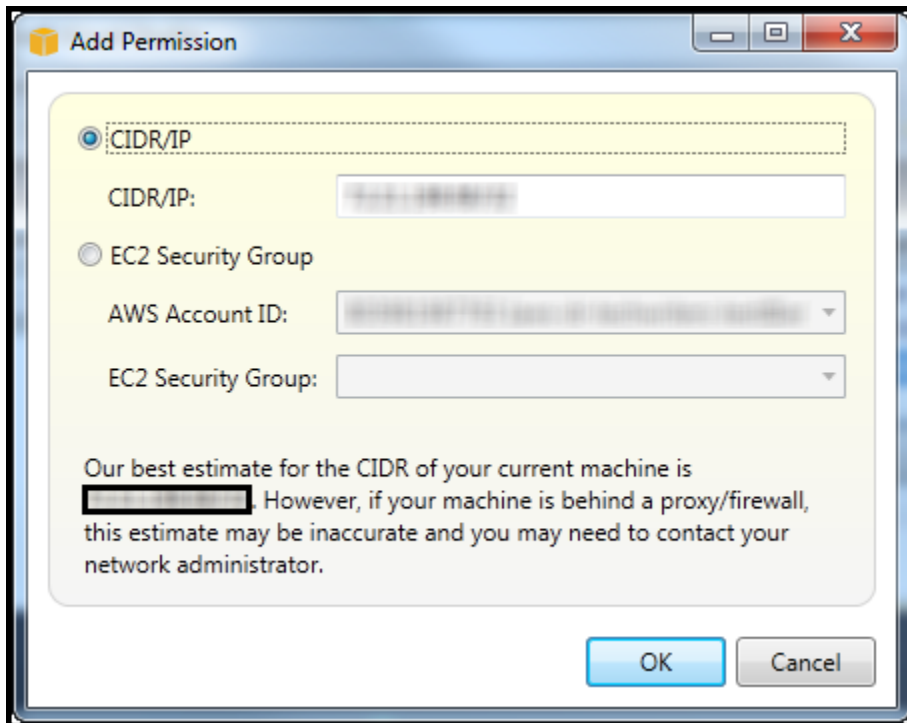
如果沒有出現安全群組索引標籤，請在 AWS Explorer 中開啟資料庫安全群組子節點的內容（按一下滑鼠右鍵）選單，然後選擇檢視。

2. 選擇 Add Permission (新增許可)。



安全群組索引標籤上的新增許可按鈕

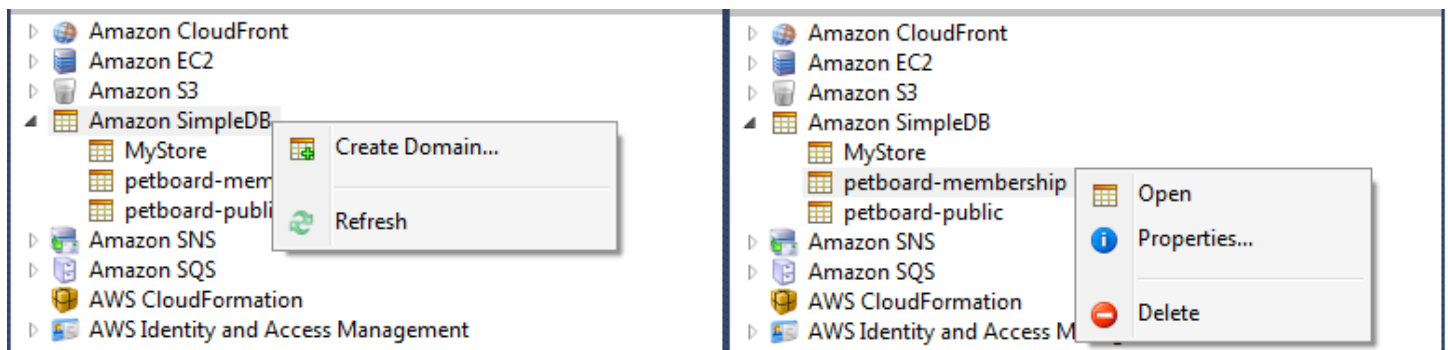
3. 在新增許可對話方塊中，您可以使用 CIDR 表示法來指定哪些 IP 地址可以存取 RDS 執行個體，或者您可以指定哪些 EC2 安全群組可以存取 RDS 執行個體。選擇 EC2 安全群組時，您可以為與相關聯的所有 EC2 執行個體指定 AWS 帳戶存取權，也可以從下拉式清單中選擇 EC2 安全群組。



AWS Toolkit 會嘗試判斷您的 IP 地址，並使用適當的 CIDR 規格自動填入對話方塊。不過，如果您的電腦透過防火牆存取網際網路，則 Toolkit 決定的 CIDR 可能不準確。

從 AWS Explorer 使用 Amazon SimpleDB

AWS Explorer 會顯示與作用中 AWS 帳戶相關聯的所有 Amazon SimpleDB 網域。您可以從 AWS Explorer 建立或刪除 Amazon SimpleDB 網域。

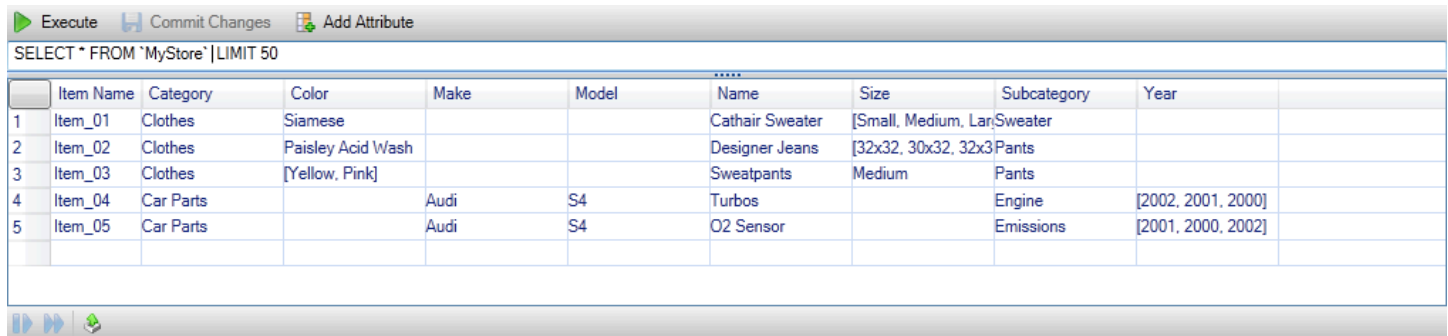


Create, delete, or open Amazon SimpleDB domains associated with your account

執行查詢和編輯結果

AWS Explorer 也可以顯示 Amazon SimpleDB 網域的網格檢視，您可以從中檢視該網域中的項目、屬性和值。您可以執行查詢，只顯示網域項目的子集。透過按兩下儲存格，您可以編輯該項目對應屬性的值。您也可以將新屬性新增至網域。

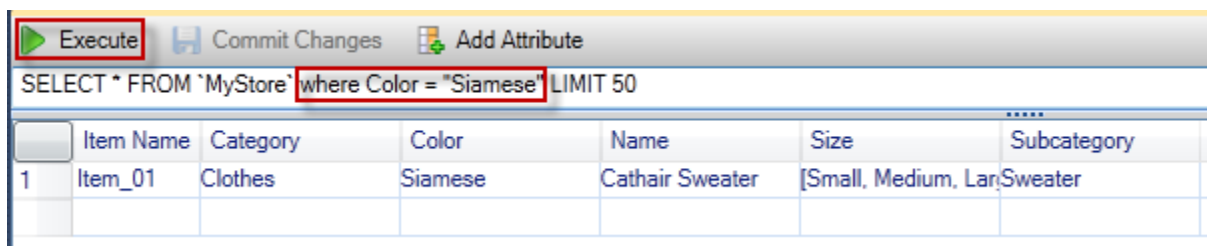
此處顯示的網域來自 隨附的 Amazon SimpleDB 範例 適用於 .NET 的 AWS SDK。



	Item Name	Category	Color	Make	Model	Name	Size	Subcategory	Year
1	Item_01	Clothes	Siamese			Cathair Sweater	[Small, Medium, Lar	Sweater	
2	Item_02	Clothes	Paisley Acid Wash			Designer Jeans	[32x32, 30x32, 32x3	Pants	
3	Item_03	Clothes	[Yellow, Pink]			Sweatpants	Medium	Pants	
4	Item_04	Car Parts		Audi	S4	Turbos		Engine	[2002, 2001, 2000]
5	Item_05	Car Parts		Audi	S4	O2 Sensor		Emissions	[2001, 2000, 2002]

Amazon SimpleDB grid view

若要執行查詢，請在網格檢視上方的文字方塊中編輯查詢，然後選擇執行。檢視會經過篩選，僅顯示符合查詢的項目。



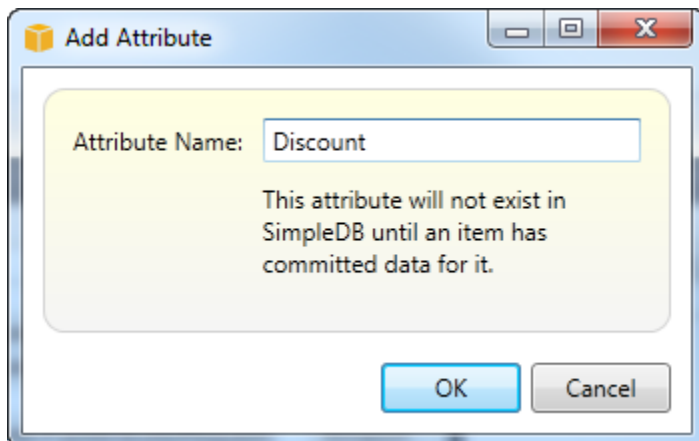
	Item Name	Category	Color	Name	Size	Subcategory
1	Item_01	Clothes	Siamese	Cathair Sweater	[Small, Medium, Lar	Sweater

Execute query from AWS Explorer

若要編輯與屬性相關聯的值，請按兩下對應的儲存格，編輯值，然後選擇遞交變更。

新增 屬性

若要新增屬性，請在檢視頂端選擇新增屬性。



Add Attribute

Attribute Name:

This attribute will not exist in SimpleDB until an item has committed data for it.

新增屬性 dialog box

若要讓 屬性成為網域的一部分，您必須為其新增值至至少一個項目，然後選擇遞交變更。



Commit changes for a new attribute

分頁查詢結果

檢視底部有三個按鈕。



Paginate and export buttons

前兩個按鈕提供查詢結果的分頁。若要顯示結果的額外頁面，請選擇第一個按鈕。若要顯示額外的十頁結果，請選擇第二個按鈕。在這種情況下，如果頁面包含在查詢中，則頁面等於 100 列或 LIMIT 值指定的結果數量。

匯出至 CSV

最後一個按鈕會將目前結果匯出至 CSV 檔案。

從 AWS Explorer 使用 Amazon SQS

Amazon Simple Queue Service (Amazon SQS) 是一種靈活的佇列服務，可在軟體應用程式中的不同執行程序之間傳遞訊息。Amazon SQS 佇列位於 AWS 基礎設施中，但傳遞訊息的程序可以位於本機、Amazon EC2 執行個體或其中的某些組合。Amazon SQS 非常適合用於協調跨多部電腦發佈工作。

Toolkit for Visual Studio 可讓您檢視與作用中帳戶相關聯的 Amazon SQS 佇列、建立和刪除佇列，以及透過佇列傳送訊息。（透過作用中帳戶，我們是指在 AWS Explorer 中選取的帳戶。）

如需 Amazon SQS 的詳細資訊，請參閱 AWS 文件中的 [SQS 簡介](#)。

建立佇列

您可以從 AWS Explorer 建立 Amazon SQS 佇列。佇列的 ARN 和 URL 將根據作用中帳戶的帳號和您在建立時指定的佇列名稱而定。

建立佇列

1. 在 AWS Explorer 中，開啟 Amazon SQS 節點的內容（按一下滑鼠右鍵）選單，然後選擇建立佇列。
2. 在建立佇列對話方塊中，指定佇列名稱、預設可見性逾時和預設交付延遲。預設可見性逾時和預設交付延遲以秒為單位指定。預設可見性逾時是指定程序取得訊息後，潛在接收程序將看不到訊息的時間量。預設交付延遲是從傳送訊息到第一次讓潛在接收程序可見的時間量。
3. 選擇確定。新的佇列會在 Amazon SQS 節點下顯示為子節點。

刪除佇列

您可以從 AWS Explorer 刪除現有的佇列。如果您刪除佇列，則與佇列相關聯的任何訊息都不再可用。

刪除佇列

1. 在 AWS Explorer 中，開啟您要刪除佇列的內容（按一下滑鼠右鍵）功能表，然後選擇刪除。

管理佇列屬性

您可以檢視和編輯 AWS Explorer 中顯示的任何佇列的屬性。您也可以從此屬性檢視傳送訊息至佇列。

管理佇列屬性

- 在 AWS Explorer 中，開啟您要管理其屬性之佇列的內容（按一下滑鼠右鍵）選單，然後選擇檢視佇列。

從佇列屬性檢視中，您可以編輯可見性逾時、訊息大小上限、訊息保留期和預設交付延遲。當您傳送訊息時，可以覆寫預設的交付延遲。在下列螢幕擷取畫面中，隱藏的文字是佇列 ARN 和 URL 的帳號元件。

Save Send Refresh

Visibility timeout (Seconds): 30 Created timestamp: 10/20/2011 1:34:49 PM

Maximum message size (Bytes): 65536 Last modified timestamp: 10/20/2011 1:34:49 PM

Message retention period (Seconds): 345600 Number of messages: 0

Default Delivery Delay (Seconds): 120 Number of messages not visible: 0

Queue ARN: arn:aws:sqs:us-east-1: :my-tk-queue

Queue URL: https://queue.amazonaws.com/ /my-tk-queue

Message Sampling

Message Id	Message Body	Sender Id	Sent
------------	--------------	-----------	------

⚠ Changes can take up to 60 seconds to propagate throughout the SQS system.

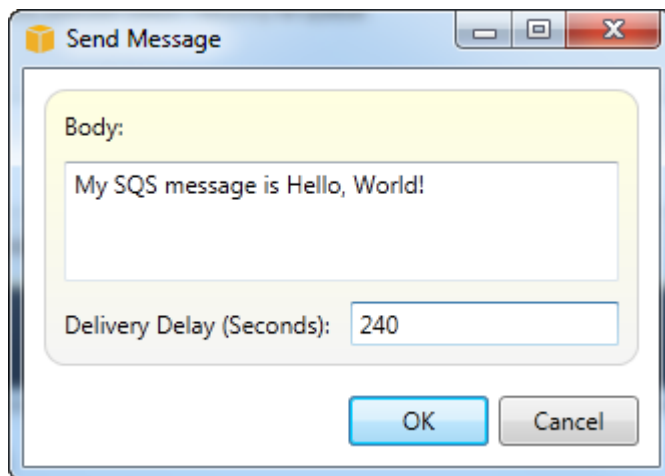
SQS queue properties view

傳送訊息至佇列

從佇列屬性檢視中，您可以將訊息傳送至佇列。

傳送訊息

1. 在佇列屬性檢視頂端，選擇傳送按鈕。
2. 輸入訊息。（選用）輸入將覆寫佇列預設交付延遲的交付延遲。在下列範例中，我們已以 240 秒的值覆寫延遲。選擇確定。



傳送訊息 dialog box

3. 等待大約 240 秒（四分鐘）。訊息會出現在佇列屬性檢視之的訊息取樣區段中。

The screenshot displays the AWS SQS console interface. At the top, there are buttons for 'Save', 'Send', and 'Refresh'. Below these are several configuration fields:

- Visibility timeout (Seconds): 30
- Maximum message size (Bytes): 65536
- Message retention period (Seconds): 345600
- Default Delivery Delay (Seconds): 120
- Created timestamp: 10/20/2011 1:34:49 PM
- Last modified timestamp: 10/20/2011 1:34:49 PM
- Number of messages: 1
- Number of messages not visible: 0

Queue ARN: `arn:aws:sqs:us-east-1:.....:my-tk-queue`
Queue URL: `https://queue.amazonaws.com/...../my-tk-queue`

Message Sampling

Message Id	Message Body	Sender Id	Sent
d58475df-2f92-49ec-a400-957bafcc5daf	My SQS message is Hello, World!	10/20/2011 2:33:02 PM

At the bottom, a warning icon indicates: "Changes can take up to 60 seconds to propagate throughout the SQS system."

SQS properties view with sent message

佇列屬性檢視中的時間戳記是您選擇傳送按鈕的時間。它不包含延遲。因此，訊息出現在佇列中且可供接收者使用的時間可能晚於此時間戳記。時間戳記會顯示在您電腦的本機時間。

身分和存取權管理

AWS Identity and Access Management (IAM) 可讓您更安全地管理對 AWS 帳戶和資源的存取。透過 IAM，您可以在主要 (根) 中建立多個使用者 AWS 帳戶。這些使用者可以擁有自己的登入資料：密碼、存取金鑰 ID 和私密金鑰，但所有 IAM 使用者都共用單一帳戶號碼。

您可以將 IAM 政策連接至使用者，以管理每個 IAM 使用者的資源存取層級。例如，您可以將政策連接至 IAM 使用者，該使用者可讓使用者存取帳戶中的 Amazon S3 服務和相關資源，但不提供任何其他服務或資源的存取權。

若要更有效率的存取管理，您可以建立 IAM 群組，這是使用者的集合。當您將政策連接到群組時，它將影響身為該群組成員的所有使用者。

除了在使用者和群組層級管理許可之外，IAM 還支援 IAM 角色的概念。就像使用者和群組一樣，您可以將政策連接到 IAM 角色。然後，您可以將 IAM 角色與 Amazon EC2 執行個體建立關聯。在 EC2 執行個體上執行的應用程式可以使用 AWS IAM 角色提供的許可存取。如需搭配 Toolkit 使用 IAM 角色的詳細資訊，請參閱[建立 IAM 角色](#)。如需 IAM 的詳細資訊，請前往[IAM 使用者指南](#)。

建立和設定 IAM 使用者

IAM 使用者可讓您授予其他人存取您的 的權限 AWS 帳戶。由於您可以將政策連接至 IAM 使用者，因此您可以精確限制 IAM 使用者可存取的資源，以及他們可以在這些資源上執行的操作。

最佳實務是，存取的所有使用者都 AWS 帳戶 應該以 IAM 使用者身分執行此操作，即使是帳戶擁有者也是如此。這可確保如果其中一個 IAM 使用者的登入資料遭到洩漏，只有這些登入資料可以停用。您不需要停用或變更帳戶的根登入資料。

從 Toolkit for Visual Studio 中，您可以透過將 IAM 政策連接至使用者，或將使用者指派給群組，將許可指派給 IAM 使用者。指派給群組的 IAM 使用者會從連接到群組的政策衍生其許可。如需詳細資訊，請參閱[建立 IAM 群組](#)，以及[新增 IAM 使用者到 IAM 群組](#)。

從 Toolkit for Visual Studio 中，您也可以為 IAM 使用者產生 AWS 登入資料（存取金鑰 ID 和私密金鑰）。如需詳細資訊，請參閱[為 IAM 使用者產生登入資料](#)

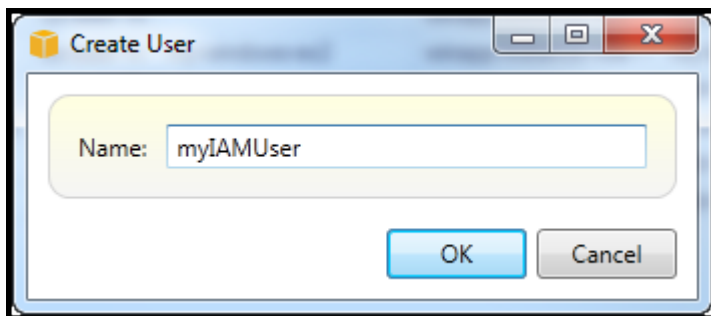


Toolkit for Visual Studio 支援指定 IAM 使用者登入資料，以透過 AWS Explorer 存取服務。由於 IAM 使用者通常無法完整存取所有 Amazon Web Services，因此可能無法使用 AWS Explorer 中的某些功能。如果您在作用中帳戶為 IAM 使用者時，使用 AWS Explorer 變更資源，然後將作用中帳戶切換至根帳戶，則在您重新整理 AWS Explorer 中的檢視之前，可能不會顯示變更。若要重新整理檢視，請選擇重新整理 () 按鈕。

如需有關如何從設定 IAM 使用者的資訊 AWS 管理主控台，請參閱《IAM 使用者指南》中的[使用使用者和群組](#)。

建立 IAM 使用者

1. 在 AWS Explorer 中，展開AWS Identity and Access Management節點，開啟使用者的內容（按一下滑鼠右鍵）選單，然後選擇建立使用者。
2. 在建立使用者對話方塊中，輸入 IAM 使用者的名稱，然後選擇確定。這是 IAM [易記的名稱](#)。如需有關 IAM 使用者名稱限制的資訊，請參閱 [IAM 使用者指南](#)。



Create an IAM user

新使用者會在AWS Identity and Access Management節點下的 使用者 下顯示為子節點。

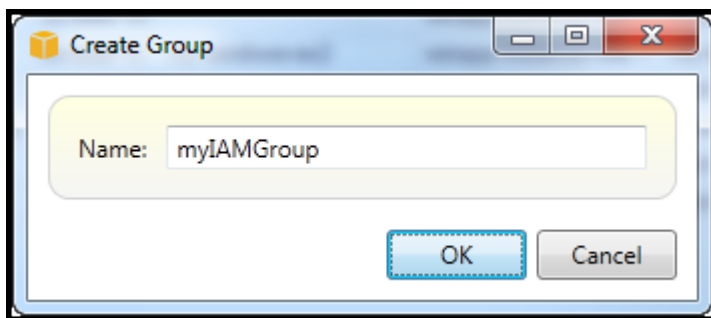
如需有關如何建立政策並將其連接至使用者的資訊，請參閱[建立 IAM 政策](#)。

建立 IAM 群組

群組提供將 IAM 政策套用至使用者集合的方法。如需有關如何管理 IAM 使用者和群組的資訊，請參閱《IAM 使用者指南》中的[使用使用者和群組](#)。

建立 IAM 群組

1. 在 AWS Explorer 的 Identity and Access Management 下，開啟群組的內容（按一下滑鼠右鍵）選單，然後選擇建立群組。
2. 在建立群組對話方塊中，輸入 IAM 群組的名稱，然後選擇確定。



Create IAM group

新的 IAM 群組會出現在 Identity and Access Management 的 Groups 子節點下。

如需如何建立政策並將其連接至 IAM 群組的詳細資訊，請參閱[建立 IAM 政策](#)。

將 IAM 使用者新增至 IAM 群組

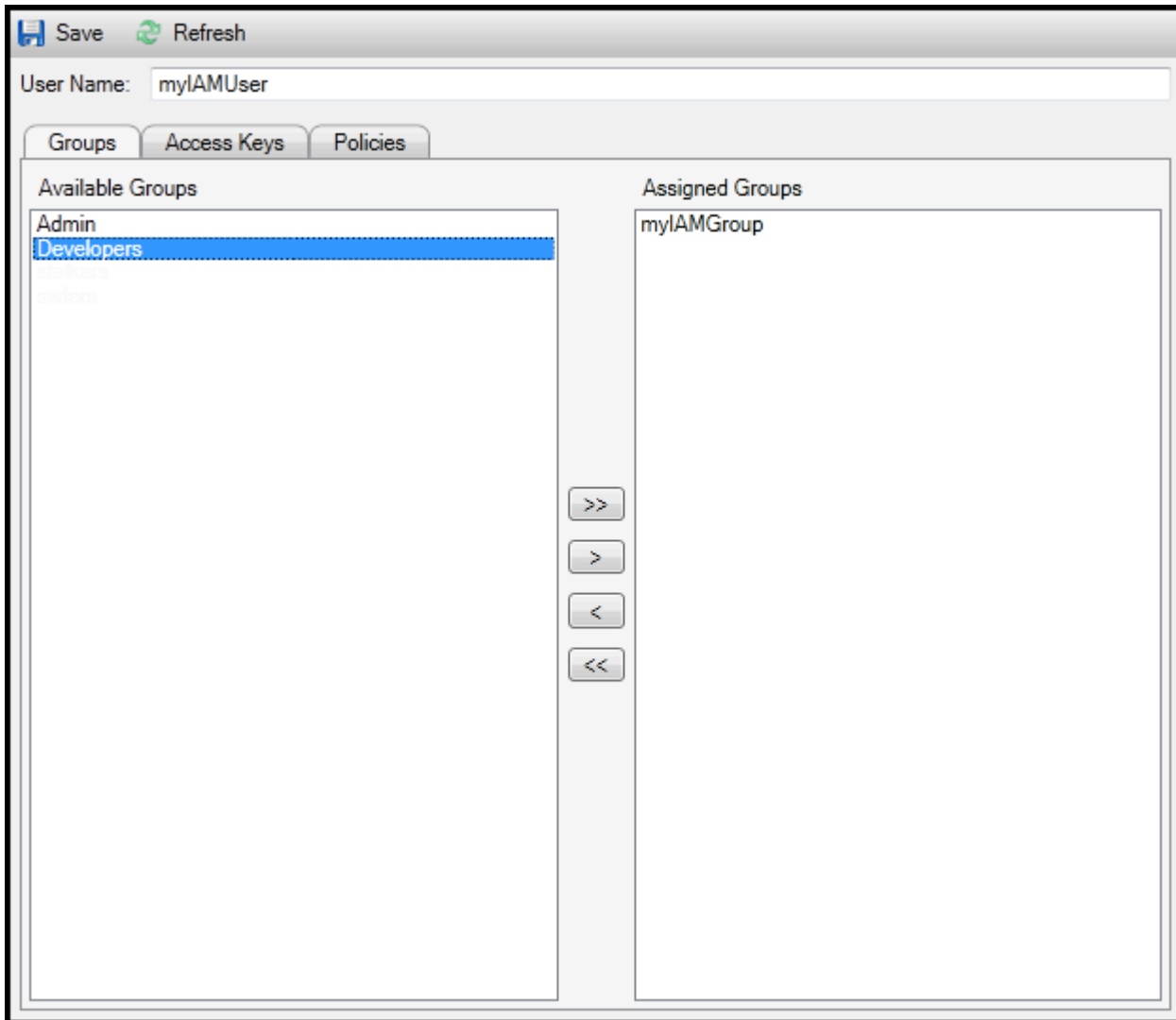
屬於 IAM 群組成員的 IAM 使用者會從連接到群組的政策衍生存取許可。IAM 群組的目的是更輕鬆地管理 IAM 使用者集合的許可。

如需連接至 IAM 群組的政策如何與連接至該 IAM 群組成員之 IAM 使用者的政策互動的相關資訊，請參閱《IAM 使用者指南》中的[管理 IAM 政策](#)。

在 AWS Explorer 中，您可以從使用者子節點將 IAM 使用者新增至 IAM 群組，而不是群組子節點。

將 IAM 使用者新增至 IAM 群組

1. 在 AWS Explorer 的 Identity and Access Management 下，開啟使用者的內容（按一下滑鼠右鍵）選單，然後選擇編輯。



Assign an IAM user to a IAM group

2. 群組索引標籤的左側窗格會顯示可用的 IAM 群組。右窗格會顯示指定 IAM 使用者已經是成員的群組。

若要將 IAM 使用者新增至群組，請在左側窗格中選擇 IAM 群組，然後選擇 > 按鈕。

若要從群組中移除 IAM 使用者，請在右窗格中選擇 IAM 群組，然後選擇 < 按鈕。

若要將 IAM 使用者新增至所有 IAM 群組，請選擇 >> 按鈕。同樣地，若要從所有群組中移除 IAM 使用者，請選擇 << 按鈕。

若要選擇多個群組，請依序選擇它們。您不需要按住 Control 金鑰。若要從您的選擇中清除群組，只需再次選擇即可。

3. 當您完成將 IAM 使用者指派給 IAM 群組後，請選擇儲存。

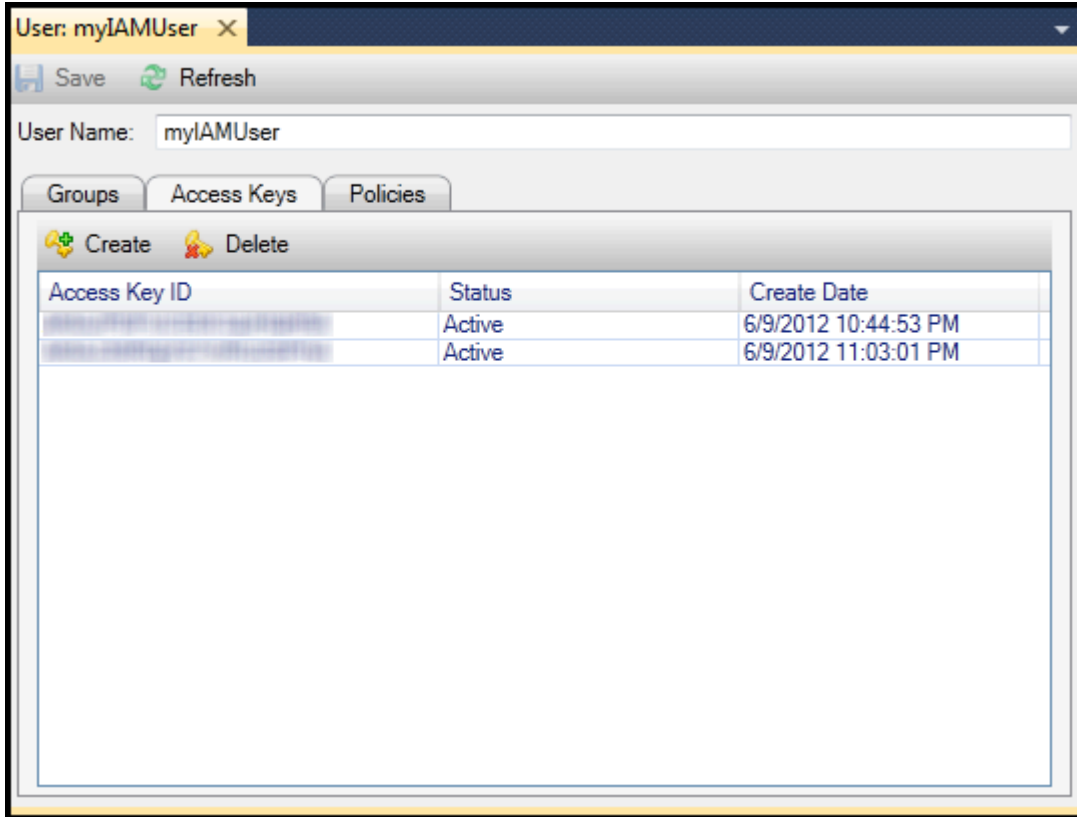
產生 IAM 使用者的登入資料

透過 Toolkit for Visual Studio，您可以產生用於進行 API 呼叫的存取金鑰 ID 和私密金鑰 AWS。您也可以透過 Toolkit 指定這些金鑰來存取 Amazon Web Services。如需如何指定登入資料以搭配 Toolkit 使用的詳細資訊，請參閱 [網頁](#)。如需如何安全地處理登入資料的詳細資訊，請參閱 [管理 AWS 存取金鑰的最佳實務](#)。

Toolkit 無法用來產生 IAM 使用者的密碼。

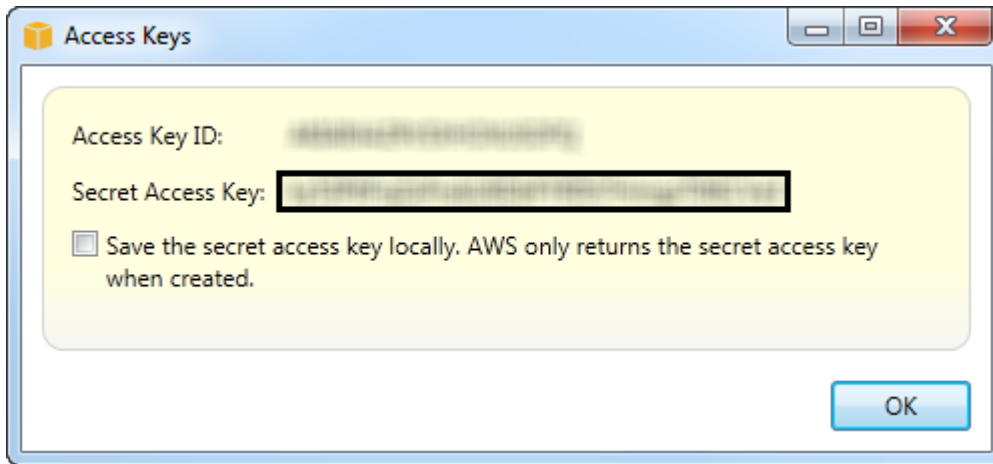
為 IAM 使用者產生登入資料

1. 在 AWS Explorer 中，開啟 IAM 使用者的內容（按一下滑鼠右鍵）選單，然後選擇編輯。



2. 若要產生登入資料，請在存取金鑰索引標籤上，選擇建立。

您可以為每位 IAM 使用者產生最多兩組登入資料。如果您已有兩組登入資料，且需要建立額外的集合，則必須刪除其中一個現有集合。

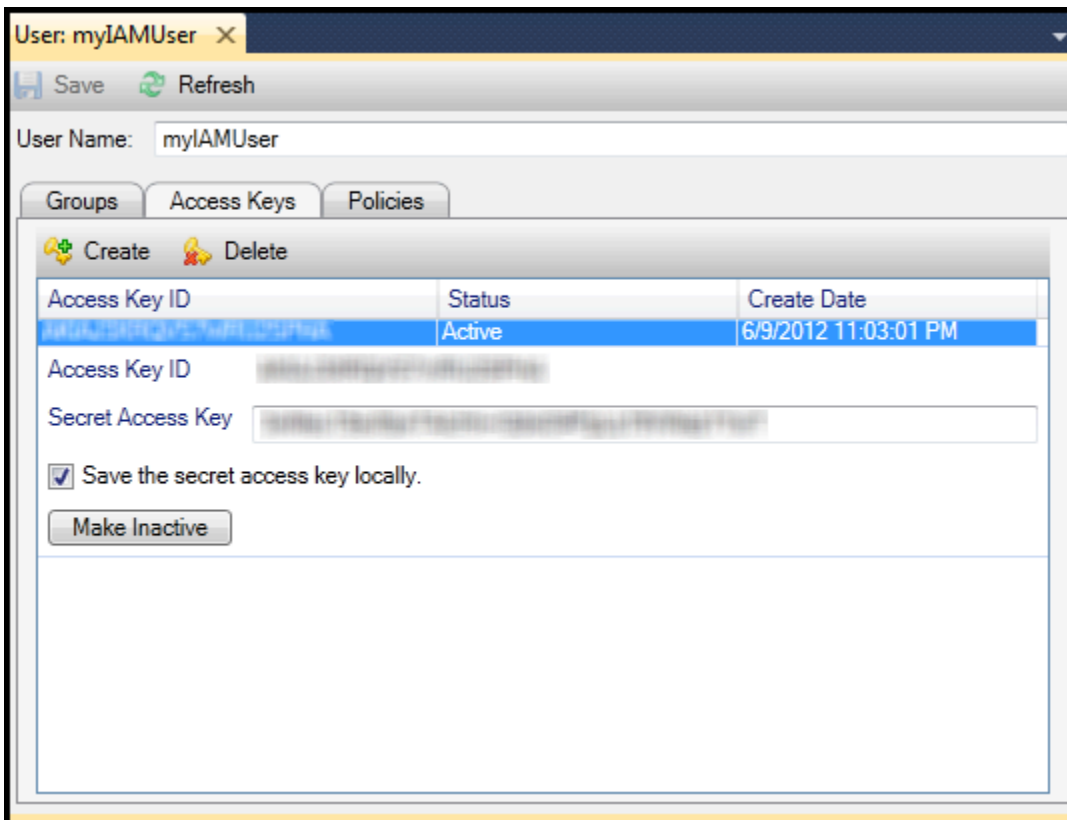


reate credentials for IAM user

如果您希望 Toolkit 將秘密存取金鑰的加密副本儲存至本機磁碟機，請選取在本機儲存秘密存取金鑰。AWS 只有在建立時才會傳回秘密存取金鑰。您也可以從對話方塊複製私密存取金鑰，並將其儲存在安全的位置。

3. 選擇確定。

產生登入資料後，您可以從存取金鑰索引標籤檢視登入資料。如果您已選取選項，讓 Toolkit 在本機儲存私密金鑰，則會顯示在此處。



Create credentials for IAM user

如果您自行儲存私密金鑰，並希望 Toolkit 儲存私密金鑰，請在私密存取金鑰方塊中輸入私密存取金鑰，然後選取在本機儲存私密存取金鑰。

若要停用登入資料，請選擇設為非作用中。（如果您懷疑登入資料已洩露，您可以執行此操作。如果您收到安全憑證的保證，您可以重新啟用憑證。）

建立 IAM 角色

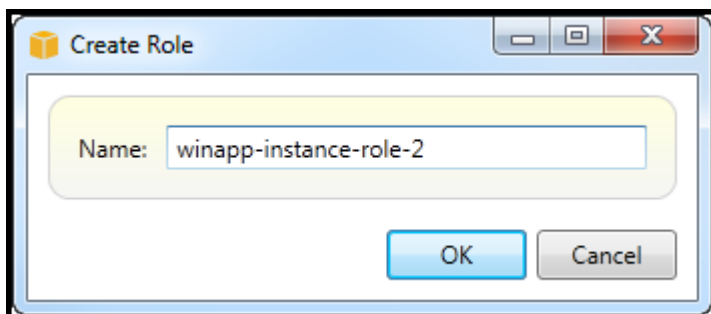
Toolkit for Visual Studio 支援 IAM 角色的建立和組態。就像使用者和群組一樣，您可以將政策連接到 IAM 角色。然後，您可以將 IAM 角色與 Amazon EC2 執行個體建立關聯。與 EC2 執行個體的關聯是透過執行個體描述檔處理，這是角色的邏輯容器。在 EC2 執行個體上執行的應用程式會自動獲得與 IAM 角色相關聯之政策指定的存取層級。即使應用程式尚未指定其他 AWS 登入資料，也是如此。

例如，您可以建立角色並將政策連接至該角色，以限制只能存取 Amazon S3。將此角色與 EC2 執行個體建立關聯後，您可以在該執行個體上執行應用程式，而應用程式將可存取 Amazon S3，但不能存取任何其他服務或資源。這種方法的優點是您不需要擔心在 EC2 執行個體上安全地傳輸和儲存 AWS 憑證。

如需 IAM 角色的詳細資訊，請參閱《[IAM 使用者指南](#)》中的[使用 IAM 角色](#)。如需 AWS 使用與 Amazon EC2 執行個體相關聯的 IAM 角色存取的程式範例，請前往 [Java](#)、[.NET](#)、[PHP](#) 和 Ruby 的 AWS 開發人員指南 ([使用 IAM 設定登入資料](#)、[建立 IAM 角色](#)和[使用 IAM 政策](#))。

建立 IAM 角色

1. 在 AWS Explorer 的 Identity and Access Management 下，開啟角色的內容（按一下滑鼠右鍵）選單，然後選擇建立角色。
2. 在建立角色對話方塊中，輸入 IAM 角色的名稱，然後選擇確定。



Create IAM role

新的 IAM 角色會出現在 Identity and Access Management 中的角色下。

如需如何建立政策並將其連接至角色的詳細資訊，請參閱[建立 IAM 政策](#)。

建立 IAM 政策

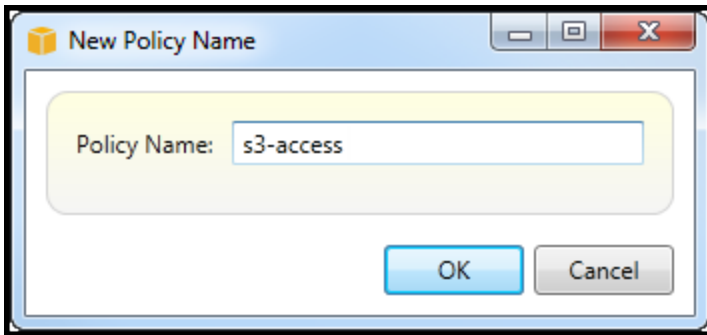
政策是 IAM 的基礎。政策可以與 IAM 實體建立關聯，例如使用者、群組或角色。政策會指定為使用者、群組或角色啟用的存取層級。

建立 IAM 政策

在 AWS Explorer 中，展開 AWS Identity and Access Management 節點，然後展開要連接政策的實體類型 (群組、角色或使用者) 節點。例如，開啟 IAM 角色的內容選單，然後選擇編輯。

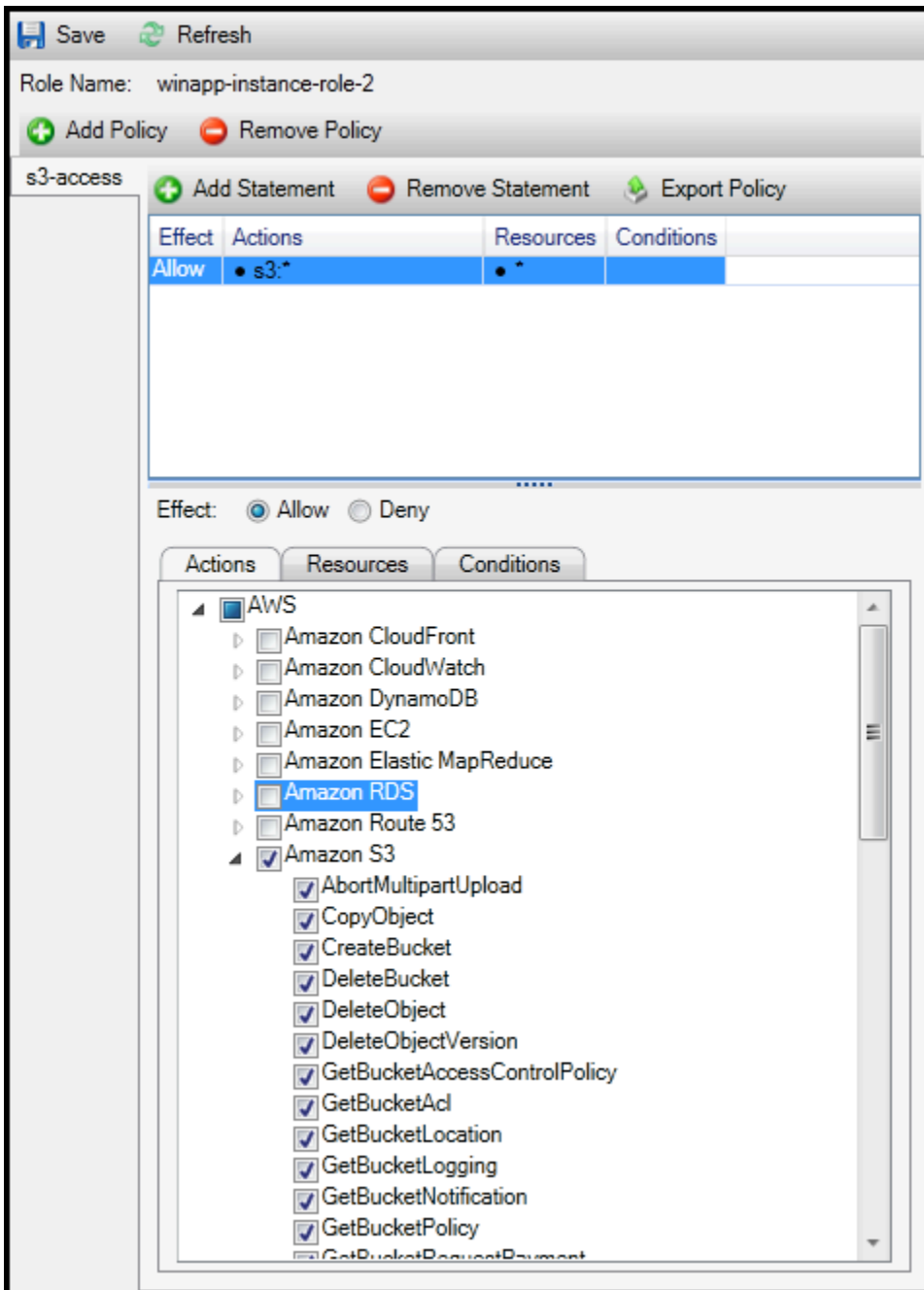
與角色相關聯的標籤會顯示在 AWS Explorer 中。選擇新增政策連結。

在新政策名稱對話方塊中，輸入政策的名稱 (例如，s3-access)。



New Policy Name dialog box

在政策編輯器中，新增政策陳述式以指定要提供給角色的存取層級 (在此範例中，為與政策相關聯的 winapp-instance-role-2。在此範例中，政策提供 Amazon S3 的完整存取權，但無法存取任何其他資源。



Specify IAM policy

如需更精確的存取控制，您可以在政策編輯器中展開子節點，以允許或不允許與 Amazon Web Services 相關聯的動作。

編輯政策後，請選擇儲存連結。

AWS Lambda

使用開發和部署以 .NET Core 為基礎的 C# Lambda 函數 AWS Toolkit for Visual Studio。AWS Lambda 是一種運算服務，可讓您執行程式碼，而無需佈建或管理伺服器。Toolkit for Visual Studio 包含適用於 Visual Studio 的 AWS Lambda .NET Core 專案範本。

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

如需 .NET Core 的詳細資訊，請參閱 Microsoft [.NET Core](#) 指南。如需 Windows、macOS 和 Linux 平台的 .NET Core 先決條件和安裝說明，請參閱 [.NET Core Downloads](#)。

下列主題說明如何 AWS Lambda 使用 Toolkit for Visual Studio 來使用。

主題

- [基本 AWS Lambda 專案](#)
- [基本 AWS Lambda 專案建立 Docker 影像](#)
- [教學課程：使用建置和測試無伺服器應用程式 AWS Lambda](#)
- [教學課程：建立 Amazon Rekognition Lambda 應用程式](#)
- [教學課程：搭配使用 Amazon Logging Framework AWS Lambda 來建立應用程式日誌](#)

基本 AWS Lambda 專案

您可以在 中使用 Microsoft .NET Core 專案範本建立 Lambda 函數 AWS Toolkit for Visual Studio。

建立 Visual Studio .NET Core Lambda 專案

您可以使用 Lambda-Visual Studio 範本和藍圖來協助加速專案初始化。Lambda 藍圖包含預先編寫的函數，可簡化彈性專案基礎的建立。

Note

Lambda 服務對不同的套件類型具有資料限制。如需資料限制的詳細資訊，請參閱《[Lambda 使用者指南](#)》中的 [Lambda 配額](#) 主題。AWS

在 Visual Studio 中建立 Lambda 專案

1. 從 Visual Studio 展開檔案功能表，展開新增，然後選擇專案。

2. 在新增專案對話方塊中，將語言、平台和專案類型下拉式方塊設定為「全部」，然後在aws lambda搜尋欄位中輸入。選擇 AWS Lambda 專案 (.NET Core - C#) 範本。
3. 在名稱欄位中，輸入 **AWSLambdaSample**，指定所需的檔案位置，然後選擇建立以繼續。
4. 從選取藍圖頁面，選取空白函數藍圖，然後選擇完成以建立 Visual Studio 專案。

檢閱專案檔案

有兩個專案檔案需要檢閱：`aws-lambda-tools-defaults.json`和 `Function.cs`。

下列範例顯示 檔案，該`aws-lambda-tools-defaults.json`檔案會自動建立為專案的一部分。您可以使用此檔案中的欄位來設定建置選項。

Note

Visual Studio 中的專案範本包含許多不同的欄位，請注意下列事項：

- `function-handler`：指定 Lambda 函數執行時執行的方法
- 在函數處理常式欄位中指定值會在發佈精靈中預先填入該值。
- 如果您重新命名函數、類別或組件，則還需要更新 `aws-lambda-tools-defaults.json` 檔案中的對應欄位。

```
{
  "Information": [
    "This file provides default values for the deployment wizard inside Visual Studio
    and the AWS Lambda commands added to the .NET Core CLI.",
    "To learn more about the Lambda commands with the .NET Core CLI execute the
    following command at the command line in the project root directory.",
    "dotnet lambda help",
    "All the command line options for the Lambda command can be specified in this
    file."
  ],
  "profile": "default",
  "region": "us-west-2",
  "configuration": "Release",
  "function-architecture": "x86_64",
  "function-runtime": "dotnet8",
  "function-memory-size": 512,
  "function-timeout": 30,
```

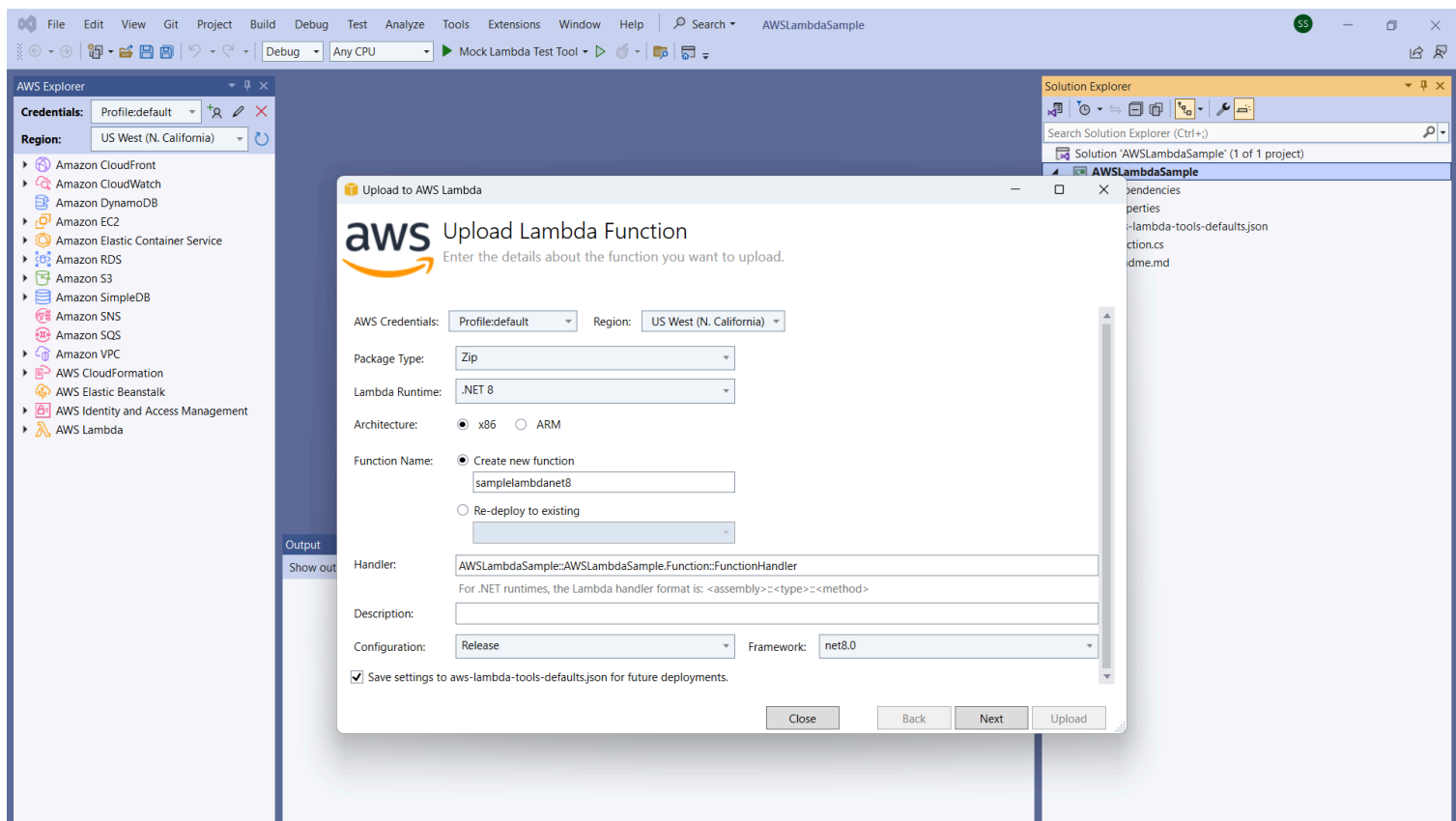
```
"function-handler": "AWSLambdaSample::AWSLambdaSample.Function::FunctionHandler"
}
```

檢查 `Function.cs` 檔案。`Function.cs` 定義 c# 函數以公開為 Lambda 函數。`FunctionHandler` 這是 Lambda 函數執行時執行的 Lambda 功能。在此專案中，有一個已定義的函數：`FunctionHandler`，它會 `ToUpper()` 呼叫輸入文字。

您的專案現在已準備好發佈至 Lambda。


發佈至 Lambda

下列程序和映像示範如何使用 將函數上傳至 Lambda AWS Toolkit for Visual Studio。



將您的函數發佈至 Lambda

1. 透過展開檢視並選擇 **AWS Explorer** 導覽至 **AWS Explorer**。
2. 在解決方案總管中，開啟您要發佈之專案的內容選單（按一下滑鼠右鍵），然後選擇發佈至 **AWS Lambda** 以開啟上傳 Lambda 函數視窗。
3. 從上傳 Lambda 函數視窗中，完成下列欄位：

- a. 套件類型：選擇 **Zip**。ZIP 檔案將作為建置程序的結果建立，並將上傳到 Lambda。或者，您可以選擇套件類型 **Image**。[教學課程：基本 Lambda 專案建立 Docker 映像](#) 說明如何使用套件類型 進行發佈 **Image**。
 - b. Lambda 執行期：從下拉式選單中選擇您的 Lambda 執行期。
 - c. 架構：選取您偏好架構的放射狀。
 - d. 函數名稱：選取建立新函數的放射狀，然後輸入 Lambda 執行個體的顯示名稱。AWS Explorer 和 AWS 管理主控台 都會參考此名稱。
 - e. 處理常式：使用此欄位來指定函數處理常式。例如：**`AWSLambdaSample::AWSLambdaSample.Function::FunctionHandler`**。
 - f. (選用) 描述：輸入要與您的執行個體一起顯示的描述性文字，從內 AWS 管理主控台。
 - g. 組態：從下拉式選單中選擇您偏好的組態。
 - h. 架構：從下拉式選單中選擇您偏好的架構。
 - i. 儲存設定：選取此方塊，將您目前的設定儲存 `aws-lambda-tools-defaults.json` 為未來部署的預設值。
 - j. 選擇下一步以繼續進階函數詳細資訊視窗。
4. 在進階函數詳細資訊視窗中，完成下列欄位：
- a. 角色名稱：選擇與您的帳戶相關聯的角色。此角色會為 函數中程式碼所做的任何 AWS 服務呼叫提供臨時登入資料。如果您沒有角色，請在下拉式清單中捲動以根據 AWS 受管政策尋找新角色，然後選擇 `AWSLambdaBasicExecutionRole`。此角色具有最低的存取許可。
-  **Note**

您的帳戶必須具有執行 IAM ListPolicies 動作的許可，否則角色名稱清單將為空白，您將無法繼續。
- b. (選用) 如果您的 Lambda 函數存取 Amazon VPC 上的資源，請選取子網路和安全群組。
 - c. (選用) 設定 Lambda 函數所需的任何環境變數。金鑰會由免費的預設服務金鑰自動加密。或者，您可以指定需要付費的 AWS KMS 金鑰。[KMS](#) 是一項受管服務，可用來建立和控制用來加密資料的加密金鑰。如果您有 AWS KMS 金鑰，您可以從清單中選取它。
5. 選擇上傳以開啟上傳函數視窗，然後開始上傳程序。

Note

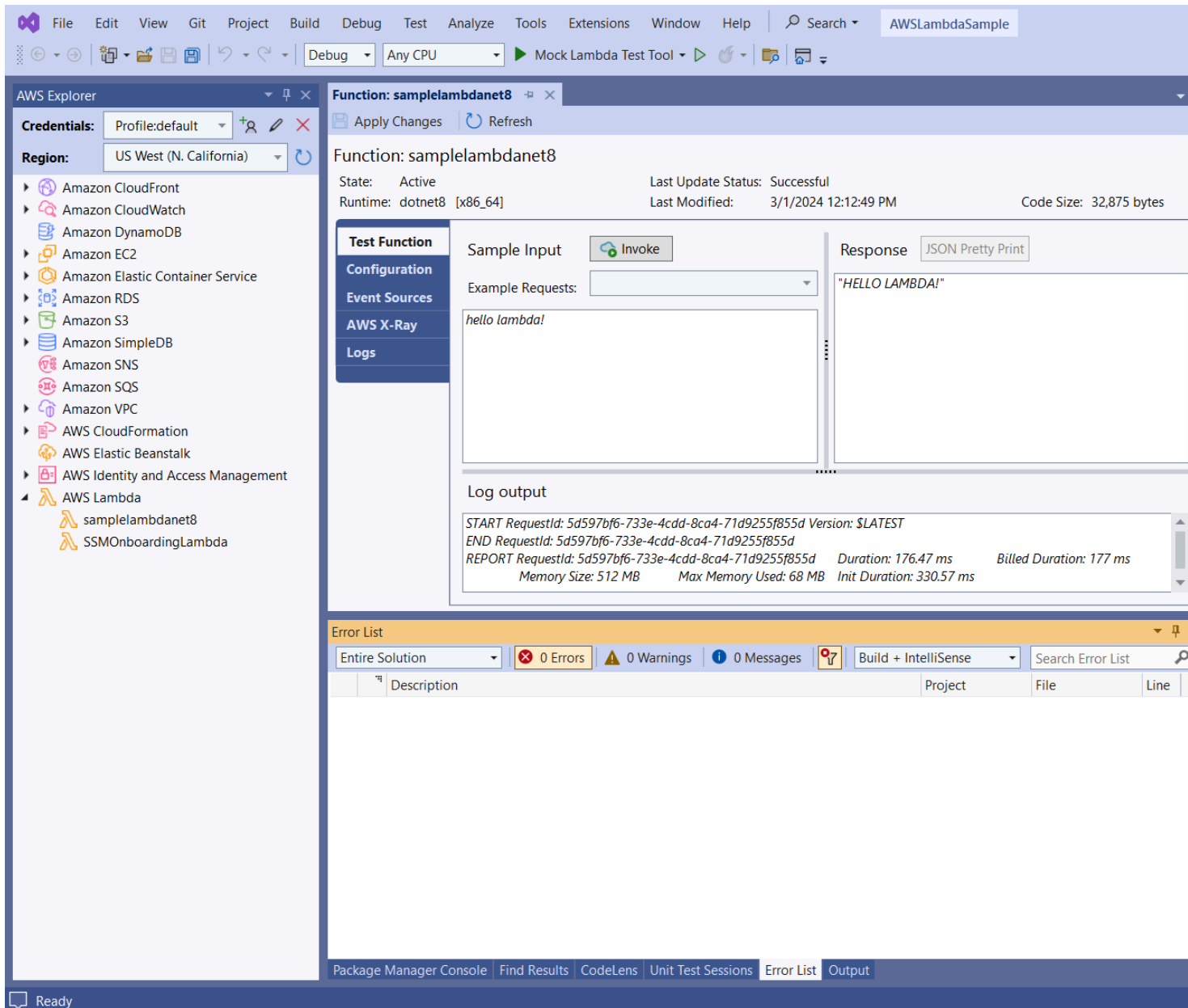
上傳函數頁面會在函數上傳時顯示 AWS。若要在上傳後保持精靈開啟，以便您可以檢視報告，請在上傳完成之前清除表單底部的成功完成時自動關閉精靈。

函數上傳後，您的 Lambda 函數即為即時。函數：檢視頁面隨即開啟，並顯示新的 Lambda 函數組態。

6. 從測試函數索引標籤 `hello lambda!`，在文字輸入欄位中輸入 `hello lambda!`，然後選擇叫用以手動叫用您的 Lambda 函數。您的文字會顯示在回應索引標籤中，並轉換為大寫。

Note

您可以隨時在節點下的 AWS Explorer 中按兩下已部署的執行個體，以重新開啟 函數：檢視AWS Lambda。



7. (選用) 若要確認您已成功發佈 Lambda 函數，請登入 AWS 管理主控台，然後選擇 Lambda。主控台會顯示所有已發佈的 Lambda 函數，包括您剛建立的函數。

清除

如果您不打算繼續使用此範例繼續開發，請刪除您部署的函數，以免因帳戶中未使用的資源而產生費用。

Note

Lambda 會自動為您監控 Lambda 函數，並透過 Amazon CloudWatch 報告指標。若要監控和疑難排解函數，請參閱《AWS Lambda 開發人員指南》中的[使用 Amazon CloudWatch 疑難排解和監控 AWS Lambda 函數](#)主題。

刪除函數

1. 從 AWS Explorer 展開AWS Lambda節點。
2. 用滑鼠右鍵按一下部署的執行個體，然後選擇刪除。

基本 AWS Lambda 專案建立 Docker 影像

您可以使用 Toolkit for Visual Studio 將 AWS Lambda 函數部署為 Docker 映像。使用 Docker，您可以更好地控制執行時間。例如，您可以選擇自訂執行時間，例如 .NET 8.0。您以與任何其他容器映像相同的方式部署 Docker 映像。本教學緊密模擬[教學課程：基本 Lambda 專案](#)，有兩個差異：

- Dockerfile 包含在專案中。
- 已選擇替代發佈組態。

如需 Lambda 容器映像的相關資訊，請參閱《AWS Lambda 開發人員指南》中的[Lambda 部署套件](#)。

如需使用 Lambda 的詳細資訊 AWS Toolkit for Visual Studio，請參閱《使用者指南》中的[使用 AWS Lambda 範本 AWS Toolkit for Visual Studio](#)主題。

建立 Visual Studio .NET Core Lambda 專案

您可以使用 Lambda Visual Studio 範本和藍圖來協助加速專案初始化。Lambda 藍圖包含預先編寫的函數，可簡化彈性專案基礎的建立。

建立 Visual Studio .NET Core Lambda 專案

1. 從 Visual Studio 展開檔案功能表，展開新增，然後選擇專案。
2. 在新增專案對話方塊中，將語言、平台和專案類型下拉式方塊設定為「全部」，然後在**aws lambda**搜尋欄位中輸入。選擇 AWS Lambda 專案 (.NET Core - C#) 範本。

3. 在專案名稱欄位中，輸入 **AWSLambdaDocker**，指定檔案位置，然後選擇建立。
4. 在選取藍圖頁面上，選擇 .NET 8（容器映像）藍圖，然後選擇完成以建立 Visual Studio 專案。您現在可以檢閱專案的結構和程式碼。

檢閱專案檔案

下列各節會檢查 .NET 8（容器映像）藍圖建立的三個專案檔案：

1. Dockerfile
2. aws-lambda-tools-defaults.json
3. Function.cs

1. Dockerfile

會 Dockerfile 執行三個主要動作：

- FROM：建立要用於此映像的基本映像。此基礎映像提供 .NET 執行期、Lambda 執行期，以及提供 Lambda .NET 程序進入點的 Shell 指令碼。
- WORKDIR：將映像的內部工作目錄建立為 /var/task。
- COPY：會將建置程序產生的檔案從其本機位置複製到映像的工作目錄。

以下是您可以指定的選用 Dockerfile 動作：

- ENTRYPOINT：基礎映像已包含 ENTRYPOINT，這是啟動映像時執行的啟動程序。如果您想要指定自己的，則您要覆寫該基本進入點。
- CMD：指示您想要執行 AWS 的自訂程式碼。它預期您的自訂方法具有完整名稱。這行需要直接包含在 Dockerfile 中，也可以在發佈過程中指定。

```
# Example of alternative way to specify the Lambda target method rather than during  
the publish process.  
CMD [ "AWSLambdaDocker::AWSLambdaDocker.Function::FunctionHandler"]
```

以下是 .NET 8（容器映像）藍圖建立的 Dockerfile 範例。

```
FROM public.ecr.aws/lambda/dotnet:8
```

```
WORKDIR /var/task

# This COPY command copies the .NET Lambda project's build artifacts from the host
# machine into the image.
# The source of the COPY should match where the .NET Lambda project publishes its build
# artifacts. If the Lambda function is being built
# with the AWS .NET Lambda Tooling, the `--docker-host-build-output-dir` switch
# controls where the .NET Lambda project
# will be built. The .NET Lambda project templates default to having `--docker-host-
# build-output-dir`
# set in the aws-lambda-tools-defaults.json file to "bin/Release/lambda-publish".
#
# Alternatively Docker multi-stage build could be used to build the .NET Lambda project
# inside the image.
# For more information on this approach checkout the project's README.md file.
COPY "bin/Release/lambda-publish" .
```

2. aws-lambda-tools-defaults.json

`aws-lambda-tools-defaults.json` 檔案用於指定 Toolkit for Visual Studio 部署精靈和 .NET Core CLI 的預設值。下列清單說明您可以在 `aws-lambda-tools-defaults.json` 檔案中設定的欄位。

- `profile` : 設定您的 AWS 設定檔。
- `region` : 設定儲存資源 AWS 的區域。
- `configuration` : 設定用於發佈函數的組態。
- `package-type` : 將部署套件類型設定為容器映像或 .zip 檔案封存。
- `function-memory-size` : 設定函數的記憶體配置，以 MB 為單位。
- `function-timeout` : 逾時是 Lambda 函數可以執行的秒數上限。您可以以 1 秒的增量調整，直到最大值達到 15 分鐘。
- `docker-host-build-output-dir` : 設定與 `Dockerfile` 中指示相關的建置程序輸出目錄。
- `image-command` : 是您方法的完整名稱，也就是您希望 Lambda 函數執行的程式碼。語法為：`{Assembly}::{Namespace}.{ClassName}::{MethodName}`。如需詳細資訊，請參閱[處理常式簽章](#)。在此設定會稍後在 Visual Studio 的發佈精靈中 `image-command` 預先填入此值。

以下是 .NET 8 (容器映像) 藍圖所建立 `aws-lambda-tools-defaults.json` 的範例。

```
{
```

```
"Information": [
  "This file provides default values for the deployment wizard inside Visual Studio
  and the AWS Lambda commands added to the .NET Core CLI.",
  "To learn more about the Lambda commands with the .NET Core CLI execute the
  following command at the command line in the project root directory.",
  "dotnet lambda help",
  "All the command line options for the Lambda command can be specified in this
  file."
],
"profile": "default",
"region": "us-west-2",
"configuration": "Release",
"package-type": "image",
"function-memory-size": 512,
"function-timeout": 30,
"image-command": "AWSLambdaDocker::AWSLambdaDocker.Function::FunctionHandler",
"docker-host-build-output-dir": "./bin/Release/lambda-publish"
}
```

3. Function.cs

Function.cs 檔案會定義要公開為 Lambda 函數的 c# 函數。FunctionHandler 是在 Lambda 函數執行時執行的 Lambda 功能。在此專案中，ToUpper() 會FunctionHandler呼叫輸入文字。

發佈至 Lambda

建置程序產生的 Docker 映像會上傳至 Amazon Elastic Container Registry (Amazon ECR)。Amazon ECR 是全受管的 Docker 容器登錄檔，可用來存放、管理和部署 Docker 容器映像。Amazon ECR 託管映像，然後 Lambda 會參考該映像，以在調用時提供程式設計的 Lambda 功能。

將您的函數發佈至 Lambda

1. 從解決方案總管開啟專案的內容選單（按一下滑鼠右鍵），然後選擇發佈至 AWS Lambda以開啟上傳 Lambda 函數視窗。
2. 從上傳 Lambda 函數頁面，執行下列動作：

Upload to AWS Lambda

aws Upload Lambda Function

Enter the details about the function you want to upload.

AWS Credentials: Profile: Default Region: US West (Oregon)

Package Type: Image

Lambda Runtime: Not Applicable to Image based Functions

Architecture: x86 ARM

Function Name: Create new function
LambdafunctionDocker
 Re-deploy to existing

Description:

Image Command: AWSLambdaDocker::AWSLambdaDocker.Function::FunctionHandler

Image Repo: awslambdadocker Image Tag: latest


Close Back Next Upload

- 對於套件類型，**Image** 已自動選取為套件類型，因為發佈精靈在您的專案Dockerfile中偵測到。
- 在函數名稱中，輸入 Lambda 執行個體的顯示名稱。此名稱是 Visual Studio 中 AWS Explorer 和 中顯示的參考名稱 AWS 管理主控台。
- 在描述中，輸入要在 中與您的執行個體一起顯示的文字 AWS 管理主控台。
- 在 Image Command 中，輸入您要 Lambda 函數執行之方法的完整路徑：
AWSLambdaDocker::AWSLambdaDocker.Function::FunctionHandler

Note


此處輸入的任何方法名稱都會覆寫 Dockerfile 內的任何 CMD 指令。只有在您的 Dockerfile 包含 CMD 以指示如何啟動 Lambda 函數時，輸入 Image Command 才為選用。

- e. 在映像儲存庫中，輸入新的或現有 Amazon Elastic Container Registry 的名稱。建置程序建立的 Docker 映像會上傳至此登錄檔。正在發佈的 Lambda 定義將參考該 Amazon ECR 映像。
 - f. 在映像標籤中，輸入 Docker 標籤以與儲存庫中的映像建立關聯。
 - g. 選擇下一步。
3. 在進階函數詳細資訊頁面上，在角色名稱中選擇與您的帳戶相關聯的角色。角色用於為 函數中程式碼所做的任何 Amazon Web Services 呼叫提供臨時登入資料。如果您沒有角色，請根據 AWS 受管政策選擇新角色，然後選擇 `AWSLambdaBasicExecutionRole`。

 Note

您的帳戶必須具有執行 IAM ListPolicies 動作的許可，否則角色名稱清單將為空白。

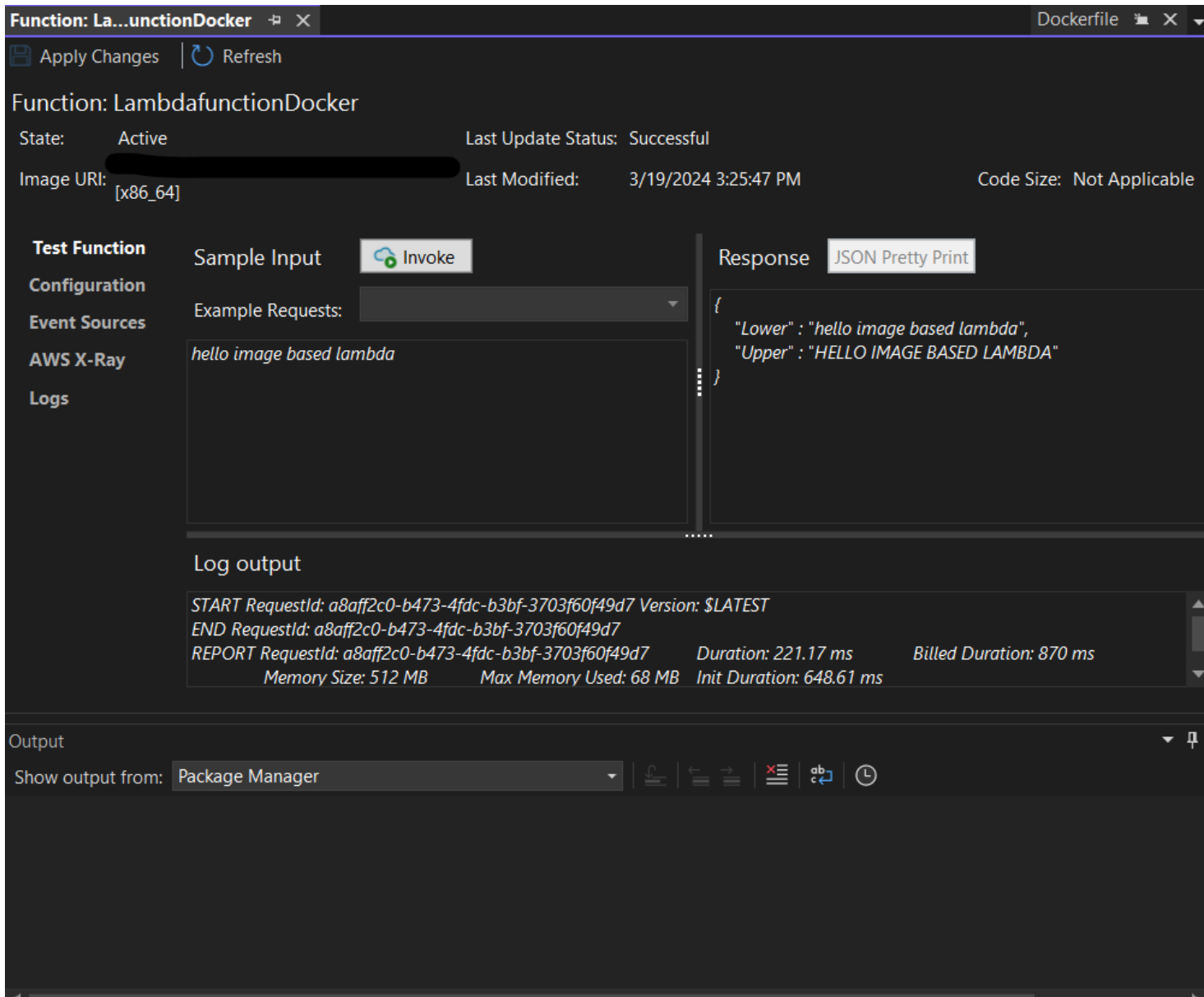
4. 選擇上傳以開始上傳和發佈程序。

 Note

上傳函數頁面會在函數上傳時顯示。發佈程序接著會根據組態參數建置映像、視需要建立 Amazon ECR 儲存庫、將映像上傳至儲存庫，以及使用該映像建立參考該儲存庫的 Lambda。

上傳函數後，函數頁面會開啟並顯示新的 Lambda 函數組態。

5. 若要手動叫用 Lambda 函數，請在測試函數索引標籤上，`hello image based lambda`在請求任意文字輸入欄位中輸入，然後選擇叫用。轉換為大寫的文字將顯示在回應中。



The screenshot displays the AWS Lambda console interface for a function named "LambdafunctionDocker". The function is in an "Active" state with a "Successful" last update status. The image URI is [x86_64]. The last modified date is 3/19/2024 3:25:47 PM, and the code size is not applicable.

The "Test Function" section shows a sample input of "hello image based lambda" and a response of {"Lower": "hello image based lambda", "Upper": "HELLO IMAGE BASED LAMBDA"}. The "Log output" section shows the following details:

```
START RequestId: a8aff2c0-b473-4fdc-b3bf-3703f60f49d7 Version: $LATEST
END RequestId: a8aff2c0-b473-4fdc-b3bf-3703f60f49d7
REPORT RequestId: a8aff2c0-b473-4fdc-b3bf-3703f60f49d7    Duration: 221.17 ms    Billed Duration: 870 ms
                Memory Size: 512 MB    Max Memory Used: 68 MB    Init Duration: 648.61 ms
```

6. 若要檢視儲存庫，請在 AWS Explorer 的 Amazon Elastic Container Service 下，選擇儲存庫。

您可以隨時在節點下的 AWS Explorer 中按兩下已部署的執行個體，以重新開啟 函數：檢視AWS Lambda。

Note

如果您的 AWS Explorer 視窗未開啟，您可以透過檢視 -> AWS Explorer 將其停駐

- 請注意組態索引標籤上的其他特定影像組態選項。此標籤提供覆寫 Dockerfile ENTRYPOINT 中 WORKDIR 可能已指定的 CMD、和 的方法。描述是您在上傳/發佈期間輸入的描述 (如果有的話)。

清除

如果您不會繼續使用此範例繼續開發，請記得刪除已部署的 函數和 ECR 映像，這樣您就不會因帳戶中未使用的資源而收到費用。

- 在節點下 AWS Explorer 中部署的執行個體上按一下滑鼠右鍵，即可刪除函數 AWS Lambda。
- 您可以在 Amazon Elastic Container Service -> 儲存庫下的 AWS Explorer 中刪除儲存庫。

後續步驟

如需有關建立和測試 Lambda 映像的資訊，請參閱[搭配使用容器映像與 Lambda](#)。

如需容器映像部署、許可和覆寫組態設定的相關資訊，請參閱[設定 函數](#)。

教學課程：使用 建置和測試無伺服器應用程式 AWS Lambda

您可以使用 AWS Toolkit for Visual Studio 範本建置無伺服器 Lambda 應用程式。Lambda 專案範本包含一個用於 AWS 無伺服器應用程式的範本，這是[AWS 無伺服器應用程式模型 \(AWS SAM\)](#) 的 AWS Toolkit for Visual Studio 實作。使用此專案類型，您可以開發一組 AWS Lambda 函數，並使用 AWS CloudFormation 來協調部署，並使用任何必要的 AWS 資源將其部署為整個應用程式。

如需設定的先決條件和資訊 AWS Toolkit for Visual Studio，請參閱[AWS Toolkit for Visual Studio 中的使用 AWS Lambda 範本](#)。

主題

- [建立新的無 AWS 伺服器應用程式專案](#)
- [檢閱無伺服器應用程式檔案](#)
- [部署無伺服器應用程式](#)
- [測試無伺服器應用程式](#)

建立新的無 AWS 伺服器應用程式專案

AWS 無伺服器應用程式專案使用無伺服器 CloudFormation 範本建立 Lambda 函數。CloudFormation 範本可讓您定義其他資源，例如資料庫、新增 IAM 角色，以及一次部署多個函數。這與 AWS 專注於開發和部署單一 Lambda 函數的 Lambda 專案不同。

下列程序說明如何建立新的 AWS 無伺服器應用程式專案。

1. 從 Visual Studio 展開檔案功能表，展開新增，然後選擇專案。
2. 在新增專案對話方塊中，確保語言、平台和專案類型下拉式方塊設定為「全部...」，然後在 **aws lambda** 搜尋欄位中輸入。
3. 選取 AWS Serverless Application with Tests (.NET Core - C#) 範本。

Note

AWS Serverless Application with Tests (.NET Core - C#) 範本可能不會填入結果頂端。

4. 按一下下一步以開啟設定您的新專案對話方塊。
5. 從設定您的新專案對話方塊中，輸入 **ServerlessPowertools** 做為名稱，然後根據您的偏好設定完成其餘欄位。選擇建立按鈕以繼續至選取藍圖對話方塊。
6. 從選取藍圖對話方塊中選擇藍圖的 Powertools AWS Lambda，然後選擇完成以建立 Visual Studio 專案。

檢閱無伺服器應用程式檔案

以下各節詳細說明為您的專案建立的三個無伺服器應用程式檔案：

1. serverless.template
2. Functions.cs
3. aws-lambda-tools-defaults.json

1. serverless.template

`serverless.template` 檔案是宣告無伺服器函數和其他 AWS 資源的 AWS CloudFormation 範本。此專案隨附的檔案包含單一 Lambda 函數的宣告，該函數將透過 Amazon API Gateway 公開為 HTTP `*Get*` 操作。您可以編輯此範本來自訂現有的函數，或新增應用程式所需的更多函數和其他資源。

以下是 `serverless.template` 檔案的範例：

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Transform": "AWS::Serverless-2016-10-31",
  "Description": "An AWS Serverless Application.",
  "Resources": {
    "Get": {
      "Type": "AWS::Serverless::Function",
      "Properties": {
        "Architectures": [
          "x86_64"
        ],
        "Handler": "ServerlessPowertools::ServerlessPowertools.Functions::Get",
        "Runtime": "dotnet8",
        "CodeUri": "",
        "MemorySize": 512,
        "Timeout": 30,
        "Role": null,
        "Policies": [
          "AWSLambdaBasicExecutionRole"
        ],
        "Environment": {
          "Variables": {
            "POWERTOOLS_SERVICE_NAME": "ServerlessGreeting",
            "POWERTOOLS_LOG_LEVEL": "Info",
            "POWERTOOLS_LOGGER_CASE": "PascalCase",
            "POWERTOOLS_TRACER_CAPTURE_RESPONSE": true,
            "POWERTOOLS_TRACER_CAPTURE_ERROR": true,
            "POWERTOOLS_METRICS_NAMESPACE": "ServerlessGreeting"
          }
        },
        "Events": {
          "RootGet": {
            "Type": "Api",
            "Properties": {
              "Path": "/",
              "Method": "GET"
            }
          }
        }
      }
    }
  }
},
```

```
"Outputs": {
  "ApiURL": {
    "Description": "API endpoint URL for Prod environment",
    "Value": {
      "Fn::Sub": "https://${ServerlessRestApi}.execute-api.
${AWS::Region}.amazonaws.com/Prod/"
    }
  }
}
```

請注意，許多...AWS::Serverless::Function...宣告欄位與 Lambda 專案部署的欄位類似。Powertools 記錄、指標和追蹤是透過下列環境變數設定：

- POWERTOOLS_SERVICE_NAME=ServerlessGreeting
- POWERTOOLS_LOG_LEVEL=Info
- POWERTOOLS_LOGGER_CASE=PascalCase
- POWERTOOLS_TRACER_CAPTURE_RESPONSE=true
- POWERTOOLS_TRACER_CAPTURE_ERROR=true
- POWERTOOLS_METRICS_NAMESPACE=ServerlessGreeting

如需環境變數的定義和其他詳細資訊，請參閱[AWS Lambda 參考資料網站的 Powertools](#)。

2. Functions.cs

Functions.cs 是一種類別檔案，其中包含對應至範本檔案中宣告之單一函數的 C# 方法。Lambda 函數會回應來自 API Gateway HTTP Get 的方法。以下是 Functions.cs 檔案的範例：

```
public class Functions
{
    [Logging(LogEvent = true, CorrelationIdPath = CorrelationIdPaths.ApiGatewayRest)]
    [Metrics(CaptureColdStart = true)]
    [Tracing(CaptureMode = TracingCaptureMode.ResponseAndError)]
    public APIGatewayProxyResponse Get(APIGatewayProxyRequest request, ILambdaContext context)
    {
        Logger.LogInformation("Get Request");

        var greeting = GetGreeting();
    }
}
```

```
        var response = new APIGatewayProxyResponse
        {
            StatusCode = (int)HttpStatusCode.OK,
            Body = greeting,
            Headers = new Dictionary (string, string) { { "Content-Type", "text/
plain" } }
        };

        return response;
    }

    [Tracing(SegmentName = "GetGreeting Method")]
    private static string GetGreeting()
    {
        Metrics.AddMetric("GetGreeting_Invocations", 1, MetricUnit.Count);

        return "Hello Powertools for AWS Lambda (.NET)";
    }
}
```

3. aws-lambda-tools-defaults.json

aws-lambda-tools-defaults.json 提供 Visual Studio 內 AWS 部署精靈的預設值，以及新增至 .NET Core CLI 的 AWS Lambda 命令。以下是此專案隨附的 aws-lambda-tools-defaults.json 檔案範例：


```
{
  "profile": "Default",
  "region": "us-east-1",
  "configuration": "Release",
  "s3-prefix": "ServerlessPowertools/",
  "template": "serverless.template",
  "template-parameters": ""
}
```

部署無伺服器應用程式

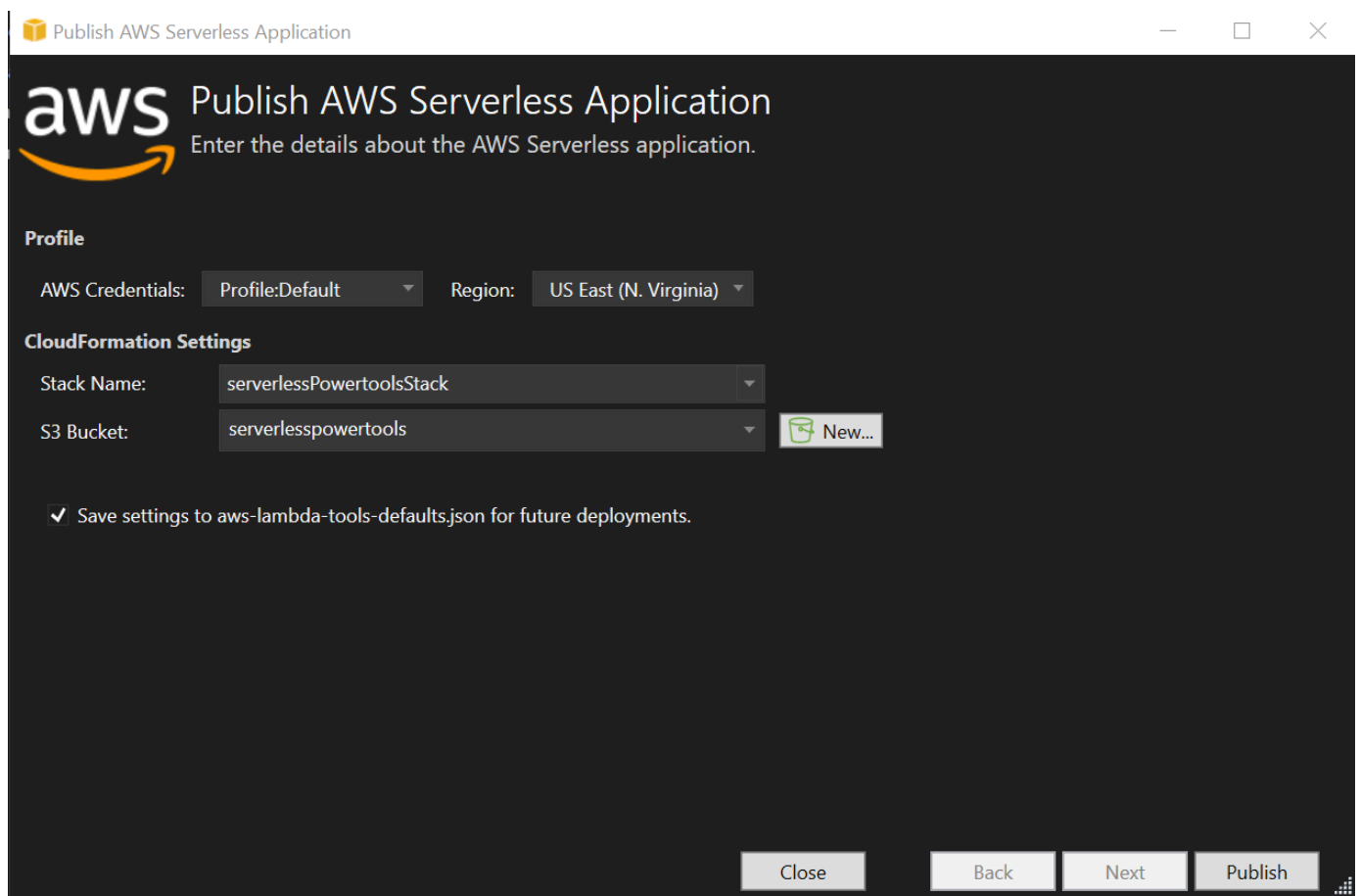
若要部署無伺服器應用程式，請完成下列步驟

1. 從解決方案總管開啟專案的內容選單（按一下滑鼠右鍵），然後選擇發佈至 AWS Lambda 以開啟發佈無 AWS 伺服器應用程式對話方塊。

2. 從發佈無 AWS 伺服器應用程式對話方塊中，在 CloudFormation 堆疊名稱欄位中輸入堆疊容器的名稱。
3. 在 S3 儲存貯體欄位中，選擇應用程式套件將上傳到的 Amazon S3 儲存貯體，或選擇新增...按鈕，然後輸入新 Amazon S3 儲存貯體的名稱。然後選擇發佈以發佈以部署您的應用程式。

 Note

您的 CloudFormation 堆疊和 Amazon S3 儲存貯體必須存在於相同的 AWS 區域中。您專案的其餘設定會在 `serverless.template` 檔案中定義。



4. 當部署完成時，堆疊檢視視窗會在發佈程序期間開啟，狀態欄位會顯示：CREATE_COMPLETE。

Resources	Time	Type	Logical ID	Physical ID	Status	Reason
Monitoring	3/29/2024 12:45:26 PM	AWS::CloudFormation::Stack	serverlessPowertoolsStack	arn:aws:cloudformation:us-east-1:50k...	CREATE_COMPLETE	
Template	3/29/2024 12:45:25 PM	AWS::ApiGateway::Stage	ServerlessRestApiProdStage	Prod	CREATE_COMPLETE	
Parameters	3/29/2024 12:45:25 PM	AWS::ApiGateway::Stage	ServerlessRestApiProdStage	Prod	CREATE_IN_PROGRESS	Resource not yet created
Outputs	3/29/2024 12:45:24 PM	AWS::ApiGateway::Stage	ServerlessRestApiProdStage		CREATE_IN_PROGRESS	Resource not yet created
	3/29/2024 12:45:23 PM	AWS::Lambda::Function	Get	serverlessPowertoolsStack-Get-Lgaks	CREATE_COMPLETE	
	3/29/2024 12:45:23 PM	AWS::ApiGateway::Deployment	ServerlessRestApiDeployment9d78fb6c57	qpdttli	CREATE_COMPLETE	
	3/29/2024 12:45:23 PM	AWS::ApiGateway::Deployment	ServerlessRestApiDeployment9d78fb6c57	qpdttli	CREATE_IN_PROGRESS	Resource not yet created
	3/29/2024 12:45:22 PM	AWS::Lambda::Permission	GetRootGetPermissionProd	serverlessPowertoolsStack-GetRootGi	CREATE_COMPLETE	
	3/29/2024 12:45:22 PM	AWS::Lambda::Permission	GetRootGetPermissionProd	serverlessPowertoolsStack-GetRootGi	CREATE_IN_PROGRESS	Resource not yet created
	3/29/2024 12:45:21 PM	AWS::ApiGateway::Deployment	ServerlessRestApiDeployment9d78fb6c57		CREATE_IN_PROGRESS	Resource not yet created
	3/29/2024 12:45:21 PM	AWS::Lambda::Permission	GetRootGetPermissionProd		CREATE_IN_PROGRESS	Resource not yet created
	3/29/2024 12:45:21 PM	AWS::ApiGateway::RestApi	ServerlessRestApi	bhntmpmjoj	CREATE_COMPLETE	
	3/29/2024 12:45:20 PM	AWS::ApiGateway::RestApi	ServerlessRestApi	bhntmpmjoj	CREATE_IN_PROGRESS	Resource not yet created
	3/29/2024 12:45:19 PM	AWS::ApiGateway::RestApi	ServerlessRestApi		CREATE_IN_PROGRESS	Resource not yet created
	3/29/2024 12:45:18 PM	AWS::Lambda::Function	Get	serverlessPowertoolsStack-Get-Lgaks	CREATE_IN_PROGRESS	Event not yet created
	3/29/2024 12:45:17 PM	AWS::Lambda::Function	Get	serverlessPowertoolsStack-Get-Lgaks	CREATE_IN_PROGRESS	Resource not yet created
	3/29/2024 12:45:16 PM	AWS::Lambda::Function	Get		CREATE_IN_PROGRESS	Resource not yet created
	3/29/2024 12:45:15 PM	AWS::IAM::Role	GetRole	serverlessPowertoolsStack-GetRole-D	CREATE_COMPLETE	
	3/29/2024 12:44:59 PM	AWS::IAM::Role	GetRole	serverlessPowertoolsStack-GetRole-D	CREATE_IN_PROGRESS	Resource not yet created
	3/29/2024 12:44:58 PM	AWS::IAM::Role	GetRole		CREATE_IN_PROGRESS	Resource not yet created
	3/29/2024 12:44:55 PM	AWS::CloudFormation::Stack	serverlessPowertoolsStack	arn:aws:cloudformation:us-east-1:50k...	CREATE_IN_PROGRESS	User Input required
	3/29/2024 12:44:49 PM	AWS::CloudFormation::Stack	serverlessPowertoolsStack	arn:aws:cloudformation:us-east-1:50k...	REVIEW_IN_PROGRESS	User Input required

測試無伺服器應用程式

當堆疊建立完成時，您可以使用AWS無伺服器URL檢視您的應用程式。如果您已完成本教學課程，但未新增任何其他函數或參數，存取無AWS伺服器URL會在Web瀏覽器中顯示下列片語：Hello Powertools for AWS Lambda (.NET)。

教學課程：建立 Amazon Rekognition Lambda 應用程式

本教學課程說明如何建立使用 Amazon Rekognition 以偵測到的標籤標記 Amazon S3 物件的 Lambda 應用程式。

如需設定的先決條件和資訊 AWS Toolkit for Visual Studio，請參閱 [AWS Toolkit for Visual Studio 中的使用 AWS Lambda 範本](#)。

建立 Visual Studio .NET Core Lambda Image Rekognition 專案

下列程序說明如何從 建立 Amazon Rekognition Lambda 應用程式 AWS Toolkit for Visual Studio。

Note

建立時，您的應用程式具有兩個專案的解決方案：包含要部署至 Lambda 之 Lambda 函數程式碼的來源專案，以及使用 xUnit 在本機測試函數的測試專案。

有時 Visual Studio 找不到專案的所有 NuGet 參考。這是因為藍圖需要必須從 NuGet 擷取的相依性。建立新專案時，Visual Studio 只會提取本機參考，而不是從 NuGet 提取遠端參考。若要修正 NuGet 錯誤：在參考上按一下滑鼠右鍵，然後選擇還原套件。

1. 從 Visual Studio 展開檔案功能表，展開新增，然後選擇專案。
2. 在新增專案對話方塊中，確保語言、平台和專案類型下拉式方塊設定為「全部...」，然後在 **aws lambda** 搜尋欄位中輸入。
3. 選取 AWS Lambda 含測試 (.NET Core - C#) 範本的。
4. 按一下下一步以開啟設定您的新專案對話方塊。
5. 從設定您的新專案對話方塊中，輸入名稱的「ImageRekognition」，然後根據您的偏好設定完成其餘欄位。選擇建立按鈕以繼續至選取藍圖對話方塊。
6. 從選取藍圖對話方塊中，選擇偵測影像標籤藍圖，然後選擇完成以建立 Visual Studio 專案。

Note

此藍圖提供用於接聽 Amazon S3 事件的程式碼，並使用 Amazon Rekognition 偵測標籤，並將其新增至 S3 物件做為標籤。

檢閱專案檔案

下列各節會檢查這些專案檔案：

1. `Function.cs`
2. `aws-lambda-tools-defaults.json`

1. `Function.cs`

在 `Function.cs` 檔案中，程式碼的第一個區段是位於檔案頂端的組件屬性。根據預設，Lambda 只接受輸入參數和傳回類型的 `System.IO.Stream`。您必須註冊序列化程式，才能針對輸入參數和傳回類

型使用類型類別。組件屬性會註冊 Lambda JSON 序列化程式，其會使用 `Newtonsoft.Json` 將串流轉換為輸入類別。您可以在組件或方法層級設定序列化程式。

以下是組件屬性的範例：

```
// Assembly attribute to enable the Lambda function's JSON input to be converted into
a .NET class.
[assembly:
    LambdaSerializer(typeof(Amazon.Lambda.Serialization.SystemTextJson.DefaultLambdaJsonSerializer))
```

類別有兩個建構函數。第一個是 Lambda 調用函數時使用的預設建構函數。此建構函數會建立 Amazon S3 和 Amazon Rekognition 服務用戶端。建構函數也會從您在部署這些用戶端時指派給函數的 IAM 角色擷取這些用戶端的 AWS 登入資料。用戶端 AWS 的區域設定為 Lambda 函數執行所在的區域。在此藍圖中，只有在 Amazon Rekognition 服務對標籤具有最低的可信度時，才想要將標籤新增至 Amazon S3 物件。Amazon Rekognition 此建構函數會檢查環境變數 `MinConfidence`，以判斷可接受的可信度等級。您可以在部署 Lambda 函數時設定此環境變數。

以下是 中第一類建構函數的範例 `Function.cs`：

```
public Function()
{
    this.S3Client = new AmazonS3Client();
    this.RekognitionClient = new AmazonRekognitionClient();

    var environmentMinConfidence =
        System.Environment.GetEnvironmentVariable(MIN_CONFIDENCE_ENVIRONMENT_VARIABLE_NAME);
    if(!string.IsNullOrEmpty(environmentMinConfidence))
    {
        float value;
        if(float.TryParse(environmentMinConfidence, out value))
        {
            this.MinConfidence = value;
            Console.WriteLine($"Setting minimum confidence to {this.MinConfidence}");
        }
        else
        {
            Console.WriteLine($"Failed to parse value {environmentMinConfidence} for
            minimum confidence. Reverting back to default of {this.MinConfidence}");
        }
    }
    else
    {
```

```
        Console.WriteLine($"Using default minimum confidence of {this.MinConfidence}");
    }
}
```

下列範例示範如何利用第二個建構函數進行測試。測試專案會設定自己的 S3 和 Rekognition 用戶端，並將其傳遞至：

```
public Function(IAmazonS3 s3Client, IAmazonRekognition rekognitionClient, float
    minConfidence)
{
    this.S3Client = s3Client;
    this.RekognitionClient = rekognitionClient;
    this.MinConfidence = minConfidence;
}
```

以下是 Function.cs 檔案內 FunctionHandler 方法的範例。

```
public async Task FunctionHandler(S3Event input, ILambdaContext context)
{
    foreach(var record in input.Records)
    {
        if(!SupportedImageTypes.Contains(Path.GetExtension(record.S3.Object.Key)))
        {
            Console.WriteLine($"Object {record.S3.Bucket.Name}:{record.S3.Object.Key}
is not a supported image type");
            continue;
        }

        Console.WriteLine($"Looking for labels in image {record.S3.Bucket.Name}:
{record.S3.Object.Key}");
        var detectResponses = await this.RekognitionClient.DetectLabelsAsync(new
DetectLabelsRequest
        {
            MinConfidence = MinConfidence,
            Image = new Image
            {
                S3Object = new Amazon.Rekognition.Model.S3Object
                {
                    Bucket = record.S3.Bucket.Name,
                    Name = record.S3.Object.Key
                }
            }
        });
    });
}
```

```
var tags = new List();
foreach(var label in detectResponses.Labels)
{
    if(tags.Count < 10)
    {
        Console.WriteLine($"\\tFound Label {label.Name} with confidence
{label.Confidence}");
        tags.Add(new Tag { Key = label.Name, Value =
label.Confidence.ToString() });
    }
    else
    {
        Console.WriteLine($"\\tSkipped label {label.Name} with confidence
{label.Confidence} because maximum number of tags reached");
    }
}

await this.S3Client.PutObjectTaggingAsync(new PutObjectTaggingRequest
{
    BucketName = record.S3.Bucket.Name,
    Key = record.S3.Object.Key,
    Tagging = new Tagging
    {
        TagSet = tags
    }
});
}
return;
}
```

FunctionHandler 是 Lambda 在建構執行個體之後呼叫的方法。請注意，輸入參數為類型 S3Event，而非 Stream。您可以因為已註冊的 Lambda JSON 序列化程式而執行此操作。S3Event 包含 Amazon S3 中觸發之事件的所有資訊。函數會循環瀏覽屬於事件一部分的所有 S3 物件，並告知 Rekognition 偵測標籤。偵測到標籤後，它們會新增為 S3 物件的標籤。

Note

此程式碼包含對的呼叫 Console.WriteLine()。當函數在 Lambda 中執行時，所有 Console.WriteLine() 重新導向至 Amazon CloudWatch Logs 的呼叫。

2. aws-lambda-tools-defaults.json

`aws-lambda-tools-defaults.json` 檔案包含藍圖設定為預先填入部署精靈中部分欄位的預設值。它也有助於設定命令列選項以與 .NET Core CLI 整合。

若要存取 .NET Core CLI 整合，請導覽至函數的專案目錄並輸入 **dotnet lambda help**。

Note

函數處理常式指出 Lambda 呼叫 以回應調用函數的方法。此欄位的格式為：`<assembly-name>::<full-type-name>::<method-name>`。命名空間必須包含在類型名稱中。

部署 函數

下列程序說明如何部署 Lambda 函數。

1. 從解決方案總管中，在 Lambda 專案上按一下滑鼠右鍵，然後選擇發佈至 AWS Lambda 以開啟上傳至 AWS Lambda 視窗。

Note

預設值會從 `aws-lambda-tools-defaults.json` 檔案擷取。

2. 從上傳到 AWS Lambda 視窗中，在函數名稱欄位中輸入名稱，然後選擇下一步按鈕以進入進階函數詳細資訊視窗。

Note

此範例使用函數名稱 **ImageRekognition**。

3. 從進階函數詳細資訊視窗中，選取授予程式碼存取 Amazon S3 和 Amazon Rekognition 資源許可的 IAM 角色。

Note

如果您遵循此範例，請選取AWSLambda_FullAccess角色。

4. 將環境變數MinConfidence設定為 60，然後選擇上傳以啟動部署程序。當函數檢視顯示在 AWS Explorer 中時，發佈程序即完成。

Upload to AWS Lambda

aws Advanced Function Details

Configure additional settings for your function.

Permissions

Select an IAM role to provide AWS credentials to our Lambda function allowing access to AWS Services like S3.

Role Name:

Execution

Memory (MB):

Timeout (Secs): (1 - 900)

VPC

If your function accesses resources in a VPC, select the list of subnets and security group IDs (these must belong to the same VPC).

VPC Subnets:

Security Groups:

Debugging and Error Handling

DLQ Resource:

Enable active tracing (AWS X-Ray) [Learn More.](#)

Environment

KMS Key:

Variable	Value
MinConfidence	60

Add...

Close Back Next Upload

- 成功部署後，請導覽至事件來源索引標籤，設定 Amazon S3 將其事件傳送至您的新函數。
- 從事件來源索引標籤中，選擇新增按鈕，然後選取要與您的 Lambda 函數連線的 Amazon S3 儲存貯體。

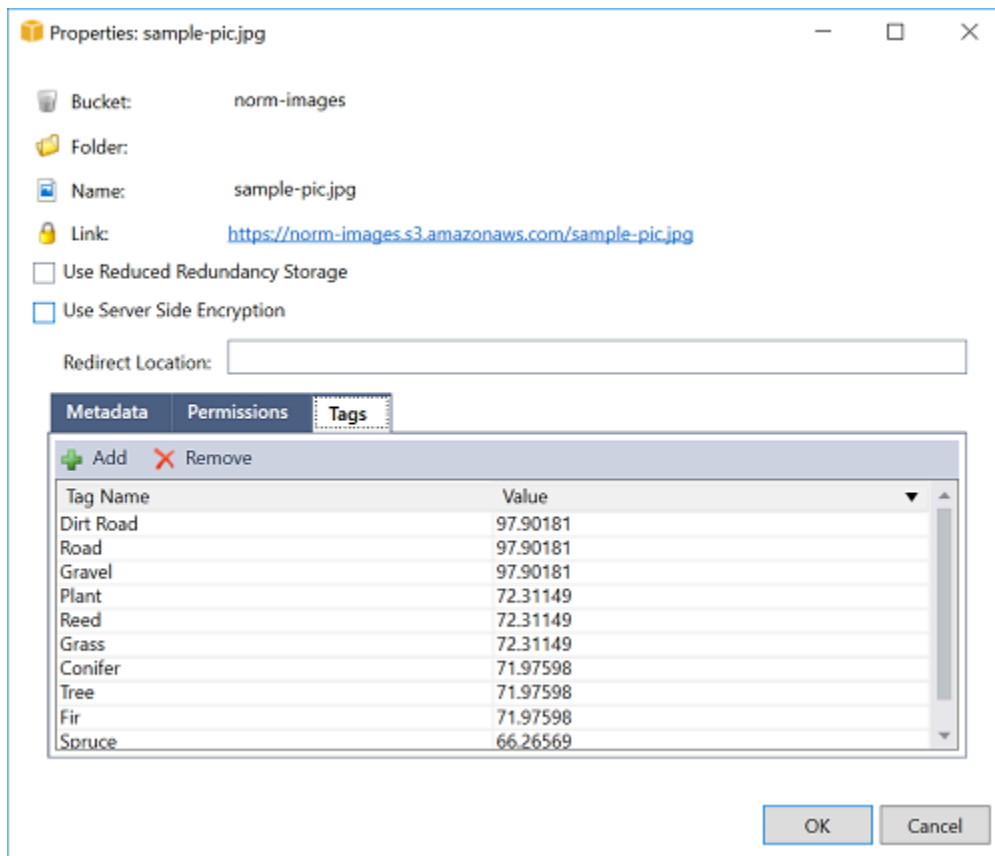
Note

儲存貯體必須與 Lambda 函數位於相同的 AWS 區域。

測試 函數

現在已部署 函數，並將 S3 儲存貯體設定為其事件來源，請從您選取的儲存貯體的 AWS Explorer 開啟 S3 儲存貯體瀏覽器。然後上傳一些影像。

上傳完成時，您可以查看函數檢視中的日誌，以確認函數執行。或者，在儲存貯體瀏覽器中的映像上按一下滑鼠右鍵，然後選擇屬性。在標籤索引標籤上，您可以檢視套用至物件的標籤。



教學課程：搭配使用 Amazon Logging Framework AWS Lambda 來建立應用程式日誌

您可以使用 Amazon CloudWatch Logs 來監控、存放和存取應用程式的日誌。若要取得 CloudWatch Logs 的日誌資料，請使用 AWS SDK 或安裝 CloudWatch Logs 代理程式來監控特定日誌資料夾。CloudWatch Logs 與數個熱門的 .NET 記錄架構整合，可簡化工作流程。

若要開始使用 CloudWatch Logs 和 .NET 記錄架構，請將適當的 NuGet 套件和 CloudWatch Logs 輸出來源新增至您的應用程式，然後像平常一樣使用您的記錄程式庫。這可讓您的應用程式使用 .NET 架構記錄訊息，將訊息傳送到 CloudWatch Logs，並在 CloudWatch Logs 主控台中顯示應用程式的日誌訊息。您也可以根據應用程式的日誌訊息，從 CloudWatch Logs 主控台設定指標和警示。

支援的 .NET 記錄架構包括：

- NLog：若要檢視，請參閱 [nuget.org NLog 套件](https://nuget.org/packages/NLog)。
- Log4net：若要檢視，請參閱 [nuget.org Log4net 套件](https://nuget.org/packages/Log4net)。
- ASP.NET Core 記錄架構：若要檢視，請參閱 [nuget.org ASP.NET Core 記錄架構套件](https://nuget.org/packages/ASP.NET%20Core%20Logging)。

以下是 NLog.config 檔案的範例，可透過將 AWS.Logger.NLog NuGet 套件和 AWS 目標新增至，同時啟用 CloudWatch Logs 和 主控台做為日誌訊息的輸出NLog.config。

```
<?xml version="1.0" encoding="utf-8" ?>
<nlog xmlns="http://www.nlog-project.org/schemas/NLog.xsd"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      throwExceptions="true">
  <targets>
    <target name="aws" type="AWSTarget" logGroup="NLog.ConfigExample" region="us-east-1"/>
    <target name="logfile" xsi:type="Console" layout="${callsite} ${message}" />
  </targets>
  <rules>
    <logger name="*" minlevel="Info" writeTo="logfile,aws" />
  </rules>
</nlog>
```

記錄外掛程式全都建置在 之上，適用於 .NET 的 AWS SDK 並在類似於 SDK 的程序中驗證您的 AWS 登入資料。下列範例詳細說明記錄外掛程式登入資料存取 CloudWatch Logs 所需的許可：

Note

AWS .NET 記錄外掛程式是開放原始碼專案。如需其他資訊、範例和指示，請參閱 [AWS 記錄 .NET GitHub](#) 儲存庫中的 [範例](#) 和 [指示](#) 主題。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogGroups"
      ],
      "Resource": [
```

```
    "arn:aws:logs:*:*:*"  
  ]  
}  
]  
}
```

部署至 AWS

Toolkit for Visual Studio 支援將應用程式部署到 AWS Elastic Beanstalk 容器或 CloudFormation 堆疊。

Note

如果您使用的是 Visual Studio Express Edition：

- 您可以使用 [Docker CLI](#) 將應用程式部署至 Amazon ECS 容器。
- 您可以使用 [AWS 管理主控台](#) 將應用程式部署到 Elastic Beanstalk 容器。

對於 Elastic Beanstalk 部署，您必須先建立 Web 部署套件。如需詳細資訊，請參閱[如何：在 Visual Studio 中建立 Web 部署套件](#)。對於 Amazon ECS 部署，您必須擁有 Docker 映像。如需詳細資訊，請參閱 [Visual Studio Tools for Docker](#)。

主題

- [在 Visual Studio AWS 中使用 發佈至](#)
- [使用 .NET Core CLI 部署 AWS Lambda 專案](#)
- [使用 AWS Toolkit for Visual Studio 搭配 Amazon Q 在 Visual Studio AWS Elastic Beanstalk 中部署至](#)
- [部署至 Amazon EC2 Container Service](#)

在 Visual Studio AWS 中使用 發佈至

發佈至 AWS 是一項互動式部署體驗，可協助您將 .NET 應用程式發佈至 AWS 部署目標，並支援以 .NET Core 3.1 及更新版本為目標的應用程式。使用 發佈，直接從您的 IDE 提供這些部署功能，以將工作流程 AWS 保留在 Visual Studio 內：

- 只要按一下，即可部署您的應用程式。
- 根據您的應用程式提供的部署建議。
- 自動建立 Dockerfile，因為 部署目的地環境（部署目標）相關且必要。
- 根據您的部署目標，最佳化應用程式建置和封裝的設定。

Note

如需有關發佈 .NET Framework 應用程式的其他資訊，請參閱《[Elastic Beanstalk 上的建立和部署 .NET 應用程式](#)》指南。

您也可以 AWS 從 .NET CLI 存取的發佈。如需詳細資訊，請參閱《[指南](#)》上的[部署 .NET 應用程式 AWS](#)。

主題

- [先決條件](#)
- [支援的應用程式類型](#)
- [將應用程式發佈至 AWS 目標](#)

先決條件

若要將 .NET 應用程式成功發佈至 AWS 服務，請將下列安裝至本機裝置：

- .NET Core 3.1+ (包括 .NET5 和 .NET6)：如需這些產品和下載資訊的其他資訊，請造訪 [Microsoft 下載網站](#)。
- Node.js 14.x 或更新版本：需要 Node.js 才能執行 AWS Cloud Development Kit (AWS CDK)。若要下載或取得 Node.js 的詳細資訊，請造訪 [Node.js 下載網站](#)。

Note

發佈以 AWS 利用 AWS CDK 將應用程式及其所有部署基礎設施部署為單一專案。如需的詳細資訊，AWS CDK 請參閱 [雲端開發套件](#) 指南。

- (選用) Docker 用於部署到容器型服務，例如 Amazon ECS。如需詳細資訊和下載 Docker，請參閱 [Docker 下載網站](#)。

支援的應用程式類型

在發佈至新的或結束的目標之前，請先在 Visual Studio 中建立或開啟下列其中一個專案類型：

- ASP.NET Core 應用程式
- .NET 主控台應用程式

- Blazor WebAssembly 應用程式

將應用程式發佈至 AWS 目標

發佈至新目標時，發佈至 AWS 將透過提出建議並使用常用設定來引導您完成程序。如果您需要發佈至先前設定的目標，您的偏好設定會儲存並可調整，或可立即用於一鍵式部署。

Note

工具組與 .NET CLI Server 整合：
發佈會在 localhost 上啟動 .NET 伺服器程序，以執行發佈程序。

發佈至新目標

以下說明如何在發佈至新目標時設定發佈至 AWS 部署偏好設定。

1. 從 AWS Explorer 展開登入資料下拉式功能表，然後選擇與部署所需的區域 AWS 和服務對應的 AWS 設定檔。
2. 展開區域下拉式功能表，然後選擇 AWS 包含部署所需 AWS 服務的區域。
3. 從 Visual Studio Solutions Explorer 窗格中，開啟專案名稱的內容選單（按一下滑鼠右鍵），然後選擇發佈至 AWS。這會開啟發佈至 AWS。
4. 從發佈至 AWS，選擇發佈至新目標以設定新的部署。

Note

若要修改您的預設部署登入資料，請選擇或按一下發佈至 AWS 中的登入資料區段旁的編輯連結。
若要略過目標組態程序，請選擇發佈至現有目標，然後從先前部署目標的清單中選擇您偏好的組態。

5. 從發佈目標窗格中，選擇 AWS 服務來管理您的應用程式部署。
6. 當您對組態感到滿意時，請選擇發佈以開始部署程序。

Note

啟動部署後，發佈 AWS 以顯示下列狀態更新：

- 在部署過程中，發佈 以顯示 AWS 部署進度的相關資訊。
- 遵循部署程序，發佈 以 AWS 指出部署成功或失敗。
- 成功部署後，資源面板會提供所建立資源的其他資訊。此資訊會根據應用程式和部署組態的類型而有所不同。

發佈至現有目標

以下說明如何將 .NET 應用程式重新發佈至現有 AWS 目標。

1. 從 AWS Explorer 展開登入資料下拉式功能表，然後選擇與部署所需的區域 AWS 和服務對應的 AWS 設定檔。
2. 展開區域下拉式功能表，然後選擇 AWS 包含部署所需 AWS 服務的區域。
3. 從 Visual Studio Solutions Explorer 窗格中，在專案的名稱上按一下滑鼠右鍵，然後選擇發佈至 AWS 以開啟發佈至 AWS。
4. 從發佈至 AWS，選擇發佈至現有目標，從現有目標清單中選取您的部署環境。

Note

如果您最近已將任何應用程式發佈至 AWS 雲端，這些應用程式會顯示在 發佈至 中 AWS。

5. 選取您要部署應用程式的發佈目標，然後按一下發佈以開始部署程序。

使用 .NET Core CLI 部署 AWS Lambda 專案

AWS Toolkit for Visual Studio 包含 Visual Studio 的 AWS Lambda .NET Core 專案範本。您可以使用 .NET Core 命令列界面 (CLI) 部署 Visual Studio 中內建的 Lambda 函數。

主題

- [先決條件](#)
- [相關主題](#)
- [列出透過 .NET Core CLI 可用的 Lambda 命令](#)
- [從 .NET Core CLI 發佈 .NET Core Lambda 專案](#)

先決條件

使用 .NET Core CLI 部署 Lambda 函數之前，您必須符合下列先決條件：

- 確定已安裝 Visual Studio 2015 Update 3。
- 安裝適用於 [Windows 的 .NET Core](#)。
- 設定 .NET Core CLI 以使用 Lambda。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 [.NET Core CLI](#)。
- 安裝 Toolkit for Visual Studio。如需詳細資訊，請參閱[安裝 AWS Toolkit for Visual Studio](#)。

相關主題

當您使用 .NET Core CLI 部署 Lambda 函數時，下列相關主題會很有幫助：

- 如需 Lambda 函數的詳細資訊，請參閱《AWS Lambda 開發人員指南》中的[什麼是 AWS Lambda?](#)
- 如需在 Visual Studio 中建立 Lambda 函數的詳細資訊，請參閱 [AWS Lambda](#)。
- 如需 Microsoft .NET Core 的詳細資訊，請參閱 Microsoft 線上文件中的 [.NET Core](#)。

列出透過 .NET Core CLI 可用的 Lambda 命令

若要列出可透過 .NET Core CLI 取得的 Lambda 命令，請執行下列動作。

1. 開啟命令提示視窗，然後導覽至包含 Visual Studio .NET Core Lambda 專案的資料夾。
2. 輸入 `dotnet lambda --help`。

```
C:\Lambda\AWSLambda1\AWSLambda1>dotnet lambda --help AWS Lambda Tools for .NET Core
functions
  Project Home: https://github.com/aws/aws-lambda-dotnet
  .
  Commands to deploy and manage Lambda functions:
  .
      deploy-function      Deploy the project to Lambda
      invoke-function      Invoke the function in Lambda with an optional
input
      list-functions       List all of your Lambda functions
      delete-function      Delete a Lambda function
```

```
get-function-config    Get the current runtime configuration for a Lambda
function
update-function-config Update the runtime configuration for a Lambda
function
.
Commands to deploy and manage AWS serverless applications using AWS CloudFormation:
.
    deploy-serverless    Deploy an AWS serverless application
    list-serverless      List all of your AWS serverless applications
    delete-serverless    Delete an AWS serverless application
.
Other Commands:
.
    package              Package a Lambda project into a .zip file ready for
deployment
.
To get help on individual commands, run the following:

    dotnet lambda help <command>
```

從 .NET Core CLI 發佈 .NET Core Lambda 專案

下列指示假設您已在 Visual Studio 中建立 AWS Lambda .NET Core 函數。

1. 開啟命令提示視窗，然後導覽至包含 Visual Studio .NET Core Lambda 專案的資料夾。
2. 輸入 `dotnet lambda deploy-function`。
3. 出現提示時，輸入要部署的函數名稱。它可以是新名稱或現有函數的名稱。
4. 出現提示時，輸入 AWS 區域（將部署 Lambda 函數的區域）。
5. 出現提示時，選取或建立 Lambda 在執行函數時將擔任的 IAM 角色。

成功完成時，會顯示建立的新 Lambda 函數訊息。

```
C:\Lambda\AWSLambda1\AWSLambda1>dotnet lambda deploy-function
Executing publish command
... invoking 'dotnet publish', working folder 'C:\Lambda\AWSLambda1\AWSLambda1\bin
\Release\netcoreapp1.0\publish'
... publish: Publishing AWSLambda1 for .NETCoreApp,Version=v1.0
... publish: Project AWSLambda1 (.NETCoreApp,Version=v1.0) will be compiled because
expected outputs are missing
... publish: Compiling AWSLambda1 for .NETCoreApp,Version=v1.0
```

```
... publish: Compilation succeeded.
... publish:      0 Warning(s)
... publish:      0 Error(s)
... publish: Time elapsed 00:00:01.2479713
... publish:
... publish: publish: Published to C:\Lambda\AWSLambda1\AWSLambda1\bin\Release\netcoreapp1.0\publish
... publish: Published 1/1 projects successfully
Zipping publish folder C:\Lambda\AWSLambda1\AWSLambda1\bin\Release\netcoreapp1.0\publish to C:\Lambda\AWSLambda1\AWSLambda1\bin\Release\netcoreapp1.0\AWSLambda1.zip
Enter Function Name: (AWS Lambda function name)
DotNetCoreLambdaTest
Enter AWS Region: (The region to connect to AWS services)
us-west-2
Creating new Lambda function
Select IAM Role that Lambda will assume when executing function:
    1) lambda_exec_LambdaCoreFunction
    2) *** Create new IAM Role ***
1
New Lambda function created
```

如果您部署現有的 函數，則部署函數只會向 AWS 區域要求。

```
C:\Lambda\AWSLambda1\AWSLambda1>dotnet lambda deploy-function
Executing publish command
Deleted previous publish folder
... invoking 'dotnet publish', working folder 'C:\Lambda\AWSLambda1\AWSLambda1\bin\Release\netcoreapp1.0\publish'
... publish: Publishing AWSLambda1 for .NETCoreApp,Version=v1.0
... publish: Project AWSLambda1 (.NETCoreApp,Version=v1.0) was previously compiled.
Skipping compilation.
... publish: publish: Published to C:\Lambda\AWSLambda1\AWSLambda1\bin\Release\netcoreapp1.0\publish
... publish: Published 1/1 projects successfully
Zipping publish folder C:\Lambda\AWSLambda1\AWSLambda1\bin\Release\netcoreapp1.0\publish to C:\Lambda\AWSLambda1\AWSLambda1\bin\Release\netcoreapp1.0\AWSLambda1.zip
Enter Function Name: (AWS Lambda function name)
DotNetCoreLambdaTest
Enter AWS Region: (The region to connect to AWS services)
us-west-2
Updating code for existing function
```

部署 Lambda 函數之後，即可開始使用。如需詳細資訊，請參閱[如何使用 AWS Lambda 的範例](#)。

Lambda 會自動為您監控 Lambda 函數，並透過 Amazon CloudWatch 報告指標。若要監控和疑難排解 Lambda 函數，請參閱[使用 Amazon CloudWatch 疑難排解和監控 AWS Lambda 函數](#)。

使用 AWS Toolkit for Visual Studio 搭配 Amazon Q 在 Visual Studio AWS Elastic Beanstalk 中部署至

AWS Elastic Beanstalk 是一種服務，可簡化為您的應用程式佈建 AWS 資源的程序。Elastic Beanstalk 提供部署應用程式所需的所有 AWS 基礎設施。此基礎設施包括：

- 託管應用程式可執行檔和內容的 Amazon EC2 執行個體。
- Auto Scaling 群組可維護適當數量的 Amazon EC2 執行個體，以支援您的應用程式。
- Elastic Load Balancing 負載平衡器，可將傳入流量路由至頻寬最高的 Amazon EC2 執行個體。

本使用者指南主題說明如何在 AWS Toolkit with Amazon Q 中使用 Elastic Beanstalk 精靈。如需 Elastic Beanstalk 特定的詳細資訊，請參閱[AWS Elastic Beanstalk 開發人員指南](#)。下列主題章節說明 AWS Toolkit with Amazon Q 的 Elastic Beanstalk 精靈。

主題

- [將傳統 ASP.NET 應用程式部署至 Elastic Beanstalk](#)
- [將 ASP.NET Core 應用程式部署至 Elastic Beanstalk \(舊版\)](#)
- [如何為您的應用程式指定 AWS 安全登入資料](#)
- [如何將應用程式重新發佈至 Elastic Beanstalk 環境 \(舊版\)](#)
- [自訂 Elastic Beanstalk 應用程式部署](#)
- [自訂 ASP.NET Core Elastic Beanstalk 部署](#)
- [適用於 .NET 和 Elastic Beanstalk 的多個應用程式支援](#)

將傳統 ASP.NET 應用程式部署至 Elastic Beanstalk

本節說明如何使用 Toolkit for Visual Studio 提供的 Publish to Elastic Beanstalk 精靈，透過 Elastic Beanstalk 部署應用程式。若要練習，您可以使用內建於 Visual Studio 的 Web 應用程式入門專案執行個體，也可以使用自己的專案。

Note

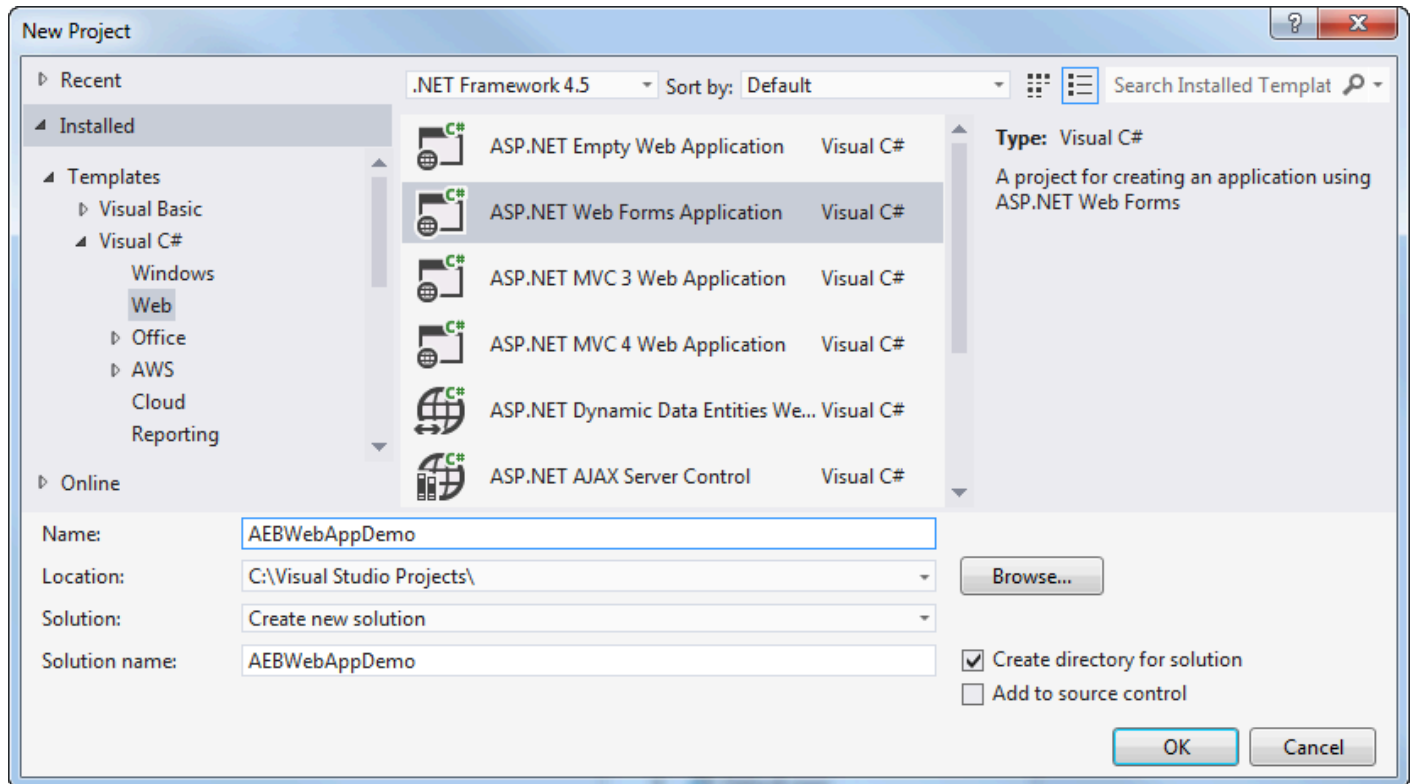
精靈也支援部署 ASP.NET Core 應用程式。如需 ASP.NET Core 的詳細資訊，請參閱 [AWS .NET 部署工具](#) 指南和更新的 [部署至 AWS](#) 目錄。

Note

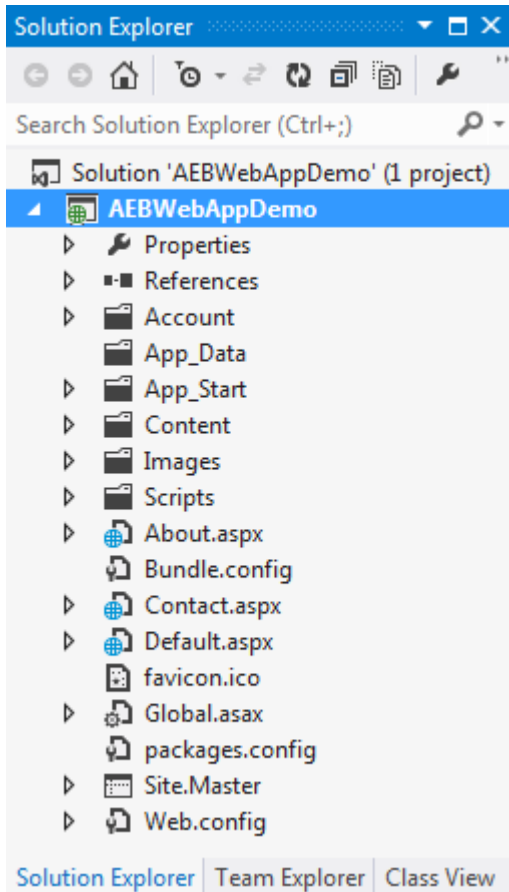
您必須先下載並安裝 [Web Deploy](#)，才能使用 Publish to Elastic Beanstalk 精靈。精靈依賴 Web Deploy 將 Web 應用程式和網站部署到網際網路資訊服務 (IIS) Web 伺服器。

建立範例 Web 應用程式啟動者專案

1. 在 Visual Studio 中，從檔案功能表中選擇新增，然後選擇專案。
2. 在 New Project (新增專案) 對話方塊的導覽窗格中，展開 Installed (已安裝)、展開 Templates (範本)，展開 Visual C#，然後選擇 Web。
3. 在 Web 專案範本的清單中，選擇其描述中包含文字 Web 和 Application 的任何範本。在此範例中，選擇 ASP.NET Web Forms 應用程式。

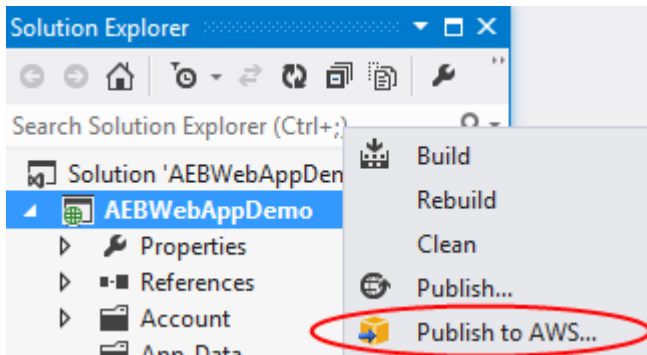


4. 在 Name (名稱) 方塊中，輸入 AEBWebAppDemo。
5. 在位置方塊中，輸入開發機器上解決方案資料夾的路徑，或選擇瀏覽，然後瀏覽並選擇解決方案資料夾，然後選擇選取資料夾。
6. 確認已選取 Create directory for solution (為方案建立目錄) 方塊。在解決方案下拉式清單中，確認已選取建立新解決方案，然後選擇確定。Visual Studio 將根據 ASP.NET Web Forms 應用程式專案範本建立解決方案和專案。然後，Visual Studio 會在新解決方案和專案出現的位置顯示 Solution Explorer。

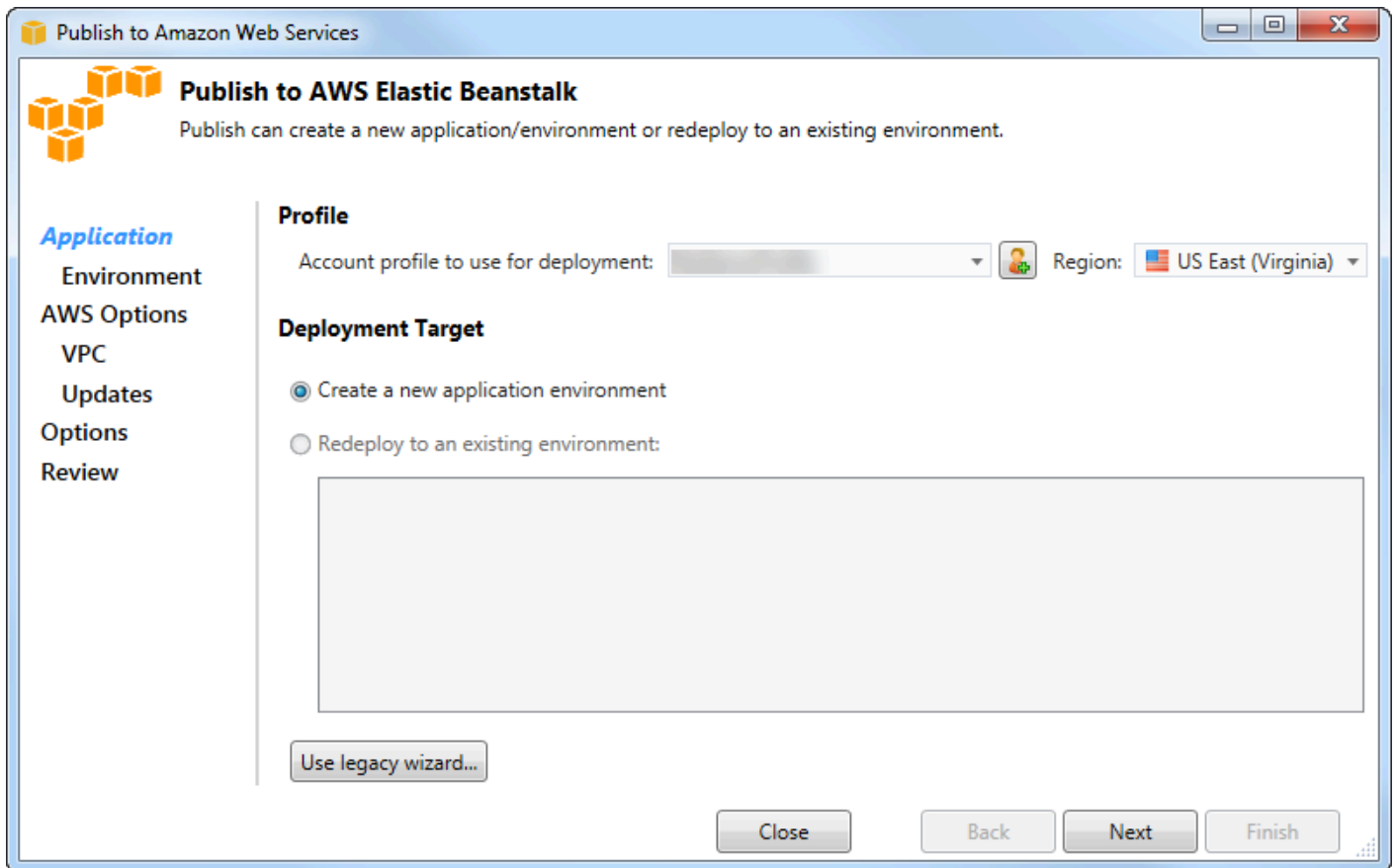


使用發佈至 Elastic Beanstalk 精靈部署應用程式

1. 在 Solution Explorer 中，開啟您在上一節中建立之專案的 AEBWebAppDemo 專案資料夾內容（按一下滑鼠右鍵）選單，或開啟您自己應用程式之專案資料夾的內容選單，然後選擇發佈至 AWS Elastic Beanstalk。



Publish to Elastic Beanstalk (發佈至 Elastic Beanstalk) 精靈隨即顯示。



2. 在設定檔中，從用於部署的帳戶設定檔下拉式清單中，選擇您要用於部署 AWS 的帳戶設定檔。

或者，如果您有要使用 AWS 的帳戶，但尚未為其建立 AWS 帳戶設定檔，您可以選擇具有加號 (+) 的按鈕來新增 AWS 帳戶設定檔。

3. 從區域下拉式清單中，選擇您希望 Elastic Beanstalk 部署應用程式的區域。

4. 在部署目標中，您可以選擇建立新的應用程式環境以執行應用程式的初始部署，或重新部署至現有環境以重新部署先前部署的應用程式。（先前的部署可能已使用精靈或已棄用的單機部署工具來執行。）如果您選擇重新部署到現有環境，精靈從目前執行的先前部署擷取資訊時，可能會有延遲。

Note

如果您選擇重新部署至現有環境，請在清單中選擇環境，然後選擇下一步，精靈會直接帶您前往應用程式選項頁面。如果您前往此路由，請跳到本節稍後說明如何使用應用程式選項頁面的說明。

5. 選擇下一步。

The screenshot shows a window titled "Publish to Amazon Web Services" with a sub-header "Application Environment". Below the sub-header is the instruction: "Enter the details for your new application environment. To create a new new environment for an existing application, select the appropriate application." On the left is a navigation menu with "Application Environment" selected. The main area contains three sections: "Application" with a dropdown menu showing "AEBWebAppDemo"; "Environment" with a dropdown menu; and "URL" with a text input field containing "http:" followed by a blurred subdomain and ".elasticbeanstalk.com", a "Check availability..." button, and a green checkmark message: "The requested URL is available". At the bottom are "Close", "Back", "Next", and "Finish" buttons.

6. 在應用程式環境頁面上，在應用程式區域中，名稱下拉式清單會提議應用程式的預設名稱。您可以從下拉式清單中選擇不同的名稱來變更預設名稱。
7. 在環境區域中的名稱下拉式清單中，輸入 Elastic Beanstalk 環境的名稱。在這種情況下，環境一詞是指您應用程式的基礎設施 Elastic Beanstalk 佈建。在此下拉式清單中可能已建議預設名稱。如果尚未提議預設名稱，您可以在有任何其他名稱可用時，輸入一個或從下拉式清單中選擇一個名稱。環境名稱不能超過 23 個字元。
8. 在 URL 區域中，方塊提議的預設子網域 `.elasticbeanstalk.com` 將是 Web 應用程式的 URL。您可以輸入新的子網域名稱來變更預設子網域。
9. 選擇檢查可用性，以確保您 Web 應用程式的 URL 尚未使用。

10. 如果 Web 應用程式的 URL 可以使用，請選擇下一步。

Publish to Amazon Web Services

AWS
Set Amazon EC2 and other AWS-related options for the deployed application.

Application
Environment
AWS Options
VPC
Updates
Options
Review

Amazon EC2 Launch Configuration

Container type *: 64bit Windows Server 2012 R2 running IIS 8.5

Instance type *: Micro Key pair *: MyKeyPair

Use custom AMI:

Use a VPC Single instance environment Enable Rolling Deployments

Deployed Application Permissions

Role: aws-elasticbeanstalk-ec2-role

The permissions for the Identity and Access Management role can be updated after the environment is created.

Relational Database Access

Select the Amazon RDS security groups to be modified to permit access from the EC2 instance(s) hosting your application.

default

Close Back Next Finish

1. 在AWS 選項頁面的 Amazon EC2 啟動組態中，從容器類型下拉式清單中選擇將用於應用程式的 Amazon Machine Image (AMI) 類型。
2. 在執行個體類型下拉式清單中，指定要使用的 Amazon EC2 執行個體類型。在此範例中，我們建議您使用 Micro。這將最大限度地減少與執行執行個體相關的成本。如需 Amazon EC2 成本的詳細資訊，請前往 [EC2 定價](#) 頁面。
3. 在金鑰對下拉式清單中，選擇要用於登入應用程式所用執行個體的 Amazon EC2 執行個體金鑰對。
4. 或者，在使用自訂 AMI 方塊中，您可以指定自訂 AMI，以覆寫容器類型下拉式清單中指定的 AMI。如需如何建立自訂 AMI 的詳細資訊，請參閱 [AWS Elastic Beanstalk 開發人員指南](#) 中的 [使用自訂 AMIs](#)，以及從 [Amazon EC2 執行個體建立 AMI](#)。
5. 或者，如果您想要在 VPC 中啟動執行個體，請選取使用 VPC 方塊。
6. 或者，如果您想要啟動單一 Amazon EC2 執行個體，然後將應用程式部署到該執行個體，請選取單一執行個體環境方塊。

如果您選取此方塊，Elastic Beanstalk 仍會建立 Auto Scaling 群組，但不會進行設定。如果您想要稍後設定 Auto Scaling 群組，您可以使用 AWS 管理主控台。

7. 或者，如果您想要控制應用程式部署到執行個體的條件，請選取啟用滾動部署方塊。只有在您尚未選取單一執行個體環境方塊時，才能選取此方塊。
 8. 如果您的應用程式使用 Amazon S3 和 DynamoDB 等 AWS 服務，提供登入資料的最佳方法是使用 IAM 角色。在部署的應用程式許可區域中，您可以選擇現有的 IAM 角色，或建立精靈用來啟動環境的 IAM 角色。使用的應用程式適用於 .NET 的 AWS SDK 會在向 AWS 服務提出請求時，自動使用此 IAM 角色提供的登入資料。
 9. 如果您的應用程式存取 Amazon RDS 資料庫，請在關聯式資料庫存取區域的下拉式清單中，選取精靈將更新的任何 Amazon RDS 安全群組旁的方塊，以便您的 Amazon EC2 執行個體可以存取該資料庫。
- 10 選擇下一步。
- 如果您選取使用 VPC，則會顯示 VPC 選項頁面。
 - 如果您選取了啟用滾動部署，但未選取使用 VPC，則滾動部署頁面將會顯示。請跳到本節稍後的說明，以說明如何使用滾動部署頁面。
 - 如果您未選取使用 VPC 或啟用滾動部署，則會顯示應用程式選項頁面。請跳到本節稍後的說明，以說明如何使用應用程式選項頁面。
- 11 如果您選取使用 VPC，請在 VPC 選項頁面上指定資訊，以在 VPC 中啟動您的應用程式。

Publish to Amazon Web Services

VPC Options

Set Amazon VPC options for the deployed application.

Application

Environment

AWS Options

VPC

Updates

Options

Review

VPC *: vpc-4e (10.0.0.0/16)

ELB Scheme *: Public Security Group *: test (sg-c1)

ELB Subnet *: subnet-c7 (10.0.2.0/24 - us-east-1a)

Instances Subnet *: subnet-45 (10.0.0.0/24 - us-east-1a)

To run AWS Elastic Beanstalk applications inside a VPC, you will need to configure at least the following:

- Create two subnets: one for your EC2 instances and one for your Elastic Load Balancer.
- Traffic must be able to be routed from your Elastic Load Balancer to your EC2 instances.
- Your EC2 instances must be able to connect to the Internet and AWS endpoints.

Elastic Load Balancer settings are not applicable to 'Single Instance' environment types.

For more information visit [AWS Elastic Beanstalk Developer Guide](#)

Close Back Next Finish

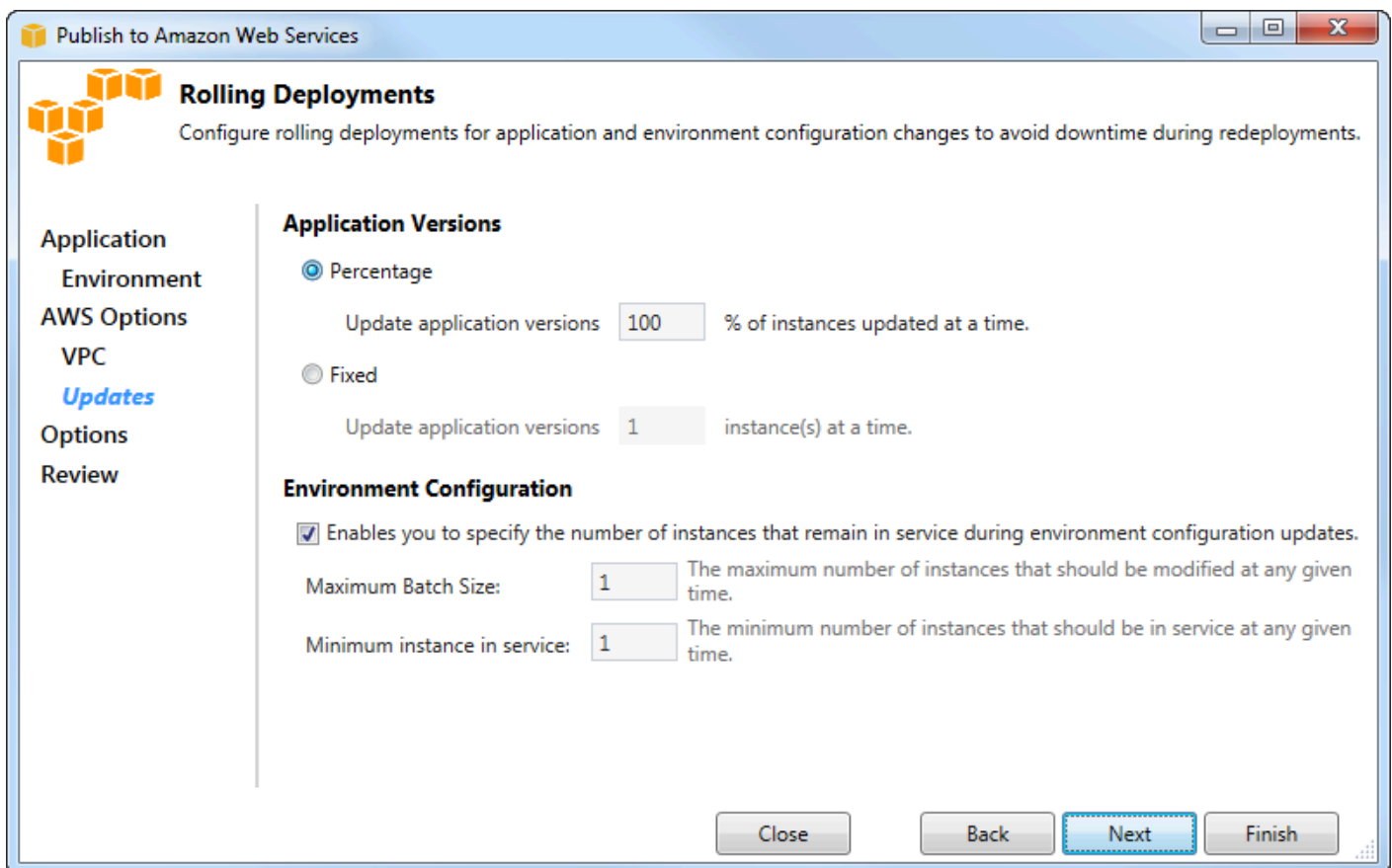
VPC 必須已建立。如果您在 Toolkit for Visual Studio 中建立 VPC，Toolkit for Visual Studio 會為您填入此頁面。如果您在 [AWS 管理主控台](#) 中建立 VPC，請在此頁面中輸入 VPC 的相關資訊。

部署至 VPC 的重要考量事項

- 您的 VPC 至少需要一個公有和一個私有子網路。
- 在 ELB 子網路下拉式清單中，指定公有子網路。Toolkit for Visual Studio 會將您應用程式的 Elastic Load Balancing 負載平衡器部署至公有子網路。公有子網路與具有指向網際網路閘道項目的路由表相關聯。您可以辨識網際網路閘道，因為它的 ID 開頭為 igw-（例如 igw-83cddaex）。您使用 Toolkit for Visual Studio 建立的公有子網路具有識別為公有的標籤值。
- 在執行個體子網路下拉式清單中，指定私有子網路。Toolkit for Visual Studio 會將您應用程式的 Amazon EC2 執行個體部署到私有子網路。
- 應用程式適用的 Amazon EC2 執行個體會透過公有子網路中執行網路位址轉譯 (NAT) 的 Amazon EC2 執行個體，從私有子網路通訊至網際網路。若要啟用此通訊，您需要一個 [VPC 安全群組](#)，允許流量從私有子網路流向 NAT 執行個體。在安全群組下拉式清單中指定此 VPC 安全群組。

如需如何將 Elastic Beanstalk 應用程式部署至 VPC 的詳細資訊，請參閱 [AWS Elastic Beanstalk 開發人員指南](#)。

1. 在 VPC 選項頁面上填入所有資訊後，選擇下一步。
 - 如果您選取啟用滾動部署，則滾動部署頁面將會出現。
 - 如果您未選取啟用滾動部署，則會顯示應用程式選項頁面。請跳到本節稍後的說明，以說明如何使用應用程式選項頁面。
2. 如果您選取啟用滾動部署，您可以在滾動部署頁面上指定資訊，以設定應用程式的新版本如何部署到負載平衡環境中的執行個體。例如，如果您的環境中有四個執行個體，而且您想要變更執行個體類型，您可以設定環境一次變更兩個執行個體。這有助於確保您的應用程式在進行變更時仍在執行。



3. 在應用程式版本區域中，選擇選項以控制一次部署到百分比或數量的執行個體。指定所需的百分比或數字。
4. 或者，如果您想要指定部署期間保留在服務的執行個體數量，請在環境組態區域中選取方塊。如果您選取此方塊，請指定一次應修改的執行個體數量上限、一次應保持服務的執行個體數量下限，或兩者。
5. 選擇下一步。

6. 在應用程式選項頁面上，您可以指定組建、網際網路資訊服務 (IIS) 和應用程式設定的相關資訊。

Build and IIS Deployment Settings

Project build configuration: Release

App pool: .NET Framework 4.5 Enable 32-bit applications

App path: Default Web Site/

Application Settings

Health check URL: /

Key	Value
-----	-------

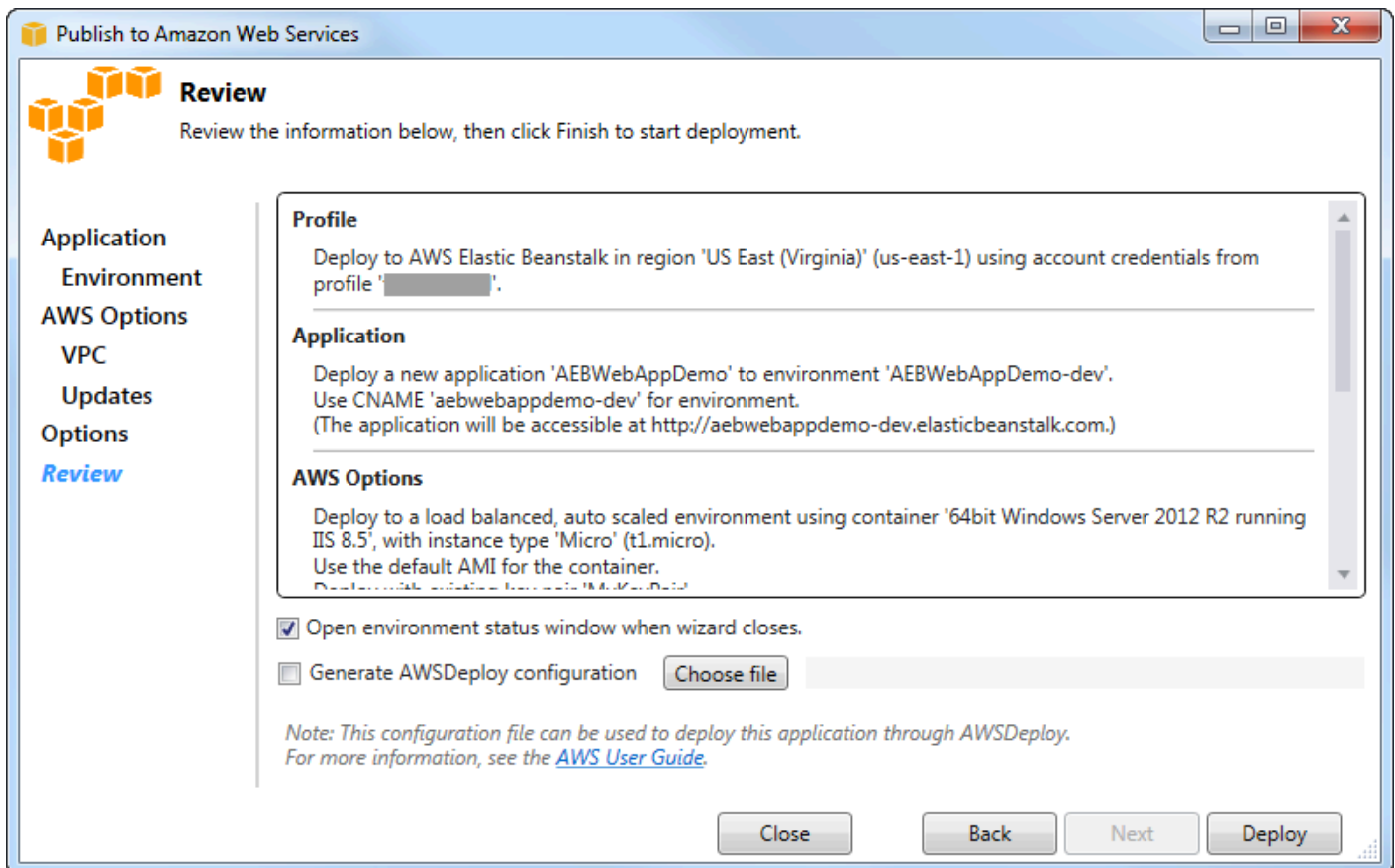
Close Back Next Finish

7. 在建置和 IIS 部署設定區域中，在專案建置組態下拉式清單中，選擇目標建置組態。如果精靈可以找到它，則版本會顯示為否則，作用中的組態會顯示在此方塊中。
8. 在應用程式集區下拉式清單中，選擇應用程式所需的 .NET Framework 版本。應該已顯示正確的 .NET Framework 版本。
9. 如果您的應用程式為 32 位元，請選取啟用 32 位元應用程式方塊。
10. 在應用程式路徑方塊中，指定 IIS 將用於部署應用程式的路徑。預設會指定預設網站/，通常會轉譯為路徑 c:\inetpub\wwwroot。如果您指定預設網站/以外的路徑，精靈會在指向您指定路徑的預設網站/路徑中放置重新導向。
11. 在應用程式設定區域中，於運作狀態檢查 URL 方塊中，輸入 Elastic Beanstalk 的 URL 以檢查 Web 應用程式是否仍然回應。此 URL 與根伺服器 URL 相關。預設會指定根伺服器 URL。例如，如果完整 URL 為 example.com/site-is-up.html，則輸入 /site-is-up.html。
12. 在索引鍵和值區域中，您可以指定要新增至應用程式 Web.config 檔案的任何索引鍵和值對。

Note

雖然不建議，但您可以使用金鑰和值的區域來指定應用程式應執行的 AWS 登入資料。偏好的方法是在 AWS 選項頁面的 Identity and Access Management Role 下拉式清單中指定 IAM 角色。不過，如果您必須使用 AWS 登入資料而非 IAM 角色來執行應用程式，請在金鑰列中選擇 AWSAccessKey。在值列中，輸入存取金鑰。針對 AWSSecretKey 重複這些步驟。

13 選擇下一步。



14. 在檢閱頁面上，檢閱您設定的選項，並在精靈關閉時選取開啟環境狀態視窗方塊。

15 如果各個項目都正確，請選擇 Deploy (部署)。

Note

當您部署應用程式時，作用中的帳戶會針對應用程式使用 AWS 的資源收取費用。

部署的相關資訊會顯示在 Visual Studio 狀態列和輸出視窗中。這可能需要幾分鐘的時間。部署完成時，輸出視窗中會顯示確認訊息。

16 若要刪除部署，請在 AWS Explorer 中展開 Elastic Beanstalk 節點，開啟部署子節點的內容（按一下滑鼠右鍵）選單，然後選擇刪除。刪除程序可能需要幾分鐘的時間。

將 ASP.NET Core 應用程式部署至 Elastic Beanstalk（舊版）

Important

本文件是指舊版服務和功能。如需更新的指南和內容，請參閱 [AWS .NET 部署工具](#) 指南和更新的 [部署至 AWS](#) 目錄。

AWS Elastic Beanstalk 是一種服務，可簡化為您的應用程式佈建 AWS 資源的程序。AWS Elastic Beanstalk 提供部署應用程式所需的所有 AWS 基礎設施。

Toolkit for Visual Studio 支援 AWS 使用 Elastic Beanstalk 將 ASP.NET Core 應用程式部署至。ASP.NET Core 是 ASP.NET 的重新設計，採用模組化架構，可將相依性開銷降至最低，並簡化您的應用程式以在雲端中執行。

AWS Elastic Beanstalk 可讓您輕鬆地以各種不同的語言部署應用程式 AWS。Elastic Beanstalk 同時支援傳統 ASP.NET 應用程式和 ASP.NET Core 應用程式。本主題說明部署 ASP.NET Core 應用程式。

使用 部署精靈

將 ASP.NET Core 應用程式部署到 Elastic Beanstalk 的最簡單方法是使用 Toolkit for Visual Studio。

如果您之前已使用 工具組來部署傳統 ASP.NET 應用程式，您會發現 ASP.NET Core 的體驗非常相似。在下列步驟中，我們將逐步解說部署體驗。

如果您之前從未使用過工具組，安裝工具組後需要做的第一件事就是向工具組註冊您的 AWS 登入資料。如需 [如何指定 Application for Visual Studio AWS 安全登入](#) 資料的詳細資訊，請參閱如何指定。

若要部署 ASP.NET Core Web 應用程式，請在解決方案總管中的專案上按一下滑鼠右鍵，然後選取發佈至 AWS...

在發佈至 AWS Elastic Beanstalk 部署精靈的第一頁上，選擇建立新的 Elastic Beanstalk 應用程式。Elastic Beanstalk「應用程式」為 Elastic Beanstalk 元件的邏輯集合，包括「環境」、「版本」和

「環境資訊」。部署精靈會產生應用程式，其中包含應用程式版本和環境的集合。環境包含執行應用程式版本的實際 AWS 資源。每次部署應用程式時，都會建立新的應用程式版本，精靈會將環境指向該版本。您可以在 [Elastic Beanstalk 元件中進一步了解這些概念](#)。

接著，設定應用程式的名稱及其第一個環境。每個環境都有與其相關聯的唯一 CNAME，您可以在部署完成時用來存取應用程式。

下一頁 AWS 的選項可讓您設定要使用 AWS 的資源類型。在此範例中，請保留預設值，金鑰對區段除外。金鑰對可讓您擷取 Windows 管理員密碼，以便登入機器。如果您尚未建立金鑰對，建議您選取建立新金鑰對。

許可

許可頁面用於將 AWS 登入資料指派給執行您應用程式的 EC2 執行個體。如果您的應用程式使用適用於 .NET 的 AWS SDK 存取其他 AWS 服務，這很重要。如果您未使用應用程式的任何其他服務，則可以保留此頁面的預設值。

應用程式選項

應用程式選項頁面上的詳細資訊與部署傳統 ASP.NET 應用程式時指定的詳細資訊不同。在這裡，您可以指定用於封裝應用程式的建置組態和架構，並指定應用程式的 IIS 資源路徑。

完成應用程式選項頁面後，請按一下下一步以檢閱設定，然後按一下部署以開始部署程序。

檢查環境狀態

在應用程式封裝並上傳至之後 AWS，您可以從 Visual Studio 中的 AWS Explorer 開啟環境狀態檢視，以檢查 Elastic Beanstalk 環境的狀態。

當環境上線時，事件會顯示在狀態列中。一旦一切完成，環境狀態將移至運作狀態良好。您可以按一下 URL 來檢視網站。您也可以從這裡將日誌從環境或遠端桌面提取到屬於 Elastic Beanstalk 環境一部分的 Amazon EC2 執行個體。

任何應用程式第一次部署需要比後續重新部署更長的時間，因為它會建立新的 AWS 資源。當您在開發期間迭代應用程式時，您可以返回精靈，或在按右鍵專案時選取重新發佈選項，以快速重新部署。

透過部署精靈，使用先前執行的設定重新發佈應用程式套件，並將應用程式套件上傳至現有的 Elastic Beanstalk 環境。

如何為您的應用程式指定 AWS 安全登入資料

您在發佈至 Elastic Beanstalk 精靈中指定的 AWS 帳戶是精靈將用來部署至 Elastic Beanstalk AWS 的帳戶。

雖然不建議，但您可能還需要指定應用程式在部署服務之後用來存取 AWS 服務 AWS 的帳戶登入資料。偏好的方法是指定 IAM 角色。在發佈至 Elastic Beanstalk 精靈中，您可以透過 AWS 選項頁面上的 Identity and Access Management Role 下拉式清單來執行此操作。在舊版發佈至 Amazon Web Services 精靈中，您可以透過 AWS 選項頁面上的 IAM 角色下拉式清單來執行此操作。

如果您必須使用 AWS 帳戶登入資料而非 IAM 角色，您可以使用下列其中一種方式為您的應用程式指定 AWS 帳戶登入資料：

- 參考對應至專案 Web.config 檔案 appSettings 元素中 AWS 帳戶登入資料的設定檔。(若要建立設定檔，請參閱 [設定 AWS 登入資料](#)。) 下列範例會指定設定檔名稱為 myProfile 的登入資料。

```
<appSettings>
  <!-- AWS CREDENTIALS -->
  <add key="AWSProfileName" value="myProfile"/>
</appSettings>
```

- 如果您使用的是發佈至 Elastic Beanstalk 精靈，請在應用程式選項頁面的索引鍵和值區域的索引鍵列中，選擇 AWS AccessKey。在值列中，輸入存取金鑰。針對 AWS SecretKey 重複這些步驟。
- 如果您使用舊版發佈至 Amazon Web Services 精靈，請在應用程式選項頁面的應用程式登入資料區域中，選擇使用這些登入資料，然後在存取金鑰和私密金鑰方塊中輸入存取金鑰和私密存取金鑰。

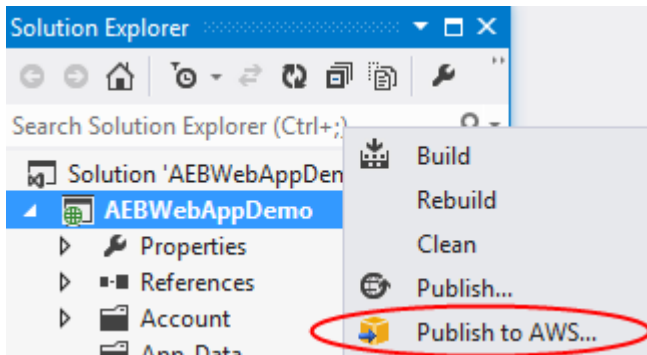
如何將應用程式重新發佈至 Elastic Beanstalk 環境 (舊版)

Important

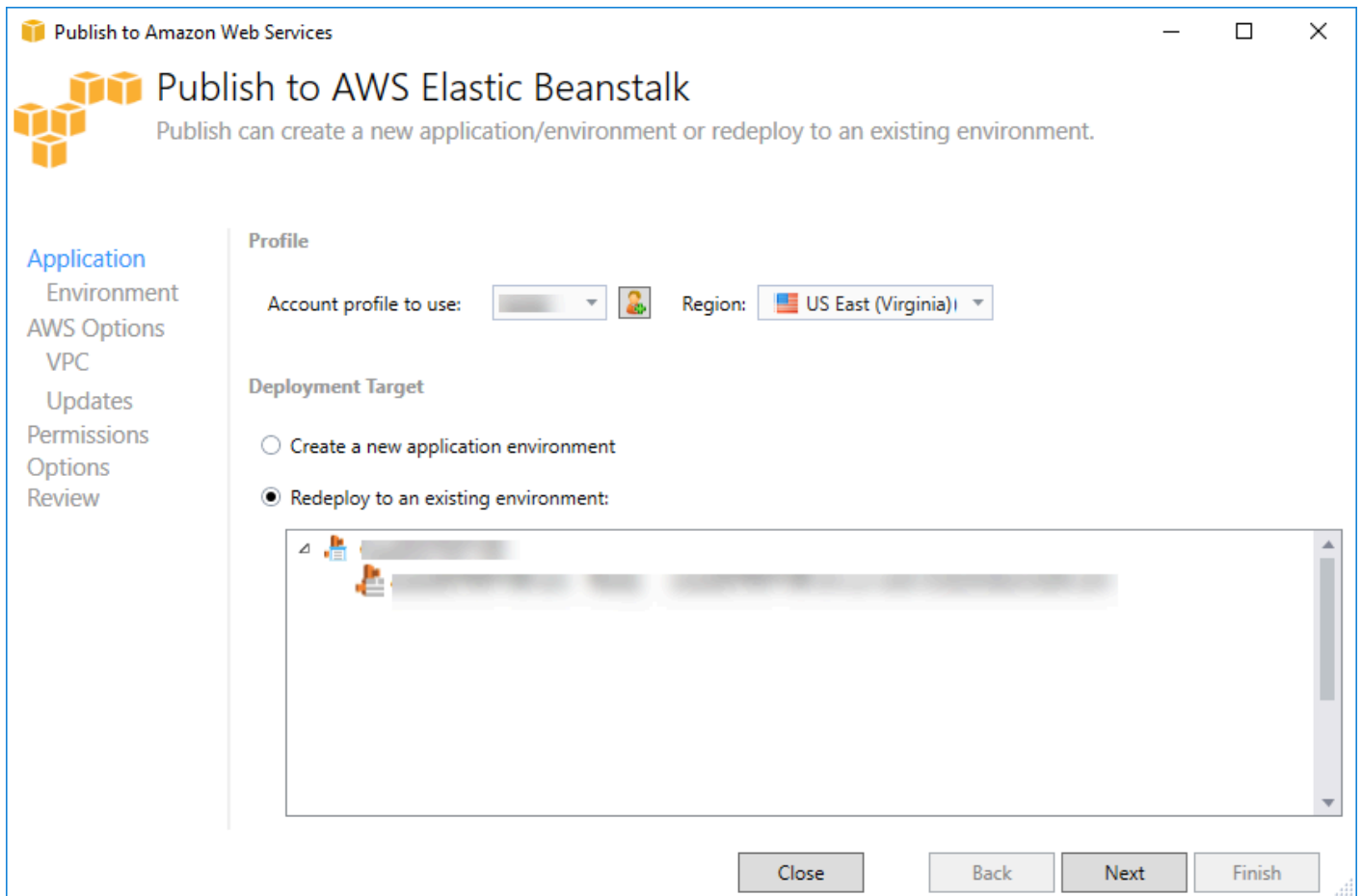
本文件是指舊版服務和功能。如需更新指南和內容，請參閱 [AWS .NET 部署工具指南](#)。

您可以透過進行離散變更，然後將新版本重新發佈到已啟動的 Elastic Beanstalk 環境，在應用程式上反覆執行。

1. 在 Solution Explorer 中，開啟您在上一節中發佈之專案的 AEBWebAppDemo 專案資料夾內容 (按一下滑鼠右鍵) 選單，然後選擇發佈至 AWS Elastic Beanstalk。



Publish to Elastic Beanstalk (發佈至 Elastic Beanstalk) 精靈隨即顯示。



2. 選取重新部署至現有環境，然後選擇您先前發佈的環境。按一下 Next (下一步)。

檢閱精靈隨即出現。

Publish to Amazon Web Services

Review

Review the information below, then click Finish to start deployment.

- Application
- Environment
- AWS Options
- VPC
- Updates
- Permissions
- Options
- Review

Profile
Publish to AWS Elastic Beanstalk in region 'US East (Virginia)' (us-east-1) using account credentials from profile '...'.

Application
Redeploy to environment '...' for application '...'.

Application Options
Use project configuration 'Debug|Any CPU' when building for deployment.
Deploy as application version 'v20170824172255'
Deploy a web application supporting .NET Core Framework netcoreapp1.1 with path 'Default Web Site/'.

Open environment status window when wizard closes.

Generate AWSDeploy configuration

Note: This configuration file can be used to deploy this application through AWSDeploy.
For more information, see the [AWS User Guide](#).

3. 按一下部署。應用程式將重新部署到相同的環境。

如果您的應用程式正在啟動或終止，則無法重新發佈。

自訂 Elastic Beanstalk 應用程式部署

本主題說明 Elastic Beanstalk Microsoft Windows 容器的部署資訊清單如何支援自訂應用程式部署。

對於想要利用 Elastic Beanstalk 的強大功能來建立和管理 AWS 資源，但想要完全控制應用程式部署方式的進階使用者，自訂應用程式部署是一項強大的功能。對於自訂應用程式部署，您可以為 Elastic Beanstalk 執行的三個不同動作建立 Windows PowerShell 指令碼。啟動部署時會使用安裝動作，從工具組或 Web 主控台呼叫 RestartAppServer API 時會使用重新啟動，並在發生新部署時解除安裝在先前的任何部署上叫用。

例如，當您的文件團隊撰寫了他們想要包含在部署中的靜態網站時，您可能有一個您想要部署的 ASP.NET 應用程式。您可以撰寫部署資訊清單來執行此操作，如下所示：

```
{
```

```
"manifestVersion": 1,
"deployments": {

  "msDeploy": [
    {
      "name": "app",
      "parameters": {
        "appBundle": "CoolApp.zip",
        "iisPath": "/"
      }
    }
  ],
  "custom": [
    {
      "name": "PowerShellDocs",
      "scripts": {
        "install": {
          "file": "install.ps1"
        },
        "restart": {
          "file": "restart.ps1"
        },
        "uninstall": {
          "file": "uninstall.ps1"
        }
      }
    }
  ]
}
}
```

為每個動作列出的指令碼必須位於相對於部署資訊清單檔案的應用程式套件中。在此範例中，應用程式套件也會包含 `documentation.zip` 檔案，其中包含由您的文件團隊建立的靜態網站。

`install.ps1` 指令碼會擷取 `zip` 檔案並設定 IIS 路徑。

```
Add-Type -assembly "system.io.compression.filesystem"
[io.compression.zipfile]::ExtractToDirectory('./documentation.zip', 'c:\inetpub\wwwroot\documentation')

powershell.exe -Command {New-WebApplication -Name documentation -PhysicalPath c:\inetpub\wwwroot\documentation -Force}
```

由於您的應用程式正在 IIS 中執行，重新啟動動作會叫用 IIS 重設。

```
iisreset /timeout:1
```

對於解除安裝指令碼，請務必清除安裝階段期間使用的所有設定和檔案。如此一來，在新版本的安裝階段，您就可以避免與先前的部署發生衝突。在此範例中，您需要移除靜態網站的 IIS 應用程式，並移除網站檔案。

```
powershell.exe -Command {Remove-WebApplication -Name documentation}  
Remove-Item -Recurse -Force 'c:\inetpub\wwwroot\documentation'
```

透過這些指令碼檔案和應用程式套件中包含的 documentation.zip 檔案，部署會建立 ASP.NET 應用程式，然後部署文件網站。

在此範例中，我們選擇一個簡單的範例來部署簡單的靜態網站，但使用自訂應用程式部署，您可以部署任何類型的應用程式，並讓 Elastic Beanstalk 管理其 AWS 資源。

自訂 ASP.NET Core Elastic Beanstalk 部署

本主題說明部署的運作方式，以及使用 Elastic Beanstalk 和 Toolkit for Visual Studio 建立 ASP.NET Core 應用程式時，您可以執行哪些自訂部署。

在您完成 Toolkit for Visual Studio 中的部署精靈後，工具組會封裝應用程式，並將其傳送至 Elastic Beanstalk。建立應用程式套件的第一步是使用新的 dotnet CLI，以使用發佈命令準備應用程式進行發佈。架構和組態會從精靈中的設定傳遞至發佈命令。因此，如果您為選擇了發行版本 configuration，並為選擇了 netcoreapp1.0framework，則工具組將執行下列命令：

```
dotnet publish --configuration Release --framework netcoreapp1.0
```

當發佈命令完成時，工具組會將新的部署資訊清單寫入發佈資料夾。部署資訊清單是名為 aws-windows-deployment-manifest.json 的 JSON 檔案，Elastic Beanstalk Windows 容器 (1.2 版或更新版本) 會讀取此檔案以判斷如何部署應用程式。例如，對於您想要在 IIS 根目錄部署的 ASP.NET Core 應用程式，工具組會產生如下所示的資訊清單檔案：

```
{  
  "manifestVersion": 1,  
  "deployments": {  
  
    "aspNetCoreWeb": [  

```

```
{
  "name": "app",
  "parameters": {
    "appBundle": ".",
    "iisPath": "/",
    "iisWebSite": "Default Web Site"
  }
}
]
```

`appBundle` 屬性指出應用程式位元與資訊清單檔案的關聯。此屬性可以指向目錄或 ZIP 封存。`iisPath` 和 `iisWebSite` 屬性指出 IIS 中託管應用程式的位置。

自訂資訊清單

工具組只會在發佈資料夾中不存在資訊清單檔案時寫入資訊清單檔案。如果檔案確實存在，工具組會更新資訊清單 `aspNetCoreWeb` 區段下列出之第一個應用程式中的 `appBundle`、`iisPath` 和 `iisWebSite` 屬性。這可讓您將 `aws-windows-deployment-manifest.json` 新增至您的專案，並自訂資訊清單。若要在 Visual Studio 中為 ASP.NET Core Web 應用程式執行此操作，請將新的 JSON 檔案新增至專案的根目錄，並命名為 `aws-windows-deployment-manifest.json`。

資訊清單必須命名為 `aws-windows-deployment-manifest.json`，且必須位於專案的根目錄。Elastic Beanstalk 容器會在根目錄中尋找資訊清單，如果找到資訊清單，則會叫用部署工具。如果檔案不存在，Elastic Beanstalk 容器會回到較舊的部署工具，其假設封存是 `ms` 部署封存。

為確保 `dotnet CLI publish` 命令包含資訊清單，請更新 `project.json` 檔案，以在 `include` 的包含區段中包含資訊清單檔案 `publishOptions`。

```
{
  "publishOptions": {
    "include": [
      "wwwroot",
      "Views",
      "Areas/**/Views",
      "appsettings.json",
      "web.config",
      "aws-windows-deployment-manifest.json"
    ]
  }
}
```

```
}
```

現在您已宣告資訊清單，使其包含在應用程式套件中，您可以進一步設定部署應用程式的方式。您可以自訂部署，超出部署精靈支援的範圍。AWS 已為 `aws-windows-deployment-manifest.json` 檔案定義了 JSON 結構描述，而且當您安裝 Toolkit for Visual Studio 時，安裝程式會註冊結構描述的 URL。

當您開啟 `aws-windows-deployment-manifest.json`，您會在結構描述下拉式方塊中看到選取的結構描述 URL。您可以導覽至 URL，以取得資訊清單中可設定內容的完整描述。選取結構描述後，Visual Studio 會在您編輯資訊清單時提供 IntelliSense。

您可以執行的其中一個自訂是設定應用程式將在其中執行的 IIS 應用程式集區。下列範例示範如何定義 IIS 應用程式集區 ("customPool")，該集區每 60 分鐘回收一次程序，並使用將其指派給應用程式 "appPool": "customPool"。

```
{
  "manifestVersion": 1,
  "iisConfig": {
    "appPools": [
      {
        "name": "customPool",
        "recycling": {
          "regularTimeInterval": 60
        }
      }
    ]
  },
  "deployments": {
    "aspNetCoreWeb": [
      {
        "name": "app",
        "parameters": {
          "appPool": "customPool"
        }
      }
    ]
  }
}
```

此外，資訊清單可以宣告 Windows PowerShell 指令碼在安裝、重新啟動和解除安裝動作之前和之後執行。例如，下列資訊清單會執行 Windows PowerShell 指令碼 `PostInstallSetup.ps1`，以在 ASP.NET Core 應用程式部署至 IIS 之後執行進一步的設定工作。新增這類指令碼時，請確定指

令碼已新增至 `project.json` 檔案中 `publishOptions` 下的包含區段，就像您對 `aws-windows-deployment-manifest.json` 檔案所做的一樣。如果沒有，指令碼不會包含在 `dotnet CLI` 發佈命令中。

```
{
  "manifestVersion": 1,
  "deployments": {
    "aspNetCoreWeb": [
      {
        "name": "app",
        "scripts": {
          "postInstall": {
            "file": "SetupScripts/PostInstallSetup.ps1"
          }
        }
      }
    ]
  }
}
```

.ebextensions 如何？

支援 Elastic Beanstalk `.ebextensions` 組態檔案，如同所有其他 Elastic Beanstalk 容器一樣。若要將 `.ebextensions` 包含在 ASP.NET Core 應用程式中，請將 `.ebextensions` 目錄新增至 `project.json` 檔案 `publishOptions` 的 `include` 區段。如需 `.ebextensions` 的詳細資訊，請參閱 [Elastic Beanstalk 開發人員指南](#)。

適用於 .NET 和 Elastic Beanstalk 的多個應用程式支援

您可以使用部署資訊清單，將多個應用程式部署至相同的 Elastic Beanstalk 環境。

部署資訊清單支援 [ASP.NET Core](#) Web 應用程式，以及 `ms` 部署傳統 ASP.NET 應用程式的封存。試想一個案例，其中您已使用前端的 ASP.NET Core 和延伸 API 的 Web API 專案來撰寫令人驚豔的新應用程式。您也有使用傳統 ASP.NET 撰寫的管理員應用程式。

工具組的部署精靈著重於部署單一專案。若要利用多個應用程式部署，您必須手動建構應用程式套件。若要開始，請撰寫資訊清單。在此範例中，您會在解決方案的根目錄撰寫資訊清單。

資訊清單中的部署區段有兩個子系：要部署的 ASP.NET Core Web 應用程式陣列，以及要部署的 `ms` 部署封存陣列。對於每個應用程式，您可以設定 IIS 路徑和應用程式位元相對於資訊清單的位置。

```
{
  "manifestVersion": 1,
  "deployments": {

    "aspNetCoreWeb": [
      {
        "name": "frontend",
        "parameters": {
          "appBundle": "./frontend",
          "iisPath": "/frontend"
        }
      },
      {
        "name": "ext-api",
        "parameters": {
          "appBundle": "./ext-api",
          "iisPath": "/ext-api"
        }
      }
    ],
    "msDeploy": [
      {
        "name": "admin",
        "parameters": {
          "appBundle": "AmazingAdmin.zip",
          "iisPath": "/admin"
        }
      }
    ]
  }
}
```

編寫資訊清單後，您將使用 Windows PowerShell 建立應用程式套件，並更新現有的 Elastic Beanstalk 環境來執行它。指令碼的撰寫假設將從包含 Visual Studio 解決方案的資料夾執行。

您在指令碼中需要做的第一件事是設定工作區資料夾，在其中建立應用程式套件。

```
$publishFolder = "c:\temp\publish"

$publishWorkspace = [System.IO.Path]::Combine($publishFolder, "workspace")
$appBundle = [System.IO.Path]::Combine($publishFolder, "app-bundle.zip")

If (Test-Path $publishWorkspace){
```

```
Remove-Item $publishWorkspace -Confirm:$false -Force
}
If (Test-Path $appBundle){
  Remove-Item $appBundle -Confirm:$false -Force
}
```

建立資料夾之後，就該準備好前端了。如同部署精靈，使用 dotnet CLI 發佈應用程式。

```
Write-Host 'Publish the ASP.NET Core frontend'
$publishFrontendFolder = [System.IO.Path]::Combine($publishWorkspace, "frontend")
dotnet publish .\src\AmazingFrontend\project.json -o $publishFrontendFolder -c Release
-f netcoreapp1.0
```

請注意，輸出資料夾使用了子資料夾「前端」，符合您在資訊清單中設定的資料夾。現在，您需要對 Web API 專案執行相同的操作。

```
Write-Host 'Publish the ASP.NET Core extensibility API'
$publishExtAPIFolder = [System.IO.Path]::Combine($publishWorkspace, "ext-api")
dotnet publish .\src\AmazingExtensibleAPI\project.json -o $publishExtAPIFolder -c
Release -f netcoreapp1.0
```

管理網站是傳統的 ASP.NET 應用程式，因此您無法使用 dotnet CLI。對於管理員應用程式，您應該使用 msbuild，傳入建置目標套件以建立 msdeploy 封存。根據預設，套件目標會在 obj\Release\Package 資料夾下建立 msdeploy 封存，因此您需要將封存複製到發佈工作區。

```
Write-Host 'Create msdeploy archive for admin site'
msbuild .\src\AmazingAdmin\AmazingAdmin.csproj /t:package /p:Configuration=Release
Copy-Item .\src\AmazingAdmin\obj\Release\Package\AmazingAdmin.zip $publishWorkspace
```

若要告知 Elastic Beanstalk 環境如何處理所有這些應用程式，請將資訊清單從您的解決方案複製到發佈工作區，然後壓縮資料夾。

```
Write-Host 'Copy deployment manifest'
Copy-Item .\aws-windows-deployment-manifest.json $publishWorkspace

Write-Host 'Zipping up publish workspace to create app bundle'
Add-Type -assembly "system.io.compression.filesystem"
[io.compression.zipfile]::CreateFromDirectory( $publishWorkspace, $appBundle)
```

現在您已擁有應用程式套件，您可以前往 Web 主控台，並將封存上傳至 Elastic Beanstalk 環境。或者，您可以繼續使用 AWS PowerShell cmdlet 來更新 Elastic Beanstalk 環境與應用程式套件。請確定

您已使用 `Set-DefaultAWSRegion` cmdlet，將目前的設定檔和區域設定為包含 Elastic Beanstalk 環境的設定檔 `Set-AWSCredentials` 和區域。

```
Write-Host 'Write application bundle to S3'
# Determine S3 bucket to store application bundle
$s3Bucket = New-EBStorageLocation
Write-S3Object -BucketName $s3Bucket -File $appBundle

$applicationName = "ASPNETCoreOnAWS"
$environmentName = "ASPNETCoreOnAWS-dev"
$versionLabel = [System.DateTime]::Now.Ticks.ToString()

Write-Host 'Update Beanstalk environment for new application bundle'
New-EBApplicationVersion -ApplicationName $applicationName -VersionLabel $versionLabel
  -SourceBundle_S3Bucket $s3Bucket -SourceBundle_S3Key app-bundle.zip
Update-EBEnvironment -ApplicationName $applicationName -EnvironmentName
  $environmentName -VersionLabel $versionLabel
```

現在，使用工具組或 Web 主控台內的 Elastic Beanstalk 環境狀態頁面來檢查更新的狀態。完成後，您將能夠導覽至部署資訊清單中 IIS 路徑集所部署的每個應用程式。

部署至 Amazon EC2 Container Service

Important

新的 發佈至 AWS 功能旨在簡化您發佈 .NET 應用程式的方式 AWS。在選擇發佈容器 AWS 之後，系統可能會詢問您是否要切換到此發佈體驗。如需詳細資訊，請參閱 [在 Visual Studio AWS 中使用 發佈至](#)。

Amazon Elastic Container Service 是高度可擴展的高效能容器管理服務，支援 Docker 容器，並可讓您輕鬆地在 Amazon EC2 執行個體的受管叢集上執行應用程式。

若要在 Amazon Elastic Container Service 上部署應用程式，您的應用程式元件必須開發為在 Docker 容器中執行。Docker 容器是軟體開發的標準化單元，包含軟體應用程式需要執行的所有一切：程式碼、執行時間、系統工具和系統程式庫等等。

Toolkit for Visual Studio 提供精靈，可簡化透過 Amazon ECS 發佈應用程式。以下各節會說明此精靈。

如需 Amazon ECS 的詳細資訊，請參閱 [Elastic Container Service 文件](#)。它包含 [Docker 基本概念和建立叢集的概觀](#)。

主題

- [為您的 ASP.NET Core 2 應用程式指定 AWS 登入資料](#)
- [將 ASP.NET Core 2.0 應用程式部署至 Amazon ECS \(Fargate\) \(舊版\)](#)
- [將 ASP.NET Core 2.0 應用程式部署至 Amazon ECS \(EC2\)](#)

為您的 ASP.NET Core 2 應用程式指定 AWS 登入資料

當您將應用程式部署到 Docker 容器時，有兩種類型的登入資料：部署登入資料和執行個體登入資料。

部署登入資料由發佈容器用於 AWS 精靈，以在 Amazon ECS 中建立環境。這包括任務、服務、IAM 角色、Docker 容器儲存庫，以及負載平衡器等項目。

執行個體（包括您的應用程式）會使用執行個體登入資料來存取不同的 AWS 服務。例如，如果您的 ASP.NET Core 2.0 應用程式讀取和寫入 Amazon S3 物件，則需要適當的許可。您可以根據環境使用不同的方法來提供不同的登入資料。例如，您的 ASP.NET Core 2 應用程式可能以開發和生產環境為目標。您可以使用本機 Docker 執行個體和登入資料進行開發，以及在生產環境中定義的角色。

指定部署登入資料

您在將容器發佈至 AWS 精靈中指定的 AWS 帳戶是精靈將用來部署至 Amazon ECS AWS 的帳戶。帳戶設定檔必須具有 Amazon Elastic Compute Cloud、Amazon Elastic Container Service 和 的許可 AWS Identity and Access Management。

如果您注意到下拉式清單中缺少選項，可能是因為您缺乏許可。例如，如果您為應用程式建立叢集，但未在將容器發佈至 AWS 精靈叢集頁面上看到叢集。如果發生這種情況，請新增缺少的許可，然後再試一次精靈。

指定開發執行個體登入資料

對於非生產環境，您可以在 `appsettings.<environment>.json` 檔案中設定您的登入資料。例如，若要在 Visual Studio 2017 的 `appsettings.Development.json` 檔案中設定您的登入資料：

1. 將 `AWSSDK.Extensions.NETCore.Setup NuGet` 套件新增至您的專案。
2. 將 AWS 設定新增至 `appsettings.Development.json`。以下組態會設定 Profile 和 Region。

```
{
```

```
"AWS": {
  "Profile": "local-test-profile",
  "Region": "us-west-2"
}
```

指定生產執行個體登入資料

對於生產執行個體，我們建議您使用 IAM 角色來控制應用程式（和服務）可存取的內容。例如，若要將具有 Amazon ECS 的 IAM 角色設定為具有 Amazon Simple Storage Service 和 Amazon DynamoDB 許可的服務主體 AWS 管理主控台：

1. 登入 AWS 管理主控台，並在 <https://console.aws.amazon.com/iam/> 開啟 IAM 主控台。
2. 在 IAM 主控台的導覽窗格中，選擇角色，然後選擇建立角色。
3. 選擇 AWS 服務角色類型，然後選擇 EC2 Container Service。
4. 選擇 EC2 Container Service 任務使用案例。服務會定義使用案例，以包含服務所需的信任政策。然後選擇 Next: Permissions (下一步：許可)。
5. 選擇 AmazonS3FullAccess 和 AmazonDynamoDBFullAccess 許可政策。勾選每個政策旁的方塊，然後選擇下一步：檢閱、
6. 針對角色名稱，輸入角色名稱或角色名稱尾碼，以協助您識別此角色的目的。角色名稱在您的 AWS 帳戶內必須是獨一無二的。它們無法透過大小寫進行區分。例如，您無法建立名為 PRODRole 和 prodrole 的角色。因為有各種實體可能會參照角色，所以您無法在建立角色之後編輯角色名稱。
7. (選用) 針對 Role description (角色說明)，輸入新角色的說明。
8. 檢閱角色，然後選擇建立角色。

您可以在將容器發佈至 AWS 精靈的 ECS 任務定義頁面上，使用此角色做為任務角色。

如需詳細資訊，請參閱 [使用服務型角色](#)。

將 ASP.NET Core 2.0 應用程式部署至 Amazon ECS (Fargate) (舊版)

Important

本文件是指舊版服務和功能。如需更新的指南和內容，請參閱 [AWS .NET 部署工具](#) 指南和更新的 [部署至 AWS](#) 目錄。

本節說明如何使用 Toolkit for Visual Studio 提供的發佈容器精靈 AWS，使用 Fargate 啟動類型透過 Amazon ECS 部署以 Linux 為目標的容器化 ASP.NET Core 2.0 應用程式。由於 Web 應用程式目標是要持續執行，它將部署為服務。

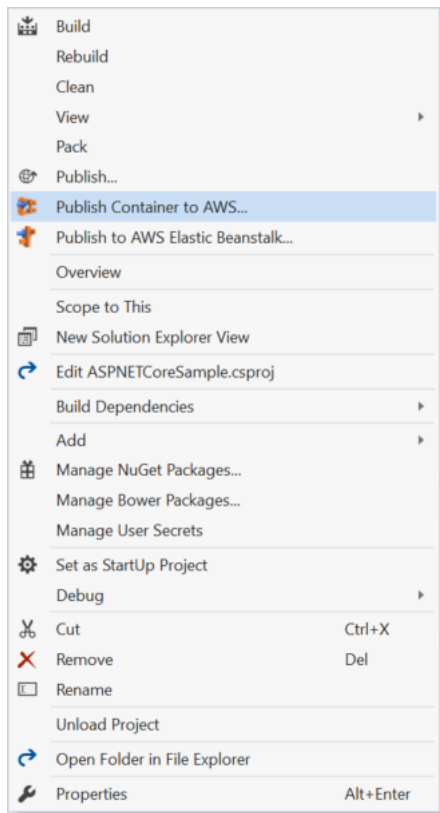
在您發佈容器之前

使用發佈容器精靈 AWS 部署 ASP.NET Core 2.0 應用程式之前：

- [指定您的 AWS 登入資料](#)，並使用 [Amazon ECS 進行設定](#)。
- [安裝 Docker](#)。您有一些不同的安裝選項，包括 [Docker for Windows](#)。
- 在 Visual Studio 中，為以 Linux 為目標的 ASP.NET Core 2.0 容器化應用程式建立（或開啟）專案。

存取要 AWS 精靈的發佈容器

若要部署以 Linux 為目標的 ASP.NET Core 2.0 容器化應用程式，請在 Solution Explorer 中的專案上按一下滑鼠右鍵，然後選取將容器發佈至 AWS。



您也可以可以在 Visual Studio Build 功能表中選取將容器發佈至 AWS。

將容器發佈至 AWS 精靈

Publish Container to AWS

aws Publish Container to AWS
Select the Amazon ECR Repository to push the Docker image to.

Profile

Account profile to use: vstools Region: US East (Virginia)

Docker Image Build

Configuration: Release

Docker Repository: aspnetcoresample Tag: latest

Deployment Target

Service on an ECS Cluster
Deploy the application as a service on an Amazon Elastic Container Service Cluster. A service is for applications like Web applications that are intended to run indefinitely.

Save settings to aws-ecs-tools-defaults.json and configure project for command line deployment.
If this is checked the dotnet CLI tool package Amazon.ECS.Tools will be added to the project. Once added you can do future deployments from the command line. Run the command "dotnet ecs --help" for more information.

Close Back Next Publish

要使用的帳戶設定檔 - 選取要使用的帳戶設定檔。

區域 - 選擇部署區域。設定檔和區域用於設定部署環境資源，以及選取預設 Docker 登錄檔。

組態 - 選取 Docker 映像建置組態。

Docker 儲存庫 - 選擇現有的 Docker 儲存庫，或輸入新儲存庫的名稱，即可建立該儲存庫。這是要推送建置容器的儲存庫。

標籤 - 選取現有標籤或輸入新標籤的名稱。標籤可以追蹤 Docker 容器的版本、選項或其他唯一組態元素等重要詳細資訊。

部署目標 - 選取 ECS 叢集上的服務。當您的應用程式需要長時間執行時（例如 ASP.NET Web 應用程式），請使用此部署選項。

將設定儲存至 **aws-docker-tools-defaults.json** 並設定專案以進行命令列部署 - 如果您想要從命令列部署的彈性，請勾選此選項。dotnet ecs deploy 從您的專案目錄使用來部署和 dotnet ecs publish 容器。

啟動組態頁面

Publish Container to AWS

aws Launch Configuration
Choose how to provide compute capacity to your application.

ECS Cluster: Create an empty cluster ASPNETCoreSample

This wizard supports creating an empty cluster which is suitable for running Fargate based services and tasks. It will not have any EC2 instances registered to it so services and tasks with the EC2 launch type will not run. The easiest way to create a cluster with EC2 instances registered is to use the AWS web console.

Launch Type: FARGATE

FARGATE will automatically provision the necessary compute capacity needed to run the application based on the CPU and Memory settings. This removes the need to add any EC2 instances to your cluster.

Allocated Compute Capacity

CPU Maximum (vCPU): 0.25 vCPU (256) Memory Maximum (GB): 512MB

Network Configuration

VPC Subnets: Security Groups:

Assign Public IP Address

Close Back Next Publish

ECS 叢集 - 選擇將執行 Docker 映像的叢集。如果您選擇建立空叢集，請提供新叢集的名稱。

啟動類型 - 選擇 FARGATE。

CPU 上限 (vCPU) - 選擇應用程式所需的運算容量上限。若要查看允許的 CPU 和記憶體值範圍，請參閱[任務大小](#)。

記憶體上限 (GB) - 選取應用程式可用的記憶體數量上限。

VPC 子網路 - 選擇單一 VPC 下的一或多個子網路。如果您選擇多個子網路，您的任務會分散到各個子網路。這可以改善可用性。如需詳細資訊，請參閱[預設 VPC 和預設子網路](#)。

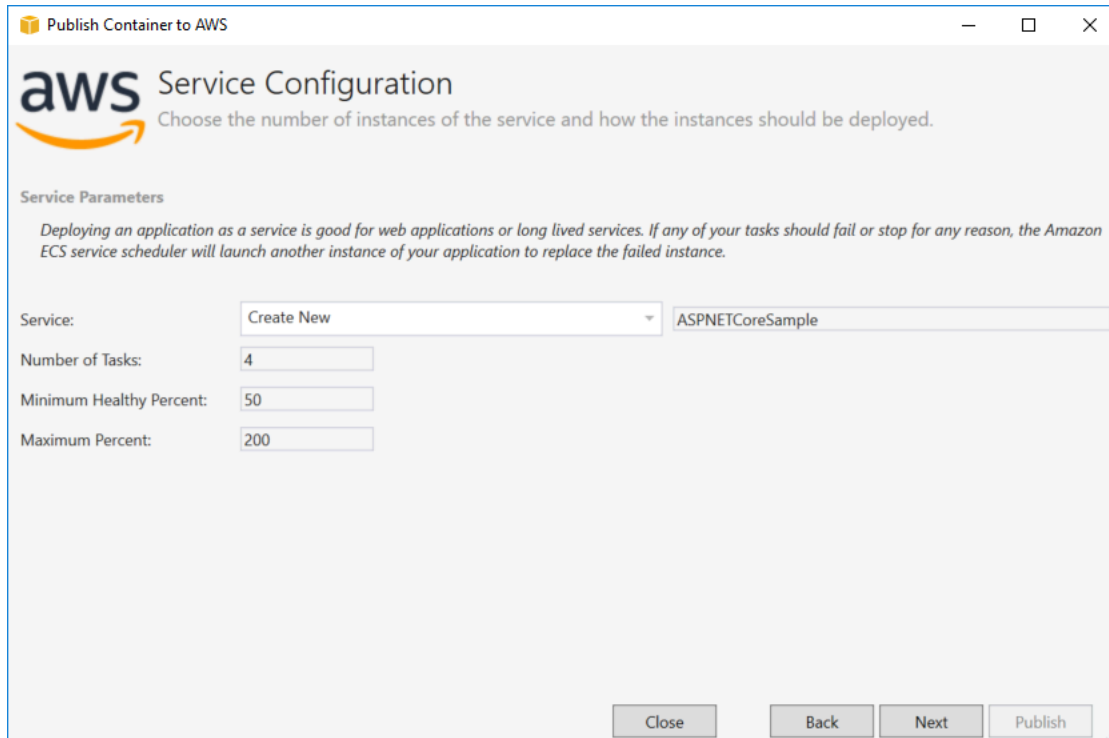
安全群組 - 選擇安全群組。

安全群組可做為相關聯 Amazon EC2 執行個體的防火牆，在執行個體層級控制傳入和傳出流量。

[預設安全群組](#)設定為允許從指派給相同安全群組的執行個體和所有傳出 IPv4 流量傳入流量。您需要允許傳出，服務才能連線到容器儲存庫。

指派公有 IP 地址 - 勾選此選項，讓您的任務可從網際網路存取。

服務組態頁面



The screenshot shows the 'Publish Container to AWS' dialog box. The title bar reads 'Publish Container to AWS'. The main heading is 'aws Service Configuration' with the subtitle 'Choose the number of instances of the service and how the instances should be deployed.' Below this, there is a section titled 'Service Parameters' with a note: 'Deploying an application as a service is good for web applications or long lived services. If any of your tasks should fail or stop for any reason, the Amazon ECS service scheduler will launch another instance of your application to replace the failed instance.' The form contains the following fields: 'Service:' with a dropdown menu set to 'Create New' and a text box containing 'ASPNETCoreSample'; 'Number of Tasks:' with a text box containing '4'; 'Minimum Healthy Percent:' with a text box containing '50'; and 'Maximum Percent:' with a text box containing '200'. At the bottom, there are four buttons: 'Close', 'Back', 'Next', and 'Publish'.

服務 - 在下拉式清單中選取其中一個服務，將您的容器部署到現有的服務。或者，選擇建立新服務以建立新的服務。叢集中不得有相同的服務名稱，但一個區域內或多個區域間的多個叢集中可以有類似的服務名稱。

任務數量 - 在叢集上部署和持續執行的任務數量。每個任務都是您容器的一個執行個體。

最小良好百分比 - 部署期間必須保持 RUNNING 狀態的任務百分比，四捨五入至最接近的整數。

百分比上限 - 部署期間允許處於 RUNNING 或 PENDING 狀態的任務百分比，四捨五入至最接近的整數。

Application Load Balancer 頁面

Publish Container to AWS

aws Application Load Balancer Configuration

Using an Application Load Balancer allows multiple instances of the application be accessible through a single URL endpoint.

Configure Application Load Balancer

It is recommended for web applications to use an Application Load Balancer which allows containers to use dynamic host port mapping. This will give the ability to run multiple instances of the web applications on the same container host without contention for port 80.

Load Balancer:

Listener Port:

Load Balancer Target Group

The Application Load Balancer will send requests to the Target Group if the request matches the specified URL path pattern. Amazon ECS will register all instances of the container with their dynamic port to the Target Group using the provided IAM role for the service.

Target Group:

Path Pattern:

Health Check Path:

設定 Application Load Balancer - 檢查以設定應用程式負載平衡器。

Load Balancer - 選取現有的負載平衡器，或選擇建立新負載平衡器，然後輸入新負載平衡器的名稱。

接聽程式連接埠 - 選取現有的接聽程式連接埠，或選擇建立新並輸入連接埠號碼。預設連接埠 80 適用於大多數 Web 應用程式。

目標群組 - 選取 Amazon ECS 會將任務註冊到服務的目標群組。

路徑模式 - 負載平衡器將使用路徑型路由。接受預設值/或提供不同的模式。路徑模式區分大小寫，長度最多可達 128 個字元，並包含 [一組選取的字元](#)。

運作狀態檢查路徑 - 運作狀態檢查目標上的目的地 ping 路徑。在預設情況下，大小上限為 /。如有需要，請輸入不同的路徑。如果您輸入的路徑無效，運作狀態檢查將會失敗，而且會被視為運作狀態不佳。

如果您部署多個服務，而且每個服務都會部署到不同的路徑或位置，您將需要自訂檢查路徑。

任務定義頁面

Publish Container to AWS

aws Task Definition
Task Definition defines the parameters for how the application will run within its Docker container.

Task Definition: ASPNETCoreSample

Container: ASPNETCoreSample

Permissions

Task Role:

Select an IAM role to provide AWS credentials to your application to access AWS Services.

Task Execution Role:

Fargate requires a role to pull private images and publish logs on your behalf.

Port Mapping

Container Port
80

Environment Variables

Variable	Value
ASPNETCORE_ENVIRONMENT	Production

Buttons: Add... Add... Close Back Next Publish

任務定義 - 選取現有的任務定義，或選擇建立新並輸入新的任務定義名稱。

容器 - 選取現有的容器，或選擇建立新容器，然後輸入新的容器名稱。

任務角色 - 選取具有應用程式存取 AWS 服務所需登入資料的 IAM 角色。這是登入資料傳入您的應用程式的方式。[了解如何為您的應用程式指定 AWS 安全登入資料](#)。

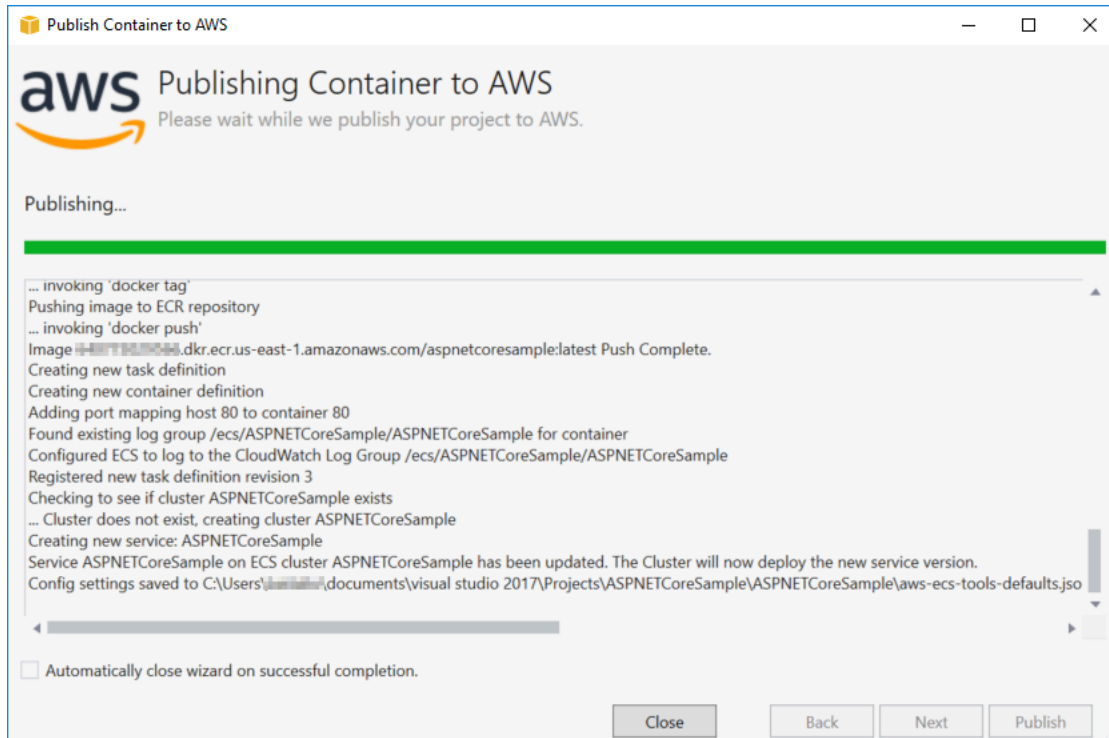
任務執行角色 - 選取具有提取私有映像和發佈日誌許可的角色。AWS Fargate 將代表您使用它。

連接埠映射 - 選擇容器上繫結至自動指派主機連接埠的連接埠號碼。

環境變數 - 新增、修改或刪除容器的環境變數。您可以修改它以符合您的部署。

當您對組態感到滿意時，請按一下發佈以開始部署程序。

將容器發佈至 AWS



事件會在部署期間顯示。成功完成時精靈會自動關閉。您可以取消核取方塊頁面底部的方塊來覆寫它。

您可以在 AWS Explorer 中找到新執行個體的 URL。展開 Amazon ECS 和叢集，然後按一下叢集。

將 ASP.NET Core 2.0 應用程式部署至 Amazon ECS (EC2)

本節說明如何使用 Toolkit for Visual Studio 提供的發佈容器精靈 AWS，使用 EC2 啟動類型透過 Amazon ECS 部署以 Linux 為目標的容器化 ASP.NET Core 2.0 應用程式。EC2 由於 Web 應用程式表示會持續執行，因此會部署為服務。

在您發佈容器之前

使用發佈容器至 AWS 部署 ASP.NET Core 2.0 應用程式之前：

- [指定您的 AWS 登入資料](#)，並使用 [Amazon ECS 進行設定](#)。
- [安裝 Docker](#)。您有一些不同的安裝選項，包括 [Docker for Windows](#)。
- 根據 Web 應用程式的需求 [建立 Amazon ECS 叢集](#)。這只需要幾個步驟。
- 在 Visual Studio 中，為以 Linux 為目標的 ASP.NET Core 2.0 容器化應用程式建立（或開啟）專案。

存取要 AWS 精靈的發佈容器

若要部署以 Linux 為目標的 ASP.NET Core 2.0 容器化應用程式，請在 Solution Explorer 中的專案上按一下滑鼠右鍵，然後選取將容器發佈至 AWS。

您也可以 Visual Studio Build 功能表中選取將容器發佈至 AWS。

將容器發佈至 AWS 精靈

要使用的帳戶設定檔 - 選取要使用的帳戶設定檔。

區域 - 選擇部署區域。設定檔和區域用於設定部署環境資源，並選取預設 Docker 登錄檔。

組態 - 選取 Docker 映像建置組態。

Docker 儲存庫 - 選擇現有的 Docker 儲存庫，或輸入新儲存庫的名稱，然後建立該儲存庫。這是推送建置容器映像的儲存庫。

標籤 - 選取現有標籤或輸入新標籤的名稱。標籤可以追蹤 Docker 容器的版本、選項或其他唯一組態元素等重要詳細資訊。

部署 - 選取 ECS 叢集上的服務。當您的應用程式需要長時間執行時（例如 ASP.NET Core 2.0 Web 應用程式），請使用此部署選項。

將設定儲存至 **aws-docker-tools-defaults.json** 並設定專案以進行命令列部署 - 如果您想要從命令列部署的彈性，請勾選此選項。dotnet ecs deploy 從您的專案目錄使用來部署和 dotnet ecs publish 容器。

啟動組態頁面

ECS 叢集 - 選擇將執行 Docker 映像的叢集。您可以使用 AWS 管理主控台 [建立 ECS 叢集](#)。

啟動類型 - 選擇 EC2。若要使用 Fargate 啟動類型，請參閱 [將 ASP.NET Core 2.0 應用程式部署到 Amazon ECS \(Fargate\)](#)。

服務組態頁面

服務 - 在下拉式清單中選取其中一個服務，將您的容器部署到現有的服務。或者，選擇建立新服務以建立新的服務。叢集中不得有相同的服務名稱，但一個區域內或多個區域間的多個叢集中可以有類似的服務名稱。

任務數量 - 在叢集上部署和持續執行的任務數量。每個任務都是您容器的一個執行個體。

最小良好百分比 - 部署期間必須保持 RUNNING 狀態的任務百分比，四捨五入至最接近的整數。

百分比上限 - 部署期間允許處於 RUNNING 或 PENDING 狀態的任務百分比，四捨五入至最接近的整數。

置放範本 - 選取任務置放範本。

當您在叢集中啟動任務時，Amazon ECS 必須根據任務定義中指定的需求，例如 CPU 和記憶體，來決定將任務放置在何處。同樣地，當您縮減任務計數時，Amazon ECS 必須判斷要終止的任務。

置放範本會控制任務在叢集中啟動的方式：

- AZ Balanced Spread (AZ 平衡分配) - 跨可用區域及跨可用區域中的容器執行個體分散任務。
- AZ Balanced BinPack (AZ 平衡 BinPack) - 使用最低可用記憶體，跨可用區域及跨可用區域中的容器執行個體分散任務。
- BinPack - 根據最低可用的 CPU 或記憶體數分散任務。
- One Task Per Host (每一主機一個任務) - 在每個容器執行個體上最多放置一個來自服務的任務。

如需詳細資訊，請參閱 [Amazon ECS 任務置放](#)。

Application Load Balancer 頁面

設定 Application Load Balancer - 檢查以設定應用程式負載平衡器。

選取服務的 IAM 角色 - 選取現有角色，或選擇建立新角色，然後建立新角色。

Load Balancer - 選取現有的負載平衡器，或選擇建立新負載平衡器，然後輸入新負載平衡器的名稱。

接聽程式連接埠 - 選取現有的接聽程式連接埠，或選擇建立新並輸入連接埠號碼。預設連接埠 80 適用於大多數 Web 應用程式。

目標群組 - 根據預設，負載平衡器會使用您為目標群組指定的連接埠和通訊協定，將請求傳送至已註冊的目標。在透過目標群組來註冊每個目標時，您可以覆寫此埠號。

路徑模式 - 負載平衡器將使用路徑型路由。接受預設值/或提供不同的模式。路徑模式區分大小寫，長度最多可達 128 個字元，並包含 [一組選取的字元](#)。

運作狀態檢查路徑 - 運作狀態檢查目標上的目的地 ping 路徑。根據預設，它/適用於 Web 應用程式。視需要輸入不同的路徑。如果您輸入的路徑無效，運作狀態檢查將會失敗，而且會被視為運作狀態不佳。

如果您部署多個服務，而且每個服務都會部署到不同的路徑或位置，您可能需要自訂檢查路徑。

ECS 任務定義頁面

任務定義 - 選取現有的任務定義，或選擇建立新並輸入新的任務定義名稱。

容器 - 選取現有的容器，或選擇建立新容器，然後輸入新的容器名稱。

記憶體 (MiB) - 提供軟性限制或硬性限制或兩者的值。

要預留給容器的記憶體軟性限制（以 MiB 為單位）。Docker 會嘗試將容器記憶體保持在軟性限制之下。容器可能會耗用更多記憶體，最高可達以記憶體參數（如果適用）指定的硬性限制，或容器執行個體上所有可用的記憶體，以先到者為準。

要呈現給容器的記憶體硬性限制（以 MiB 為單位）。如果您的容器嘗試使用超過此處指定的記憶體，容器便會終止。

任務角色 - 選取 IAM 角色的任務角色，允許容器代表您呼叫其相關聯政策中指定的 AWS APIs。這是將登入資料傳入應用程式的方式。[了解如何為您的應用程式指定 AWS 安全登入資料](#)。

連接埠映射 - 新增、修改或刪除容器的連接埠映射。如果負載平衡器已開啟，主機連接埠預設為 0，而連接埠指派將為動態。

環境變數 - 新增、修改或刪除容器的環境變數。

當您對組態感到滿意時，請按一下發佈以開始部署程序。

將容器發佈至 AWS

事件會在部署期間顯示。成功完成時精靈會自動關閉。您可以取消核取方塊頁面底部的方塊來覆寫它。

您可以在 AWS Explorer 中找到新執行個體的 URL。展開 Amazon ECS 和叢集，然後按一下叢集。

故障診斷 AWS Toolkit for Visual Studio

下列各節包含有關 AWS Toolkit for Visual Studio 和從 工具組使用 AWS 服務的一般疑難排解資訊。

Note

安裝和set-up-specific疑難排解資訊，請參閱本使用者指南中的[疑難排解安裝問題](#)主題。

主題

- [疑難排解最佳實務](#)
- [檢視和篩選 Amazon Q 安全性掃描](#)
- [工具 AWS 組未正確安裝](#)
- [防火牆和代理設定](#)

疑難排解最佳實務

以下是疑難排解 AWS Toolkit for Visual Studio 問題時建議的最佳實務。

- 修復 Visual Studio 並重新啟動您的系統
- 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 AWS 工具組日誌。如需如何尋找您的 AWS Toolkit 日誌的詳細說明，請參閱本指南主題中的[如何尋找您的 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 GitHub 儲存庫的問題區段中報告您未解決[AWS Toolkit for Visual Studio 的問題](#)。

修復 Visual Studio 並重新啟動您的系統

1. 關閉所有執行中的 Visual Studio 執行個體。
2. 從 Windows 開始選單中，啟動 Visual Studio Installer。
3. 對 Visual Studio (受影響) 的安裝執行修復。這可讓 Visual Studio 重建其已安裝擴充功能的索引。

4. 在重新啟動 Visual Studio 之前重新啟動 Windows。

如何尋找您的 AWS Toolkit 日誌

1. 在 Visual Studio 主功能表中，展開延伸模組。
2. 選擇 AWS 工具組以展開 AWS 工具組功能表，然後選擇檢視工具組日誌。
3. 當 AWS Toolkit 日誌資料夾在您的作業系統中開啟時，請依日期排序檔案，並尋找任何包含目前問題相關資訊的日誌檔案。

檢視和篩選 Amazon Q 安全性掃描

若要在 Visual Studio 中檢視 Amazon Q 安全性掃描，請展開 Visual Studio 主功能表中的檢視標題，然後選擇錯誤清單，以開啟 Visual Studio 錯誤清單。

根據預設，Visual Studio 錯誤清單會顯示程式碼基底的所有警告和錯誤。若要從 Visual Studio 錯誤清單中篩選 Amazon Q 安全性掃描問題清單，請完成下列程序來建立篩選條件。

Note

只有在執行安全性掃描並偵測到問題之後，才會顯示 Amazon Q 安全性掃描問題清單。Amazon Q 安全性掃描調查結果會在 Visual Studio 中顯示為警告。若要從錯誤清單檢視 Amazon Q 安全性掃描問題清單，必須選取錯誤清單標題中的警告選項。

1. 從 Visual Studio 主功能表中，展開檢視標題，然後選擇錯誤清單以開啟錯誤清單窗格。
2. 在錯誤清單窗格中的標頭列上按一下滑鼠右鍵，以開啟內容功能表。
3. 從內容選單中，展開顯示資料欄，然後在展開的選單中選取工具。
4. 工具欄會新增至您的錯誤清單。
5. 從工具欄標題中，選取篩選圖示，然後選擇 Amazon Q 來篩選 Amazon Q 安全性掃描問題清單。

工具 AWS 組未正確安裝

問題：

啟動 Visual Studio 後一分鐘內，AWS Toolkit for Visual Studio 以下訊息分別會出現在輸出窗格和資訊列中：

Some Toolkit components could not be initialized. Some functionality may not work during this IDE session.

The AWS Toolkit is not properly installed.

解決方案：

更新或安裝擴充功能可能會導致一些 Visual Studio 的內部快取檔案out-of-sync。下列程序說明如何在下次啟動 Visual Studio 時重建這些檔案。

Note

此解決方案可能會影響您的 Visual Studio 自訂。完成此程序後，AWS Toolkit 擴充功能應列為已安裝，不再報告錯誤訊息。如果您在完成下列步驟後繼續遇到此問題，請參閱 AWS Toolkit for Visual Studio GitHub 儲存庫中的[問題編號 452](#)，以取得其他資訊。

1. 安裝最新版本的 Visual Studio 2022。

Note

最低必要版本為 17.11.5。

2. 關閉所有執行中的 Visual Studio 執行個體。
3. 在 Windows 中，以管理員身分開啟開發人員命令提示字元。
4. 從開發人員命令提示字元中，執行下列命令：`devenv /updateconfiguration /resetExtensions`，然後等待命令完成。
5. 命令完成後，重新啟動 Visual Studio。
6. 在 Visual Studio 中，AWS 延伸項目現在會列為已安裝，且不再報告此問題頂端列出的錯誤訊息。

防火牆和代理設定

防火牆和代理設定故障診斷

安全掃描軟體可能會干擾您從 AWS Toolkit 語言伺服器下載檔案的能力，方法是從下載中移除檔案或防止下載。

若要檢查您的防火牆和代理設定，請從與 Visual Studio 執行個體安裝在相同系統上的網際網路瀏覽器導覽至 <https://aws-toolkit-language-servers.amazonaws.com/codewhisperer/0/manifest.json>。如果您遇到錯誤或頁面無法載入，則可能有防火牆或代理篩選條件阻止您到達 `aws-toolkit-language-servers.amazonaws.com`。

自訂憑證

AWS Toolkit for Visual Studio 使用在 Node.js 執行時間上執行的語言伺服器。如需如何檢查網路是否使用自訂憑證的詳細資訊，請參閱《第 1 版 AWS Command Line Interface 使用者指南》中的主題中的 [組態和憑證檔案設定 AWS CLI](#)。

若要設定代理設定並定義憑證，您必須設定 `HTTPS_PROXY` 您的 `env` 變數，並為 `NODE_OPTIONS` 和 `NODE_EXTRA_CA_CERTS` 金鑰建立 Windows 環境變數。

若要設定您的 `HTTPS_PROXY` `env` 變數，請完成下列步驟。

1. 從 Visual Studio 主功能表選擇工具，然後選擇選項。
2. 從選項功能表中，展開 AWS Toolkit，然後選擇 Proxy。
3. 從 Proxy 功能表中，定義您的主機和連接埠。

Note

如需 `HTTPS_PROXY` 從設定的相關資訊 AWS CLI，請參閱 AWS Command Line Interface 《使用者指南》中的 [針對主題使用 HTTP 代理 AWS CLI](#)。

為下列金鑰建立 Windows 環境變數。

- `NODE_OPTIONS = --use-openssl-ca`
- `NODE_EXTRA_CA_CERTS = Path/To/Corporate/Certs`

Note

如需擷取公司根憑證的詳細資訊，請參閱在 <https://learn.microsoft.com> [使用私有金鑰文章匯出憑證](#)。如需 Windows 環境變數金鑰的詳細資訊，請參閱 [Node.js v23.3.0 文件](#)，網址為 <https://nodejs.org>。

允許列出和其他步驟

除了干擾 AWS Toolkit 語言伺服器之外，防火牆設定還可以防止 Amazon Q 上傳到 Amazon S3 並呼叫服務 API。為了將這些錯誤的可能性降至最低，我們建議允許下列端點在連接埠 443 (HTTPS) 上存取傳出網際網路：

- <https://codewhisperer.us-east-1.amazonaws.com/>
- <https://amazonq-code-transformation-us-east-1-c6160f047e0.s3.amazonaws.com/>
- <https://aws-toolkit-language-servers.amazonaws.com/>
- <https://q.us-east-1.amazonaws.com>
- <https://client-telemetry.us-east-1.amazonaws.com>
- <https://cognito-identity.us-east-1.amazonaws.com>
- <https://oidc.us-east-1.amazonaws.com>

如需端點的詳細清單，請參閱《使用者指南》中的[更新防火牆和開道以允許存取](#)主題。如需為 Amazon Q 設定公司代理的詳細資訊，請參閱《Amazon Q 開發人員使用者指南》中的[在 Amazon Q 主題中設定公司代理](#)。如果您持續遇到防火牆和代理問題，請收集您的 AWS Toolkit Logs，並透過 AWS Toolkit for Visual Studio GitHub 儲存庫[AWS Toolkit for Visual Studio 的問題](#)區段聯絡 AWS Toolkit for Visual Studio 團隊。如需收集工具 AWS 組日誌的詳細資訊，請參閱本使用者指南主題的疑難排解最佳實務一節中的資訊。

的安全性 AWS Toolkit for Visual Studio

雲端安全是 Amazon Web Services (AWS) 最重視的一環。身為 AWS 客戶的您，將能從資料中心和網路架構的建置中獲益，以滿足組織最為敏感的安全要求。安全性是 AWS 與您之間共同責任。[共同責任模型](#) 將此描述為雲端本身的安全和雲端內部的安全。

雲端的安全性 – AWS 負責保護執行 AWS 雲端中提供的所有服務的基礎設施，並為您提供可安全使用的服務。我們的安全責任是最高優先順序 AWS，我們的安全有效性由第三方稽核人員定期測試和驗證，作為[AWS 合規計劃](#)的一部分。

雲端的安全性 – 您的責任取決於您使用 AWS 的服務，以及其他因素，包括資料的敏感度、組織的需求，以及適用的法律和法規。

此 AWS 產品或服務會透過其支援的特定 Amazon Web Services (AWS) 服務，遵循[共同責任模型](#)。如需 AWS 服務安全資訊，請參閱[AWS 服務安全文件頁面](#)，以及[AWS 合規計劃在 AWS 合規工作範圍內的服務](#)。

主題

- [中的資料保護 AWS Toolkit for Visual Studio](#)
- [身分和存取權管理](#)
- [此 AWS 產品或服務的合規驗證](#)
- [此 AWS 產品或服務的彈性](#)
- [此 AWS 產品或服務的基礎設施安全](#)
- [中的組態和漏洞分析 AWS Toolkit for Visual Studio](#)

中的資料保護 AWS Toolkit for Visual Studio

AWS [共同責任模型](#)適用於 AWS Toolkit for Visual Studio 與 Amazon Q 中的資料保護。如本模型所述，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](#)。如需有關歐洲資料保護的相關資訊，請參閱AWS 安全性部落格上的[AWS 共同責任模型和 GDPR](#) 部落格文章。

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 AWS Toolkit 搭配 Amazon Q 或使用主控台 AWS CLI、API 或 AWS SDKs 的其他 AWS 服務 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。

身分和存取權管理

AWS Identity and Access Management (IAM) 是一種 AWS 服務，可協助管理員安全地控制對 AWS 資源的存取。IAM 管理員可控制誰可以進行驗證（登入）和授權（具有許可）來使用 AWS 資源。IAM 是您可以免費使用 AWS 服務的。

主題

- [目標對象](#)
- [使用身分驗證](#)
- [使用政策管理存取權](#)
- [AWS 服務 如何使用 IAM](#)
- [對 AWS 身分和存取進行故障診斷](#)

目標對象

您使用 AWS Identity and Access Management (IAM) 的方式會有所不同，取決於您在 中執行的工作 AWS。

服務使用者 – 如果您使用 AWS 服務 執行任務，管理員會為您提供所需的登入資料和許可。當您使用更多 AWS 功能來執行工作時，您可能需要額外的許可。了解存取許可的管理方式可協助您向管理員請

求正確的許可。如果您無法存取 中的功能 AWS，請參閱 [對 AWS 身分和存取進行故障診斷](#) 或 AWS 服務 您正在使用的 使用者指南。

服務管理員 – 如果您負責公司 AWS 的資源，您可能擁有的完整存取權 AWS。您的任務是判斷服務使用者應存取哪些 AWS 功能和資源。接著，您必須將請求提交給您的 IAM 管理員，來變更您服務使用者的許可。檢閱此頁面上的資訊，了解 IAM 的基本概念。若要進一步了解貴公司如何使用 IAM AWS，請參閱您正在使用的 使用者指南 AWS 服務。

IAM 管理員：如果您是 IAM 管理員，建議您掌握如何撰寫政策以管理 AWS 存取權的詳細資訊。若要檢視您可以在 IAM 中使用的以 AWS 身分為基礎的政策範例，請參閱 AWS 服務 您正在使用的 使用者指南。

使用身分驗證

身分驗證是您 AWS 使用身分憑證登入的方式。您必須以 AWS 帳戶根使用者、IAM 使用者或擔任 IAM 角色身分進行身分驗證。

您可以使用身分來源的登入資料，例如 AWS IAM Identity Center (IAM Identity Center)、單一登入身分驗證或 Google/Facebook 登入資料，以聯合身分的形式登入。如需有關登入的詳細資訊，請參閱《AWS 登入 使用者指南》中的 [如何登入您的 AWS 帳戶](#)。

對於程式設計存取，AWS 提供 SDK 和 CLI 以密碼編譯方式簽署請求。如需詳細資訊，請參閱《IAM 使用者指南》中的 [API 請求的 AWS 第 4 版簽署程序](#)。

AWS 帳戶 根使用者

當您建立時 AWS 帳戶，您會從一個名為 AWS 帳戶 theroot 使用者的登入身分開始，該身分具有對所有 AWS 服務 和 資源的完整存取權。強烈建議不要使用根使用者來執行日常任務。有關需要根使用者憑證的任務，請參閱《IAM 使用者指南》中的 [需要根使用者憑證的任務](#)。

聯合身分

最佳實務是要求人類使用者使用聯合身分提供者，以 AWS 服務 使用臨時憑證存取。

聯合身分是您企業目錄、Web 身分提供者的使用者，或使用來自身分來源的 AWS 服務 憑證存取 Directory Service。聯合身分會擔任角色，而該角色會提供臨時憑證。

若需集中化管理存取權限，建議使用 AWS IAM Identity Center。如需詳細資訊，請參閱 AWS IAM Identity Center 使用者指南中的 [什麼是 IAM Identity Center?](#)。

IAM 使用者和群組

IAM 使用者https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users.html是一種身分具備單人或應用程式的特定許可權。建議以臨時憑證取代具備長期憑證的 IAM 使用者。如需詳細資訊，請參閱《IAM 使用者指南》中的[要求人類使用者使用聯合身分提供者來 AWS 使用臨時憑證存取](#)。

[IAM 群組](#)會指定 IAM 使用者集合，使管理大量使用者的許可權更加輕鬆。如需詳細資訊，請參閱《IAM 使用者指南》中的[IAM 使用者的使用案例](#)。

IAM 角色

IAM 角色https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html的身分具有特定許可權，其可以提供臨時憑證。您可以透過[從使用者切換到 IAM 角色（主控台）](#)或呼叫 AWS CLI 或 AWS API 操作來擔任角色。如需詳細資訊，請參閱《IAM 使用者指南》中的[擔任角色的方法](#)。

IAM 角色適用於聯合身分使用者存取、臨時 IAM 使用者許可、跨帳戶存取權與跨服務存取，以及在 Amazon EC2 執行的應用程式。如需詳細資訊，請參閱《IAM 使用者指南》中的[IAM 中的快帳戶資源存取](#)。

使用政策管理存取權

您可以透過建立政策並將其連接到身分或資源 AWS 來控制 AWS 中的存取。政策定義與身分或資源相關聯的許可。當委託人提出請求時 AWS，會評估這些政策。大多數政策會以 JSON 文件 AWS 的形式存放在中。如需進一步了解 JSON 政策文件，請參閱《IAM 使用者指南》中的[JSON 政策概觀](#)。

管理員會使用政策，透過定義哪些主體可在哪些條件下對哪些資源執行動作，以指定可存取的範圍。

預設情況下，使用者和角色沒有許可。IAM 管理員會建立 IAM 政策並將其新增至角色，供使用者後續擔任。IAM 政策定義動作的許可，無論採用何種方式執行。

身分型政策

身分型政策是附加至身分 (使用者、使用者群組或角色) 的 JSON 許可政策文件。這類政策控制身分可對哪些資源執行哪些動作，以及適用的條件。如需了解如何建立身分型政策，請參閱《IAM 使用者指南》中的[透過客戶管理政策定義自訂 IAM 許可](#)。

身分型政策可分為內嵌政策 (直接內嵌於單一身分) 與受管政策 (可附加至多個身分的獨立政策)。如需了解如何在受管政策及內嵌政策之間做選擇，請參閱《IAM 使用者指南》中的[在受管政策與內嵌政策之間選擇](#)。

資源型政策

資源型政策是附加到資源的 JSON 政策文件。範例包括 IAM 角色信任政策與 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。您必須在資源型政策中[指定主體](#)。

資源型政策是位於該服務中的內嵌政策。您無法在資源型政策中使用來自 IAM 的 AWS 受管政策。

存取控制清單 (ACL)

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

Amazon S3 AWS WAF 和 Amazon VPC 是支援 ACLs 的服務範例。如需進一步了解 ACL，請參閱《Amazon Simple Storage Service 開發人員指南》中的[存取控制清單 \(ACL\) 概觀](#)。

其他政策類型

AWS 支援其他政策類型，可設定更多常見政策類型授予的最大許可：

- 許可界限 — 設定身分型政策可授與 IAM 實體的最大許可。如需詳細資訊，請參閱《IAM 使用者指南》中的[IAM 實體許可界限](#)。
- 服務控制政策 (SCP) — 為 AWS Organizations 中的組織或組織單位指定最大許可。如需詳細資訊，請參閱《AWS Organizations 使用者指南》中的[服務控制政策](#)。
- 資源控制政策 (RCP) — 設定您帳戶中資源可用許可的上限。如需詳細資訊，請參閱《AWS Organizations 使用者指南》中的[資源控制政策 \(RCP\)](#)。
- 工作階段政策 — 在以程式設計方式為角色或聯合身分使用者建立臨時工作階段時，以參數形式傳遞的進階政策。如需詳細資訊，請參閱《IAM 使用者指南》中的[工作階段政策](#)。

多種政策類型

當多種類型的政策適用於請求時，產生的許可會更複雜而無法理解。若要了解如何 AWS 在涉及多種政策類型時決定是否允許請求，請參閱《IAM 使用者指南》中的[政策評估邏輯](#)。

AWS 服務 如何使用 IAM

若要深入了解 如何使用 AWS 服務 大多數 IAM 功能，請參閱《IAM 使用者指南》中的[AWS 使用 IAM 的服務](#)。

若要了解如何 AWS 服務 搭配 IAM 使用特定，請參閱相關服務使用者指南的安全章節。

對 AWS 身分和存取進行故障診斷

使用以下資訊來協助您診斷和修正使用 AWS 和 IAM 時可能遇到的常見問題。

主題

- [我無權在 中執行動作 AWS](#)
- [我未獲得執行 iam:PassRole 的授權](#)
- [我想要允許 以外的人員 AWS 帳戶 存取我的 AWS 資源](#)

我無權在 中執行動作 AWS

如果您收到錯誤，告知您未獲授權執行動作，您的政策必須更新，允許您執行動作。

下列範例錯誤會在mateojackson IAM 使用者嘗試使用主控台檢視一個虛構 *my-example-widget* 資源的詳細資訊，但卻無虛構 `aws:GetWidget` 許可時發生。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
aws:GetWidget on resource: my-example-widget
```

在此情況下，必須更新 mateojackson 使用者的政策，允許使用 `aws:GetWidget` 動作存取 *my-example-widget* 資源。

如果您需要協助，請聯絡您的 AWS 管理員。您的管理員提供您的簽署憑證。

我未獲得執行 iam:PassRole 的授權

如果您收到錯誤，告知您未獲授權執行 `iam:PassRole` 動作，您的政策必須更新，允許您將角色傳遞給 AWS。

有些 AWS 服務 可讓您將現有角色傳遞給該服務，而不是建立新的服務角色或服務連結角色。如需執行此作業，您必須擁有將角色傳遞至該服務的許可。

名為 marymajor 的 IAM 使用者嘗試使用主控台在 AWS 中執行動作時，發生下列範例錯誤。但是，動作要求服務具備服務角色授予的許可。Mary 沒有將角色傳遞給服務的許可。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

在這種情況下，Mary 的政策必須更新，允許她執行 `iam:PassRole` 動作。

如果您需要協助，請聯絡您的 AWS 管理員。您的管理員提供您的簽署憑證。

我想要允許以外的人員 AWS 帳戶存取我的 AWS 資源

您可以建立一個角色，讓其他帳戶中的使用者或您組織外部的人員存取您的資源。您可以指定要允許哪些信任物件取得該角色。針對支援基於資源的政策或存取控制清單 (ACL) 的服務，您可以使用那些政策來授予人員存取您的資源的許可。

如需進一步了解，請參閱以下內容：

- 若要了解是否 AWS 支援這些功能，請參閱 [AWS 服務 如何使用 IAM](#)。
- 若要了解如何 AWS 帳戶 在您擁有的 資源之間提供存取權，請參閱《[IAM 使用者指南](#)》中的 [在您擁有 AWS 帳戶 的另一個 IAM 使用者中提供存取權](#)。
- 若要了解如何將資源的存取權提供給第三方 AWS 帳戶，請參閱《IAM 使用者指南》中的 [將存取權提供給第三方 AWS 帳戶 擁有](#)。
- 如需了解如何透過聯合身分提供存取權，請參閱《IAM 使用者指南》中的 [將存取權提供給在外部進行身分驗證的使用者 \(聯合身分\)](#)。
- 如需了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱《IAM 使用者指南》中的 [IAM 中的跨帳戶資源存取](#)。

此 AWS 產品或服務的合規驗證

若要了解 AWS 服務 是否在特定合規計劃範圍內，請參閱 [AWS 服務 合規計劃範圍內](#) 然後選擇您感興趣的合規計劃。如需一般資訊，請參閱 [AWS 合規計劃](#)。

您可以使用 下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱 [下載報告 in AWS Artifact](#)

您使用 時的合規責任 AWS 服務 取決於資料的機密性、您公司的合規目標，以及適用的法律和法規。如需使用 時合規責任的詳細資訊 AWS 服務，請參閱 [AWS 安全文件](#)。

此 AWS 產品或服務會透過其支援的特定 Amazon Web Services (AWS) 服務，遵循 [共同責任模型](#)。如需 AWS 服務安全資訊，請參閱 [AWS 服務安全文件頁面](#)，以及 [AWS 合規計劃在 AWS 合規工作範圍內的服務](#)。

此 AWS 產品或服務的彈性

AWS 全球基礎設施是以 AWS 區域 和可用區域為基礎建置。

AWS 區域 提供多個實體分隔和隔離的可用區域，這些可用區域與低延遲、高輸送量和高備援聯網連接。

透過可用區域，您可以設計與操作的應用程式和資料庫，在可用區域之間自動容錯移轉而不會發生中斷。可用區域的可用性、容錯能力和擴展能力，均較單一或多個資料中心的傳統基礎設施還高。

如需 AWS 區域和可用區域的詳細資訊，請參閱 [AWS 全球基礎設施](#)。

此 AWS 產品或服務會透過其支援的特定 Amazon Web Services (AWS) 服務，遵循[共同責任模型](#)。如需 AWS 服務安全資訊，請參閱[AWS 服務安全文件頁面](#)，以及[AWS 合規計劃在 AWS 合規工作範圍內的服務](#)。

此 AWS 產品或服務的基礎設施安全

此 AWS 產品或服務使用 受管服務，因此受到 全球網路安全的 AWS 保護。如需 AWS 安全服務以及如何 AWS 保護基礎設施的資訊，請參閱[AWS 雲端安全](#)。若要使用基礎設施安全的最佳實務來設計您的 AWS 環境，請參閱安全支柱 AWS Well-Architected Framework 中的[基礎設施保護](#)。

您可以使用 AWS 發佈的 API 呼叫，透過網路存取此 AWS 產品或服務。使用者端必須支援下列專案：

- Transport Layer Security (TLS)。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 具備完美轉送私密(PFS)的密碼套件，例如 DHE (Ephemeral Diffie-Hellman)或 ECDHE (Elliptic Curve Ephemeral Diffie-Hellman)。現代系統(如 Java 7 和更新版本)大多會支援這些模式。

此外，請求必須使用存取金鑰 ID 和與 IAM 主體相關聯的私密存取金鑰來簽署。或者，您可以透過 [AWS Security Token Service](#) (AWS STS) 來產生暫時安全憑證來簽署請求。

此 AWS 產品或服務會透過其支援的特定 Amazon Web Services (AWS) 服務，遵循[共同責任模型](#)。如需 AWS 服務安全資訊，請參閱[AWS 服務安全文件頁面](#)，以及[AWS 合規計劃在 AWS 合規工作範圍內的服務](#)。

中的組態和漏洞分析 AWS Toolkit for Visual Studio

Toolkit for Visual Studio 會在開發新功能或修正時發佈至 [Visual Studio Marketplace](#)。這些更新有時包含安全性更新，因此請務必讓 AWS Toolkit 與 Amazon Q 保持最新狀態。

確認已啟用擴充功能的自動更新

1. 選擇工具、延伸項目和更新 (Visual Studio 2017) 或延伸項目、管理延伸項目 (Visual Studio 2019) 來開啟延伸項目管理員。

2. 選擇變更您的延伸項目和更新設定 (Visual Studio 2017) , 或變更延伸項目的設定 (Visual Studio 2019)。
3. 調整環境的設定。

如果您選擇停用擴充功能的自動更新，請務必以適合您環境的間隔檢查 AWS Toolkit with Amazon Q 的更新。

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

文件歷史紀錄

下表說明 AWS Toolkit for Visual Studio 使用者指南中最近重要的變更。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
內容入門的更新	對入門和連線至 AWS 內容所做的更新，以反映在 UI 中所做的變更。	2025 年 4 月 24 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 AWS Toolkit for Visual Studio 具有 Amazon Q 擴充功能之 中的所有服務和功能。	2025 年 3 月 20 日
對防火牆和代理設定進行故障診斷	新增疑難排解主題，以解決和 Amazon Q 的防火牆 AWS Toolkit for Visual Studio 和代理設定。	2024 年 12 月 15 日
對安裝更新進行故障診斷	更新安裝問題內容以考量 Microsoft 的更新。	2024 年 11 月 20 日
內容入門的更新	對入門和連線至 AWS 內容所做的更新，以反映在 UI 中所做的變更。	2024 年 10 月 24 日
連線至 的更新 AWS	對連線至 AWS 內容所做的更新。	2024 年 9 月 26 日
Amazon EC2 AMI 內容的更新	內容已更新，以記錄 Amazon EC2 AMI 程序的變更。	2024 年 9 月 13 日

AWS 工具組元件無法初始化	新增故障診斷主題，以解決 AWS Toolkit for Visual Studio 元件未初始化的問題。	2024 年 9 月 13 日
檢視和篩選 Amazon Q 安全性掃描	新增故障診斷主題，以協助檢視和篩選 Amazon Q 安全性掃描。	2024 年 7 月 31 日
適用於的 Amazon Q AWS Toolkit for Visual Studio	Amazon Q 現在可供使用 AWS Toolkit for Visual Studio。	2024 年 6 月 30 日
內容更新和維護	更新 UI 和 AWS 樣式準則變更的內容。	2024 年 3 月 6 日
內容更新和維護	更新 UI 和 AWS 樣式準則變更的內容。	2024 年 3 月 6 日
內容更新和維護	更新 UI 和 AWS 樣式準則變更的內容。	2024 年 3 月 6 日
內容更新和維護	更新 UI 和 AWS 樣式準則變更的內容。	2024 年 3 月 6 日
內容更新和維護	更新 UI 和 AWS 樣式準則變更的內容。	2024 年 3 月 6 日
設定和身分驗證的更新	已更新設定和身分驗證主題，以改善安全性和工具組入門體驗。請參閱 入門 和 身分驗證和存取 主題 TOCs 以檢視變更。	2023 年 6 月 22 日
身分驗證和存取	提供 AWS 登入資料現在是身分驗證和存取。重構 TOC 和子主題以符合 AWS 樣式和私密要求。	2023 年 5 月 4 日

[設定區段和主題的更新](#)

本使用者指南中的[設定 AWS Toolkit for Visual Studio](#)章節和主題已更新，以改善的加入體驗 AWS Toolkit for Visual Studio。

2023 年 1 月 30 日

[設定區段和主題的更新](#)

本使用者指南中的[設定 AWS Toolkit for Visual Studio](#)章節和主題已更新，以改善的加入體驗 AWS Toolkit for Visual Studio。

2023 年 1 月 30 日

[新增 2022 年 AWS Toolkit for Visual Studio 資訊](#)

Visual Studio 2022 的支援已新增至 AWS Toolkit for Visual Studio。

2022 年 12 月 20 日

[發佈以 AWS 引導的更新](#)

文件更新以反映對 GA 啟動服務所做的變更。

2022 年 7 月 6 日

[標題更新和重新定位](#)

進行了次要標題變更，以更好地反映內容。指南現在位於發佈指南中 AWS。

2022 年 7 月 6 日

[部署至 AWS：標題和內容更新](#)

本指南章節正式標題為：使用 AWS Toolkit 部署、具有更新的目錄 (TOC)，現在標題為：部署至 AWS。下列指南已完成棄用，無法再存取：部署至 Elastic Beanstalk (舊版) 和部署至 AWS CloudFormation (舊版)。如需 Elastic Beanstalk 和 Cloudformation 部署的更新內容，請參閱本指南中更新的 TOC。

2022 年 7 月 6 日

[部署 ASP.NET Core 2.0 應用程式 \(Fargate\) 現在是舊版指南](#)

本文件是指舊版服務和功能。如需更新的指南和內容，請參閱 [AWS .NET 部署工具](#) 指南和更新的 [部署至 AWS](#) 目錄。

2022 年 7 月 6 日

[部署 ASP.NET 應用程式現在是舊版指南](#)

本文件是指舊版服務和功能。如需更新的指南和內容，請參閱 [AWS .NET 部署工具](#) 指南和更新的 [部署至 AWS](#) 目錄。

2022 年 7 月 6 日

[部署 ASP.NET 應用程式現在是舊版指南](#)

本文件是指舊版服務和功能。如需更新的指南和內容，請參閱 [AWS .NET 部署工具](#) 指南和更新的 [部署至 AWS](#) 目錄。

2022 年 7 月 6 日

[新指南主題：在 Visual Studio 中使用 CloudWatch Logs](#)

為 [Visual Studio 指南中的 Amazon CloudWatch Logs 整合](#) 建立新的概觀主題。

2022 年 6 月 29 日

[新指南主題：設定 Visual Studio 的 CloudWatch Logs 整合](#)

為 [Visual Studio 指南中的 Amazon CloudWatch Logs 整合](#) 建立新的設定區段。

2022 年 6 月 29 日

[Visual Studio 的 CloudWatch Logs 整合](#)

為 Visual Studio 中的 Amazon CloudWatch Logs 整合建立新的指南，包括指南主題：[設定 Visual Studio 的 CloudWatch Logs](#) 和 [在 Visual Studio 中使用 CloudWatch Logs](#)。

2022 年 6 月 29 日

[發佈至 AWS](#)

發佈至 AWS 不再處於預覽狀態。更新以反映 UI 的變更和發佈建議的改進。

2022 年 6 月 1 日

[新發佈至 AWS 可供預覽](#)

增強的部署體驗，提供適用於您應用程式的 AWS 服務指引。

2021 年 10 月 21 日

AWS 憑證的 SSO 和 MFA 支援	更新以記錄 AWS 憑證中對 AWS 單一登入 (IAM Identity Center) 和多重要素驗證的新支援。	2021 年 4 月 21 日
基本 AWS Lambda 專案建立 Docker 影像	新增對 Lambda 容器映像的支援。	2020 年 12 月 1 日
安全內容	已新增安全內容。	2020 年 2 月 6 日
提供 AWS 登入資料	更新有關在共用 AWS 登入資料檔案中建立登入資料設定檔的資訊。	2019 年 6 月 20 日
在 AWS Toolkit for Visual Studio 中使用 AWS Lambda 專案	支援 Visual Studio 2019 已新增至 AWS Toolkit for Visual Studio。	2019 年 3 月 28 日
教學課程：建立 Amazon Rekognition Lambda 應用程式	支援 Visual Studio 2019 已新增至 AWS Toolkit for Visual Studio。	2019 年 3 月 28 日
教學課程：使用 AWS Lambda 建置和測試無伺服器應用程式	支援 Visual Studio 2019 已新增至 AWS Toolkit for Visual Studio。	2019 年 3 月 28 日
設定 AWS Toolkit for Visual Studio	Visual Studio 2019 的支援已新增至 AWS Toolkit for Visual Studio。	2019 年 3 月 28 日
部署 ASP.NET Core 2.0 應用程式 (Fargate)	支援 Visual Studio 2019 已新增至 AWS Toolkit for Visual Studio。	2019 年 3 月 28 日
部署 ASP.NET Core 2.0 應用程式 (EC2)	支援 Visual Studio 2019 已新增至 AWS Toolkit for Visual Studio。	2019 年 3 月 28 日

在 Visual Studio 中建立 an AWS CloudFormation 範本專案	支援 Visual Studio 2019 已新增至 AWS Toolkit for Visual Studio。	2019 年 3 月 28 日
Container Service 的詳細檢視	新增 Explorer 所提供 Amazon Elastic Container Service 叢集和容器儲存庫詳細檢視的相關資訊 AWS。	2018 年 2 月 16 日
部署至 Amazon EC2 Container Service	新增部署至 Amazon EC2 容器服務的相關資訊。	2018 年 2 月 16 日
使用 Fargate 部署容器服務	新增如何使用 Fargate 啟動類型透過 Amazon ECS 部署以 Linux 為目標的容器化 ASP.NET Core 2.0 應用程式的相關資訊。	2018 年 2 月 16 日
使用 EC2 部署容器服務	新增如何使用 EC2 啟動類型透過 Amazon ECS 部署以 Linux 為目標的容器化 ASP.NET Core 2.0 應用程式的相關資訊。EC2	2018 年 2 月 16 日
部署至 Amazon EC2 Container Service 的登入資料	新增如何在部署至 Amazon EC2 容器服務時指定登入資料的相關資訊。	2018 年 2 月 16 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。