使用者指南

AWS Toolkit for Visual Studio

AWS Toolkit for Visual Studio: 使用者指南

Table of Contents

AWS Toolkit for Visual Studio

這是的使用者指南AWS Toolkit for Visual Studio。如果您正在尋找 AWS Toolkit for VS Code，請參閱 AWS Toolkit for Visual Studio Code.

什麼是 Toolkit for Visual Studio

這 AWS Toolkit for Visual Studio 是 Visual Studio IDE 的外掛程式，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。Toolkit for Visual Studio 的工具組支援 2019 年及更新版本。如需有關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「安裝與設定」主題。

> **Note**
>
> Toolkit for Visual Studio 也發布了視覺工作室 2008 年，2010，2012，2013 年，2015 年和 2017 年的版本。但是，不再支持這些版本。如需詳細資訊，請參閱本使用指南中的〈安裝和設定〉主題。

Toolkit for Visual Studio 包含下列功能，可增強您的開發體驗。

AWS 瀏覽器

AWS [檔案總管] 工具視窗 (可從 IDE 的 [檢視] 功能表取得) 可讓您從 Visual Studio IDE 內部與許多 AWS 服務互動。支持的數據服務包括 Amazon 簡單存儲服務（Amazon S3），Amazon SimpleDB，亞馬遜 Simple Notification Service (Amazon SNS)，Amazon Simple Queue Service (Amazon SQS) 和亞馬遜。 CloudFront AWS 資源管理器還提供 Amazon Elastic Compute Cloud (Amazon EC2) 管理、 AWS Identity and Access Management (IAM) 使用者和政策管理、無伺服器應用程式和功能部署至 AWS Lambda 和的 Web 應用程式的存取權。 AWS Elastic Beanstalk AWS CloudFormation

認證和區域管理

AWS Explorer 支援多個 AWS 帳戶 (包括 IAM 使用者帳戶) 和區域，可讓您輕鬆將顯示的檢視從一個帳戶變更為另一個帳戶，或檢視和管理不同地區的資源和服務。

Amazon EC2

您可以從 AWS 檔案總管檢視可用的 Amazon 機器映像 (AMI)、從這些 AMI 建立 Amazon EC2 執行個體，然後使用 Windows 遠端桌面連線到這些執行個體。 AWS Explorer 也會啟用支援功能，例如建立和管理金鑰配對和安全性群組的功能。

AWS Lambda

您可以使用 Lambda 來託管無伺服器 .NET Core C# 函數和無伺服器應用程式。使用藍圖快速建立新的無伺服器專案，並取得開發無伺服器應用程式的開始。

AWS CodeCommit

CodeCommit 與視覺工作室團隊資源管理器集成。這樣可以輕鬆地克隆和創建保存的存儲庫 CodeCommit，以及從 IDE 中處理源代碼更改。

Amazon DynamoDB

DynamoDB 是一種快速、可高度擴展、高可用性、符合成本效益的非關聯式資料庫服務。適用於 Visual Studio 的工具組提供了在開發環境中使用 Amazon DynamoDB 的功能。使用適用 Toolkit for Visual Studio，您可以在 DynamoDB 資料表中建立和編輯屬性，以及在資料表上執行掃描作業。

Amazon S3

您可以透過拖放或從 Amazon S3 下載內容，快速輕鬆地將內容上傳到 Amazon S3 儲存貯體。您也可以方便地在值區中的物件上設定權限、中繼資料和標記。

Amazon RDS

AWS 資源管理器可以幫助您在視覺工作室中創建和管理 Amazon RDS 資產。使用 Microsoft SQL 伺服器的 Amazon RDS 執行個體也可以新增至伺服器資源管理器。

AWS Elastic Beanstalk

您可以使用 Elastic Beanstalk 將 .NET Web 應用程式專案部署到. AWS您可以將應用程式部署到單一執行個體環境，或從 IDE 內部部署到完全負載平衡、自動調整規模的環境。您也可以快速方便地部署應用程式的新版本，而無需離開 Visual Studio。如果您的應用程式使用 Amazon RDS 中的 SQL Server，部署精靈也可以設定 Elastic Beanstalk 中的應用程式環境與 Amazon RDS 中的資料庫執行個

體之間的連線。適用於 Visual Studio 的工具組也包含獨立的命令列部署工具。使用部署工具讓部署成為建置程序的自動部分，或在 Visual Studio 以外的其他指令碼案例中包含部署。

AWS CloudFormation

您可以使用 Toolkit for Visual Studio 來編輯 AWS CloudFormation JSON 格式範本，並支援編輯器 IntelliSense 和語法醒目提示。使用 AWS CloudFormation 範本，您可以描述要實例化以託管應用程式的資源。然後，您可以在 IDE 中部署範本 AWS CloudFormation。範本中描述的資源是為您佈建的，讓您可以專注於開發應用程式的功能。

AWS Identity and Access Management (IAM)

您可以在 AWS Explorer 中建立 IAM 使用者、角色和政策，並將政策附加到使用者。

相關資訊

若要開啟問題或檢視目前未解決的問題，請造訪 https://github.com/aws/ aws-toolkit-visual-studio /問題。

要了解有關視覺工作室的更多信息，請訪問 https://visualstudio.microsoft.com/vs/。

Amazon Q 和 Amazon CodeWhisperer

什麼是 Amazon Q

從 2024 年 4 月 30 日起，Amazon 現已成 CodeWhisperer 為 Amazon Q 開發人員的一員，其中包括內嵌程式碼建議和安全掃描。

若要進一步了解與中的 Amazon Q 開發人員合作 AWS Toolkit for Visual Studio，請參閱 Amazon Q 開發人員使用者指南中的 IDE 中的 Amazon Q 開發人員主題。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

您可以下載、安裝和設定 Toolkit for Visual Studio，透過 IDE 中的視覺工作室 Marketplace。如需詳細指示，請參閱本使用者指南的入門主題中的〈安裝 Visual Studio AWS 工具組〉一節。

從視覺工作室 Marketplace 下載工具包

瀏覽至網頁瀏覽器中的下載網站，以下載 AWS Visual Studio 安裝檔案的工具組。

來自的其他 IDE 工具包 AWS

除了工具組之外，AWS還提供適用於 VS 程式碼和 JetBrains.

AWS Toolkit for Visual Studio Code連結

- 請按照此鏈接AWS Toolkit for Visual Studio Code從 VS 代碼 Marketplace 下載。
- 若要進一步瞭解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

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

開始使用

可AWS Toolkit for Visual Studio讓您從 Visual Studio 整合式開發環境 (IDE) 取得您的AWS服務和資源。

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

安裝與設定 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

必要條件

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

- 視覺工作室 19 或更高版本
- 視窗 10 或更新版本的視窗
- 管理員存取視窗和視覺工作室
- 有效 AWS IAM 登入資料

> **Note**
>
> 不支援 AWS Toolkit for Visual Studio 的版本可供視覺工作室使用。若要下載不受支援的版本，請瀏覽至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，請造訪主AWS 控台閘道。

安裝 AWS Toolkit for Visual Studio

若要安裝 AWS Toolkit for Visual Studio，請從下列程序中尋找您的 Visual Studio 版本，並完成必要的步驟。所有版本的下載連結都 AWS Toolkit for Visual Studio 可以在AWS Toolkit for Visual Studio登陸頁面中找到。

> **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 適用於視覺工作室 2022

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

1. 從主功能表導覽至擴充功能，然後選擇管理擴充功能。
2. 在搜尋方塊中搜尋AWS。
3. 選擇相關版本的下載按鈕，然後按照安裝提示進行操作。

   > **Note**
   >
   > 您可能需要手動關閉並重新啟動 Visual Studio，才能完成安裝程序。

4. 當下載和安裝完成時，您可以從 [檢視] 功能表中 AWS Toolkit for Visual Studio 選擇 AWS [檔案總管] 來開啟。

安裝 AWS Toolkit for Visual Studio 適用於視覺工作室 2019

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

1. 從主功能表導覽至擴充功能，然後選擇管理擴充功能。
2. 在搜尋方塊中搜尋AWS。
3. 選擇下載按鈕，然後按照提示進行操作。

> **Note**
>
> 您可能需要手動關閉並重新啟動 Visual Studio，才能完成安裝程序。

4. 當下載和安裝完成時，您可以從 [檢視] 功能表中 AWS Toolkit for Visual Studio 選擇 AWS [檔案總管] 來開啟。

解除安裝 AWS Toolkit for Visual Studio

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

卸載視 AWS Toolkit for Visual Studio 覺工作室 2022

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

1. 從主功能表導覽至擴充功能，然後選擇管理擴充功能。
2. 從「管理擴充功能」導覽功能表，展開「已安裝」標題。
3. 找到 AWS Toolkit for Visual Studio 2022 擴展程序，然後選擇卸載按鈕。

> **Note**
>
> 如 AWS Toolkit for Visual Studio 果在導覽功能表的 [已安裝] 區段中看不到，您可能需要重新啟動 Visual Studio。

4. 按照屏幕上的提示完成卸載過程。

卸載 AWS Toolkit for Visual Studio 適用於視覺工作室

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

1. 從主功能表導覽至「工具」，然後選擇「管理擴充功能」。
2. 從「管理擴充功能」導覽功能表，展開「已安裝」標題。

3. 找到 AWS Toolkit for Visual Studio 2019 擴展程序，然後選擇卸載按鈕。
4. 按照屏幕上的提示完成卸載過程。

卸載視 AWS Toolkit for Visual Studio 覺工作室

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

1. 從主功能表導覽至 [工具]，然後選擇 [擴充功能和更新]。
2. 從「擴充功能和更新」導覽功能表中，展開「已安裝」標題。
3. 找到 AWS Toolkit for Visual Studio 2017 擴展程序，然後選擇卸載按鈕。
4. 按照屏幕上的提示完成卸載過程。

卸載 AWS Toolkit for Visual Studio 適用於視覺工作室

若要解除安裝 AWS Toolkit for Visual Studio 2013 或 2015 年，請完成以下步驟：

1. 在 Windows 控制台中，開啟「程式和功能」。

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

2. 從已安裝的程式清單中，開啟 (以滑鼠右鍵按一下) Windows AWS 工具的內容功能表。
3. 選擇「解除安裝」，然後按照提示完成解除安裝程序。

> **Note**
>
> 您的 Samples 目錄在解除安裝過程中不會被刪除。如果您已修改樣本，則會保留此目錄。必須手動移除此目錄。

連接到 AWS

大多數 Amazon Web Services（AWS）服務和資源都是通過一個 AWS 帳戶進行管理。不需要 AWS 帳戶即可使用 AWS Toolkit for Visual Studio，但是 Toolkit 功能在沒有連接的情況下受到限制。

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

必要條件

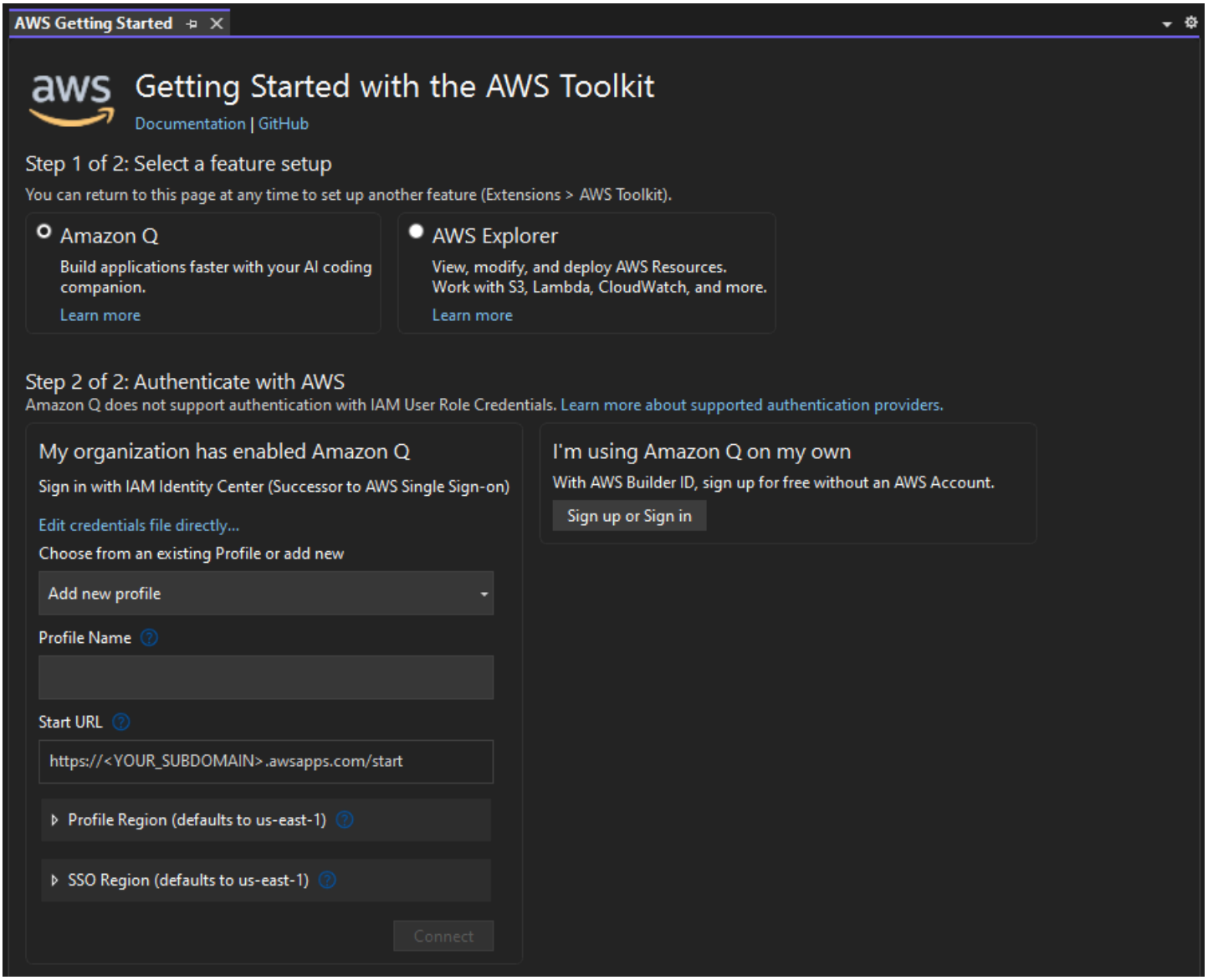
如果您是新手 AWS 或尚未建立帳戶，則有 3 個主要步驟可將 Visual Studio 的工具組與您的 AWS 帳戶連線：

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

AWS 從工具組連線到

若要從 Visual Studio 的工具組連線到您的 AWS 帳戶，請完成下列程序，開啟 AWS 工具組使用者介面入門 (連線 UI)。

1. 從 Visual Studio 主功能表中，展開擴充功能，然後展開AWS 工具組。
2. 從 [AWS 工具組] 功能表選項中選擇 [入門]。
3. 開始使用 AWS 工具組連線使用者介面會在 Visual Studio 中開啟。



下表說明哪些驗證方法與每個功能相容。若要深入了解 3 種驗證方法 AWS IAM Identity Center、AWS Identity and Access Management 認證和 AWS Builder ID，請參閱本使用者指南中的「驗證和存取」目錄。

> **Note**
>
> 目前，在 Visual Studio 的工具包 CodeCatalyst 中使用時，您只需要在克隆第三方存儲庫時使用 AWS 構建器 ID 進行授權。

Amazon Q 開發	AWS 瀏覽器	Amazon CodeCatalyst

☑ AWS 生成器 ID	☒ AWS 生成器 ID	☑ AWS 生成器 ID
☑ 身分識別中心	☑ 身分識別中心	☒ 身分識別中心
☒ AWS IAM 登入資料	☑ AWS IAM 登入資料	☒ AWS IAM 登入資料

Amazon Q 開發人員的身份驗證

若要開始使用 Amazon Q 開發人員，請使用您的 AWS IAM Identity Center 或 AWS 產生器 ID 登入資料進行身份驗證並連接。

下列程序說明如何驗證 Toolkit 並將其與您的 AWS 帳戶連線。

使用 IAM 身分中心進行驗證和連線

1. 從 AWS 工具組連線入門使用者介面中，選取 Amazon Q 開發人員徑向以擴展 Amazon Q 開發人員身份驗證選項。

   > **Note**
   >
   > 如果沒有預存的登入資料，請繼續執行步驟 3 以新增或更新您的 IAM 身分中心登入資料。

2. 在「我的組織已啟用 Amazon Q Developer」區段中，展開「從現有設定檔中選擇」或「新增新」下拉式功能表，以從儲存的登入資料清單中進行選擇。
3. 從「設定檔類型」下拉式功能表中選擇 AWS IAM Identity Center
4. 在「設定檔名稱」文字欄位中，輸入您要驗證**`Profile Name`**的 IAM 身分中心設定檔。
5. 在「開始 URL」文字欄位中，輸入**`Start URL`**附加至 IAM 身分中心登入資料的資訊。
6. 從「設定檔區域」(預設為 us-east-1) 下拉式功能表中，選擇您要驗證的 IAM 身分中心使用者設定檔所定義的設定檔區域。
7. 從 SSO 區域 (預設為 us-east-1) 下拉式功能表中，選擇 IAM 身分中心登入資料所定義的 SSO 區域，然後選擇「Connect」按鈕以開啟「 AWS 使用 IAM 身分中心登入」對話方塊。
8. 從「 AWS 使用 IAM 身分中心登入」對話方塊中，選擇「繼續瀏覽器」按鈕，以在預設 Web 瀏覽器中開啟「 AWS 授權請求」網站。
9. 確認 IDE 中的安全代碼與 Web 瀏覽器中顯示的「 AWS 授權請求確認碼」相符，然後選擇「提交並繼續」按鈕以繼續。

10. 依照預設網頁瀏覽器中的提示進行操作，授權程序完成時會收到通知，您可以安全地關閉瀏覽器，然後返回 Visual Studio。

使用 AWS 生成器 ID 進行身份驗證並連接

1. 從 AWS 工具組連線入門使用者介面中，選取 Amazon Q 開發人員徑向以擴展 Amazon Q 開發人員身份驗證選項。
2. 從 [我自己使用 Amazon Q 開發人員] 區段中，選擇 [註冊] 或 [登入] 按鈕以開啟 [使用 AWS 產生器 ID 登入] 對話方塊。
3. 選擇「繼續瀏覽器」按鈕，在預設的網頁瀏覽器中開啟「 AWS 授權」請求網站。
4. 確認 IDE 中的安全代碼與 Web 瀏覽器中顯示的「 AWS 授權請求確認碼」相符，然後選擇「提交並繼續」按鈕以繼續。
5. 依照預設網頁瀏覽器中的提示進行操作，授權程序完成時會收到通知，您可以安全地關閉瀏覽器，然後返回 Visual Studio。

AWS 資源管理器的驗證

若要從工具組開始使用 AWS 檔案總管，請使用您的 IAM 身分中心登入資料或 IAM 登入資料進行驗證並連線。

下列程序說明如何驗證 Toolkit 並將其與您的 AWS 帳戶連線。

使用 IAM 身分中心進行驗證和連線

1. 從「 AWS 工具組入門」連線使用者介面中，選取AWS 資源管理器徑向以擴展 Amazon Q 開發人員身份驗證選項。
2. 從下**`Profile Type`**拉式功能表中選擇AWS IAM Identity Center。
3. 在「設定檔名稱」文字欄位中，輸入您要使用**`Profile Name`**的 IAM 身分中心設定檔。
4. 在「開始 URL」文字欄位中，輸入**`Start URL`**附加至 IAM 身分中心登入資料的資訊。
5. 從「設定檔區域」(預設為 us-east-1) 下拉式功能表中，選擇您要驗證的 IAM 身分中心使用者設定檔所定義的設定檔區域。
6. 從 SSO 區域 (預設為 us-east-1) 下拉式功能表中，選擇您的 IAM 身分中心登入資料所定義的 SSO 區域。
7. 選擇 [繼續瀏覽器] 按鈕，在預設的網頁瀏覽器中開啟 [AWS 授權要求] 網站。

8. 確認 IDE 中的安全代碼與 Web 瀏覽器中顯示的「AWS 授權請求確認碼」相符，然後選擇「提交並繼續」按鈕以繼續。
9. 依照預設網頁瀏覽器中的提示進行操作，授權程序完成時會收到通知，您可以安全地關閉瀏覽器，然後返回 Visual Studio。

使用 IAM 登入資料進行身份驗證

1. 從「 AWS 工具組入門」連線使用者介面中，選取AWS 資源管理器徑向以擴展 Amazon Q 開發人員身份驗證選項。
2. 從下**Profile Type**拉式功能表中選擇「IAM 使用者角色」。
3. 在「設定檔名稱」文字欄位中**Profile Name**，輸入您要驗證的設定檔。
4. 在「存取金鑰 ID」文字欄位中**Access Key ID**，輸入您要驗證的設定檔。
5. 在「密碼金鑰」文字欄位中**Secret Key**，輸入您要驗證的設定檔。
6. 從儲存位置 (預設為共用認證檔案) 下拉式功能表中，指定您要將認證與共用認證檔案一起儲存，還是要將身分證明與 .NET 加密儲存一起儲存。
7. 從「設定檔區域」(預設為 us-east-1) 下拉式功能表中，選擇附加至您要驗證之設定檔的設定檔區域。

疑難排解安裝問題 AWS Toolkit for Visual Studio

下列資訊可解決安裝時的常見安裝問題AWS Toolkit for Visual Studio。

如果您在安裝時遇到錯誤，AWS Toolkit for Visual Studio或者不清楚安裝是否完成，請檢閱以下各節中的資訊。

系統管理員權限

AWS Toolkit for Visual Studio擴充功能需要管理員權限，才能確保所有AWS服務和功能均可存取。

如果您擁有本機系統管理員權限，系統管理員權限可能不會直接延伸到 Visual Studio 執行個體。

若要以系統管理員權限在本機啟動 Visual Studio：

1. 在視窗中，找出應用程式啟動器 (圖示)。
2. 開啟內容功能表 (按一下滑鼠右鍵) Visual Studio 圖示，以開啟內容功能表。
3. 從內容功能表中選取「以管理員身分執行」

若要以系統管理員權限從遠端啟動 Visual Studio：

1. 在 Windows 中，找出您用來連線至 Visual Studio 遠端執行個體之應用程式的應用程式啟動器。
2. 開啟 (按一下滑鼠右鍵) 應用程式的關聯式功能表，以開啟關聯式功能表。
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 都可能會提示您確認系統管理認證。

取得安裝記錄

如果您已完成上述之前 [系統管理員權限] 區段中的步驟，且已確認您正在以系統管理員權限執行或連線至 Visual Studio，則取得安裝記錄檔可協助診斷其他問題。

若要AWS Toolkit for Visual Studio從`.vsix`檔案手動安裝並產生安裝記錄檔，請完成以下步驟。

1. 在AWS Toolkit for Visual Studio登陸頁面中，依照「下載」連結儲存您要安裝的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] 子功能表，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具有下列格式的`vsixinstaller`命令：

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

4. 以您要在其中建立安裝記錄檔之目錄的檔案名稱和完整檔案路徑取`[file path to log file]`代。具有指定檔案路徑和檔案名稱的`vsixinstaller`命令範例如下所示：

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

5. 取`[file path to Toolkit installation file]`代為所在目錄的完整檔案路徑。`AWSToolkitPackage.vsix`

   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
   ```

6. 檢查以確保您的檔案名稱和路徑正確無誤，然後執行vsixinstaller指令。

   完整vsixinstaller指令的範例如下所示：

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

安裝不同的擴展

如果您已取得安裝記錄檔，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 擴充功能。安裝不同的 Visual Studio 擴充功能可提供您安裝問題的其他洞察力。如果您無法安裝任何 Visual Studio 擴充功能，可能需要疑難排解 Visual Studio 的問題，而不是AWS Toolkit for Visual Studio。

聯絡支援

如果您已檢閱本指南中包含的所有章節，並且需要其他資源或支援，則可以從 AWS Toolkit for Visual StudioGithub 問題網站檢視過去的問題或開啟新問題。

若要協助加速解決您的問題：

- 檢查過去和當前的問題，看看其他人是否遇到類似的情況。
- 請詳細記錄您為解決問題所採取的每個步驟。
- 儲存您從安裝或其他擴充功能取得的AWS Toolkit for Visual Studio任何記錄檔。
- 將您的AWS Toolkit for Visual Studio安裝記錄檔附加至新問題。

配置文件和窗口綁定

Toolkit. Visual Studio 來 Visual Studio 來 Visual Studio 來 Visual Studio 來 Visual Studio 來 Visual Studio

使用適用於 Visual Studio 之工具組的發行工具、精靈和其他功能時，請注意下列事項：

- 「檔案AWS總管」視窗會一次繫結至單一設定檔和區域。從檔案AWS總管開啟的視窗預設為該繫結設定檔和區域。
- 開啟新視窗後，您可以使用AWS Explorer 的該執行個體切換至不同的設定檔或區域。
- Visual Studio 發行工具組和功能的工具組會自動預設為在檔案AWS總管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將繼續使用新的設定檔和區域設定。
- 如果您開啟了多個 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。
- 檔案AWS總管會儲存指定的最後一個設定檔和區域，最後關閉的 Visual Studio 執行個體會保留其值。

身份驗證和訪問

您不需要驗證，即可 AWS 開始使用適用 AWS Toolkit for Visual Studio。但是，大多數 AWS 資源都是通過 AWS 帳戶管理的。若要存取 Visual Studio 服務和功能的所有 AWS 工具組，您至少需要 2 種類型的帳戶驗證：

1. 為您的 AWS 帳戶提供 AWS Identity and Access Management (IAM) 或AWS IAM Identity Center身份驗證。大部分的 AWS 服務和資源都是透過 IAM 和 IAM 身分中心進行管理。
2. 對於某些其他 AWS 服務，AWS 生成器 ID 是可選的。

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 IAM 身分中心登入資料 AWS Toolkit for Visual Studio
- AWS IAM 登入資料
- AWS 產生器 ID
- 工具包中的多因素身份驗證（MFA）
- 設定外部認證

AWS IAM 身分中心登入資料 AWS Toolkit for Visual Studio

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

如需有關如何設定軟體開發套件 (SDK) 的 IAM 身分中心的詳細指示 AWS Toolkit for Visual Studio，請參閱 SDK 和工具參考指南的 IAM 身分中心身分驗證一節。AWS

使用 IAM 身分中心進行驗證 AWS Toolkit for Visual Studio

若要透過將 IAM 身分中心設定檔新 AWS Toolkit for Visual Studio 增至您的`credentials`或`config`檔案，使用 IAM 身分中心進行驗證，請完成以下步驟。

1. 在偏好的文字編輯器中，開啟儲存在`<hone-directory>\.aws\credentials`檔案中的 AWS 認證資訊。
2. `credentials file`在該區段下方`[default]`，為具名的 IAM 身分中心設定檔新增範本。以下是範例範本：

> **Important**
>
> 在檔案中建立項目時，請勿使用「設定credential檔」一詞，因為會與credential檔案命名慣例產生衝突。
> profile_只有在檔案中配置具名的設定config檔時，才包含前置字。

```
[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**：指向組織的 IAM 身分中心使用者入口網站的 URL。
- **sso_region**：包含您的 IAM 身分中心入口網站主機的 AWS 區域。這可能與稍後在預設region參數中指定的「 AWS 區域」不同。
- **sso_account_id**：包含 IAM 角色的 AWS 帳戶 ID，該角色具有您要授與此 IAM 身分中心使用者的權限。
- **sso_role_name**: 使用此設定檔透過 IAM 身分中心取得登入資料時，定義使用者許可的 IAM 角色名稱。
- **region**: 此 IAM 身分中心使用者登入的預設 AWS 區域。

> **Note**
>
> 您也可以 AWS CLI 透過執行aws configure sso命令，將已啟用 IAM 身分中心的設定檔新增至您的。執行此命令之後，您可以為代管 IAM 身分中心目錄的 IAM 身分中心起始 URL (sso_start_urlregion) 和 AWS 區域 () 提供值。
> 如需詳細資訊，請參閱《使用指南》中的 < 設定 AWS CLI 以使用 AWS 單一登入 > 。AWS Command Line Interface

使用 IAM 身分中心登入

使用 IAM 身分中心設`sso_start_url`定檔登入時，預設瀏覽器會啟動至`credential file`. 您必須先驗證您的 IAM 身分中心登入資源，然後才能存取中的 AWS 資源 AWS Toolkit for Visual Studio。如果您的憑據過期，則必須重複連接過程才能獲取新的臨時憑據。

AWS IAM 登入資料

AWS IAM 登入資料會透過本機儲存的存取金鑰，使用您的 AWS 帳戶進

以下各節說明如何設定 IAM 登入資料，以便從中使用您的 AWS 帳戶進行驗證 AWS Toolkit for Visual Studio。

> **Important**
>
> 在設定 IAM 登入資料以使用您的 AWS 帳戶進行驗證之前，請注意：
>
> - 如果您已透過其他 AWS 服務 (例如 AWS AWS CLI) 設定 IAM 登入資料，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 驗證。如需 AWS IAM 最佳做法的其他資訊，請參閱身分與存取管理使用者指南中AWS IAM 中的安全性最佳做法一節。
> - 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。請改為使用身分識別提供者的同盟，例如 AWS IAM Identity Center. 如需詳細資訊，請參閱什麼是 IAM 身分中心？ 在《AWS IAM Identity Center 使用者指南》中。

建立 IAM 使用者

您必須先完成 AWS SDK 和工具參考指南中的「 AWS Toolkit for Visual Studio 使用長期登入資料進行驗證」主題中的「步驟 1：建立 IAM 使用者」和「步驟 2：取得您的存取金鑰」，才能使用您的 AWS 帳戶進行驗證。

> **Note**
>
> 步驟 3：更新共享憑據是可選的。
> 如果您完成步驟 3， AWS Toolkit for Visual Studio 系統會自動從偵測您的認證`credentials file`。

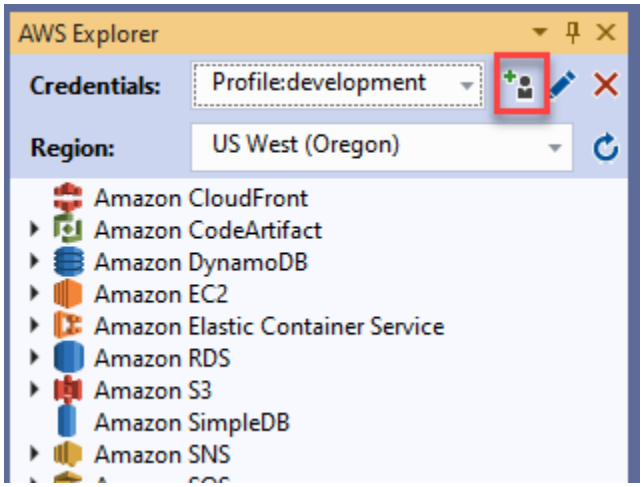
如果您尚未完成步驟 3，則會逐 AWS Toolkit for Visual Studio 步引導您完成建立程序，`credentials file`如下方的〈從〉一 AWS Toolkit for Visual Studio節建立認證檔案中所述。

建立認證檔

若要將使用者新增至或`credentials file`從中建立使用者 AWS Toolkit for Visual Studio：

> **Note**
>
> 從工具包添加新的用戶配置文件時：
>
> - 如果`credentials file`已存在，則會將新使用者資訊加入至既有檔案。
> - 如果`credentials file`不存在，則創建一個新文件。

1. 從 AWS 資源管理器中選擇新帳戶配置文件圖標以打開新帳戶配置文件對話框。



2. 完成 [新帳戶設定檔] 對話方塊中的必填欄位，然後選擇 [確定] 按鈕以建立 IAM 使用者。

從工具組編輯 IAM 使用者登入資料

若要從工具組編輯 IAM 使用者登入資料，請完成以下步驟：

1. 從 AWS 檔案總管的認證下拉式清單中，選擇要編輯的 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. 在「編輯設定檔」對話方塊中完成更新，然後選擇「確定」按鈕以儲存變更。

若要從工具組刪除 IAM 使用者登入資料，請完成以下步驟：

1. 從 AWS 檔案總管的認證下拉式清單中，選擇要刪除的 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. 確認您要刪除設定檔，以便將其從您的`Credentials file`.

> **Important**
>
> 無法從中編輯支援進階存取功能的設定檔，例如 IAM 身分中心或 [編輯設定檔] 對話方塊中的多因素驗證 (MFA)。 AWS Toolkit for Visual Studio若要變更這些類型的縱斷面，您必須`credentials file`使用文字編輯器來編輯。

從文字編輯器編輯 IAM 使用者登入資料

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。視窗`credential file`中的預設位置為`C:\Users\`*`USERNAME`*`\.aws\credentials`。

有關的位置和結構的更多詳細信息`credential files`，請參閱 AWS SDK 和工具參考指南的「共享配置和憑據文件」部分。

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

您可以使用指令在中建立 IAM 使用者的`credentials file`另一個工具`aws configure`。 AWS CLI

如需建立 IAM 使用者的詳細資訊， AWS CLI 請參閱使用AWS CLI 者指南中的〈設定〉 AWS CLI主題。

適用於 Visual Studio 的工具組支援下列組態屬性：

```
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 產生器 ID

AWS 產生器 ID 是一種額外的 AWS 身份驗證方法，可能需要使用某些服務或功能，例如透過 Amazon 複製第三方儲存庫 CodeCatalyst。

有關 AWS Builder ID 驗證方法的詳細資訊，請參閱AWS 登入使用者指南中的使用 AWS Builder ID 登入主題。

如需複製儲存庫的其他資訊 AWS Toolkit for Visual Studio，請參閱本使用者指南中的「使用 Amazon」 CodeCatalyst 主題。 CodeCatalyst

工具包中的多因素身份驗證（MFA）

多重要素驗證 (MFA) 可為您的帳戶提供額外的 AWS 安全性。MFA 要求使用者在存取 AWS 網站或服務時，透過 AWS 支援的 MFA 機制提供登入認證和唯一驗證。

AWS 支援一系列虛擬裝置和硬體裝置進行 MFA 驗證。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 AWS的使用多重要素驗證 (MFA)。

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角色分隔以指派權限給 IAM 使用者。如需角色分隔的詳細資訊，請參閱使用指南中的建立角色以將許可委派給 IAM 使用AWS Identity and Access Management 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 [其他 AWS 帳戶]。
4. 輸入您所需的帳戶 ID，然後標記 [需要 MFA] 核取方塊。

> **Note**
>
> 若要尋找您的 12 位數帳號 (ID)，請前往主控台的導覽列，然後選擇 [支援]、[Support Support 中心]。

5. 選擇下一步：許可。
6. 將現有原則附加至您的角色，或為其建立新原則。您在此頁面上選擇的政策決定 IAM 使用者可以透過 Toolkit 存取哪些 AWS 服務。
7. 附加政策後，請選擇「下一步：標籤」以選擇將 IAM 標籤新增至您的角色。然後選擇「下一步：查看」以繼續。
8. 在「複查」頁面中，輸入必要的「角色」名稱 (例如，工具組角色)。您也可以新增選擇性的「角色」描述。
9. 選擇建立角色。
10. 顯示確認訊息時 (例如「角色工具組-角色已建立」)，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 使用者擔任該角色時，您需要此 ARN。)

步驟 2：建立具有角色許可的 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. 選擇導覽列中的 [使用者]，然後選擇 [新增使用者]。
3. 在 [新增使用者] 頁面中，輸入必要的使用者名稱 (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」：「權限」、「下一步」：「標籤」和「下一步」：「檢閱」可在下一頁中移 您不會在此階段新增權限，因為使用者會承擔角色的權限。
5. 在「檢閱」頁面中，系統會通知您此使用者沒有權限。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。(在認證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 使用者擔任該角色

下列程序會建立內嵌原則，以允許使用者擔任角色 (以及該角色的權限)。

1. 在 IAM 主控台的「使用者」頁面中，選擇剛建立的 IAM 使用者 (例如，工具組使用者)。
2. 在 [摘要] 頁面的 [權限] 索引標籤中，選擇 [新增內嵌原則]。
3. 在 [建立原則] 頁面中，選擇 [選擇服務]，在 [尋找服務] 中輸入 ST S，然後從結果中選擇 STS。
4. 對於 「動作」，開始輸入術語AssumeRole。AssumeRole勾選出現的核取方塊。
5. 在「資源」段落中，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] 對話方塊中，針對 [指定角色的 ARN]，新增您在步驟 1 中建立之角色的 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 [建立原則] 頁面，選擇 [指定要求條件 (選用)]，標示 [需要 MFA] 核取方塊，然後選擇 [關閉] 以確認。
9. 選擇 Review policy (檢閱政策)
10. 在 [檢閱原則] 頁面中，輸入原則的 [名稱]，然後選擇 [建立原則]。

    「權限」索引標籤會顯示直接附加至 IAM 使用者的新內嵌政策。

步驟 4：為 IAM 使用者管理虛擬 MFA 裝置

1. 下載虛擬 MFA 應用程序並將其安裝到您的智能手機。

   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，從導覽列選擇 [使用者]，然後選擇擔任角色的使用者 (在本例中為工具組使用者)。
3. 在 [摘要] 頁面中，選擇 [安全性認證] 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]，然後使用智慧型手機上安裝的虛擬 MFA 應用程式掃描代碼。
6. 掃描 QR 碼後，虛擬 MFA 應用程式會產生一次性 MFA 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 [安全認證] 索引標籤中，複製新指派的 MFA 裝置的 ARN。

   ARN 包括您的 12 位數帳戶 ID，其格式類似於以下內容：`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

下列程序會在從 Visual Studio 的工具組存取 AWS 服務時，建立允許 MFA 的設定檔。

您建立的設定檔包括您在上一個步驟中複製並儲存的三項資訊：

- IAM 使用者的存取金鑰 (存取金鑰 ID 和秘密存取金鑰)
- 將許可委派給 IAM 使用者的角色 ARN
- 指派給 IAM 使用者的虛擬 MFA 裝置的 ARN

在包含您 AWS 認證的 AWS 共用認證檔案或 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
```

在提供的範例中定義了兩個設定檔：

- `[toolkit-user]`設定檔包括您在步驟 2 中建立 IAM 使用者時產生並儲存的存取金鑰和秘密存取金鑰。
- `[mfa]`設定檔定義支援多重要素驗證的方式。有三個項目：

  ◦`source_profile`：指定其認證用來擔任此設定檔中此`role_arn`設定所指定角色的設定檔。在這種情況下，它是設`toolkit-user`定檔。

  ◦`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 資源名稱 (ARN)。在這種情況下，它是您在步驟 1 中建立的角色的 ARN。

- `mfa_serial`：指定使用者擔任角色時必須使用的 MFA 裝置識別碼或序號。在這種情況下，它是您在步驟 3 中設置的虛擬設備的 ARN。

設定外部認證

如果您有產生或查詢不直接支援的認證的方法 AWS，您可以將包含該設定的`credential_process`設定檔新增至共用認證檔案。此設定會指定執行以產生或擷取要使用的驗證認證的外部命令。例如，您可以在檔案中包含類似下列項目的項`config`目：

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

如需有關使用外部認證及相關安全風險的詳細資訊，請參閱《使用AWS Command Line Interface 者指南》中的使用外部處理程序的 Sourcing 認證。

使用 AWS 服務

下列主題說明如何開始使用 Visual Studio 工 AWS 具組中的 AWS 服務。

主題

亞馬遜CodeCatalyst的視覺AWS工作室工具包

什麼是 Amazon CodeCatalyst？

Amazon CodeCatalyst 是適用於軟體開發團隊的雲端協作空間。使用適用於 Visual Studio 的AWS工具組，您可以直接從AWS工具組檢視和管理CodeCatalyst資源。如需有關的詳細資訊CodeCatalyst，請參閱 Amazon CodeCatalyst 使用者指南。

下列主題說明如何連接 Visual Studio 的工AWS具組，以CodeCatalyst及如何透CodeCatalyst過 Visual Studio 的工AWS具組來使用。

主題

開始使用亞馬遜CodeCatalyst和視覺AWS工作室的工具包

若要CodeCatalyst從 Visual Studio 的工AWS具組開始使用亞馬遜，請完成以下操作。

主題

安裝AWS工具包

在您整合 Visual Studio 的AWS工具組與您的CodeCatalyst帳戶之前，請確定您使用的是最新版本的 Visual Studio AWS 工具組。如需有關如何安裝和設定 Visual Studio AWS 工具組的最新版本的詳細資訊，請參閱本使用者指南的 < 設定 Visual Studio 的AWS工具組 > 一節。

創建CodeCatalyst帳戶和AWS生成器 ID

除了安裝最新版本的 Visual Studio 的AWS工具組之外，您必須擁有使用中的AWS產生器識別碼和CodeCatalyst帳戶，才能與 Visual Studio 的AWS工具組連線。如果您沒有有效的 AWS Builder ID 或CodeCatalyst帳戶，請參閱CodeCatalyst使用者指南中的「設定方式」一CodeCatalyst節。

> **Note**
>
> AWS產生器 ID 與您的AWS認證不同。如需有關如何使用 AWS Builder ID 註冊和驗證的指示，請參閱本使用者指南中的驗證和存取：AWS產生器 ID 主題。
> 如需有關AWS建置器 ID 的詳細資訊，請參閱《AWS一般參考使用指南》中的「AWS產生器 ID」主題。

連接AWS工具包 CodeCatalyst

若要將 Visual Studio 的AWS工具組與您的CodeCatalyst帳戶連線，請完成下列步驟。

1. 從 Git 的功能表項目中，選擇複製儲存庫... 。
2. 從「瀏覽儲存庫」區段中，選取 Amazon CodeCatalyst 作為供應商。
3. 在「連線」區段中，選擇「使用AWS產生器 ID 連線」，在偏好的網頁瀏覽器中開啟CodeCatalyst主控台。
4. 在瀏覽器中，在提供的欄位中輸入您的 AWS Builder ID，然後按照指示繼續操作。
5. 出現提示時，選擇 [允許] 以確認 Visual Studio 的AWS工具組與您的CodeCatalyst帳戶之間的連線。連線程序完成後，CodeCatalyst會顯示確認訊息，指出您可以安全地關閉瀏覽器。

使用亞馬遜CodeCatalyst資源從視覺工作室的工AWS具包

以下各節提供適用於 Visual Studio AWS 工具組的亞馬遜CodeCatalyst資源管理功能的概觀。

主題

- 複製儲存庫

複製儲存庫

CodeCatalyst是一種基於雲的服務，需要您連接到雲才能處理CodeCatalyst項目。若要在本機上處理專案，您可以將CodeCatalyst存放庫複製到本機電腦，並在下次連線到雲端時與CodeCatalyst專案同步。

若要將存放庫複製到本機電腦，請完成以下步驟。

1. 從 Git 的功能表項目中，選擇複製儲存庫... 。
2. 從「瀏覽儲存庫」區段中，選取 Amazon CodeCatalyst 作為供應商。

> **Note**
>
> 如果「連線」區段顯示Not Connected訊息，請先完成本使用手冊中「驗證與存取：AWSBuilder ID」一節中的步驟，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目標資料夾。

> **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區域。
7. 存儲庫被克隆後，Visual Studio 將加載您克隆的解決方案

> **Note**
>
> 如果 Visual Studio 未在複製的存放庫中開啟解決方案，您的 Visual Studio 選項可以從 [原始檔控制] 功能表的 [Git 全域設定] 中的 [開啟 Git 存放庫時自動載入解決方案] 中調整。

疑難排解

以下是CodeCatalyst從 Visual Studio 的工AWS具組使用 Amazon 時解決已知問題的疑難排解主題。

主題

- 憑證

憑證

如果您在嘗試從中克隆基於 git 的存儲庫時遇到一個對話框詢問憑據CodeCatalyst，則可能會全局配置您的AWSCodeCommit憑據幫助程序，從而導致干擾。CodeCatalyst如需有關AWSCodeCommit認證協助程式的其他資訊，請參閱AWSCodeCommit使用指南中的 < 使用 AWS CLI 認證協助程式在 Windows 上設定 HTTPS 連線到AWSCodeCommit儲存庫的步驟 > 一節。

若要限制AWSCodeCommit認證協助程式僅處理 CodeCommit URL，請完成以下步驟。

1. 在以下位置打開全局 git 配置文件：`%userprofile%\.gitconfig`
2. 在檔案中找到下列區段：

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

3. 將該部分更改為以下內容：

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

4. 儲存您的變更，然後完成複製儲存庫的步驟。

亞馬遜 CloudWatch Logs 整合 Visual Studio

Amazon CloudWatch Logs 整合AWSToolkit for Visual Studio，讓您能夠監視、儲存和存取 CloudWatch 記錄資源，而不必離開您的 IDE。進一步了解如何設定 CloudWatch 服務以及如何使用 CloudWatch 記錄檔功能，請從下列主題中選擇。

主題

- 設定 CloudWatch Logs 整合
- 使用 CloudWatch Visual Studio 中的日誌

設定 CloudWatch Logs 整合

在您可以使用亞馬遜之前 CloudWatch LoToolkit for Visual Studio 整合記錄，您需要AWS帳戶。您可以建立新的AWS來自的帳戶AWS登入網站。大部分 CloudWatch 可透過使用作用中存取 Toolkit for Visual Studio 中的記錄檔功能AWS登入資料。如果特定功能需要其他配置，則需求包含在使用 CloudWatch 日誌指南。

有關設置的其他信息和選項 CloudWatch 記錄檔，請參閱開始設定亞馬遜部分 CloudWatch 日誌指南。

使用 CloudWatch Visual Studio 中的日誌

亞馬遜 CloudWatch Logs 整合可讓您監控、存放和存取 CloudWatch 來自AWSToolkit for Visual Studio。有權訪問 CloudWatch 記錄功能 (不需要離開您的 IDE) 藉由簡化 CloudWatch 記錄開發流程並減少工作流程的中斷情況。下列主題說明如何使用的基本特徵和函數 CloudWatch 日誌整合。

主題

- CloudWatch 日誌群組
- CloudWatch 日誌串流

- CloudWatch 日誌事件
- 額外的訪問權限 CloudWatch日誌

CloudWatch 日誌群組

一個log group是一組log streams共用相同保留、監控和存取控制設定。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

所以此View Log Groups功能顯示的日誌群組清單 CloudWatch日誌群組總管。

若要存取檢視記錄群組功能並開啟 CloudWatch 日誌群組 Explorer，請完成下列步驟。

1. 來自AWSExplorer, 展開亞馬遜 CloudWatch。
2. 按兩下日誌群組或開啟內容功能表 (按一下內容功能表)，然後選取檢視，開啟CloudWatch 日誌群組 Explorer。

> **Note**
>
> 所以此 CloudWatch 記錄群組總管會在 [解決方案總管] 的相同視窗位置中開啟。

篩選日誌群組

您的個人帳戶可以包含數千個不同的記錄群組。若要簡化搜尋特定群組，請使用filtering功能如下所述。

1. 來自CloudWatch 日誌群組總管，將游標置於視窗頂端的搜尋列中。
2. 開始輸入與您要尋找的記錄群組相關的前置詞。
3. CloudWatch 日誌群組總管系統會自動更新，以顯示符合您在上一步中指定之搜尋字詞的結果。

刪除日誌群組

若要刪除特定日誌群組，請參閱下列程序。

1. 來自CloudWatch 日誌群組總管，以滑鼠右鍵按一下您想要刪除的日誌群組。
2. 出現提示時，確認您想要刪除目前選取的日誌群組。

3. 選擇合適的是按鈕刪除選取的記錄群組，然後重新整理CloudWatch 日誌群組總管。

重新整理日誌群組

若要重新整理目前顯示在中的記錄群組清單CloudWatch 日誌群組 Explorer，選擇重新整理圖示按鈕位於工具欄。

複製日誌群組 ARN

若要複製特定記錄群組的 ARN，請完成下列步驟。

1. 來自CloudWatch 日誌群組總管，以滑鼠右鍵按一下要從中複製 ARN 的「記錄群組」。
2. 選擇複製 ARN從功能表選項。
3. ARN 現在會複製到您的本機剪貼簿，並準備好貼上。

CloudWatch 日誌串流

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，記錄串流會依最近的事件時間戳記排序。
> - 與記錄資料流相關聯的資料行可以按遞增或遞減順序排序，方法是切換插入符位於列標題中。
> - 篩選後的項目只能依據Log Stream Name (日誌串流名稱)。

檢視日誌串流

1. 來自CloudWatch 日誌群組總管在日誌群組上按兩下日誌群組，或在日誌群組上按一下日誌群組檢視日誌串流從內容功能表。
2. 新的標籤隨即開啟文件window，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. 來自日誌串流」頁籤中的文件視窗中，將游標設定在搜尋列中。

2. 開始輸入與您要尋找的記錄資料流相關的前置詞。
3. 當您輸入時，目前的顯示會自動更新，以根據您的輸入篩選「記錄串流」。

重新整理理理論壇

若要重新整理目前顯示在中的日誌串流清單文件視窗中，選擇重新整理圖示按鈕，位於工具欄，旁邊搜索欄。

複製日誌串流 ARN

若要複製特定記錄資料流的 ARN，請完成下列步驟。

1. 來自日誌串流」頁籤中的文件視窗中，以滑鼠右鍵按一下您想要從中複製 ARN 的日誌串流。
2. 選擇複製 ARN從功能表選項。
3. ARN 現在會複製到您的本機剪貼簿，並準備好貼上。

下載日誌串流

所以此匯出日誌串流功能會下載並將選取的記錄資料流儲存在本機，以便透過自訂工具和軟體存取，以進行其他處理。

1. 來自日誌串流」頁籤中的文件視窗中，以滑鼠右鍵按一下您想要下載的日誌串流。
2. 選擇匯出日誌串流開啟匯出至文字檔案對話方塊。
3. 選擇您要在本機儲存檔案的位置，並在提供的文字欄位中指定名稱。
4. 通過選擇確認下載確定。下載狀態會顯示在視覺工作室任務狀態中心

CloudWatch 日誌事件

日誌事件是由應用程式或正受到監控的資源所記錄的活動日誌 CloudWatch。

日誌事件動作

記錄事件會顯示為表格。依預設，事件會從最舊的事件排序到最新的事件。

下列動作會與記錄檔事件相關聯：

- 自動換行文字模式：您可以通過單擊事件切換包裝文本。
- 文字換行按鈕：位於document window **toolbar**，此按鈕可切換所有項目的文字換行開啟和關閉。

- 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. 來自文件」視窗中，選擇包含日誌串流清單的標籤。
2. 按兩下記錄串流，或在記錄串流上按一下滑鼠右鍵，然後選取檢視日誌串流從功能表開始自。
3. 一個新的日誌事件選項卡將在文件window，其中包含與您選擇的記錄資料流相關聯的記錄事件表格。

篩選日誌事件

有三種方法可供您篩選記錄事件：依內容、時間範圍或兩者。若要依內容和時間範圍篩選記錄事件，請先依內容或時間範圍篩選郵件，然後以其他方法篩選這些結果。

若要依內容篩選記錄事件：

1. 來自日誌事件」頁籤中的文件視窗中，將游標設定在搜尋列中，位於視窗頂端。
2. 開始輸入與您要搜尋的記錄事件相關的字詞或片語。
3. 當您輸入時，目前的顯示會自動開始篩選記錄事件。

> **Note**
>
> 篩選條件模式區分大小寫。您可以將完全相符字詞和詞組括在雙引號中，以非英數字元括住，以改善搜尋結果。如需篩選條件模式的詳細資訊，請參閱篩選條件和模式語法在亞馬遜主題 CloudWatch 指南。

若要檢視特定時間範圍內產生的記錄事件：

1. 來自日誌事件」頁籤中的文件視窗中，選擇行事曆圖示按鈕，位於工具欄。
2. 使用提供的欄位指定您想要搜尋的時間範圍。
3. 當您指定日期和時間限制時，篩選的結果會自動更新。

> **Note**
>
> 所以此清除篩選條件選項清除所有當前 date-and-time 篩選選取。

重新整理日誌事件

若要重新整理目前顯示在中的記錄事件清單日誌事件標籤上選擇重新整理圖示按鈕，位於工具欄。

額外的訪問權限 CloudWatch日誌

您可以存取 CloudWatch 與其他相關聯的日誌AWS服務和資源直接來自AWS視覺工作室中的工具包。

Lambda

若要檢視與 Lambda 函數相關聯的日誌串流，請執行下列動作：

> **Note**
>
> Lambda 執行角色必須擁有適當許可，才能將日誌傳送至 CloudWatch記錄檔。如需所需 Lambda 許可的詳細資訊 CloudWatch 記錄檔，請參閱https://docs.aws.amazon.com/lambda/latest/dg/monitoring-cloudwatchlogs.html#monitoring-cloudwatchlogs-prereqs

1. 來自AWS工具組總管, 展開Lambda。
2. 在您要檢視的函數上按一下您想要檢視的函數，然後檢視日誌以開啟相關的記錄資料流文件Windows。

若要使用 Lambda 整合檢視日誌串流`function view`：

1. 來自AWS工具組總管, 展開Lambda。
2. 在您要檢視的函數上按一下您想要檢視的函數，然後檢視功能以開啟功能檢視文件Windows。
3. 來自`function view`，切換至日誌」索引標籤中，會顯示與所選 Lambda 函數相關聯的記錄資料流。

ECS

欲檢視與 ECS 任務容器相關聯的記錄資源，請完成下列程序。

> **Note**
>
> 為了使亞馬遜 ECS 服務將日誌發送到 CloudWatch，指定 Amazon ECS 任務的每個容器都必須符合所需的組態。有關所需設置和配置的其他信息，請參閱指南使用AWS日誌驅動程式。

1. 來自AWS工具組總管, 展開Amazon ECS。
2. 選擇您想要檢視的 Amazon ECS 叢集以開啟新的 Amazon ECS 叢集ECS 叢集」頁籤中的文件Windows。
3. 在導覽功能表中，位於ECS 叢集標籤上選擇工作來列出與叢集相關的所有 Toolkit (列出叢集)。
4. 來自工作顯示，選取工作並選擇檢視日誌鏈接，位於左下角。

> **Note**
>
> 此顯示會列出叢集中包含的所有工作，`View Logs`只有符合所需記錄組態的每項工作才會顯示連結。
>
> - 如果工作只與單一容器相關聯，檢視日誌鏈接打開該容器的日誌流。
> - 如果工作與多個容器相關聯，檢視日誌鏈接打開檢視 CloudWatch ECS 任務的日誌」對話方塊中，使用容器：下拉式功能表以選擇您要檢視記錄的容器，然後選擇確定。

5. 新的標籤將在中開啟文件視窗會顯示與您的容器選取項相關聯的記錄資料流。

管理 Amazon EC2 執行個體

AWS資源管理器提供 Amazon Machine Image (AMI) 和 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體的詳細視圖。通過這些視圖，您可以從 AMI 啟動 Amazon EC2 實例、連接到該實例以及停止或終止實例，所有這些都可以從 Visual Studio 開發環境中進行。您可以使用實例視圖從您的實例創建 AMI。如需詳細資訊，請參閱「」從 Amazon EC2 執行個體創建 AMI。

亞馬遜計算機映像和 Amazon EC2 實例視圖

從AWS瀏覽器中，您可以顯示 Amazon Machine Image (AMI) 和 Amazon EC2 執行個體的視圖。InAWS資源管理器中，展開Amazon EC2節點。

要顯示 AMI 視圖，請在第一個子節點上AMI，開啟內容功能表 (按一下滑鼠右鍵)，然後選擇檢視。

要顯示 Amazon EC2 實例視圖，請在執行個體節點中，開啟內容功能表 (按一下滑鼠右鍵)，然後選擇檢視。

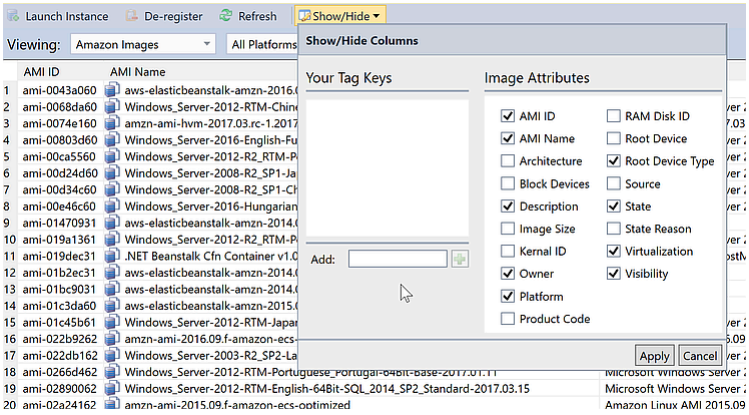
您也可以通過雙擊相應的節點來顯示任一視圖。

- 視圖的作用域限定為AWS瀏覽器 (例如，美國西部 (加利佛尼亞北部) 區域)。
- 您可以通過單擊並拖動來重新排列列。若要排序欄位中的值，請按一下該欄位。

- 您可以使用下拉式清單和檢視配置視圖。初始視圖顯示任何平台類型（Windows 或 Linux）的 AMI，這些平台類型由AWSExplorer

ShowHide 列/Hide 列

您也可以在ShowHide /Hide下拉菜單，以配置顯示哪些列。如果關閉視圖並重新打開該視圖，您選擇的列將持續存在。



ShowHide 列/Hide 列用於 AMI 和實例視圖的 UI

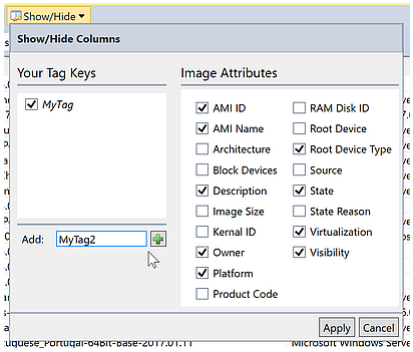
標記 AMI、實例和卷

您也可以使用ShowHide /Hide下拉列表為您擁有的 AMI、Amazon EC2 實例或卷添加標籤。標籤是名稱值對，讓您可以將元數據附加至 AMI、執行個體和卷。標籤名稱的作用域既適用於您的賬户，也分別限定為您的 AMI 和實例。例如，如果您對 AMI 和實例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，請前往使用標籤中的Amazon EC2 Linux 執行個體使用者指南。

新增標籤

1. 在 中Add方塊中，輸入標籤的名稱。選擇帶加號 (+) 的綠色按鈕，然後選擇套用。



向 AMI 或 Amazon EC2 實例添加標籤

新標記以斜體顯示，表示尚未與該標記關聯任何值。

在列表視圖中，標記名稱顯示為新列。當至少有一個值與標記關聯時，該標記將在AWS Management Console。

2. 要為標記添加值，請雙擊該標記列中的單元格，然後鍵入值。要刪除標籤值，請雙擊單元格並刪除文本。

   如果清除ShowHide /Hide下拉式清單中，相應的欄位會從視圖中消失。標記以及與 AMI、實例或卷關聯的任何標記值都會保留。

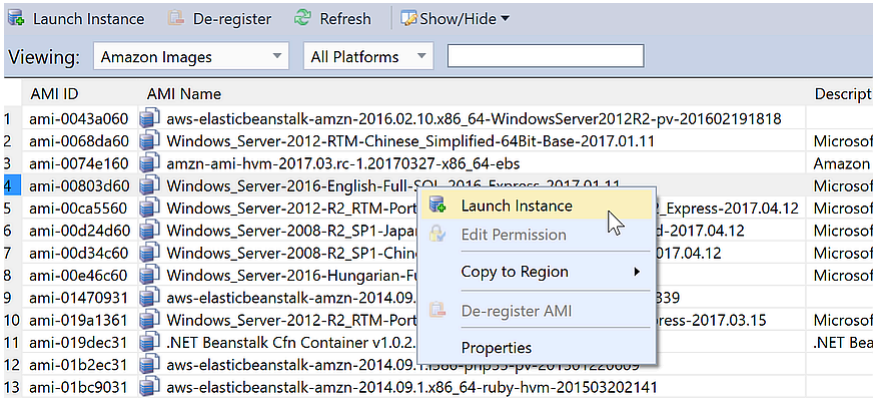
> **Note**
>
> 如果清除ShowHide /Hide下拉列表中沒有任何關聯值，則AWS工具包將完全刪除該標籤。它不會再顯示在清單視圖中或ShowHide /Hide。要再次使用該標籤，請使用ShowHide / Hide對話框以重新創建它。

啟動 Amazon EC2 執行個體

AWS資源管理器提供啟動 Amazon EC2 執行個體所需的所有功能。在本節中，我們將選擇 Amazon Machine Image (AMI)，進行配置，然後將其作為 Amazon EC2 執行個體啟動。

啟動 Amazon EC2 執行個體

1. 在 AMI 視圖頂部的左側下拉列表中，選擇Amazon Image。在右側的下拉式清單中，選擇視窗。在篩選方塊中，輸入ebs用於彈性塊存儲。可能需要一點時間才能刷新視圖。
2. 在清單中選擇 AMI，開啟內容功能表 (按一下滑鼠右鍵)，然後選擇啟動執行個體. .



AMI 列表

3. 在 中啟動 Amazon EC2 執行個體對話框中，為您的應用程式配置 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

   輸入您的執行個體的名稱。此名稱不可超過 256 個字元。

   金鑰對

   key pair 用於獲取您使用遠端桌面協定 (RDP) 登入 EC2 執行個體所使用的 Windows 密碼。選擇您有權訪問私鑰的 key pair，或選擇創建 key pair 的選項。如果您在 Toolkit 中創建 key pair，則 Toolkit 可以為您存儲私鑰。

   存儲在 Toolkit 中的密鑰對進行加密。您可以在%LOCALAPPDATA%\AWSToolkit\keypairs（通常：C:\Users\<user>\AppData\Local\AWSToolkit\keypairs。 您可以將加密金 key pair 導出到.pemfile.

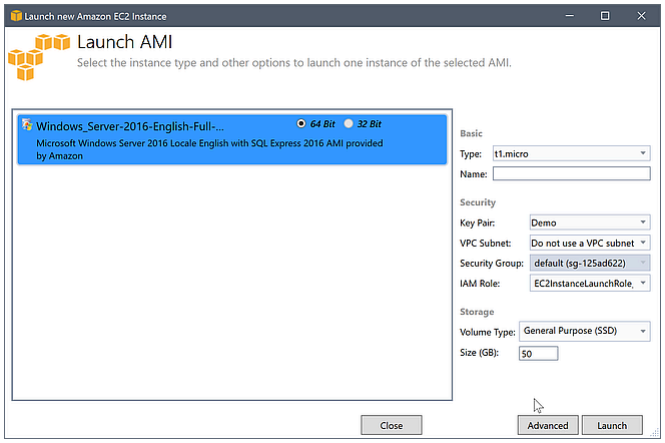
   a. 在 Visual Studio 中，選擇檢視，然後按一下AWS探險者。
   b. 按一下Amazon EC2，然後選擇金鑰對。
   c. 將列出密鑰對，並且由 Toolkit 創建/管理的密鑰對標記為儲存在 AwstoolKit。
   d. 右鍵單擊您創建的 key pair，然後選擇匯出私有金鑰。私鑰將取消加密並存儲在您指定的位置。

   安全群組

   安全組控制 EC2 執行個體將接受的網絡流量類型。選擇允許連接 RDP 所使用的連接埠 (RDP 所使用的連接埠) 流量傳入的安全組，讓您可以連接至 EC2 執行個體。如需如何使用 Toolkit 創建安全組的詳細資訊，請參管理安全組AWS探險者。

   執行個體描述檔

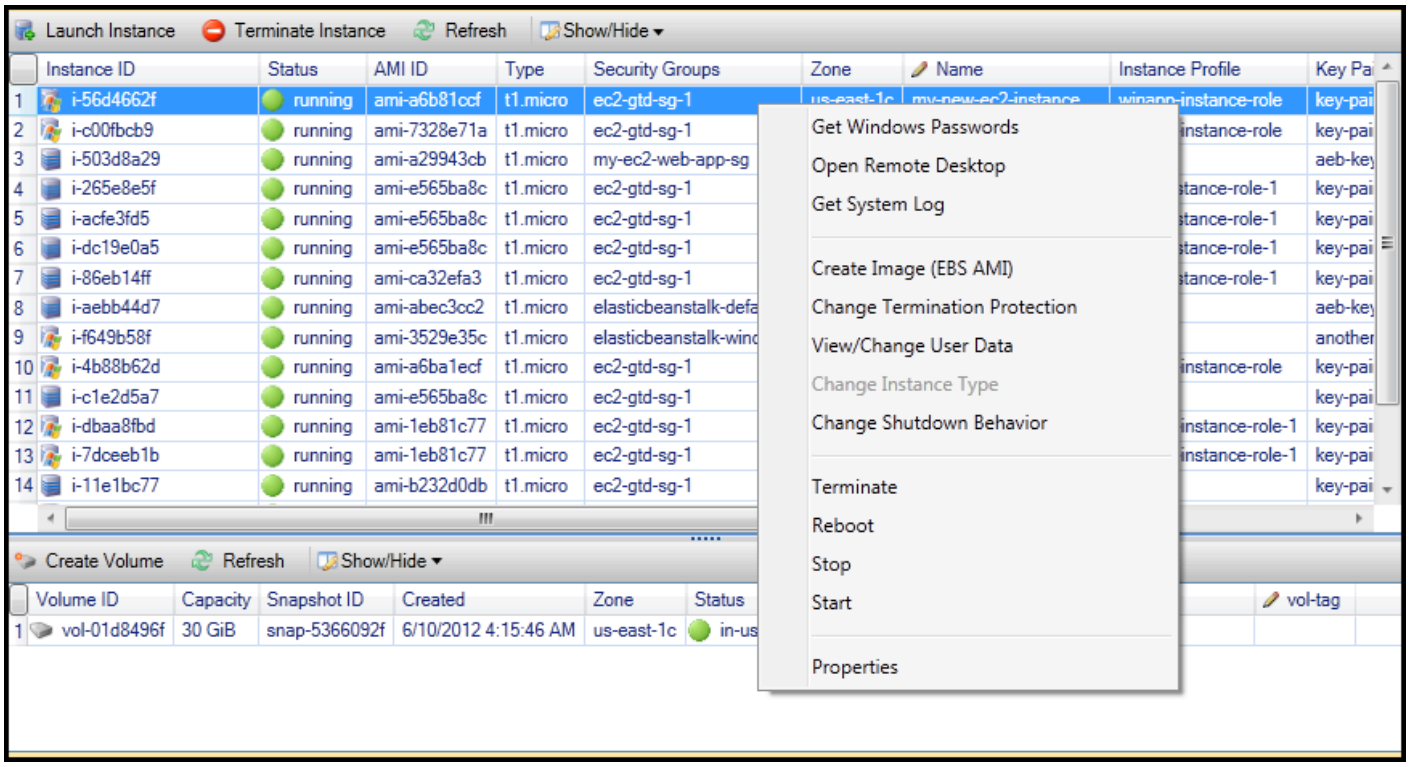
   執行個體描述檔是 IAM 角色的邏輯容器。選擇執行個體配置文件時，您會將相應的 IAM 角色與 EC2 執行個體建立關聯。IAM 角色配置了指定 Amazon Web Services 和賬户資源訪問權限的策略。一個 EC2 執行個體與 IAM 角色關聯時，在執行個體上執行的應用程式軟體將以 IAM 角色指定的許可執行。這樣便能讓應用程式軟體無需指定AWS憑據，使軟體更加安全。如需 IAM 角色的詳細資訊，請前往IAM User Guide。



EC2啟動 AMI對話方塊

4. 選擇 Launch (啟動)。

InAWS資源管理器，在執行個體子節點Amazon EC2，開啟內容功能表 (按一下滑鼠右鍵)，然後選擇檢視。所以此AWS工具包顯示與活動賬户關聯的 Amazon EC2 實例的列表。您可能需要選擇重新整理以查看您的新執行個體。當實例首次出現時，它可能處於掛起狀態，但過了一會兒，它會轉換到正在運行狀態。



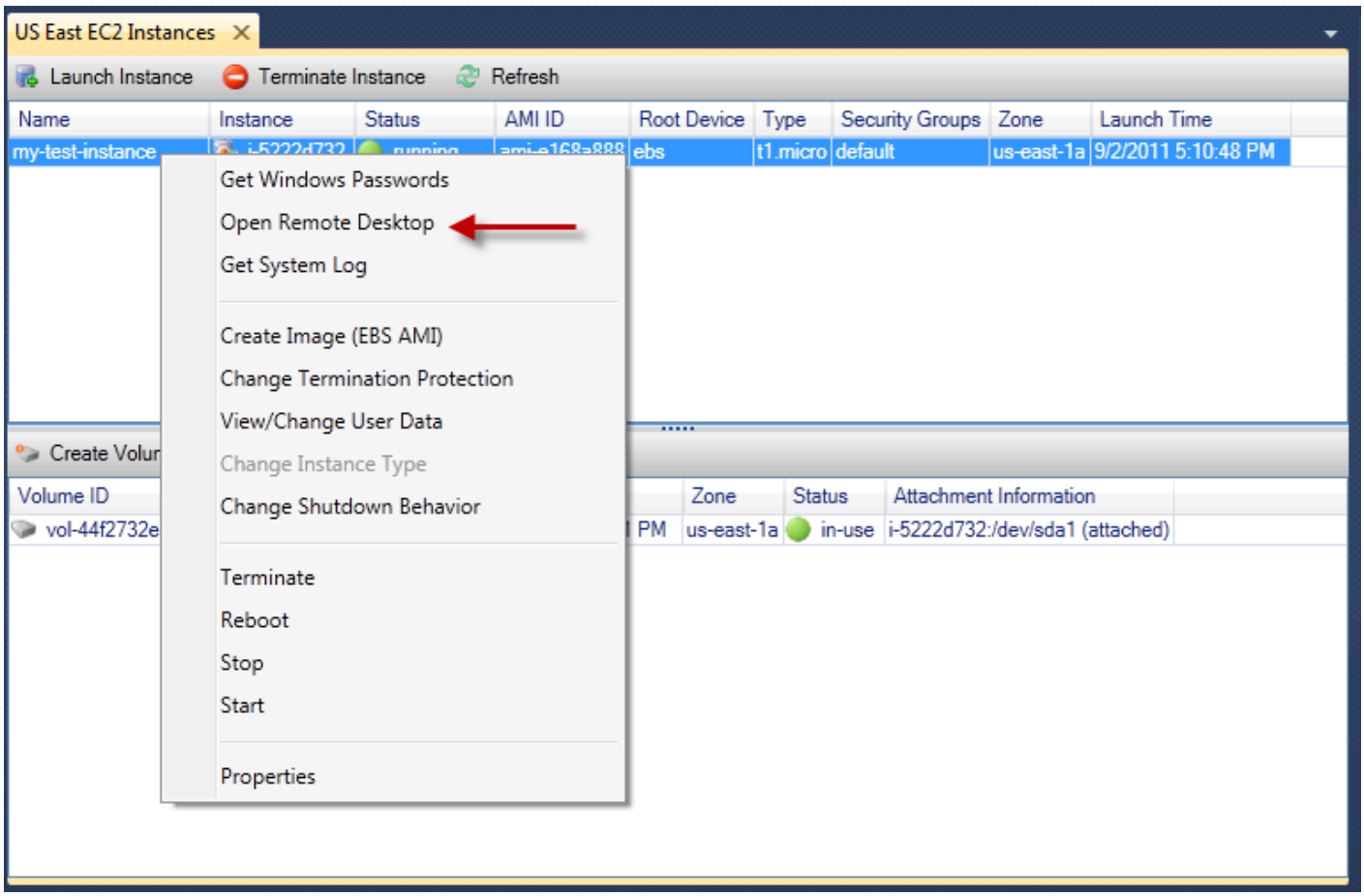
連接 Amazon EC2 執行個體

您可以使用 Windows 遠端桌面連接至 Windows 服務器執行個體。對於身份驗證，AWSToolkit 允許您檢索實例的管理員密碼，也可以簡單地使用與實例關聯的存儲 key pair。在以下過程中，我們將使用存儲的 key pair。

使用 Windows 遠端桌面連接至 Windows 服務器執行個體

1. 在 EC2 執行個體清單中，按一下您要連接的 Windows Server 執行個體。從內容功能表選擇開啟遠端桌面。

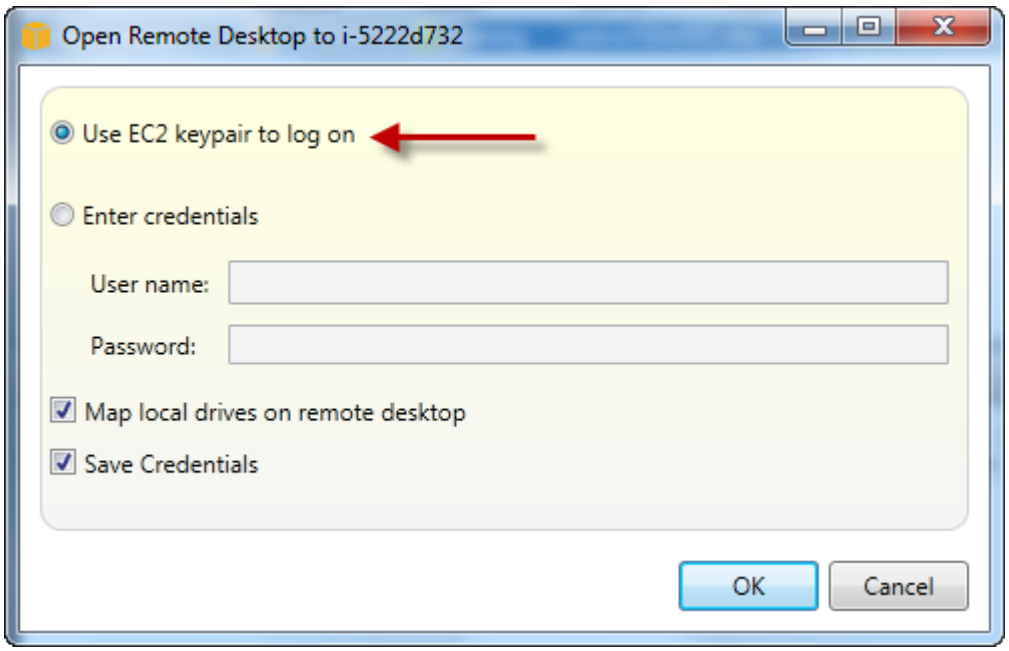
   如果要使用管理員密碼進行身份驗證，可以選擇獲取窗口密碼。



   EC2 實例上下文菜單

2. 在 中開啟遠端桌面對話方塊中，選擇使用 EC2 密鑰對登錄選擇，接着選擇確定。

   如果您沒有將 key pair 與AWS工具包中，指定包含私有金鑰的 PEM 文件。

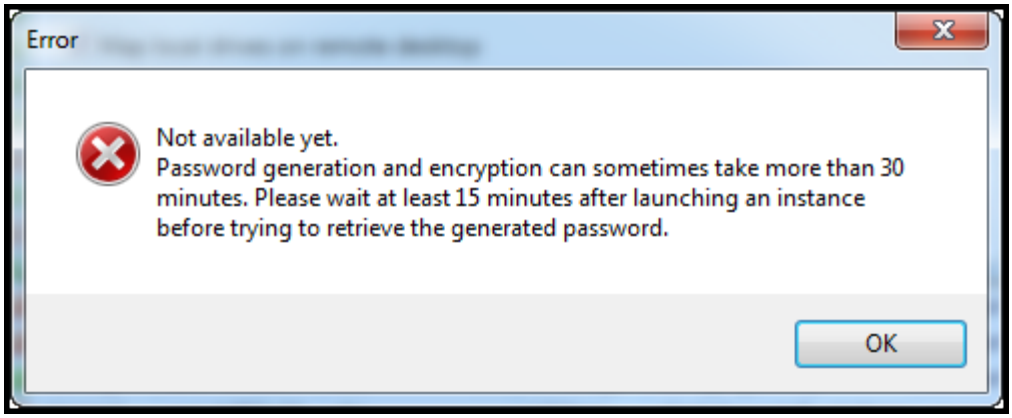


開啟遠端桌面對話方塊

3. 所以此遠端桌面窗口將會打開。您無需登錄，因為 key pair 進行身份驗證。您將以 Amazon EC2 執行個體上的管理員身份運行。

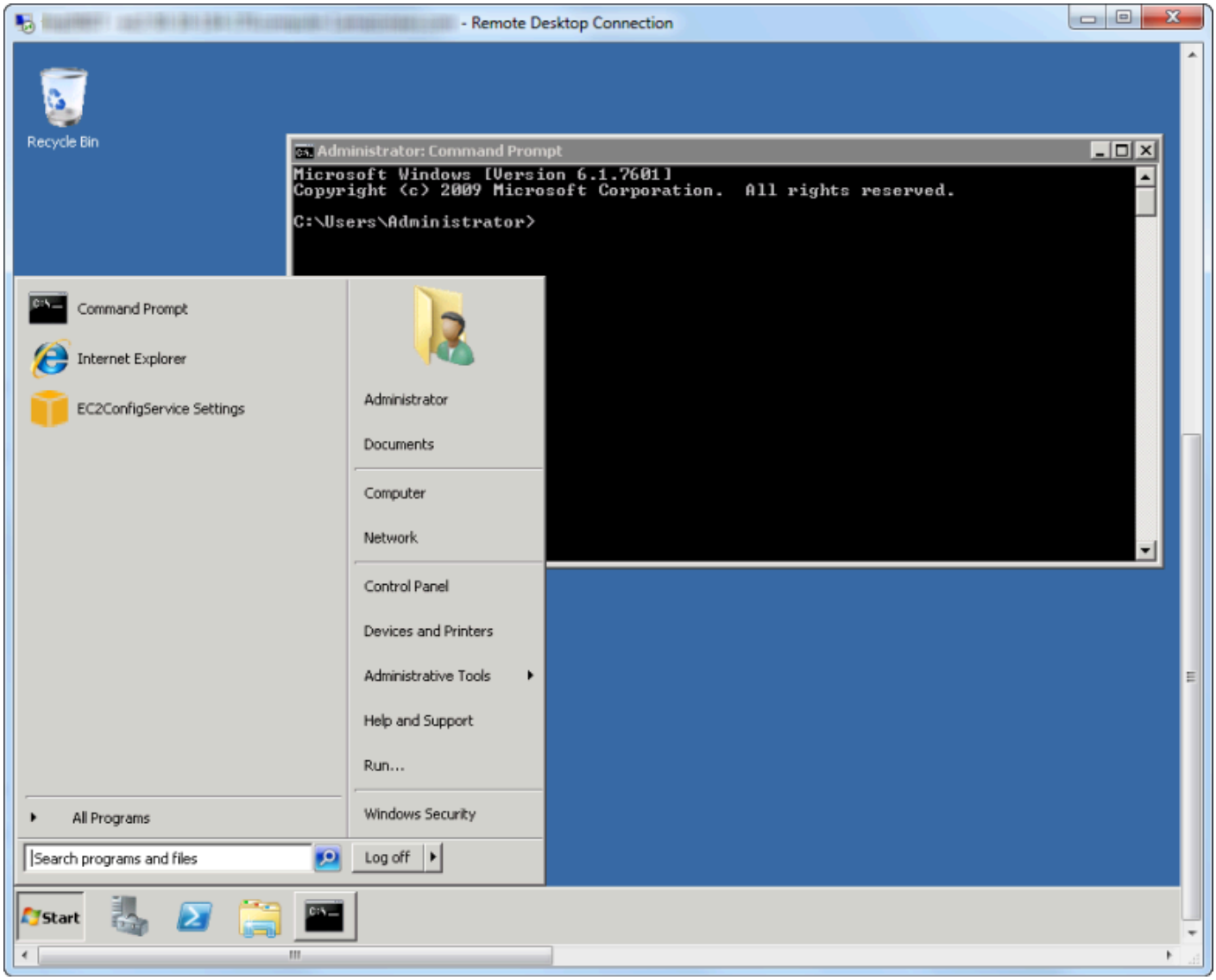
   如果 EC2 實例最近才啟動，您可能由於以下兩個原因無法連接：

   - 遠程桌面服務可能尚未啟動並運行。請等待幾分鐘後再試一次。
   - 密碼信息可能尚未傳輸到實例。在這種情況下，您會看到類似如下的消息框。



密碼尚不可用

以下屏幕截圖顯示了以管理員身份通過遠程桌面連接的用户。



遠端桌面

結束 Amazon EC2 執行個體

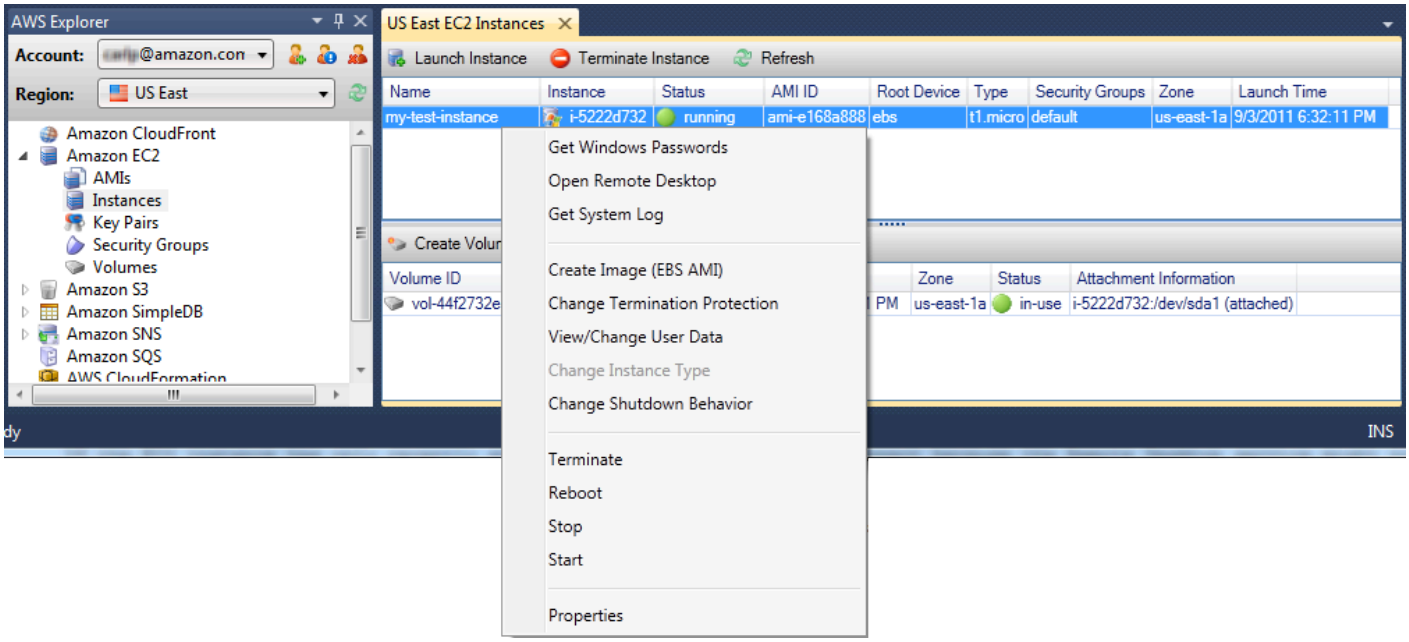
使用AWS工具包，您可以從 Visual Studio 停止或終止運行中的 Amazon EC2 執行個體。要停止實例，EC2 實例必須使用 Amazon EBS 卷。如果 EC2 實例未使用 Amazon EBS 卷，則您唯一的選擇是終止該實例。

如果停止執行個體，EBS 捲上存放的數據會保留。如果終止實例，則存儲在實例的本地存儲設備上的所有數據都將丟失。無論是停止還是終止哪種情況，都不會繼續向您支付 EC2 實例的費用。但是，如果您停止實例，則將繼續為實例停止後持續存在的 EBS 存儲收取費用。

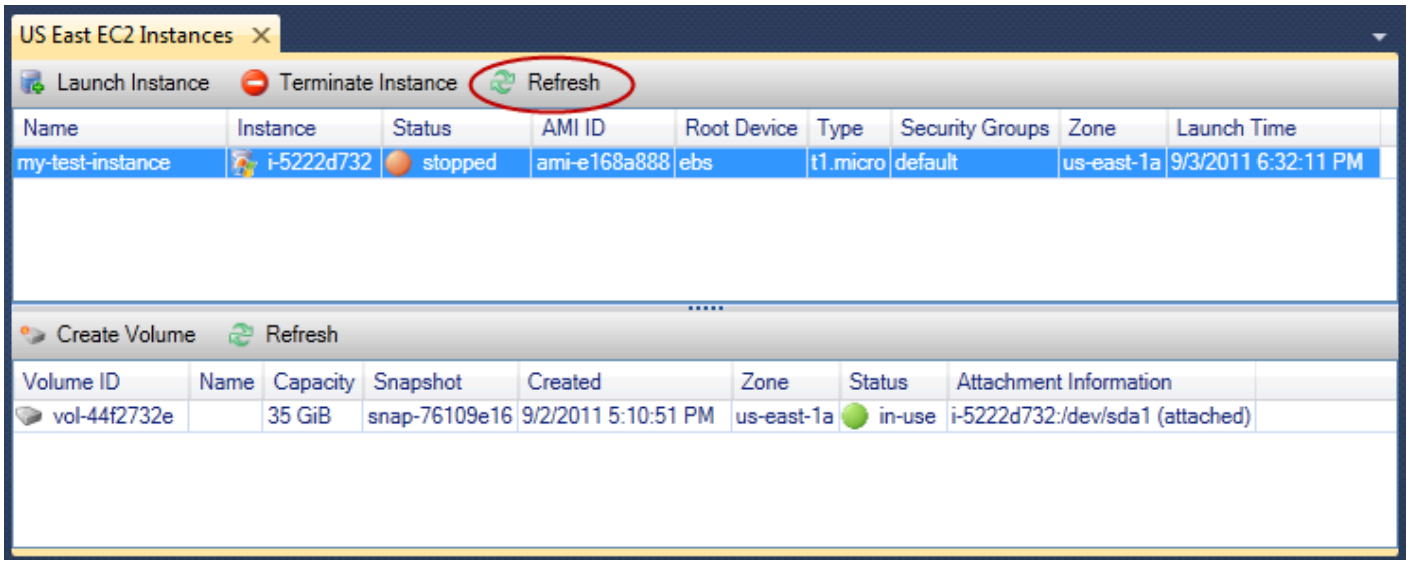
結束實例的另一種可能方法是使用遠程桌面連接到實例，然後從 Windows啟動選單中，使用Shutdown。在這種情況下，您可以將實例配置為停止或終止。

停止 Amazon EC2 執行個體

1. InAWS資源管理器中，展開Amazon EC2節點中，開啟內容 (按右鍵) 功能表，然後選擇執行個體選擇，接着選擇檢視。在 中執行個體列表中，右鍵單擊要停止的實例，然後選擇停止。選擇是以確認您要停止執行個體。

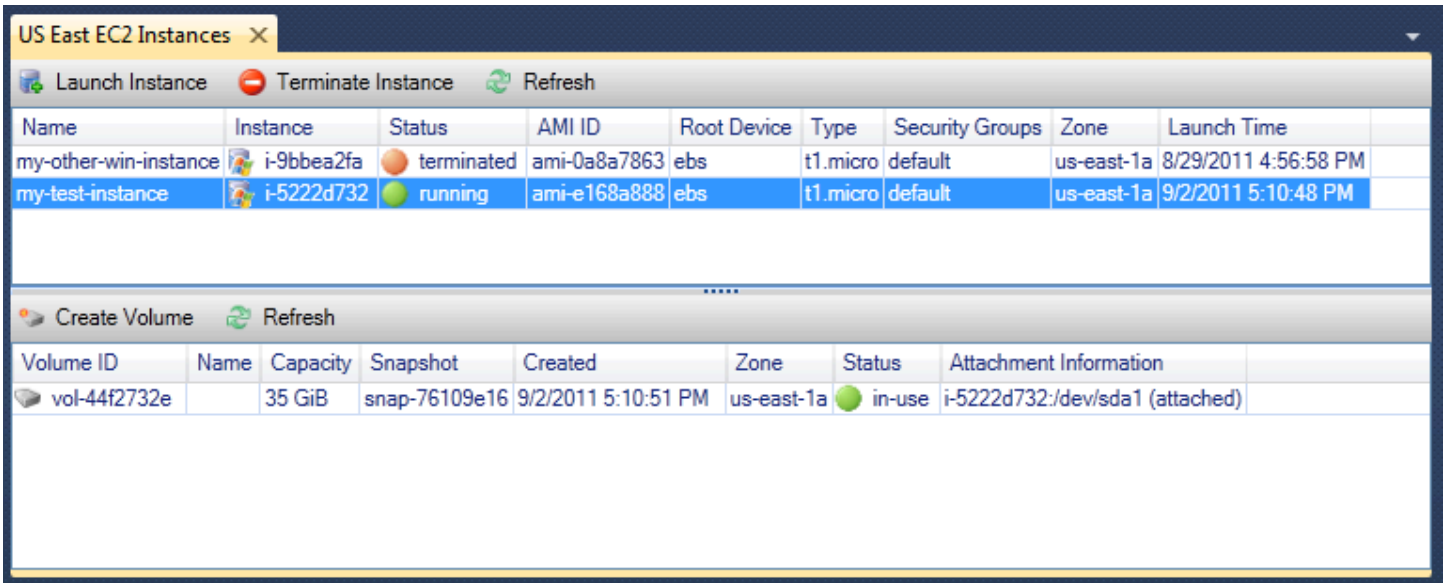


2. 在執行個體列表中，選擇重新整理查看 Amazon EC2 執行個體狀態的變化。由於我們停止而不是終止實例，因此與實例關聯的 EBS 卷仍處於活動狀態。



終止的實例保持可見

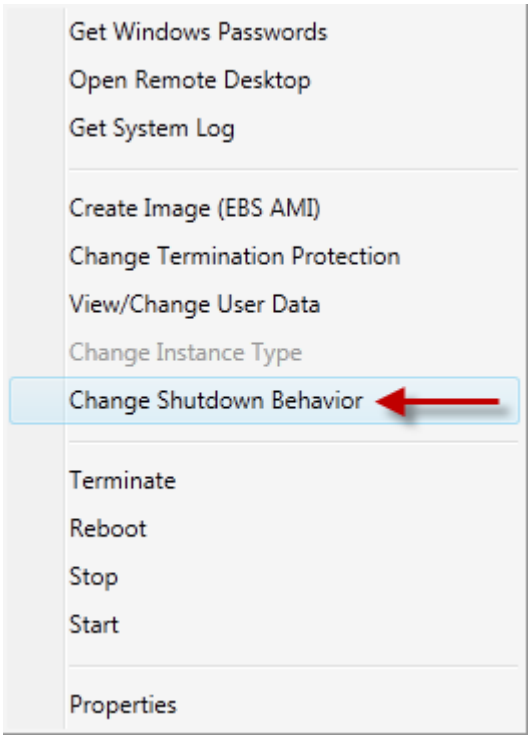
如果您終止某個實例，該實例將繼續顯示在執行個體列表以及正在運行或已停止的實例。最終,AWS回收這些執行個體，它們會從清單中消失。您無需為處於終止狀態的實例收費。



指定 EC2 實例關閉時的行為

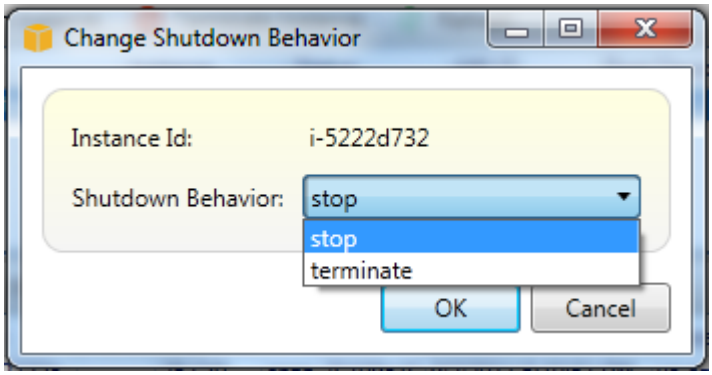
所以此AWS使用工具包，您可以指定 Amazon EC2 實例將停止還是終止Shutdown選擇啟動選單。

1. 在 中執行個體清單中，按一下 Amazon EC2 執行個體，然後選擇更改關機行為。



更改關機行為選單項目

2. 在 中更改關機行為對話框中，從關機行為下拉式清單中，選擇停止或者終止。



管理 Amazon ECS 執行個體

AWSExplorer 提供 Amazon Elastic Container Service (Amazon ECS) 叢集和容器儲存庫的詳細視圖。您可以從 Visual Studio 開發環境中創建、刪除和管理羣集和容器詳細信息。

修改服務屬性

您可以從羣集視圖查看服務詳細信息、服務事件和服務屬性。

1. InAWS選擇 Explorer，開啟叢集的內容 (按一下滑鼠右鍵) 功能表，然後選擇檢視。
2. 在彈性雲服務器羣集視圖中，單擊服務，然後按一下滑鼠右側的詳細資訊選項卡中的詳細信息視圖。您可以單擊活動查看事件消息，然後部署設定為部署狀態。
3. 按一下 Edit (編輯)。您可以更改所需的任務計數以及最小和最大健康百分比。
4. 按一下Save接受更改，或Cancel以恢復到現有值。

停止任務

您可以在羣集視圖中查看任務的當前狀態並停止一個或多個任務。

停止任務

1. InAWS在 Explorer 中，開啟叢集 (按一下滑鼠右鍵) 功能表，然後選擇檢視。
2. 在彈性雲服務器羣集視圖中，單擊工作在左側。
3. 確定所需任務狀態已設定為Running。選擇要停止的單個任務，然後單擊停止或按一下滑鼠全停止選擇並停止所有正在運行的任務。

4. 在 中停止任務對話方塊中，選擇是。

刪除服務

您可以從羣集視圖中刪除羣集中的服務。

刪除叢集服務

1. InAWS在 Explorer 中，開啟您要刪除的服務的叢集的內容 (按一下滑鼠右鍵) 功能表，然後選擇檢視。
2. 在彈性雲服務器羣集視圖中，單擊服務，然後單擊刪除。
3. 在 中刪除叢集對話框中，如果集羣中存在負載均衡器和目標組，則可以選擇將它們與羣集一起刪除。刪除服務時不會使用它們。
4. 在 中刪除叢集對話方塊中，選擇確定。當集羣被刪除時，它將從AWSExplorer

刪除叢集

您可以刪除 Amazon Elastic Container Service 叢集AWSExplorer

刪除叢集

1. InAWS在 Explorer 中，開啟您要刪除叢集的內容 (按一下滑鼠右鍵) 功能表，然後選擇叢集節點Amazon ECS，然後選擇刪除。
2. 在 中刪除叢集對話方塊中，選擇確定。當集羣被刪除時，它將從AWSExplorer

建立儲存庫

您可以創建亞馬遜彈性容器註冊表存儲庫，從AWSExplorer

建立一組儲存庫

1. InAWS在 Explorer 中，開啟內容功能表 (按一下滑鼠右鍵)儲存庫節點Amazon ECS，然後選擇建立儲存庫。
2. 在 中建立儲存庫對話框中，提供存儲庫名稱，然後選擇確定。

刪除儲存庫

您可以將 Amazon 彈性容器註冊表存儲庫從AWSExplorer

刪除儲存庫

1. InAWS在 Explorer 中，開啟內容功能表 (按一下滑鼠右鍵)儲存庫節點Amazon ECS，然後選擇刪除儲存庫。
2. 在 中刪除儲存庫對話框中，即使存儲庫包含圖像，也可以選擇刪除該存儲庫。否則，只有空的話才會刪除。按一下是。

管理安全羣組AWS探險者

Visual Studio Toolkit for Visual Studio) 使您能夠建立和配置安全羣組，以與 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體和AWS CloudFormation。當您啟動 Amazon EC2 執行個體或將應用程式部署到AWS CloudFormation，您可以指定要與 Amazon EC2 執行個體建立關聯的安全羣組。（部署到AWS CloudFormation會建立 Amazon EC2 執行個體。）

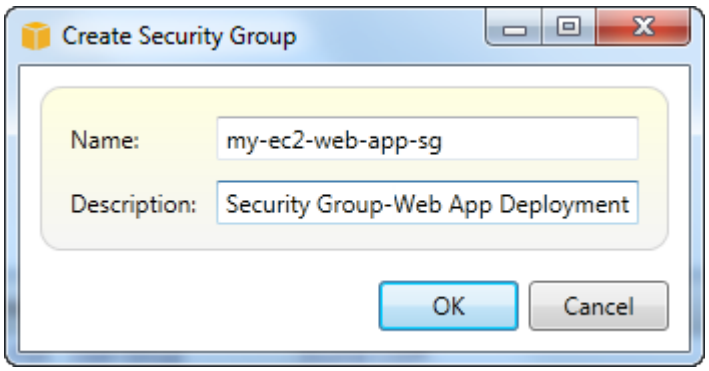
安全群組對於傳入網路流量的功能，就像防火牆一樣。安全組指定在 Amazon EC2 實例上允許使用哪些類型的網絡流量。它還可以指定僅能從特定 IP 地址或僅接受來自指定用户或其他安全羣組的傳入流量。

建立安全群組

在本節中，我們將建立安全羣組。安全羣組建立時，不會有任何許可設定。以額外操作處理設定許可。

建立安全群組

1. InAWS資源管理器，在Amazon EC2按一下滑鼠右鍵 (按一下滑鼠右鍵) 功能表，在安全群組節點，然後選擇檢視。
2. 在EC2 安全群組選項卡上，選擇建立安全羣組。
3. 在 中建立安全羣組對話方塊中，鍵入安全羣組的名稱和描述，然後選擇確定。

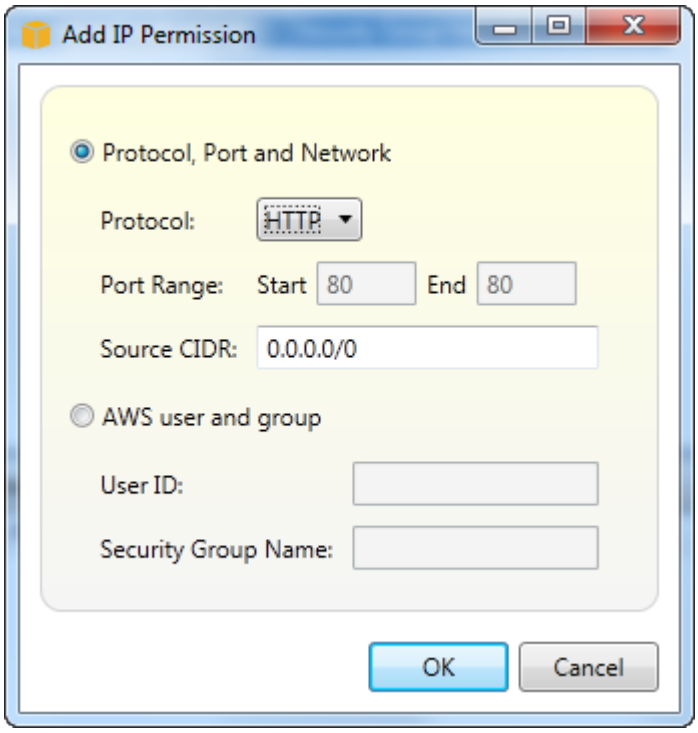


為安全群組新增許可

在本節中，我們將新增許可到安全羣組，以允許通過 HTTP 和 HTTPS 協定的 Web 流量。我們還允許其他電腦使用 Windows 遠端桌面通訊協定 (RDP) 進行連線。

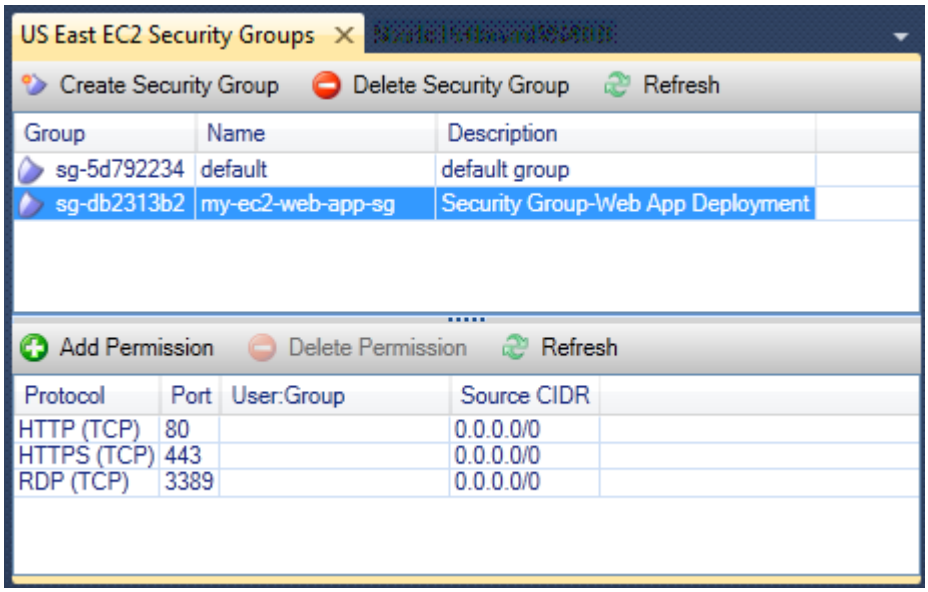
若要為安全群組新增許可

1. 在EC2 安全群組選項卡上，選擇安全羣組，然後選擇新增許可按鈕。
2. 在 中新增許可對話方塊中，選擇協議、端口和網絡單選按鈕，然後從協議下拉式清單中，選擇HTTP。端口範圍會自動調整為端口 80，這是 HTTP 的默認端口。所以此源 CIDR字段默認設定為 0.0.0/0，指定將從任何外部 IP 地址接受 HTTP 網路流量。選擇 OK (確定)。



為此安全組打開端口 80 (HTTP)

3. 針對 HTTPS 和 RDP 重複這個程序。您的安全羣組許可現在應該與下列類似。



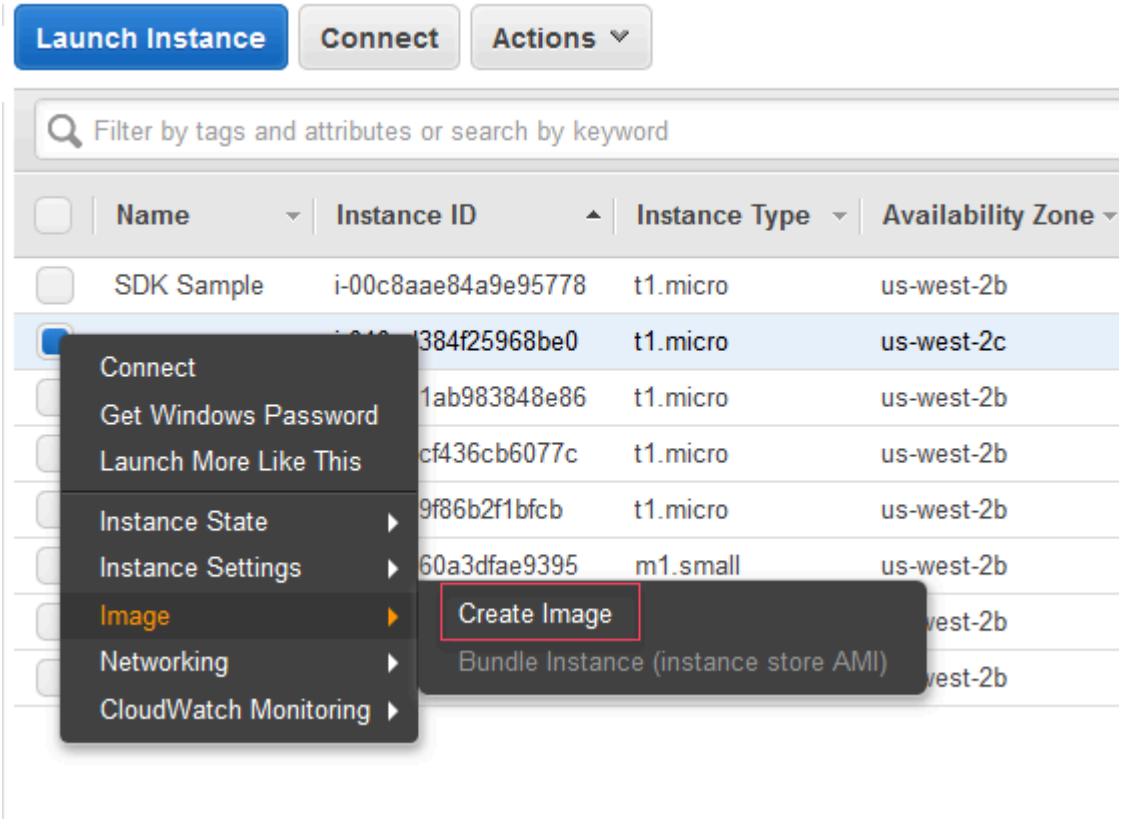
您也可以通過指定用户 ID 和安全羣組名稱來設定安全羣組中的許可。在這種情況下，此安全羣組中的 Amazon EC2 執行個體會接受所有由指定安全羣組中 Amazon EC2 執行個體所傳入的網路流量。您還必須指定用户 ID，以區分安全羣組名稱；安全羣組名稱在所有AWS。如需安全羣組的詳細資訊，請轉到EC2 文檔。

從 Amazon EC2 執行個體建立 AMI

從 Amazon EC2 執行個體檢視中，您可以從執行中或停止的執行個體建立 Amazon 機器映像 (AMI)。如需 AMI 的詳細資訊，請參閱《Amazon E lastic Compute Cloud Windows 執行個體使用者指南》中的 Elastic Compute (AMI) 主題。

從執行個體建立 AMI

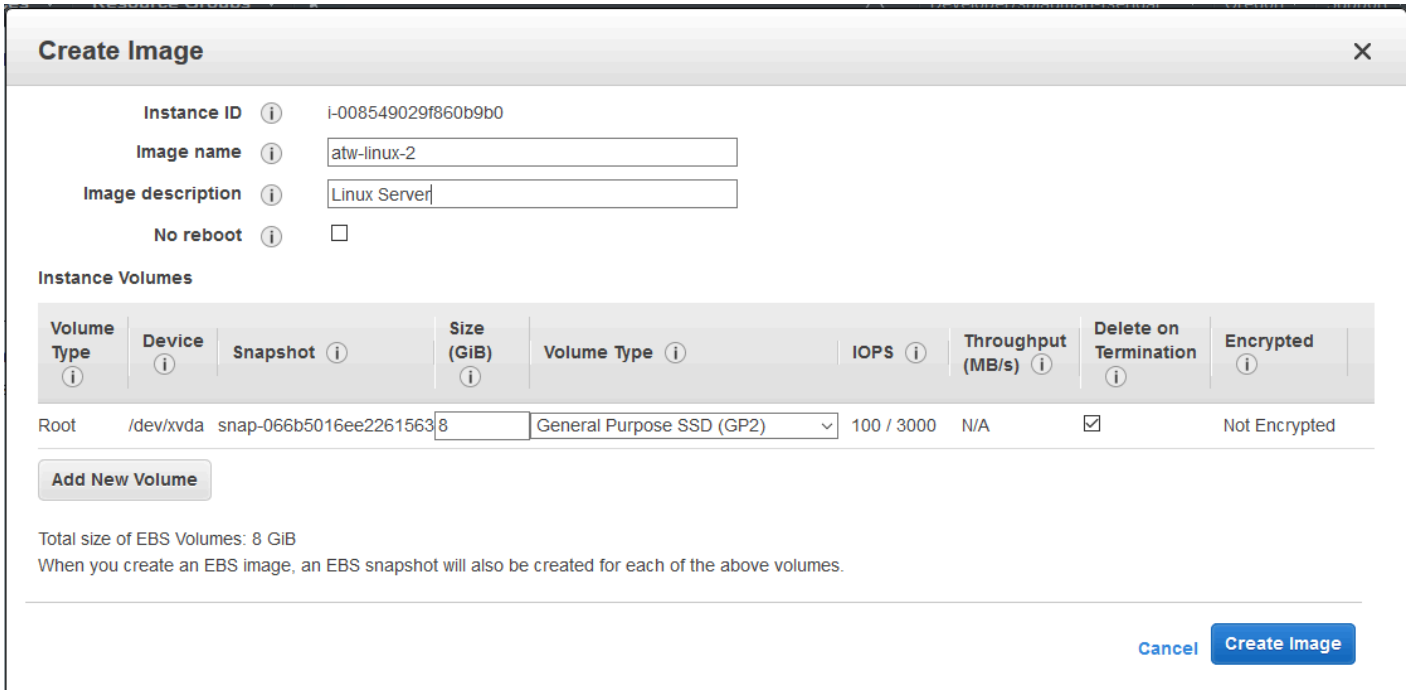
1. 在您要用作 AMI 基礎的執行個體上按一下滑鼠右鍵，然後從內容功能表中選擇 [建立映像]。



建立影像右鍵功能表

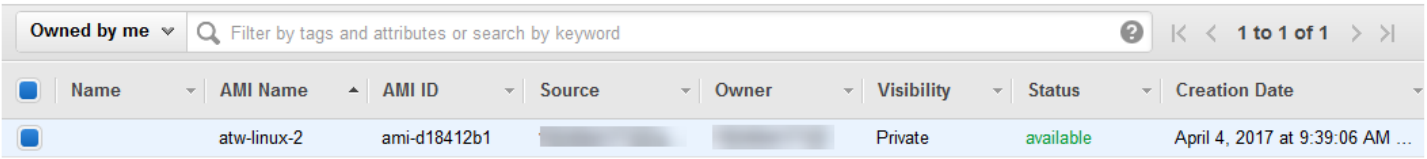
2. 在 [建立影像] 對話方塊中，輸入唯一的名稱和描述，然後選擇 [建立影像]。根據預設，Amazon EC2 會關閉執行個體，拍攝任何已連接磁碟區的快照，建立並註冊 AMI，然後重新開機執行個體。如果您不想關閉執行個體，請選擇 [不重新啟動]。

> **Warning**
>
> 若您選擇 No reboot (不重新開機)，我們無法保證建立映像的檔案系統完整性。



建立影像對話方塊

建立 AMI 可能需要幾分鐘。建立之後，它會出現在AWS檔案總管的 AMI 檢視中。若要顯示此檢視，請在AWS資源管理器中按兩下 Amazon EC2 | AMI 節點。若要查看您的 AMI，請從 [檢視] 下拉式清單中選擇 [由我擁有]。您可能需要選擇「重新整理」才能查看您的 AMI。當 AMI 第一次出現時，它可能處於待處理狀態，但稍後，它會轉換到可用狀態。



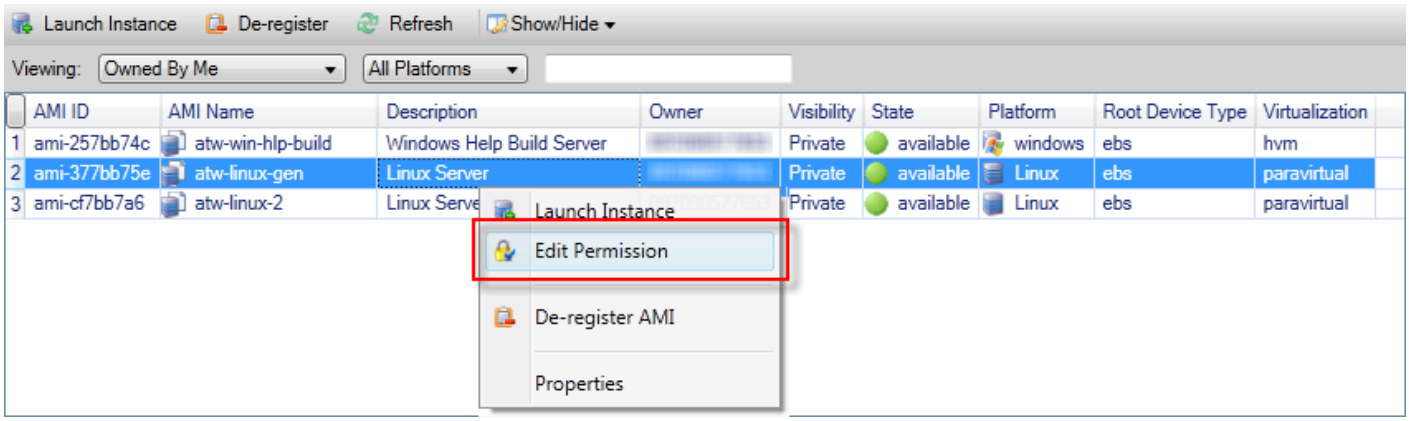
已建立的 AMI 清單

設定 Amazon Machine Images

您可以從AMI在視AWSExplorer 您可以使用設定 AMI 許可對話框從 AMI 複製權限。

設定 AMI 的許可

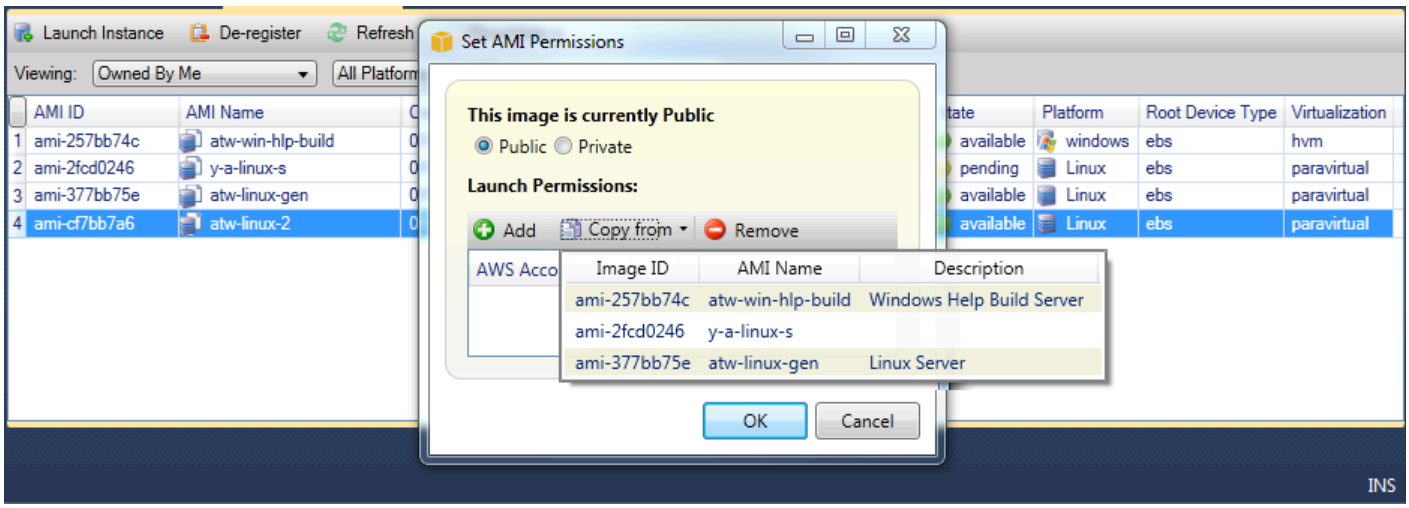
1. 在 中AMI在視AWS，開啟 AMI 上的上下文 (按一下滑鼠右鍵)，然後選擇編輯許可。



2. 有三個選項可用設定 AMI 許可對話方塊：

   - 要授予啟動權限，請選擇Add，然後鍵入AWS您要向其授予啟動權限的用户。
   - 要刪除啟動權限，請選擇AWS要刪除啟動權限的用户，然後選擇Remove (移除)。
   - 要將許可從 AMI 複製到另一個 AMI，從清單中選取 AMI，然後選擇從 COPY。對您選擇的 AMI 具有啟動權限的用户將獲得當前 AMI 的啟動權限。您可以將此過程與其他 AMI 重複在從 COPY列表將多個 AMI 中的權限複製到目標 AMI 中。

     所以此從 COPY列表中僅包含那些由帳户所擁有的 AMI，當AMI視圖顯示來自AWSExplorer 因此，從 COPY列表可能不會顯示任何 AMI，如果活動帳户沒有其他 AMI 所有。



複製 AMI 許可對話方塊

Amazon Virtual Private Cloud (VPC)

Amazon Virtual Private Cloud (Amazon VPC) 可讓您將 Amazon Virtual Private Cloud (Amazon Web Services VPC) 啟動到您定義的虛擬網路。這個虛擬網路與您在資料中心中操作的傳統網路相似，且具備使用AWS。如需詳細資訊，請前往Amazon VPC User Guide。

Visual Studio 工具包使開發人員能夠訪問 VPC 功能類似於AWS Management Console而是來自視覺工作室開發環境。所以此Amazon VPC節點AWS資源管理器包括下列區域的子節點。

- 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

創建用於部署的公私 VPCAWS Elastic Beanstalk

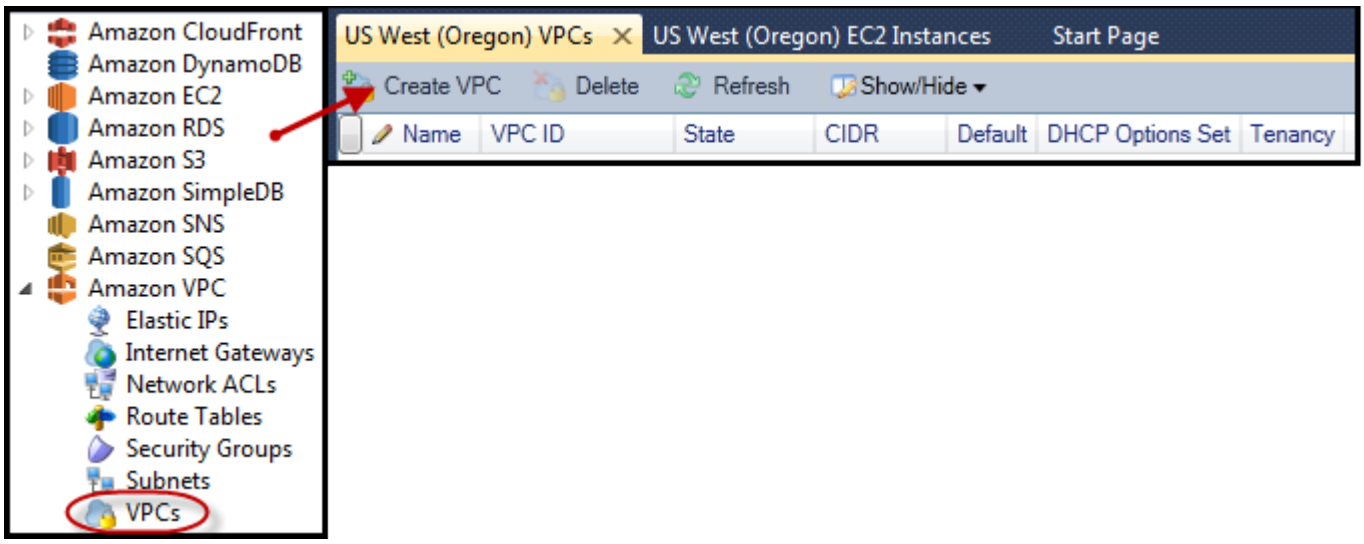
本節將介紹如何建立同時含公有和私有子網路的 Amazon VPC。公有子網路包含 Amazon EC2 實例，其執行網路位址轉譯 (NAT)，以使私有子網路中的執行個體與公有網路通訊。這兩個子網路必須位於同一個可用區域 (AZ)。

這 VPC 部署AWS Elastic Beanstalk環境。在這種情況下，託管您的應用程序的 Amazon EC2 實例駐留在私有子網中；用於將傳入流量路由到應用程序的 Elastic Load Balancing 器位於公有子網中。

如需網路位址轉譯 (NAT) 的詳細資訊，請前往NAT 執行個體中的Amazon Virtual Private Cloud 用户指南。如需如何將部署配置為使用 VPC 的範例，請參部署至 Elastic Beanstalk。

創建公私子網 VPC

1. 在 中Amazon VPC中的節點AWS資源管理器中，打開VPC子節點，然後選擇建立 VPC。

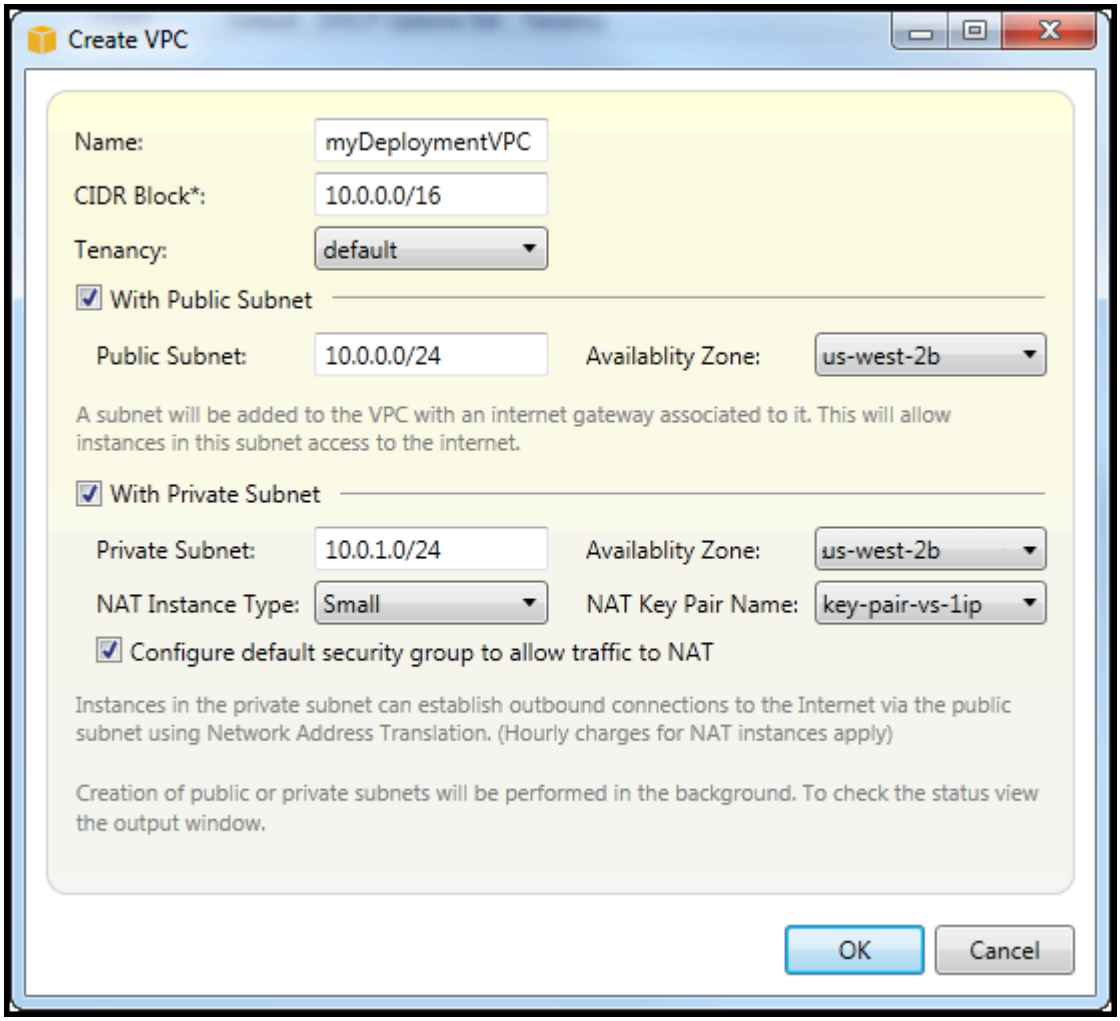


2. 按如下方式配置 VPC：

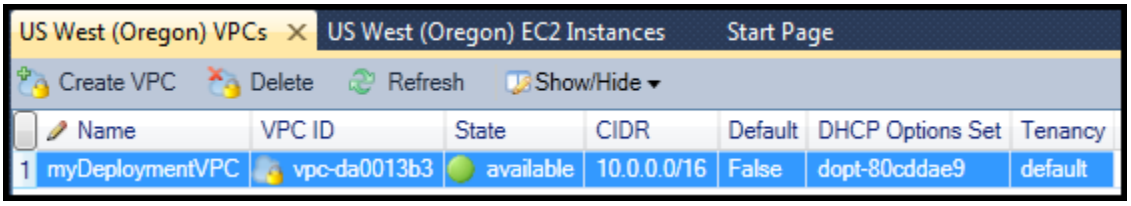
   - 輸入 VPC 的名稱。
   - 選取使用公有子網路與私有子網路核取方塊。
   - 從可用區域下拉列表框中，選擇一個可用區。確保為兩個子網使用相同的可用區。
   - 對於私有子網，請在NAT 金鑰對名稱，提供金 key pair。此 key pair 用於執行從私有子網到公有互聯網的網絡地址轉換的 Amazon EC2 實例。
   - 選取將默認安全組配置為允許流量到 NAT核取方塊。

   輸入 VPC 的名稱。選取使用公有子網路與私有子網路核取方塊。從可用區域下拉列表框中，選擇一個可用區。確保為兩個子網使用相同的可用區。對於私有子網，請在NAT 金鑰對名稱，提供金 key pair。此 key pair 用於執行從私有子網到公有互聯網的網絡地址轉換的 Amazon EC2 實例。選取將默認安全組配置為允許流量到 NAT核取方塊。

   選擇 OK (確定)。

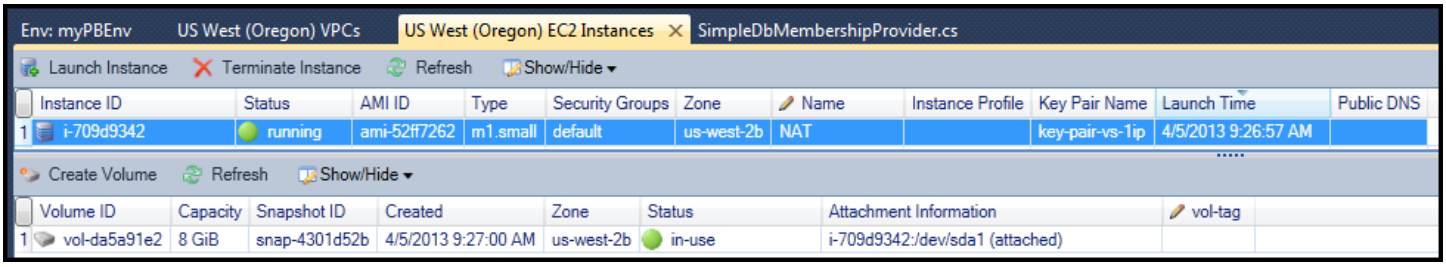


您可 VPC 在VPC標籤AWSExplorer。



NAT 實例可能需要幾分鐘的時間來啟動。當它可用時，您可以通過展開Amazon EC2中的節點AWS資源管理器，然後打開執行個體子節點。

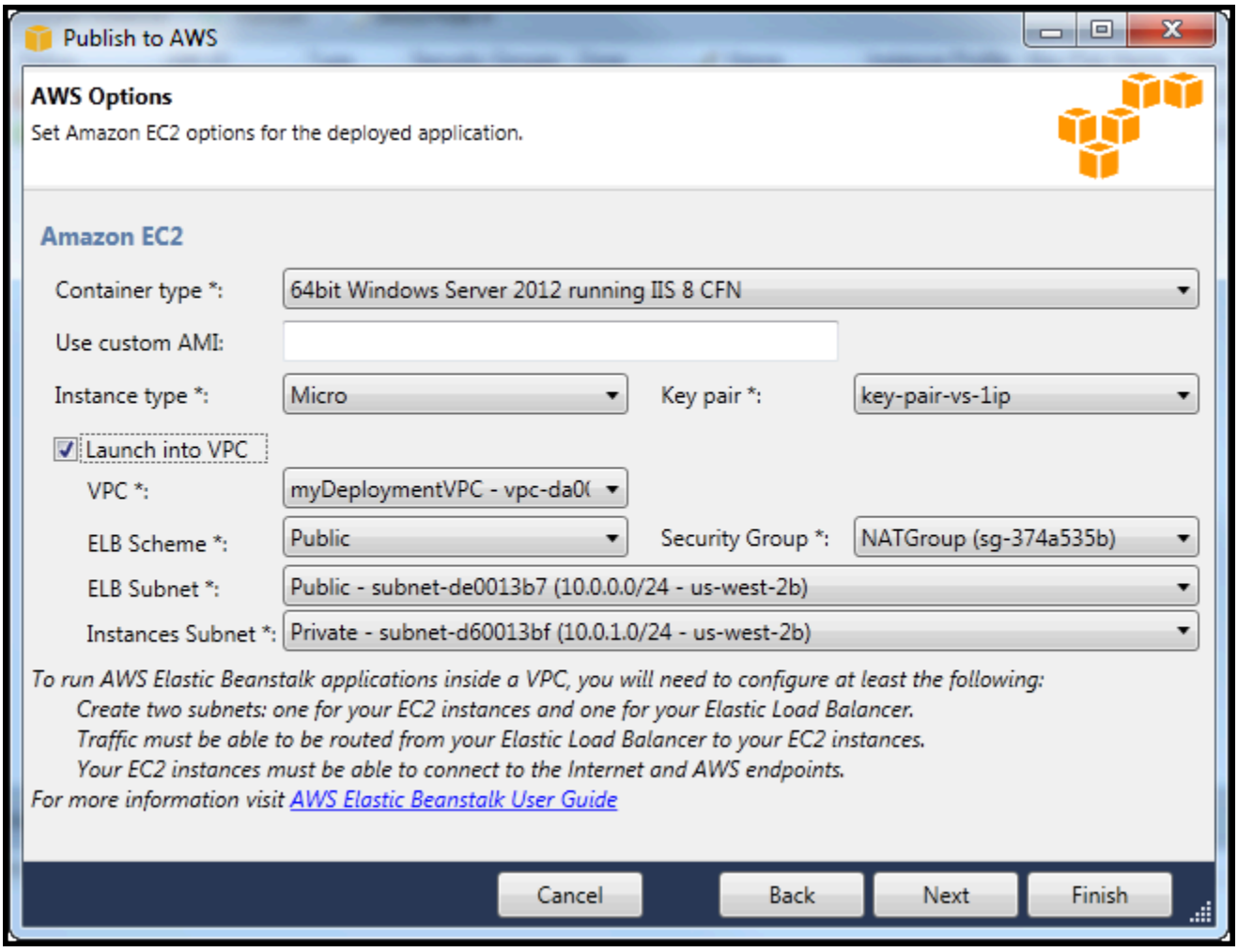
同時AWS Elastic Beanstalk（Amazon EBS）卷將為 NAT 實例自動建立。如需 Elastic Beanstalk 的詳細資訊，請前往AWS Elastic Beanstalk(EBS)中的Amazon EC2 Linux 執行個體使用者指南。



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，則工具包將填充發佈至Amazon Web Services對話框，其中包含 VPC 的配置信息。

工具包將僅使用來自工具包中創建的 VPC 的信息填充對話框，而不是使用AWS Management Console。這是因為 Toolkit 創建 VPC 時，它會標記 VPC 的組件，以便它可以訪問它們的信息。

部署嚮導的以下屏幕截圖顯示了一個對話框示例，該對話框中使用在 Toolkit 中創建的 VPC 中創建的值填充。



刪除 VPC

要刪除 VPC，您必須先終止 VPC 中的所有 Amazon EC2 實例。

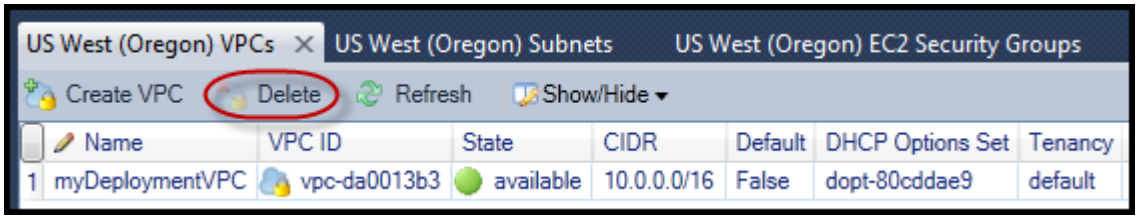
1. 如果已將應用程序部署到AWS Elastic Beanstalk環境中 VPC 請刪除環境。這將終止託管您的應用程序的所有 Amazon EC2 實例以及彈性負載均衡器。

   如果您嘗試直接終止託管應用程序的實例而不刪除環境，Auto Scaling 服務將自動創建新實例以替換已刪除的實例。如需詳細資訊，請前往Auto Scaling 開發者指南。

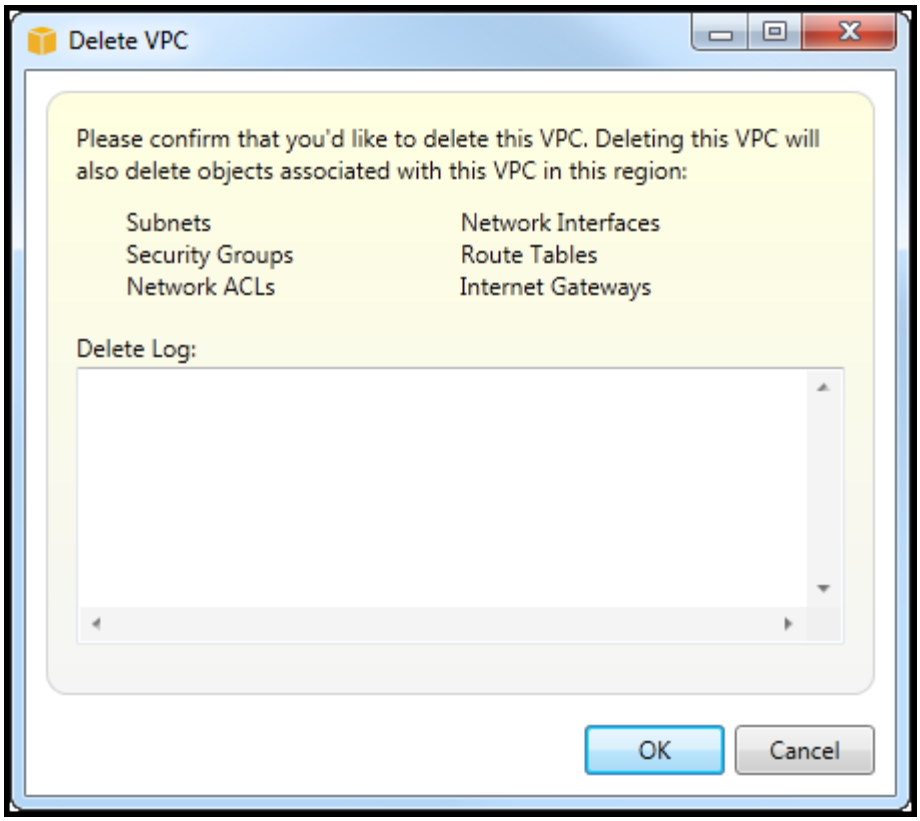
2. 刪除 VPC 的 NAT 實例。

   您無需刪除與 NAT 實例關聯的 Amazon EBS 卷即可刪除 VPC。但是，如果您不刪除該卷，即使您刪除了 NAT 實例和 VPC，也將繼續為其付費。

3. 在VPC選項卡上選擇刪除鏈接來刪除 VPC。



4. 在 中刪除 VPC對話方塊中，選擇確定。



使用 AWS CloudFormation 範本編輯器

Toolkit for Visual Studio 包括 AWS CloudFormation 範本編輯器和 AWS 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範本 (空白或從現有堆疊或範例 AWS CloudFormation 範本複製)。
- 使用自動 JSON 驗證、自動完成、程式碼折疊和語法醒目提示來編輯範本。
- 自動建議範本中欄位值的內建函數和資源參考參數。
- 功能表項目，可從 Visual Studio 為範本執行一般動作。

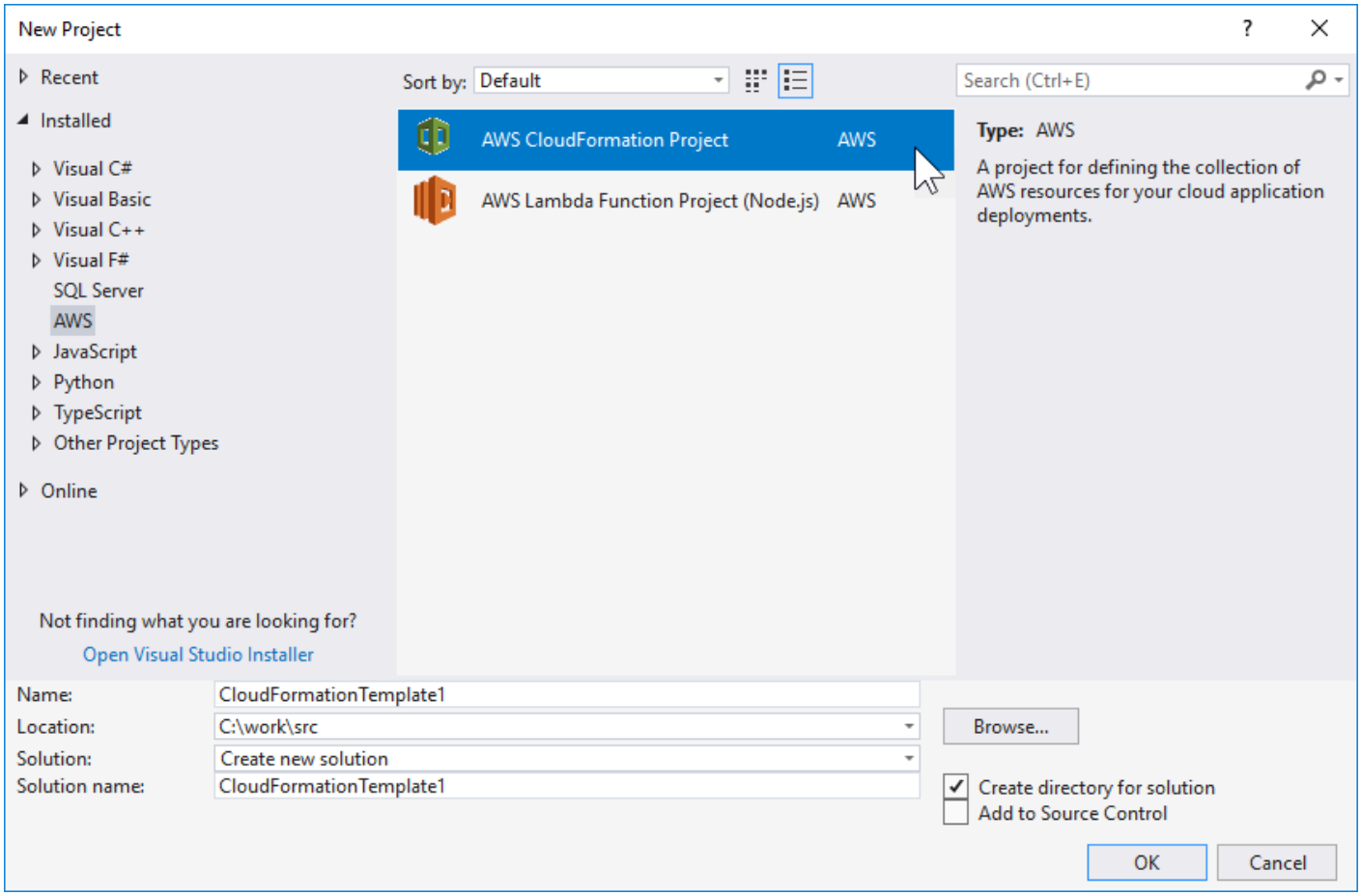
主題

建立AWS CloudFormation視覺工作室中的模板項目

創建模板項目

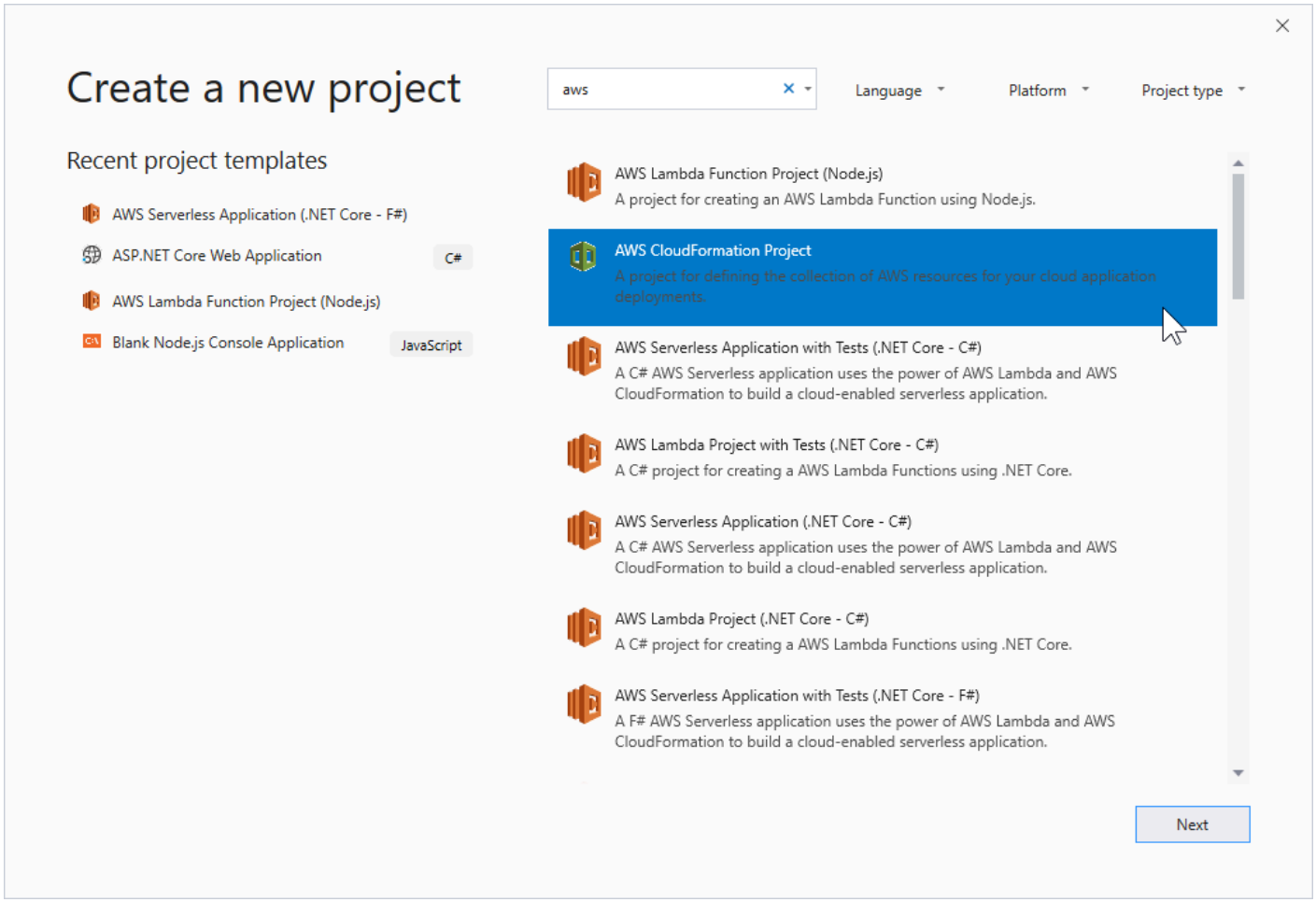
1. 在 Visual Studio 中，選擇File (檔案)，選擇新的，然後選擇專案。
2. 若為 Visual Studio 2017：

   在 中新的專案對話方塊中，展開已安裝並選擇AWS。



若為 Visual Studio 2019：

在 中新的專案對話框中，確保語言、平台，以及專案類型下拉框設置為「所有...」，然後鍵入aws中的搜尋欄位。

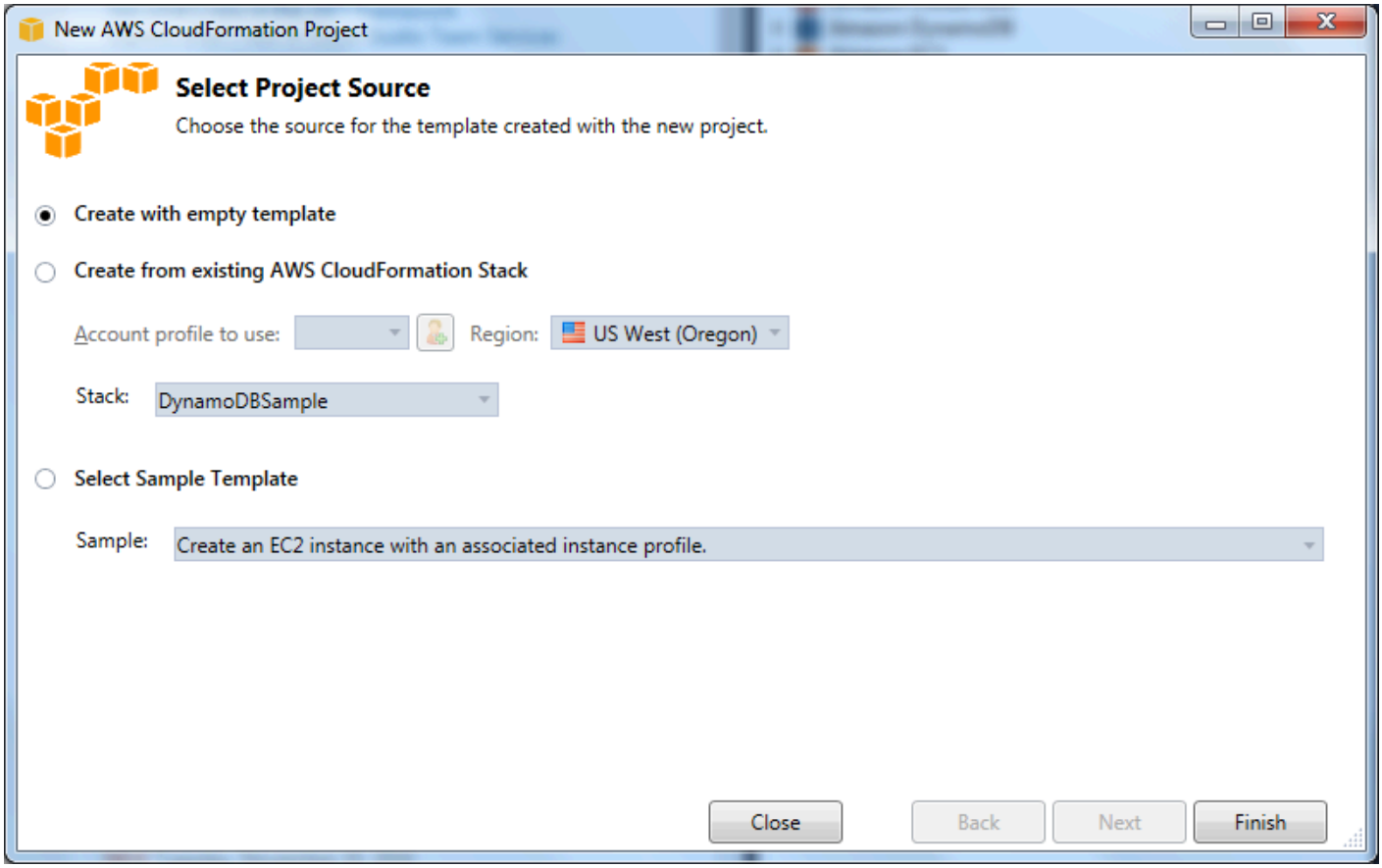


3. 選取AWSCloudFormation 專案範本。
4. 若為 Visual Studio 2017：

   輸入所需的名稱、位置等，為模板項目，然後單擊確定。

   若為 Visual Studio 2019：

   按一下 Next (下一步)。在下一個對話框中，輸入所需名稱、位置等，為模板項目，然後單擊建立。
5. 在選取專案來源頁面上，請選擇要建立範本的來源：
   - 使用空模板創建生成新的、空白的AWS CloudFormation範本。
   - 使用現有建立範本AWS| CFN | 堆棧使用現有堆疊建立範本AWS帳戶。（堆棧不需要具有CREATE_COMPLETE。）
   - 選取範例範本生成一個模板，從AWS CloudFormation範例範本。

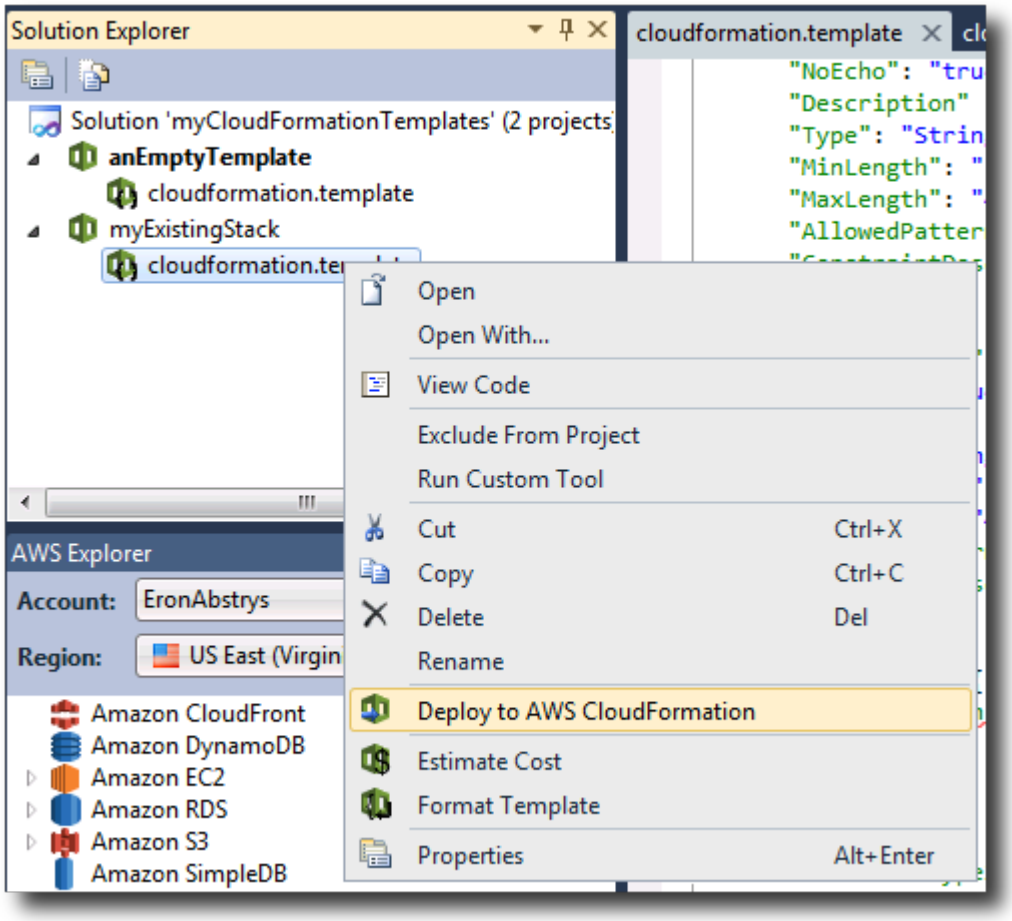


6. 要完成您的AWS CloudFormation模板項目中，選擇完成。

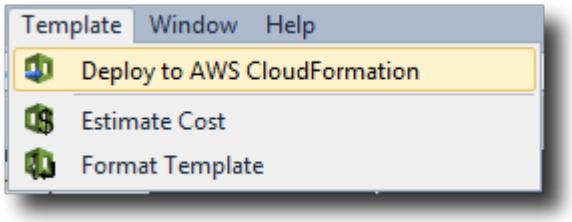
部署AWS CloudFormationVisual Studio 中的範本

部署 CFN 模板

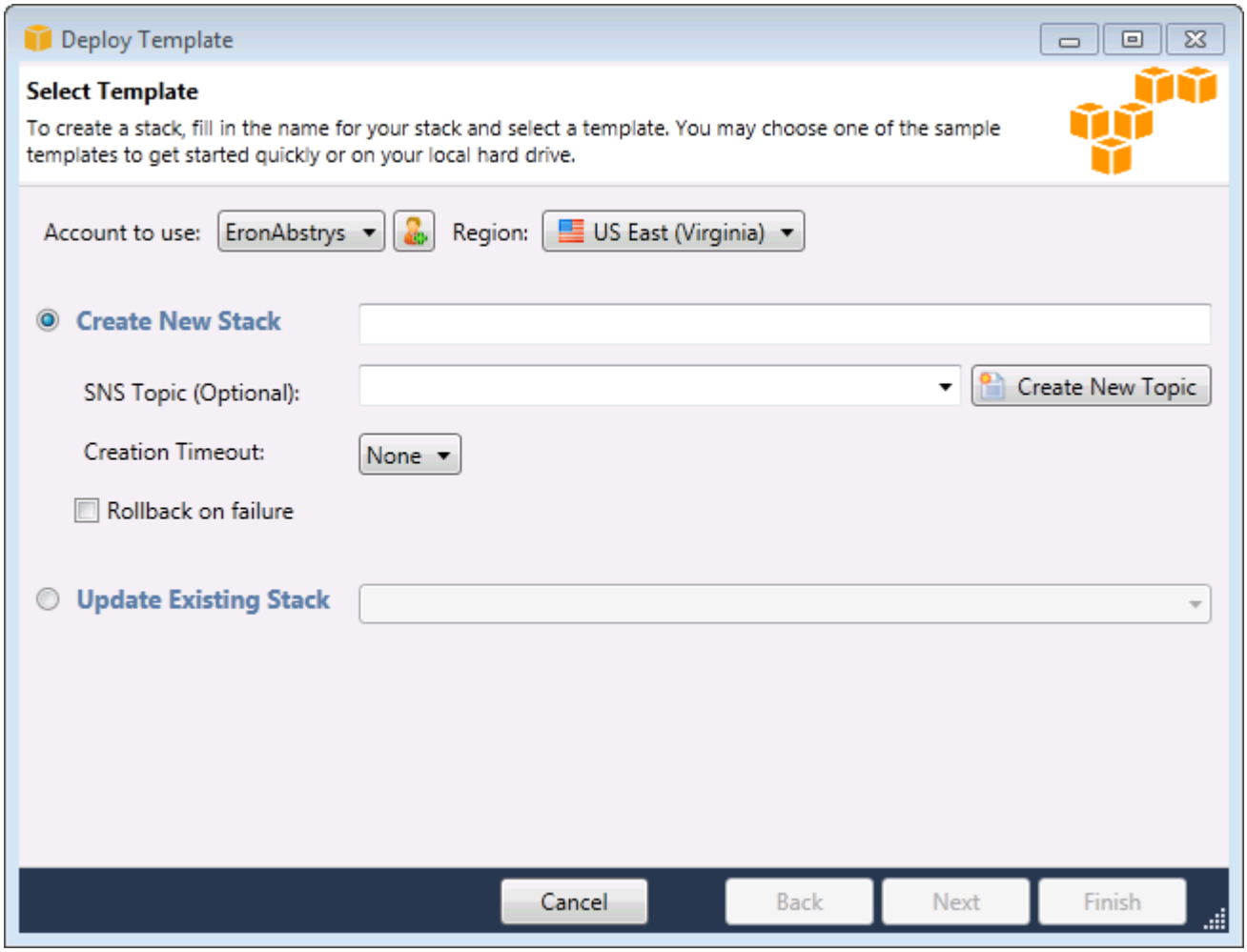
1. 在 Creation Explorer 中，開啟您想要部署的模板的內容 (按一下滑鼠右鍵) 選單，然後選擇部署至 AWS CloudFormation。



或者，要部署您當前正在編輯的模板，請從Template (範本)菜單中，選擇部署至AWS CloudFormation。



2. 在部署模板頁面上，選擇AWS 帳戶用於啟動堆棧和將啟動堆棧的區域。

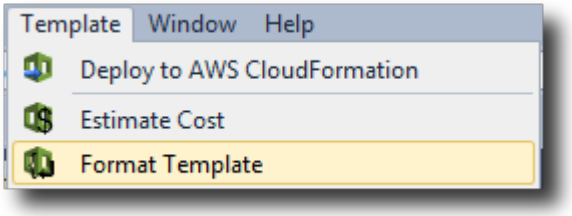


3. 選擇建立新堆疊，然後輸入堆疊名稱。
4. 選擇下列其中任何一個選項 (或不選擇)：
   - 若要接收堆疊進度的通知，請從SNS 主題下拉式清單中，選擇 SNS 主題。您也可以創建 SNS 主題，方法是選擇建立新主題並在框中輸入電子郵件地址。
   - 使用建立逾時以指定多長AWS CloudFormation應允許在堆疊宣告失敗 (並且進行還原，除非Rollback on failure (在發生故障時轉返)選項被清除）。
   - 使用Rollback on failure (在發生故障時轉返)如果您想要堆疊在發生故障時還原（即刪除其本身）。如果您想要進行偵錯，讓堆疊維持啟用 (即便啟動失敗)，請不要核取這個選項。
5. 選擇完成以啟動堆疊。

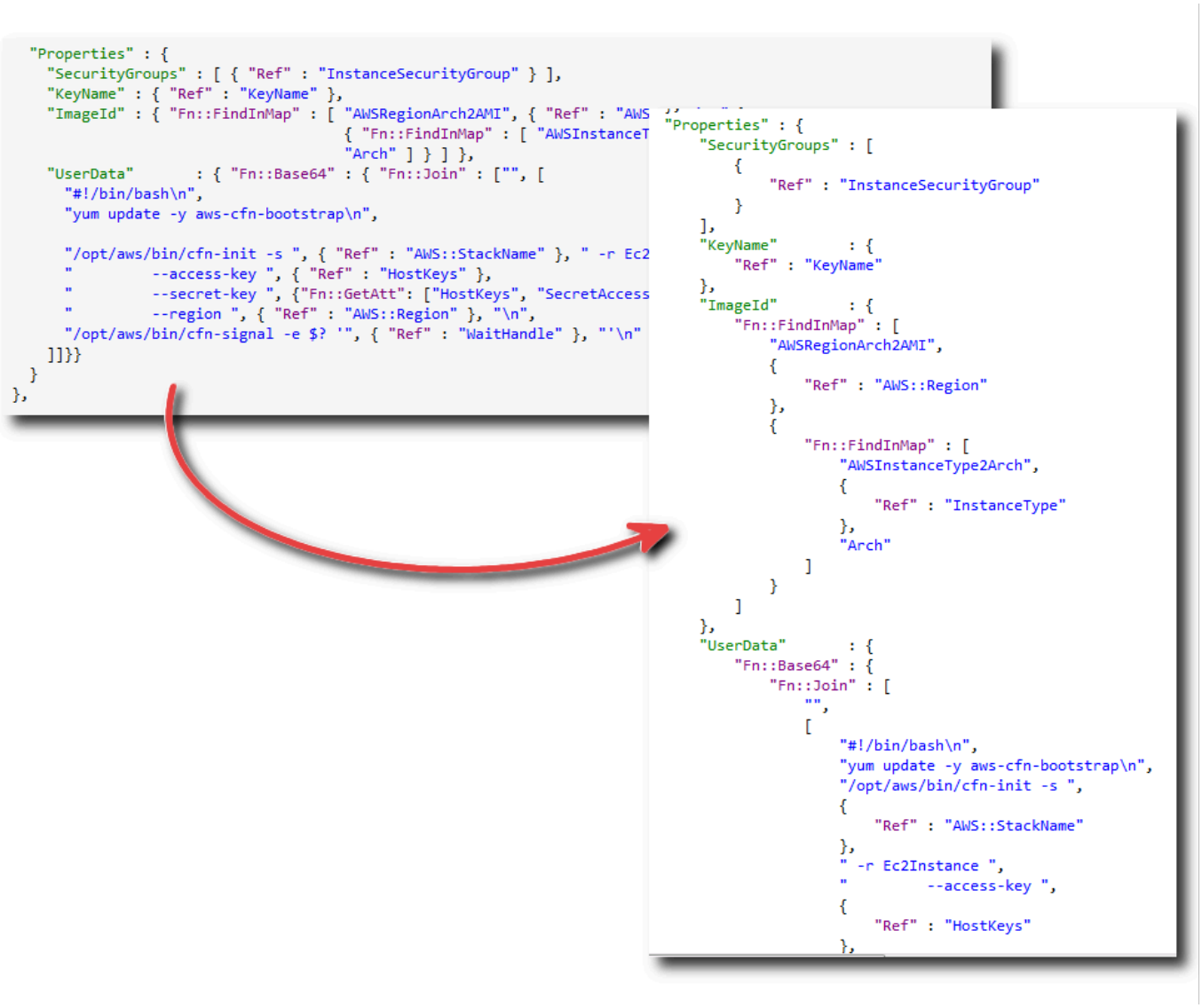
格式AWS CloudFormationVisual Studio 中的範本

- 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。

或者，要設置當前正在編輯的模板的格式，請從Template (範本)菜單中，選擇格式。



您的 JSON 代碼將被格式化，以便清楚地顯示其結構。



使用 Amazon S3AWS探險者

Amazon Simple Storage Service (Amazon S3) 使您能夠儲存和檢索任何連接到互聯網的數據。您存儲在 Amazon S3 上的所有數據都與您的賬户相關聯，默認情況下，只能由您訪問。使用適用 Toolkit for Visual Studio，您可以在 Amazon S3 上存儲數據，並查看、管理、檢索和分發這些數據。

Amazon S3 使用存儲桶的概念，您可以將其視為類似於文件系統或邏輯驅動器。存儲桶可以包含類似於目錄和對象的文件夾，這些文件夾和對象類似於文件。在本節中，我們將在演示 Toolkit for Visual Studio 公開的 Amazon S3 功能時使用這些概念。

> **Note**
>
> 若要使用此工具，您的 IAM 策略必須授予`s3:GetBucketAcl`、`s3:GetBucket`，以及`s3:ListBucket`動作。如需詳細資訊，請參閱「」概觀AWSIAM 政策。

建立 Amazon S3 儲存貯體

儲存貯體是 Amazon S3 中最基本的儲存體。

建立 S3 儲存貯體

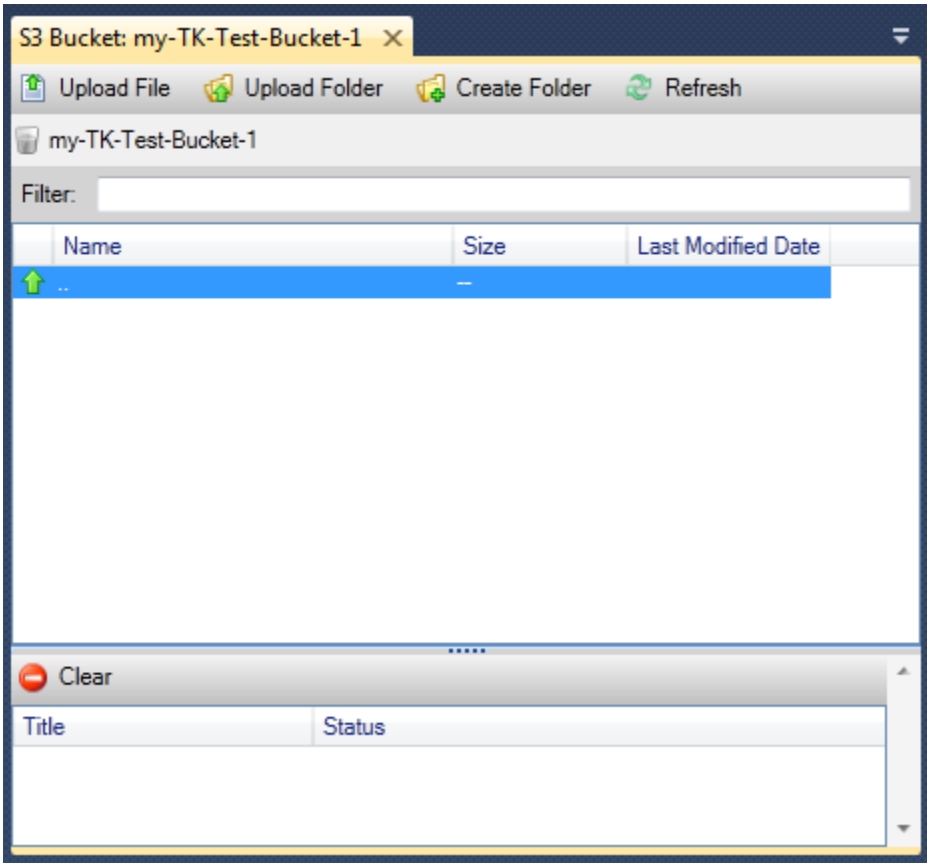
1. InAWSExplorer（按一下滑鼠右鍵）功能表，然後選擇Amazon S3節點，然後選擇建立儲存貯體。
2. 在 中建立儲存貯體對話方塊中，輸入儲存貯體的名稱。儲存貯體名稱必須在AWS。如需其他約束條件的詳細資訊，請前往Amazon S3 檔。
3. 選擇 OK (確定)。

管理 Amazon S3 儲存貯體AWS探險者

InAWS資源管理器，當您開啟 (按一下滑鼠右鍵) Amazon S3 存儲存段的操作功能表。

瀏覽

顯示儲存貯體中包含的數據元的視圖。在此處，您可以從本地計算機創建文件夾或上傳文件或整個目錄和文件夾。下方窗格顯示有關上載過程的狀態消息。要清除這些消息，請選擇Clear圖示。您也可以通過雙擊AWSExplorer。



屬性

顯示一個對話方塊，您可以在其中執行以下操作：

- 將該範圍的 Amazon S3 權限設置為：
  - 您作為存儲桶擁有者。
  - 所有已通過身份驗證的用户AWS。
  - 每個人都有互聯網接入。
- 打開存儲桶的日誌記錄。
- 使用 Amazon 簡單通知服務 (Amazon SNS) 設置通知，以便如果您使用的是低冗餘存儲 (RRS)，在發生數據丟失時會收到通知。RRS 是 Amazon S3 存儲選項，它提供的持久性低於標準存儲，但成本更低。如需詳細資訊，請參閱「」S3 常見問答集。
- 使用儲存貯體中的數據建立靜態網站。

政策

使您能夠設置AWS Identity and Access Management(IAM) 策略。如需詳細資訊，請前往IAM 文件以及IAM和S3。

建立預先簽章 URL

使您能夠生成可分發的有時間限制的 URL，以便提供對存儲存內容的訪問權限。如需詳細資訊，請參閱「」如何建立預先簽章 URL。

查看多部分上傳

使您能夠查看分段上傳。Amazon S3 支持將大型數據元上傳分解為分段，以提高上傳過程的效率。如需詳細資訊，請前往S3 文檔中的分段上傳。

刪除

允許您刪除存儲桶。您僅能刪除空的儲存貯體。

上傳檔案和資料夾至 Amazon S3

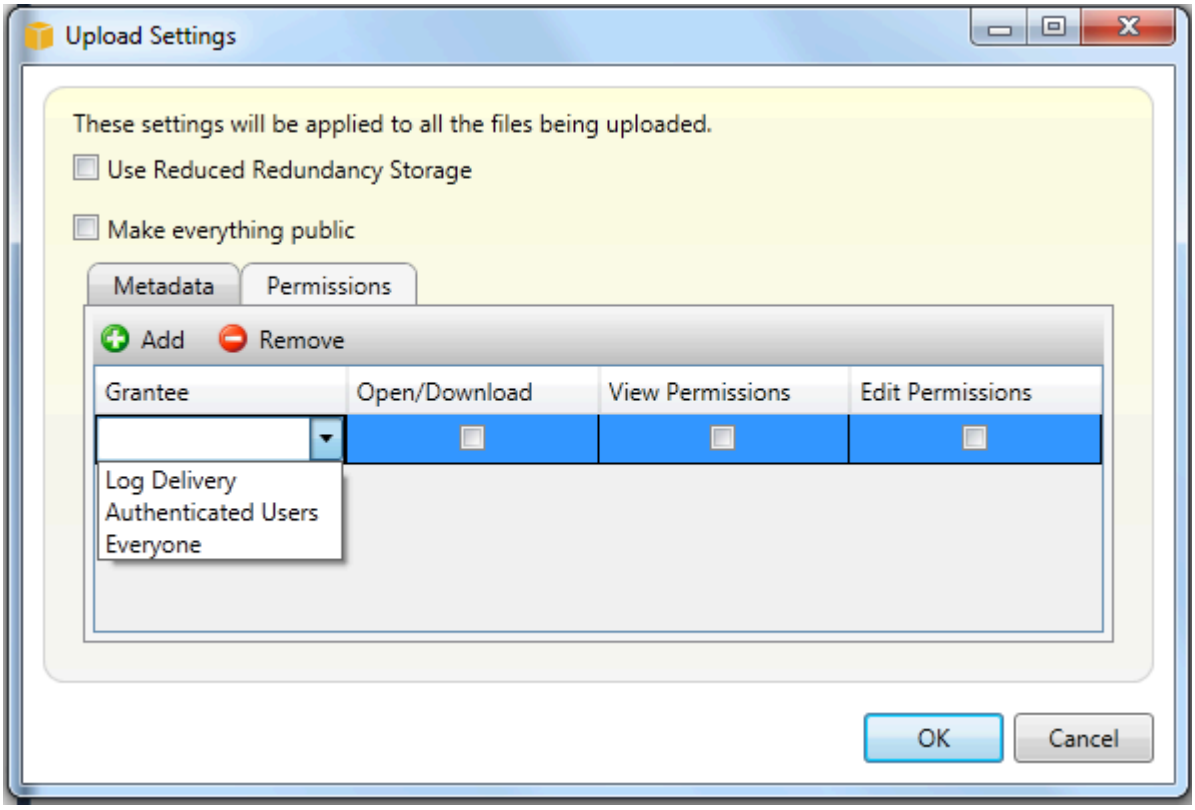
您可以使用AWS資源管理器將文件或整個文件夾從本地計算機傳輸到任何存儲桶。

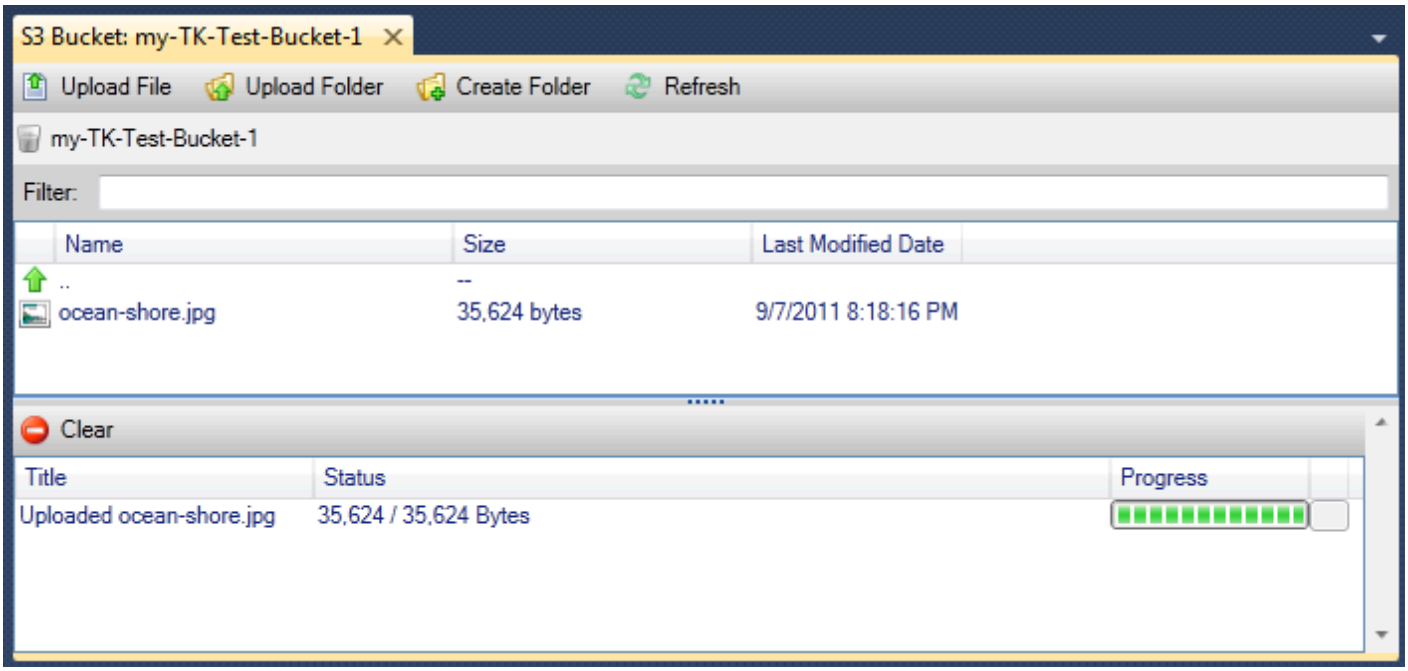
> **Note**
>
> 如果您上傳的文件或文件夾名稱與 Amazon S3 存儲桶中已存在的文件或文件夾具有相同名稱，您上傳的文件將覆蓋現有文件，而不會發出警告。

上傳檔案至 S3

1. InAWS資源管理器中，展開Amazon S3節點，然後雙擊存放區或開啟內容功能表，然後選擇瀏覽。
2. 在 中瀏覽視圖中，選擇上傳檔或者上傳文件夾。
3. 在 中Open (檔案)對話方塊中，導覽至要上載的檔案，選擇它們，然後選擇開啟。如果要上傳文件夾，請導航到該文件夾並選擇該文件夾，然後選擇開啟。

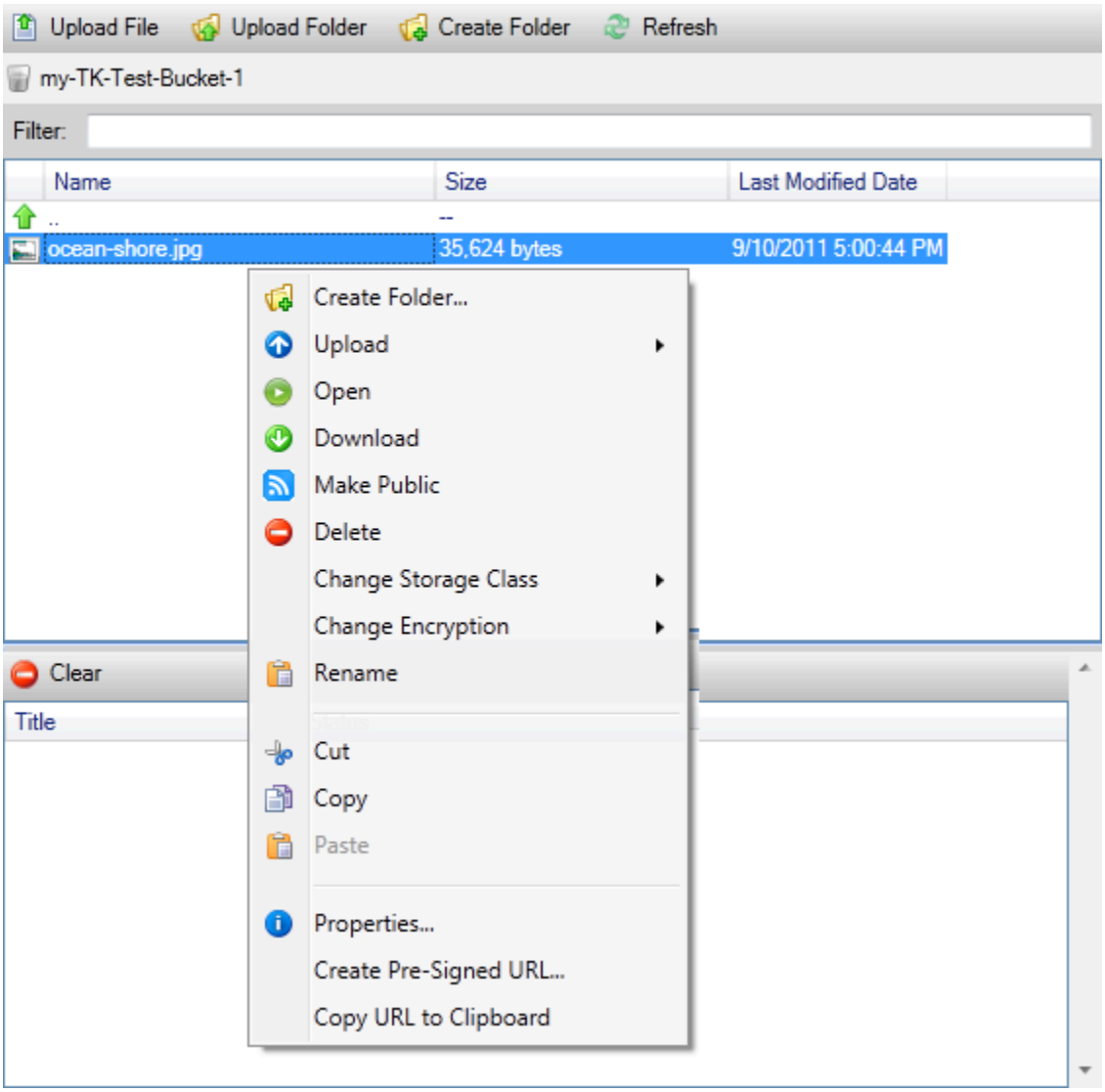
   所以此上傳設定對話框允許您設置元數據和對要上傳的文件或文件夾的權限。選取設定公有資料複選框等效於設置開啟/下載的許可每個人。您可以選擇使用低冗餘儲存對於上傳的文件。





來自 Amazon S3 檔案操作AWSToolkit for Visual Studio

如果您在 Amazon S3 視圖中選擇一個檔案，然後開啟 (按一下滑鼠右鍵) 內容功能表，則可以對該檔案執行各種操作。



建立資料夾

允許您在當前儲存貯體中建立資料夾。（相當於選擇建立資料夾鏈接。）

上傳

允許您上傳檔案或資料夾。（相當於選擇上傳檔或者上傳文件夾(鏈接)。

Open

嘗試在默認瀏覽器中開啟所選的檔案。根據文件類型和默認瀏覽器的功能，該文件可能不會顯示。它可能只是由您的瀏覽器下載。

下載

開啟資料夾樹對話框以允許您下載選定的文件。

設定公有資料

將選定文件的權限設置為開啟/下載和每個人。（相當於選擇設定公有資料複選方塊上傳設定對話方塊。）

刪除

刪除選定的文件或文件夾。您也可以刪除文件或文件夾，方法是選擇文件或文件夾並按Delete。

變更儲存方案

將存儲類別設置為標準或低冗餘儲存 (RRS)。要查看當前存儲類設置，請選擇屬性。

變更加密

允許您設定檔案的服務器端加密。若要查看當前加密設定，請選擇屬性。

Rename (重新命名)

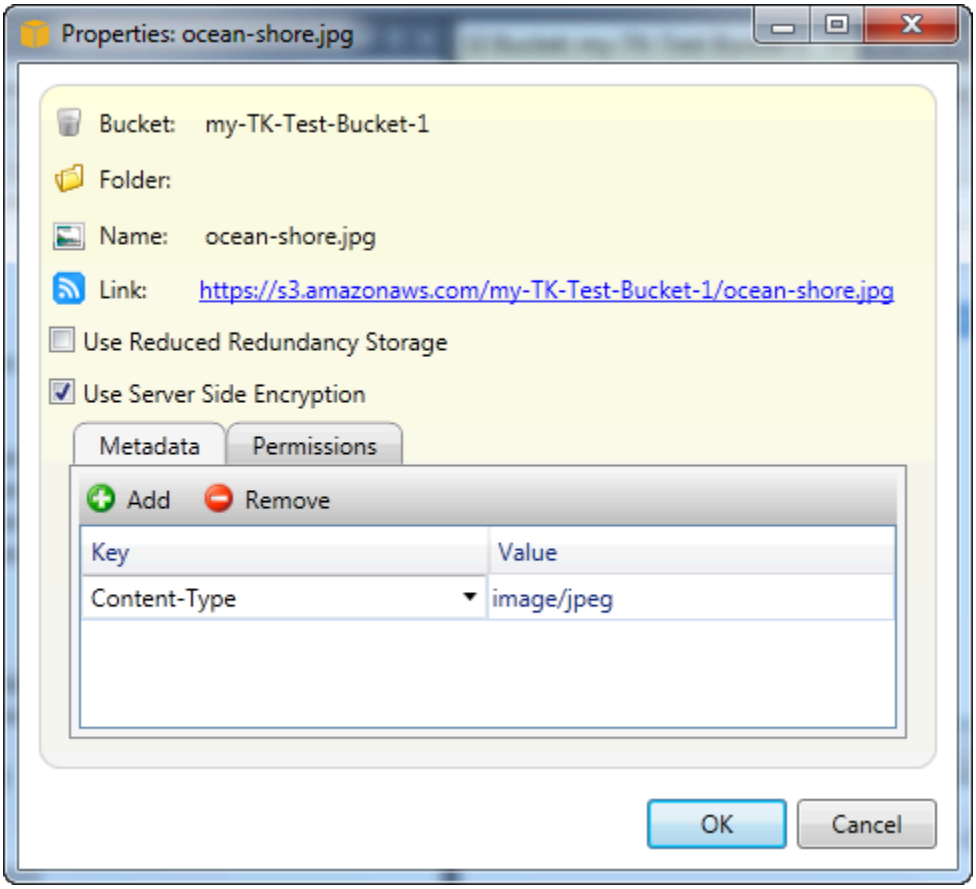
使您能夠重命名文件。您無法重新命名資料夾。

剪切 | 複製 | 粘貼

允許您在文件夾之間或存儲桶之間剪切、複製和粘貼文件或文件夾。

屬性

顯示一個對話框，使您能夠設置文件的元數據和權限，以及在低冗餘存儲 (RRS) 和標準之間切換文件的存儲，併為文件設置服務器端加密。此對話框還顯示指向該文件的 https 鏈接。如果選擇此鏈接，則 Toolkit for Visual Studio 將在默認瀏覽器中打開該文件。如果您對文件設置為開啟/下載和每個人，其他人將能夠通過此鏈接訪問該文件。我們建議您創建和分發預簽名 URL，而不是分發此鏈接。



建立預先簽章 URL

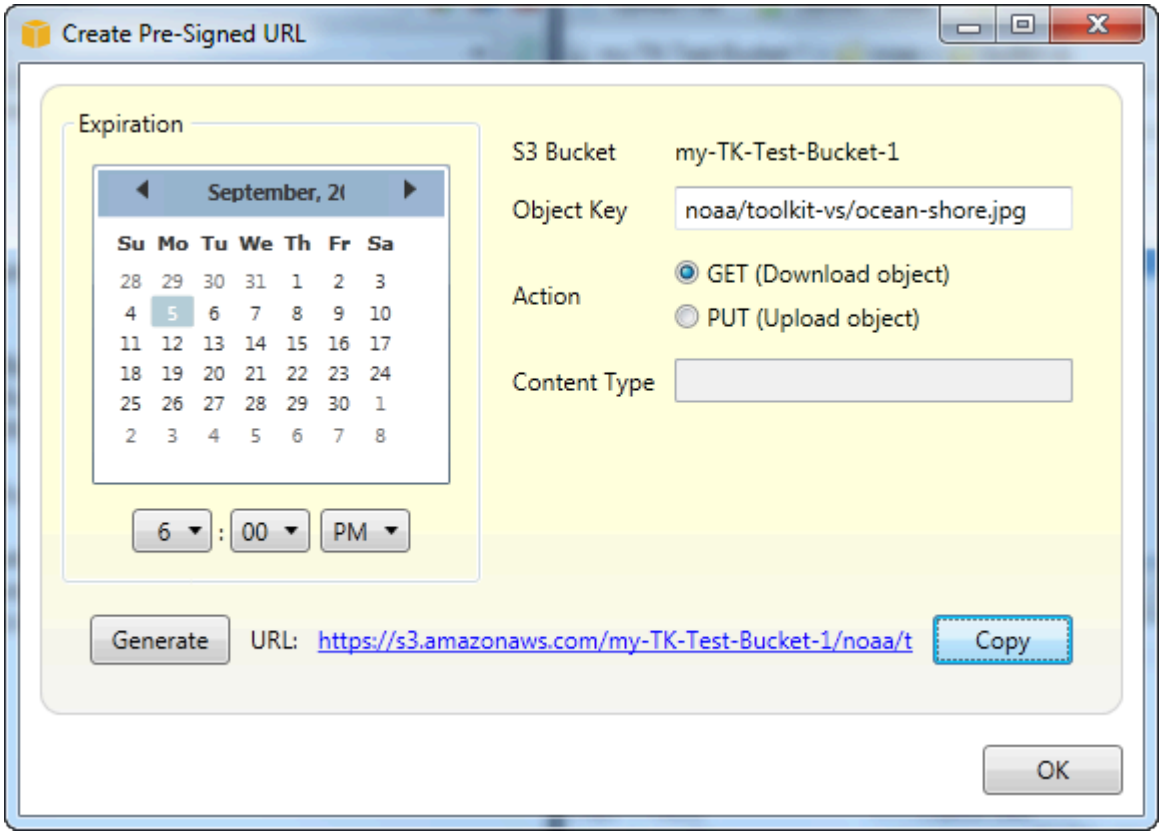
使您能夠創建一個有時間限制的預簽名 URL，您可以分發該 URL，使其他人能夠訪問您存儲在 Amazon S3 上的內容。

如何建立預先簽章 URL

您可以為存放區中的一個或多個檔案創建預先簽章 URL。然後，其他人可以使用此 URL 訪問存儲桶或文件。URL 將在您創建 URL 時指定的一段時間後便會過期。

若要建立預先簽章 URL

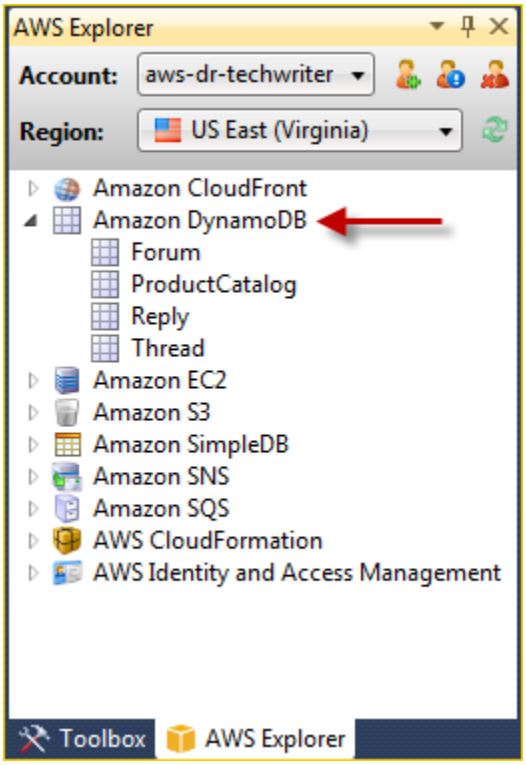
1. 在 中建立預先簽章 URL對話方塊中，設定 URL 的過期日期和時間。默認設定是從當前時間開始的 1 小時。
2. 選擇Generate按鈕。
3. 若要將 URL 複製到剪貼簿，請選擇複製。



使用 DynamoDBAWS探險者

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 移除資料儲存體中的傳統擴展性限制，同時保持了低延遲及可預期的效能。適用於 Visual Studio 的工具包提供了可在開發環境之下搭配使用 DynamoDB 的功能。如需 DynamoDB 的詳細資訊，請參DynamoDB」(位於 Amazon Web Services 網站)。

在 Toolkit for Visual Studio 中，AWS資源管理器會顯示與活動的AWS 帳戶。



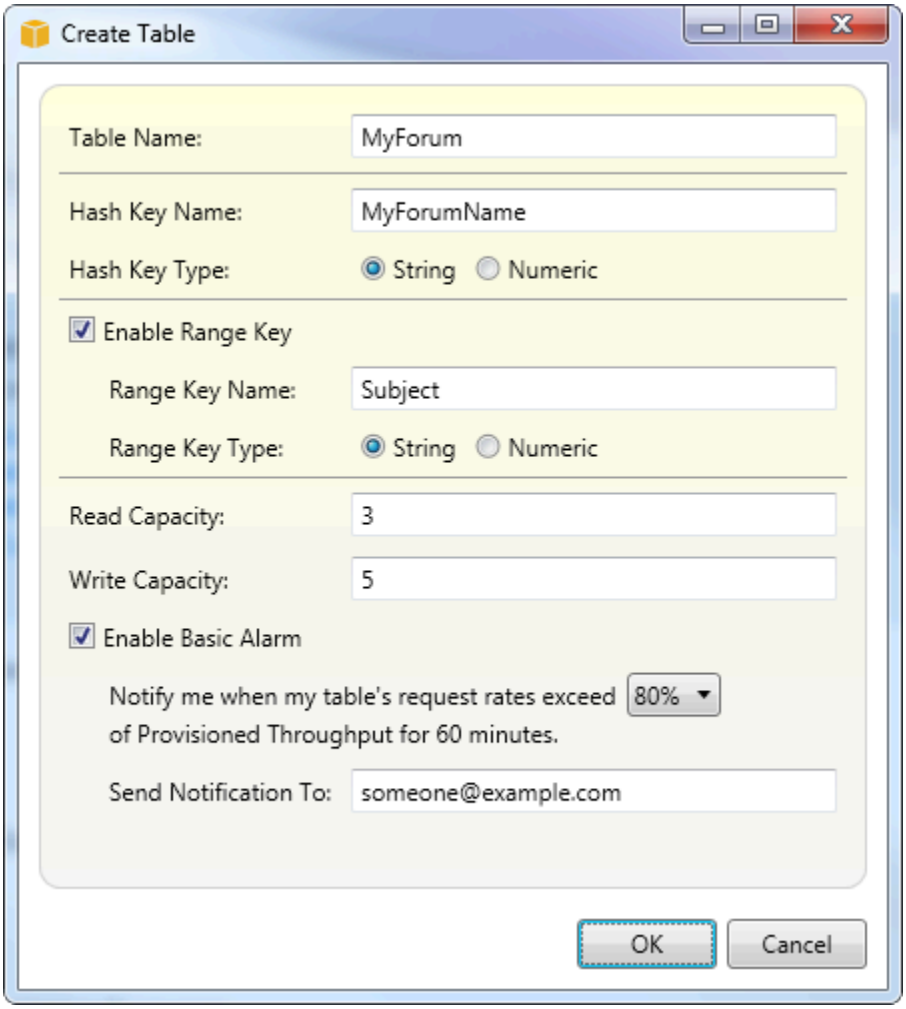
建立 DynamoDB 資料表

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

若要建立資料表AWS探險者

1. InAWSExplorer (按一下滑鼠右鍵) 功能表，開啟Amazon DynamoDB，然後選擇建立資料表。
2. 在 中建立資料表精靈 (位於)資料表名稱中，輸入資料表的名稱。
3. 在 中哈希鍵名稱字段中，鍵入主哈希鍵屬性，然後從哈希鍵類型按鈕，選擇散列鍵類型。DynamoDB 建立一個未排序的雜湊索引，使用的是使用範圍主索引鍵屬性的主索引鍵屬性和選用排序範圍索引。如需有關主要散列鍵屬性的詳細資訊，請前往主索引鍵區段中Amazon DynamoDB 開發人員指南。
4. (選用) 選取啟用範圍鍵。在 中範圍金鑰名稱字段中，鍵入範圍鍵屬性，然後從範圍鍵類型按鈕，選取範圍鍵類型。
5. 在 中讀取容量字段中，輸入讀取容量單位數目。在 中寫入容量字段中，輸入寫入容量單位的數目。您必須指定至少三個讀取容量單位和五個寫入容量單位。如需有關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，當表格的請求費率過高時提醒您。選擇在發送警報之前必須超過的每 60 分鐘預配置吞吐量的百分比。位於傳送通知到中，輸入電子郵件地址。

7. 按一下確定來建立資料表。

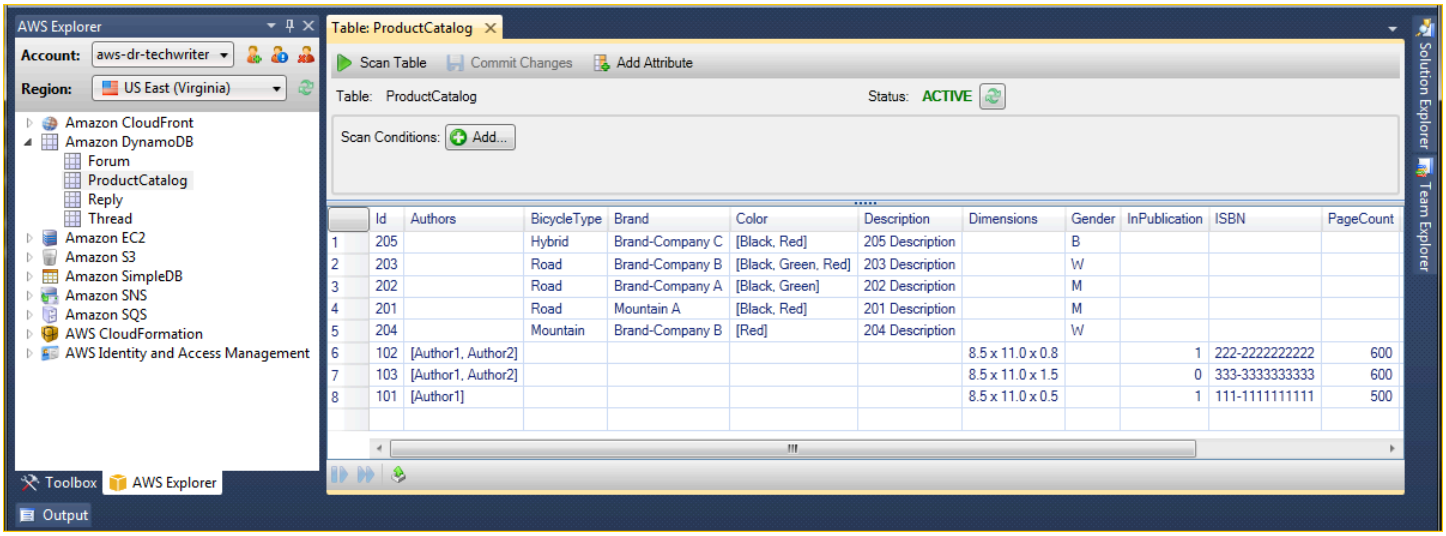


如需 DynamoDB 資料表的詳細資訊，請前往數據模型概念-資料表、項目與屬性。

以網格檢視 DynamoDB 資料表

若要開啟其中一份 DynamoDB 資料表的網格檢視，請在AWS瀏覽器中，按兩下與資料表對應的子節點。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。

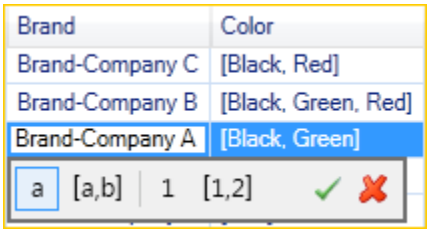
屬性可以擁有屬於字串或數字的值。有些屬性的值包含字串或數字的「集合」。集合值會顯示為逗號分隔的清單，並且用方括號括住。



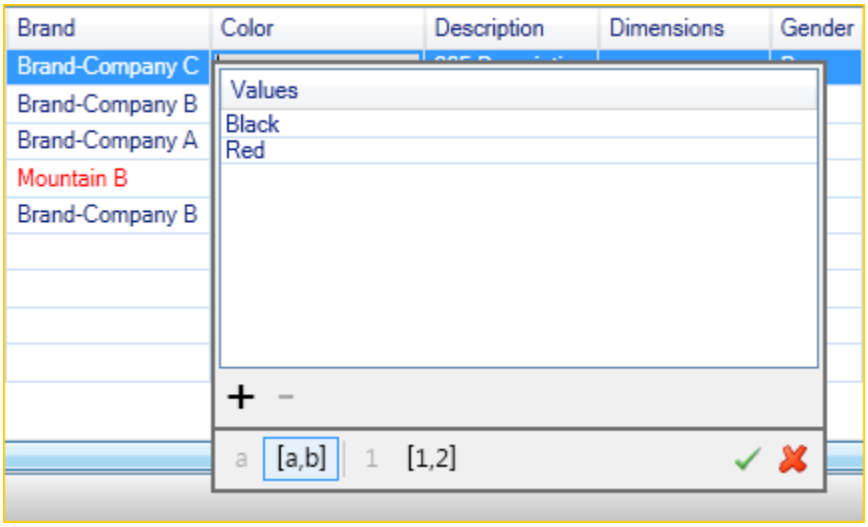
	Id	Authors	BicycleType	Brand	Color	Description	Dimensions	Gender	InPublication	ISBN	PageCount
1	205		Hybrid	Brand-Company C	[Black, Red]	205 Description		B			
2	203		Road	Brand-Company B	[Black, Green, Red]	203 Description		W			
3	202		Road	Brand-Company A	[Black, Green]	202 Description		M			
4	201		Road	Mountain A	[Black, Red]	201 Description		M			
5	204		Mountain	Brand-Company B	[Red]	204 Description		W			
6	102	[Author1, Author2]					8.5 x 11.0 x 0.8		1	222-2222222222	600
7	103	[Author1, Author2]					8.5 x 11.0 x 1.5		0	333-3333333333	600
8	101	[Author1]					8.5 x 11.0 x 0.5		1	111-1111111111	500

編輯和新增屬性和值

只要按兩下儲存格，您便可以編輯項目的對應屬性值。對於集合值屬性，您也可以在集合中新增或刪除其中的個別值。



除了變更屬性值之外，您還可以在某些限制之下變更屬性值的格式。例如，任何數值都能轉換為字串值。如果您有一個內容是數字的字串值，其內容是 125，則您可以使用單元格編輯器，將該值的格式從字串轉換成數值。您也可以將單一值轉換為設定值。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。

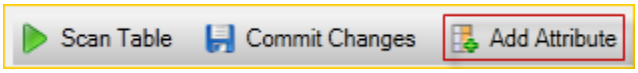


編輯屬性值之後，選擇綠色核取標記進行確認您的變更。如果您想要放棄您的變更，請選擇紅色 X。

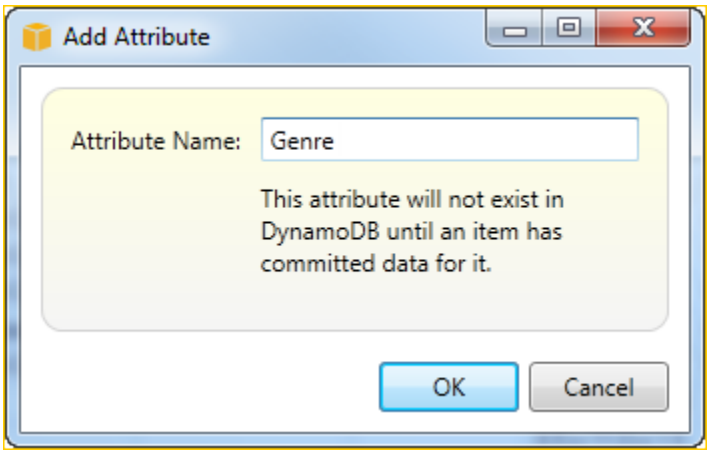
確認您的變更後，屬性值將以紅色顯示。這表示屬性已更新，但新的值尚未寫回 DynamoDB 資料庫中。要將更改寫回 DynamoDB，請選擇提交更改。若要放棄您的變更，請選擇掃描資料表並在工具組詢問您是否要在掃描前遞交變更時，選擇否。

新增屬性

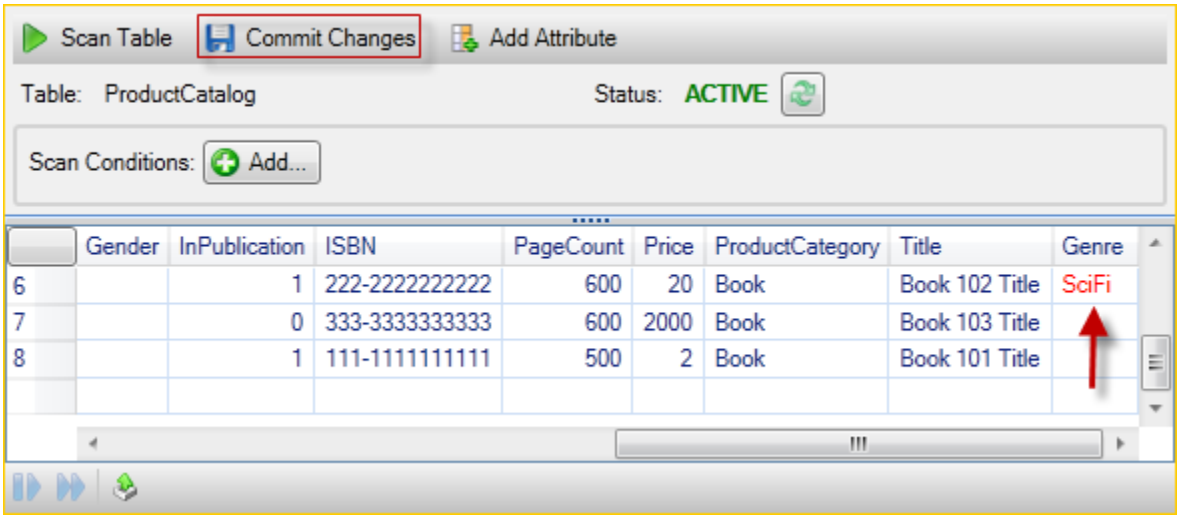
在網格檢視中，您還可以向資料表中添加屬性。若要新增屬性，請選擇新增屬性。



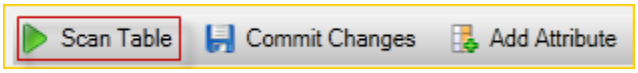
在 中新增屬性對話方塊中，輸入屬性的名稱，然後選擇確定。



要使新屬性成為表的一部分，您必須為其添加至少一個項目的值，然後選擇提交更改按鈕。要放棄新屬性，只需關閉表格的網格視圖，而不選擇提交更改。



掃描 DynamoDB 資料表

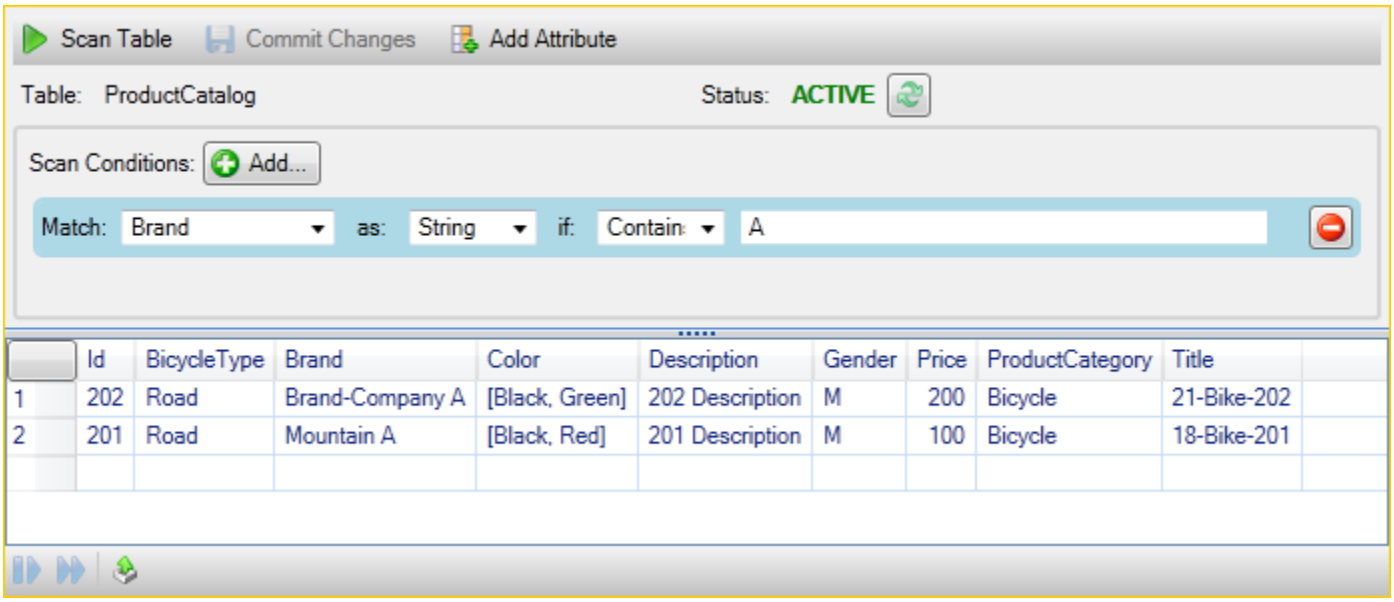


您可以從工具組中對 DynamoDB 資料表執行掃描。進行掃描時，您可以定義一組條件，而掃描會從資料表中傳回符合您所設定條件的所有項目。掃描作業非常耗用資源，因此應謹慎使用，避免中斷資料表處理更高優先性之生產工作的流量。如需如何使用掃描操作的詳細資訊，請前往Amazon DynamoDB 開發人員指南。

若要在 DynamoDB 資料表中執行掃描，請從AWS探險者

1. 在網格檢視中，選擇掃描條件：添加按鈕。
2. 在掃描子句編輯器中，選擇要比對的屬性、其值的解釋方式 (字串、數字、設置值)、其比對方式 (例如，以什麼開頭或包含內容) 及其應該和什麼常值進行比對。
3. 根據需要，新增更多 Scan 子句以進行搜索。掃描只會傳回符合您所有掃描子句之條件的項目。當比對字串值時，掃描會執行大小寫比較。
4. 在網格檢視頂部的按鈕欄中，選擇掃描資料表。

若要移除掃描子句，請選擇紅色按鈕，並選擇每個子句右邊的白線。



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，然後選擇掃描資料表。

為掃描結果編製分頁

檢視的底部有三個按鈕。

前兩個藍色按鈕可為掃描結果提供分頁。第一個按鈕將顯示一個附加的結果頁面。第二個按鈕將顯示另外十頁的結果。在這種情況下，頁面等於 1 MB 的內容。

將掃描結果導出到 CSV

第三個按鈕會將結果從當前掃描中輸出為 CSV 文件。

使用AWS CodeCommit與 Visual Studio Team Explorer

您可以使用AWS Identity and Access Management(IAM) 用户帳户創建 Git 證書，並使用它們從團隊資源管理器中創建和克隆存儲庫。

的登入資料類型AWS CodeCommit

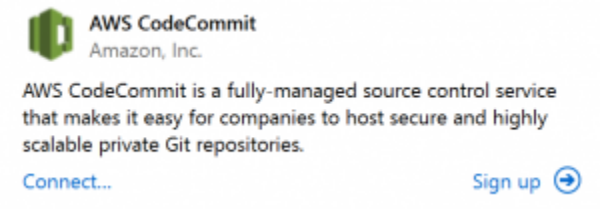
最AWS Toolkit for Visual Studio用户知道設置AWS包含其訪問密鑰和私有密鑰的憑據配置文件。這些登入資料配置文件用於 Visual Studio 以啟用對服務 API 的調用，例如，在AWS或啟動 Amazon EC2 執行個體。的整合AWS CodeCommit也使用這些憑據配置文件。但是，要使用 Git 本身，您需要額外的憑據，特別是 HTTPS 連接的 Git 憑據。您可以在使用 Git 登入資料的 HTTPS 使用者設定中的AWS CodeCommit使用者指南。

您可以創建 Git 憑據AWS CodeCommit僅適用於 IAM 用户賬户。您無法為 root 帳户創建它們。您最多可以為服務創建兩組這些憑據，儘管您可以將一組憑據標記為非活動，但非活動集仍會計入兩個集的限制。請注意您可以隨時刪除並重新建立登入資料。當您使用AWS CodeCommit從視覺工作室，您的傳統AWS憑據用於使用服務本身，例如，當您創建和列出存儲庫時。當使用託管的實際 Git 存儲庫AWS CodeCommit，您可以使用 Git 憑據。

作為對AWS CodeCommit，可視化工作室的工具包會自動為您創建和管理這些 Git 憑據，並將它們與AWS登入資料設定檔。在團隊資源管理器中執行 Git 操作時，您無需擔心掌握正確的憑據集。一旦你連接到團隊資源管理器AWS憑據配置文件，每當您使用 Git 遠程數據時，都會自動使用關聯的 Git 憑據。

連接到 AWS CodeCommit

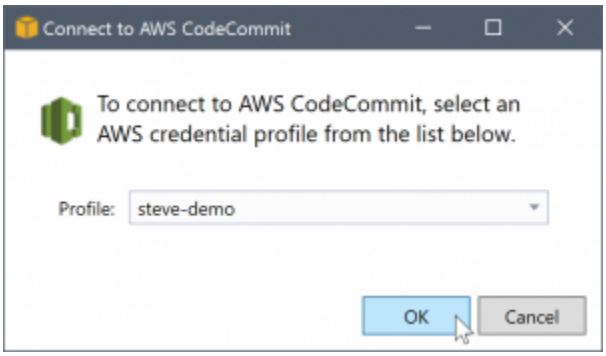
當您在視覺工作室 2015 年或更高版本中打開團隊資源管理器窗口時，您將看到AWS CodeCommit條目在「管理連接」的「託管服務提供程序」部分中。



選擇註冊會在瀏覽器視窗開啟 Amazon Web Services 首頁。當您選擇時會發生什麼情況連線取決於 Toolkit to Visual Studio 是否可以使用AWS訪問密鑰和私有密鑰，以使其能夠調用AWS會代表您。您可能已經通過使用新的入門頁面設置了憑據配置文件，該頁面在 Visual Studio 的 Toolkit 找不到任何本地存儲的憑據時，在 IDE 中顯示該頁面。或者您可能已經使用 Toolkit for Visual Studio，AWS Tools for Windows PowerShell，或AWS CLI並且已經擁有AWSToolkit 供 Visual Studio 使用的登入資料設定檔。

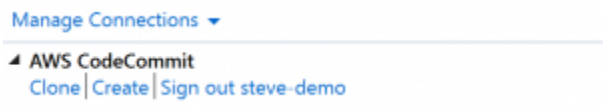
當您選擇連線時，Toolkit for Visual Studio 將啟動查找要在連接中使用的憑據配置文件的過程。如果 Toolkit for Visual Studio 找不到憑據配置文件，則會打開一個對話框，邀請您輸入AWS 帳戶。我們強烈建議您使用 IAM 使用者賬户，而非您的根登入資料。此外，如前所述，您最終需要的 Git 證書只能為 IAM 用户創建。提供訪問密鑰和私有密鑰並創建憑據配置文件後，Team Explorer 和AWS CodeCommit已準備就緒。

如果 TToolkit for Visual Studio 找到多個AWS登入資料檔時，系統將提示您選取您要在 Team Explorer 中使用的帳户。



如果您只有一個憑據配置文件，Toolkit for Visual Studio 將繞過配置文件選擇對話框並立即連接：

當團隊資源管理器和AWS CodeCommit，邀請對話框將關閉，並顯示連接面板。

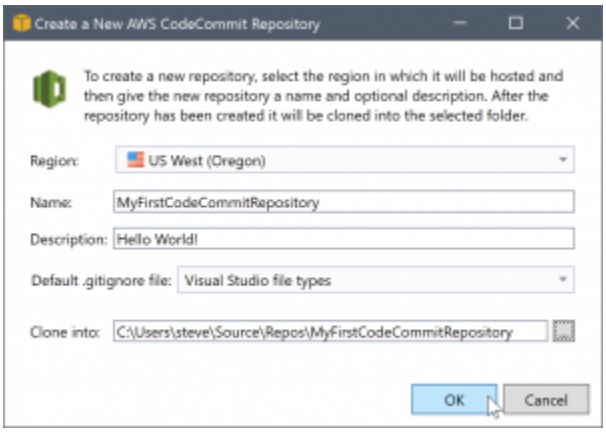


由於您沒有在本地克隆的存儲庫，因此面板僅顯示您可以執行的操作：複製、建立，以及註銷。與其他提供者一樣,AWS CodeCommit只能綁定到單個AWS任何指定時間登入資料設定檔。要切換帳户，請使用註銷刪除連接，以便您可以使用其他帳户啟動新連接。

既然您已建立連線，就可以單擊建立鏈接。

建立儲存庫

當您單擊建立鏈接，建立新的AWS CodeCommit儲存庫對話方塊會開啟。



AWS CodeCommit存儲庫是按區域組織的，所以在Region (區域)您可以選擇儲存庫所在的區域。該列表中包含的所有區域AWS CodeCommit系統支援 。您可以為我們的新存儲庫提供名稱（必填）和說明（可選）。

對話框的默認行為是在新存儲庫的文件夾位置後綴為存儲庫名稱（輸入名稱時，文件夾位置也會更新）。要使用其他文件夾名稱，請編輯複製到文件夾路徑後輸入存儲庫名稱。

您也可以選擇自動建立初始的`.gitignore`檔案。所以此AWS Toolkit for Visual Studio為 Visual Studio 文件類型提供了一個內置默認值。您也可以選擇沒有文件或使用要跨存儲庫重複使用的自定義現有文件。只需選擇使用自訂，然後導航到要使用的自定義文件。

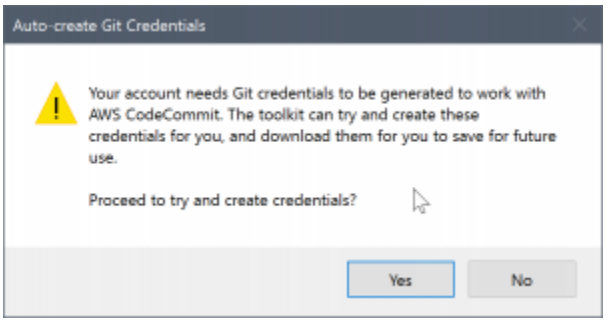
獲得存儲庫名稱和位置後，就可以單擊OK (OK)並開始創建存儲庫。Toolkit for Visual Studio 請求服務創建存儲庫，然後在本地克隆新存儲庫，同時添加為 .gitignore 文件的初始提交（如果您使用的話）。此時您開始使用 Git 遙控器，因此 Visual Studio 的工具包現在需要訪問前面描述的 Git 憑據。

設置 Git 憑據

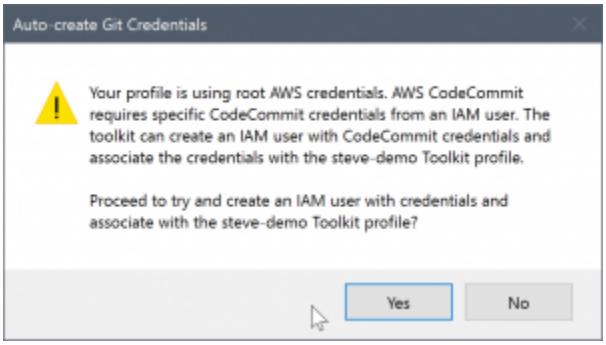
到目前為止，您一直在使用AWS訪問密鑰和私有密鑰來請求服務創建您的存儲庫。現在你需要使用 Git 本身來完成實際的克隆操作，Git 不理解AWS訪問密鑰和私有密鑰。相反，您需要向 Git 提供用户名和密碼憑據，以便在與遠程的 HTTPS 連接上使用。

正如設定 Git 登入資料，您要使用的 Git 登入資料必須與 IAM 使用者關聯。您無法為根憑據生成它們。您應該始終設定AWS憑證配置文件，以包含 IAM 用户訪問和私有密鑰，而不是根密鑰。Toolkit for Visual Studio 可以嘗試設定 Git 登入資料AWS CodeCommit，並將它們與AWS之前用於在團隊資源管理器中連接的憑據配置文件。

當您選擇OK (OK)中的建立新的AWS CodeCommit儲存庫對話框並成功創建存儲庫時，Toolkit for Visual Studio 將檢查AWS在團隊資源管理器中連接的憑據配置文件，以確定AWS CodeCommit存在，並且與配置文件本地關聯。如果是這樣，Toolkit for Visual Studio 將指示團隊資源管理器在新存儲庫上開始克隆操作。如果 Git 憑據本地不可用，則 Toolkit for Visual Studio 會檢查團隊資源管理器中連接中使用的帳户憑據的類型。如果證書是針對 IAM 用户的，按照我們的建議，則會顯示以下消息。

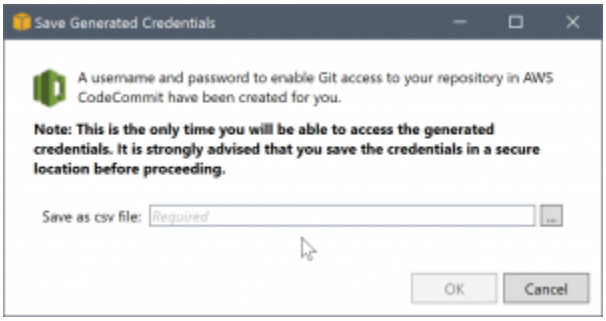


如果憑據是根憑據，則會顯示以下消息。



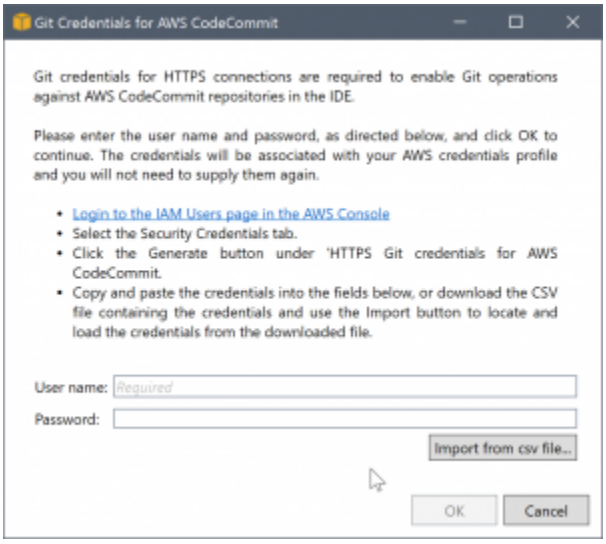
在這兩種情況下，Visual Studio 的工具包都提供了嘗試為您創建必要的 Git 憑據的工作。在第一種情況下，它需要為 IAM 用户創建一組 Git 證書。使用根賬户時，Toolkit for Visual Studio 首先嘗試創建 IAM 用户，然後繼續為該新用户創建 Git 證書。如果 Toolkit for Visual Studio 必須創建一個新用户，則應用AWS CodeCommit超級用户託管策略添加到該新用户帳户。此策略僅允許訪問AWS CodeCommit，並使所有操作都可以使用AWS CodeCommit除了刪除存儲庫。

創建憑據時，您只能查看一次。因此，Toolkit for Visual Studio 會提示您將新創建的憑據另存為`.csv`文件，然後再繼續。



這也是我們強烈建議的，並確保將它們保存到一個安全的位置！

在某些情況下，Toolkit for Visual Studio 無法自動創建登入資料。例如，您可能已經為AWS CodeCommit（二），或者您可能沒有足夠的編程權限，無法讓 Visual Studio 工具包為您執行工作（如果您以 IAM 用户身份登錄）。在這些情況下，您可以登入AWS Management Console管理憑據或從管理員處獲取憑據。您可以將它們輸入到用於的 Git 憑據AWS CodeCommit對話方塊，Visual Studio 的工具包會顯示該對話方塊。

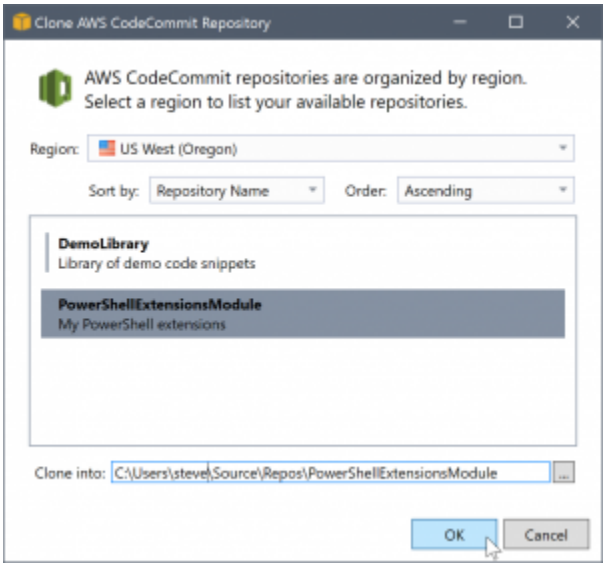


現在 Git 的憑據可用，新存儲庫的克隆操作將繼續進行（請參閱團隊資源管理器中操作的進度指示）。如果您選擇使用默認`.gitignore`文件時，它會提交給存儲庫，並帶有「初始提交」的註釋。

這就是設置憑據和在團隊資源管理器中創建存儲庫的所有內容。一旦所需的憑據到位，您將來在創建新存儲庫時看到的只是建立新的AWS CodeCommit儲存庫對話方塊本身。

複製儲存庫

要克隆現有存儲庫，請返回AWS CodeCommit在團隊資源管理器中。按一下複製鏈接以打開複製AWS CodeCommit儲存庫對話框，然後選擇要克隆的存儲庫以及要將其放置在磁盤上的位置。



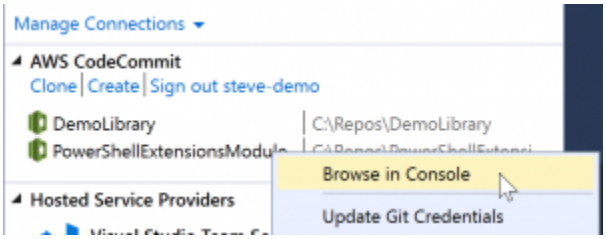
選擇區域後，Toolkit for Visual Studio 將查詢服務以發現該區域中可用的存儲庫，並將它們顯示在對話框的中心列表部分。此外，還會顯示每個存儲庫的名稱和可選描述。您可以對列表重新排序，以便按存儲庫名稱或上次修改日期對其進行排序，並按升序或降序對每個列表進行排序。

選擇存儲庫後，您可以選擇要克隆到的位置。這默認為 Team Explorer 的其他插件中使用的相同存儲庫位置，但您可以瀏覽或輸入任何其他位置。默認情況下，存儲庫名稱後綴到所選路徑上。但是，如果需要特定路徑，只需在選擇文件夾後編輯該文本框即可。當您單擊框中的任何文本OK (OK)將是您將在其中找到克隆存儲庫的文件夾。

選擇存儲庫和文件夾位置後，單擊OK (OK)以繼續執行克隆操作。就像創建存儲庫一樣，您可以看到團隊資源管理器中報告的克隆操作的進度。

使用 儲存庫

克隆或創建存儲庫時，請注意連接的本地存儲庫列在 Team Explorer 的連接面板中的操作鏈接下。通過這些條目，您可以方便地訪問存儲庫以瀏覽內容。只需右鍵單擊存儲庫，然後選擇在控制台中瀏覽。



您也可以使用更新 Git 憑據更新與憑據配置文件關聯的存儲 Git 憑據。如果您已旋轉憑據，則此功能非常有用。該命令將打開用於的 Git 憑據AWS CodeCommit對話框，您可以在其中輸入或導入新憑據。

存儲庫上的 Git 操作按照您的期望工作。您可以進行本地提交，當您準備要共用時，請使用 Team Explorer 中的同步選項。因為 Git 憑據已經存儲在本地，並與我們連接的AWS憑據配置文件，我們將不會被提示再次提供它們來執行AWS CodeCommit遠端。

在視覺工作室中使用 CodeArtifact

AWS CodeArtifact是一種完全受管的工件資料庫服務，可讓組織輕鬆地安全地存儲和共享用於應用程序開發的軟件套件。您可以將 CodeArtifact 與流行的構建工具和軟件包管理器（如 NuGet 和 .NET 核心 CLI 和可視工作室）一起使用。您還可以配置 CodeArtiFit，以便從外部公共存儲庫（如組織。

在 CodeArtifact T 中，您的軟件包存儲在存儲庫中，然後存儲在域中。所以此AWS Toolkit for Visual Studio簡化了 Visual Studio 與您的 CodeArtifact 存儲庫的配置，從而可以輕鬆地從 CodeArtifact 直接和 nuget.org 中使用 Visual Studio 中的軟件包。

將您的 CodeArtifact 存儲庫添加為 NuGet 軟件包源

要使用 CodeArtifact 中的軟件包，您需要將您的存儲庫添加為NuGet 套件管理工具Visual Studio

將您的存儲庫添加為軟件包源

1. InAWS資源管理工具，請導航至AWS CodeArtifact節點。
2. 打開要添加的資料庫的上下文 (按一下右鍵) 選單，然後選擇複製 NuGet 源端點。
3. 導覽至。套件來源下面的NuGet 套件管理工具節點中的工具選單。
4. In套件來源中，選擇加號 (+)，編輯名稱，然後粘貼前面複製的 NuGet 源終端節點 URL來源欄位。
5. 選中新添加的軟件包源旁邊的複選框以啟用它。

> **Note**
>
> 我們建議將外部連接添加到組織添加到 CodeArtifact 並禁用核數字Visual Studio 中的套件。當使用外部連接時，所有從組織存儲在 CodeArtifact 中。如果組織出於任何原因，您需要的軟件包仍然可用。如需外部連線的詳細資訊，請參添加外部連接中的AWS CodeArtifact使用者指南。

6. 選擇確定關閉菜單。

如需使用 Visual Studio CodeArtifact 詳細資訊，請參使用 Visual Studio 的代碼工具中的AWS CodeArtifact使用者指南。

Amazon RDS EditionAWS探險者

Amazon Relational Database Service (Amazon RDS) 是一種服務，您可以在雲端中預配和管理 SQL 關聯式資料庫系統。Amazon RDS 支援三種類型的數據庫系統：

- MySQL Community Edition
- Oracle Enterprise Edition
- 微軟 SQL 服務器（快速版、標準版或 Web 版）

如需詳細資訊，請參閱 Amazon RDS 使用者指南。

這裏討論的很多功能也可以通過AWS管理主控台，以獲得 Amazon RDS。

主題

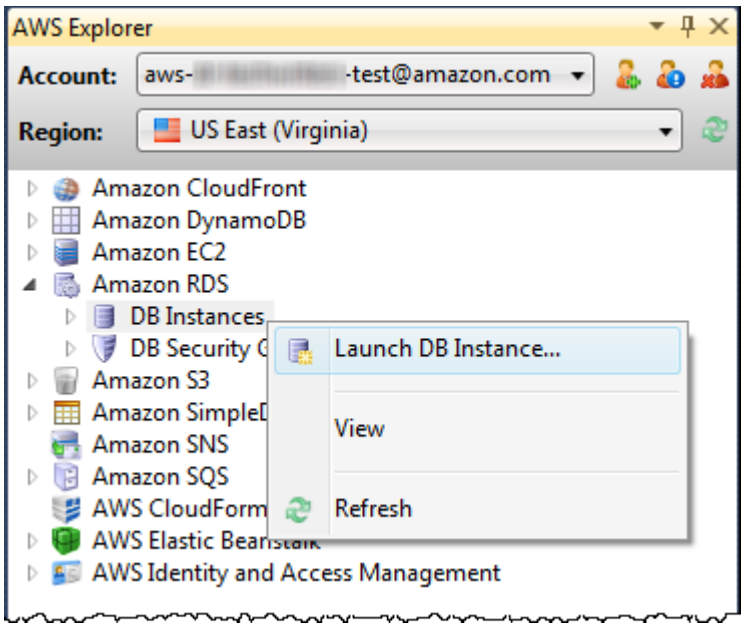
- Launch Amazon RDS 資料庫執行個體
- 在 RDS 實例中建立 Microsoft SQL Server 資料庫
- Amazon RDS 安全羣組

Launch Amazon RDS 資料庫執行個體

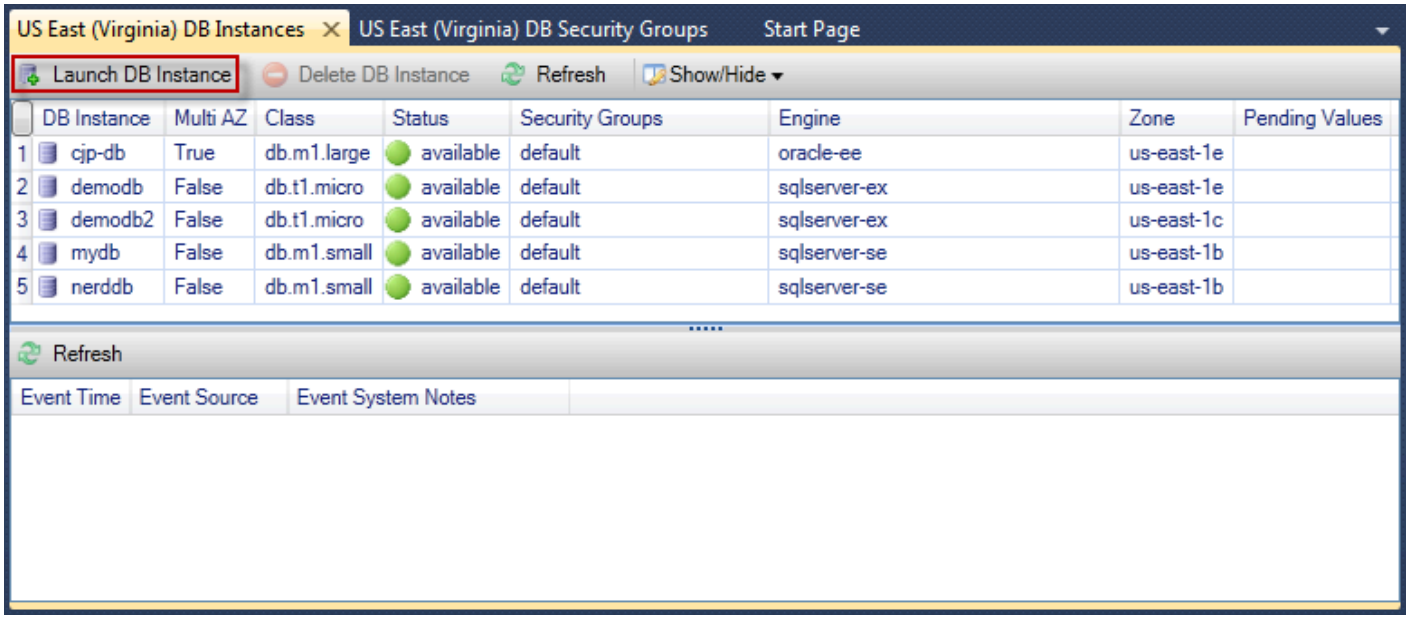
搭配AWS資源管理器，您可以啟動 Amazon RDS 支持的任何數據庫引擎的實例。以下演練顯示了啟動 Microsoft SQL Server 標準版實例的用户體驗，但用户體驗與所有支持的引擎相似。

若要啟動 Amazon RDS 執行個體

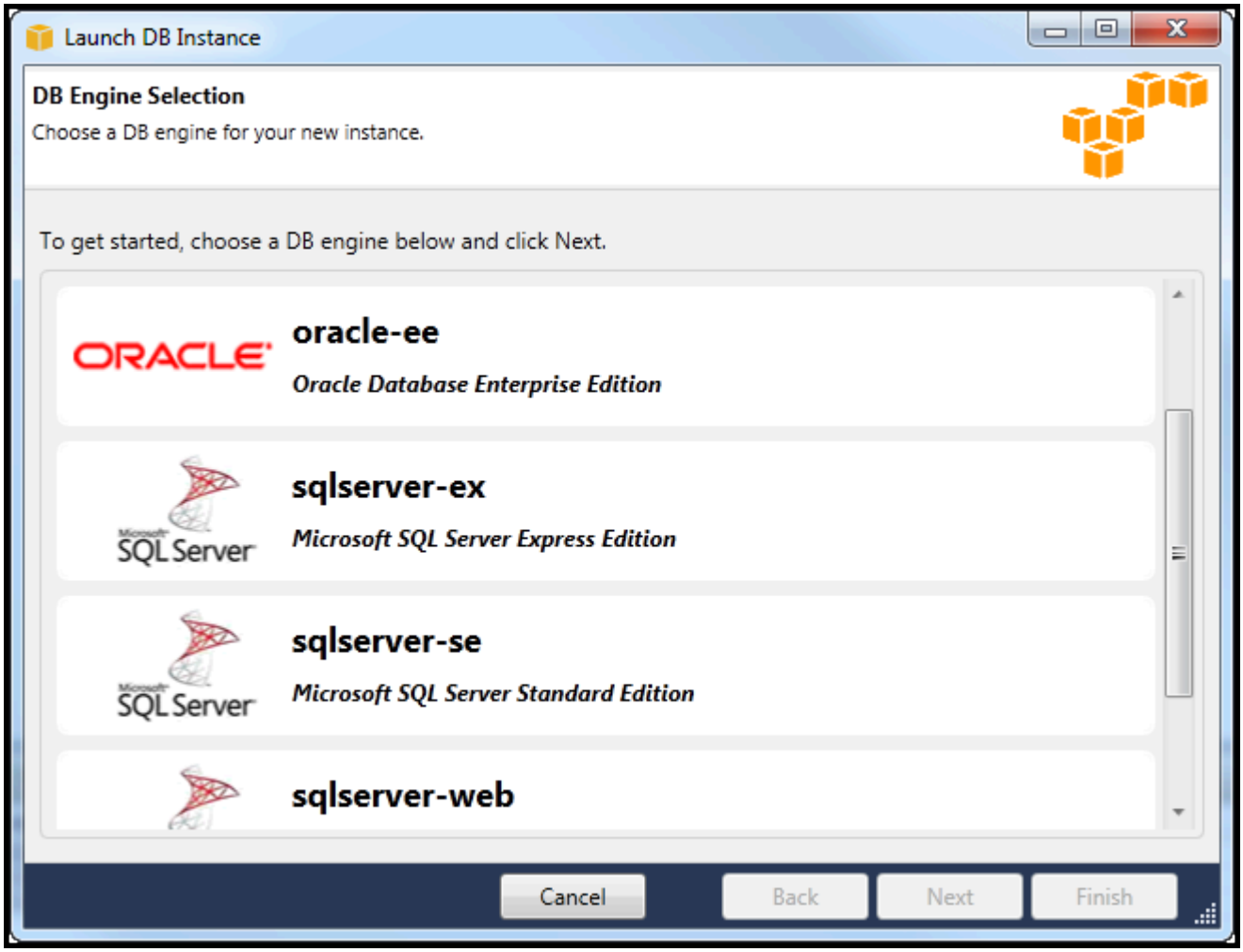
1. InAWSExplorer (按一下滑鼠右鍵) 功能表，然後開啟Amazon RDS節點，然後選擇Launch DB Instance (。



或在資料庫執行個體選項卡上選擇Launch DB Instance (。



2. 在 中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。對於此演練，請選擇 Microsoft SQL Server 標準版 (SQL服務器-se)，然後選擇下一頁。



3. 在 中DB 引擎執行個體選項對話方塊中，選擇配置選項。

   在 中數據庫引擎實例選項和類部分中，您可以指定下列設定。

   License Model (授權模式)

引擎類型	授權
Microsoft SQL Server	已包含授權
MySql	一般公共許可
Oracle	選擇自有授權

根據資料庫引擎的類型，授權模型會有所不同。Microsoft SQL Server 授權包含 MySql 一般公共授權合約

數據庫實例版本

選擇您要使用的資料庫引擎版本。如果只支持一個版本，則會為您選取它。

資料庫執行個體類別

選擇資料庫引擎的執行個體類。實例類的定價有所不同。如需詳細資訊，請參閱「」Amazon RDS 定價。

執行多可用區部署

選擇此選項可創建多可用區部署，以增強數據持久性和可用性。Amazon RDS 在不同的可用區中配置和維護數據庫的備用副本，以便在出現計劃或計劃外停機時自動故障切換。有關多可用區部署定價的信息，請參閱Amazon RDS詳細資訊頁面。SQL 服務器不支持此選項。

自動升級次要版本

選取此選項可使AWS自動為您的 RDS 實例執行次要版本更新。

在 中RDS 資料庫執行個體部分中，您可以指定下列設定。

Allocated Storage (配置的儲存體)

引擎	最小值 (GB)	最大值 (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

分配存儲的最小值和最大值取決於數據庫引擎的類型。引擎最小值 (GB) 最大值 (GB) MySQL 5 1024 甲骨文企業版 10 1024 微軟 SQL 服務器快速版 30 1024 微軟 SQL 服務器標準版 250 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。它將以小寫形式顯示在AWSExplorer。

Master User Name (主要使用者名稱)

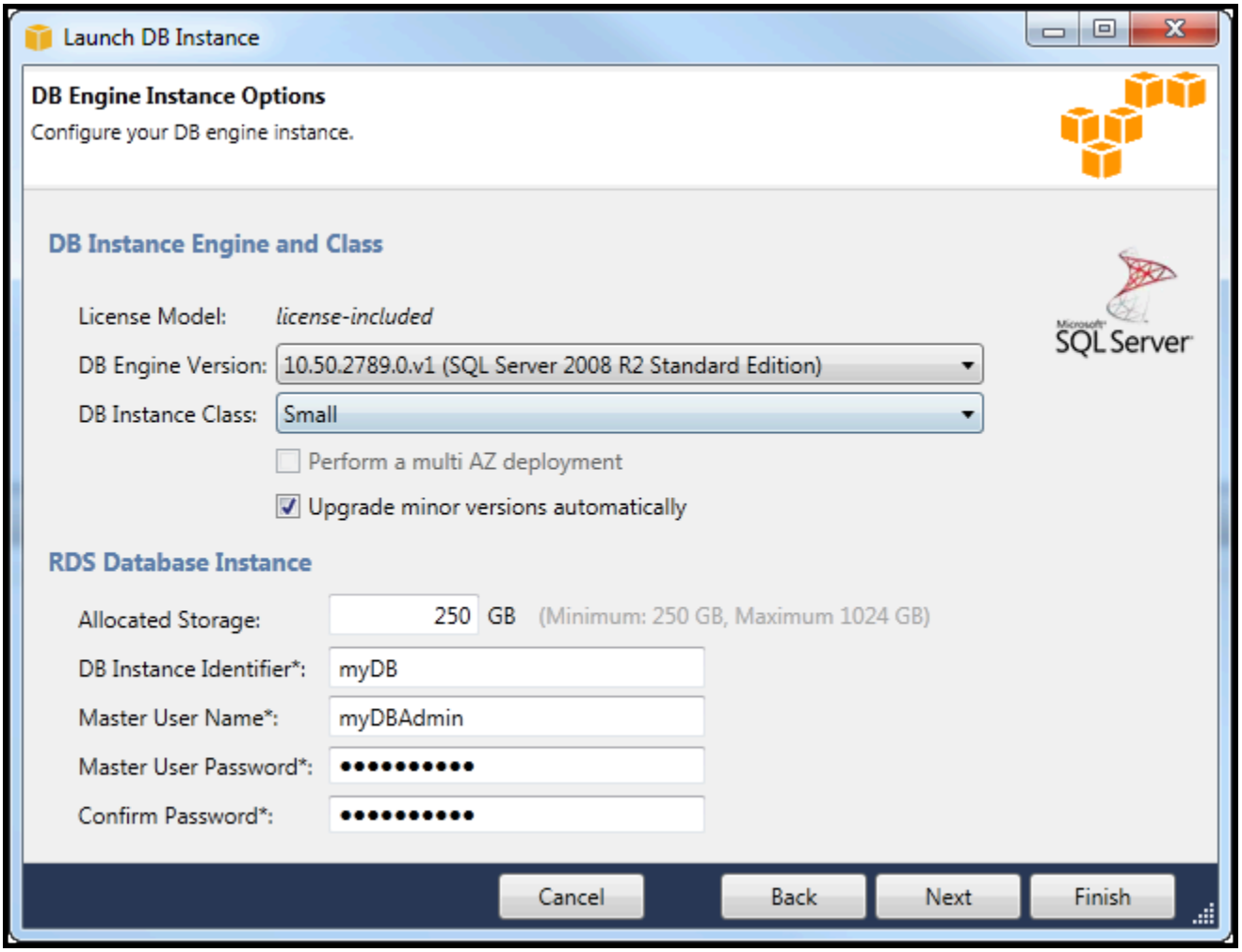
鍵入資料庫執行個體的管理員的名稱。

Master User Password (主要使用者密碼)

鍵入資料庫執行個體的管理員的密碼。

確認密碼

再次鍵入密碼以驗證密碼是否正確。



1. 在 中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)

   這是實例用於在網絡上通信的 TCP 端口。如果您的計算機通過防火牆訪問 Internet，請將此值設置為防火牆允許通信的端口。

   可用區域

   如果您希望在您所在地區的特定可用區啟動實例，請使用此選項。在特定區域中可能無法在所有可用區域中使用您指定的資料庫執行個體。

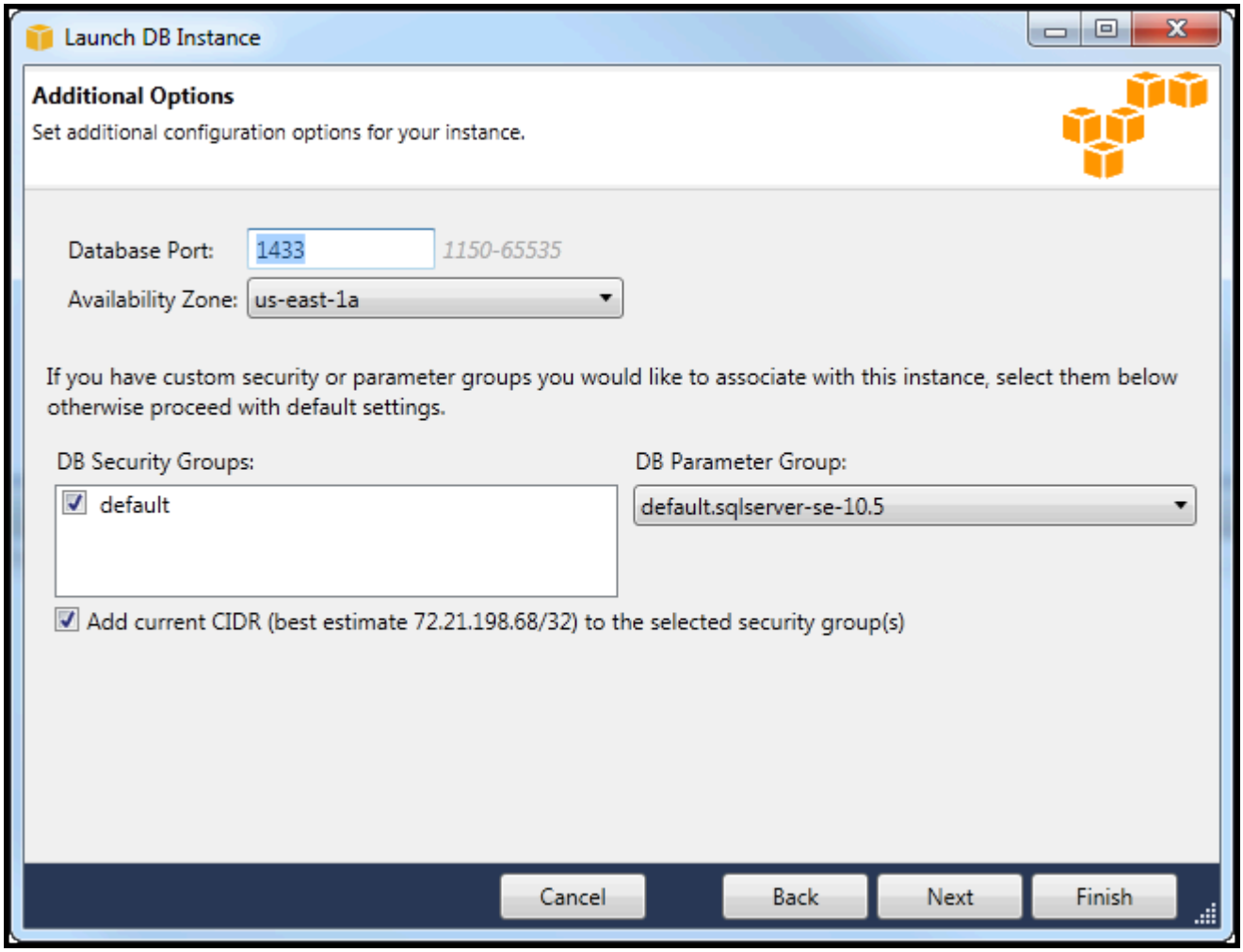
   RDS 安全羣組

   選擇 RDS 安全羣組（或組）來與您的執行個體建立關聯。RDS 安全組指定 IP 地址、Amazon EC2 實例和AWS 帳戶允許訪問您的實例。如需 RDS 安全羣組的詳細資訊，請參閱Amazon RDS 安全羣組。Toolkit for Visual Studio 將嘗試確定您當前的 IP 地址，並提供將此地址添加到與您的實例關聯的安全組的選項。但是，如果您的計算機通過防火牆訪問 Internet，則 Toolkit 為您的計算機生成的 IP 地址可能不準確。要確定要使用的 IP 地址，請諮詢系統管理員。

   資料庫參數群組

   （可選）從此下拉列表中，選擇要與您的實例關聯的數據庫參數組。使用數據庫參數組，您可以更改實例的默認配置。如需詳細資訊，請前往Amazon Relational Database Service 用户指南和這篇文章。

   在此對話框中指定了設置後，請選擇下一頁。

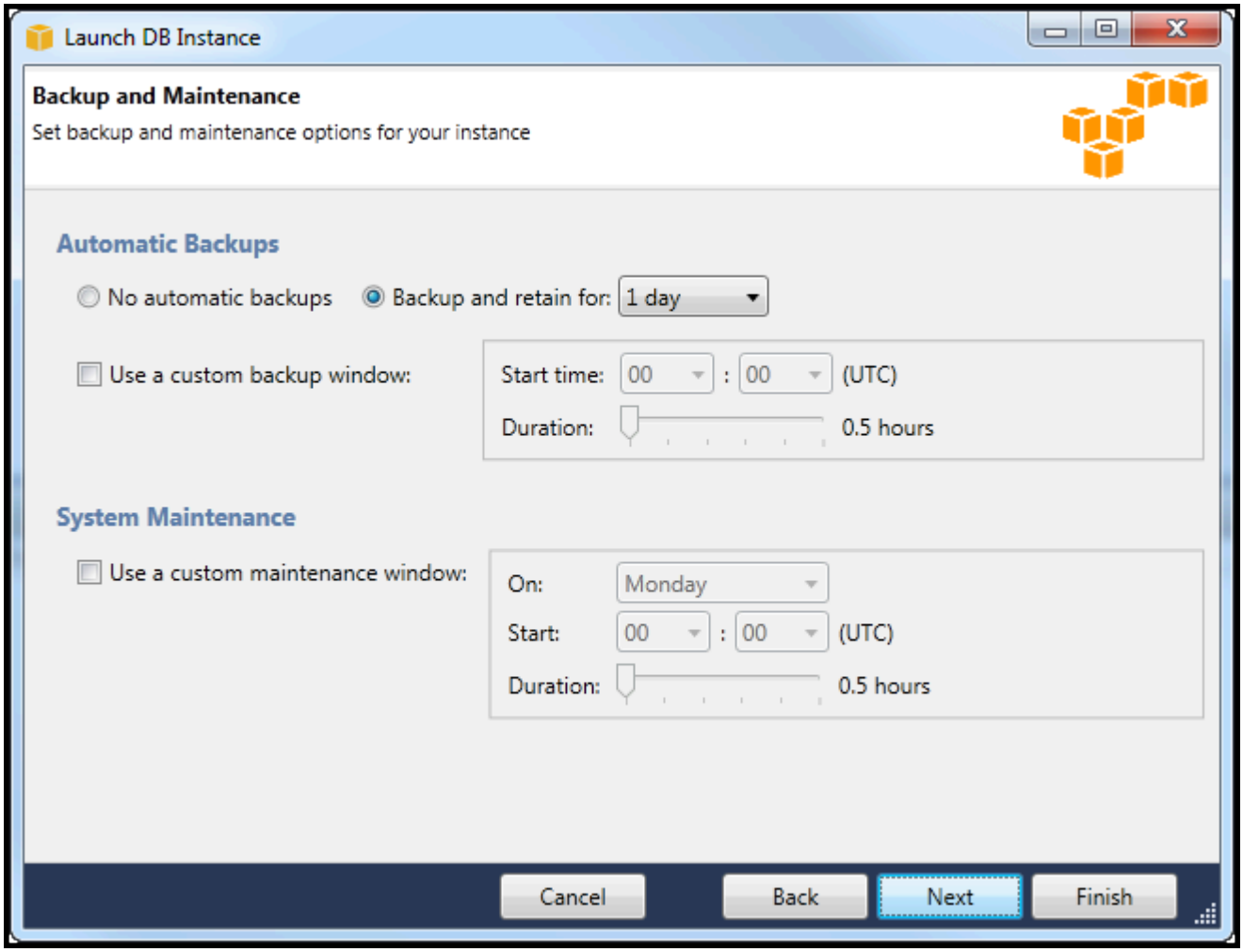


2. 所以此Backup 和維護對話框中，您可以指定 Amazon RDS 是否應備份您的實例，如果應該備份，備份應保留多長時間。您還可以指定執行備份的時段。

   此對話框還允許您指定是否希望 Amazon RDS 對您的實例執行系統維護。維護包括常規修補程序和次要版本升級。

   為系統維護指定的時間窗口不能與為備份指定的時間窗口重疊。

   選擇 Next (下一步)。



3. 嚮導中的最後一個對話框允許您查看實例的設置。如果您需要修改設定，請使用Back按鈕。如果所有設定都正確，請選擇啟動。

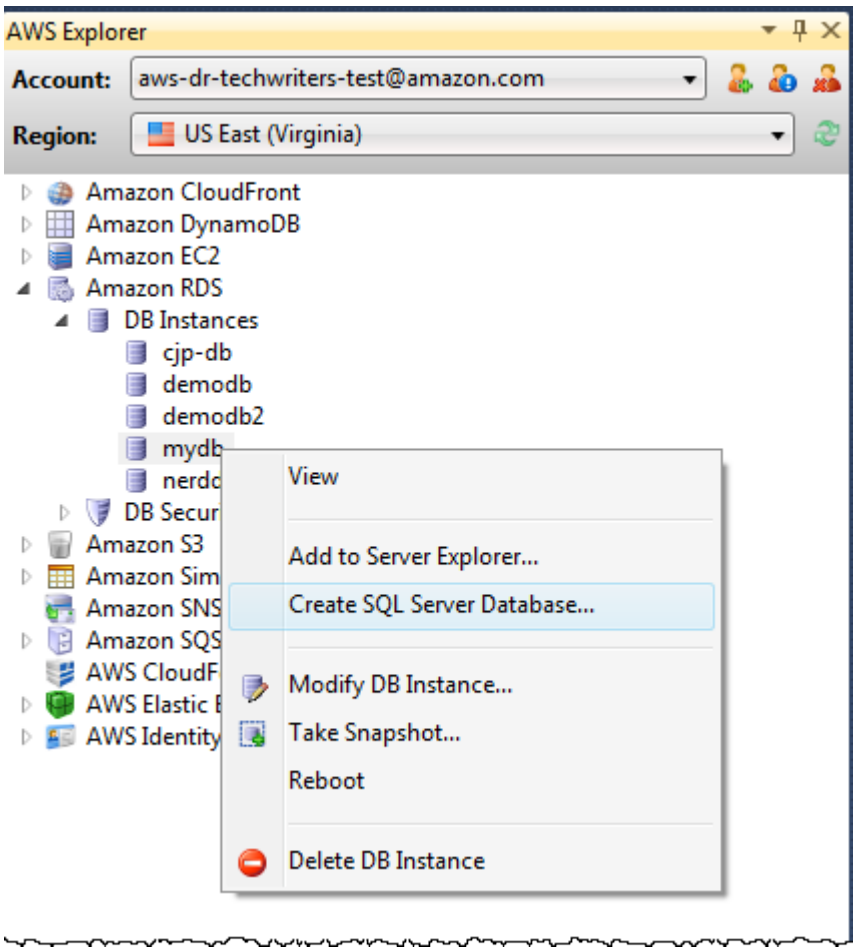
在 RDS 實例中建立 Microsoft SQL Server 資料庫

微軟 SQL 服務器的設計方式是，在啟動亞馬遜 RDS 實例後，您需要在 RDS 實例中創建 SQL Server 數據庫。

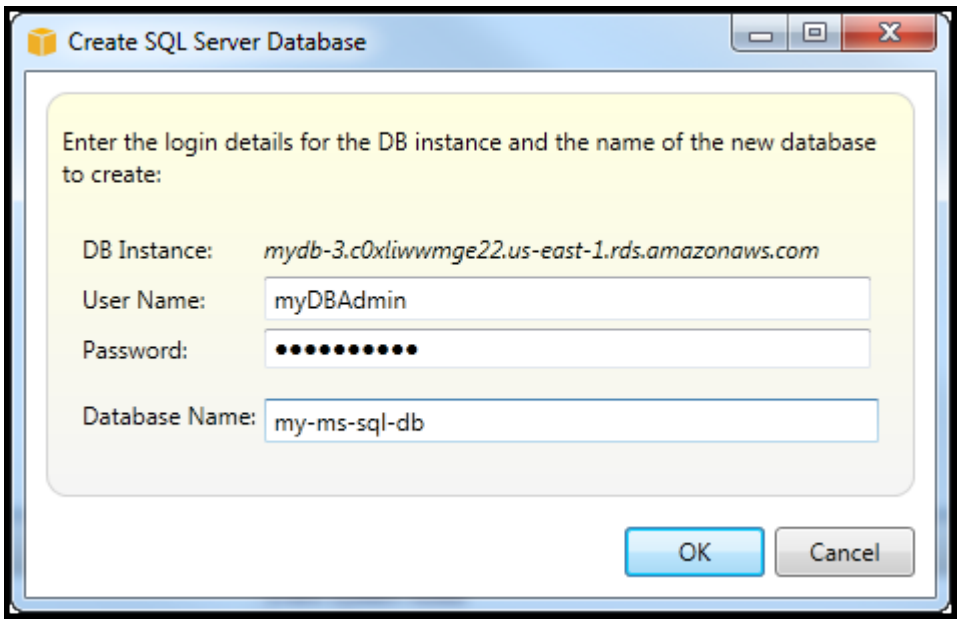
如需如何建立 Amazon RDS 實例的詳細資訊，請參啟動 Amazon RDS 資料庫實例。

若要建立 Microsoft SQL Server 資料庫

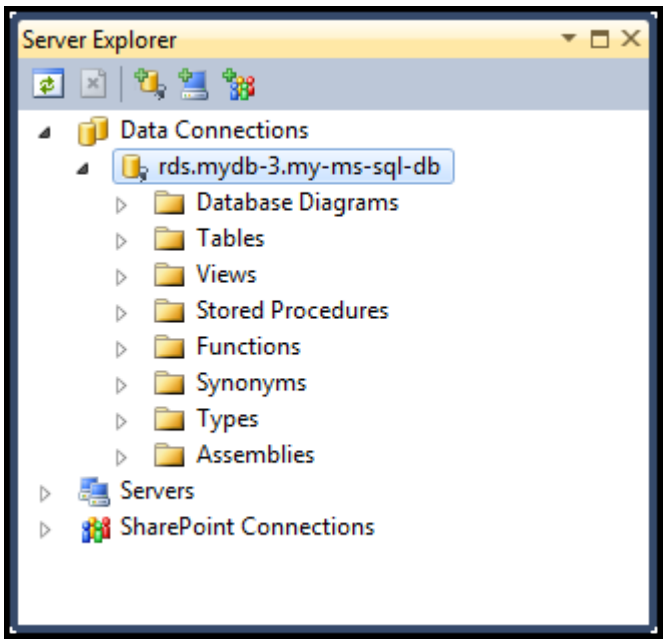
1. InAWS資料庫中，開啟 RDS 實例的內容功能表 (按一下滑鼠右鍵)，然後選擇建立 SQL Server 資料庫。



2. 在 中建立 SQL Server 資料庫對話框中，鍵入您在創建 RDS 實例時指定的密碼，鍵入 Microsoft SQL Server 數據庫的名稱，然後選擇確定。



3. 可 Toolkit for Visual Studio 創建 SQL Server 數據庫並將其添加到可視工作室服務器資源管理器。



Amazon RDS 安全羣組

Amazon RDS 安全組使您能夠管理對 Amazon RDS 實例的網絡訪問。對於安全組，您可以使用 CIDR 表示法指定 IP 地址集，您的 Amazon RDS 實例只能識別來自這些地址的網絡流量。

儘管 Amazon RDS 安全組的運行方式類似，但它們與 Amazon EC2 安全組不同。您可以將 EC2 安全羣組添加到 RDS 安全羣組。所有作為 EC2 安全組成員的 EC2 實例都可以訪問 RDS 安全組成員的 RDS 實例。

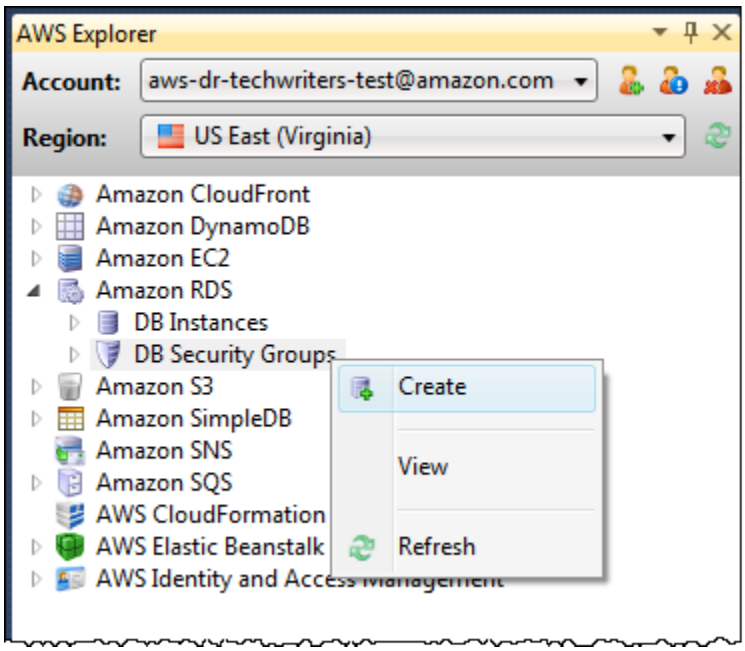
如需 Amazon RDS 安全羣組的詳細資訊，請前往RDS 安全羣組。如需 Amazon EC2 安全羣組的詳細資訊，請前往EC2 用户指南。

建立 Amazon RDS 安全羣組

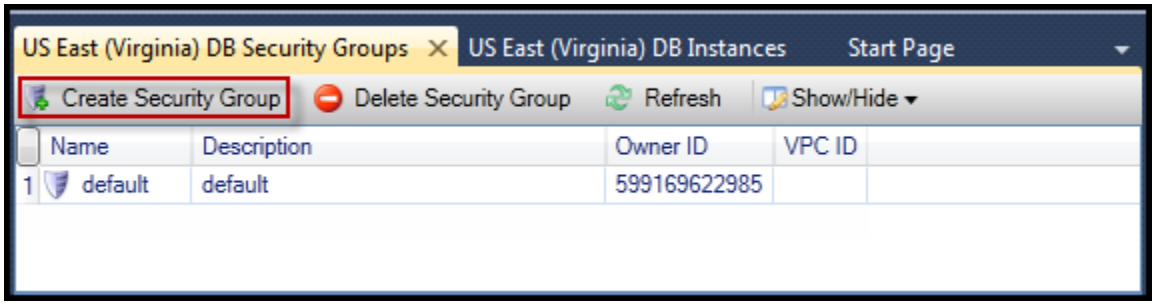
您可使用 Toolkit for Visual Studio 來建立 RDS 安全羣組。如果您使用AWSToolkit 啟動 RDS 實例時，嚮導將允許您指定一個 RDS 安全組以用於您的實例。您可使用下列程序來建立該安全羣組，然後再啟動嚮導。

若要建立 Amazon RDS 安全羣組

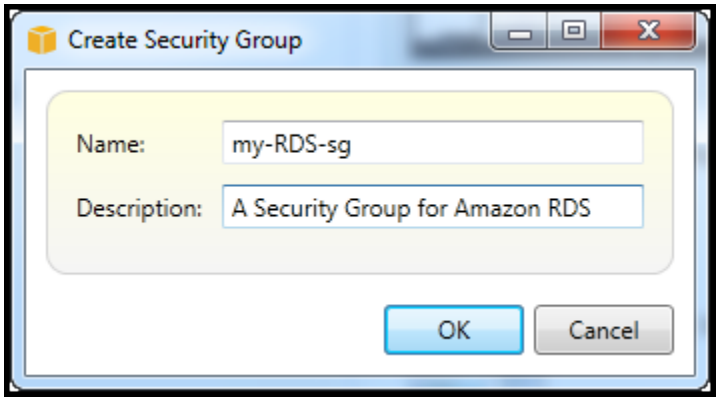
1. InAWS資源管理器中，展開Amazon RDS節點，開啟內容 (按一下滑鼠右鍵) 功能表，然後選擇資料庫安全群組子節點，然後選擇建立。



或者，在安全群組標籤上選擇建立安全羣組。如果未顯示此標籤，請開啟內容 (按一下滑鼠右鍵) 功能表資料庫安全群組子節點，然後選擇檢視。



2. 在 中建立安全羣組對話框中，鍵入安全羣組的名稱和描述，然後選擇確定。



設定 Amazon RDS 安全羣組的訪問許可

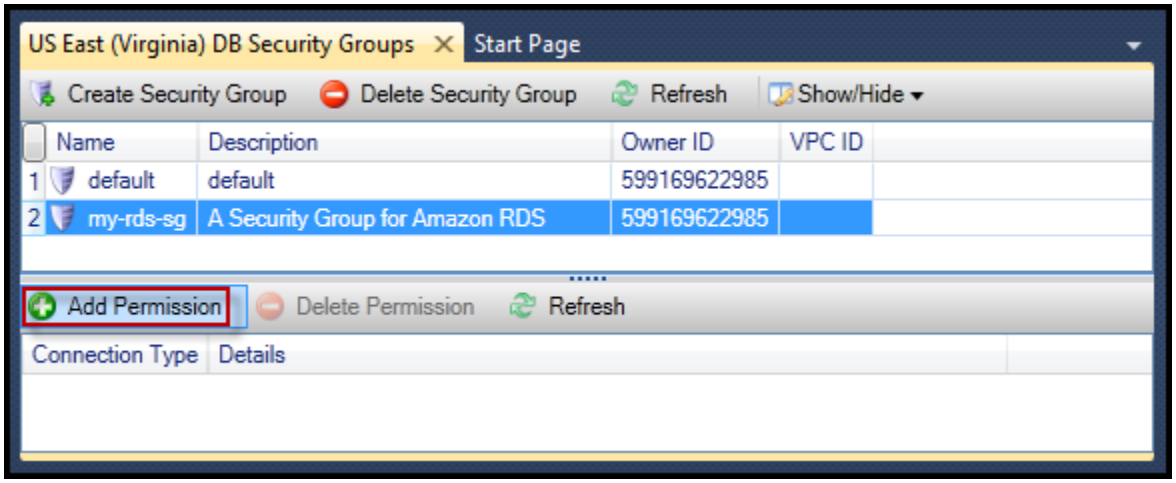
默認情況下，新的 Amazon RDS 安全組不提供網絡訪問權限。要啟用對使用安全組的 Amazon RDS 實例的訪問權限，請使用以下過程設置其訪問權限。

設置 Amazon RDS 安全組的訪問權限

1. 在安全群組選項卡上，從清單視圖中選擇安全羣組。如果您的安全羣組未出現在清單中，請選擇重新整理。如果您的安全組仍未顯示在列表中，請驗證您正在查看正確的AWS區域。安全群組標籤AWS工具包專屬於特定區域。

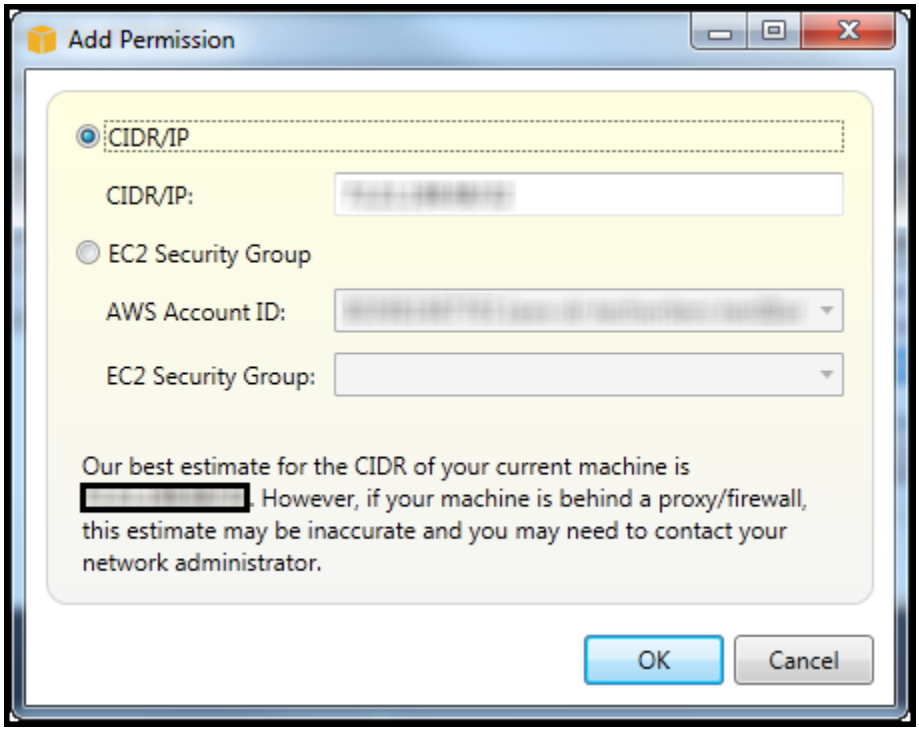
   如果沒有安全群組選項卡出現，位於AWS瀏覽器，開啟內容 (按一下右鍵) 功能表，然後選擇資料庫安全群組子節點，然後選擇檢視。

2. 選擇 Add Permission (新增許可)。



新增許可按鈕安全群組標籤

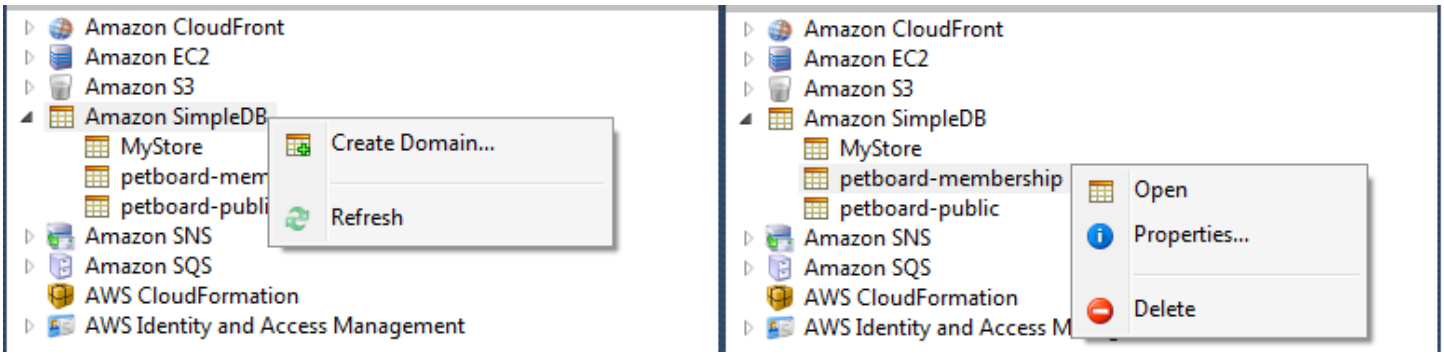
3. 在 中新增許可對話框中，您可以使用 CIDR 表示法指定哪些 IP 地址可以訪問 RDS 實例，也可以指定哪些 EC2 安全組可以訪問 RDS 實例。當您選擇EC2 安全羣組，您可以為與AWS 帳戶或者您可以從下拉式清單中選擇 EC2 安全羣組。



所以此AWS工具包會嘗試確定您的 IP 地址並使用相應的 CIDR 規範自動填充對話框。但是，如果您的計算機通過防火牆訪問互聯網，工具包確定的 CIDR 可能不準確。

使用 Amazon SimpleDBAWS探險者

AWS資源管理器顯示與活動AWS帳户。從AWS資源管理器，您可以創建或刪除 Amazon SimpleDB 域。

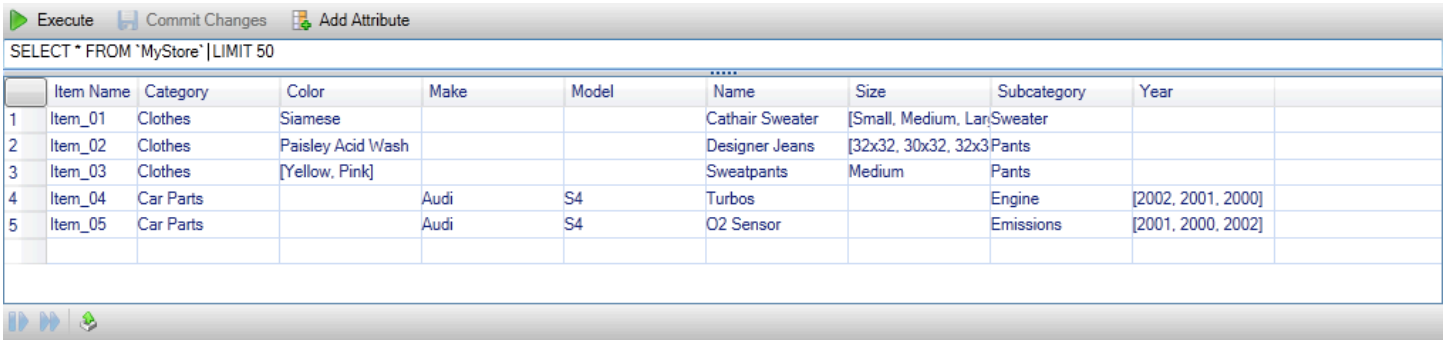


Create, delete, or open Amazon SimpleDB domains associated with your account

執行查詢和編輯結果

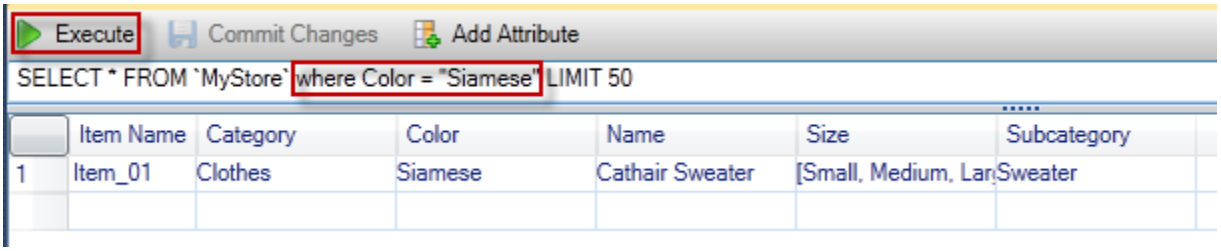
AWSExplorer 也可以顯示 Amazon SimpleDB 域的網格視，您可以從中查看該域中的項目、屬性和值。您可以執行查詢，以便只顯示域項目的子集。通過雙擊某個單元格，您便可以編輯項目的對應屬性值。您也可以新增屬性到域。

此處顯示的域名來自 Amazon SimpleDB 示例AWS SDK for .NET。



Amazon SimpleDB grid view

要執行查詢，請在網格視圖頂部的文本框中編輯查詢，然後選擇Execute。將篩選視圖以僅顯示與查詢匹配的項目。

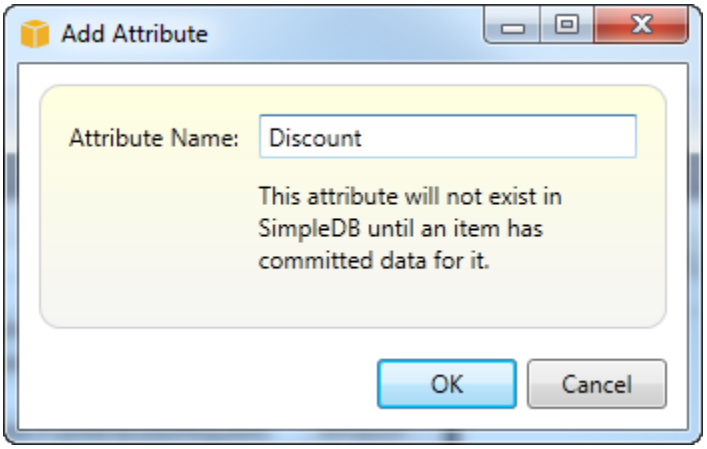


Execute query from AWS Explorer

要編輯與屬性關聯的值，請雙擊相應的單元格，編輯值，然後選擇提交更改。

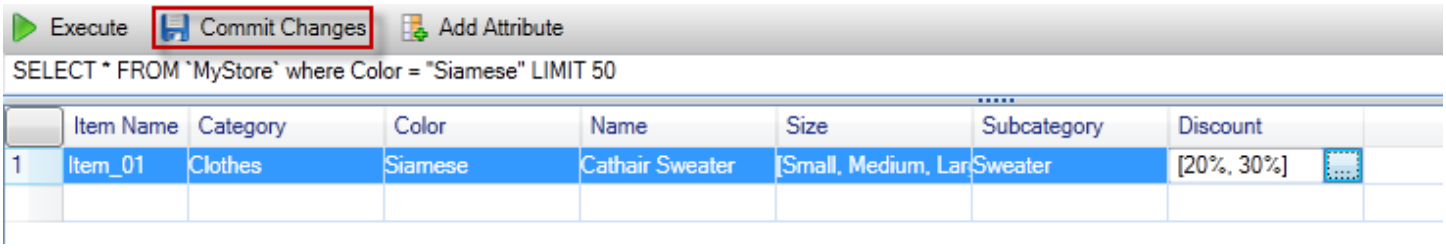
新增屬性

若要新增屬性，請在視圖頂端，選擇新增屬性。



新增屬性 dialog box

要使屬性成為域的一部分，您必須向至少一個項目添加一個值，然後選擇提交更改。



Commit changes for a new attribute

分頁查詢結果

視圖底部有三個按鈕。

Paginate and export buttons

前兩個按鈕提供查詢結果的分頁。要顯示其他頁面的結果，請選擇第一個按鈕。要再顯示十頁的結果，請選擇第二個按鈕。在此上下文中，頁面等於 100 行或 LIMT 值指定的結果數（如果查詢中包含）。

匯出至 CSV

最後一個按鈕將當前結果匯出到 CSV 檔案。

使用 Amazon SQSAWS探險者

Amazon Simple Queue Service (Amazon SQS) 是一種靈活的佇列服務，可在軟件應用程式中的不同執行流程之間傳送訊息。Amazon SQS 隊列位於AWS基礎設施，但傳遞消息的進程可以位於本地、Amazon EC2 實例上，也可以位於這些實例的某些組合上。Amazon SQS 是協調跨多台計算機分配工作的理想選擇。

Toolkit for Visual Studio 使您能夠查看與活動帳户關聯的 Amazon SQS 隊列、創建和刪除隊列以及通過隊列發送消息。（通過活躍賬户，我們指AWSExplorer。

如需有關 Amazon SQS 的詳細資訊，請前往SQS 簡介中的AWS文件中)。

建立佇列

您可以從AWSExplorer 隊列的 ARN 和 URL 將基於活動帳户的帳户號碼和您在創建時指定的隊列名稱。

建立佇列

1. InAWSExplorer，開啟內容 (按右鍵) 選單，然後選擇Amazon SQS節點，然後選擇建立佇列。
2. 在 中建立佇列對話框中，指定隊列名稱、默認可見性超時和默認傳遞延遲。默認的可見性超時和默認傳遞延遲以秒為單位指定。默認的可見性超時是指在給定進程獲取消息後對潛在接收進程不可見的消息的時間長度。默認傳遞延遲是從發送消息的那一刻起到其首次對潛在接收進程可見的時間長度。
3. 選擇 OK (確定)。新隊列將作為一個子節點顯示在Amazon SQS節點。

刪除佇列

您可以將現有隊列從AWSExplorer 如果刪除隊列，則與該隊列關聯的所有消息將不再可用。

刪除佇列

1. InAWSExplorer，開啟您要刪除的佇列的內容 (按右鍵) 選單，然後選擇刪除。

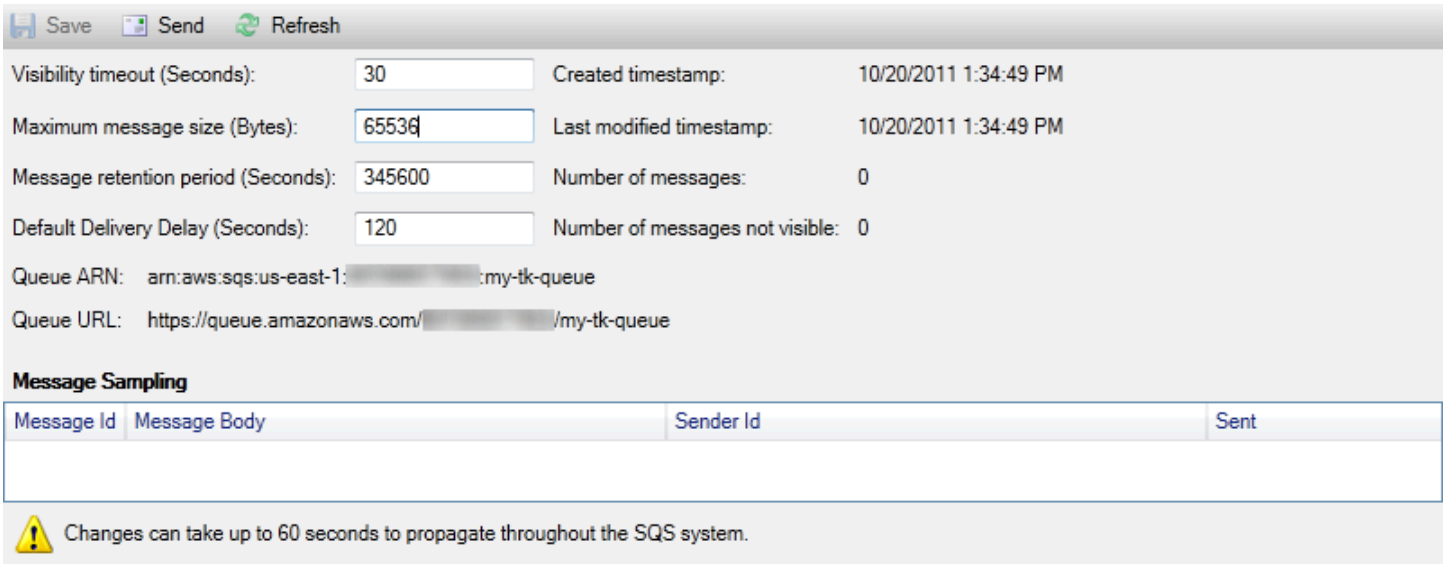
管理佇列屬性

您可以查看和編輯AWSExplorer 您還可以從此屬性視圖向佇列發送訊息。

管理隊列屬性

- InAWSExplorer，開啟您要管理其屬性的佇列的內容 (按右鍵) 選單，然後選擇檢視佇列。

  從隊列屬性視圖中，您可以編輯可見性超時、最大郵件大小、郵件保留期和默認傳遞延遲。傳送訊息時，可以覆寫默認傳送延遲。在下面的屏幕截圖中，模糊的文本是隊列 ARN 和 URL 的帳户號組件。



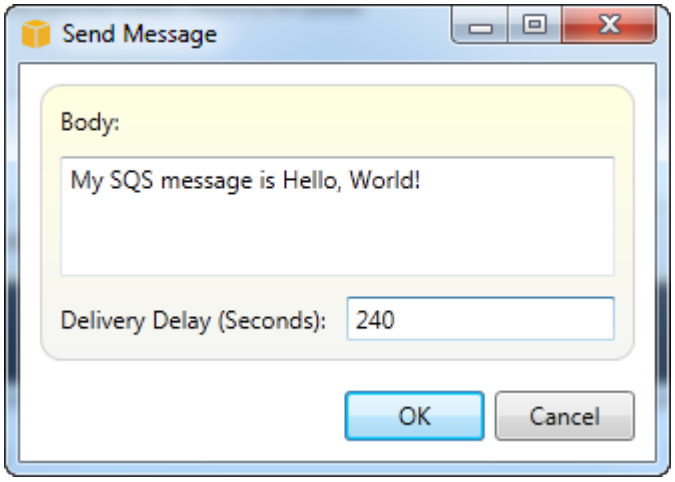
SQS queue properties view

傳送訊息至佇列

從隊列屬性視圖中，您可以向隊列發送消息。

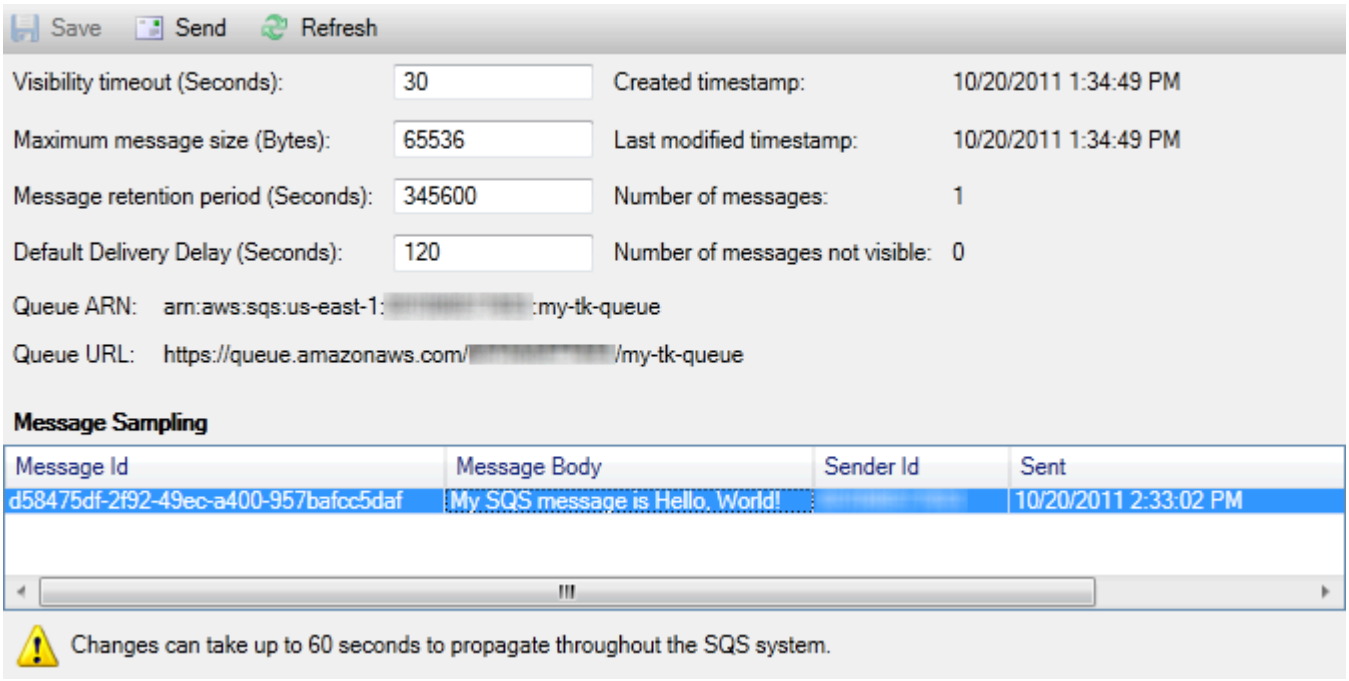
傳送訊息

1. 在佇列屬性視圖頂端，選擇傳送按鈕。
2. 鍵入消息。（可選）輸入配送延遲，該延遲將覆蓋隊列的默認傳遞延遲。在下面的示例中，我們用 240 秒的值覆蓋了延遲。選擇 OK (確定)。



傳送訊息 dialog box

3. 等待大約 240 秒 (四分鐘)。訊息將顯示在訊息抽樣部分中的隊列屬性視圖。



SQS properties view with sent message

隊列屬性視圖中的時間戳是您選擇傳送按鈕。它不包括延遲。因此，消息出現在隊列中並可供接收方使用的時間可能晚於此時間戳。時間戳以計算機的本地時間顯示。

Identity and Access Management

AWS Identity and Access Management(IAM)，您可以更安全地管理您的AWS 帳戶和資源. 藉助 IAM，您可以在主（根)AWS 帳戶。這些用户可以擁有自己的證書：密碼、訪問密鑰 ID 和私有密鑰，但所有 IAM 用户共享一個賬户號碼。

您可以通過向用户附加 IAM 策略來管理每個 IAM 用户的資源訪問級別。例如，您可以將策略附加到 IAM 用户，該策略授予用户訪問 Amazon S3 服務和您賬户中相關資源的權限，但該策略不提供對任何其他服務或資源的訪問權限。

如需更高效的存取管理，您可以建立 IAM 羣組，即使用者的集合。當您將策略附加到組時，它將影響作為該組成員的所有用户。

除了管理用户和組級別的權限外，IAM 還支持 IAM 角色的概念。與用户和組一樣，您可以將策略附加到 IAM 角色。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，可以訪問AWS使用 IAM 角色所提供的許可。如需將 IAM 角色與 Toolkit 結合使用的詳細資訊，請參建立 IAM 角色。如需 IAM 的詳細資訊，請前往IAM User Guide。

建立和配置 IAM 使用者

IAM 用户允許您授予他人訪問您的AWS 帳戶。由於您能夠將策略附加到 IAM 用户，因此您可以精確限制 IAM 用户可以訪問的資源以及他們可以對這些資源執行的操作。

根據最佳實務，所有使用者AWS 帳戶應以 IAM 用户身份執行此操作，甚至是賬户的所有者。這可確保如果其中一個 IAM 使用者的登入資料遭到盜用，只能停用這些登入資料。無需停用或更改帳户的根憑據。

您可以從 Toolkit to Visual Studio，您可以通過將 IAM 政策連接到 IAM 使用者或將該使用者指派到羣組中，將許可指派給 IAM 使用者。被指派到該羣組的 IAM 使用者將獲取連接到該羣組的政策許可。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。

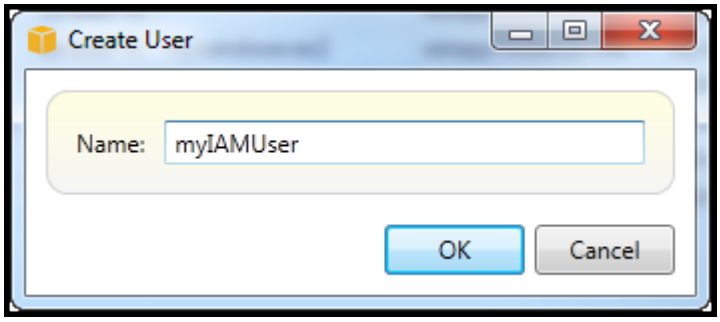
您還可以從 Toolkit for Visual Studio 中，您也可以生成AWS登入資料（存取密鑰 ID 和私密鑰）。如需詳細資訊，請參閱「」生成 IAM 使用者的登入資料

適用 Toolkit for Visual Studio 支持指定 IAM 使用者登入資料，以便通過AWSExplorer 由於 IAM 用户通常沒有對所有 Amazon Web Services 的完全訪問權限，因此AWSExplorer 可能不適用。如果您使用AWS資源管理器在活動賬户是 IAM 用户時更改資源，然後將活動賬户切換到根賬户，則更改可能不會顯示，直到您刷新AWSExplorer 若要刷新視圖，請選擇 Reresh () 按鈕。

如需如何從AWS Management Console，請前往使用使用者和用户(IAM 使用者指南)。

建立 IAM 使用者

1. InAWS資源管理器中，展開AWS Identity and Access Management節點中，開啟內容 (按右鍵) 功能表，然後選擇使用者然後選擇建立使用者。
2. 在 中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友好名稱。如需 IAM 使用者名稱限制條件的詳細資訊，請參IAM User Guide。



Create an IAM user

新使用者將顯示為使用者在下方AWS Identity and Access Management節點。

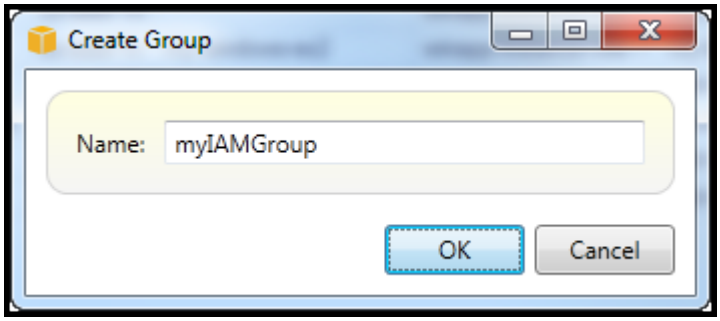
如需如何建立政策，並將它連接到使用者的資訊，請參建立 IAM 政策。

建立 IAM 群組

組提供了一種將 IAM 策略應用於用户集合的方法。如需如何管理 IAM 使用者和羣組的詳細資訊，請參使用使用者和用户(IAM 使用者指南)。

若要建立 IAM 群組

1. InAWSExplorerIdentity and Access Management下，開啟內容 (按右鍵) 功能表，然後選擇Groups (群組)並選擇建立羣組。
2. 在 中建立羣組對話方塊中，輸入 IAM 羣組的名稱，然後選擇確定。



Create IAM group

新 IAM 組將顯示在Groups (群組)子節點Identity and Access Management。

如需如何建立政策，並將它連接到 IAM 羣組的詳細資訊，請參建立 IAM 政策。

將 IAM 使用者新增至 IAM 群組

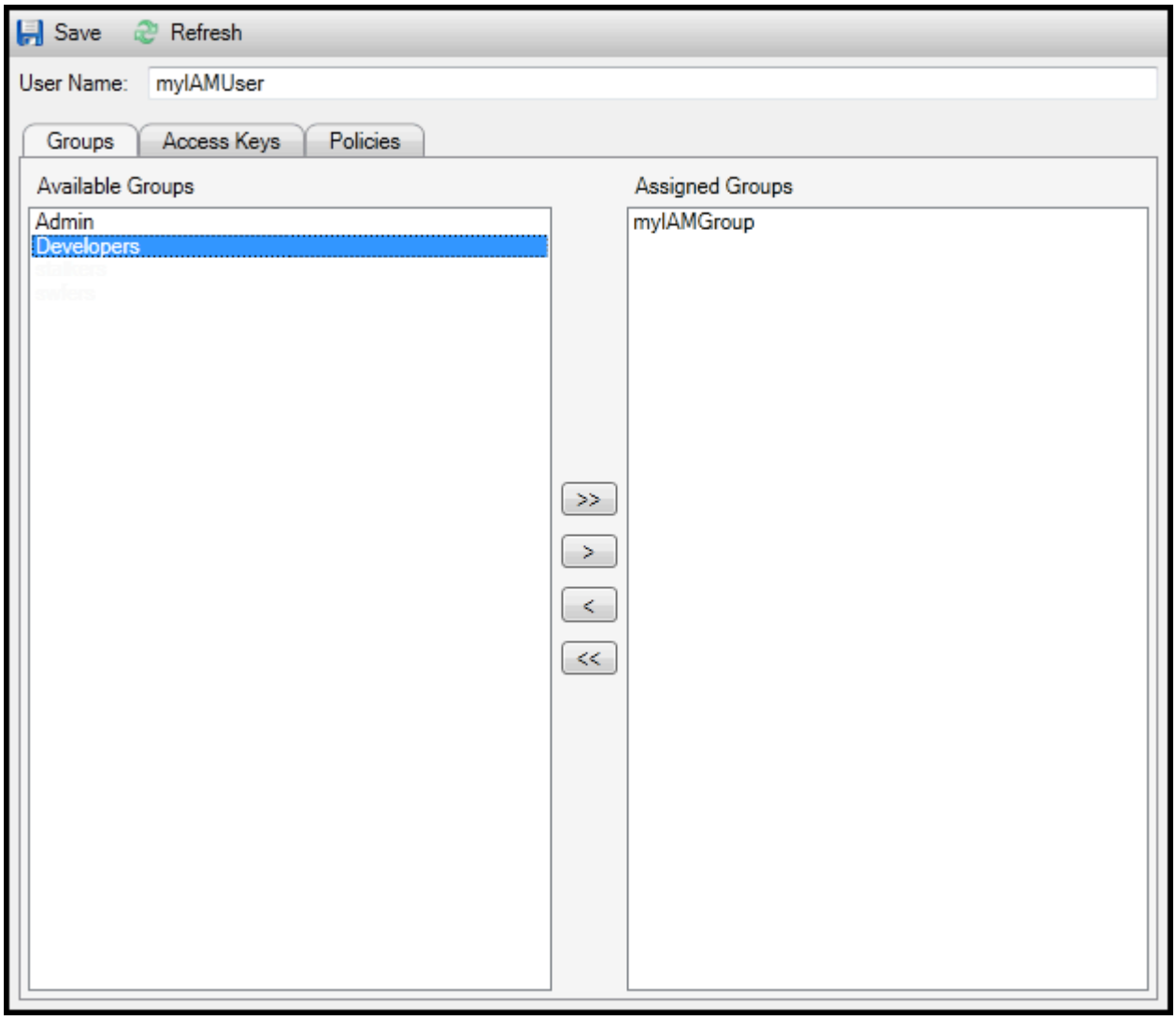
IAM 羣組成員的 IAM 使用者將從連接到該羣組的政策中獲取存取許可。IAM 羣組的目的是使您能夠更輕鬆地管理 IAM 使用者集合的許可。

有關附加到 IAM 組的策略如何與附加到作為該 IAM 組成員的 IAM 用户的策略進行交互的信息，請轉到《IAM 使用者指南》中的管理 IAM 政策。

InAWS資源管理器，您可以從使用者子節點，而不是Groups (群組)子節點。

若要將 IAM 使用者新增至 IAM 群組

1. InAWSExplorerIdentity and Access Management下，開啟內容 (按右鍵) 功能表，然後選擇使用者並選擇Edit (編輯)。



Assign an IAM user to a IAM group

2. 的左側窗格顯示Groups (群組)選項卡顯示可用的 IAM 組。右窗格顯示指定 IAM 用户已經是其成員的組。

   若要將 IAM 使用者添加到羣組中，請在左窗中選擇 IAM 羣組，然後選擇>按鈕。

   要從羣組中移除 IAM 用户，請在右窗格中選擇 IAM 組，然後選擇<按鈕。

   要將 IAM 用户添加到所有 IAM 組，請選擇>>按鈕。同樣，要從所有組中移除 IAM 用户，請選擇<<按鈕。

   要選擇多個組，請按順序選擇它們。您不需要按住 Control (Control) 鍵。要從您的選擇中清除某個組，只需再次選擇該組。

3. 完成將 IAM 用户分配給 IAM 組後，選擇Save。

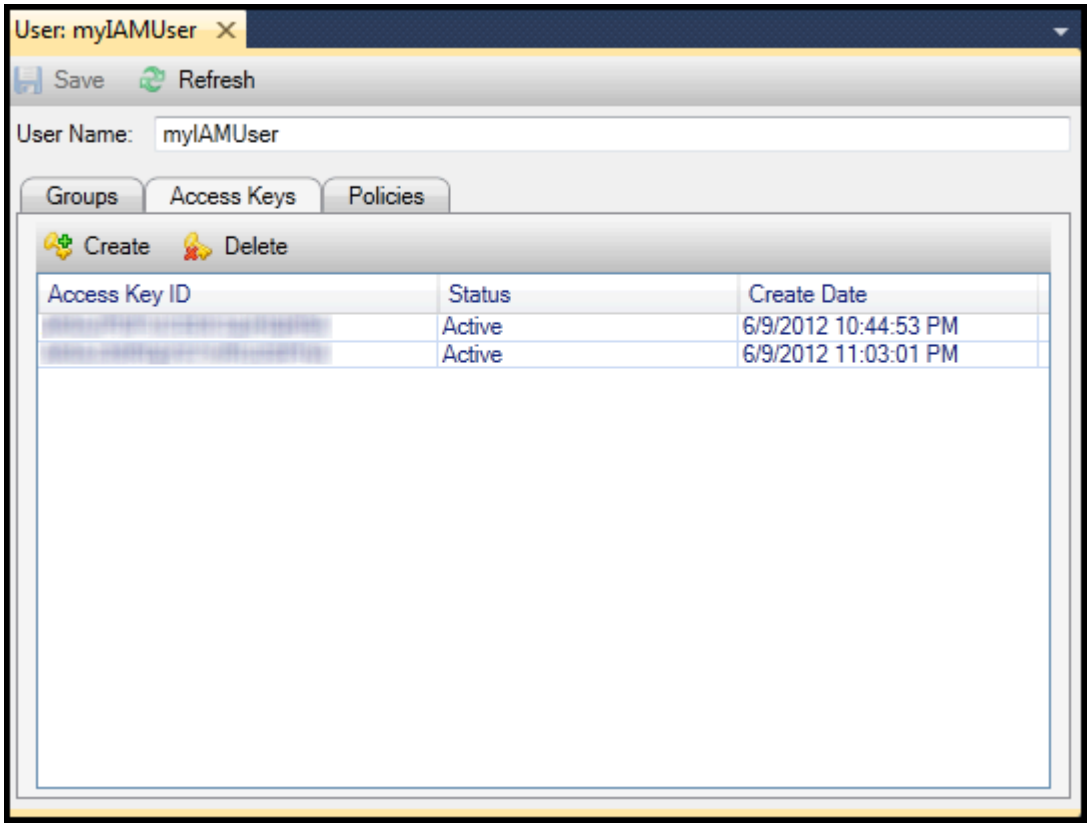
生成 IAM 使用者的登入資料

使用適用 Toolkit for Visual Studio，您可以生成訪問密鑰 ID 和密鑰，用於對AWS。也可以指定這些密鑰，以便通過工具包訪問 Amazon Web Services。如需如何指定與 Toolkit 一起使用的登入資料的詳細資訊，請參 Creds。如需如何安全地處理登入資料的詳細資訊，請參管理的最佳實務AWS存取金鑰。

工具包不能用於為 IAM 使用者生成密碼。

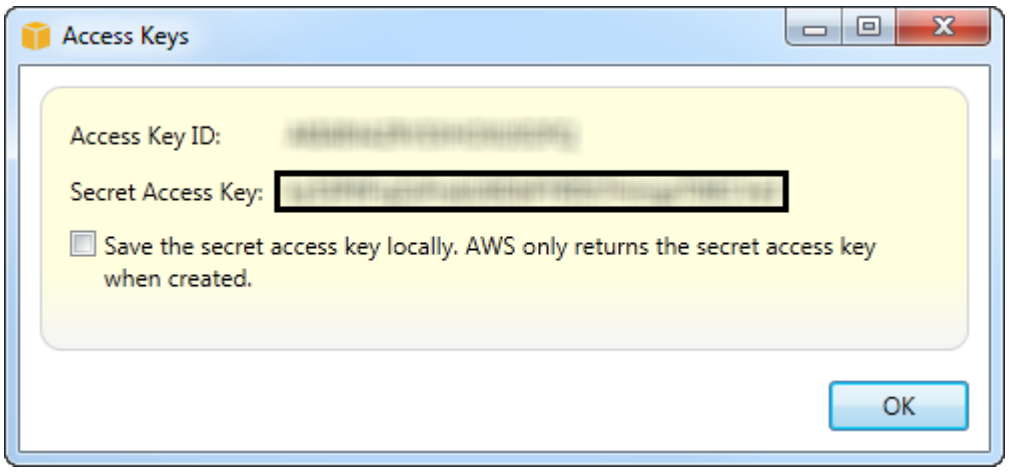
生成 IAM 使用者的登入資料

1. InAWS，請為 IAM 使用者打開上下文 (右鍵按一下) 菜單，然後選擇Edit (編輯)。



2. 要生成憑據，請在存取金鑰選項卡上選擇建立。

   您可以為每位 IAM 使用者產生最多兩組登入資料。如果您已經有兩組登入資料，並需要建立額外的登入資料，您必須刪除其中一組現有登入資料。

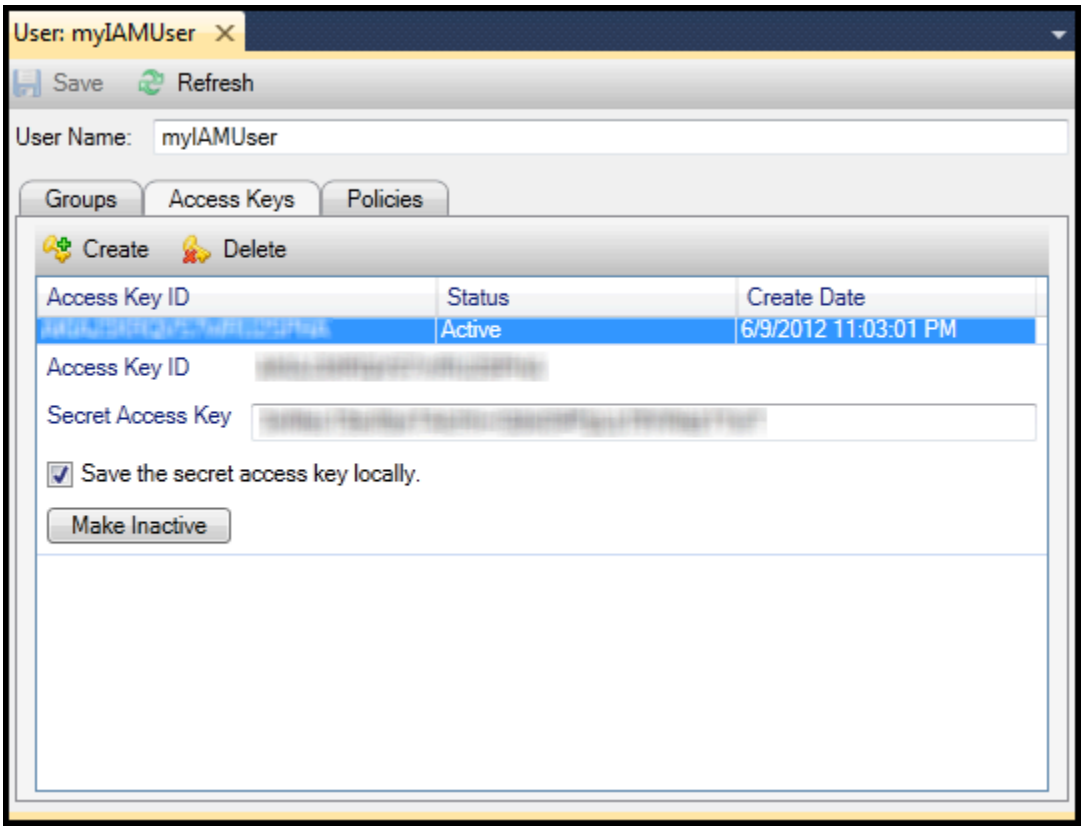


reate credentials for IAM user

如果希望 Toolkit 將私有訪問密鑰的加密副本保存到本地驅動器，請選擇在本地保存私有訪問密鑰。AWS僅在創建時返回私有訪問密鑰。您還可以從對話框中複製私有訪問密鑰並將其保存在安全位置。

3. 選擇 OK (確定)。

在您生成登入資料之後，您可以從存取金鑰選項卡。如果您選擇了讓 Toolkit 在本地保存密鑰的選項，則該密鑰將顯示在此處。



Create credentials for IAM user

如果您自己保存了密鑰，並希望 Toolkit 保存密鑰，請在Secret Access Key (私密存取金鑰)框中，鍵入私有訪問密鑰，然後選擇在本機保存私密存取金鑰。

若要停用憑證，請選擇Make Inactive (設為閒置)。（如果您懷疑憑據已被盜用，則可以執行此操作。如果您收到證書是安全的，則可以重新激活憑據。）

建立 IAM 角色

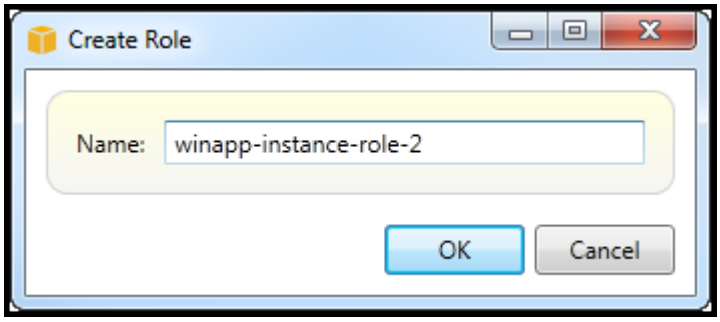
適用 Toolkit for Visual Studio 支援 IAM 角色的建立和配置。與用户和組一樣，您可以將策略附加到 IAM 角色。您可以將 IAM 角色與 Amazon EC2 執行個體建立關聯。與 EC2 執行個體的關聯是通過執行個體描述檔，這是角色的邏輯容器。在 EC2 實例上運行的應用程序將自動授予與 IAM 角色關聯的策略所指定的訪問級別。即使應用程式沒有指定其他AWS登入資料。

例如，您可以建立僅限存取 Amazon S3 的角色，並將政策連接到該角色。將此角色與 EC2 實例關聯後，您可以在該實例上運行應用程序，該應用程序將有權訪問 Amazon S3，但不能訪問任何其他服務或資源。這種方法的優點是您不需要關注安全傳輸和存儲AWS證書。

如需 IAM 角色的詳細資訊，請前往《IAM 使用者指南》中的使用 IAM 角色。有關訪問的程序示例AWS使用與 Amazon EC2 實例關聯的 IAM 角色，請轉到AWS的開發人員指南爪哇、.NET、PHP, 和 Ruby (使用 IAM 設置證書、建立 IAM 角色，和處理 IAM 政策。

若要建立一個 IAM 角色

1. InAWSExplorerIdentity and Access Management下，開啟內容 (按右鍵) 功能表，然後選擇角色然後選擇建立角色。
2. 在 中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 IAM 角色將顯示在角色在Identity and Access Management。

如需如何建立政策，並將它連接到角色的詳細資訊，請參建立 IAM 政策。

建立 IAM 政策

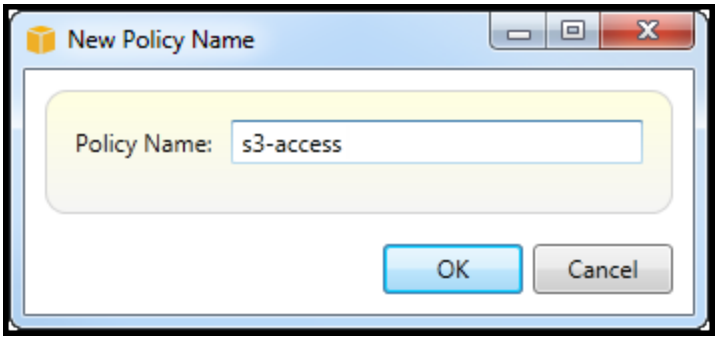
策略是 IAM 的基礎。策略可以與 IAM 關聯實體（如使用者、羣組或角色）。政策指定為使用者、組或角色啟用的訪問級別。

建立 IAM 政策

InAWS資源管理器中，展開AWS Identity and Access Management節點，然後展開實體類型的節點（Groups (群組)、角色, 或使用者），您將附加策略。例如，打開 IAM 角色的上下文菜單，然後選擇Edit (編輯)。

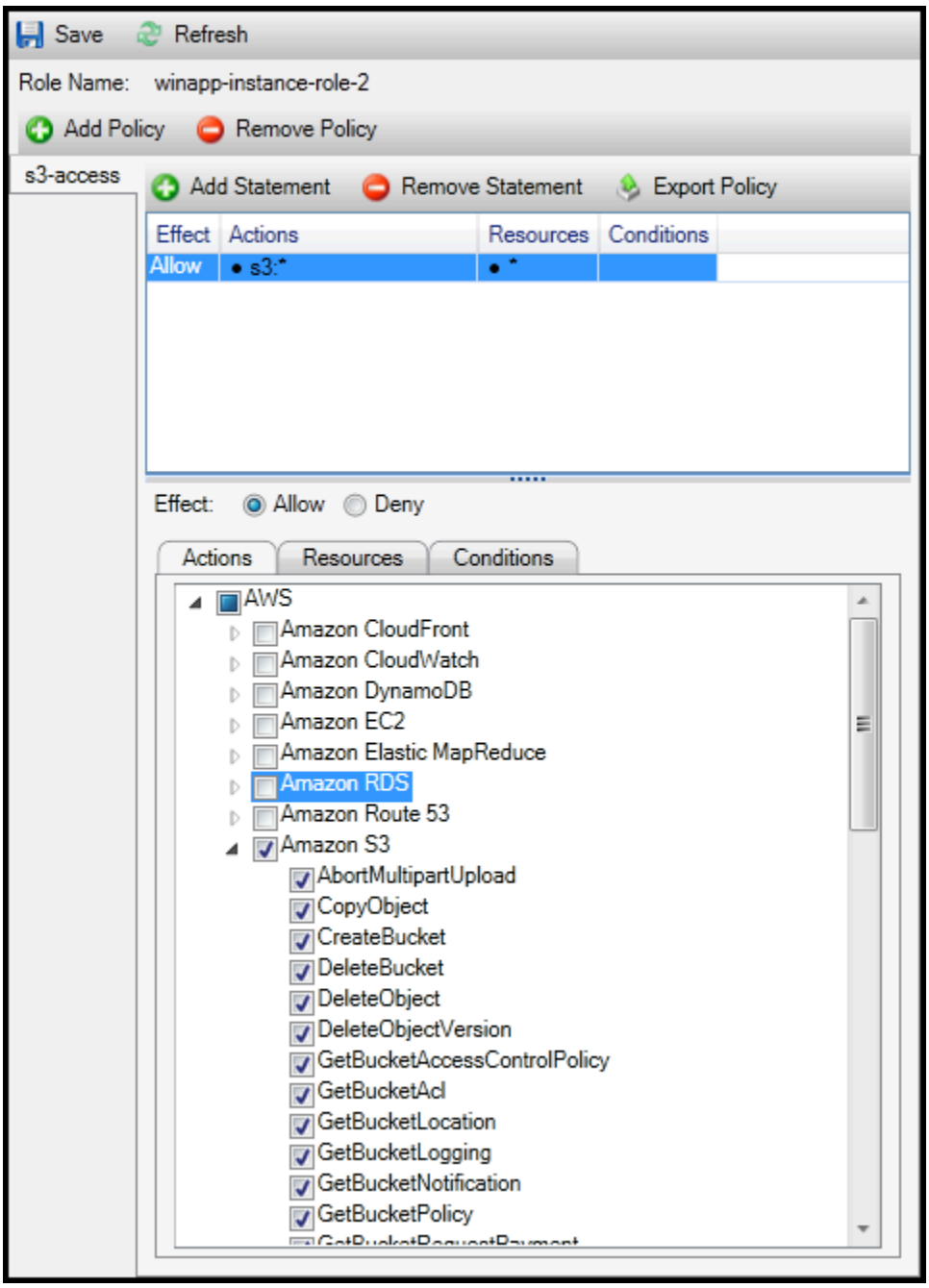
與該角色關聯的選項卡將出現在AWSExplorer 選擇添加政策鏈接。

在 中新建立政策名稱對話方塊中，輸入政策的名稱 (例如 s3-訪問)。



New Policy Name dialog box

在策略編輯器中，添加策略語句以指定要向角色提供的訪問級別（在本示例中，與策略關聯的 winapp-實例-角色-2）。在此示例中，策略提供了對 Amazon S3 的完全訪問權限，但不能訪問任何其他資源。



Specify IAM policy

要獲得更精確的訪問控制，您可以在策略編輯器中展開子節點，以允許或禁止與 Amazon Web Services 相關聯的操作。

編輯策略後，選擇Save鏈接。

AWS Lambda

開發和部 Lambda 以. 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。該 Toolkit for Visual Studio 包包括. AWS Lambda NET 核心專案範本視覺工作室。

如需有關的詳細資訊 AWS Lambda，請參閱 AWS Lambda 開發人員指南。

如需 .NET 核心的詳細資訊，請參閱 .Microsoft 核心指南。如需 Windows、macOS 和 Linux 平台的 .NET 核心必要條件和安裝指示，請參閱 .NET 核心下載。

下列主題說明如何使用適 AWS Lambda 用 Toolkit for Visual Studio。

主題

- 基本 AWS Lambda 項目
- 基本 AWS Lambda 項目創建碼頭圖像
- 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：使用 Amazon 記錄架構與建 AWS Lambda 立應用程式日誌

基本 AWS Lambda 項目

您可以使 Lambda Microsoft 核心專案範本，在 AWS Toolkit for Visual Studio.

建立視覺工作室 .NET 核心 Lambda 專案

您可以使用 Lambda 視覺工作室範本和藍圖來協助加速專案初始化。Lambda 藍圖包含預先撰寫的函數，可簡化彈性專案基礎的建立作業。

> **Note**
>
> Lambda 服務對不同的封裝類型有資料限制。如需有關資料限制的詳細資訊，請參閱 Lambda 使用者指南中的 AWS Lambda 配額主題。

若要在視覺工作室中建立 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aws lambda在「搜尋」欄位中輸入。選擇 L AWS ambda 專案 (.NET 核心-C#) 範本。
3. 在「名稱」欄位中，輸入**AWSLambdaSample**，指定您想要的檔案「位置」，然後選擇「建立」以繼續。
4. 從 [選取藍圖] 頁面中，選取空白函式藍圖，然後選擇 [完成] 以建立 Visual Studio 專案。

檢閱專案檔

有兩個專案檔案可供檢閱：aws-lambda-tools-defaults.json和Function.cs。

下列範例顯示aws-lambda-tools-defaults.json檔案，該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：
>
> - 函數處理程序：指定 Lambda 函數運行時運行的方法
> - 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。
> - 如果重命名函數，類或程序集，那麼您還需要更新aws-lambda-tools-defaults.json文件中的相應字段。

```
{
  "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,
```

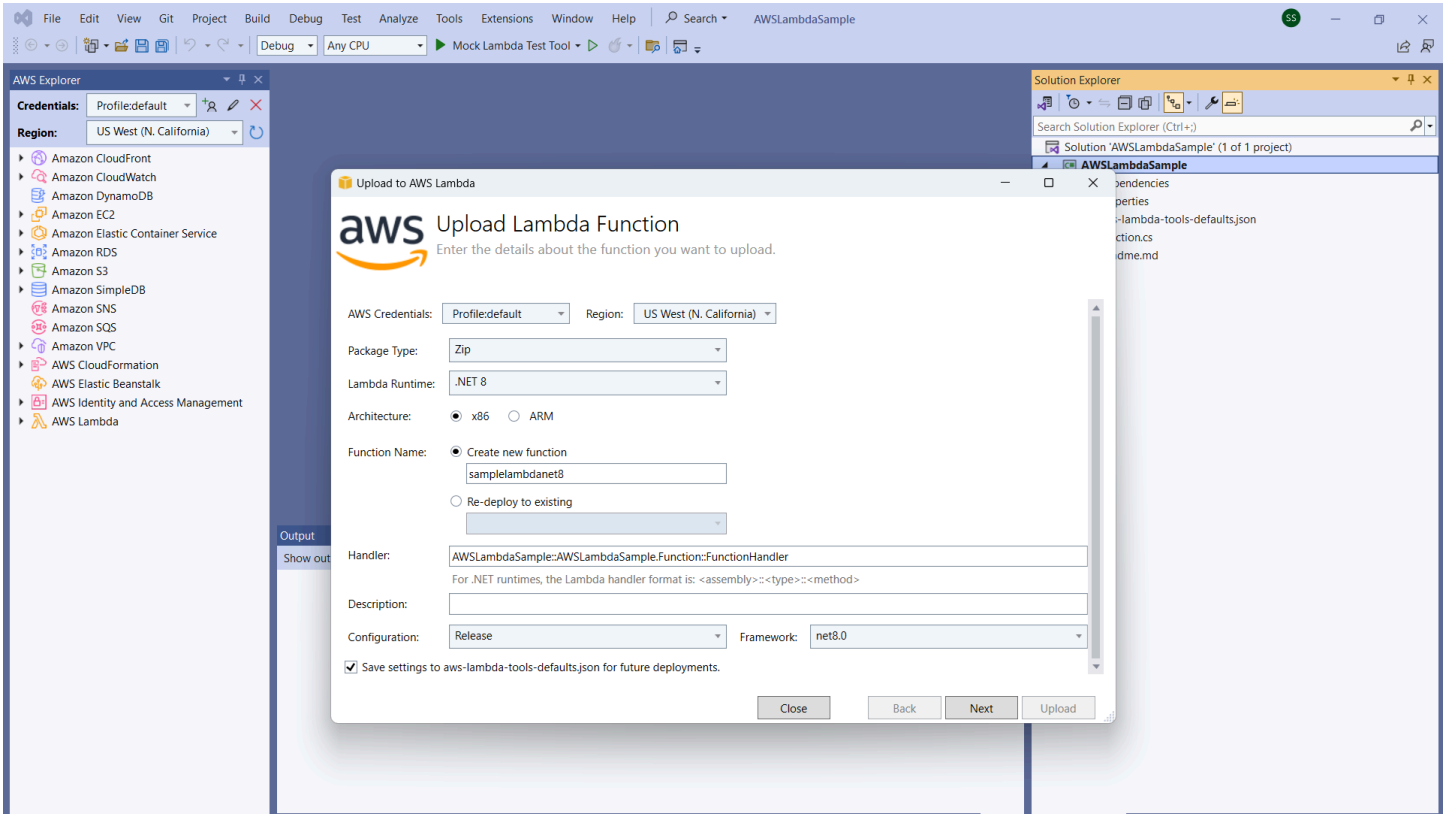
```
  "function-timeout": 30,
  "function-handler": "AWSLambdaSample::AWSLambdaSample.Function::FunctionHandler"
}
```

檢查Function.cs檔案。 Function.cs定義 c# 函數以公開為 Lambda 函數。這FunctionHandler是 Lambda 函數執行時執行的 Lambda 功能。在這個項目中，定義了一個函數：FunctionHandler，它調ToUpper()用輸入文本。

您的專案現在已準備就緒，可以發佈到 Lambda。

發佈至 Lambda

下列程序和影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. 展開「檢視」並選擇「 AWS 檔案總管」，導覽至AWS 檔案總管
2. 在解決方案總管中，開啟 (以滑鼠右鍵按一下) 要發佈之專案的內容功能表，然後選擇「發佈至 AWS Lambda」以開啟「上傳 Lambda 函數」視窗。
3. 在「上傳 Lambda 函數」視窗中，完成下列欄位：

a. Package 類型：選擇**Zip**。建置程序的結果會建立一個 ZIP 檔案，並將上傳至 Lambda。或者，您也可以選擇「P ackage 類型」**Image**。教學課程：基本 Lambda 專案建立 Docker 映像檔說明如何使用 P ackage 類型**Image**發佈。

b. Lambda 執行階段：從下拉式功能表中選擇 Lambda 執行階段

c. 架構：為您偏好的架構選擇徑向。

d. 函數名稱：為建立新函數選取徑向，然後輸入 Lambda 執行個體的顯示名稱。 AWS 檔案總管和 AWS Management Console 顯示器都會參考此名稱。

e. 處理程式：使用此欄位可指定函數處理常式。例如：**AWSLambdaSample::AWSLambdaSample.Function::FunctionHandler**。

f. (選擇性) 說明：從中輸入要與執行環境一起顯示的描述性文字 AWS Management Console。

g. 組態：從下拉式功能表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：選取此方塊可將目前的設定儲存`aws-lambda-tools-defaults.json`為 future 部署的預設值。

j. 選擇下一步，繼續進行「進階功能詳細資訊」視窗。

4. 在「進階功能詳細資訊」視窗中，完成下列欄位：

a. 角色名稱：選擇與您帳戶相關聯的角色。此角色會為函式中的程式碼所發出的任何 AWS 服務呼叫提供暫時認證。如果您沒有角色，請捲動以在下拉式選取器中尋找根據 AWS 受管理策略的新角色，然後選擇AWSLambdaBasicExecutionRole。此角色具有最小的存取權限。

> **Note**
>
> 您的帳戶必須具有執行 IAM ListPolicies 動作的權限，否則「角色名稱」清單將為空，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。

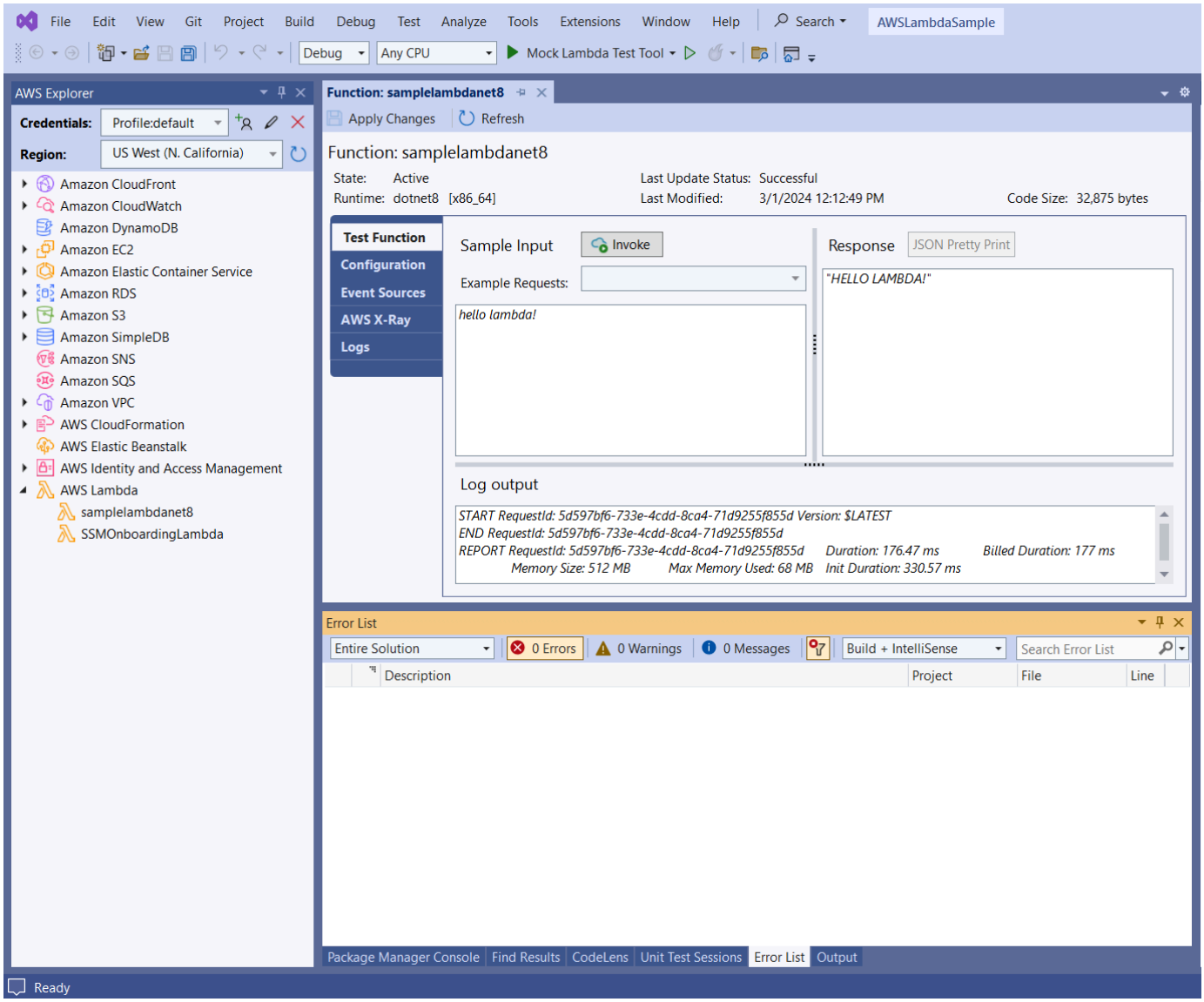
c. (選擇性) 設定 Lambda 函數需要的任何環境變數。這些密鑰由默認服務密鑰自動加密，該密鑰是免費的。或者，您可以指定一個 AWS KMS 鍵，其中有一個收費。KMS 是一項受管服務，可用來建立和控制用於加密資料的加密金鑰。如果您有 AWS KMS 金鑰，您可以從清單中選取金鑰。

5. 選擇「上傳」以開啟「上傳功能」視窗並開始上傳處理。

> **Note**
>
> 上傳函數時，會顯示 [上載功能] 頁面 AWS。若要在上傳之後保持精靈開啟，以便您可以檢視報表，請在上傳完成前清除表單底部表單底部的 [成功完成時自動關閉精靈]。
> 函數上傳之後，您的 Lambda 函數就會上線。「函數：檢視」頁面隨即開啟，並顯示新的 Lambda 函數組態。

6. 在「測試函數」標籤`hello lambda!`中，輸入文字輸入欄位，然後選擇叫用以手動呼叫 Lambda 函數。您的文本顯示在「響應」標簽中，并轉換為大寫字母。

> **Note**
>
> 您可以隨時重新開啟 F unction: 檢視，方法是在AWS Lambda節點下方的AWS 檔案總管中連按兩下已部署的執行個體。



7. (選擇性) 若要確認您已成功發佈 Lambda 函數，請登入， AWS Management Console 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。

清理

如果您不打算繼續使用此範例進行開發，請刪除您部署的函數，這樣就不會針對帳戶中未使用的資源向您收取費用。

> **Note**
>
> Lambda 會自動為您監控 Lambda 函數，並透過 Amazon 報告指標 CloudWatch。若要監控和疑難排解您的函數，請參閱 AWS Lambda 開發人員指南中的使用 Amazon 疑難排解和監控 AWS Lambda 函數 CloudWatch主題。

若要刪除您的函數

1. 從AWS 總管中展開AWS Lambda節點。
2. 在部署的執行個體上按一下滑鼠右鍵，然後選擇

基本 AWS Lambda 項目創建碼頭圖像

您可以使用 Toolkit for Visual Studio，將您的 AWS Lambda 函式部署為泊塢視窗映像。使用 Docker，您可以更好地控制運行時。例如，您可以選擇自訂執行階段，例如 .NET 8.0。您可以使用與任何其他容器映像相同的方式部署 Docker 映像檔。本教程密切模仿教程：基本 Lambda 項目，有兩個區別：

- 一個碼頭文件包含在項目中。
- 會選擇替代的發行組態。

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 的其他資訊 AWS Toolkit for Visual Studio，請參閱本使用者指南 AWS Toolkit for Visual Studio主題中的使用 AWS Lambda 範本。

建立視覺工作室 .NET 核心 Lambda 專案

您可以使用 Lambda 視覺工作室範本和藍圖來協助加速專案初始化。Lambda 藍圖包含預先撰寫的函數，可簡化彈性專案基礎的建立作業。

若要建立 .NET 核心 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**`aws lambda`**在「搜尋」欄位中輸入。選擇 L AWS ambda 專案 (.NET 核心-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。

審閱專案檔

下列各節將檢查 .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

A `Dockerfile` 執行三個主要動作：

- `FROM`：建立用於此影像的基本影像。此基本映像檔提供 .NET 執行階段、Lambda 執行階段和殼層指令碼，可提供 Lambda .NET 程序的進入點。
- `WORKDIR`：將映像檔的內部工作目錄建立為`/var/task`。
- `COPY`: 將從構建過程生成的文件從其本地位置複製到圖像的工作目錄中。

下列是您可以指定的選用`Dockerfile`動作：

- `ENTRYPOINT`：基本映像已包含一個`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，則會覆寫該基準入口點。
- `CMD`：指示 AWS 您要執行的自定義代碼。它期望您的自定義方法具有完全合格的名稱。這一行需要直接包含在 Dockerfile 中，或者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) 藍圖所建立的 Docker 檔案範例。

```
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

該`aws-lambda-tools-defaults.json`文件是用來指定的默認值工 Toolkit for Visual Studio 部署嚮導和. NET 核心 CLI。下列清單說明您可以在`aws-lambda-tools-defaults.json`檔案中設定的欄位。

- `profile`：設置您的 AWS 個人資料。
- `region`：設定儲存資源的 AWS 區域。
- `configuration`：設置用於發布函數的配置。
- `package-type`：將部署包類型設置為容器映像或 .zip 文件存檔。
- `function-memory-size`：以 MB 為單位設置函數的內存分配。
- `function-timeout`: 逾時是 Lambda 函數可以執行的時間上限 (以秒為單位)。您可以以 1 秒的增量調整，最大值為 15 分鐘。
- `docker-host-build-output-dir`：設定建置程序的輸出目錄，該目錄與中的指示相關聯。`Dockerfile`
- `image-command`: 是您的方法的完全限定名稱，您希望 Lambda 函數運行的代碼。語法為：`{Assembly}::{Namespace}.{ClassName}::{MethodName}`。如需詳細資訊，請參閱處理常式簽章。這`image-command`裡的設定會在稍後的 Visual Studio 的發行精靈中預先填入這個值。

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

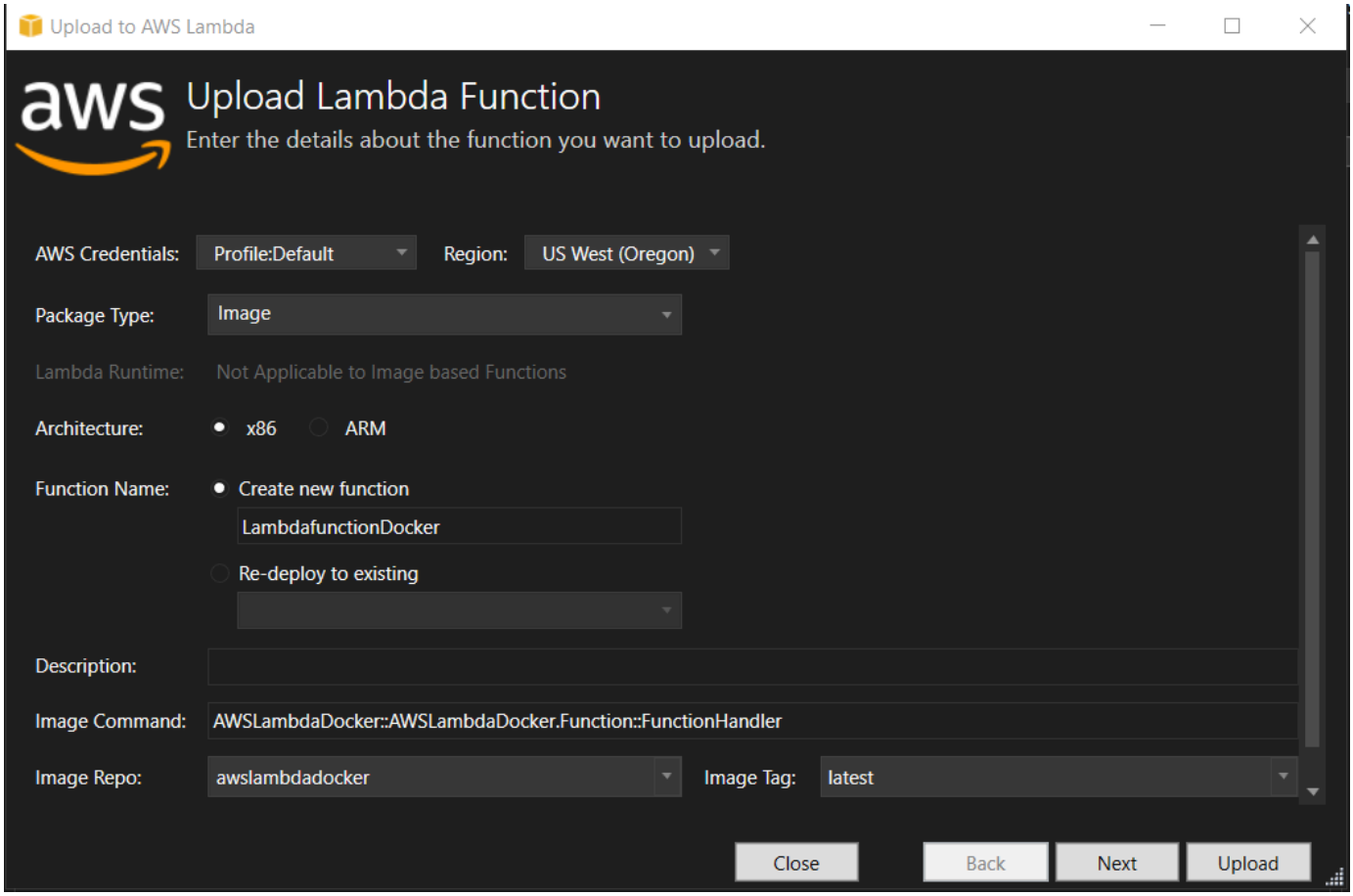
該Function.cs文件定義了 C# 函數被公開為 Lambda 函數。FunctionHandler是 Lambda 函數執行時執行的 Lambda 功能。在這個項目中，FunctionHandler調ToUpper()用輸入文本。

發佈到 Lambda

由構建過程生成的碼頭映像會上傳到 Amazon Elastic Container Registry (Amazon ECR)。Amazon ECR 是一種全受管的 Docker 容器登錄，可用來存放、管理和部署 Docker 容器映像。Amazon ECR 會託管映像，Lambda 接著會參考該映像檔，以便在叫用時提供程式設計的 Lambda 功能。

若要將您的函數發佈至 Lambda

1. 從 [方案總管] 中，開啟 (按一下滑鼠右鍵) 專案的內容功能表，然後選擇 [發佈 AWS Lambda至] 以開啟 [上傳 Lambda 函數] 視窗。
2. 在「上傳 Lambda 函數」頁面中，執行下列動作：



a. 對於「P ackage 類型」，**Image**已自動選取為「Package 類型」，因為發佈精靈在您的專案**Dockerfile**中偵測到了「封裝類型」。

b. 在函數名稱中，輸入 Lambda 執行個體的顯示名稱。這個名稱是在 AWS 資源管理器中顯示的參考名稱和 AWS Management Console.

c. 在「說明」中，輸入要與執行個體一起顯示的文字 AWS Management Console。

d. 對於映像命令，請輸入您希望 Lambda 函數執行之方法的完整路徑：**AWSLambdaDocker::AWSLambdaDocker.Function::FunctionHandler**

> **Note**
>
> 在此處輸入的任何方法名稱都將覆蓋 Docker 文件中的任何 CMD 指令。只有在`Dockerfile`包含指示如何啟動 Lambda 函數`CMD`時，輸入影像命令才是選用的。

e. 對於映像存放庫，請輸入新的或現有的 Amazon 彈性容器登錄名稱。建置程序建立的 Docker 映像會上傳至此登錄。正在發布的 Lambda 定義將引用該 Amazon ECR 映像。

f. 在「影像標籤」中，輸入要與儲存庫中的映像建立關聯的 Docker 標籤。

g. 選擇下一步。

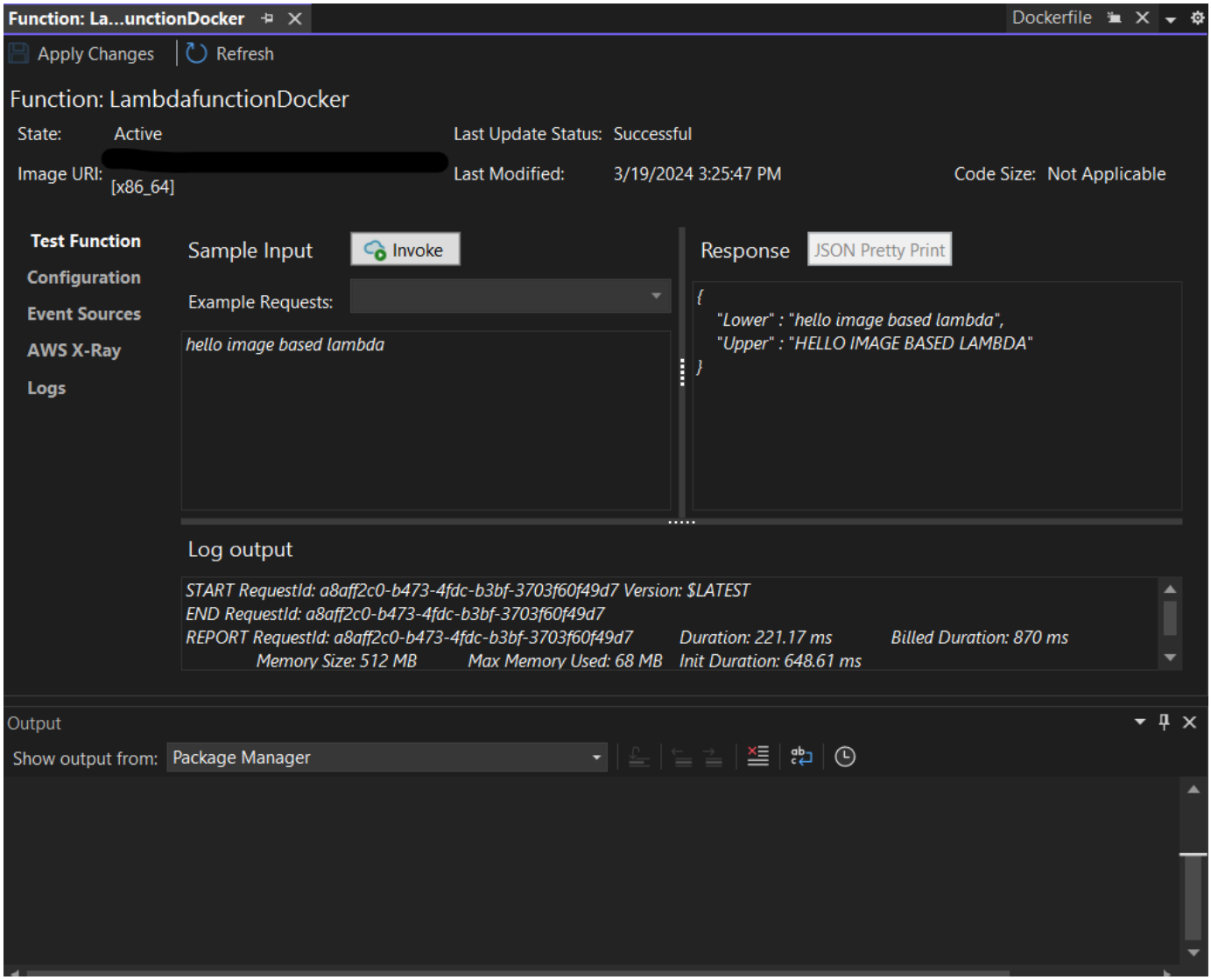
3. 在 [進階功能詳細資料] 頁面的角色名稱中，選擇與您帳戶相關聯的角色。該角色用於為函數中的程式碼發出的任何 Amazon Web Services 呼叫提供臨時登入資料。如果您沒有角色，請選擇 [根據 AWS 受管理的策略新增角色]，然後選擇AWSLambdaBasicExecutionRole。

> **Note**
>
> 您的帳戶必須具有執行 IAM ListPolicies 動作的權限，否則「角色名稱」清單將為空。

4. 選擇「上載」以開始上傳與發佈程序。

> **Note**
>
> 上載函數時會顯示 [上載功能] 頁面。然後，發佈程序會根據組態參數建立映像、視需要建立 Amazon ECR 儲存庫、將映像上傳到儲存庫，然後建立使用該映像參考該存放庫的 Lambda。
>
> 上傳函數之後，「函數」頁面會開啟並顯示新的 Lambda 函數組態。

5. 若要手動呼叫 Lambda 函數，請在 [測試函數] 索引標籤上輸`hello image based lambda`入要求自由文字輸入欄位，然後選擇 [叫用]。你的文字（轉換為大寫）會顯示在「回應」中。



6. 若要檢視儲存庫，請在AWS 資源管理器的 Amazon 彈性容器服務下，選擇儲存庫。

   您可以隨時重新開啟 F unction: 檢視，方法是在AWS Lambda節點下方的AWS 檔案總管中連按兩下已部署的執行個體。

   > **Note**
   >
   > 如果您的 AWS 資源管理器窗口未打開，則可以通過查看-> AWS 資源管理器將其停靠

7. 請注意組態索引標籤上的其他影像特定組態選項。此選項卡提供了一種覆蓋 `ENTRYPOINTCMD`，並且可能`WORKDIR`已在 Dockerfile 中指定的方法。「描述」是您在上傳/發佈期間輸入的說明 (如果有的話)。

清理

如果您不打算繼續使用此範例進行開發，請記得刪除已部署的函數和 ECR 映像檔，這樣就不會針對帳戶中未使用的資源向您收取費用。

- 若要刪除函數，請以滑鼠右鍵按一下位於 AWS Explorer 中AWS Lambda節點下方的已部署執行個體。
- 您可以在 Amazon 彈性容器服務-> 儲存庫下的AWS 資源管理器中刪除儲存庫。

後續步驟

如需建立和測試 Lambda 映像的相關資訊，請參閱搭配 Lambda 使用容器映像。

如需容器映像部署、權限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 範本建立無伺服器 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，即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 AWS Lambda 函數集合，並將它們與任何必要的 AWS 資源作為一個整個應用程式部署，用 AWS CloudFormation 來協調部署。

如需有關設定的先決條件和資訊 AWS Toolkit for Visual Studio，請參閱使用 Visual Studio AWS 工具組中的 AWS Lambda 範本。

主題

- 建立新的 AWS 無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 AWS 無伺服器應用程式專案

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

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，確定 [語言]、[平台] 和 [專案類型] 下拉式方塊已設定為 [全部...]，然後**aws lambda**在 [搜尋] 欄位中輸入。
3. 選取含測試的AWS 無伺服器應用程式 (.NET 核心-C#) 範本。

   > **Note**
   >
   > 具有測試的AWS 無伺服器應用程式 (.NET Core-C#) 範本可能不會填入結果的頂端。

4. 按一下「下一步」以開啟「規劃新專案」對話方塊。
5. 在「配置新專案」對話方塊中，輸入 **ServerlessPowertools**「名稱」，然後根據您的偏好填寫其餘欄位。選擇建立按鈕以繼續執行「選取藍圖」對話方塊。
6. 從 [選取藍圖] 對話方塊中選擇適用於 AWS Lambda藍圖的 Powertools，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

以下各節詳細介紹三個為您的專案建立的無伺服器應用程式檔案：

1. 無伺服器. 範本
2. Functions.cs
3. aws-lambda-tools-defaults.json

1。無服務器. 模板

`serverless.template`檔案是用來宣告您的無伺服器函式和其他 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，該函數將以HTTP *Get*作業形式透過 Amazon API Gateway 公開。您可以編輯此範本以自訂現有功能，或新增應用程式所需的更多功能和其他資源。

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...宣告欄位都類似於 Lambda 專案部署的欄位。Powertools 記錄日誌、測量結果和追蹤是透過下列環境變數進行設定：

- 電源工具 _ 服務 _ 名稱 = ServerlessGreeting
- 電源工具 _ 日誌級別 = 信息
- 電動工具記錄器案例 = PascalCase
- 電源工具 _ 跟踪器 _ 捕獲 _ 響應 = 真
- 電源工具 _ 跟踪器 _ 捕獲 _ 錯誤 = 真
- 指標 _ 名稱空間 = ServerlessGreeting

有關環境變量的定義和其他詳細信息，請參閱 Powertools 的 AWS Lambda 參考資料網站。

2. Functions.cs

Functions.cs是一個包含 C# 方法的類文件，該文件映射到模板文件中聲明的單個函數。Lambda 函數會回應來自 API Gateway 的HTTP Get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提供部 AWS 署精靈內的預設值，以及新增至 .NET 核心 CLI 的 AWS Lambda 命令。以下是此專aws-lambda-tools-defaults.json案包含的檔案範例：

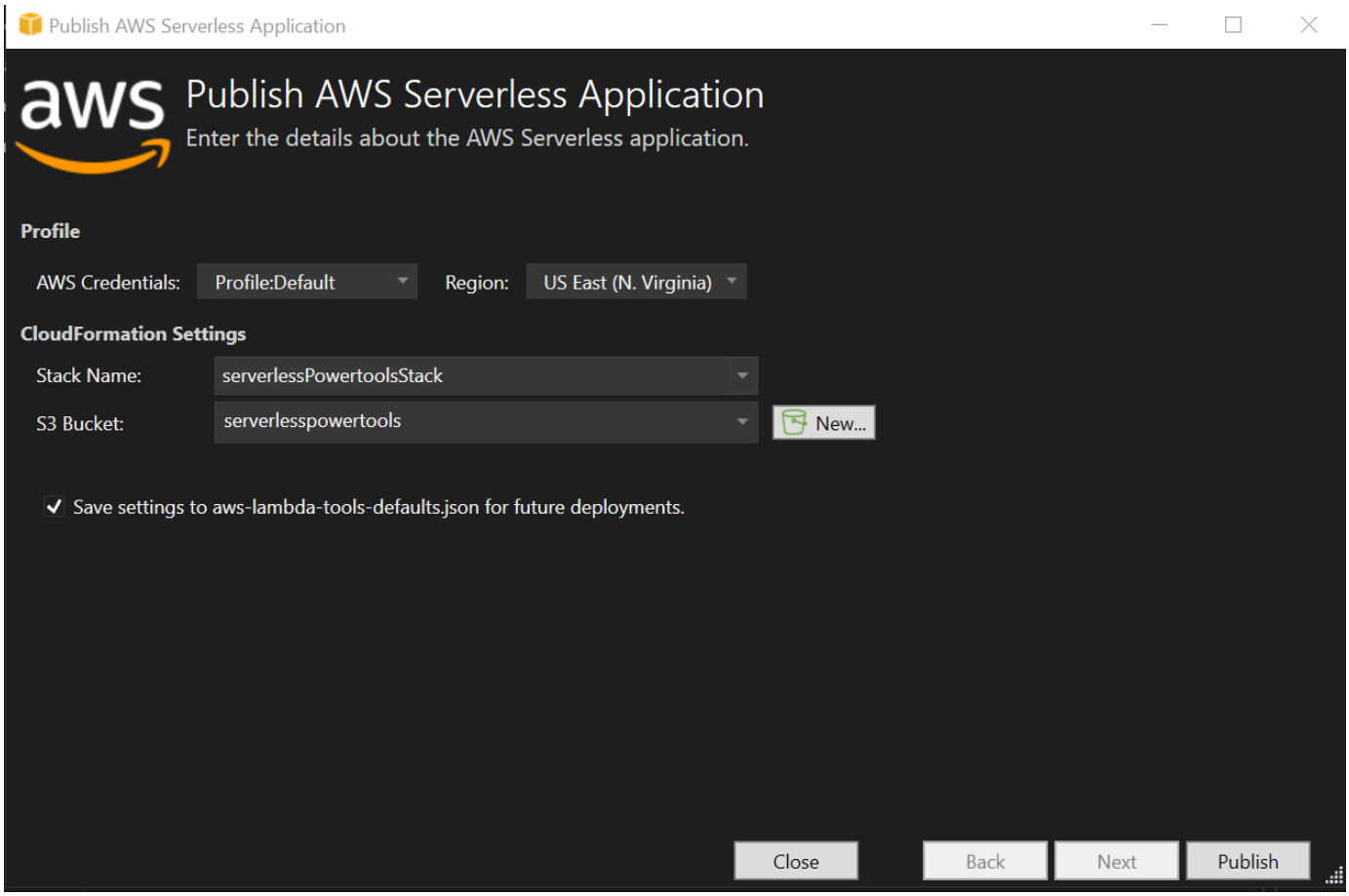
```
{
  "profile": "Default",
  "region": "us-east-1",
  "configuration": "Release",
  "s3-prefix": "ServerlessPowertools/",
  "template": "serverless.template",
  "template-parameters": ""
}
```

部署無伺服器應用程式

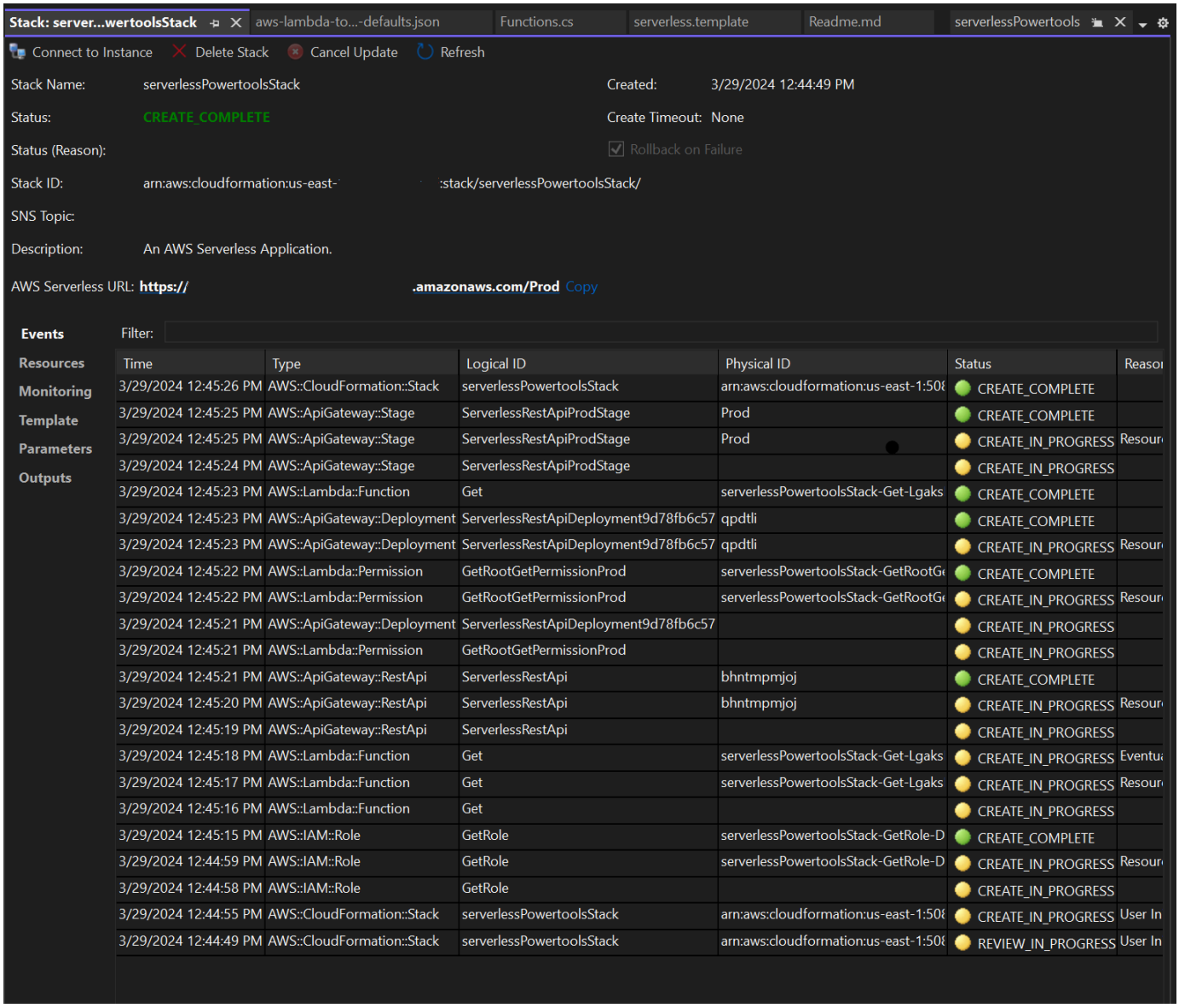
若要部署無伺服器應用程式，請完成以下步驟

1. 從解決方案總管中，開啟 (按一下滑鼠右鍵) 專案的內容功能表，然後選擇「發佈到 AWS Lambda」以開啟「發佈 AWS 無伺服器應用程式」對話方塊。
2. 從「發佈 AWS 無伺服器應用程式」對話方塊的「 AWS 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，選擇應用程式套件包將上傳到的 Amazon S3 儲存貯體，或選擇新增儲存貯體... 按鈕並輸入一個新的 Amazon S3 存儲桶的名稱。然後選擇「發佈」以發佈以部署您的應用程式。

> **Note**
>
> 您的 AWS CloudFormation 堆疊和 Amazon S3 儲存貯體必須位於相同的 AWS 區域。專案的其餘設定在`serverless.template`檔案中定義。



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



測試無伺服器應用程式

堆疊建立完成後，您可以使用AWS 無伺服器 URL 檢視應用程式。如果您在未新增任何其他函數或參數的情況下完成此教學課程，存取 AWS 無伺服器 URL 會在您的網頁瀏覽器中顯示以下片語：`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 Lambda 應用程式，該應用程式使用 Amazon Rekognition 標記含偵測到的標籤的 Amazon S3 物件。

如需有關設定的先決條件和資訊 AWS Toolkit for Visual Studio，請參閱使用 Visual Studio AWS 工具組中的 AWS Lambda 範本。

建立視覺工作室 .NET 核心 Lambda 影像 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**
>
> 建立後，您的應用程式會有一個包含兩個專案的解決方案：包含要部署到 Lambda 的 Lambda 函數程式碼的原始程式碼，以及使用 xUnit 在本機測試函數的測試專案。
> 有時候 Visual Studio 找不到您專案的所有 NuGet 參考資料。這是因為藍圖需要必須從中 NuGet擷取的相依性。建立新專案時，Visual Studio 只會從中擷取本機參考，而不會從中擷取遠端參考 NuGet。若要修正 NuGet 錯誤：以滑鼠右鍵按一下您的參考檔，然後選擇

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，確定 [語言]、[平台] 和 [專案類型] 下拉式方塊已設定為 [全部...]，然後**`aws lambda`**在 [搜尋] 欄位中輸入。
3. 選擇AWS Lambda 與測試（ .NET 核心-C＃ ）模板。
4. 按一下「下一步」以開啟「規劃新專案」對話方塊。
5. 在 [設定新專案] 對話方塊中，輸入 "ImageRekognition" 做為 [名稱]，然後根據您的偏好填寫其餘欄位。選擇建立按鈕以繼續執行「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**
>
> 此藍圖提供程式碼以偵聽 Amazon S3 事件，並使用 Amazon Rekognition 偵測標籤，並將其新增至 S3 物件做為標籤。

審閱專案檔案

以下各節將檢查這些專案檔案：

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。默認情況下，Lambda 僅接受類型的輸入參數和返回類型System.IO.Stream。您必須註冊序列化程式，才能將類型類別用於輸入參數和傳回類型。組件屬性註冊 Lambda JSON 序列化程序，它使用流Newtonsoft.Json轉換為類型類。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
```

該類有兩個構造函數。第一個是 Lambda 叫用函數時使用的預設建構函式。此建構函式會建立 Amazon S3 和 Amazon Rekognition 服務用戶端。建構函式也會在您部署該函式時，從指派給函數的 IAM 角色擷取這些用戶端的 AWS 認證。用戶端的 AWS 區域設定為執行 Lambda 函數的區域。在此藍圖中，如果 Amazon Rekognition 服務對標籤具有最低信賴度，您只想將標籤新增至 Amazon S3 物件。此構造函數檢查環境變量MinConfidence以確定可接受的置信水平。您可以在部署 Lambda 函數時設定此環境變數。

以下是中第一個類別建構函式的範例Function.cs：

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

    var environmentMinConfidence =
 System.Environment.GetEnvironmentVariable(MIN_CONFIDENCE_ENVIRONMENT_VARIABLE_NAME);
    if(!string.IsNullOrWhiteSpace(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}");
    }
}
```

下面的例子演示了如何第二個構造函數可以用於測試。測試專案會設定自己的 S3 和 Rekognition 用戶端，並將它們傳入：

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

下面是Function.cs文件中的FunctionHandler方法的一個例子。

```
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 在構建實例後調用的方法。請注意，輸入參數的類型S3Event不是Stream。您可以這樣做，因為已註冊的 Lambda JSON 序列化程式。包S3Event含 Amazon S3 中觸發之事件的所有相關資訊。此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。偵測到標籤之後，它們會作為標籤新增至 S3 物件。

> **Note**
>
> 程式碼包含對`Console.WriteLine()`. 當函數在 Lambda 中執行時，所有呼叫都會`Console.WriteLine()`重新導向至 Amazon CloudWatch 日誌。

2. aws-lambda-tools-defaults.json

該`aws-lambda-tools-defaults.json`檔案包含藍圖已設定為預先填入部署精靈中某些欄位的預設值。它也有助於設定與 .NET 核心 CLI 整合的命令列選項。

若要存取 .NET Core CLI 整合，請瀏覽至函式的專案目錄並輸入**`dotnet lambda help`**。

> **Note**
>
> 函數處理常式會指出 Lambda 在回應叫用函數時呼叫的方法。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。

部署功能

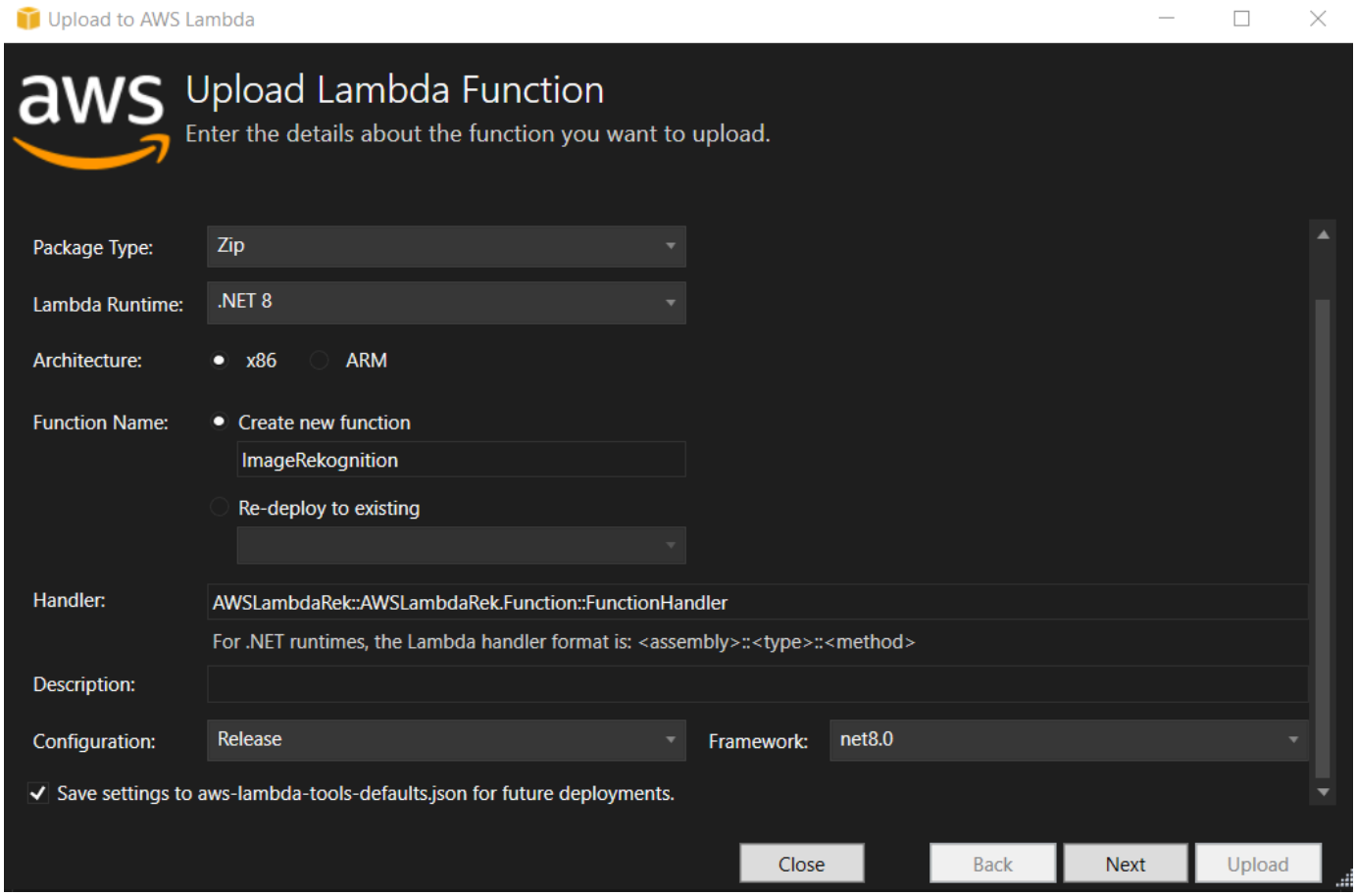
下列程序說明如何部署 Lambda 函數。

1. 在解決方案總管中，以滑鼠右鍵按一下 Lambda 專案，然後選擇「發佈到 AWS Lambda」以開啟「上傳到 AWS Lambda」

   > **Note**
   >
   > 預置值會從`aws-lambda-tools-defaults.json`檔案中擷取。

2. 從「上傳至 AWS Lambda」視窗的「功能名稱」欄位中輸入名稱，然後選擇「下一步」按鈕，以前進至「進階功能明細」視窗。

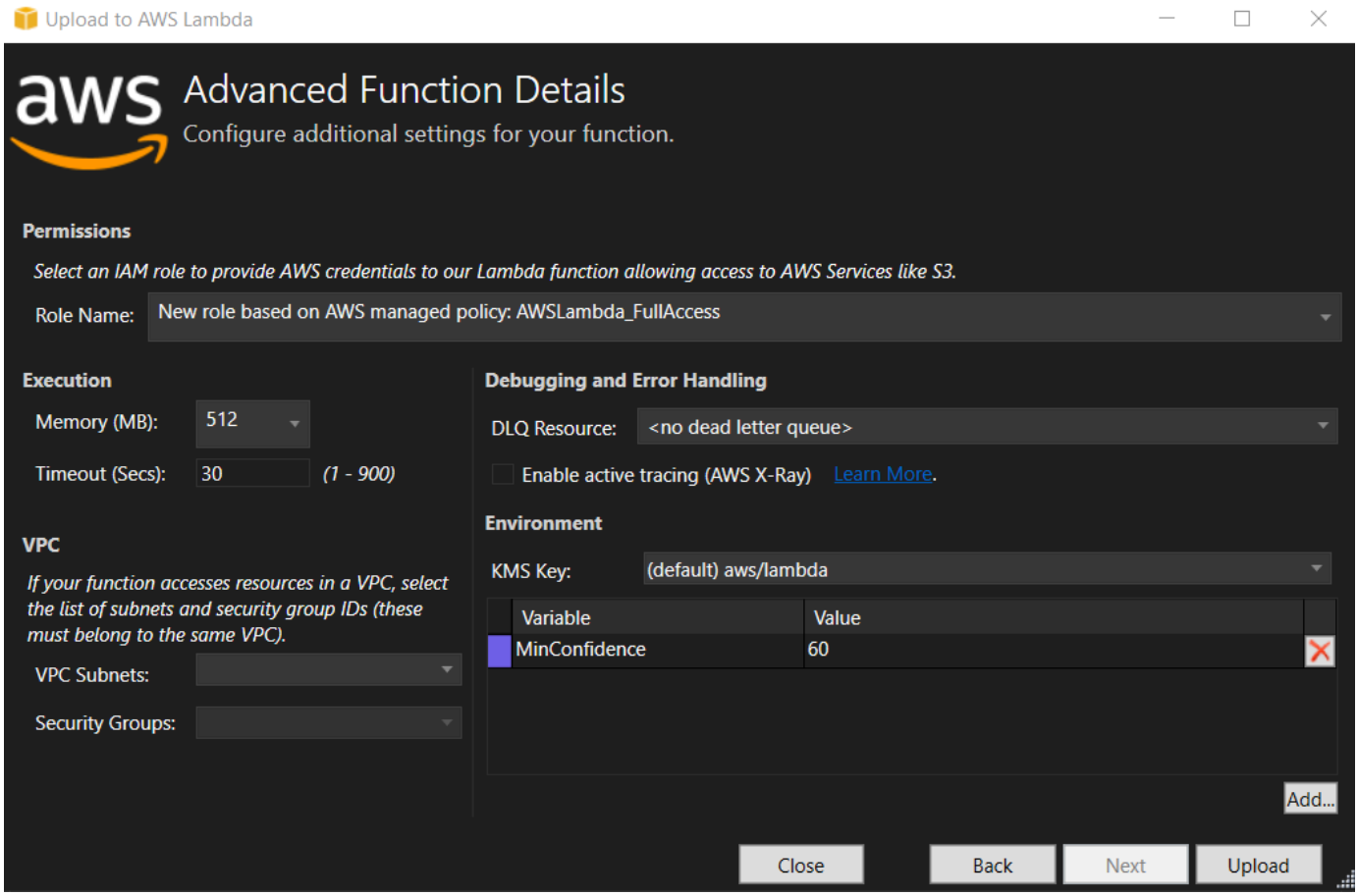
   > **Note**
   >
   > 此範例使用函數名稱**`ImageRekognition`**。



3. 在「進階功能詳細資料」視窗中，選取一個 IAM 角色，以授予程式碼存取 Amazon S3 和 Amazon Rekognition 資源的權限。

> **Note**
>
> 如果您正在遵循此範例，請選取`AWSLambda_FullAccess`角色。

4. 將環境變數設定`MinConfidence`為 60，然後選擇上傳以啟動部署程序。當「函數」檢視顯示在AWS 檔案總管中時，就會完成發佈程序。



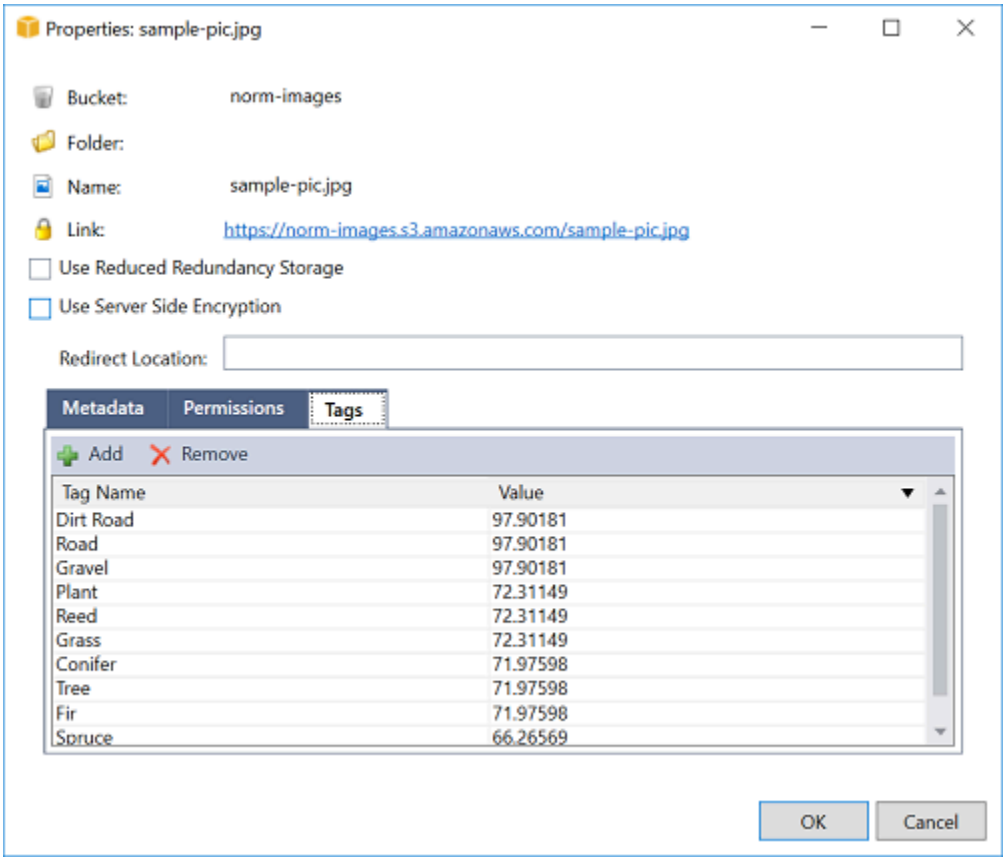
5. 成功部署後，透過導覽至「事件來源」索引標籤，設定 Amazon S3 將其事件傳送到新功能。
6. 在「事件來源」索引標籤中，選擇「新增」按鈕，然後選取要與 Lambda 函數連接的 Amazon S3 儲存貯體。

> **Note**
>
> 值區必須與 Lambda 函數位於相同的 AWS 區域。

測試 函數

現在已部署該函數，並將 S3 儲存貯體設定為其事件來源，請從 AWS Explorer 為您選取的儲存貯體開啟 S3 儲存貯體瀏覽器。然後上傳一些圖片。

上傳完成後，您可以通過查看函數視圖中的日誌來確認您的功能是否運行。或者，在值區瀏覽器中的影像上按一下滑鼠右鍵，然後選擇「 在「標籤」頁籤上，您可以檢視套用至物件的標籤。



教學課程：使用 Amazon 記錄架構與建 AWS Lambda 立應用程式日誌

您可以使用 Amazon CloudWatch 日誌來監控、存放和存取應用程式的日誌。若要將記錄資料取得至 CloudWatch 記錄檔，請使用 AWS SDK 或安裝 CloudWatch Logs 代理程式來監視某些記錄檔資料夾。 CloudWatch 日誌與幾個流行的 .NET 日誌框架集成，從而簡化了工作流程。

要開始使用 CloudWatch 日誌和 .NET 日誌框架，請將適當的 NuGet 包和 CloudWatch 日誌輸出源添加到您的應用程序，然後像平常一樣使用日誌庫。這使您的應用程序可以使用 .NET 框架記錄消息，將其發送到 CloudWatch 日誌，並在日誌控制台中顯示應用程序的 CloudWatch 日誌消息。您也可以根據應用程式的 CloudWatch 記錄訊息，從記錄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。
- Log4net：若要檢視，請參閱核心網路套件。
- ASP.NET 核心記錄架構：若要檢視，請參閱核心記錄架構套件。

以下是一個檔案範例，該NLog.config檔案範例會透過將AWS.Logger.NLog NuGet 套件和 AWS target 新增至來啟用 CloudWatch 記錄檔和主控台作為記錄訊息的輸出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>
```

記錄外掛程式全部建置在上方， AWS SDK for .NET 並在類似於 SDK 的程序中驗 AWS 證您的認證。下列範例詳細說明記錄外掛程式認證存 CloudWatch 取記錄所需的權限：

> **Note**
>
> AWS .NET 記錄外掛程式是一個開放原始碼專案。如需其他資訊、範例和指示，請參閱AWS 記錄 .NET GitHub 存放庫中的範例和指示主題。

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

```
    ]
}
```

部署到AWS

適用於 Visual Studio 的工具組支援將應用程式部署到AWS Elastic Beanstalk容器或AWS CloudFormation堆疊。

> **Note**
>
> 如果您使用的是快速版：
>
> - 您可以使用泊塢窗 CLI 將應用程式部署到 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 How in Visual Studio。對於 Amazon ECS 部署，您必須擁有泊塢視窗映像。如需詳細資訊，請參閱 Tools in Visual Studio。

主題

使用發佈至AWS位於 Visual Studio

發佈至AWS是互動式部署體驗，可協助您將 .NET 應用程式發佈至AWS部署目標，支援以 .NET Core 3.1 及更高版本為目標的應用程式。使用發佈至AWS將這些部署功能直接從 IDE 提供，讓您的工作流程保持在 Visual Studio 中：

- 只需單擊一下即可部署應用程序的能力。
- 根據您的應用程式提供部署建議。
- 自動建立 Docker 檔案，如您的部署目的地環境 (部署目標) 所需的相關性。
- 根據部署目標的要求，最佳化建置和封裝應用程式的設定。

> **Note**
>
> 如需有關發佈 .NET Framework 應用程式的其他資訊，請參閱指南在 Elastic Beanstalk 上建立和部署 .NET 應用程式
> 您也可以存取發佈至AWS來自. 如需詳細資訊，請參閱 。在上部署 .NET 應用程式AWS指南。

主題

先決條件

若要成功將 .NET 應用程式發佈到AWS服務中，將以下內容安裝到您的本地設備上：

- . NET 核心 3.1+（其中包括 .NET 和 .NET 6）：如需有關這些產品的其他資訊和下載資訊，請造訪微軟下載網站。
- Node.js 14.x 或更新版本：Node.js 是必需的才能運行AWS Cloud Development Kit (AWS CDK)。若要下載或取得有關 Node.js 的詳細資訊，請造訪Node.js 下載網站。

  > **Note**
  >
  > 發佈至AWS利用AWS CDK將您的應用程式及其所有部署基礎結構部署為單一專案。如需有關 的詳細資訊AWS CDK請參閱Cloud Development Kit指南。

- (選擇性) 在部署到以容器為基礎的服務 (例如 Amazon ECS) 時，會使用 Docker。如需更多詳細資訊和下載 Docker，請參閱Docker 下載網站。

支援的應用程式

在發佈到新的或現有的目標之前，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 ConStudio 應用程式
- 布拉索爾 WebAssembly 應用程式

發佈至AWS目標

發佈至新目標時，「發佈至」AWS將會透過提出建議和使用常用設定，引導您完成程序。如果您需要發佈到先前設定的目標，則會儲存您的偏好設定並進行調整，或者可立即用於單鍵部署。

發佈至新目標

以下說明如何設定您的發佈至AWS發佈至新目標時的部署偏好設定。

1. 來自AWS探險者中，展開登入資料下拉式功能表，然後選擇AWS與該區域對應的配置文件和AWS部署所需的服務。
2. 展開Region (區域)下拉式功能表，然後選擇AWS包含的區域AWS部署所需的服務。
3. 來自 Visual Studio方案 Explorer」窗格中開啟 (按一下滑鼠右鍵) 專案名稱的操作功能表，然後選擇發佈至AWS。這將開啟發佈至AWS。
4. 從發佈至AWS，選擇發佈至新目標以規劃新的部署。

> **Note**
>
> 若要修改預設部署認證，請選擇或按一下Edit (編輯)位於旁邊的鏈接登入資料區段，位於發佈至AWS。
> 若要略過目標組態處理作業，請選擇發佈至現有目標，然後從先前的部署目標清單中挑選您偏好的組態。

5. 來自發佈目標」窗格中，選擇一個AWS管理應用程式部署的服務。
6. 當您對您的設定感到滿意時，請選擇發布以開始部署程序。

> **Note**
>
> 啟動部署之後，發佈至AWS會顯示下列狀態更新：
>
> - 在部署過程中，發佈至AWS顯示部署進度相關資訊。
> - 在部署程序之後，發佈至AWS指出部署成功或失敗。
> - 成功部署之後，資源面板提供有關所建立資源的其他資訊。此資訊會依據應用程式類型和部署設定而有不同。

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應用程式AWS目標。

1. 來自AWS探險者中，展開登入資料下拉式功能表，然後選擇AWS與該區域對應的配置文件和AWS部署所需的服務。
2. 展開Region (區域)下拉式功能表，然後選擇AWS包含的區域AWS部署所需的服務。
3. 來自 Visual Studio方案 Explorer窗格中，請在專案名稱上按一下滑鼠右鍵，然後發佈至AWS開啟發佈至AWS。
4. 從發佈至AWS，選擇發佈至現有目標，從現有目標的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到AWS雲，這些應用程序顯示在發布到AWS。

5. 選取您欲部署應用程式的發佈目標，然後按一下發布以開始部署程序。

部署AWS Lambda使用 .NET Core CLI 進行專案

所以此AWS Toolkit for Visual Studio包括AWS Lambda.NET Core 專案範本。您可以使用 .NET Core 命令列界面 (CLI) 部署在 Visual Studio 中建立 Lambda 函數。

主題

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：

- 請確定 Visual Studio 2015 Update 3 已安裝。
- 安裝.NET Core 為 Windows。
- 設定 .NET Core CLI 與 Lambda 搭配使用 Lambda。如需詳細資訊，請參閱「」.NET Core CLI中的AWS Lambda開發人員指南。

- 安裝 TToolkit 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，請參閱什麼是AWSLambda中的AWS Lambda開發人員指南。
- 如需在視覺工作室中建立 Lambda 函數的相關資訊，請參閱AWS Lambda。
- 如需有關 Microsoft NET Core 的詳細資訊，請參閱.NET Core在微軟的在線文檔中。

列出 Lambda 命令可透過 .NET Core CLI

若要列出可透過 .NET 核心 CLI 使用的 Lambda 命令，請執行下列動作。

1. 開啟命令提示字元視窗，並導覽到檔案夾，包含 Visual Studio .NET Core Lambda 專案。
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 核心 CLI 發佈 .NET 核心 Lambda 專案

下列指示假設您已建立AWS Lambda. NET 核心功能。

1. 開啟命令提示字元視窗，並導覽至檔案所在的資料夾，您的 Visual Studio Lambda 專案。
2. 輸入 `dotnet lambda deploy-function`。
3. 出現提示時，輸入要部署的函數名稱。它可以是新的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，訊息建立新 Lambda 函數會顯示。

```
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\AWSLamb
da1\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
```

如果您部署現有的功能，部署函數只會要求AWS區域。

```
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\AWSLamb
da1\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，就可以使用了。如需詳細資訊，請參閱「」如何使用 的範例AWSLambda。

Lambda 會為您自動監控 Lambda 函數，並透過 Amazon Amazon 報告指標 CloudWatch。若要監控 Lambda 函數並進行疑難排解，請參閱故障診斷與監控AWSLambda 函數與 Amazon CloudWatch。

Elastic Beanstalk

AWS Elastic Beanstalk是簡化佈建程序的服務AWS適用於應用程式的資源。Elastic Beanstalk 提供所有功能AWS部署應用程式所需的基礎結構。此基礎架構包括：

- 為您的應用程式託管可執行檔和內容的 Amazon EC2 執行個體。
- Auto Scaling 群組可維持適當數目的 Amazon EC2 執行個體，以支援應用程式。
- Elastic Load Balancing 負載平衡器，可將傳入流量路由至具有最多頻寬的 Amazon EC2 執行個體。

Visual Studio 的工具組提供了一個精靈，可透過 Elastic Beanstalk 簡化發佈應用程式。下列各節會加以說明。

如需 Elastic Beanstalk 的詳細資訊，請前往Elastic Beanstalk 文檔。

主題

將傳統的 ASP.NET 應用程序部署到 Elastic Beanstalk

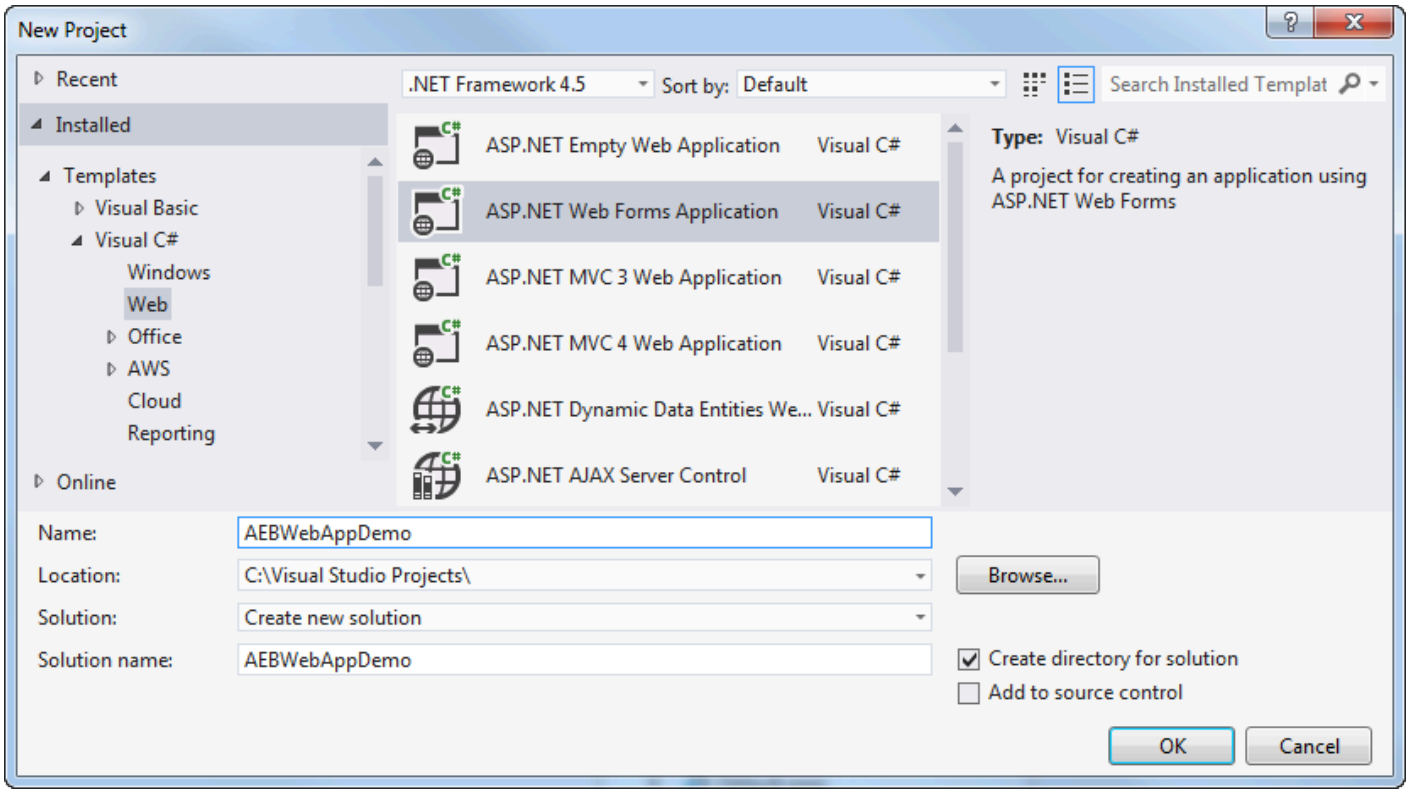
本節說明如何使用發佈至 Elastic Beanstalk 精靈 (做為 Toolkit for Visual Studio 的一部分所提供)，透過 Elastic Beanstalk 部署應用程式。若要練習，您可以使用 Visual Studio 內建的 Web 應用程式入門專案執行個體，也可以使用自己的專案。

> **Note**
>
> 此精靈也支援部署 ASP.NET 核心應用程式。如需 ASP.NET 核心的相關資訊，請參閱 AWS.NET 部署工具指南和更新的部署至AWS目錄。

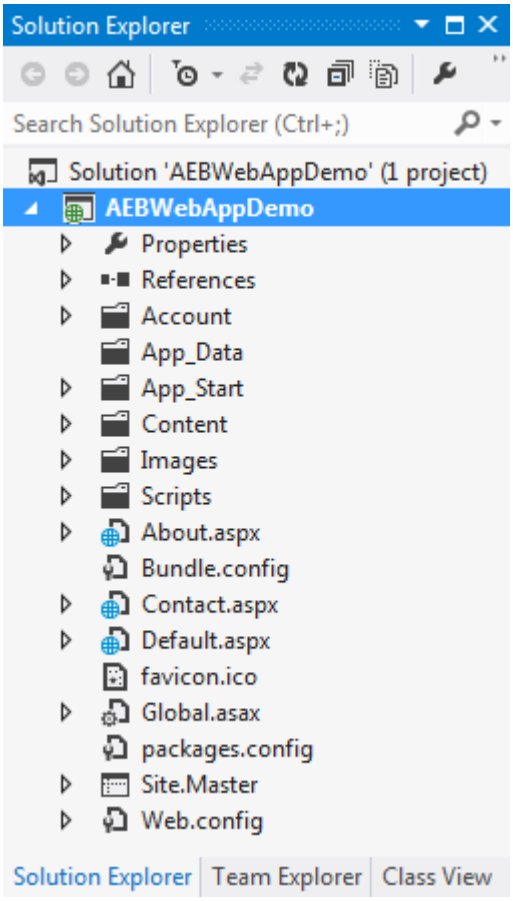
> **Note**
>
> 您必須先下載並安裝 Web 部署，才能使用「發佈成 Elastic Beanstalk」精靈。精靈依賴 Web 部署將 Web 應用程式和網站部署到網際網路資訊服務 (IIS) Web 伺服器。

建立範例 Web 應用程式入門專案

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，請選擇 ASP.NET Web 表單應用程式。

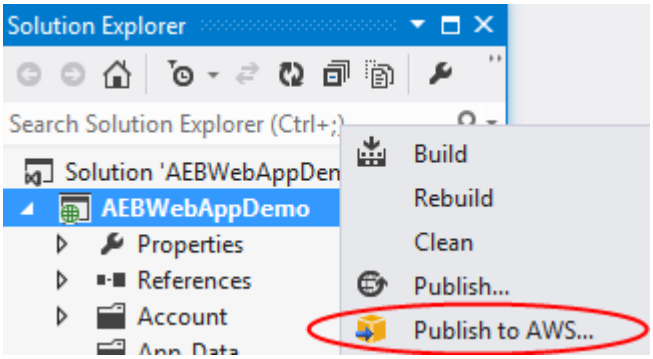


4. 在 Name (名稱) 方塊中，輸入 `AEBWebAppDemo`。
5. 在 [位置] 方塊中，輸入開發電腦上方案資料夾的路徑，或選擇 [瀏覽]，然後瀏覽並選擇解決方案資料夾，然後選擇 [選取資料夾]。
6. 確認已選取 Create directory for solution (為方案建立目錄) 方塊。在 [解決方案] 下拉式清單中，確認已選取 [建立新方案]，然後選擇 [確定]。視覺工作室將創建一個基於 ASP.NET Web 窗體應用程序項目模板的解決方案和項目。然後 Visual Studio 將顯示解決方案資源管理器，其中新的解決方案和項目出現。

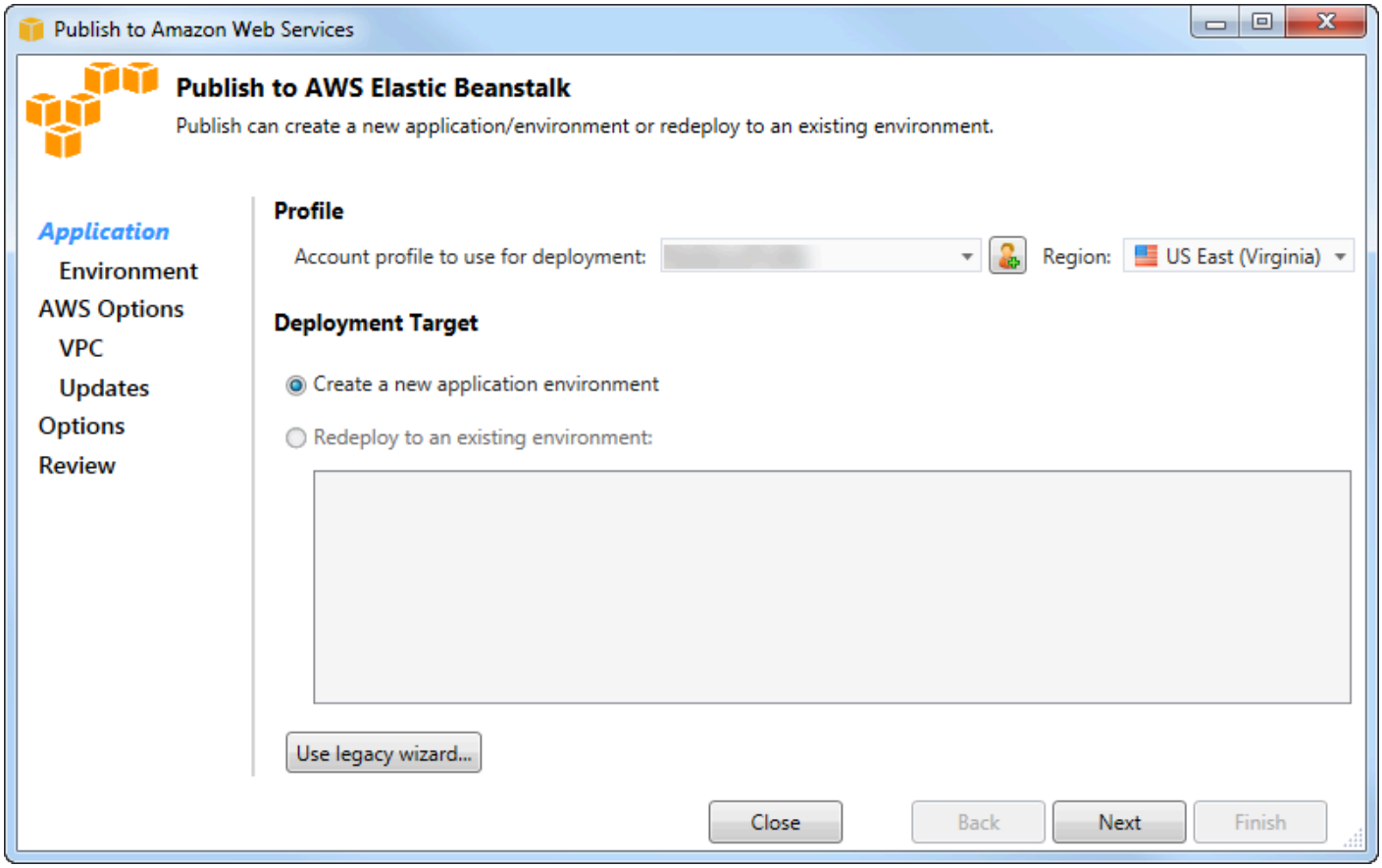


使用發佈至 Elastic Beanstalk 精靈部署應用程式

1. 在 [方案總管] 中，針對您在上一節中建立的專案開啟 AEBWebAppDemo 專案資料夾的內容 (按一下滑鼠右鍵) 功能表，或為您自己的應用程式開啟專案資料夾的內容功能表，然後選擇 [發佈至 EAWS lastic Beanstalk]。



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



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

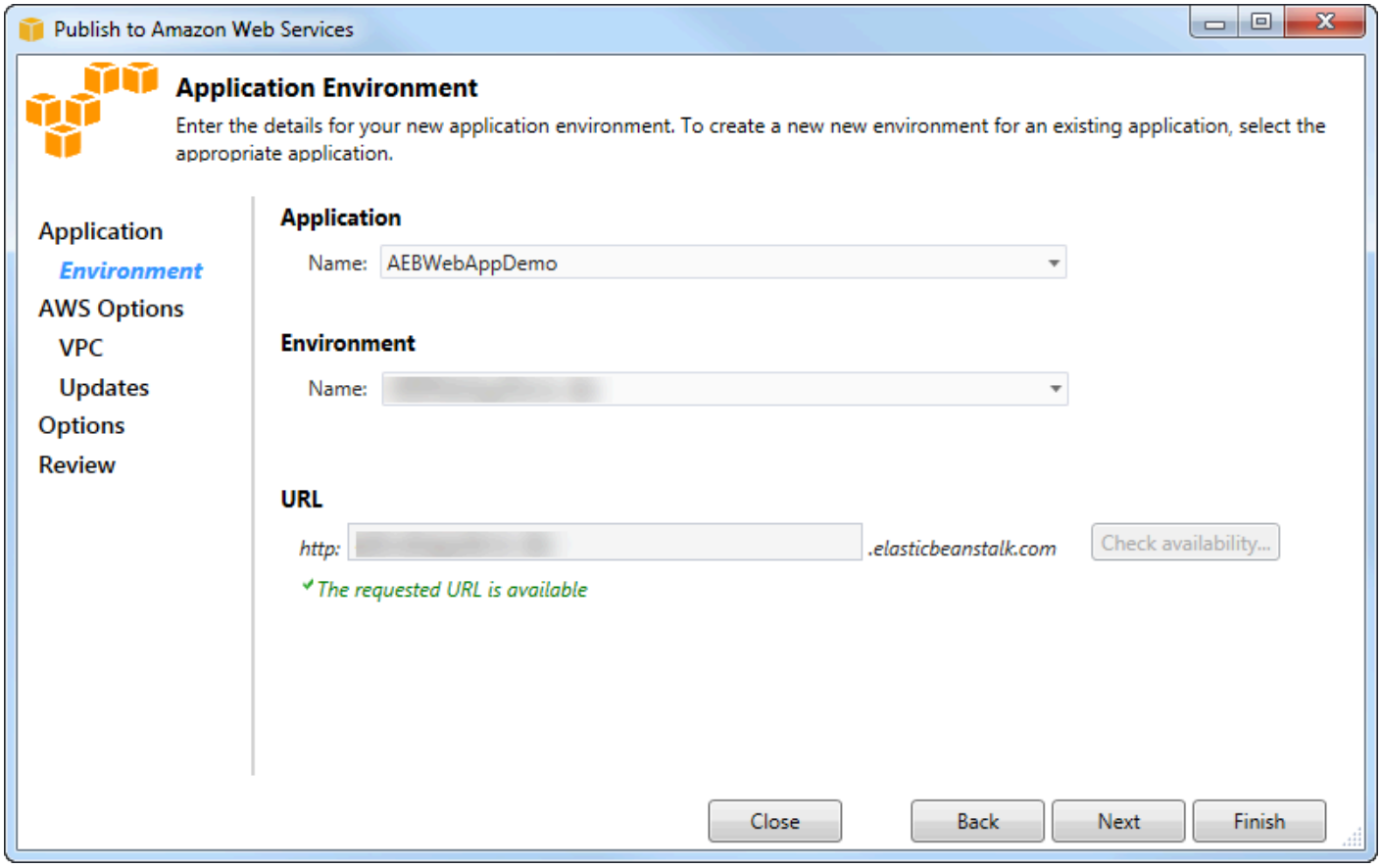
   或者，如果您有要使用的AWS帳戶，但尚未為其建立AWS帳戶設定檔，則可以選擇帶有加號 (+) 的按鈕來新增AWS帳戶設定檔。

3. 從「區域」下拉式清單中，選擇您要 Elastic Beanstalk 部署應用程式的目標區域。

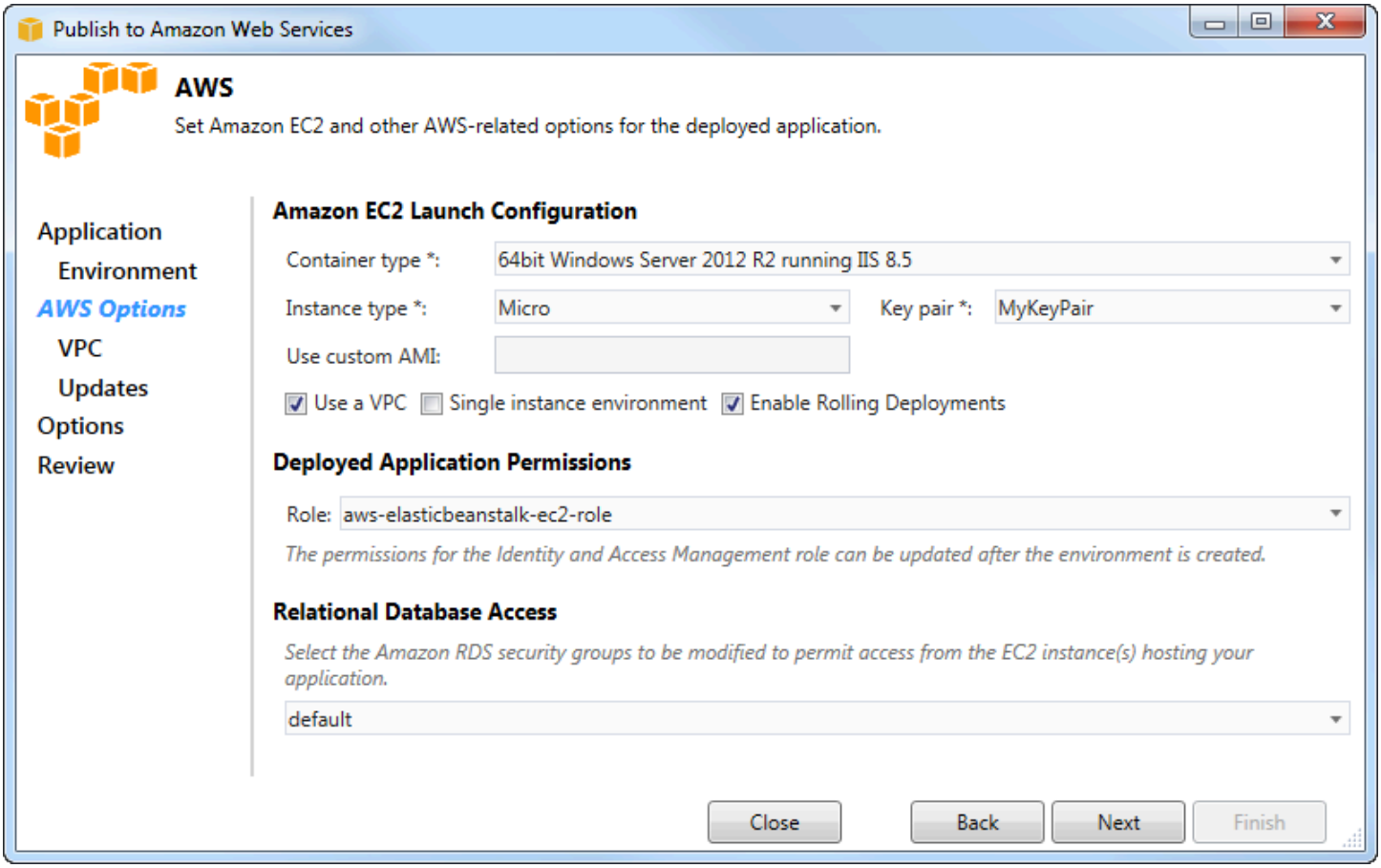
4. 在部署目標中，您可以選擇建立新的應用程式環境來執行應用程式的初始部署，或選擇重新部署至現有環境以重新部署先前部署的應用程式。(先前的部署可能是使用精靈或已取代的「獨立部署工具」來執行)。如果您選擇重新部署至現有環境，精靈可能會從目前正在執行的先前部署擷取資訊時發生延遲。

   > **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」，精靈會直接帶您進入「應用程式選項」頁面。如果您採用此路線，請略過本節稍後說明如何使用「應用程式選項」頁面的指示。

5. 選擇 Next (下一步)。



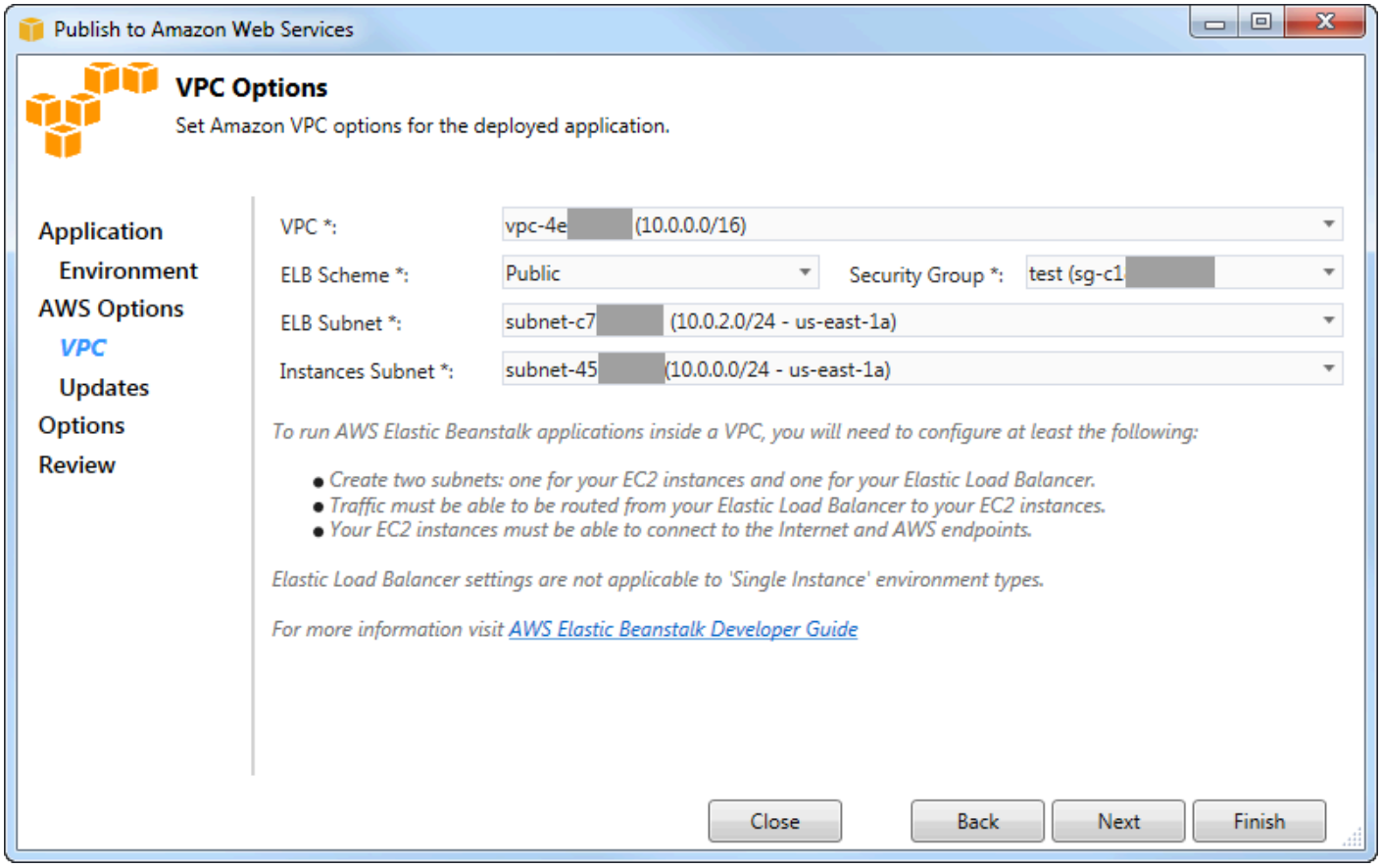
6. 在 [應用程式環境] 頁面的 [應用程式] 區域中，[名稱] 下拉式清單會建議應用程式的預設名稱。您可以從下拉式清單中選擇其他名稱來變更預設名稱。
7. 在「環境」區域的「名稱」下拉式清單中，輸入 Elastic Beanstalk 環境的名稱。在此內容中，「環境」一詞指的是您應用程式的基礎結構 Elastic Beanstalk 條款。此下拉式清單中可能已建議預設名稱。如果尚未建議使用預設名稱，您可以鍵入一個或從下拉式清單中選擇一個名稱 (如果有其他名稱)。環境名稱長度不可超過 23 個字元。
8. 在 URL 區域中，方塊會提出一個預設子網域，`.elasticbeanstalk.com`該子網域將會是您 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] 以確定 Web 應用程式的 URL 尚未在使用中。
10. 如果您的 Web 應用程式的 URL 可以使用，請選擇 [下一步]。



1. 在 [選AWS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您的應用程式的 Amazon 機器映像 (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。這將將執行個個體相關的成本降至最低。如需 Amazon EC2 成本的詳細資訊，請前往 EC2 定價頁面。
3. 在 key pair 下拉式清單中，選擇 Amazon EC2 執行個體金鑰配對，以登入將用於應用程式的執行個體。
4. 或者，在 [使用自訂 AMI] 方塊中，您可以指定將覆寫 [容器類型] 下拉式清單中指定的 AMI 的自訂 AMI。如需有關如何建立自訂 AMI 的詳細資訊，請前往 EAWS lastic Beanstalk 開發人員指南中的使用自訂 AM I，並從 Amazon EC2 執行個體建立 AMI。
5. 您可以選擇性地選擇您想要在 VPC 中啟動您的執行個個個體，請選擇 Telete VPC (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。

如果您選取此核取方塊，Elastic Beanstalk 仍會建立 Auto Scaling 群組，但不會對其進行設定。如果您想要稍後配置「Auto Scaling」群組，可以使用AWS Management Console.

7. 或者，如果您要控制將應用程式部署至執行個體的條件，請選取啟用滾動式部署方塊。僅當您尚未選取「單一例證」環境方塊時，才可以選取此方塊。
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式就是使用 IAM 角色。在「已部署的應用程式許可」區域中，您可以選擇現有的 IAM 角色，也可以建立精靈將用來啟動環境的角色。使用的應用程式AWS SDK for .NET會在向AWS服務提出要求時，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 安全群組旁邊的方塊，以便 Amazon EC2 執行個體可以存取該資料庫。
10. 選擇 Next (下一步)。
    - 如果您選取 [使用 VPC]，將會顯示 [VPC 選項] 頁面。
    - 如果您選取了「啟用輪替部署」，但未選取「使用 VPC」，則會顯示「滾動式部署」頁面。請跳至本段落稍後說明如何使用「機動式建置」頁面的指示。
    - 如果未選取「使用 VPC」或「啟用輪替部署」，將會顯示「應用程式選項」頁面。請跳至本節稍後說明如何使用「應用程式選項」頁面的指示。
11. 如果您選取 [使用 VPC]，請在 [VPC 選項] 頁面上指定資訊，以將應用程式啟動到 VPC 中。



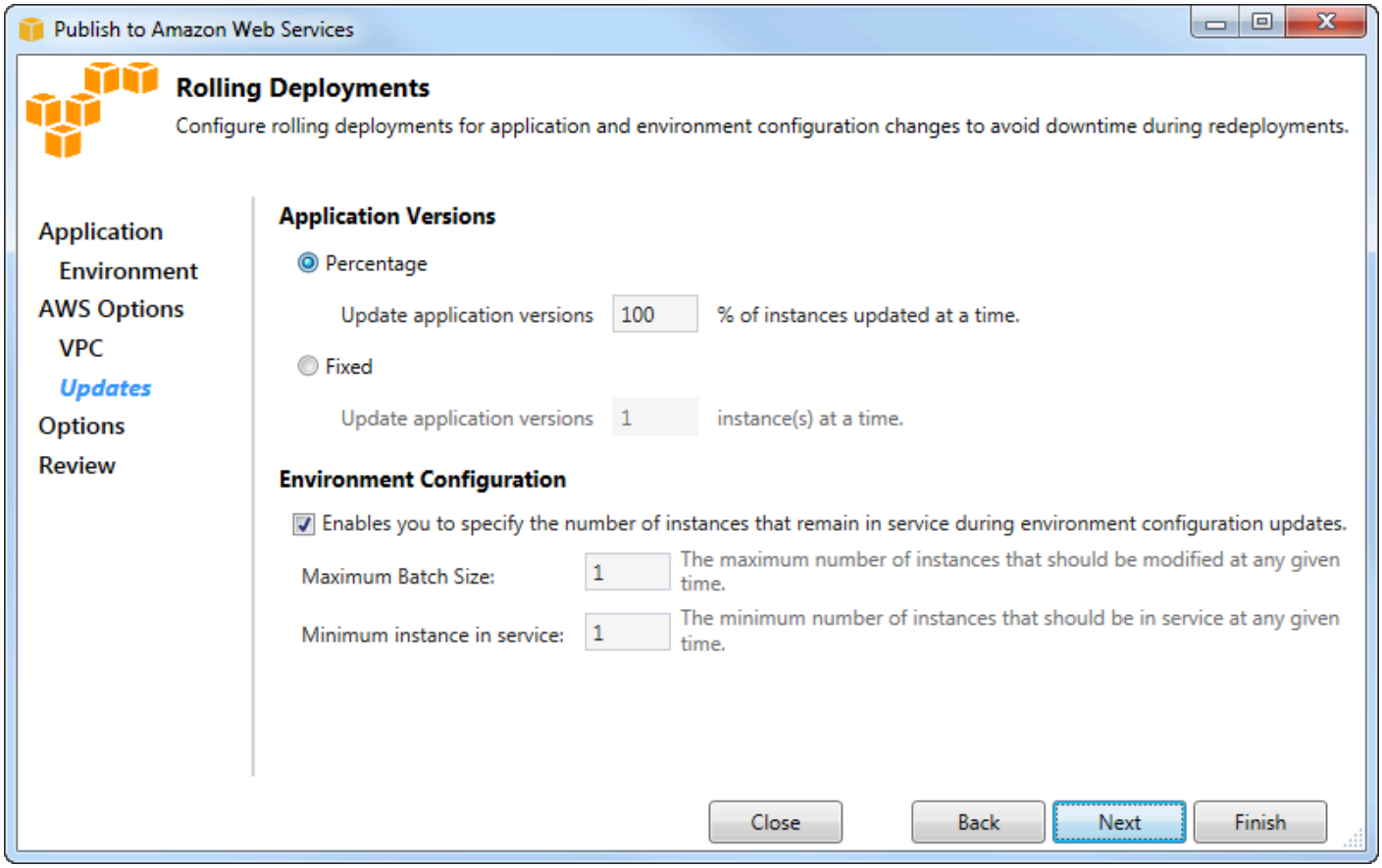
VPC 必須已建立了 VPC。如果您在 Visual Studio 的工具組中建立 VPC，Toolkit for Visual Studio 將會為您填入此頁面。如果您在AWS管理主控台中建立 VPC，請在此頁面中輸入 VPC 的相關資訊。

部署到 VPC 的關鍵注意事項

- 您的 VPC 需要至少一個公有私有私有私有 VPC 需要的私有 VPC
- 在 ELB 子網路下拉式清單中，指定公用子網路。Visual Studio 的工具組會為您的應用程式部署 Elastic Load Balancing 負載平衡器至公用子網路。公有有有有一個指向網際網路由閘道的路由資料表相關聯。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 Visual Studio 的工具組所建立的公用子網路具有可將其識別為公用的標記值。
- 在執行個體子網路下拉式清單中，指定私有子網路。適用 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。
- 應用程式的 Amazon EC2 執行個體透過執行網路位址轉譯 (NAT) 的公有子網路中的 Amazon EC2 執行個體，從私有子網路與網際網路通訊。若要啟用此通訊，您需要一個允許流量從私有子網路流向 NAT 執行個體的 VPC 安全群組。在 [安全性群組] 下拉式清單中指定此 VPC 安全性群組。

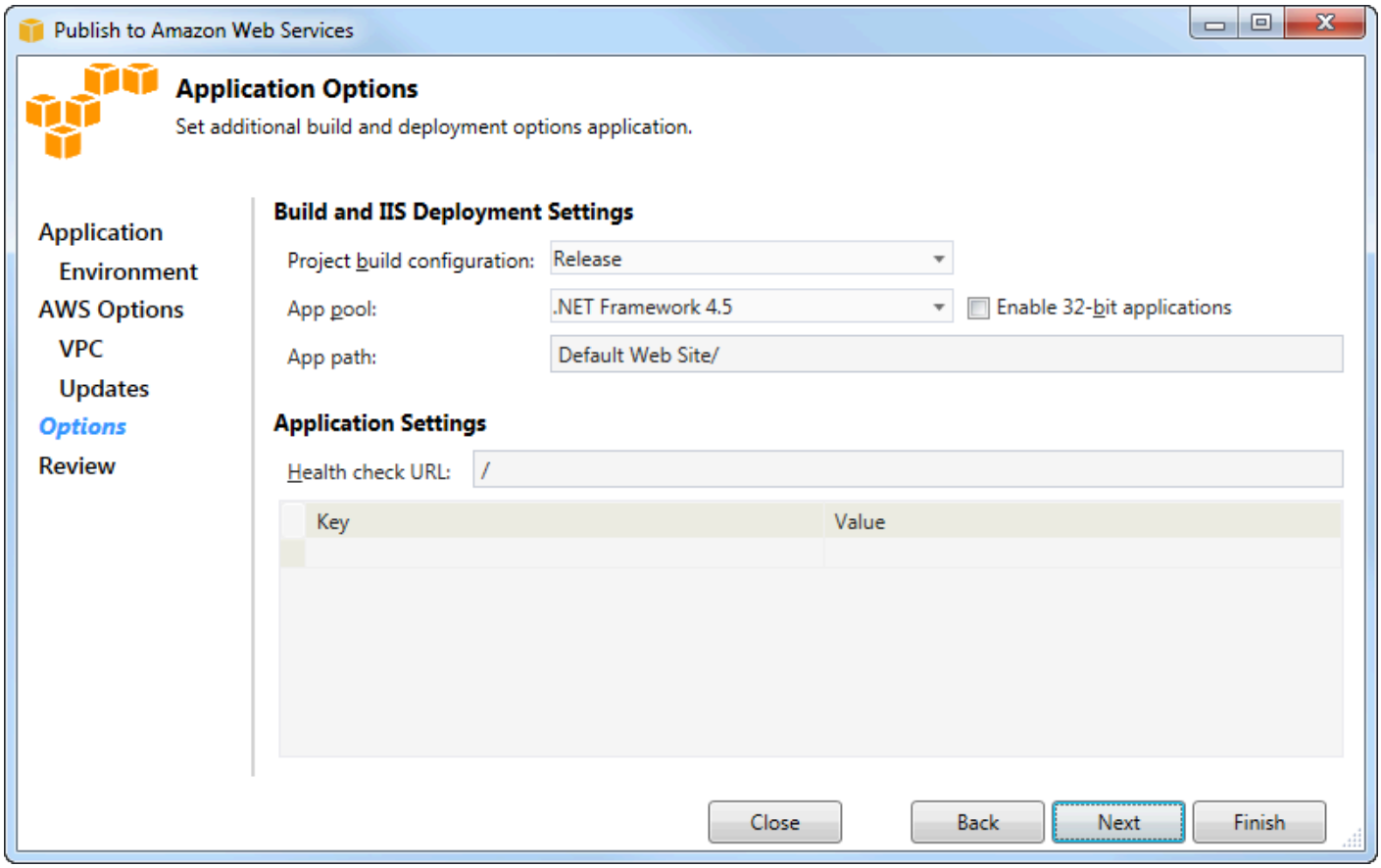
如需有關如何將彈性 Beanstalk 應用程式部署到 VPC 的詳細資訊，請前往 EAWS lastic Beanstalk 開發人員指南。

1. 在 [VPC 選項] 頁面上填寫完所有資訊之後，請選擇 [下一步]。
   - 如果您選取「啟用輪替部署」，將會顯示「輪替部署」頁面。
   - 如果未選取「啟用滾動式部署」，則會顯示「應用程式選項」頁面。請跳至本節稍後說明如何使用「應用程式選項」頁面的指示。
2. 如果您選取「啟用輪替式部署」，您可以在「機動部署」頁面上指定資訊，以設定如何將應用程式的新版本建置到負載平衡環境中的執行處理。舉例來說，您可以將環境設定為一次變更兩個執行個個個個體的執行個個個個個體的環境，您可以將該環境設定成一次變更兩個執行個個個個體的執行個個 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，選擇一個選項，一次控制部署為某個百分比或數目的執行個體。指定所需的百分比或數字。
4. 或者，如果您要指定部署期間保留在服務中的執行個體數目，請選取 [環境組態] 區域中的核取方塊。如果選取此方塊，請指定一次應修改的執行處理數目上限、一次應保持使用中的執行處理數目下限，或同時指定兩者。
5. 選擇 Next (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。



7. 在 [組建和 IIS 部署設定] 區域的 [專案組建設定] 下拉式清單中，選擇目標組建組態。如果精靈可以找到它，否則會出現釋放，使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。應該已顯示了正確的 .NET 架構版本。

9. 如果您的應用程式是 32 位元，請選取啟用 32 位元應用程式方塊

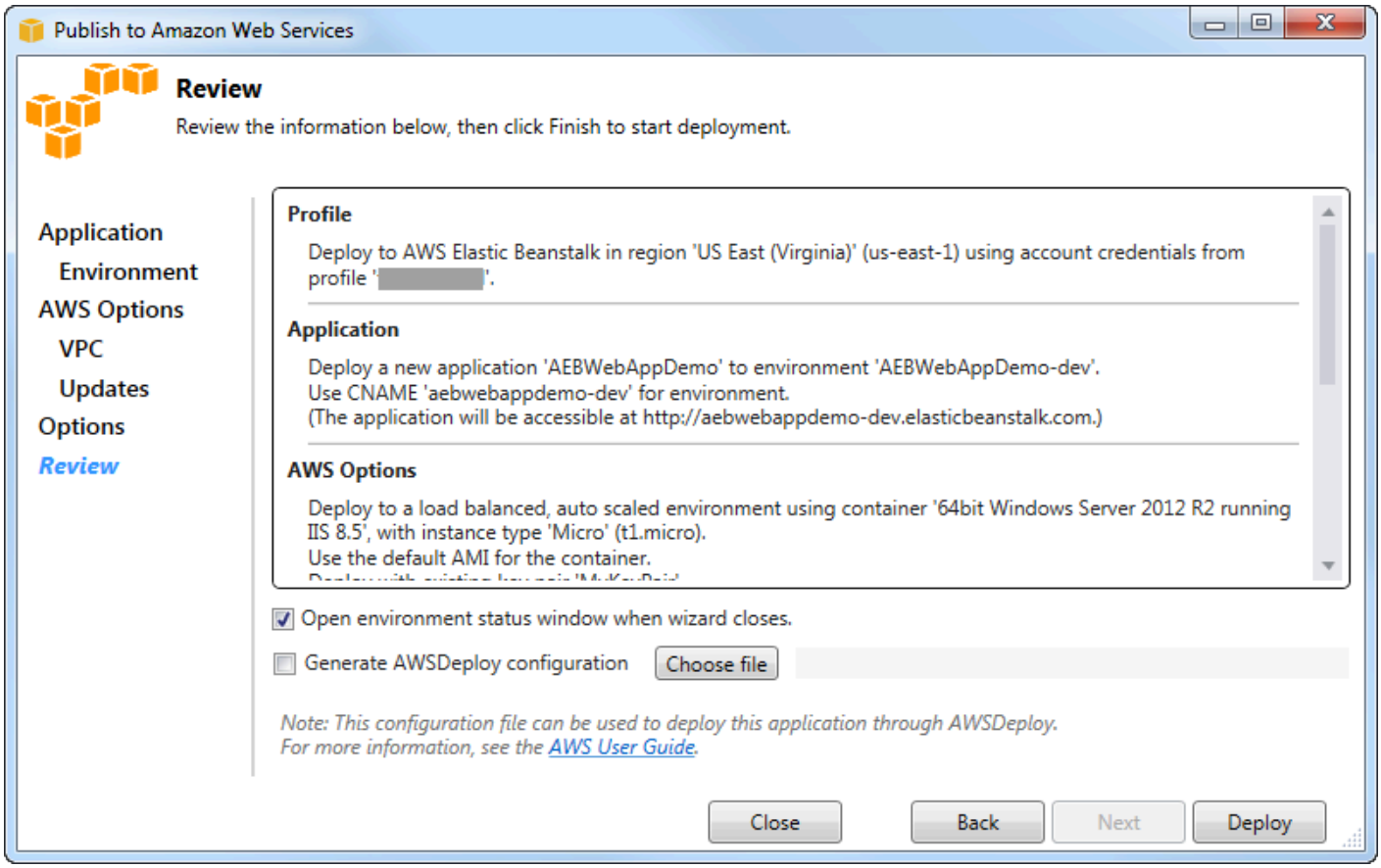
10. 在 [應用程式路徑] 方塊中，指定 IIS 將用於部署應用程式的路徑。依預設，會指定「預設網站/」，通常會轉譯為路徑`c:\inetpub\wwwroot`。如果您指定預設網站/以外的路徑，精靈會將重新導向放置在指向您指定的路徑的預設網站/路徑中。

11. 在 [應用程式設定] 區域的 [Health 全狀況] 核取方塊中，輸入 Elastic Beanstalk 的 URL 以檢查，以判斷您的 Web 應用程式是否仍有回應。此 URL 是相對於根伺服器 URL 的 URL。根伺服器 URL 預設為指定。例如，如果完整的 URL 是`example.com/site-is-up.html`，您可以輸入`/site-is-up.html`。

12. 在「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配對。

> **Note**
>
> 雖然不建議使用，但您可以使用「鍵」和「值」區域來指定應用程式執行的AWS認證。偏好的方法是在 [AWS選項] 頁面的 [Identity and Access Management 角色] 下拉式清單中指定 IAM 角色。但是，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。對重複這些步驟AWSSecretKey。

13選擇 Next (下一步)。



14.在 [複查] 頁面上，複查您設定的選項，然後選取 [精靈關閉時開啟環境狀態視窗] 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，作用中帳戶會對應用程式使用的AWS資源產生費用。

有關部署的資訊將顯示在 Visual Studio 狀態列和 [輸出] 視窗中。可能需要幾分鐘的時間。部署完成後，「輸出」視窗中會顯示確認訊息。

16若要刪除部署，請在AWS Explorer (檔案總管) 中，展開 Explorer Elastic Beanstalk 節點的內容 (右鍵) 功能表，開啟部署的子節點的內容 (右鍵) 功能表，開啟部署的子節點的內容 (刪除程序可能需要幾分鐘的時間才會完成。

將 ASP.NET Core 應用程式

> **Important**
>
> 本文件涉及舊版服務和功能。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是一項可簡化應用程式佈建AWS資源程序的服務。 AWS Elastic Beanstalk提供部署應用程式所需的所有AWS基礎結構。

該 Toolkit for Visual Studio 持部署 ASP.NET 核心應用程序AWS使用 Elastic Beanstalk。ASP.NET 核心是 ASP.NET 的重新設計與模組化的架構，可將相依性負荷降到最低，並簡化您的應用程式，以便在雲端中執行。

AWS Elastic Beanstalk可讓您輕鬆部署各種不同語言的應用程式至AWS. Elastic Beanstalk 支持傳統的 ASP.NET 應用程序和 ASP.NET 核心應用程序。本主題說明部署 ASP.NET 核心應用程式。

使用精靈

至 Elastic Beanstalk 部署 ASP.NET Core 應用程 Toolkit for Visual Studio

如果您在部署傳統 ASP 之前使用過該工具包。NET 應用程式中，您會發現 ASP.NET 核心的經驗非常相似。在下面的步驟中，我們將逐步介紹部署體驗。

如果您之前從未使用過該工具組，則在安裝工具組後需要做的第一件事是使用工具組註冊您的登AWS入資料。如需如何執行此動作的詳細資料，請參閱如何為 Visual Studio 的應用程式指定AWS安全性認證文件。

若要部署 ASP.NET Core Web 應用程式，請以滑鼠右鍵按一下 [方案總管] 中的專案，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」。部署精靈會產生包含應用程式版本和環境集合的應用程式。環境包含執行應用程式版本的實際AWS資源。您每次部署應用程式的新版本，並且精靈 您可以在 Elastic Beanstalk 組件中了解有關這些概念的更多信息。 。

接下來，為應用程序及其第一個環境設置名稱。每個環境都有一個與其關聯的唯一 CNAME，您可以在部署完成時用來存取應用程式。

下一頁「AWS選項」可讓您配置要使用的AWS資源類型。在此範例中，保留預設值 (金鑰配對區段除外)。金鑰配對可讓您擷取 Windows 管理員密碼，以便登入電腦。如果您尚未建立 key pair，您可能需要選取 [建立新 key pair]。

許可

「權限」頁面用於將AWS登入資料指派給執行應用程式的 EC2 執行個體。如果您的應用程式使用存取其他AWS服務，這一點很重要。AWS SDK for .NET如果您沒有使用應用程序中的任何其他服務，則可以將此頁面保留為默認值。

應用選項

[應用程式選項] 頁面上的詳細資料與部署傳統 ASP.NET 應用程式時指定的詳細資料不同。在這裡，您可以指定用於封裝應用程式的組建設定和架構，並指定應用程式的 IIS 資源路徑。

完成「應用程式選項」頁面後，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

應用程式封裝並上傳至之後AWS，您可以從 Visual Studio 中的AWS檔案總管開啟環境狀態檢視，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。一旦一切都完成，環境狀態就會移至健康狀態。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。

任何應用程式的第一次部署會比後續重新部署花費更長的時間，因為它會建立新的AWS資源。在開發過程中迭代應用程序時，您可以通過返回嚮導或在右鍵單擊項目時選擇「重新發布」選項來快速重新部署。

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

如何指定AWS應用程式的安全登入資料

所以此AWS您指定的帳戶發佈至 Elastic Beanstalk精靈是AWS精靈將用於部署至 Elastic Beanstalk 的帳戶。

雖然不建議，您可能還需要指定AWS您的應用程式將用來存取的帳戶認證AWS部署後的服務。偏好的方法是指定 IAM 角色。在 中發佈至 Elastic Beanstalk精靈，您是透過執行此操作Identity and Access Management 角色 drop-down list on the AWS選項(憑證已建立！) 頁面上的名稱有些許差異。-在舊版發佈到 Amazon Web Services精靈，您是透過執行此操作IAM 角色 drop-down list on the AWS選項(憑證已建立！) 頁面上的名稱有些許差異。

如果您必須使用AWS您可以指定帳戶登入資料而非 IAM 角色AWS使用下列其中一種方式，為您的申請提供帳戶登入資料：

- 參考對應的縱斷面AWS中的帳戶登入資料`appSettings`項目的元素`Web.config`file. (若要建立設定檔，請參閱設定AWS登入資料。) 以下範例指定設定檔名稱為的登入資料`myProfile`。

```
<appSettings>
  <!-- AWS CREDENTIALS -->
  <add key="AWSProfileName" value="myProfile"/>
</appSettings>
```

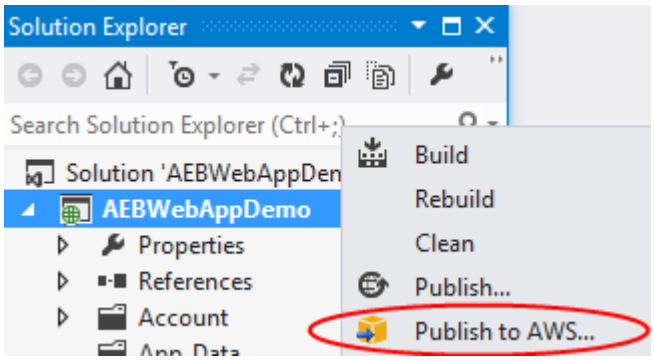
- 如果您使用的是發佈至 Elastic Beanstalk精靈，在應用程式選項頁面，在金鑰的列金鑰和數值區域，選擇AWSAccessKey。在 中數值列中，輸入存取金鑰。為您重複這些步驟AWSSecretKey。
- 如果您使用的是舊版發佈到 Amazon Web Services精靈，在應用程式選項頁面，在應用程式認證區域，選擇使用這些登入資料，然後輸入存取金鑰和私密存取金鑰存取金鑰和私密金鑰框。

如何將應用程式重新發佈到 Elastic Beanstalk 環境 (舊版)

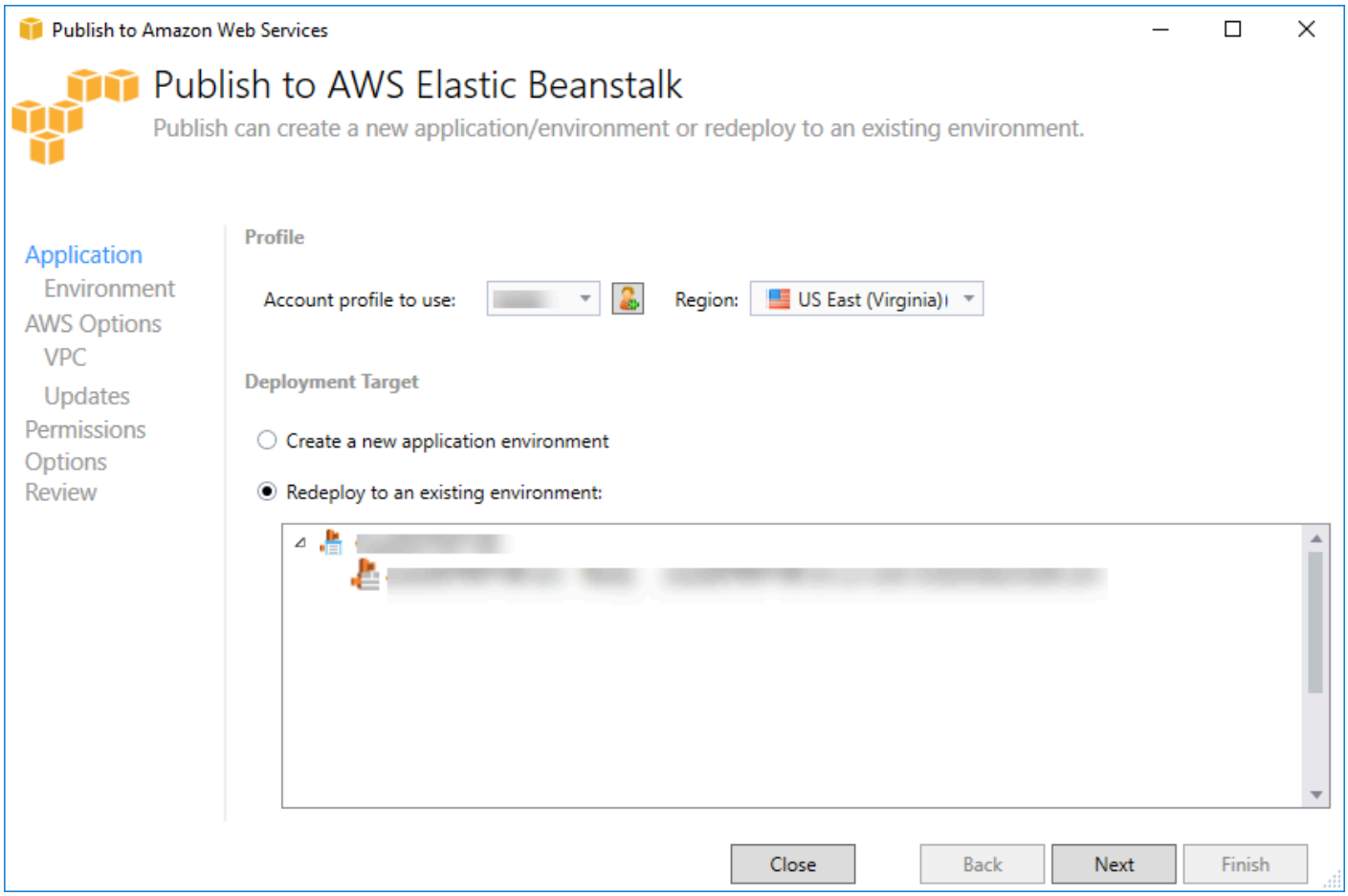
> **Important**
>
> 本文件涉及舊版服務和功能。如需更新的指南和內容，請參閱 AWS.NET 部署工具指南和更新的部署至AWS目錄。

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，在應用程式上重複執行。

1. 在 Solution Explorer 中，開啟 AEBWebAppDemo 專案資料夾的內容選單 (按一下滑鼠右鍵)，然後選擇 Publish in (發佈至)AWS Elastic Beanstalk。

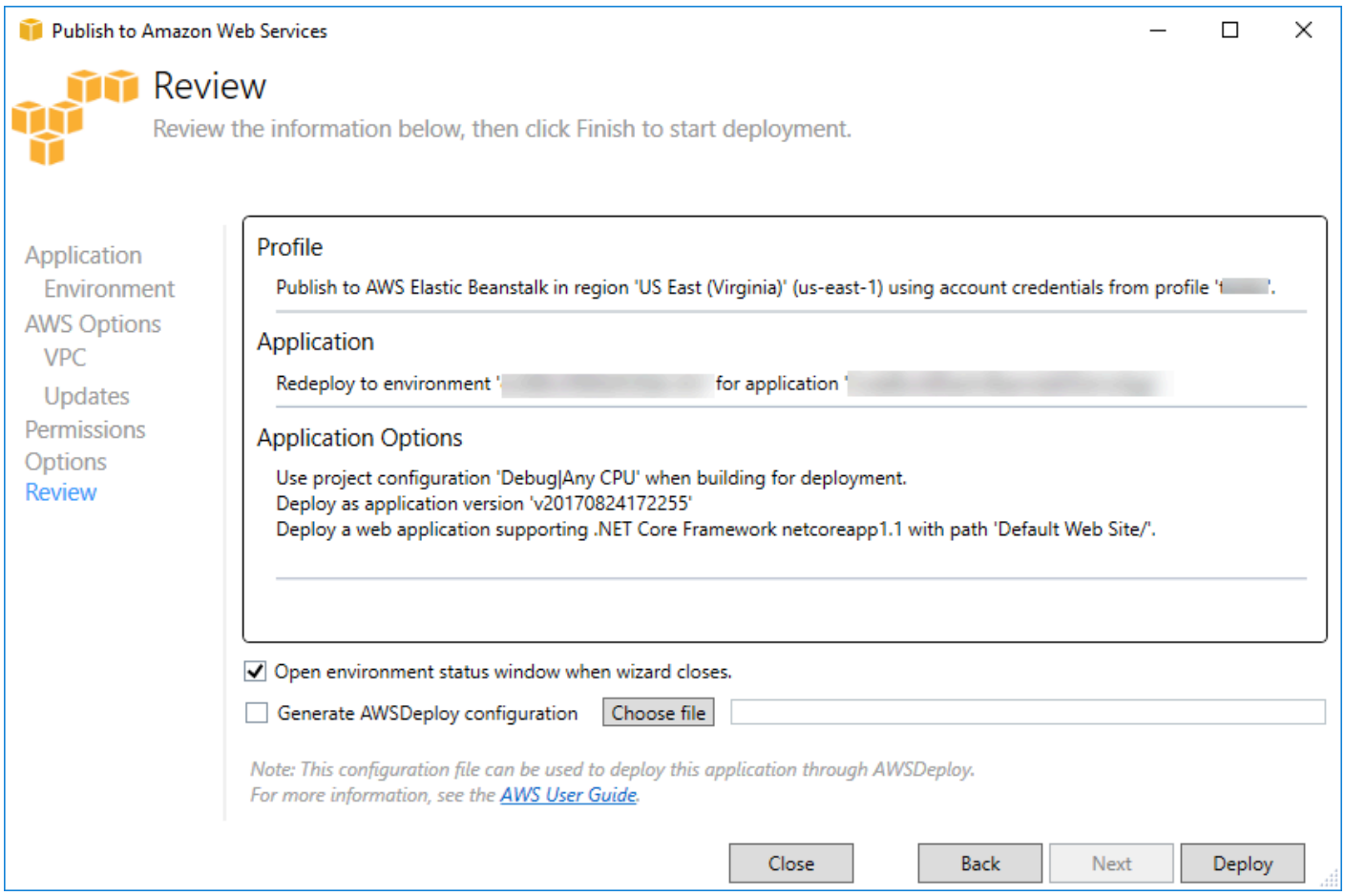


Publish to Elastic Beanstalk (發佈至 Elastic Beanstalk) 精靈隨即顯示。



2. 選取「重新部署至現有環境」，然後選擇您先前發佈的目標環境。按一下 Next (下一步)。

[檢閱] 精靈隨即出現。



3. 按一下部署。應用程式將重新部署至相同的環境。

如果您的應用程式正在啟動或終止程序，則無法重新發佈。

自訂 Elastic Beanstalk 應用程式部署

本主題介紹了彈性 Beanstalk 的 Microsoft Windows 容器的部署清單如何支持自定義應用程序部署。

自定義應用程序部署是一項強大的功能，對於那些希望利用 Elastic Beanstalk 的強大功能來創建和管理其AWS資源，但希望完全控制其應用程序的部署方式。對於自定義應用程序部署，您可以為 Elastic Beanstalk 執行的三個不同操作創建 Windows PowerShell 腳本。在啟動部署時使用安裝操作，在`RestartAppServer`API 可以從工具包或 Web 控制台調用，並在任何先前的部署中調用該 API，只要發生新部署，則會在任何先前的部署中調用該 API。

例如，您可能有一個要部署的 ASP.NET 應用程序，而您的文檔團隊編寫了他們希望包含在部署中的靜態網站。你可以通過像這樣編寫你的部署清單來做到這一點：

```
{
```

```
  "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"
          }
        }
      }
    ]
  }
}
```

為每個操作列出的腳本必須位於相對於部署清單文件的應用程序包中。在此示例中，應用程序包還將包含一個 documentation.zip 文件，其中包含由您的文檔團隊創建的靜態網站。

所以此`install.ps1`腳本提取 zip 文件並設置 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}
```

由於您的應用程序在 IIS 中運行，因此重新啟動操作將調用 IIS 重置。

```
iisreset /timeout:1
```

對於卸載腳本，清理安裝階段使用的所有設置和文件非常重要。這樣，在新版本的安裝階段，您可以避免與之前的部署發生任何衝突。對於此示例，您需要刪除靜態網站的 IIS 應用程序並刪除網站文件。

```
powershell.exe -Command {Remove-WebApplication -Name documentation}
Remove-Item -Recurse -Force 'c:\inetpub\wwwroot\documentation'
```

使用這些腳本文件和應用程序包中包含的 documentation.zip 文件，部署將創建 ASP.NET 應用程序，然後部署文檔站點。

在此示例中，我們選擇了一個簡單的示例來部署一個簡單的靜態網站，但使用自定義應用程序部署，您可以部署任何類型的應用程序，並讓 Elastic Beanstalk 管理AWS為它提供資源。

自訂 ASP.NET Core Elastic Elastic Beanstalk 部署

本主題介紹了使用 Elastic Beanstalk 和 Toolkit for Visual Studio 創建 ASP.NET 核心應用程序時部署的工作原理以及您可以執行哪些操作自定義部署。

在適用於 Visual Studio 的工具包中完成部署嚮導後，該工具包會捆綁應用程序並將其發送到 Elastic Beanstalk。創建應用程序包的第一步是使用新的 dotnet CLI 通過使用發布命令。框架和配置將從嚮導中的設置向下傳遞到發布命令。所以，如果你選擇發行版本為了`configuration`和網絡反射應用 1.0(針對)`framework`，則工具包將執行下列命令：

```
dotnet publish --configuration Release --framework netcoreapp1.0
```

當發布命令完成後，工具包會將新的部署清單寫入發佈文件夾中。部署清單是一個名為aws-windows-windows-deploys-deploys-windows-deploys-windows-deploys-windows-deploy，Elastic Beanstalk Windows 容器（版本 1.2 或更高版本）讀取該文件以確定如何部署應用程序。例如，對於要在 IIS 的根目錄部署的 ASP.NET 核心應用程序，該工具包會生成如下所示的清單文件：

```
{
  "manifestVersion": 1,
  "deployments": {

    "aspNetCoreWeb": [
      {
        "name": "app",
```

```
        "parameters": {
          "appBundle": ".",
          "iisPath": "/",
          "iisWebSite": "Default Web Site"
        }
      }
    ]
  }
}
```

所以此appBundle屬性指示應用程序位與清單文件相關的位置。此屬性可以指向目錄或 ZIP 歸檔文件。所以此iisPath和iisWebSite屬性指示 IIS 中託管應用程序的位置。

自訂資訊清單

只有在發佈文件夾中不存在清單文件時，工具包才會寫入清單文件。如果文件確實存在，工具包會更新appBundle、iisPath和iisWebSite屬性中列出的第一個應用程序中的aspNetCoreWeb部分。這可讓您添加aws-windows-windows-deploys-deploys-windows-deploys-windows-deploys-windows-deploy添加到您的項目中，並自定義清單。若要在 Visual Studio 中為 ASP.NET 核心 Web 應用程序執行此操作，將新的 JSON 文件添加到項目的根目錄並將其命名為aws-windows-windows-deploys-deploys-windows-deploys-windows-deploys-windows-deploy。

清單必須命名為aws-windows-windows-deploys-deploys-windows-deploys-windows-deploys-windows-deploy並且它必須位於項目的根部。Elastic Beanstalk 容器在根目錄中查找清單，如果發現它將調用部署工具。如果該文件不存在，則 Elastic Beanstalk 容器會回退到舊的部署工具中，該工具假定存檔是msDeploy存檔。

確保點網 CLIpublish命令包含清單，請更新project.json文件將清單文件包含在include在publishOptions。

```
{
   "publishOptions": {
     "include": [
       "wwwroot",
       "Views",
       "Areas/**/Views",
       "appsettings.json",
       "web.config",
       "aws-windows-deployment-manifest.json"
     ]
   }
```

```
  }
```

既然您已聲明清單以便將其包含在應用程序包中，您可以進一步配置要部署應用程序的方式。您可以在部署嚮導支持的範圍之外自定義部署。AWS已經定義了 JSON 模式aws-windows-windows-deploys-windows-deploys-windows-deploys-windows-deploys-windows-，並在安裝工具包的 Visual Studio 時，安裝程序將註冊該架構的 URL。

當您打開`windows-deployment-manifest.json`，您將看到在「架構」下拉框中選擇的架構 URL。您可以導航到 URL 以獲取清單中可以設置的內容的完整描述。選擇架構後，Visual Studio 將在編輯清單時提供 IntelliSense。

您可以執行的一個自訂操作是配置 IIS 應用程式集區，以便在其下運行應用程式。以下示例說明如何定義 IIS 應用程序池（「CustomPool」），該池每 60 分鐘回收一次流程，並使用`"appPool": "customPool"`。

```
{
  "manifestVersion": 1,
  "iisConfig": {
    "appPools": [
      {
        "name": "customPool",
        "recycling": {
          "regularTimeInterval": 60
        }
      }
    ]
  },
  "deployments": {
    "aspNetCoreWeb": [
      {
        "name": "app",
        "parameters": {
          "appPool": "customPool"
        }
      }
    ]
  }
}
```

此外，清單可以聲明 Windows PowerShell 腳本在安裝、重新啟動和卸載操作之前和之後運行。例如對於下面的清單運行`PostInstallSetup.ps1`在 ASP.NET 核心應用程序部署到 IIS 後執行進一步

的安裝工作。添加這樣 publishOptions 腳本時，請確保腳本已添加到`project.json`文件，就像你使用`aws-windows-deployment-manifest.json`file. 如果不這麼做，腳本不會作為 dotnet CLI 的一部分發布命令。

```
{
  "manifestVersion": 1,
  "deployments": {
    "aspNetCoreWeb": [
      {
        "name": "app",
        "scripts": {
          "postInstall": {
            "file": "SetupScripts/PostInstallSetup.ps1"
          }
        }
      }
    ]
  }
}
```

關於 .eb擴展名？

Elastic Beanstalk.ebextensions配置文件與所有其他 Elastic Beanstalk 容器一樣受支持。若要在 ASP.NET 核心應用程序中包含 .eb擴展名，請添加`.ebextensions`目錄到`include`部分下`publishOptions`中的`project.json`file. 有關 .eb擴發的更多信息，請查看Elastic Beanstalk 開發人員指南。

對 .NET 和 Elastic Beanstalk 的多應用 Support

使用部署清單，您可以將多個應用程序部署到同一 Elastic Beanstalk 環境中。

部署清單支持ASP.NET CoreWeb 應用程序以及傳統 ASP.NET 應用程序的 ms部署歸檔。想象一下，您已經使用 ASP.NET 核心作為前端編寫了一個新的驚人的應用程序，併為擴展 API 編寫了一個 Web API 項目。您還有一個使用傳統 ASP.NET 編寫的管理員應用程序。

工具包的部署嚮導側重於部署單個項目。要利用多個應用程序部署，您必須手動構建應用程序捆綁包。要開始，請編寫清單。對於此示例，您將在解決方案的根目錄中編寫清單。

清單中的部署部分有兩個子項：一個要部署的 ASP.NET 核心 Web 應用程序數組，以及要部署的 msdeploy 歸檔數組。對於每個應用程序，您可以設置 IIS 路徑以及相對於清單的應用程序位的位置。

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

在寫入清單後，您將使用 Windows PowerShell 創建應用程序包，並更新現有的 Elastic Beanstalk 環境以運行該應用程序包。編寫該腳本時假設它將從包含 Visual Studio 解決方案的文件夾中運行。

您需要在腳本中執行的第一件事是設置一個工作區文件夾，以便在其中創建應用程序包。

```
$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
}
```

一旦你創建了文件夾，就是時候準備好前端了。與部署嚮導一樣，使用 dotnet CLI 發佈應用程序。

```
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，子文件夾「前端」用於輸出文件夾，與您在清單中設置的文件夾匹配。現在您需要搭配 Web API 項目執行相同的作業。

```
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
```

管理站點是傳統的 ASP.NET 應用程序，因此您不能使用 dotnet CLI。對於管理應用程序，您應該使用 msbuild，傳入構建目標包來創建 msdeploy 歸檔文件。默認情況下，軟件包目標會在`obj\Release\Package`文件夾，因此您需要將歸檔文件複製到發佈工作區。

```
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
```

要告訴 Elastic Beanstalk 環境如何處理所有這些應用程序，請將清單從解決方案複製到發佈工作區，然後壓縮文件夾。

```
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序包，您可以轉到 Web 控制台並將檔案上傳到 Elastic Beanstalk 環境中。或者，您可以繼續使用AWSPowerShell cmdlet 使用應用程式包更新 Elastic Beanstalk 環境。確保您已

將當前配置文件和區域設置為包含您的 Elastic Beanstalk 環境的配置文件和區域，方法是使用`Set-AWSCredentials`和`Set-DefaultAWSRegion`Cmdlet。

```
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，使用工具包或 Web 控制台中的 Elastic Beanstalk 環境狀態頁檢查更新的狀態。完成後，您將能夠導航到部署清單中設置的 IIS 路徑部署的每個應用程序。

部署至 Amazon EC2 Container Service

> **Important**
>
> 新發佈至AWS功能旨在簡化將 .NET 應用程序發佈到AWS。系統可能會詢問您是否要切換到此發佈體驗，然後選擇發佈容器至AWS。如需詳細資訊，請參閱 使用發佈至AWS位於 Visual Studio。

Amazon Elastic Container Service 是一項可高度擴展、擁有高效能的容器管理服務，其不僅支援 Docker 容器，還能讓您在 Amazon EC2 實例的受管叢集上輕鬆執行應用程式。

若要在 Amazon Elastic Container Service 中部署應用程式，您的應用程式元件必須經過建構才能在 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。

適用 Toolkit for Visual Studio 提供了一個嚮導，可簡化通過 Amazon 彈性雲服務器發佈應用程序。下列各節會加以說明。

如需亞馬遜 ECS 的詳細資訊，請前往彈性容器服務文檔。它包括Docker 基本概念和建立叢集。

主題

- 指定AWSASP.NET Core 2 應用程式的登入資料
- 將 ASP.NET 核心 2.0 應用程式部署到亞馬遜 ECS (Fargate) (舊版)
- 將 ASP.NET Core 2.0 應用程式部署至亞馬遜 ECS (EC2)

指定AWSASP.NET Core 2 應用程式的登入資料

將應用程序部署到 Docker 容器時，有兩種類型的憑據正在播放：部署憑據和實例憑據。

發佈容器使用部署憑據來AWS嚮導在亞馬遜彈性雲服務器中創建環境。這包括諸如任務、服務、IAM 角色、Docker 容器存儲庫以及如果您選擇的話，還包括負載均衡器。

實例憑據（包括您的應用程序）用於訪問不同的AWS服務。例如，如果 ASP.NET Core 2.0 應用程序讀取和寫入 Amazon S3 對象，則需要適當的權限。您可以根據環境使用不同的方法提供不同的憑據。例如，您的 ASP.NET Core 2 應用程式可能會針對開發和生產環境。您可以使用本地 Docker 實例和證書進行開發，並在生產中使用定義的角色。

指定部署憑據

所以此AWS帳户中指定的發佈容器至AWS精靈是AWS帳户，嚮導將用於部署到亞馬遜雲服務器。賬户配置文件必須具有 Amazon 彈性計算雲、亞馬遜彈性容器服務和AWS Identity and Access Management。

如果您注意到下拉列表中缺少選項，則可能是因為您缺乏權限。例如，如果您為應用程序創建了集羣，但在發佈容器至AWS嚮導「羣集」頁面。如果發生這種情況，請添加缺少的權限，然後再次嘗試該嚮導。

指定開發實例憑據

對於非生產環境，您可以在應用程式設置中配置您的憑據。 <environment>.json 文件。例如，要在 Visual Studio 2017 中的應用程序設置開發 .JSON 文件中配置您的憑據，請執行以下操作：

1. 將 AWSSDK.Extens.NetCore.Setup NuGet 軟件包添加至您的項目。
2. AddAWS設置添加到應用程序設置。開發 .JSON。下面的配置設置`Profile`和`Region`。

```
{
    "AWS": {
```

```
        "Profile": "local-test-profile",
        "Region": "us-west-2"
    }
}
```

指定生產實例憑據

對於生產實例，我們建議您使用 IAM 角色來控制應用程序（和服務）可以訪問的內容。例如，要將 IAM 角色配置為亞馬遜雲服務器作為服務主體，具有亞馬遜簡單存儲服務和 Amazon DynamoDB 的權限，請訪問AWS Management Console：

1. 登入 AWS Management Console，並開啟位於 https://console.aws.amazon.com/iam/ 的 IAM 主控台。
2. 在 IAM 主控台的導覽窗格中，選擇 Roles (角色)，然後選擇 Create role (建立角色)。
3. 選擇AWSService (服務)角色類型，然後選擇EC2 Container Service。
4. 選擇EC2 容器服務任務使用案例。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 (下一步)：Permissions (許可)。
5. 選擇AmazonS3FullAccess和AmazonDynamoDBFullAccess許可政策。選中每個策略旁邊的複選框，然後選擇下一頁: 檢閱、
6. 適用於Role name (角色名稱)中，輸入角色名稱或角色名稱後綴，以協助您識別此角色的用途。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 Create role (建立角色)。

您可以將此角色用作任務角色在彈性雲服務器任務定義的頁面發佈容器至AWS精靈。

如需詳細資訊，請參閱「」使用基於服務的角色。

將 ASP.NET 核心 2.0 應用程式部署到亞馬遜 ECS (Fargate) (舊版)

> **Important**
>
> 本文件涉及舊版服務和功能。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。

本節說明如何使用將容器發佈到AWS精靈 (做為 Toolkit for Visual Studio 的一部分提供)，使用 Fargate 啟動類型，透過 Amazon ECS 部署容器化 ASP.NET 核心 2.0 應用程式以 Linux 為目標。由於 Web 應用程式目標是要持續執行，它將部署為服務。

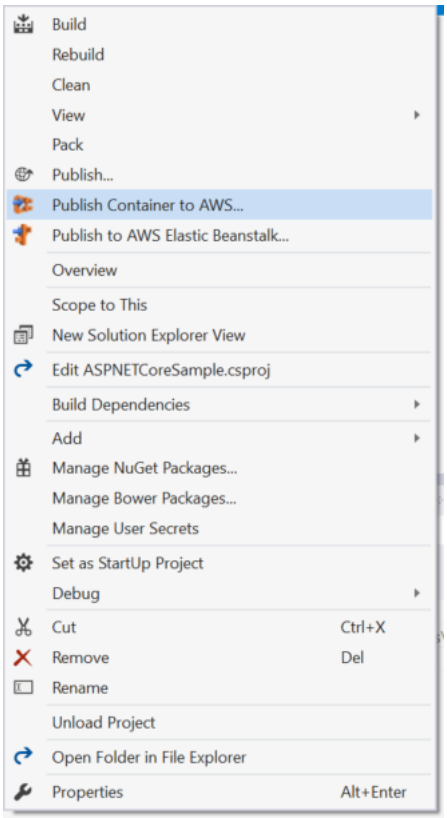
發佈容器之前

在使用 [發佈容器至AWS精靈] 部署 ASP.NET 核心 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。
- 安裝碼頭工人。您有幾個不同的安裝選項，包括 Windows 的泊塢視窗。
- 在視覺工作室中，為針對 Linux 的 ASP.NET 核心 2.0 容器化應用程式建立 (或開啟) 專案。

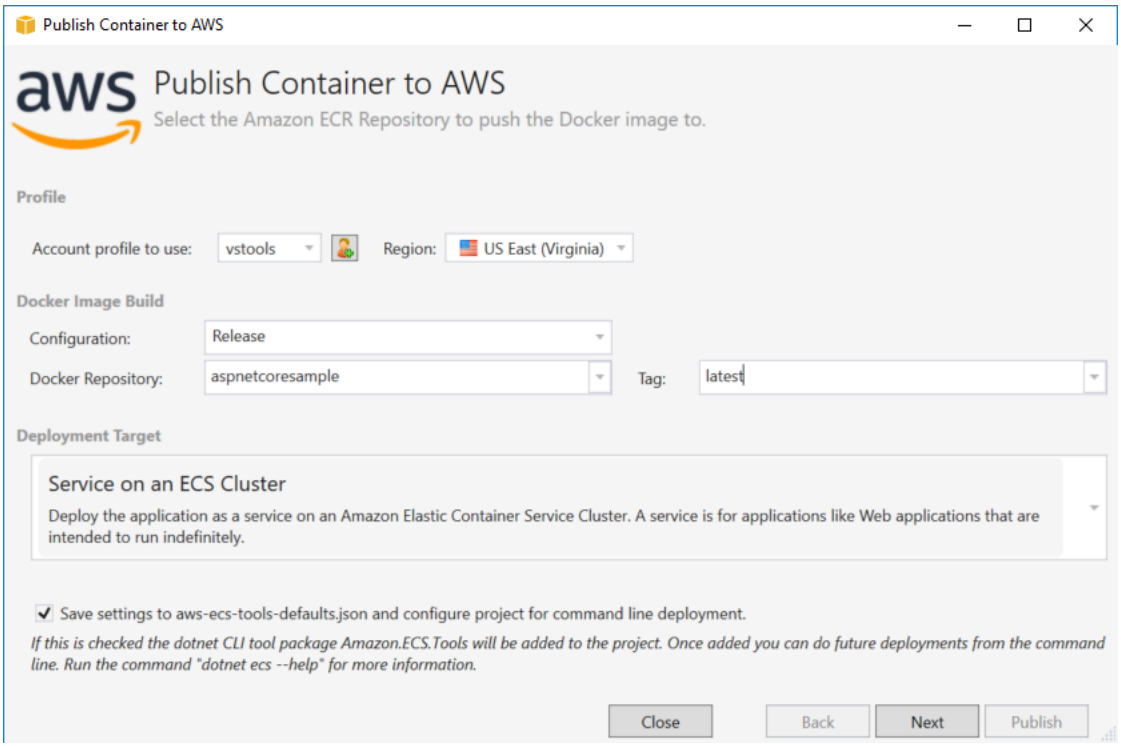
存取「發佈容器至AWS」精靈

若要部署以 Linux 為目標的 ASP.NET Core 2.0 容器化應用程式，請在方案總管中的專案上按一下滑鼠右鍵，然後選取 [將容器發佈至]AWS。



您也可以在 [建置] 功能表AWS上選取 [將容器發佈至]。

將容器發佈至AWS精靈



要使用的帳號設定檔-選取要使用的帳號設定檔。

區域-選擇部署區域。設定檔和區域可用來設定您的部署環境資源，以及選取預設的 Docker 登錄。

配置-選擇 Docker 映像構建配置。

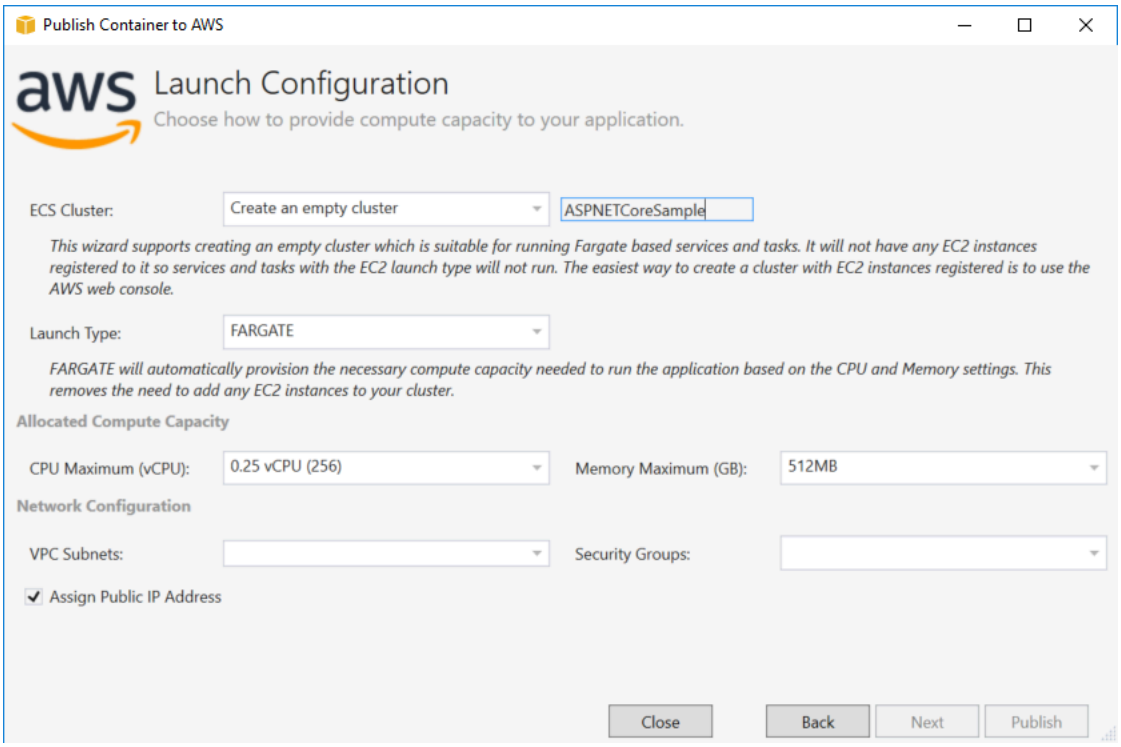
碼頭庫-選擇現有的 Docker 存儲庫或鍵入一個新的存儲庫的名稱，它將被創建。這是構建容器推送到的存儲庫。

標籤-選取現有標籤或輸入新標籤的名稱。標籤可以跟踪重要的詳細信息，例如 Docker 容器的版本，選項或其他唯一配置元素。

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 (例如 ASP.NET Web 應用程式) 時，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`從您的項目目錄中使用以部署和`dotnet ecs publish`容器。

啟動組態組態



ECS 叢集-挑選將執行 Docker 映像檔的叢集。如果您選擇建立空叢集，請提供新叢集的名稱。

啟動類型-選擇遠門。

CPU 最大值 (vCPU)-選擇應用程式所需的最大運算容量。若要查看允許的 CPU 和記憶體值範圍，請參閱工作大小。

記憶體上限 (GB)-選取應用程式可用的最大記憶體容量。

VPC 子網路-在單一 VPC 下選擇一或多個子網路。如果您選擇一個以上的子網路，您的工作將會分配到它們之間。這可以提高可用性。如需更多詳細資訊，請參閱預設 VPC 和預設子網路。

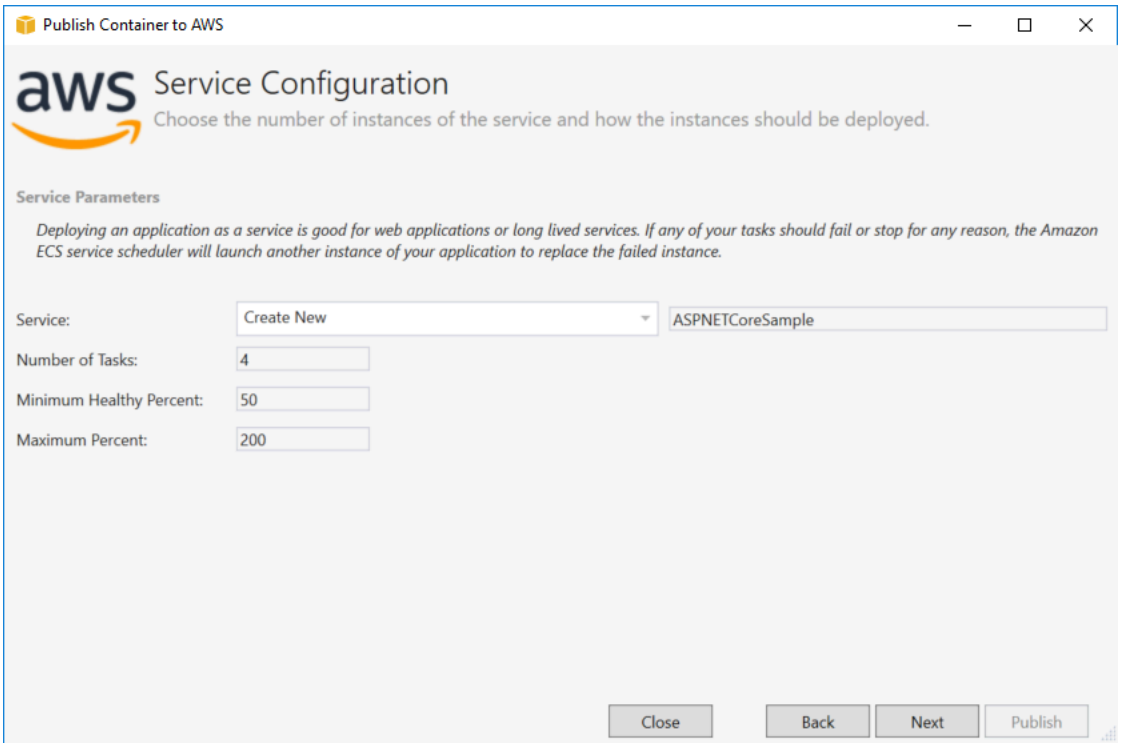
安全群組-選擇安全性群組。

安全群組就像是防火牆，用於關聯的 Amazon EC2 執行個體，用於關聯的 Amazon EC2 執行個體，用於關聯的 Amazon EC2 執行個體，用於

預設安全群組設定為允許來自指派給相同安全群組的執行個體和所有輸出 IPv4 流量的輸入流量。您需要允許輸出，以便服務可以連接到容器存放庫。

分配公共 IP 地址-選中此選項以使您的任務可從互聯網訪問。

服務組態組態組



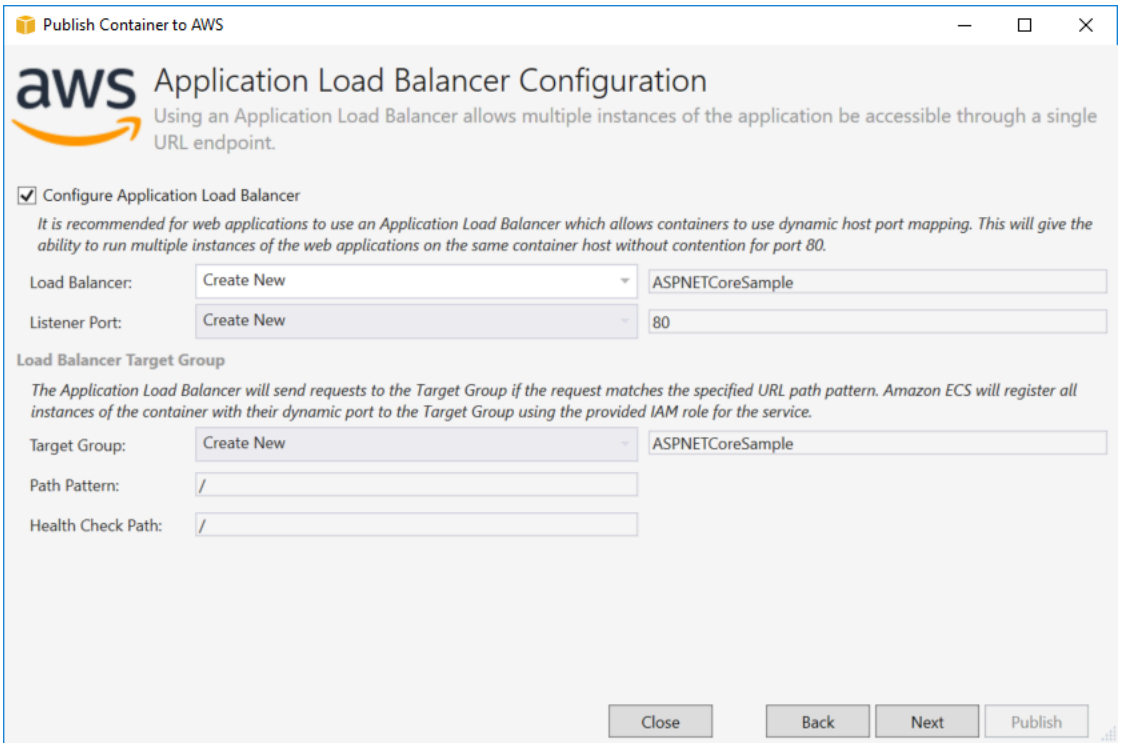
服務-在下拉式清單中選取其中一個服務，將容器部署至現有服務。或選擇「新建」以建立新服務。叢集中不得有相同的服務名稱，但一個區域內或多個區域間的多個叢集中可以有類似的服務名稱。

任務數量-要部署在您叢集上並保持執行的任務數量。每個任務都是容器的一個實例。

最小健全狀況百分比-部署期間必須保持在RUNNING狀態的工作百分比 (捨入至最接近的整數)。

最大百分比-部署期間，RUNNING或PENDING狀態中允許的工作百分比，四捨五入至最接近的整數。

Application Load Balancer 頁面



設定 Application Load Balancer-核取以設定應用程式負載平衡器。

Load Balancer-選取現有的負載平衡器，或選擇新建並輸入新負載平衡器的名稱。

監聽器連接埠-選取現有的監聽器連接埠，或選擇新建並輸入連接埠號碼。預設連接埠80適用於大多數 Web 應用程式。

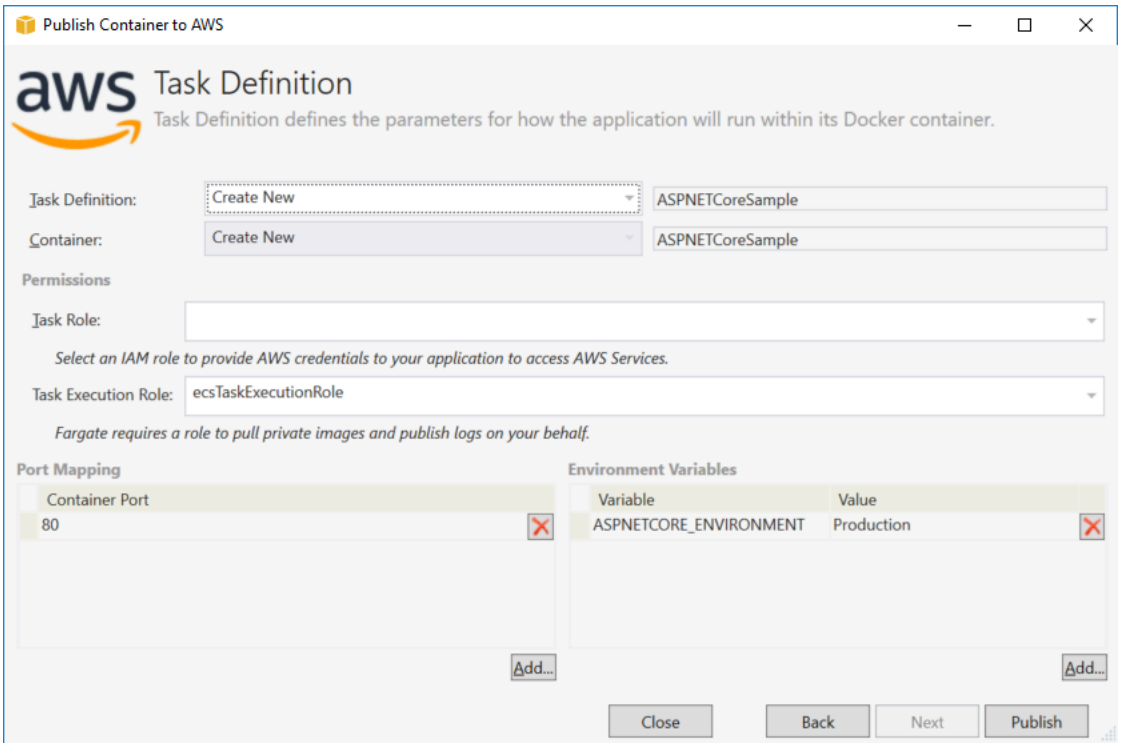
目標群組-選取 Amazon ECS 將向服務註冊任務的目標群組。

路徑模式-負載平衡器將使用基於路徑的路由。接受預設值/或提供不同的樣式。名稱模式區分大小寫，長度最多可達 128 個字元，而且包含選取的字元組。

Health 全狀況檢查路徑-執行狀態檢查之目標上的 Ping 路徑。在預設情況下，大小上限為 /。如果需要，請輸入其他路徑。如果您輸入的路徑無效，健全狀況檢查將會失敗，且會被視為健康狀況不良。

如果您部署多個服務，且每個服務都將部署到不同的路徑或位置，則需要自訂檢查路徑。

任務定義頁面頁



作業定義-選取現有作業定義，或選擇「新建」並輸入新作業定義名稱。

貨櫃-選取現有的貨櫃，或選擇「新建」，然後輸入新的貨櫃名稱。

任務角色-選取具有您的應用程式存取AWS服務所需登入資料的 IAM 角色。這是認證傳遞到您的應用程序的方式。請參閱如何為您的應用程式指定AWS安全認證。

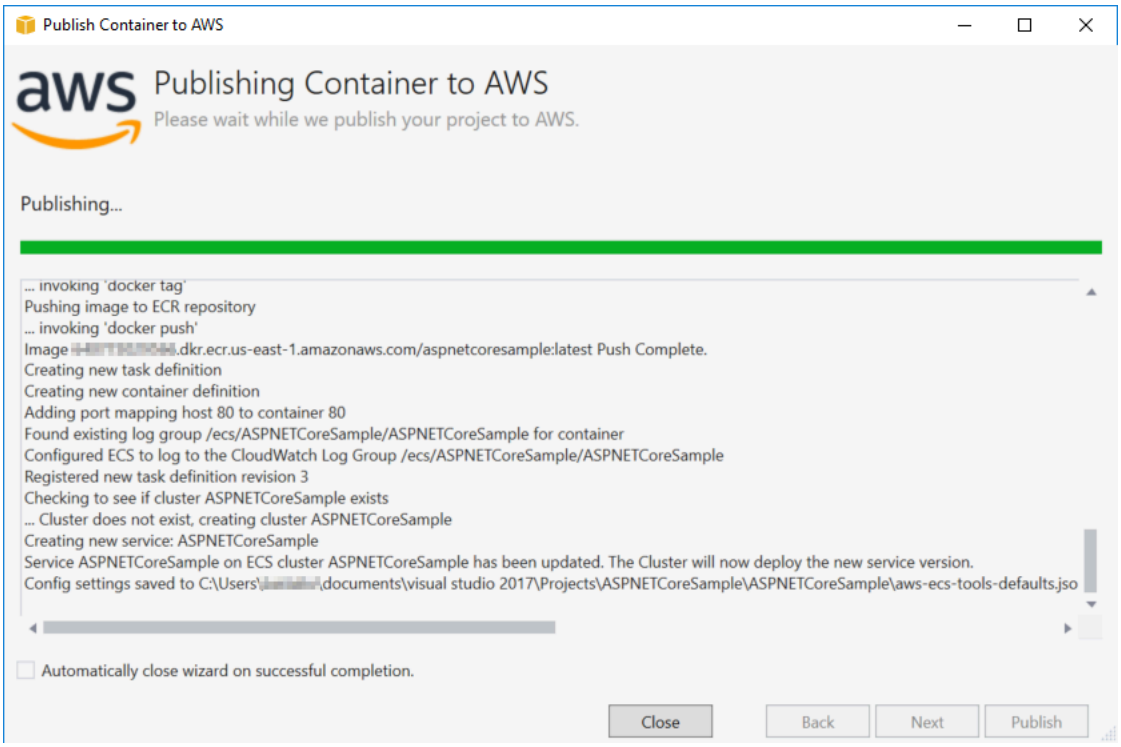
工作執行角色-選取具有提取私人映像和發佈記錄檔之權限的角色。 AWSFargate 會代表您使用這項操作。

連接埠對映-選擇容器上的連接埠號碼，該號碼繫結到自動指派的主機連接埠。

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。

當您滿意組態時，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檔案總管中找到新執行個體的 URL。展開 Amazon ECS 和叢集，然後按一下您的叢集。

將 ASP.NET Core 2.0 應用程式部署至亞馬遜 ECS (EC2)

本節將介紹如何使用發佈容器至AWS嚮導（作為適用 Toolkit for Visual Studio 的一部分提供），使用 EC2 啟動類型通過亞馬遜雲服務器部署面向 Linux 的容器化 ASP.NET 核心 2.0 應用程序。由於 Web 應用程式目標是要持續執行，它將部署為服務。

發佈容器之前

使用之前發佈容器至AWS部署 ASP.NET Core 2.0 應用程式：

- 指定您的AWS證書和使用 Amazon ECS 進行設定。
- 安裝 Docker。您有幾種不同的安裝選項，包括Docker (Windows)。
- 建立 Amazon ECS 叢集基於 Web 應用程式的需求。它只需要幾個步驟。
- 在可視工作室中，創建（或打開）針對 Linux 的 ASP.NET 核心 2.0 容器化應用程序的項目。

訪問發佈容器至AWS巫師

若要部署鎖定 Linux 的 ASP.NET Core 2.0 容器化應用程式，在 Solution Explorer (方案總管) 專案中按一下滑鼠右鍵，然後選擇發佈容器至AWS。

您也可以選擇發佈容器至AWS在可視工作室生成菜單上。

發佈容器至AWS精靈

要使用的帳户檔案-選擇要使用的帳户檔案。

Region (區域)-選擇部署區域。配置文件和區域用於設置部署環境資源並選擇默認的 Docker 註冊表。

組態-選擇 Docker 映像構建配置。

Docker 存儲庫-選擇一個現有的 Docker 存儲庫或鍵入新存儲庫的名稱，然後創建它。這是構建的容器映像被推送到的存儲庫。

標籤-選擇現有標籤或鍵入新標記的名稱。標籤可以跟蹤重要的細節，例如版本、選項或 Docker 容器的其他唯一配置元素。

部署-選取ECS 叢集上的服務。如果應用程序要長時間運行（如 ASP.NET Core 2.0 Web 應用程序），請使用此部署選項。

將設定保存至**aws-docker-tools-defaults.json**併為命令行部署配置項目-如果您希望從命令行靈活部署，請選中此選項。使用`dotnet ecs deploy`部署和`dotnet ecs publish`容器。

啟動組態頁面

ECS 叢集-選擇將運行 Docker 映像的集羣。您可以建立 ECS 叢集使用AWS管理主控台。

啟動類型-選擇 EC2 (EC2)。要使用 Fargate 啟動類型，請參見將 ASP.NET Core 2.0 應用程式部署至亞馬遜 ECS (Fargate)。

服務組態頁面

Service (服務)-在下拉菜單中選擇其中一個服務，將容器部署到現有服務中。或者選擇建立新的建立新服務。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-在您的叢集上部署和保持執行的任務數量。每個任務都是您的容器的執行個體。

最低健康百分比-必須保留的任務的百分比RUNNING狀態會無條件進入至最接近的整數。

最大百分比-任務的百分比在RUNNING或者PENDING狀態會無條件捨去至最接近的整數。

置放模板-選擇任務放置模板。

當您將任務啟動到叢集時，Amazon ECS 必須根據任務定義中所指定的需求 (例如 CPU 和記憶體) 來判斷要放置任務的位置。同樣地，當您縮減任務計數時，Amazon ECS 必須判斷要終止的任務。

放置模板控制如何將任務啟動到集羣中：

- 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 任務放置。

Application Load Balancer 頁面

配置 Application Load Balancer-選中此選項可以配置應用程式負載平衡器。

為服務選擇 IAM 角色-選擇一個現有角色或選擇建立新的並建立新角色。

負載平衡器-選擇現有負載平衡器或選擇建立新的，然後鍵入新負載均衡器的名稱。

接聽程式連接埠-選擇一個現有的監聽器端口，或選擇建立新的並鍵入連接埠號碼。默認端口80，適用於大多數 Web 應用程序。

目標群組-根據預設，負載平衡器會使用您針對目標組所指定的埠號和通訊協定，來將請求傳送到登錄的目標。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-負載平衡器將使用以路徑為基礎的路由。接受預設/或者提供不同的模式。路徑模式區分大小寫，長度最多可達 128 個字元，而且包含選擇字符集。

運作 Health 態檢查路徑-目標上運作狀態檢查目的地的 ping 路徑。默認情況下，它是/並且適用於 Web 應用程序。如果需要，請輸入其他路徑。如果輸入的路徑無效，運行狀況檢查將失敗，並且將被視為運行狀況不佳。

如果部署多個服務，並且每個服務都將部署到不同的路徑或位置，則可能需要自定義檢查路徑。

彈性雲服務器任務定義頁

任務定義-選擇現有任務定義，或選擇建立新的並鍵入新的任務定義名稱。

容器-選擇一個現有的容器或選擇建立新的，然後輸入新容器名稱。

內存 (MiB)-為軟性限制或者硬性限制或兩者皆是。

所以此軟性限制(MiB) 來保留給容器使用的記憶體。Docker 會嘗試將容器記憶體保持在軟性限制下。容器會佔用更多記憶體，最多達到以記憶體參數指定的硬性限制 (如適用)，或容器執行個體上的所有可用記憶體，以先達到者為準。

所以此硬性限制提供給容器使用的記憶體 (MiB)。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-為 IAM 角色選擇一個任務角色，該角色允許容器調用AWS代表您在相關聯政策中指定的API。這就是憑據傳遞到您的應用程序的方式。請參閱如何指定AWS應用程序的安全證書。

連接埠映射-新增、修改或刪除容器的連接埠映射。如果負載均衡器處於打開狀態，則主機端口默認為0，端口分配將是動態的。

環境變數-新增、修改或刪除容器的環境變量。

當您滿意時，請單擊發布開始部署過程。

發佈容器至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AWSExplorer。展開 Amazon ECS 和 Clusters，然後按一下您的叢集。

疑難排解 AWS Toolkit for Visual Studio

以下各節包含有關工具組中服務 AWS Toolkit for Visual Studio 及使用 AWS 服務的一般疑難排解資訊。

> **Note**
>
> 安裝與 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

疑難排解最佳做

以下是疑難排解 AWS Toolkit for Visual Studio 問題時建議的最佳作法。

- 在傳送報告之前，嘗試重新建立您的問題或錯誤。
- 在娛樂過程中，請詳細記錄每個步驟，設置和錯誤消息。
- 收集 AWS 工具包日誌。如需如何尋找 AWS Toolkit 記錄檔的詳細說明，請參閱本指南主題中的如何尋找 AWS 記錄程序。
- 在存放 AWS Toolkit for Visual Studio GitHub 庫的「問題」區段中檢查開啟的要求、已知的解決方案，或報告您未解決的AWS Toolkit for Visual Studio 問題。

如何找到您的 AWS 工具包日誌

1. 從視覺工作室主功能表中，展開擴充功能。
2. 選擇AWS 工具組以展開 [工 AWS 具組] 功能表，然後選擇 [檢視工具組記錄]。
3. 在作業系統中開啟 AWS 工具組記錄檔資料夾時，請依日期排序檔案，並找出任何包含與您目前問題相關資訊的記錄檔。

Amazon CodeWhisperer 登錄和註銷被禁用

如果您遇到停用 [登入] 和 [登出] 功能表項目的 CodeWhisperer 服務問題，請完成下列步驟以疑難排解問題。

1. 從 Windows 檔案總管，瀏覽至位於下列位置的 AWS 工具組快取資料夾：`%LOCALAPPDATA%/aws/toolkits/language-servers/CodeWhisperer`。
2. 清除快取資料夾的內容。
3. 關閉並重新開啟目前的解決方案。

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。安全是 AWS 與您之間共同承擔的責任。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。

雲的安全性 — AWS 負責保護運行 AWS 雲中提供的所有服務的基礎設施，並為您提供可以安全使用的服務。我們的安全責任是我們的首要任務 AWS，並且我們的安全性有效性是由第三方審計師定期測試和驗證，作為AWS 合規計劃的一部分。

雲端安全性 — 您的責任取決於您使用的 AWS 服務，以及其他因素，包括資料的敏感性、組織的需求，以及適用的法律和法規。

本 AWS 產品或服務透過其支援的特定 Amazon 網路服務 (AWS) 服務，遵循共同的責任模式。如需 AWS 服務安全性資訊，請參閱AWS 服務安全性說明文件頁面和符合性計劃 AWS 遵循工作範圍的 AWS 服務。

主題

- 資料保護 AWS Toolkit for Visual Studio
- 身分和存取權管理
- 本 AWS 產品或服務的合規驗證
- 本 AWS 產品或服務的復原能力
- 本 AWS 產品或服務的基礎架構安全性
- 中的組態和弱點分析 AWS Toolkit for Visual Studio

資料保護 AWS Toolkit for Visual Studio

AWS 共用責任模型適用於 Visual Studio 的 AWS 工具組中的資料保護。如此模型中所述， AWS 負責保護執行所有 AWS 雲端. 您負責維護在此基礎設施上託管內容的控制權。您也同時負責所使用 AWS 服務 的安全組態和管理任務。如需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 共同的責任模型和 GDPR 部落格文章。

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

- 每個帳戶均要使用多重要素驗證 (MFA)。
- 使用 SSL/TLS 與 AWS 資源進行通訊。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。
- 使用 AWS 加密解決方案，以及其中的所有默認安全控制 AWS 服務。
- 使用進階的受管安全服務 (例如 Amazon Macie)，協助探索和保護儲存在 Amazon S3 的敏感資料。
- 如果您在透過命令列介面或 API 存取時需要經 AWS 過 FIPS 140-2 驗證的加密模組，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) 140-2 概觀。

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。這包括當您使用適用於 Visual Studio 或其他 AWS 服務 使用主控台 AWS CLI、API 或 AWS SDK 的工具組時。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) 可協助系統管理員安全地控制 AWS 資源存取權。 AWS 服務 IAM 管理員控制哪些人可以通過身份驗證 (登入) 和授權 (具有權限) 來使用 AWS 資源。IAM 是您可以使用的 AWS 服務 ，無需額外付費。

主題

- 物件
- 使用身分驗證
- 使用政策管理存取權
- 如何 AWS 服務 使用 IAM
- 疑難排解 AWS 身分和存取

物件

您使用 AWS Identity and Access Management (IAM) 的方式會有所不同，具體取決於您在進行的工作 AWS。

服務使用者 — 如果您 AWS 服務 用於執行工作，則管理員會為您提供所需的認證和權限。當您使用更多 AWS 功能來完成工作時，您可能需要其他權限。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，請參閱疑難排解 AWS 身分和存取或 AWS 服務 您正在使用的使用指南。

服務管理員 — 如果您負責公司的 AWS 資源，您可能擁有完整的存取權 AWS。決定您的服務使用者應該存取哪些 AWS 功能和資源是您的工作。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搭配使用 IAM AWS，請參閱 AWS 服務 您正在使用的使用者指南。

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 AWS 基於身分識別的政策範例，請參閱 AWS 服務 您正在使用的使用者指南。

使用身分驗證

驗證是您 AWS 使用身分認證登入的方式。您必須以 IAM 使用者身分或假設 IAM 角色進行驗證 (登入 AWS)。 AWS 帳戶根使用者

您可以使用透過 AWS 身分識別來源提供的認證，以聯合身分識別身分登入。 AWS IAM Identity Center (IAM 身分中心) 使用者、貴公司的單一登入身分驗證，以及您的 Google 或 Facebook 登入資料都是聯合身分識別的範例。您以聯合身分登入時，您的管理員先前已設定使用 IAM 角色的聯合身分。當您使 AWS 用同盟存取時，您會間接擔任角色。

根據您的使用者類型，您可以登入 AWS Management Console 或 AWS 存取入口網站。如需有關登入的詳細資訊 AWS，請參閱《AWS 登入 使用指南》 AWS 帳戶中的如何登入您的。

如果您 AWS 以程式設計方式存取，請 AWS 提供軟體開發套件 (SDK) 和命令列介面 (CLI)，以使用您的認證以加密方式簽署要求。如果您不使用 AWS 工具，則必須自行簽署要求。如需使用建議的方法自行簽署請求的詳細資訊，請參閱 IAM 使用者指南中的簽署 AWS API 請求。

無論您使用何種身分驗證方法，您可能都需要提供額外的安全性資訊。例如， AWS 建議您使用多重要素驗證 (MFA) 來增加帳戶的安全性。如需更多資訊，請參閱 AWS IAM Identity Center 使用者指南中的多重要素驗證和 IAM 使用者指南中的在 AWS中使用多重要素驗證 (MFA)。

AWS 帳戶 根使用者

當您建立時 AWS 帳戶，您會從一個登入身分開始，該身分可完整存取該帳戶中的所有資源 AWS 服務和資源。此身分稱為 AWS 帳戶 root 使用者，可透過使用您用來建立帳戶的電子郵件地址和密碼登入來存取。強烈建議您不要以根使用者處理日常任務。保護您的根使用者憑證，並將其用來執行只能由根使用者執行的任務。如需這些任務的完整清單，了解需以根使用者登入的任務，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

最佳作法是要求人類使用者 (包括需要系統管理員存取權的使用者) 使用與身分識別提供者的同盟，才能使用臨時登入資料進行存取 AWS 服務 。

聯合身分識別是來自企業使用者目錄的使用者、Web 身分識別提供者、Identi ty Center 目錄，或使用透過身分識別來源提供的認證進行存取 AWS 服務 的任何使用者。 AWS Directory Service同盟身分存取時 AWS 帳戶，他們會假設角色，而角色則提供臨時認證。

對於集中式存取權管理，我們建議您使用 AWS IAM Identity Center。您可以在 IAM Identity Center 中建立使用者和群組，也可以連線並同步到自己身分識別來源中的一組使用者和群組，以便在所有應用程式 AWS 帳戶 和應用程式中使用。如需 IAM Identity Center 的相關資訊，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 使用者是您內部的身分，具 AWS 帳戶 有單一人員或應用程式的特定許可。建議您盡可能依賴暫時憑證，而不是擁有建立長期憑證 (例如密碼和存取金鑰) 的 IAM 使用者。但是如果特定使用案例需要擁有長期憑證的 IAM 使用者，建議您輪換存取金鑰。如需更多資訊，請參閱 IAM 使用者指南中的為需要長期憑證的使用案例定期輪換存取金鑰。

IAM 群組是一種指定 IAM 使用者集合的身分。您無法以群組身分簽署。您可以使用群組來一次為多名使用者指定許可。群組可讓管理大量使用者許可的程序變得更為容易。例如，您可以擁有一個名為 IAMAdmins 的群組，並給予該群組管理 IAM 資源的許可。

使用者與角色不同。使用者只會與單一人員或應用程式建立關聯，但角色的目的是在由任何需要它的人員取得。使用者擁有永久的長期憑證，但角色僅提供暫時憑證。如需進一步了解，請參閱 IAM 使用者指南中的建立 IAM 使用者 (而非角色) 的時機。

IAM 角色

IAM 角色是您 AWS 帳戶 內部具有特定許可的身分。它類似 IAM 使用者，但不與特定的人員相關聯。您可以切換角色，在中暫時擔任 IAM 角色。 AWS Management Console 您可以透過呼叫 AWS CLI 或 AWS API 作業或使用自訂 URL 來擔任角色。如需使用角色的方法更多相關資訊，請參閱 IAM 使用者指南中的使用 IAM 角色。

使用暫時憑證的 IAM 角色在下列情況中非常有用：

- 聯合身分使用者存取 – 若要向聯合身分指派許可，請建立角色，並為角色定義許可。當聯合身分進行身分驗證時，該身分會與角色建立關聯，並獲授予由角色定義的許可。如需有關聯合角色的相關資訊，請參閱 IAM 使用者指南中的為第三方身分提供者建立角色。如果您使用 IAM Identity Center，

則需要設定許可集。為控制身分驗證後可以存取的內容，IAM Identity Center 將許可集與 IAM 中的角色相關聯。如需有關許可集的資訊，請參閱 AWS IAM Identity Center 使用者指南中的許可集。

- 暫時 IAM 使用者許可 – IAM 使用者或角色可以擔任 IAM 角色來暫時針對特定任務採用不同的許可。
- 跨帳戶存取權 – 您可以使用 IAM 角色，允許不同帳戶中的某人 (信任的委託人) 存取您帳戶中的資源。角色是授予跨帳戶存取權的主要方式。但是，對於某些策略 AWS 服務，您可以將策略直接附加到資源（而不是使用角色作為代理）。若要了解跨帳戶存取權角色和資源型政策間的差異，請參閱 IAM 使用者指南中的 IAM 角色與資源類型政策的差異。
- 跨服務訪問 — 有些 AWS 服務 使用其他 AWS 服務功能。例如，當您在服務中進行呼叫時，該服務通常會在 Amazon EC2 中執行應用程式或將物件儲存在 Amazon Simple Storage Service (Amazon S3) 中。服務可能會使用呼叫主體的許可、使用服務角色或使用服務連結角色來執行此作業。
  - 轉寄存取工作階段 (FAS) — 當您使用 IAM 使用者或角色在中執行動作時 AWS，您會被視為主體。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 會使用主體呼叫的權限 AWS 服務，並結合要求 AWS 服務 向下游服務發出要求。只有當服務收到需要與其 AWS 服務 他資源互動才能完成的請求時，才會發出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱《轉發存取工作階段》。
  - 服務角色 – 服務角色是服務擔任的 IAM 角色，可代表您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需更多資訊，請參閱 IAM 使用者指南中的建立角色以委派許可給 AWS 服務。
  - 服務連結角色 — 服務連結角色是連結至. AWS 服務服務可以擔任代表您執行動作的角色。服務連結角色會顯示在您的中， AWS 帳戶 且屬於服務所有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。
- 在 Amazon EC2 上執行的應用程式 — 您可以使用 IAM 角色來管理在 EC2 執行個體上執行的應用程式以及發出 AWS CLI 或 AWS API 請求的臨時登入資料。這是在 EC2 執行個體內儲存存取金鑰的較好方式。若要將 AWS 角色指派給 EC2 執行個體並提供給其所有應用程式，請建立連接至執行個體的執行個體設定檔。執行個體設定檔包含該角色，並且可讓 EC2 執行個體上執行的程式取得暫時憑證。如需更多資訊，請參閱 IAM 使用者指南中的利用 IAM 角色來授予許可給 Amazon EC2 執行個體上執行的應用程式。

若要了解是否要使用 IAM 角色或 IAM 使用者，請參閱 IAM 使用者指南中的建立 IAM 角色 (而非使用者) 的時機。

使用政策管理存取權

您可以透 AWS 過建立原則並將其附加至 AWS 身分識別或資源來控制中的存取。原則是一個物件 AWS ，當與身分識別或資源相關聯時，會定義其權限。 AWS 當主參與者 (使用者、root 使用者或角色

工作階段) 提出要求時，評估這些原則。政策中的許可決定是否允許或拒絕請求。大多數原則會 AWS 以 JSON 文件的形式儲存在中。如需 JSON 政策文件結構和內容的更多相關資訊，請參閱 IAM 使用者指南中的 JSON 政策概觀。

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

預設情況下，使用者和角色沒有許可。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

IAM 政策定義該動作的許可，無論您使用何種方法來執行操作。例如，假設您有一個允許 `iam:GetRole` 動作的政策。具有該原則的使用者可以從 AWS Management Console AWS CLI、或 AWS API 取得角色資訊。

身分型政策

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要了解如何建立身分類型政策，請參閱 IAM 使用者指南中的建立 IAM 政策。

身分型政策可進一步分類成內嵌政策或受管政策。內嵌政策會直接內嵌到單一使用者、群組或角色。受管理的策略是獨立策略，您可以將其附加到您的 AWS 帳戶. 受管政策包括 AWS 受管政策和客戶管理的策略。若要了解如何在受管政策及內嵌政策間選擇，請參閱 IAM 使用者指南中的在受管政策和內嵌政策間選擇。

資源型政策

資源型政策是連接到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 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。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中指定主體。主參與者可以包括帳戶、使用者、角色、同盟使用者或。 AWS 服務

資源型政策是位於該服務中的內嵌政策。您無法在以資源為基礎的政策中使用 IAM 的 AWS 受管政策。

存取控制清單 (ACL)

存取控制清單 (ACL) 可控制哪些委託人 (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 和 Amazon VPC 是支援 ACL 的服務範例。 AWS WAF若要進一步了解 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 支援其他較不常見的原則類型。這些政策類型可設定較常見政策類型授予您的最大許可。

- 許可界限 – 許可範圍是一種進階功能，可供您設定身分型政策能授予 IAM 實體 (IAM 使用者或角色) 的最大許可。您可以為實體設定許可界限。所產生的許可會是實體的身分型政策和其許可界限的交集。會在 `Principal` 欄位中指定使用者或角色的資源型政策則不會受到許可界限限制。所有這類政策中的明確拒絕都會覆寫該允許。如需許可範圍的更多相關資訊，請參閱 IAM 使用者指南中的 IAM 實體許可範圍。
- 服務控制策略（SCP）— SCP 是 JSON 策略，用於指定中組織或組織單位（OU）的最大權限。AWS Organizations AWS Organizations 是一種用於分組和集中管理您企業擁 AWS 帳戶 有的多個服務。若您啟用組織中的所有功能，您可以將服務控制政策 (SCP) 套用到任何或所有帳戶。SCP 限制成員帳戶中實體的權限，包括每個 AWS 帳戶根使用者帳戶。如需組織和 SCP 的更多相關資訊，請參閱 AWS Organizations 使用者指南中的 SCP 運作方式。
- 工作階段政策 – 工作階段政策是一種進階政策，您可以在透過編寫程式的方式建立角色或聯合使用者的暫時工作階段時，作為參數傳遞。所產生工作階段的許可會是使用者或角色的身分型政策和工作階段政策的交集。許可也可以來自資源型政策。所有這類政策中的明確拒絕都會覆寫該允許。如需更多資訊，請參閱 IAM 使用者指南中的工作階段政策。

多種政策類型

將多種政策類型套用到請求時，其結果形成的許可會更為複雜、更加難以理解。要了解如何在涉及多個政策類型時 AWS 確定是否允許請求，請參閱《IAM 使用者指南》中的政策評估邏輯。

如何 AWS 服務 使用 IAM

若要深入瞭解如何使 AWS 服務 用大多數 IAM 功能，請參閱 IAM 使用者指南中的與 IAM 搭配使用的 AWS 服務。

要了解如何將特定的 IAM AWS 服務 與 IAM 搭配使用，請參閱相關服務用戶指南的安全部分。

疑難排解 AWS 身分和存取

使用下列資訊可協助您診斷和修正使用和 IAM 時可能會遇到的 AWS 常見問題。

主題

我沒有執行操作的授權 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 awes:*GetWidget* 許可時發生。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 awes: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 awes:*GetWidget* 動作存取 *my-example-widget* 資源。

如果您需要協助，請聯絡您的 AWS 系統管理員。您的管理員提供您的簽署憑證。

我沒有授權執行 iam：PassRole

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 允許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 沒有將角色傳遞至該服務的許可。

```
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 系統管理員。您的管理員提供您的簽署憑證。

我想允許我以外的人訪 AWS 帳戶 問我的 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：

- 若要瞭解是否 AWS 支援這些功能，請參閱如何 AWS 服務 使用 IAM。
- 若要了解如何提供對您所擁有資源 AWS 帳戶 的存取權，請參閱 IAM 使用者指南中您擁有的另一 AWS 帳戶 個 IAM 使用者提供存取權限。
- 若要了解如何將資源存取權提供給第三方 AWS 帳戶，請參閱 IAM 使用者指南中的提供第三方 AWS 帳戶 擁有的存取權。
- 若要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)。
- 若要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 角色與資源型政策的差異。

本 AWS 產品或服務的合規驗證

若要瞭解 AWS 服務 是否屬於特定規範遵循方案的範圍內，請參閱AWS 服務 遵循規範計劃方案中的，並選擇您感興趣的合規方案。如需一般資訊，請參閱AWS 規範計劃AWS。

您可以使用下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱下載中的報告中的 AWS Artifact。

您在使用時的合規責任取決 AWS 服務 於資料的敏感性、公司的合規目標以及適用的法律和法規。AWS 提供下列資源以協助遵循法規：

- 安全性與合規性快速入門指南 — 這些部署指南討論架構考量，並提供部署以安全性和合規性 AWS 為重點的基準環境的步驟。
- 在 Amazon Web Services 上架構 HIPAA 安全性與合規性 — 本白皮書說明公司如何使用建立符合 HIPAA 資格的應 AWS 用程式。

> **Note**
>
> 並非所有人 AWS 服務 都符合 HIPAA 資格。如需詳細資訊，請參閱 HIPAA 資格服務參照。

- AWS 合規資源AWS — 此工作簿和指南集合可能適用於您的產業和所在地。
- AWS 客戶合規指南 — 透過合規的角度瞭解共同的責任模式。這份指南總結了在多個架構 (包括美國國家標準技術研究所 (NIST)、支付卡產業安全標準委員會 (PCI) 和國際標準化組織 (ISO)) 中，保 AWS 服務 護指引並對應至安全控制的最佳實務。
- 使用AWS Config 開發人員指南中的規則評估資源 — 此 AWS Config 服務會評估您的資源組態符合內部實務、產業準則和法規的程度。
- AWS Security Hub— 這 AWS 服務 提供了內部安全狀態的全面視圖 AWS。Security Hub 使用安全控制，可評估您的 AWS 資源並檢查您的法規遵循是否符合安全業界標準和最佳實務。如需支援的服務和控制清單，請參閱 Security Hub controls reference。
- Amazon GuardDuty — 透過監控環境中的 AWS 帳戶可疑和惡意活動，藉此 AWS 服務 偵測您的工作負載、容器和資料的潛在威脅。 GuardDuty 可協助您滿足特定合規性架構所要求的入侵偵測需求，如 PCI DSS 等各種合規性需求。
- AWS Audit Manager— 這 AWS 服務 有助於您持續稽核您的 AWS 使用情況，以簡化您管理風險的方式，以及遵守法規和業界標準的方式。

本 AWS 產品或服務透過其支援的特定 Amazon 網路服務 (AWS) 服務，遵循共同的責任模式。如需 AWS 服務安全性資訊，請參閱AWS 服務安全性說明文件頁面和符合性計劃 AWS 遵循工作範圍的 AWS 服務。

本 AWS 產品或服務的復原能力

AWS 全球基礎架構是圍繞 AWS 區域 和可用區域建立的。

AWS 區域 提供多個實體分離和隔離的可用區域，這些區域透過低延遲、高輸送量和高度備援的網路連線。

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。

如需區域和可用區域的相關 AWS 資訊，請參閱AWS 全域基礎結構。

本 AWS 產品或服務透過其支援的特定 Amazon 網路服務 (AWS) 服務，遵循共同的責任模式。如需 AWS 服務安全性資訊，請參閱AWS 服務安全性說明文件頁面和符合性計劃 AWS 遵循工作範圍的 AWS 服務。

本 AWS 產品或服務的基礎架構安全性

此 AWS 產品或服務使用受管理的服務，因此受到 AWS 全球網路安全性的保護。有關 AWS 安全服務以及如何 AWS 保護基礎結構的詳細資訊，請參閱AWS 雲端安全 若要使用基礎架構安全性的最佳做法來設計您的 AWS 環境，請參閱安全性支柱架構良 AWS 好的架構中的基礎結構保護。

您可以使用 AWS 已發佈的 API 呼叫，透過網路存取本「 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。

本 AWS 產品或服務透過其支援的特定 Amazon 網路服務 (AWS) 服務，遵循共同的責任模式。如需 AWS 服務安全性資訊，請參閱AWS 服務安全性說明文件頁面和符合性計劃 AWS 遵循工作範圍的 AWS 服務。

中的組態和弱點分析 AWS Toolkit for Visual Studio

Toolkit for Visual Studio 被釋放到視覺工作室 Marketplace 作為新的功能或修復的開發。這些更新有時會包含安全性更新，因此請務必將 Toolkit for Visual Studio 保持在最新狀態。

確認已啟用擴充功能的自動更新

1. 開啟擴充功能管理員，方法是選擇 [工具、擴充功能和更新 (Visual Studio 2017)] 或 [擴充功能]、[管理擴充功能] (Visual Studio 2019)。
2. 選擇 [變更您的擴充功能和更新設定 (視覺工作室 2017)，或變更擴充功能的設定 (視覺工作室 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 Toolkit for Visual Studio 的更新。

AWS Toolkit for Visual Studio 使用者指南的文件記錄

上次文件更新日期：2021 年 4 月 21 日

文件歷史紀錄

下表說明《 AWS Toolkit for Visual Studio 使用者指南》最近的重要變更。如需獲得此文件更新的通知，您可以訂閱 RSS 摘要。

變更	描述	日期
內容更新與維護	更新 UI 和 AWS 樣式準則變更的內容。	2024年3月6日
內容更新與維護	更新 UI 和 AWS 樣式準則變更的內容。	2024年3月6日
內容更新與維護	更新 UI 和 AWS 樣式準則變更的內容。	2024年3月6日
內容更新與維護	更新 UI 和 AWS 樣式準則變更的內容。	2024年3月6日
內容更新與維護	更新 UI 和 AWS 樣式準則變更的內容。	2024年3月6日
設定和驗證的更新	設定和驗證主題已更新，以改善安全性和工具組上線體驗。請參閱入門和驗證以及存取主題目錄以檢視變更。	2023 年 6 月 22 日
身份驗證和訪問	提供 AWS 憑據現在是身份驗證和訪問。重構目錄和子主題以符合 AWS 樣式和安全性需求。	2023 年 5 月 4 日

新增的一般疑難排解	疑難排解主題包含與相關服務的 AWS Toolkit for Visual Studio 一般疑難排解資訊。	2023年4月30日
設定區段和主題的更新	本使用者指南中的 AWS Toolkit for Visual Studio章節和主題設定已更新，以改善的登機體驗 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 AWS Toolkit for Visual Studio章節和主題設定已更新，以改善的登機體驗 AWS Toolkit for Visual Studio。	2023 年 1 月 30 日
新增 2022 年 AWS Toolkit for Visual Studio 資訊	視覺工作室 2022 年的 Support 已新增至 AWS Toolkit for Visual Studio.	2022 年 12 月 20 日
發佈至 AWS 指南的更新	文件更新，以反映針對 GA 啟動的服務所做的變更。	2022 年 7 月 6 日
標題更新和重新定位	進行了輕微的標題更改以更好地反映內容。指南現在位於「發佈 AWS 指南」中。	2022 年 7 月 6 日
部署到 AWS：標題和內容更新	正式標題為：使用 AWS 工具組進行部署的指南區段具有更新的目錄 (目錄)，現在標題為：部署到 AWS. 下列指南已完成淘汰，無法再存取：部署至 Elastic Beanstalk (舊版) 和部署至 AWS CloudFormation (舊版)。您可以從本指南中的更新目錄中找到有關 Elastic Beanstalk 和雲形成的更新內容。	2022 年 7 月 6 日

部署 ASP.NET 核心 2.0 應用程式 (Fargate) 現在是舊版指南	本文件涉及舊版服務和功能。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	本文件涉及舊版服務和功能。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	本文件涉及舊版服務和功能。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
新的指南主題：使用 CloudWatch 記錄	在視覺工作室指南中為 Amazon CloudWatch 日誌整合創建了新的概述主題。	2022 年 6 月 29 日
新的指南主題：設定 CloudWatch 記錄檔整合	在視覺工作室指南中為 Amazon CloudWatch 日誌集成創建了新的設置部分。	2022 年 6 月 29 日
CloudWatch 記錄檔整合	在 Visual Studio 中建立 Amazon CloudWatch 日誌整合的新指南，其中包括指南主題：設定 CloudWatch 日誌和在 Visual Studio 中使用 CloudWatch 日誌。	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 的變更以及發佈建議的改進。	2022 年 6 月 1 日
新增發佈至 AWS 可供預覽	增強的部署體驗，提供適合您應用程式的 AWS 服務指引。	2021 年 10 月 21 日

登入資料的 SSO 和 MFA 支援 AWS	已更新，以記錄登入資料中對 AWS 單一登入 (IAM 身分中心) 和多重要素身份驗 AWS 證的新支援。	2021 年 4 月 21 日
基本 AWS Lambda 項目創建碼頭圖像	增加了對 Lambda 容器映像的支持。	2020 年 12 月 1 日
安全性內容	已新增安全內容。	2020 年 2 月 6 日
提供 AWS 認證	已更新為在共用 AWS 認證檔案中建立認證設定檔的相關資訊。	2019 年 6 月 20 日
在視覺工作室的 AWS 工具組中使用 AWS Lambda 專案	視覺工作室 2019 年的 Support 已新增至 AWS 工 Toolkit for Visual Studio。	2019 年 3 月 28 日
教學課程：建立 Amazon Rekognition Lambda 應用程式	視覺工作室 2019 年的 Support 已新增至 AWS 工 Toolkit for Visual Studio。	2019 年 3 月 28 日
教學課程：使用 Lambda 建置和測試無伺服器應用 AWS 程式	視覺工作室 2019 年的 Support 已新增至 AWS 工 Toolkit for Visual Studio。	2019 年 3 月 28 日
正在設定 AWS Toolkit for Visual Studio	視覺工作室 2019 的 Support 已新增至 AWS Toolkit for Visual Studio.	2019 年 3 月 28 日
部署一個核心 2.0 應用程序（Fargate）	視覺工作室 2019 年的 Support 已新增至 AWS 工 Toolkit for Visual Studio。	2019 年 3 月 28 日
部署核心 2.0 應用程式	視覺工作室 2019 年的 Support 已新增至 AWS 工 Toolkit for Visual Studio。	2019 年 3 月 28 日

創建一個 AWS CloudFormation 模板項目	視覺工作室 2019 年的 Support 已新增至 AWS 工 Toolkit for Visual Studio。	2019 年 3 月 28 日
容器服務的詳細視圖	已新增有關資 AWS 源管理器所提供之 Amazon 彈性容器服務叢集和容器儲存庫的詳細檢視資訊。	2018 年 2 月 16 日
部署到 Amazon EC2 Container Service	已新增部署至 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	已新增有關如何使用 Fargate 啟動類型透過 Amazon ECS 部署容器化 ASP.NET 核心 2.0 應用程式以 Linux 為目標的資訊。	2018 年 2 月 16 日
使用 EC2 部署容器服務	已新增有關如何使用 EC2 啟動類型透過 Amazon ECS 部署容器化 ASP.NET 核心 2.0 應用程式以 Linux 為目標的資訊。	2018 年 2 月 16 日
部署到 Amazon EC2 Container Service 的登入資料	已新增有關如何在部署到 Amazon EC2 容器服務時指定登入資料的資訊。	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。