用户指南

AWS 适用于 Visual Studio 的工具包

AWS 适用于 Visual Studio 的工具包: 用户指南

Table of Contents

AWS Toolkit for Visual Studio

这是 AWS Toolkit for Visual Studio 的用户指南。如果您正在查找有关 AWS Toolkit for VS Code 的信息，请参阅《User Guide for the AWS Toolkit for Visual Studio Code》。

什么是 Toolkit for Visual Studio？

AWS Toolkit for Visual Studio 是 Visual Studio IDE 的插件，可让您更轻松地开发、调试和部署使用亚马逊 Web Services 的.NET 应用程序。Visual Studio 2019 及更高版本支持 Visual Studio 的工具包。有关如何下载和安装该套件的详细信息，请参阅本用户指南中的安装和设置主题。

> **Note**
>
> Visual Studio 的工具包也已在 Visual Studio 2008、2010、2012、2013、2015 和 2017 版本中发布。但是，这些版本已不再受支持。有关更多信息，请参阅本用户指南中的安装和设置主题。

Toolkit for Visual Studio 包含以下增强开发体验的功能。

AWS 探险家

AWS 资源管理器工具窗口可从 IDE 的 “查看” 菜单中找到，它使您可以从 Visual Studio IDE 内部与许多 AWS 服务进行交互。支持的数据服务包括亚马逊简单存储服务 (Amazon S3)、亚马逊 SimpleDB、亚马逊简单通知服务 (Amazon SNS)、亚马逊简单队列服务 (Amazon SQS)、亚马逊简单队列服务 (Amazon SQS) Simple Queue Service 和亚马逊。 CloudFront AWS Explorer 还允许访问亚马逊弹性计算云 (Amazon EC2) 管理 AWS Identity and Access Management 、(IAM) 用户和策略管理、向和部署无服务器应用程序和功能以及 AWS Lambda 向和部署 Web 应用程序。 AWS Elastic Beanstalk AWS CloudFormation

凭证和区域管理

AWS Explorer 支持多个 AWS 账户（包括 IAM 用户账户）和区域，使您可以轻松地将显示的视图从一个账户更改为另一个账户，或者查看和管理不同区域的资源和服务。

Amazon EC2

在 AWS 资源管理器中，您可以查看可用的亚马逊系统映像 (AMI)，使用这些 AMI 创建 Amazon EC2 实例，然后使用 Windows 远程桌面连接到这些实例。 AWS Explorer 还支持支持功能，例如创建和管理密钥对和安全组的功能。

AWS Lambda

您可以使用 Lambda 托管无服务器 .NET 内核 C# 函数和无服务器应用程序。使用蓝图可以快速创建新的无服务器项目，在开发您的无服务器应用程序时实现良好的开端。

AWS CodeCommit

CodeCommit 与 Visual Studio 团队资源管理器集成。这使得克隆和创建存储库以及在 CodeCommit IDE 中处理源代码更改变得容易。

Amazon DynamoDB

DynamoDB 是一项快速、高度可扩展、高度可用且经济实惠的非关系数据库服务。Toolkit for Visual Studio 提供了在开发上下文中使用 Amazon DynamoDB 的功能。利用 Toolkit for Visual Studio，您可以在 DynamoDB 表中创建和编辑属性并对表运行扫描操作。

Amazon S3

您可以使用拖放操作轻松快速地将内容上传到 Amazon S3 桶或从 Amazon S3 下载内容。此外，您还可以为存储桶中的对象方便地设置权限、元数据和标签。

Amazon RDS

AWS 资源管理器可以帮助你在 Visual Studio 中创建和管理 Amazon RDS 资产。使用 Microsoft SQL Server 的 Amazon RDS 实例还可以添加到 Visual Studio 的服务器浏览器。

AWS Elastic Beanstalk

您可以使用 Elastic Beanstalk 将 .NET Web 应用程序项目部署到 AWS。您可以从 IDE 中将应用程序部署到单实例环境，或者部署到完全负载均衡的自动扩展环境。您还可以方便快速地部署应用程序的新版本而无需退出 Visual Studio。如果您的应用程序在 Amazon RDS 中使用 SQL Server，则部署向导还可以设置 Elastic Beanstalk 中应用程序环境与 Amazon RDS 中数据库实例之间的连接。Toolkit for

Visual Studio 还包括独立的命令行部署工具。使用部署工具可以使得部署成为您的构建流程中的一个自动部分，或将部署包含在 Visual Studio 之外的其他脚本场景中。

AWS CloudFormation

你可以使用 Visual Studio 的 Toolkit for Visual Studio 编辑支持编辑 IntelliSense 器和语法突出显示的 AWS CloudFormation JSON 格式的模板。使用 AWS CloudFormation 模板，您可以描述要实例化以托管应用程序的资源。然后在 IDE 中将模板部署到 AWS CloudFormation。模板中描述的资源已为您预配置，使您能够腾出时间专注于开发应用程序的功能。

AWS Identity and Access Management (IAM)

AWS 在 Explorer 中，您可以创建 IAM 用户、角色和策略，并将策略附加到用户。

相关信息

要打开问题或查看当前未解决的问题，请访问 https://github.com/aws/aws-toolkit-visual-studio/issues。

要了解有关 Visual Studio 的更多信息，请访问 https://visualstudio.microsoft.com/vs/。

亚马逊 Q 和亚马逊 CodeWhisperer

什么是 Amazon Q

截至 2024 年 4 月 30 日，亚马逊现已成 CodeWhisperer 为 Amazon Q Developer 的一部分，其中包括内联代码建议和安全扫描。

要了解有关与 Amazon Q 开发者合作的更多信息 AWS Toolkit for Visual Studio，请参阅 Amazon Q 开发者用户指南中的 IDE 中的 Amazon Q 开发者主题。有关 Amazon Q 计划和定价的详细信息，请参阅 Amazon Q 定价指南。

下载 Toolkit for Visual Studio

您可以通过 IDE 中的 Visual Studio Marketplace 下载、安装和设置 Toolkit for Visual Studio。有关详细说明，请参阅本用户指南入门主题中的安装 AWS Toolkit for Visual Studio 部分。

从 Visual Studio Marketplace 下载 Toolkit

在网络浏览器中导航到 AWS Visual Studio 下载站点，下载 Toolkit for Visual Studio 安装文件。

其他来自 AWS 的 IDE 工具包

除了 Visual Studio 的 Toolkit for Visual Studio 之外，AWS还提供适用于 VS Code 的 IDE 工具包和 JetBrains

AWS Toolkit for Visual Studio Code 链接

- 点击此链接，从 VS Code Marketplace 下载 AWS Toolkit for Visual Studio Code。
- 要了解有关 AWS Toolkit for Visual Studio Code 的更多信息，请参阅 AWS Toolkit for Visual Studio Code 用户指南。

AWS Toolkit for JetBrains 链接

- 点击此链接AWS Toolkit for JetBrains从 Mark JetBrains etplace 下载。
- 要了解有关 AWS Toolkit for JetBrains 的更多信息，请参阅 AWS Toolkit for JetBrains 用户指南。

入门

借助 AWS Toolkit for Visual Studio，您可以直接通过 Visual Studio 集成式开发环境（IDE）使用 AWS 服务和资源。

为了帮助您入门，以下主题介绍了如何安装、设置和配置 AWS Toolkit for Visual Studio。

主题

安装和设置 AWS Toolkit for Visual Studio

以下主题介绍如何下载、安装、设置和卸载 AWS Toolkit for Visual Studio。

主题

先决条件

以下是设置支持的 AWS Toolkit for Visual Studio版本的先决条件。

- Visual Studio 19 或更高版本
- Windows 10 或更高版本的 Windows
- 对 Windows 和 Visual Studio 的管理员访问权限
- 有效 AWS 的 IAM 证书

> **Note**
>
> 不支持的版本适用于 Visual Studio 2008、2010、2012、2013、2015 和 2017。 AWS Toolkit for Visual Studio 要下载不带支持的版本，请导航到 AWS Toolkit for Visual Studio 登录页面，然后从下载链接列表中选择所需的版本。
> 要了解有关 IAM 凭证的更多信息或注册账户，请访问 AWS 控制台门户。

正在安装 AWS Toolkit for Visual Studio

要安装 AWS Toolkit for Visual Studio，请从以下步骤中找到您的 Visual Studio 版本并完成必要的步骤。 AWS Toolkit for Visual Studio 可在AWS Toolkit for Visual Studio登录页面上找到所有版本的下载链接。

> **Note**
>
> 如果您在安装时遇到问题 AWS Toolkit for Visual Studio，请参阅本指南中的安装问题疑难解答主题。

AWS Toolkit for Visual Studio 为 Visual Studio 2022 安装

要从 Visual Studio 安装 AWS Toolkit for Visual Studio 2022，请完成以下步骤：

1. 在主菜单中导航到扩展，然后选择管理扩展。
2. 在搜索框中搜索 AWS。
3. 选择 Visual Studio 2022 相关版本的下载按钮，然后按照安装提示进行操作。

   > **Note**
   >
   > 您可能需要手动关闭并重新启动 Visual Studio 才能完成安装过程。

4. 下载和安装完成后，您可以从 “查看” 菜单中选择 “AWS 资源管理器” AWS Toolkit for Visual Studio 来打开。

AWS Toolkit for Visual Studio 为 Visual Studio 2019 安装

要从 Visual Studio 安装 AWS Toolkit for Visual Studio 2019，请完成以下步骤：

1. 在主菜单中导航到扩展，然后选择管理扩展。
2. 在搜索框中搜索 AWS。
3. 选择 Visual Studio 2017 和 2019 的下载按钮，然后按照提示进行操作。

> **Note**
>
> 您可能需要手动关闭并重新启动 Visual Studio 才能完成安装过程。

4. 下载和安装完成后，您可以从 “查看” 菜单中选择 “AWS 资源管理器” AWS Toolkit for Visual Studio 来打开。

正在卸载 AWS Toolkit for Visual Studio

要卸载 AWS Toolkit for Visual Studio，请从以下步骤中找到您的 Visual Studio 版本并完成必要的步骤。

正在卸载 Visual Studio 2022 版 AWS Toolkit for Visual Studio

要从 Visual Studio 卸载 AWS Toolkit for Visual Studio 2022，请完成以下步骤：

1. 在主菜单中导航到扩展，然后选择管理扩展。
2. 在管理扩展导航菜单中，展开已安装标题。
3. 找到 AWS Toolkit for Visual Studio 2022 扩展并选择卸载按钮。

> **Note**
>
> 如果在导航菜单的 “已安装” 部分中看 AWS Toolkit for Visual Studio 不到，则可能需要重新启动 Visual Studio。

4. 按照屏幕提示完成卸载过程。

正在卸载 Visual Studio 2019 版 AWS Toolkit for Visual Studio

要从 Visual Studio 中卸载 AWS Toolkit for Visual Studio 2019，请完成以下步骤：

1. 在主菜单中导航到工具，然后选择管理扩展。
2. 在管理扩展导航菜单中，展开已安装标题。

3. 找到 AWS Toolkit for Visual Studio 2019 扩展并选择卸载按钮。
4. 按照屏幕提示完成卸载过程。

正在卸载 Visual Studio 2017 版 AWS Toolkit for Visual Studio

要在 Visual Studio 中卸载 AWS Toolkit for Visual Studio 2017，请完成以下步骤：

1. 在主菜单中，导航到工具，然后选择扩展和更新。
2. 在扩展和更新导航菜单中，展开已安装标题。
3. 找到 AWS Toolkit for Visual Studio 2017 扩展并选择卸载按钮。
4. 按照屏幕提示完成卸载过程。

正在卸载 Visual Studio 2013 或 2015 版 AWS Toolkit for Visual Studio

要卸载 AWS Toolkit for Visual Studio 2013 或 2015，请完成以下步骤：

1. 在 Windows 控制面板中，打开程序和功能。

> **Note**
>
> 您可以通过 Windows 命令提示符或 Windows 运行对话框运行 `appwiz.cpl`，立即打开“程序和功能”。

2. 从已安装程序列表中，打开 AWS Tools for Windows 的上下文菜单（右键单击）。
3. 选择卸载，然后按照提示完成卸载过程。

> **Note**
>
> 在卸载过程中，您的 Samples 目录不会被删除。如果您修改了示例，则此目录会保留。必须手动移除此目录。

正在连接到 AWS

大多数 Amazon Web Services (AWS) 服务和资源都是通过 AWS 账户管理的。无需 AWS 账户即可使用 AWS Toolkit for Visual Studio，但是，如果没有连接，Toolkit 的功能将受到限制。

如果您之前设置过 AWS 帐户并通过其他 AWS 服务（例如 AWS Command Line Interface）进行身份验证，则 Visual Studio 的 Toolkit for Visual Studio 会自动检测您的凭据。

先决条件

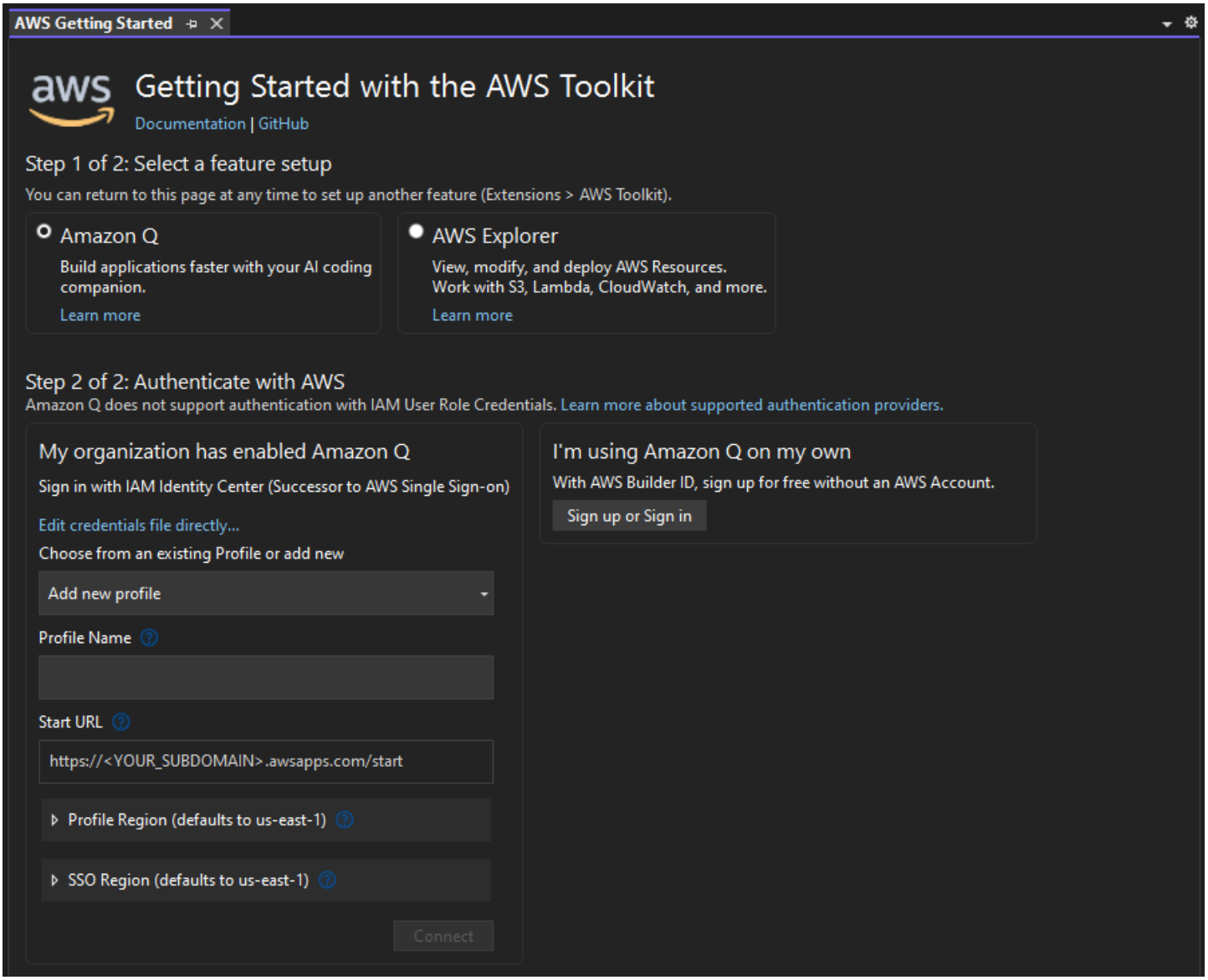
如果你是新用户 AWS 或尚未创建账户，那么要将 Visual Studio Toolkit for Visual Studio 与你的 AWS 账户相关联，主要有 3 个步骤：

1. 注册 AWS 帐户：您可以从注册门户网站AWS 注册一个 AWS 帐户。有关设置新 AWS 帐户的详细信息，请参阅《AWS 设置用户指南》中的概述主题。
2. 设置身份验证：使用 Visual Studio Toolkit for Visual Studio 中的 AWS 账户进行身份验证有三种主要方法。要了解上述每种方法的更多信息，请参阅本《用户指南》中的身份验证和访问权限主题。
3. 使用@@ 工具包进行 AWS 身份验证：完成本用户指南以下各节中的步骤，即可通过工具包与您的 AWS 账户建立联系。

AWS 从工具包连接到

要从 Visual Studio 的 Toolkit for Visual Studio 连接到您的 AWS 帐户，请完成以下步骤打开 AWS Toolkit 用户界面入门（连接 UI）。

1. 在 Visual Studio 主菜单中，展开 “扩展”，然后展开AWS 工具包。
2. 从 “AWS 工具包” 菜单选项中选择 “入门”。
3. AWS 工具包连接入门用户界面将在 Visual Studio 中打开。



下表介绍了与每种功能兼容的身份验证方法。要详细了解 3 种身份验证方法 AWS IAM Identity Center、 AWS Identity and Access Management 凭据和 AWS Builder ID 中的每种方法，请参阅本用户指南中的身份验证和访问目录。

> **Note**
>
> 目前，在 Visual Studio 的 Toolkit for Visual Studio CodeCatalyst 中使用时，你只需要在克隆第三方仓库时使用 AWS 生成器 ID 进行授权。

Amazon Q 开发者版	AWS 探险家	Amazon CodeCatalyst

☑ AWS 建造者 ID	☒ AWS 建造者 ID	☑ AWS 建造者 ID
☑ IAM Identity Center	☑ IAM Identity Center	☒ IAM 身份中心
☒ AWS IAM 证书	☑ AWS IAM 证书	☒ AWS IAM 证书

Amazon Q 开发者身份验证

要开始使用 Amazon Q Developer，请使用您的 AWS IAM Identity Center 或 AWS 建筑商 ID 凭证进行身份验证并进行连接。

以下过程介绍了如何进行身份验证并将 Toolkit 与您的 AWS 账户相连接。

通过 IAM Identity Center 进行身份验证和连接

1. 从 AWS Toolkit 连接入门用户界面中，选择 Amazon Q 开发者半导体以展开 Amazon Q 开发者身份验证选项。

   > **Note**
   >
   > 如果不存在存储的证书，请继续执行步骤 3 以添加或更新您的 IAM Identity Center 证书。

2. 在 “我的组织已启用 Amazon Q Developer” 部分，展开 “从现有个人资料中选择” 或添加新下拉菜单，从存储的凭证列表中进行选择。
3. 从配置文件类型下拉菜单中，选择 AWS IAM Identity Center
4. 在配置文件名称文本字段中，输入您要进行身份验证的 IAM Identity Center 配置文件。**`Profile Name`**
5. 在 “开始 URL” 文本字段中**`Start URL`**，输入附加到您的 IAM 身份中心证书的。
6. 从个人资料区域（默认为 us-east-1）下拉菜单中，选择由您正在进行身份验证的 IAM Identity Center 用户个人资料定义的个人资料区域。
7. 从 SSO 区域（默认为 us-east-1）下拉菜单中，选择由您的 IA M 身份中心证书定义的 SSO 区域，然后选择 Connect 按钮打开使用 IAM 身份中心登录对话框。 AWS
8. 从 “使用 AWS IAM Identity Center 登录” 对话框中，选择 “继续浏览器” 按钮，在默认 Web 浏览器中打开 AWS 授权请求站点。
9. 确认 IDE 中的安全代码与 Web 浏览器中显示的 AWS 授权请求确认码相匹配，然后选择 “提交并继续” 按钮继续。

10. 按照默认 Web 浏览器中的提示进行操作，当授权过程完成时，您会收到通知，可以安全地关闭浏览器，然后返回 Visual Studio。

使用 AWS 建筑商 ID 进行身份验证并进行连接

1. 从 AWS Toolkit 连接入门用户界面中，选择 Amazon Q 开发者半导体以展开 Amazon Q 开发者身份验证选项。
2. 在 “我自己使用 Amazon Q Developer” 部分中，选择 “注册” 或 “登录” 按钮，打开 “使用 AWS 建筑商 ID 登录” 对话框。
3. 选择 “继续浏览器” 按钮，在默认 Web 浏览器中打开 “ AWS 授权请求” 站点。
4. 确认 IDE 中的安全代码与 Web 浏览器中显示的 AWS 授权请求确认码相匹配，然后选择 “提交并继续” 按钮继续。
5. 按照默认 Web 浏览器中的提示进行操作，当授权过程完成时，您会收到通知，可以安全地关闭浏览器，然后返回 Visual Studio。

AWS Explorer 的身份验证

要开始使用工具包中的 AWS Explorer，请使用您的 IAM 身份中心证书或 IAM 凭证进行身份验证和连接。

以下过程介绍了如何进行身份验证并将 Toolkit 与您的 AWS 账户相连接。

通过 IAM Identity Center 进行身份验证和连接

1. 从 AWS Toolkit 连接入门用户界面中，选择 AWS Explorer 径向以展开 Amazon Q 开发者身份验证选项。
2. 从 **`Profile Type`** 下拉菜单中，选择 AWS IAM Identity Center。
3. 在配置文件名称文本字段中，输入您要使用的 **`Profile Name`** IAM Identity Center 配置文件。
4. 在 “开始 URL” 文本字段中**`Start URL`**，输入附加到您的 IAM 身份中心证书的。
5. 从个人资料区域（默认为 us-east-1）下拉菜单中，选择由您正在进行身份验证的 IAM Identity Center 用户个人资料定义的个人资料区域。
6. 从 SSO 区域（默认为 us-east-1）下拉菜单中，选择由您的 IA M 身份中心证书定义的 SSO 区域。
7. 选择 “继续浏览器” 按钮，在默认 Web 浏览器中打开 “AWS 授权请求” 站点。

8. 确认 IDE 中的安全代码与 Web 浏览器中显示的AWS 授权请求确认码相匹配，然后选择 “提交并继续” 按钮继续。
9. 按照默认 Web 浏览器中的提示进行操作，当授权过程完成时，您会收到通知，可以安全地关闭浏览器，然后返回 Visual Studio。

使用 IAM 凭证进行身份验证和连接

1. 从 AWS Toolkit 连接入门用户界面中，选择 AWS Explorer 径向以展开 Amazon Q 开发者身份验证选项。
2. 从**Profile Type**下拉菜单中选择 IAM 用户角色。
3. 在配置文件名称文本字段中，输入要进行身份验证**Profile Name**的配置文件的。
4. 在访问密钥 ID 文本字段中**Access Key ID**，输入要进行身份验证的配置文件的。
5. 在 “密钥” 文本字段中**Secret Key**，输入要进行身份验证的配置文件的。
6. 从 “存储位置”（默认为 “共享凭据文件”）下拉菜单中，指定是使用共享凭据文件还是使用 .NET 加密存储来存储凭据。
7. 从个人资料区域（默认为 us-east-1）下拉菜单中，选择附加到您要进行身份验证的个人资料的个人资料区域。

的安装问题疑难解答 AWS Toolkit for Visual Studio

已知以下信息可以解决安装时出现的常见安装问题AWS Toolkit for Visual Studio。

如果您在安装时遇到错误AWS Toolkit for Visual Studio或者不清楚安装是否已完成，请查看以下各节中的信息。

视觉工作室的管理员权限

该AWS Toolkit for Visual Studio扩展需要管理员权限，以确保所有AWS服务和功能均可访问。

如果您拥有本地管理员权限，则您的管理员权限可能无法直接扩展到您的 Visual Studio 实例。

要在本地启动具有管理员权限的 Visual Studio

1. 在 Windows 中，找到 Visual Studio 应用程序启动器（图标）。
2. 打开（右键单击）Visual Studio 图标的上下文菜单以打开快捷菜单。

3. 从上下文菜单中选择 “以管理员身份运行”。

要使用管理员权限远程启动 Visual Studio：

1. 在 Windows 中，找到用于连接到 Visual Studio 远程实例的应用程序的应用程序启动器。
2. 打开（右键单击）应用程序的快捷菜单以打开快捷菜单。
3. 从上下文菜单中选择 “以管理员身份运行”。

> **Note**
>
> 无论你是在本地启动程序还是远程连接，Windows 都可能会提示你确认管理凭据。

获取安装日志

如果您已完成上述 “管理员权限” 部分中的步骤，并且已确认您正在使用管理员权限运行或连接到 Visual Studio，则获取安装日志文件可以帮助诊断其他问题。

要AWS Toolkit for Visual Studio从`.vsix`文件手动安装并生成安装日志文件，请完成以下步骤。

1. 在AWS Toolkit for Visual Studio登录页面上，点击下载链接，保存要安装的AWS Toolkit for Visual Studio版本的`.vsix`文件。
2. 从 Visual Studio 主菜单中，展开 “工具” 标题，展开 “命令行” 子菜单，然后选择 Visual Studio 开发者命令提示符。
3. 在 Visual Studio 开发者**`vsixinstaller`**命令提示符下输入以下格式的命令：

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

4. `[file path to log file]`替换为要在其中创建安装日志的目录的文件名和完整文件路径。带有您指定的文件路径和文件名的`vsixinstaller`命令示例如下：

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

5. `[file path to Toolkit installation file]`替换为所在目录的完整文件路径。`AWSToolkitPackage.vsix`

   包含 Toolkit 安装文件的完整文件路径的`vsixinstaller`命令示例应与以下内容类似：

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

6. 检查确保您的文件名和路径正确，然后运行vsixinstaller命令。

   完整vsixinstaller命令的示例类似于以下内容：

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

安装不同的视觉工作室扩展

如果您已获得安装日志文件，但仍然无法确定安装过程失败的原因，请检查您是否能够安装其他 Visual Studio 扩展。安装不同的 Visual Studio 扩展可以进一步了解您的安装问题。如果你无法安装任何 Visual Studio 扩展，则可能需要解决 Visual Studio 的问题，而不是AWS Toolkit for Visual Studio。

联系支持人员

如果您已查看本指南中包含的所有部分并需要其他资源或支持，则可以从 AWS Toolkit for Visual StudioGithub Issues 网站查看过去的问题或打开新问题。

为了帮助加快问题的解决方案：

- 检查过去和当前的问题，看看其他人是否遇到过类似的情况。
- 详细记录你为解决问题而采取的每一个步骤。
- 保存您在安装AWS Toolkit for Visual Studio或其他扩展程序时获得的所有日志文件。
- 将您的AWS Toolkit for Visual Studio安装日志文件附加到新问题上。

配置文件和窗口绑定

Toolkit for Visual Studio

在使用 Toolkit for Visual Studio 的发布工具、向导和其他功能时，请注意以下几点：

- EAWS xplorer 窗口一次绑定到单个配置文件和区域。从AWS资源管理器打开的 Windows 默认为该绑定的配置文件和区域。
- 打开新窗口后，您可以使用AWS Explorer 的该实例切换到不同的配置文件或区域。

- 适用于 Visual Studio 的工具包发布工具和功能会自动默认为AWS资源管理器中设置的配置文件和区域。
- 如果在发布工具、向导或功能中指定了新的配置文件或区域：之后创建的所有资源将继续使用新的配置文件和区域设置。
- 如果您打开了多个 Visual Studio 实例，则每个实例可以绑定到不同的配置文件和区域。
- AWS资源管理器保存上次指定的配置文件和区域，最后关闭的 Visual Studio 实例的值将保持不变。

身份验证和访问

您无需进行身份验证即可开始使用适用于 Visual Studio 的 AWS Toolkit for Visual Studio。 AWS 但是，大多数 AWS 资源都是通过 AWS 账户管理的。要访问所有 Visual Studio AWS Toolkit for Visual Studio 服务和功能，您至少需要两种类型的帐户身份验证：

1. 对您的 AWS 账户进行AWS Identity and Access Management (IAM) AWS IAM Identity Center身份验证或身份验证。大多数 AWS 服务和资源都是通过 IAM 和 IAM 身份中心管理的。
2. 对于某些其他 AWS 服务，AWS 生成器 ID 是可选的。

以下主题包含其他详细信息以及每种凭证类型和身份验证方法的设置说明。

主题

AWS 中的 IAM 身份中心证书 AWS Toolkit for Visual Studio

AWS IAM Identity Center 是管理 AWS 账户身份验证的推荐最佳实践。

有关如何为软件开发套件 (SDK) 设置 IAM 身份中心等的详细说明 AWS Toolkit for Visual Studio，请参阅《软件开发工具包和工具参考指南》的 IAM 身份中心身份验证部分。AWS

通过 IAM 身份中心进行身份验证 AWS Toolkit for Visual Studio

要 AWS Toolkit for Visual Studio 通过向您的`credentials`或`config`文件中添加 IAM 身份中心配置文件来通过 IAM 身份中心进行身份验证，请完成以下步骤。

1. 在首选的文本编辑器中，打开存储在`<hone-directory>\.aws\credentials`文件中的 AWS 凭据信息。
2. 在 `credentials file` 的 `[default]` 部分下，为 IAM Identity Center 命名配置文件添加模板。以下是示例模板：

> **Important**
>
> 在 `credential` 文件中创建条目时，请勿使用 profile 一词，因为这会与 `credential` 文件命名约定发生冲突。
> 仅当配置 `config` 文件中的命名配置文件时，才包含前缀词 `profile_`。

```
[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`**：指向贵组织 IAM Identity Center 用户门户的 URL。
- **`sso_region`**：包含您的 IAM 身份中心门户主机的 AWS 区域。这可能与稍后在默认`region`参数中指定的 AWS 区域不同。
- **`sso_account_id`**：包含您要向此 IAM 身份中心用户授予权限的 IAM 角色的 AWS 账户 ID。
- **`sso_role_name`**：使用此配置文件通过 IAM Identity Center 获取凭证时，定义用户权限的 IAM 角色的名称。
- **`region`**：此 IAM 身份中心用户登录的默认 AWS 区域。

> **Note**
>
> 您还可以通过运行`aws configure sso`命令将启用 IAM Identity Center AWS CLI 的配置文件添加到您的配置文件中。运行此命令后，您需要为 IAM 身份中心起始 URL (`sso_start_url`) 和托管 IAM 身份中心目录的 AWS 区域 (`region`) 提供值。
> 有关更多信息，请参阅《AWS Command Line Interface 用户指南》中的 “将 AWS CLI 配置为使用 AWS 单点登录”。

使用 IAM Identity Center 登录

使用 IAM Identity Center 配置文件登录时，默认浏览器将启动到 `credential file` 中指定的 `sso_start_url`。您必须先验证自己的 IAM 身份中心登录信息，然后才能访问中的 AWS 资源 AWS Toolkit for Visual Studio。如果凭证过期，则必须重复连接过程以获取新的临时凭证。

AWS IAM 证书

AWS IAM 凭证通过本地存储的访问密钥对您的 AWS 账户进行身份验证。

以下各节介绍如何设置 IAM 证书，以便通过您的 AWS 账户进行身份验证 AWS Toolkit for Visual Studio。

> **Important**
>
> 在设置 IAM 凭证以使用您的 AWS 账户进行身份验证之前，请注意：
>
> - 如果您已经通过其他 AWS 服务（例如 AWS AWS CLI）设置了 IAM 证书，则会 AWS Toolkit for Visual Studio 自动检测这些证书。
> - AWS 建议使用 AWS IAM Identity Center 身份验证。有关 AWS IAM 最佳实践的更多信息，请参阅 Identity and A ccess Managem AWS ent 用户指南的 “IAM 中的安全最佳实践” 部分。
> - 为了避免安全风险，在开发专用软件或处理真实数据时，请勿使用 IAM 用户进行身份验证，相反，请使用与身份提供商（例如）的联合 AWS IAM Identity Center。有关更多信息，请参阅《AWS IAM Identity Center User Guide》中的 What is IAM Identity Center?。

创建 IAM 用户

在将设置为使用 AWS 账户 AWS Toolkit for Visual Studio 进行身份验证之前，您需要在《AWS 软件开发工具包和工具参考指南》的 “使用长期证书进行身份验证” 主题中完成步骤 1：创建您的 IAM 用户和步骤 2：获取访问密钥。

> **Note**
>
> 步骤 3：更新共享凭证是可选步骤。
> 如果您完成了步骤 3，则会 AWS Toolkit for Visual Studio 自动从中检测您的凭证`credentials file`。

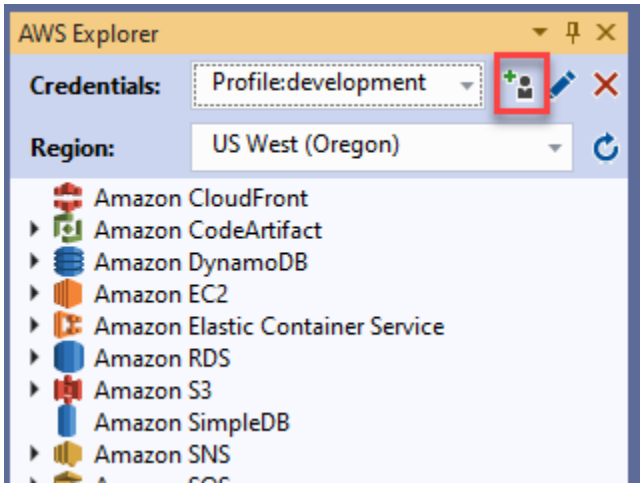
如果您尚未完成步骤 3，则将 AWS Toolkit for Visual Studio 引导您完成创建过程，`credentials file`如下面的 “从中创建证书文件 AWS Toolkit for Visual Studio” 部分所述。

创建凭证文件

要向 AWS Toolkit for Visual Studio添加用户或在其中创建 `credentials file`，请执行以下操作：

> **Note**
>
> 在 Toolkit 中添加新的用户配置文件时：
>
> - 如果 `credentials file` 已经存在，则新的用户信息将添加到现有文件中。
> - 如果 `credentials file` 不存在，则会创建一个新文件。

1. 从 AWS 资源管理器中选择 “新建账户资料” 图标以打开 “新建账户资料” 对话框。



2. 在新建账户配置文件对话框中填写必填字段，然后选择确定按钮创建 IAM 用户。

在 Toolkit 中编辑 IAM 用户凭证

要在 Toolkit 中编辑 IAM 用户凭证，请完成以下步骤：

1. 从 AWS 资源管理器的证书下拉列表中，选择要编辑的 IAM 用户证书。
2. 选择编辑配置文件图标以打开编辑配置文件对话框。

3. 在编辑配置文件对话框中完成更新，然后选择确定按钮保存更改。

要从 Toolkit 中删除 IAM 用户凭证，请完成以下步骤：

1. 从 AWS Explorer 的 “证书” 下拉列表中，选择要删除的 IAM 用户证书。
2. 选择删除配置文件图标以打开删除配置文件提示。
3. 确认您要删除该配置文件，以将其从 `Credentials file` 中移除。

> **Important**
>
> 无法在 AWS Toolkit for Visual Studio中通过编辑配置文件对话框编辑支持高级访问功能 [例如 IAM Identity Center 或多重身份验证（MFA）] 的配置文件。要更改这些类型的配置文件，必须使用文本编辑器编辑 `credentials file`。

使用文本编辑器编辑 IAM 用户凭证

除了使用管理 IAM 用户外 AWS Toolkit for Visual Studio，您还可以使用首选`credential files`的文本编辑器进行编辑。在 Windows 中，`credential file` 的默认位置是 `C:\Users\`*`USERNAME`*`\.aws\credentials`。

有关 `credential files` 位置和结构的更多详细信息，请参阅《AWS SDKs and Tools Reference Guide》的 Shared config and credentials files 部分。

通过 AWS Command Line Interface (AWS CLI) 创建 IAM 用户

AWS CLI 是另一个工具，您可以使用命令在中创建 IAM 用户`aws configure`。`credentials file`

有关从中创建 IAM 用户的 AWS CLI 详细信息，请参阅AWS CLI 用户指南中的配置 AWS CLI主题。

Toolkit for Visual Studio 支持以下配置属性：

```
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 生成器 ID

AWS Builder ID 是一种额外的 AWS 身份验证方法，可能需要使用某些服务或功能，例如通过 Amazon 克隆第三方存储库 CodeCatalyst。

有关 AWS 生成器 ID 身份验证方法的详细信息，请参阅《登录用户指南》中的 “使用 AWS 生成器 ID AWS 登录” 主题。

有关 CodeCatalyst 从中克隆存储库的更多信息 AWS Toolkit for Visual Studio，请参阅本用户指南中的使用Amazon CodeCatalyst 主题。

Toolkit for Visual Studio 中的多重身份验证（MFA）

多重身份验证 (MFA) 为您的账户提供了额外的安全保障。 AWS MFA 要求用户在访问网站或服务时提供登录凭证和来自支持的 AWS MFA 机制的唯一身份验证。 AWS

AWS 支持一系列虚拟设备和硬件设备进行 MFA 身份验证。以下是通过智能手机应用程序启用的虚拟 MFA 设备示例。要了解有关 MFA 设备选项的更多信息，请参阅《IAM 用户指南》中的在 AWS中使用多重身份验证（MFA）。

步骤 1：创建 IAM 角色，以向 IAM 用户委派访问权限

以下过程介绍如何设置向 IAM 用户分配权限的角色委派。有关角色委派的详细信息，请参阅《AWS Identity and Access Management 用户指南》中的创建向 IAM 用户委派权限的角色主题。

1. 转到 IAM 控制台，地址：https://console.aws.amazon.com/iam。
2. 在导航栏中选择角色，然后选择创建角色。
3. 在创建角色页面中，选择另一个 AWS 账户。
4. 输入所需的账户 ID 并选中需要 MFA 复选框。

> **Note**
>
> 要查找您的 12 位账号（ID），请在控制台顶部的导航栏上，选择支持，然后选择支持中心。

5. 选择下一步：权限。
6. 将现有策略附加到角色或为其创建新策略。您在此页面上选择的策略决定了 IAM 用户可以通过 Toolkit 访问哪些 AWS 服务。
7. 附加策略后，选择下一步：标签，设置向角色添加 IAM 标签的选项。然后，选择下一步：审核以继续。
8. 在审核页面中，输入所需的角色名称（例如 toolkit-role）。您也可以添加可选的角色描述。
9. 选择 创建角色。
10. 当显示确认消息（例如“角色 toolkit-role 已创建”）时，请在消息中选择该角色的名称。
11. 在摘要页面中，选择复制图标以复制角色 ARN 并将其粘贴到一个文件中。（在配置代入角色的 IAM 用户时，您需要此 ARN。）

步骤 2：创建代入角色权限的 IAM 用户

此步骤将创建一个没有权限的 IAM 用户，以便可以添加内联策略。

1. 转到 IAM 控制台，地址：https://console.aws.amazon.com/iam。
2. 在导航栏中，选择用户，然后选择添加用户。
3. 在添加用户页面中，输入所需的用户名（例如 toolkit-user），然后选中编程访问复选框。
4. 选择下一步：权限、下一步：标签和下一步：审核，进行翻页。您不必在此阶段添加权限，因为用户将代入通过角色委派的权限。
5. 在审核页面中，您会被告知此用户没有权限。选择 创建用户。
6. 在成功页面中，选择下载.csv 以下载包含访问密钥 ID 和秘密访问密钥的文件。（在凭证文件中定义用户的配置文件时，您将需要这两个信息。）
7. 选择关闭。

步骤 3：添加允许 IAM 用户代入角色的策略

以下过程将创建一个内联策略，允许用户代入角色（以及该角色的权限）。

1. 在 IAM 控制台的用户页面中，选择您刚刚创建的 IAM 用户（例如 toolkit-user）。
2. 在摘要页面的权限选项卡上，选择添加内联策略。
3. 在创建策略页面中，选择选择服务，在查找服务中输入 STS，然后从结果中选择 STS。
4. 在 “操作” 中，开始输入术语AssumeRole。当AssumeRole复选框出现时，将其选中。
5. 在资源部分，确保选中特定，然后单击添加 ARN 以限制访问。
6. 对于添加 ARN 对话框中的为角色指定 ARN，添加您在步骤 1 中创建的角色的 ARN。

   添加该角色的 ARN 后，与该角色关联的可信账户和角色名称将显示在账户和带路径的角色名称中。
7. 选择添加。
8. 返回创建策略页面，选择指定请求条件（可选），选中需要 MFA 复选框，然后选择关闭进行确认。
9. 选择查看策略
10. 在查看策略页面中，为策略输入名称，然后选择创建策略。

    权限选项卡中将显示直接附加到 IAM 用户的新内联策略。

步骤 4：为 IAM 用户管理虚拟 MFA 设备

1. 将虚拟 MFA 应用程序下载并安装到智能手机。

   有关支持的应用程序列表，请参阅多重身份验证资源页面。
2. 在 IAM 控制台中，从导航栏中选择用户，然后选择代入角色的用户（在本例中为 toolkit-user）。
3. 在摘要页面中，选择安全凭证选项卡，然后为已分配的 MFA 设备选择管理。
4. 在管理 MFA 设备窗格中，选择虚拟 MFA 设备，然后选择继续。
5. 在设置虚拟 MFA 设备窗格中，选择显示 QR 码，然后使用安装在智能手机上的虚拟 MFA 应用程序扫描二维码。
6. 扫描二维码后，虚拟 MFA 应用程序会生成一次性 MFA 代码。在 MFA 代码 1 和 MFA 代码 2 中连续输入两个 MFA 代码。
7. 选择 Assign MFA (分配 MFA)。
8. 返回用户的安全凭证选项卡，复制新分配的 MFA 设备的 ARN。

   ARN 包含您的 12 位账户 ID，其格式类似于以下内容：`arn:aws:iam::123456789012:mfa/toolkit-user`。在下个步骤中定义 MFA 配置文件时，您需要用到此 ARN。

步骤 5：创建配置文件以允许 MFA

以下过程创建了在 Visual Studio 的 Toolkit for Visual Studi AWS o 中访问服务时允许 MFA 的配置文件。

您创建的配置文件包括您在前面的步骤中复制和存储的三条信息：

- IAM 用户的访问密钥（访问密钥 ID 和秘密访问密钥）
- 向 IAM 用户委派权限的角色的 ARN
- 分配给 IAM 用户的虚拟 MFA 设备的 ARN

在包含您的 AWS 凭据的 AWS 共享凭证文件或 SDK 商店中，添加以下条目：

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

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

提供的示例中定义了两个配置文件：

- `[toolkit-user]` 配置文件包含您在步骤 2 中创建 IAM 用户时生成和保存的访问密钥和秘密访问密钥。
- `[mfa]` 配置文件定义了如何支持多因素身份验证。该配置文件有三个条目：

  ◦ `source_profile`：指定配置文件，将使用其凭证代入此配置文件中的 `role_arn` 设置所指定的角色。在本例中，该条目为 `toolkit-user` 配置文件。

  ◦ `role_arn`：指定 IAM 角色的 Amazon 资源名称（ARN），您将通过此角色执行使用此配置文件请求的操作。在本例中，该条目您在步骤 1 中创建的角色的 ARN。

  ◦`mfa_serial`：指定用户在代入角色时必须使用的 MFA 设备的标识或序列号。在本例中，该条目是您在步骤 3 中设置的虚拟设备的 ARN。

设置外部凭证

如果 AWS不直接支持您生成或查找凭证的方法，则可以在共享凭证文件中添加一个包含 `credential_process` 设置的配置文件。此设置指定运行的外部命令，以生成或检索要使用的身份验证凭证。例如，您可以在 `config` 文件中包含类似于以下内容的条目：

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

有关使用外部凭证和相关安全风险的更多信息，请参阅《AWS Command Line Interface 用户指南》中的使用外部进程获取凭证。

使用 AWS 服务

以下主题介绍如何开始使用适用于 Visual Studio 的 AWS Toolkit 中的 AWS 服务。

主题

亚马逊CodeCatalyst的 Visual Studio AWS 工具包

什么是 Amazon CodeCatalyst？

亚马逊CodeCatalyst是软件开发团队的基于云的协作空间。使用 Visual Studio AWS 工具包，您可以直接从 Visual Studio AWS 工具包中查看和管理CodeCatalyst资源。有关的更多信息CodeCatalyst，请参阅《亚马逊CodeCatalyst用户指南》。

以下主题介绍如何连接 Visual Studio AWS 工具包CodeCatalyst以及如何CodeCatalyst通过 Visual Studio AWS 工具包使用。

主题

亚马逊入门CodeCatalyst和 Visual Studio AWS 工具包

要通过 Visual Studio AWS 工具包开始与亚马逊CodeCatalyst合作，请完成以下操作。

主题

安装 Visual Studio AWS 工具包

在将 Visual Studio AWS 工具包集成到您的CodeCatalyst账户之前，请确保您使用的是 Visual Studio AWS 工具包的最新版本。有关如何安装和设置最新版本的 Visual Studio AWS 工具包的详细信息，请参阅本用户指南的 “设置 Visual Studio AWS 工具包” 部分。

创建CodeCatalyst账户和AWS生成器 ID

除了安装最新版本的 Visual Studio AWS 工具包外，您还必须拥有有效的AWS生成器 ID 和 CodeCatalyst帐户才能连接 Visual Studio AWS 工具包。如果您没有有效的 AWS Builder ID 或 CodeCatalyst帐户，请参阅《CodeCatalyst用户指南》中的 “设置方式CodeCatalyst” 部分。

> **Note**
>
> AWS生成器 ID 与您的AWS证书不同。有关如何注册和使用 AWS Builder ID 进行身份验证的说明，请参阅本用户指南中的 “身份验证和访问：AWSBuilder ID” 主题。
> 有关AWS生成器 ID 的详细信息，请参阅《AWS一般参考用户指南》中的 AWSBuilder ID 主题。

将适用于视觉工作室的AWS工具包与 CodeCatalyst

要将 Visual Studio AWS 工具包与您的CodeCatalyst帐户关联，请完成以下步骤。

1. 从 Visual Studio 的 Git 菜单项中，选择克隆存储库... 。
2. 在浏览存储库部分中，选择亚马逊CodeCatalyst作为提供商。
3. 在 “连接” 部分中，选择 “使用AWS生成器 ID 连接”，在首选 Web 浏览器中打开CodeCatalyst控制台。
4. 在浏览器提供的字段中输入您的AWS生成器 ID，然后按照说明继续。
5. 出现提示时，选择 “允许” 以确认 Visual Studio AWS 工具包与您的CodeCatalyst帐户之间的连接。连接过程完成后，CodeCatalyst会显示一条确认信息，表明关闭浏览器是安全的。

使用 Visual Studio AWS 工具包中的亚马逊CodeCatalyst资源

以下各节概述了适用于 Visual Studio AWS 工具包的亚马逊CodeCatalyst资源管理功能。

主题

克隆存储库

CodeCatalyst是一项基于云的服务，需要您连接到云端才能处理CodeCatalyst项目。要在本地处理项目，您可以将CodeCatalyst存储库克隆到本地计算机，并在下次连接到云端时与CodeCatalyst项目同步。

要将存储库克隆到本地计算机，请完成以下步骤。

1. 从 Visual Studio 的 Git 菜单项中，选择克隆存储库... 。
2. 在浏览存储库部分中，选择亚马逊CodeCatalyst作为提供商。

> **Note**
>
> 如果 “连接” 部分显示一条`Not Connected`消息，请先完成本用户指南的 “身份验证和访问：AWSBuilder ID” 部分中的步骤，然后再继续。

3. 选择要从中克隆存储库的空间和项目。
4. 从 “存储库” 部分中，选择要克隆的存储库。

5. 在路径部分中，选择要将存储库克隆到的文件夹。

> **Note**
>
> 此文件夹最初必须为空才能成功克隆。

6. 选择 “克隆” 开始克隆存储库。
7. 克隆存储库后，Visual Studio 将加载您克隆的解决方案

> **Note**
>
> 如果 Visual Studio 未在克隆存储库中打开解决方案，则可以从 “源代码控制” 菜单的 “打开 Git 存储库时自动加载解决方案” 设置中调整您的 Visual Studi o 选项。

问题排查

以下是解决在 Visual Studio AWS 工具包中使用亚马逊时出现CodeCatalyst的已知问题的疑难解答主题。

主题

- 凭证

凭证

如果您在尝试从中克隆基于 git 的存储库时遇到一个要求您提供凭据的对话框CodeCatalyst，则您的AWSCodeCommit凭据助手可能会被全局配置，从而造成干扰。CodeCatalyst有关AWSCodeCommit凭证帮助程序的更多信息，请参阅《AWSCodeCommit用户指南》的 “AWS使用 CLI 凭据帮助程序设置 HTTPS 连接到 Windows 上的AWSCodeCommit存储库的步骤” 部分。

要将AWSCodeCommit凭证助手限制为仅处理 CodeCommit URL，请完成以下步骤。

1. 在以下位置打开全局 git 配置文件：`%userprofile%\.gitconfig`
2. 在文件中找到以下部分：

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

3. 将该部分更改为以下内容：

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

4. 保存您的更改，然后完成克隆存储库的步骤。

亚马逊 CloudWatch Lisual Tudio

Amazon CloudWatch Logs ToolkitAWSToolkit for Visual Studio 使你能够监视、存储和访问 CloudWatch 记录资源，无需离开 IDE。了解有关设置 CloudWatch 服务和 Visudio CloudWatch 日志功能，请从以下主题中进行选择。

主题

- 设置 CloudWatch Locial Studio
- 使用 CloudWatch 日志流式传输

设置 CloudWatch Locial Studio

您可以先使用亚马逊 CloudWatch LoglkiToolkit for Visual StudioAWSaccount. 你可以创建新的AWS账号来自AWS登录site（站点）。大多数 CloudWatch Toolkit for Visual Studio 中提供的日志功能可以通过激活访问AWS凭证。如果某项特定功能需要额外的配置，这些要求将包含在使用 CloudWatch 日志指南。

有关设置的更多信息和选项 CloudWatch 日志，请参阅开始设置部分的亚马逊 CloudWatch 日志指南。

使用 CloudWatch 日志流式传输

亚马逊 CloudWatch 日志流式传输，使您能够监控、存储和访问 CloudWatch 中的日志的AWSToolkit for Visual Studio 有访问权限 CloudWatch 无需离开您的 IDE 即可记录要素通过简化 CloudWatch 记录开发流程，减少对工作流程的干扰。以下主题介绍如何使用 CloudWatch 日志集成。

主题

- CloudWatch 日志组

- CloudWatch 日志流
- CloudWatch 日志事件
- 额外访问权限 CloudWatch日志

CloudWatch 日志组

一个log group是一组log streams具有相同保留期、监控和访问控制设置。对可属于一个日志组的日志流数没有限制。

查看日志组

这些区域有：View Log Groups功能显示日志组 CloudWatch日志组。

要访问 “查看日志组” 功能并打开 CloudWatch 日志组，请完成以下步骤。

1. 从AWSExplorer，亚马逊 CloudWatch.
2. 双击日志组或打开上下文菜单（右键单击）并选择查看以打开CloudWatch 日志组.

> **Note**
>
> 这些区域有： CloudWatch 日志组资源管理器将在与解决方案资源管理器相同的窗口位置打开。

筛选日志组

您的个人账户可以包含数千个不同的日志组。要简化对特定群组的搜索，请使用filtering功能如下所述。

1. 从CloudWatch 日志组中，将光标置于位于窗口顶部的搜索栏。
2. 开始键入与要查找的日志组相关的前缀。
3. CloudWatch 日志组会自动更新以显示与您在上一步中指定的搜索词相匹配的结果。

删除日志组

要删除特定的日志组，请参考以下步骤。

1. 从CloudWatch 日志组中，右键单击您要删除的日志组。

2. 在提示时，确认您希望删除当前选定的日志组。
3. 选择是按钮删除选定的日志组，然后刷新CloudWatch 日志组.

刷新日志组

要刷新当前显示在CloudWatch 日志组，选择Resh 图标按钮位于工具条.

复制日志组 ARN

要复制特定日志组的 ARN，请完成下述步骤。

1. 从CloudWatch 日志组中，右键单击要从中复制 ARN 的日志组。
2. 选择复制 ARN选项。
3. ARN 现已复制到您的本地剪贴板并准备粘贴。

CloudWatch 日志流

日志流是共享同一个源的一系列日志事件。

> **Note**
>
> 在查看日志流时，请注意以下属性：
>
> - 默认情况下，日志流按最近的事件时间戳排序。
> - 与日志流关联的列可以按升序或降序排序，方法是切换插入符号位于列标题中。
> - 筛选的条目只能按以下顺序排序日志流名称.

查看日志流

1. 从CloudWatch 日志组双击日志组，然后选择查看日志流从上下文菜单中。
2. 将在文档窗口，其中包含与您的日志组关联的日志流的列表。

筛选日志流

1. 从日志流选项卡，位于文档窗口中，将光标设置在搜索栏。
2. 开始键入与您要查找的日志流相关的前缀。

3. 键入时，当前显示会自动更新，以根据您的输入过滤日志流。

刷新日志流

要刷新显示在文档窗口中，选择Resh 图标按钮，位于工具条，位于搜索栏.

复制日志ARN

要复制特定日志流的 ARN，请完成下述步骤。

1. 从日志流选项卡，位于文档窗口中，右键单击您要从中复制的日志流。
2. 选择复制 ARN选项。
3. ARN 现已复制到您的本地剪贴板并准备粘贴。

下载日志流式传输

这些区域有：导出日志流功能会在本地下载并存储选定的日志流，通过自定义工具和软件可以访问该日志流以进行其他处理。

1. 从日志流选项卡，位于文档窗口中，右键单击您要下载的日志流。
2. 选择导出日志流以打开导出到文本文件对话框。
3. 选择要在本地存储文件的位置，并在提供的文本字段中指定名称。
4. 通过选择确认下载确定. 下载的状态显示在视觉工作室任务状态中心

CloudWatch 日志事件

日志事件是对受监控的应用程序或资源记录的活动的记录。 CloudWatch.

日志事件操作

日志事件显示为表格。默认情况下，事件按从最早的事件到最近的事件进行排序。

以下操作与 Visual Studio 中的日志事件相关联：

- 换行文本模式：您可以通过单击事件来切换换换文本。
- 文本换行按钮：位于`document window` **`toolbar`**，此按钮可为所有条目打开和关闭文本换行。
- 将留言复制到剪贴板：选择要复制的邮件，然后右键单击所选内容并选择Copy键盘快捷键`Ctrl + C`)。

查看日志事件

1. 从文档] 窗口中，选择一个包含日志流列表的选项卡。
2. 双击日志流，或右键单击日志流并选择查看日志流从菜单中。
3. New 的日志事件选项卡将在文档窗口，其中包含与所选日志流关联的日志事件表。

筛选日志事件

有三种方法可以过滤日志事件：按内容、时间范围或两者兼而有之。要按内容和时间范围过滤日志事件，请先按内容或时间范围过滤消息，然后用另一种方法过滤这些结果。

要按内容过滤日志事件，请执行以下操作：

1. 从日志事件选项卡，位于文档窗口中，将光标置于位于窗口顶部的搜索栏。
2. 开始键入与要搜索的日志事件相关的术语或短语。
3. 键入时，当前显示会自动开始过滤日志事件。

> **Note**
>
> 筛选条件模式区分大小写。您可以通过将精确的字词和短语用双引号括起来 (* "" *)。有关筛选模式的更详细信息，请参阅筛选条件和模式语法亚马逊的话题 CloudWatch 指南。

要查看在特定时间范围内生成的日志事件，请执行以下操作：

1. 从日志事件选项卡，位于文档窗口中，选择日历图标按钮，位于工具条.
2. 使用提供的字段，指定要搜索的时间范围。
3. 当您指定日期和时间限制时，筛选的结果会自动更新。

> **Note**
>
> 这些区域有：清除筛选条件选项会清除所有你当前的 date-and-time 筛选选择。

刷新日志事件

要刷新显示在日志事件选项卡上，选择Resh 图标按钮，位于工具条.

额外访问权限 CloudWatch日志

您可以访问 CloudWatch 日志流式传输AWS服务和资源直接来自AWS视觉工作室中的工具包。

Lambda

要查看与 Lambda 函数关联的日志流，请执行以下操作：

> **Note**
>
> 您的 Lambda 执行角色必须具有适当的权限以将日志流式传输。 CloudWatch日志。有关所需的 Lambda 权限的更多信息， CloudWatch 日志，请参阅https://docs.aws.amazon.com/lambda/latest/dg/monitoring-cloudwatchlogs.html#monitoring-cloudwatchlogs-prereqs

1. 从AWS工具包资源管理器，展开Lambda.
2. 右键单击您要查看的函数，然后选择查看日志在中打开关联的日志流文档Windows。

使用 Lambda 集成查看日志流`function view`：

1. 从AWS工具包资源管理器，展开Lambda.
2. 右键单击您要查看的函数，然后选择视图函数在中打开函数视图文档Windows。
3. 从`function view`切换到日志选项卡上，将显示与所选 Lambda 函数关联的日志流。

ECS

要查看与 ECS 任务容器关联的日志资源，请完成以下过程。

> **Note**
>
> 为了让 Amazon ECS 服务将日志发送到 CloudWatch，则给定 Amazon ECS 任务的每个容器都必须满足所需的配置。有关所需设置和配置的其他信息，请参阅指南使用AWS日志日志驱动程序.

1. 从AWS工具包资源管理器，展开Amazon ECS.
2. 选择您要查看的 Amazon ECS 集群ECS 集群选项卡，位于文档Windows。

3. 从位于左侧的导航菜单中，ECS 集群选项卡上，选择任务列出与集群关联的所有任务。
4. 从任务显示，选择一个任务，然后选择查看日志链接，位于左下角。

> **Note**
>
> 此显示列出了集群中包含的所有任务，`View Logs`链接仅对满足所需日志配置的每个任务可见。
>
> - 如果 Task 仅与单个容器关联，则查看日志link 打开该容器的日志流。
> - 如果一个 Task 与多个容器关联，则查看日志链接打开查看 CloudWatch 适用于 ECS Task 的对话框中，使用容器：下拉菜单选择要查看其日志的容器，然后选择确定.

5. 这将打开一个新的选项卡。文档窗口显示与您的容器选择关联的日志流。

管理 Amazon EC2 实例

AWSAWS Explorer 提供了 Amazon 系统映像 (AMI) 和 Amazon Elastic Compute Cloud (Amazon EC2) 实例的详细视图。从这些视图中，您可从 AMI 启动 Amazon EC2 实例，连接到此实例，停止或终止此实例，所有这些操作都是在 Visual Studio 开发环境中执行的。您可使用实例视图从您的实例创建 AMI。有关更多信息，请参阅从 Amazon EC2 实例创建 AMI。

Amazon 系统映像和 Amazon EC2 实例视图

从AWSExplorer 中，您可显示 Amazon 系统映像 (AMI) 和 Amazon EC2 实例的视图。InAWS资源管理器，展开Amazon EC2节点。

要显示 AMI 视图，请在第一个子节点 AMI 上，打开上下文（右键单击）菜单，然后选择 View (查看)。

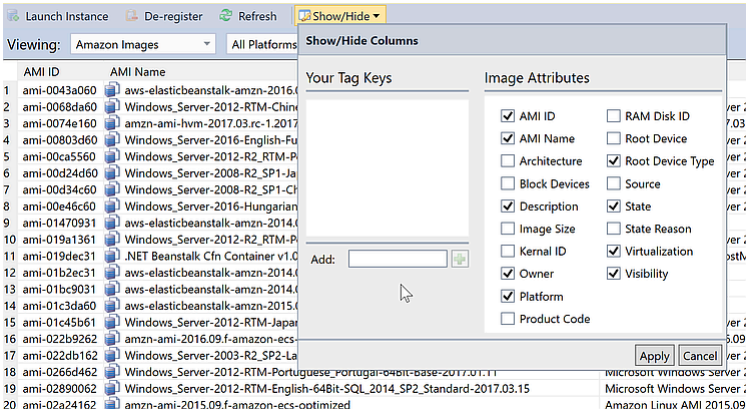
要显示 Amazon EC2 实例视图，请在 Instances (实例) 节点上，打开上下文（右键单击）菜单，然后选择 View (查看)。

您也可通过双击相应的节点来显示任一视图。

- 视图的范围为中指定的区域AWSExplorer（例如，美国西部（加利福尼亚北部）区域）。
- 您可通过单击并拖动来重新整理列。要对列中的值进行排序，请单击列标题。
- 可使用 Viewing (查看) 中的下拉列表和筛选器框来配置视图。初始视图将显示中指定的账户所拥有的任何平台类型（Windows 或 Linux）的 AMI。AWSExplorer。

显示/隐藏列

您还可选择位于视图顶部的 Show/Hide (显示/隐藏) 下拉列表来配置显示的列。如果您关闭并重新打开视图，您选择的列将保留。



AMI 和实例视图的 Show/Hide Columns (显示/隐藏列) UI

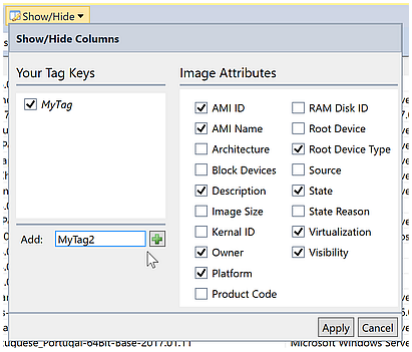
为 AMI、实例和卷添加标签

您也可以使用显示/Hide为 AMI、Amazon EC2 实例或您拥有的卷添加标签。标签是可让您将元数据附加到 AMI、实例和卷的名称-值对。标签名称的范围限定于您的账户，也可独立于您的 AMI 和实例。例如，如果您对 AMI 和实例使用相同的标签名称，则不会发生冲突。标签名称不区分大小写。

有关标签的更多信息，请参阅使用标签中的适用于 Linux 实例的 Amazon EC2 用户指南.

添加标签

1. 在 Add (添加) 框中，键入标签的名称。选择带加号 (+) 的绿色按钮，然后选择 Apply (应用)。



向 AMI 或 Amazon EC2 实例添加标签

新标签以斜体形式显示，这表示此标签未与任何值关联。

在列表视图中，标签名称显示为新列。如果标签已与至少一个值关联，则标签将在AWS Management Console.

2. 要为标签添加值，请双击标签对应的列中的单元格，然后键入值。要删除标签值，请双击单元格并删除文本。

   如果在 Show/Hide (显示/隐藏) 下拉列表中清除标签，则对应的列将从视图中消失。标签将与 AMI、实例或卷所关联的任何标签值一起保留。

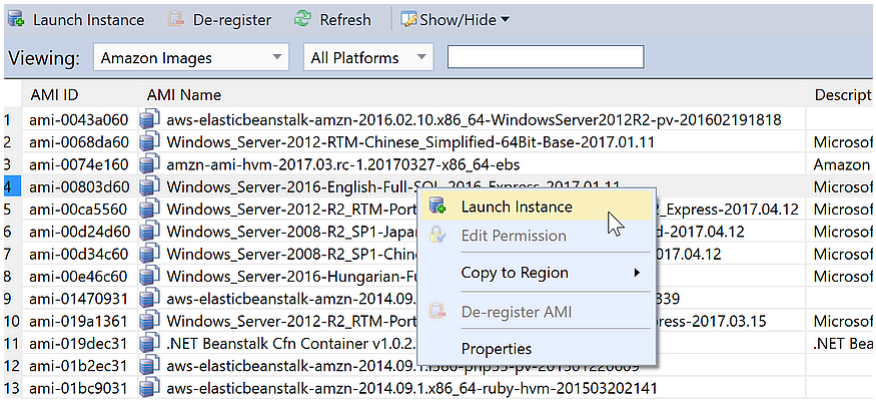
> **Note**
>
> 如果你清除了中的标签显示/Hide没有关联值的下拉列表，AWSToolkit 将完全删除标签。它将不再显示在列表视图或 Show/Hide (显示/隐藏) 下拉列表中。要再次使用标签，请使用 Show/Hide (显示/隐藏) 对话框重新创建它。

启动 Amazon EC2 实例

AWSAWS Explorer 提供了启动 Amazon EC2 实例所需的全部功能。在本节中，我们将选择并配置 Amazon 系统映像 (AMI)，然后将它作为 Amazon EC2 实例启动。

启动 Windows Server Amazon EC2 实例

1. 在 AMI 视图顶部左侧的下拉列表中，选择 Amazon Images (Amazon 映像)。在右侧的下拉列表中，选择 Windows。在筛选器框中，为 Elastic Block 存储键入 ebs。刷新视图可能需要一点时间。
2. 选择列表中的 AMI，打开上下文（右键单击）菜单，然后选择 Launch Instance (启动实例)。



AMI 列表

3. 在 Launch New Amazon EC2 Instance (启动新 Amazon EC2 实例) 对话框中，为您的应用程序配置 AMI。

   实例类型

   选择要启动的 EC2 实例的类型。您可以在 EC2 Pricing (EC2 定价) 页面上找到实例类型和定价信息的列表。

   名称

   为您的实例键入名称。此名称不能超过 256 个字符。

   密钥对

   密钥对用于获取您用于通过远程桌面协议 (RDP) 登录到 EC2 实例的 Windows 密码。选择您有权访问私有密钥的密钥对，或选择用于创建密钥对的选项。如果您在 Toolkit 中创建密钥对，则 Toolkit 可为您存储私有密钥。

   存储在 Toolkit 中的密钥对将被加密。`%LOCALAPPDATA%\AWSToolkit\keypairs`（通常是：`C:\Users\<user>\AppData\Local\AWSToolkit\keypairs`）。 您可以将加密 key pair 导出到`.pem`文件。

   a. 在 Visual Studio 中，选择查看然后单击AWS探险者.

   b. 单击 Amazon EC2，选择 Key Pairs (密钥对)。

   c. 此时将列出密钥对，并且由 Toolkit 创建/管理的密钥对将标记为 Stored in AWSToolkit (存储在 AWSToolkit 中)。

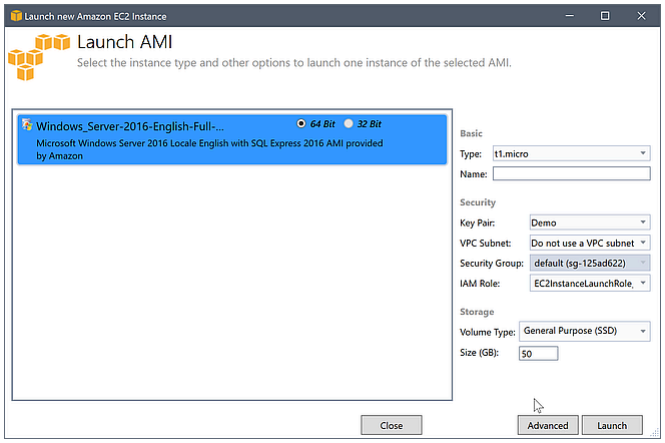
   d. 右键单击您创建的密钥对，然后选择 Export Private Key (导出私有密钥)。私有密钥将取消加密并保存在您指定的位置。

   安全组

   安全组控制 EC2 实例将接受的网络流量的类型。选择将允许端口 3389（RDP 使用的端口）上的传入流量的安全组，以便能连接到 EC2 实例。有关如何使用 Toolkit 创建安全组的信息，请参阅从中管理安全组AWS探险者.

   实例配置文件

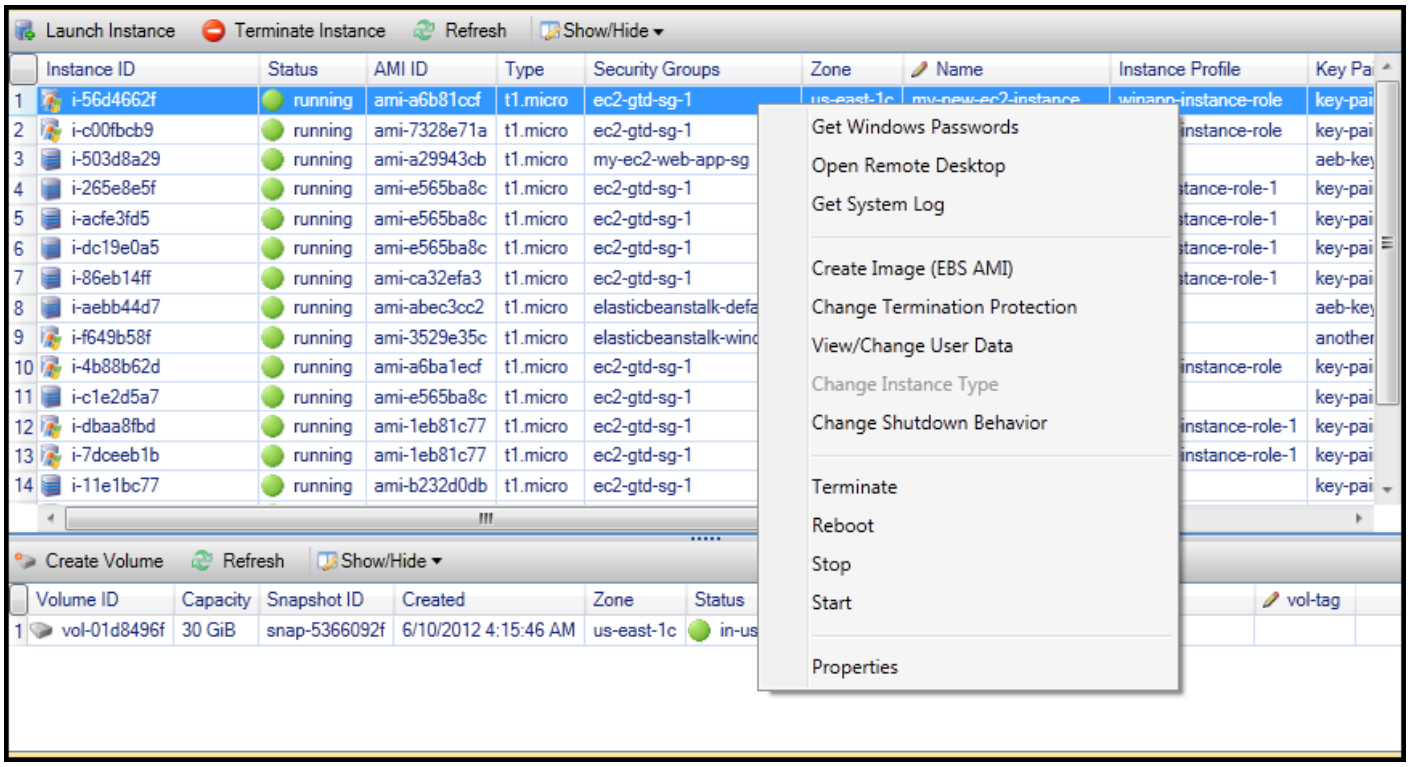
   实例配置文件是 IAM 角色的逻辑容器。当您选择实例配置文件时，会将对应的 IAM 角色与 EC2 实例关联。IAM 角色使用策略配置，这些策略指定对 Amazon Web Services 和账户资源的访问权限。当 EC2 实例与 IAM 角色关联时，在实例上运行的应用程序软件将使用 IAM 角色指定的权限运行。这使得应用程序软件无需指定任何操作即可运行。AWS凭证自己的凭证，从而使软件更安全。有关 IAM 角色的更多信息，请转到 IAM 用户指南。



EC2 Launch AMI (启动 AMI) 对话框

4. 选择启动。

InAWS资源管理器，在实例的子节点Amazon EC2，打开上下文 (右键单击) 菜单，然后选择查看. 这些区域有：AWSToolkit 将显示与有效账户关联的 Amazon EC2 实例的列表。您可能需要选择 Refresh (刷新) 才能查看您的新实例。当实例第一次显示时，它可能处于挂起状态，但在几分钟后，它将过渡到运行状态。



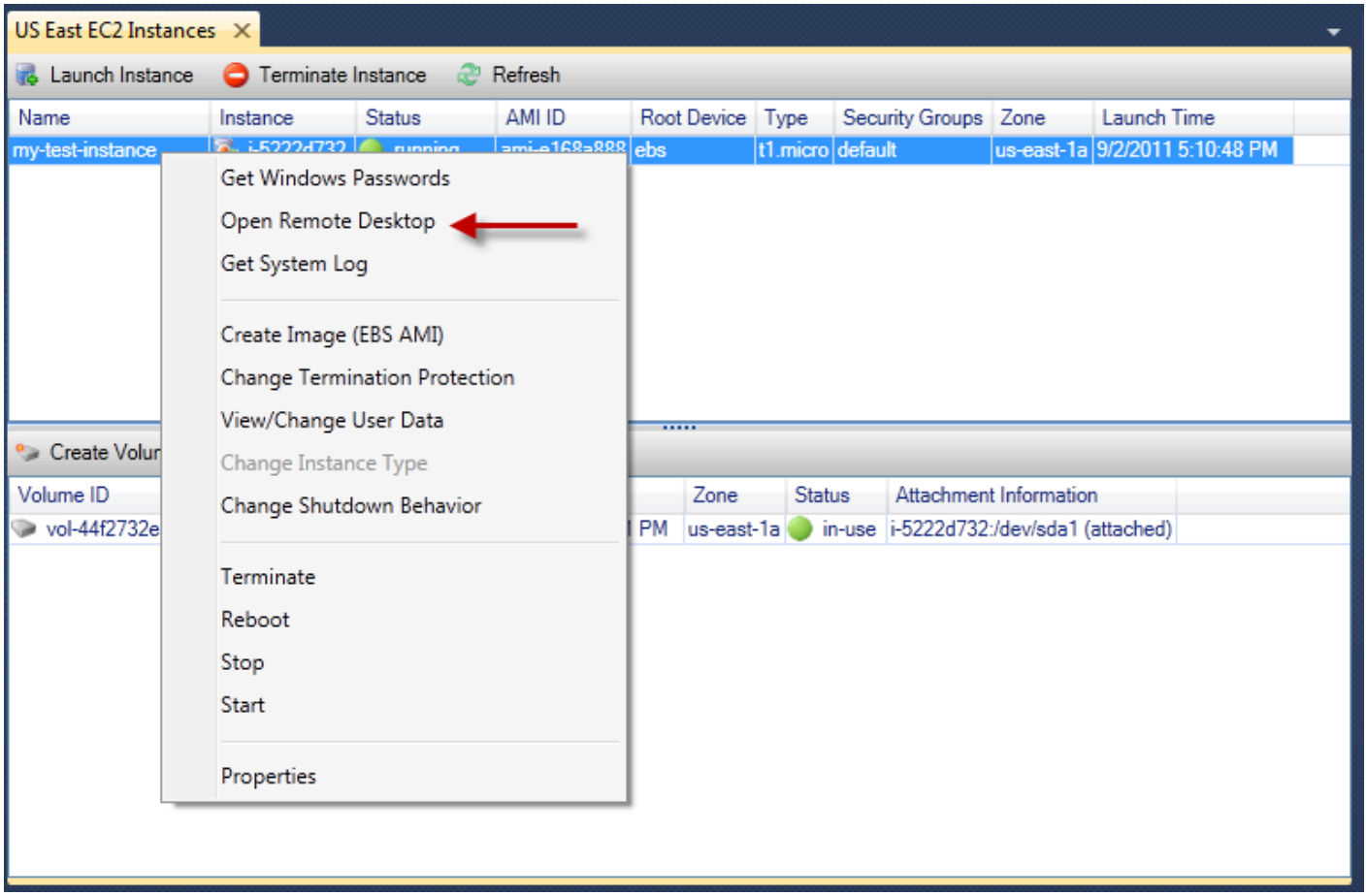
连接到 Amazon EC2 实例

您可使用 Windows 远程桌面连接到 Windows Server 实例。对于身份验证，AWSToolkit 可让您检索实例的管理员密码，也可使用与实例关联的已存储 key pair。在以下过程中，我们将使用存储的密钥对。

使用 Windows 远程桌面连接到 Windows Server 实例

1. 在 EC2 实例列表中，右键单击要连接到的 Windows Server 实例。从上下文菜单中，选择 Open Remote Desktop (打开远程桌面)。

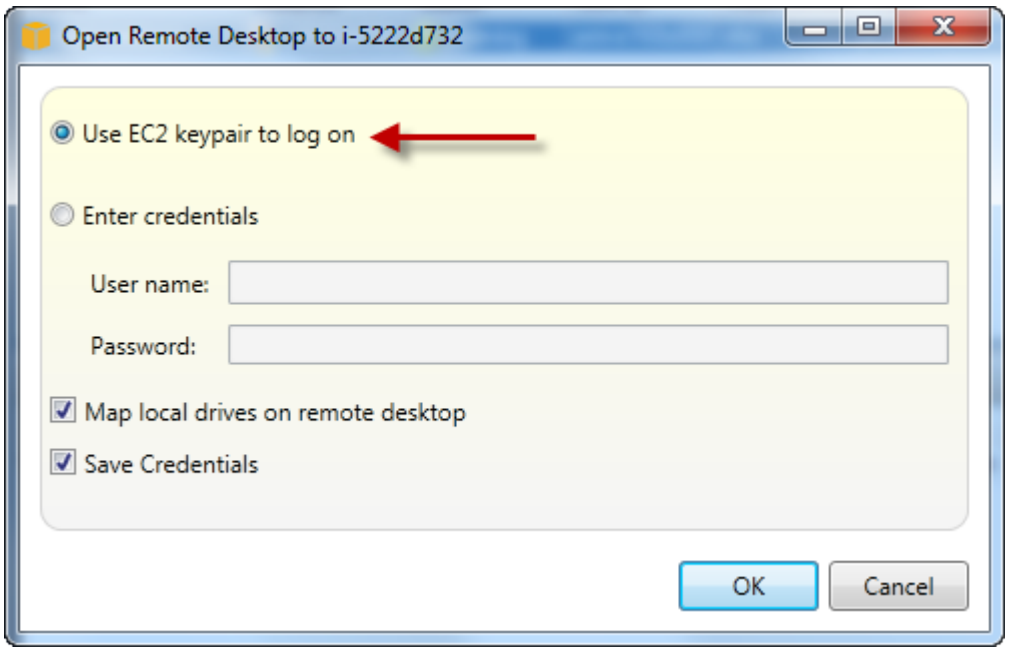
   如果您要使用管理员密码进行身份验证，请选择 Get Windows Passwords (获取 Windows 密码)。



   EC2 实例上下文菜单

2. 在 Open Remote Desktop (打开远程桌面) 对话框中，选择 Use EC2 keypair to log on (使用 EC2 密钥对进行登录)，然后选择 OK (确定)。

   如果您未 key pair 用AWSToolkit 中，指定包含私有密钥的 PEM 文件。

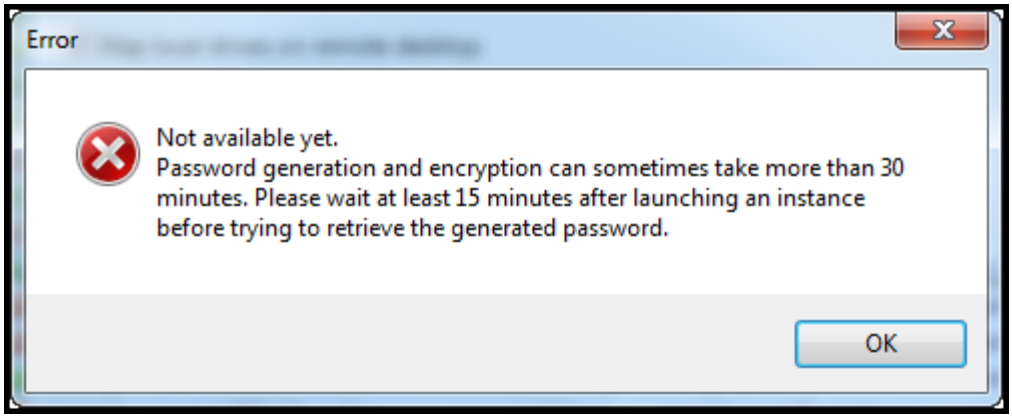


Open Remote Desktop (打开远程桌面) 对话框

3. Remote Desktop (远程桌面) 窗口将打开。由于已使用密钥对进行身份验证，因此您无需登录。您将以管理员身份在 Amazon EC2 实例上运行。

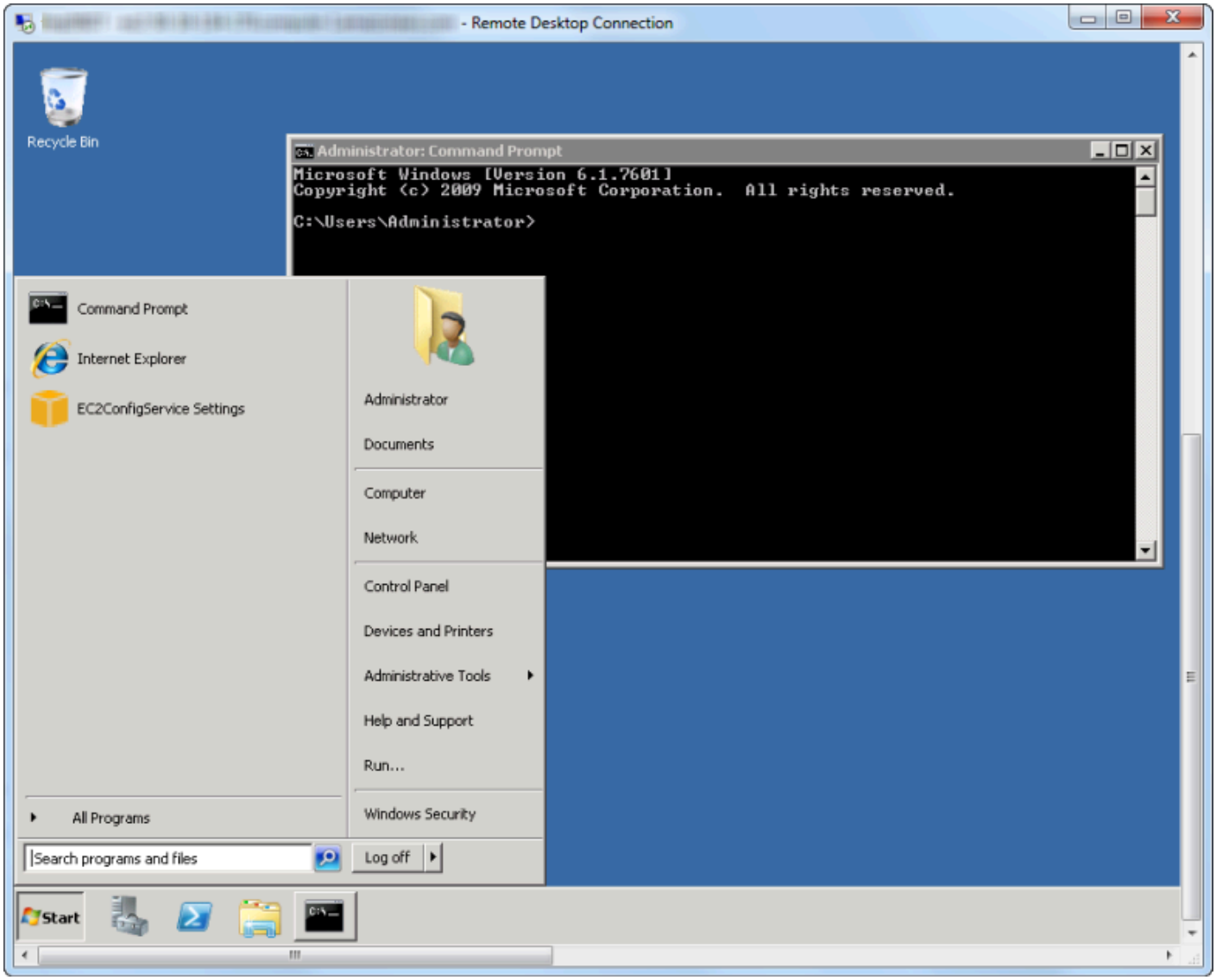
   如果 EC2 实例仅最近启动，您可能因以下两个原因无法连接：

   - 远程桌面服务可能尚未启动并运行。请等待几分钟，然后重试。
   - 密码信息可能尚未传输到实例。在此情况下，您将看到与下面类似的消息框。



密码尚不可用

以下屏幕截图显示以管理员身份通过远程桌面连接的用户。



远程桌面

结束 Amazon EC2 实例

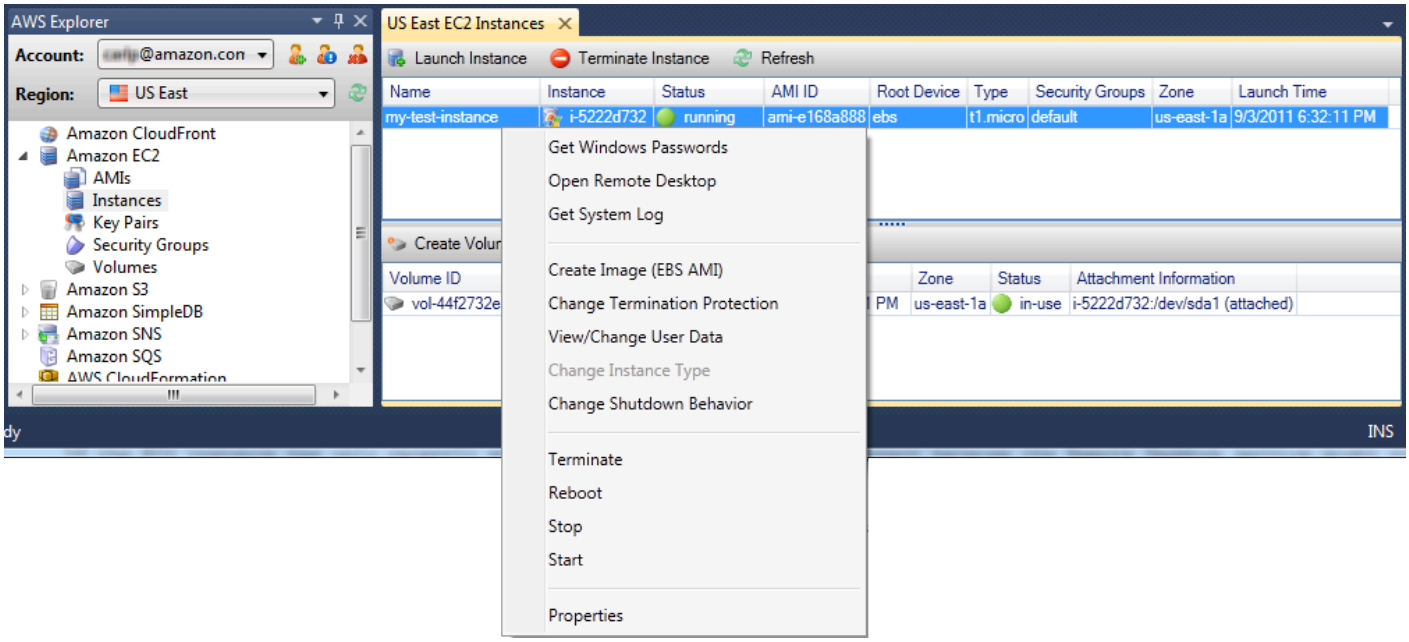
使用AWSToolkit 中，您可从 Visual Studio 停止或终止正在运行的 Amazon EC2 实例。要停止此实例，EC2 实例必须使用 Amazon EBS 卷。如果 EC2 实例未使用 Amazon EBS 卷，您的唯一选择是终止此实例。

如果您停止实例，EBS 卷上存储的数据将保留。如果您终止实例，实例的本地存储设备上存储的所有数据将丢失。在停止或终止的情况下，将停止向您收取 EC2 实例的费用。但是，如果您停止实例，将继续向您收取实例停止后保留 EBS 存储的费用。

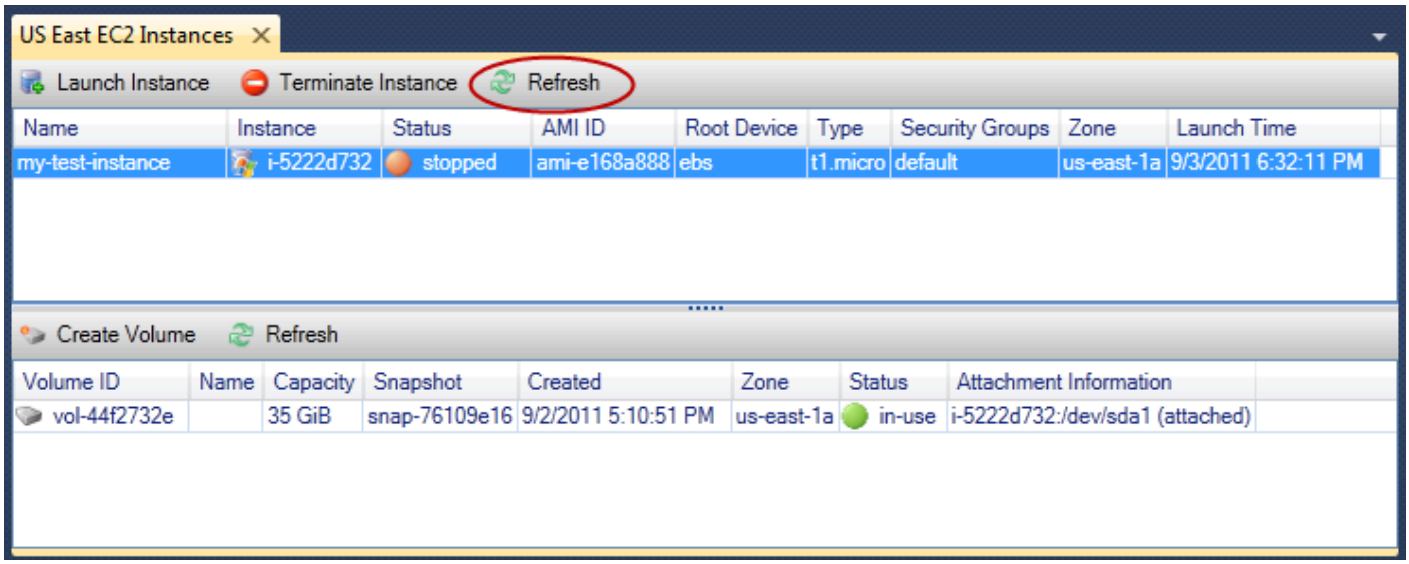
另一种结束实例的可能方式是，使用远程桌面连接到实例，然后从 Windows 开始菜单中，使用关机。您可将实例配置为在此方案中停止或终止。

停止 Amazon EC2 实例

1. InAWS资源管理器，展开Amazon EC2节点中，打开的上下文 (右键单击) 菜单实例，然后选择查看. 在 Instances (实例) 列表中，右键单击要停止的实例，然后从上下文菜单中选择 Stop (停止)。选择 Yes (是) 确认您要停止实例。

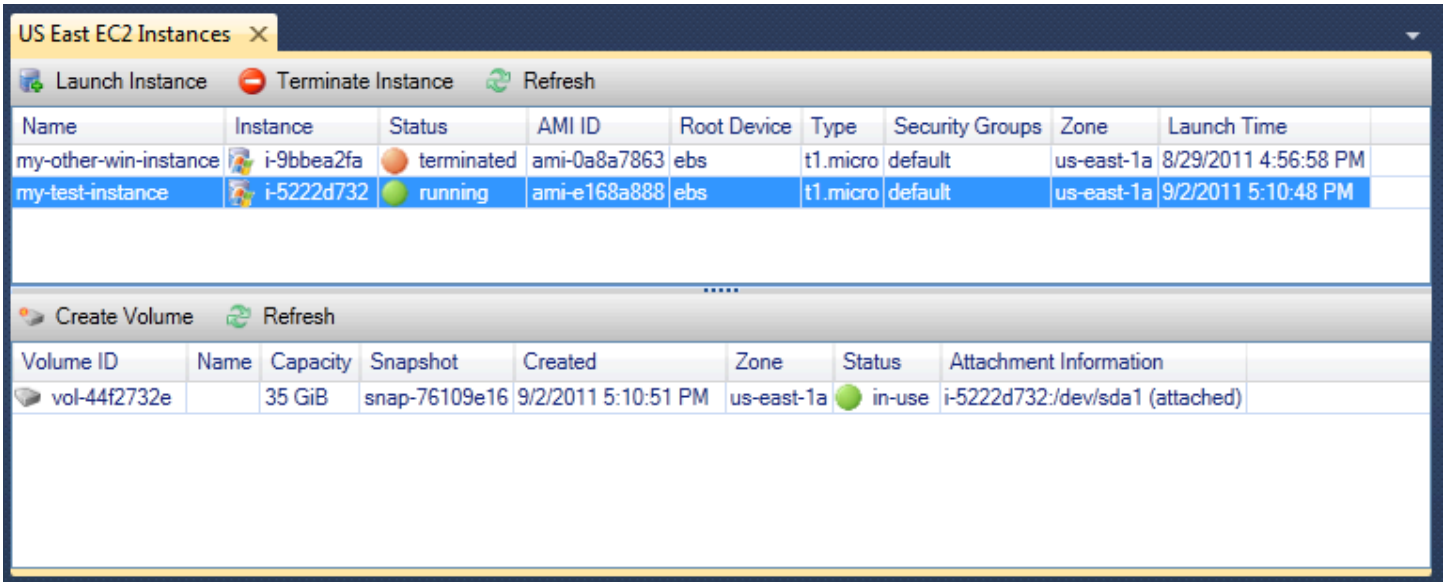


2. 在 Instances (实例) 列表的顶部，选择 Refresh (刷新) 以查看 Amazon EC2 实例状态的更改。由于我们已停止而不是终止实例，因此与实例关联的 EBS 卷仍处于活动状态。



终止的实例仍可见

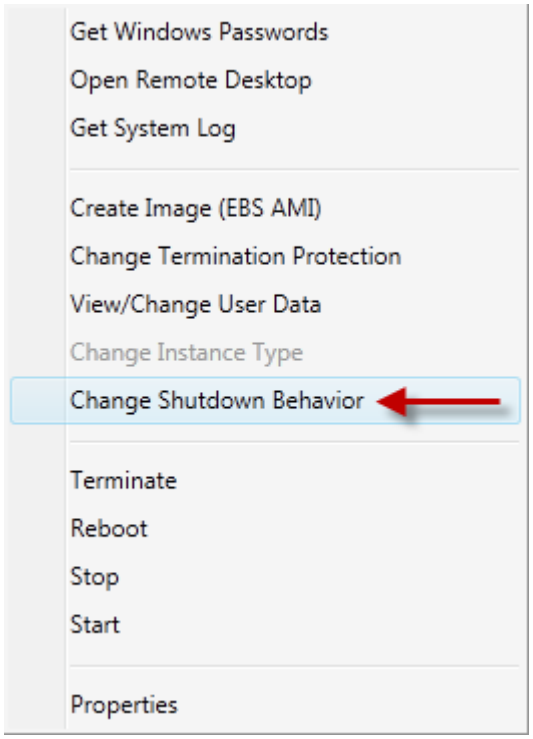
如果您终止一个实例，此实例将与正在运行的或已停止的实例一起继续显示在 Instance (实例) 列表中。最终，AWS回收这些实例，并且它们将从列表中消失。不会向您收取处于已终止状态的实例的费用。



指定 EC2 实例在关闭时的行为

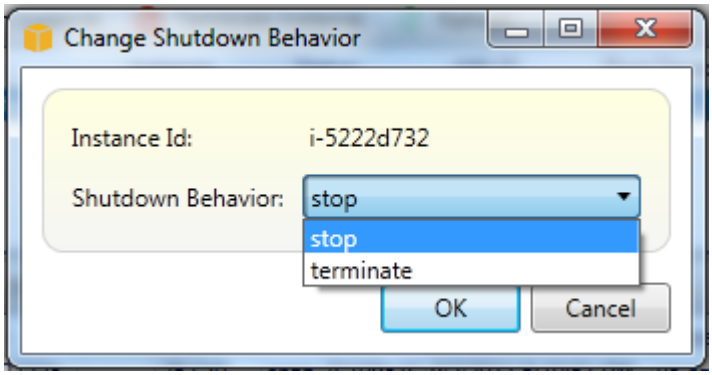
这些区域有：AWS利用 Toolkit，您可指定 Amazon EC2 实例在下列情况下是停止还是终止。Shutdown已从启动菜单。

1. 在 Instances (实例) 列表中，右键单击 Amazon EC2 实例，然后选择 Change shutdown behavior (更改关机行为)。



Change Shutdown Behavior (更改关机行为) 菜单项

2. 在 Change Shutdown Behavior (更改关机行为) 对话框中，从 Shutdown Behavior (关机行为) 下拉列表中，选择 Stop (停止) 或 Terminate (终止)。



管理 Amazon ECS 实例

AWSExplorer 提供了详细的 Amazon Elastic Container Service (Amazon ECS) 群集和容器存储库视图。您可以在 Visual Studio 开发环境中创建、删除和管理群集和容器详细信息。

修改服务属性

您可以在集群视图中查看服务详细信息、服务事件和服务属性。

1. InAWS选择 Explorer，打开要管理的集群的上下文（右键单击）菜单，然后选择。查看.
2. 在 ECS集群视图中，单击左侧的 Services (服务)，然后单击详细信息视图中的 Details (详细信息) 选项卡。您可以单击 Events (事件) 以查看事件消息，也可以单击 Deployments (部署) 以查看部署状态。
3. 单击 Edit (编辑)。您可以更改所需任务计数以及最小和最大的正常运行状态百分比。
4. 单击 Save (保存) 以接受更改，或单击 Cancel(取消) 以恢复为现有值。

停止任务

您可以查看任务的当前状态并停止群集视图中的一个或多个任务。

停止任务

1. InAWS选择 Explorer，打开含有要停止任务的集群的上下文（右键单击）菜单，然后选择。查看.
2. 在 ECS 集群视图中，单击左侧的 Tasks (任务)。
3. 确保将 Desired Task Status (所需任务状态) 设置为 `Running`。选择要停止的各个任务，然后单击 Stop (停止) 或单击 Stop All (全部停止) 以选择和停止所有正在运行的任务。
4. 在 Stop Tasks (停止任务) 对话框中选择Yes (是)。

删除服务

您可以从群集视图中删除群集中的服务。

删除集群服务

1. InAWS选择 Explorer，打开含有要删除服务的集群的上下文（右键单击）菜单，然后选择。查看.
2. 在 ECS 集群视图中，单击左侧的 Services (服务)，然后单击 Delete (删除)。
3. 在 Delete Cluster (删除集群) 对话框中，如果您的集群中有负载均衡器和目标组，您可以选择将它们与集群一同删除。删除该服务后，将不会再使用它们。
4. 在 Delete Cluster (删除集群) 对话框中，选择 OK (确定)。群集被删除后，也将从AWSExplorer。

删除集群

您可以从中删除 Amazon Elastic Container Service 集群AWSExplorer。

删除集群

1. InAWS在 Explorer 中，打开要删除的集群的上下文（右键单击）菜单。集群的节点Amazon ECS选择，然后选择Delete.
2. 在 Delete Cluster (删除集群) 对话框中，选择 OK (确定)。群集被删除后，也将从AWSExplorer。

创建存储库

您可以从以下位置创建 Amazon Elastic 容器注册库AWSExplorer。

创建存储库

1. InAWS打开 Explorer 中的上下文（右键单击）菜单。存储库节点下Amazon ECS选择，然后选择创建存储库.
2. 在 Create Repository (创建存储库) 对话框中，为存储库命名，然后选择 OK (确定)。

删除存储库

您可以从中删除 Amazon Elastic Container Registry 存储库AWSExplorer。

删除存储库

1. InAWS打开 Explorer 中的上下文（右键单击）菜单。存储库节点下Amazon ECS选择，然后选择删除存储库.
2. 在 Delete Repository (删除存储库) 对话框中，您可以选择删除存储库（即使其中包含映像）。否则，它只有为空时才会被删除。单击 Yes (是)。

从管理安全组AWS探险者

通过 Toolkit for Visual Studio，您可以创建和配置安全组以与 Amazon Elastic Compute Cloud (Amazon EC2) 实例配合使用，以及AWS CloudFormation. 当您启动 Amazon EC2 实例或将应用程序部署到时AWS CloudFormation，您可指定要与 Amazon EC2 实例关联的安全组。（将部署到AWS CloudFormation创建 Amazon EC2 实例。）

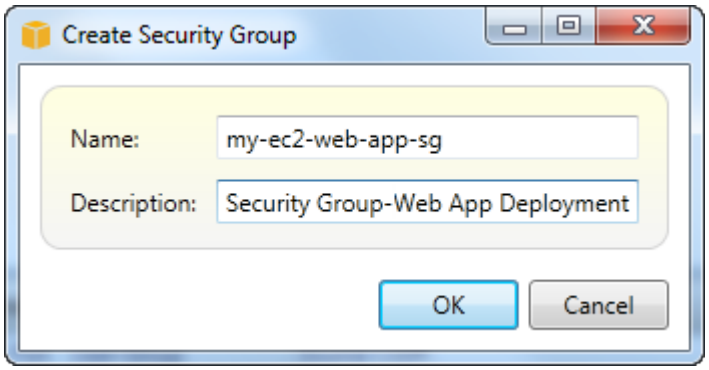
安全组的行为类似于传入网络流量的防火墙。安全组指定 Amazon EC2 实例上允许的网络流量的类型。它还指定仅接受来自特定 IP 地址或指定用户或其他安全组的传入流量。

正在创建安全组

在此部分中，我们将创建一个安全组。创建安全组后，安全组将未配置任何权限。权限配置是通过其他操作来处理的。

创建安全组

1. InAWS资源管理器，在Amazon EC2节点中，打开上下文（右键单击）菜单个安全组节点，然后选择查看.
2. 在 EC2 Security Groups (EC2 安全组) 选项卡上，选择 Create Security Group (创建安全组)。
3. 在 Create Security Group (创建安全组) 对话框中，键入该安全组的名称和描述，然后选择 OK (确定)。

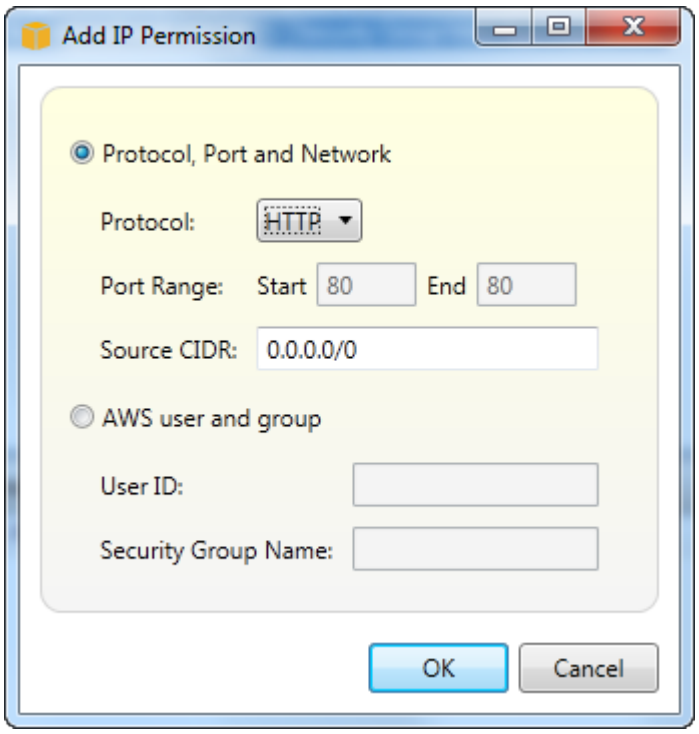


向安全组添加权限

在此部分中，我们将向安全组添加权限以通过 HTTP 和 HTTPS 协议允许 Web 流量。我们还将允许其他计算机通过使用 Windows 远程桌面协议 (RDP) 进行连接。

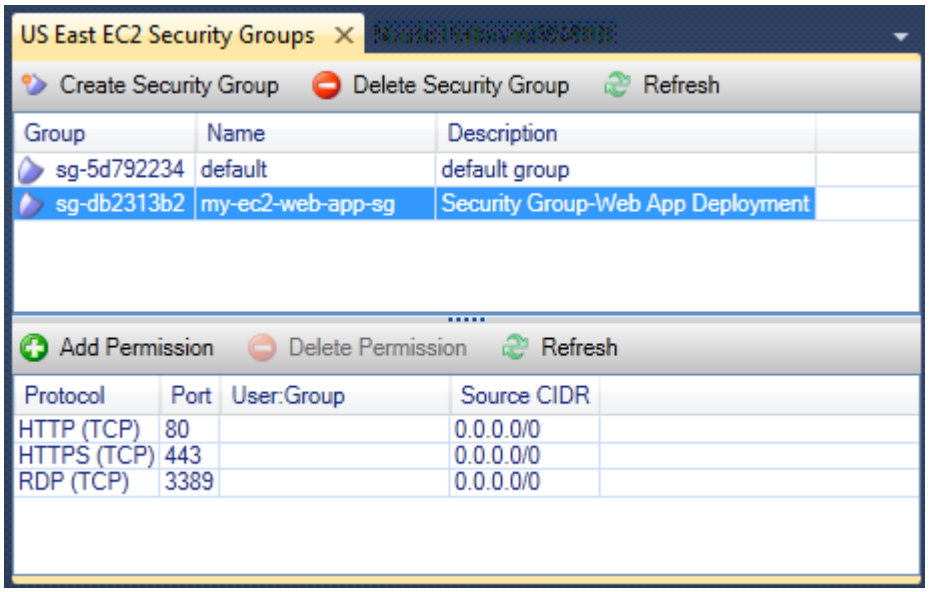
向安全组添加权限

1. 在 EC2 Security Groups (EC2 安全组) 选项卡上，选择安全组，然后选择 Add Permission (添加权限) 按钮。
2. 在 Add IP Permission (添加 IP 权限) 对话框中，选择 Protocol, Port and Network (协议、端口和网络) 单选按钮，然后从 Protocol (协议) 下拉列表中，选择 HTTP。端口范围将自动调整为端口 80（HTTP 的默认端口）。Source CIDR (源 CIDR) 字段默认为 0.0.0.0/0，这指定将接受来自任何外部 IP 地址的 HTTP 网络流量。选择 OK（确定）。



为此安全组打开端口 80 (HTTP)

3. 对 HTTPS 和 RDP 重复此过程。您的安全组权限现在看起来应与下面内容类似。



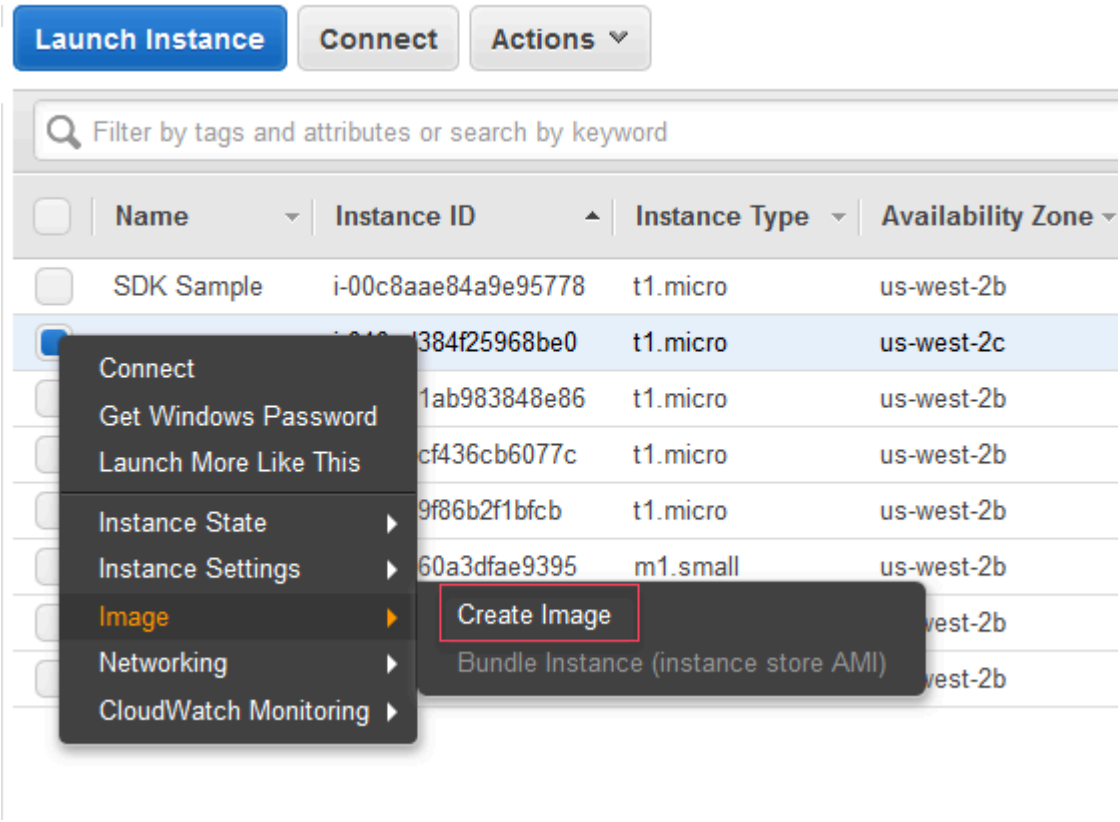
您还可通过指定用户 ID 和安全组名称在安全组中设置权限。在此情况下，此安全组中的 Amazon EC2 实例将接受来自指定安全组中的 Amazon EC2 实例的所有传入网络流量。您还必须指定用户 ID 作为区分安全组名称的方法；安全组名称在所有AWS. 有关安全组的更多信息，请转到 EC2 文档。

从 Amazon EC2 实例创建 AMI

通过 Amazon EC2 Instances (Amazon EC2 实例) 视图，您可以从正在运行的或已停止的实例中创建 Amazon 系统映像 (AMI)。有关 AMI 的更多详细信息，请参阅适用于 Windows 实例的 Amazon Elastic Compute Cloud 用户指南中的 Amazon 机器映像 (AMI) 主题。

从实例创建 AMI

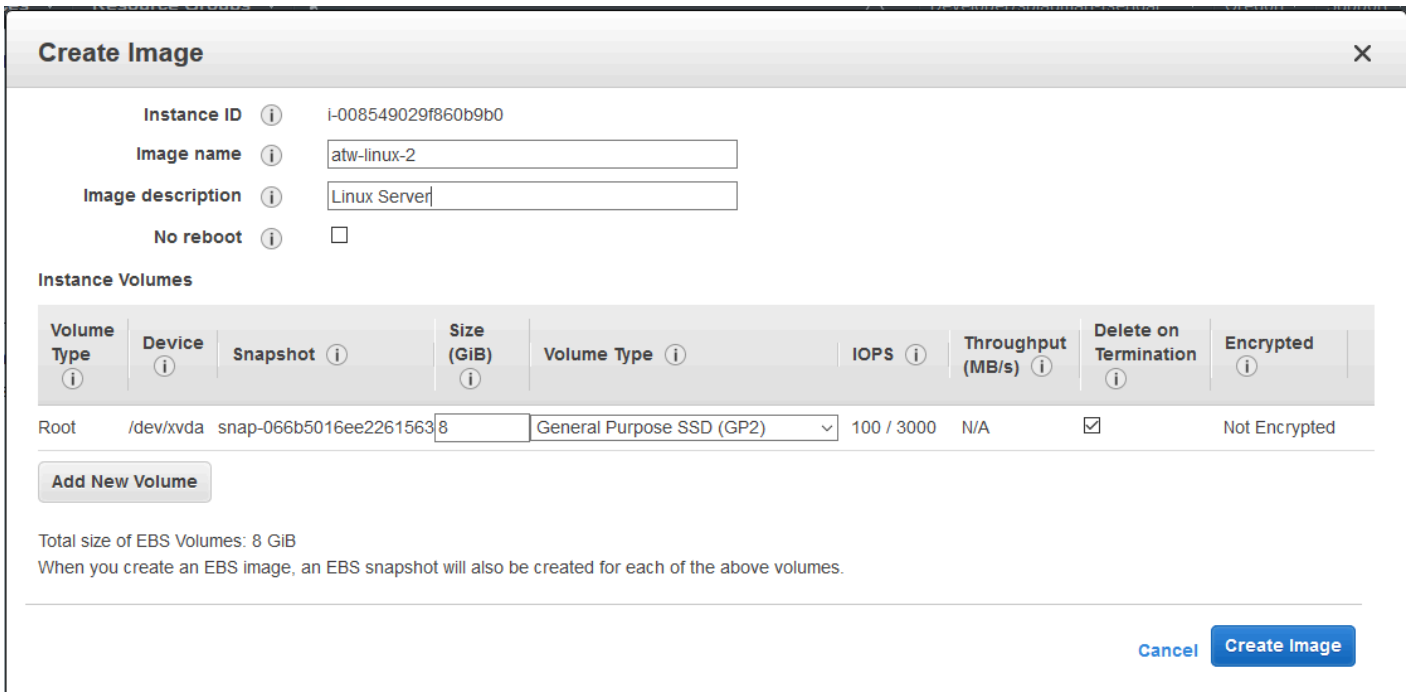
1. 右键单击要用作 AMI 的基础的实例，然后从上下文菜单中选择 Create Image (创建映像)。



Create Image (创建映像) 上下文菜单

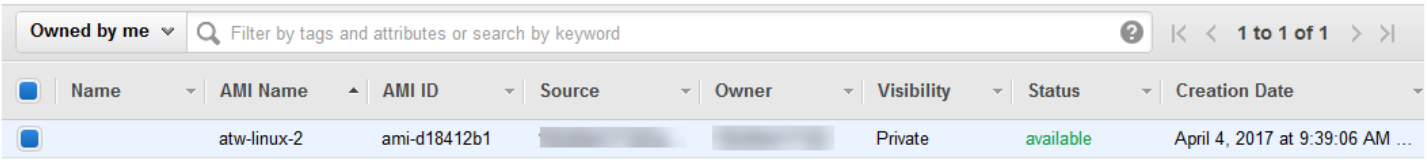
2. 在 Create Image (创建映像) 对话框中，键入唯一的名称和描述，然后选择 Create Image (创建映像)。默认情况下，Amazon EC2 将关闭实例，为附加的任意卷制作快照，创建和注册 AMI，然后重新启动实例。如果不希望关闭实例，请选择 No，请选择 No，请选择 No。

> **Warning**
>
> 如果您选择 No reboot 选项，则我们无法保证所创建映像的文件系统完整性。



Create Image (创建映像) 对话框

创建 AMI 可能需要几分钟。创建后，它将出现在AWS资源管理器的 AMI 视图中。要显示此视图，请在AWS资源管理器中双击 Amazon EC2 | AMI 节点。要查看您的 AMI，请从 Viewing (查看) 下拉列表中，选择 Owned By Me (由我拥有)。您可能需要选择 Refresh (刷新) 以查看您的 AMI。当 AMI 首次显示时，它可能处于挂起状态，但过一段时间后，它将转变为可用状态。



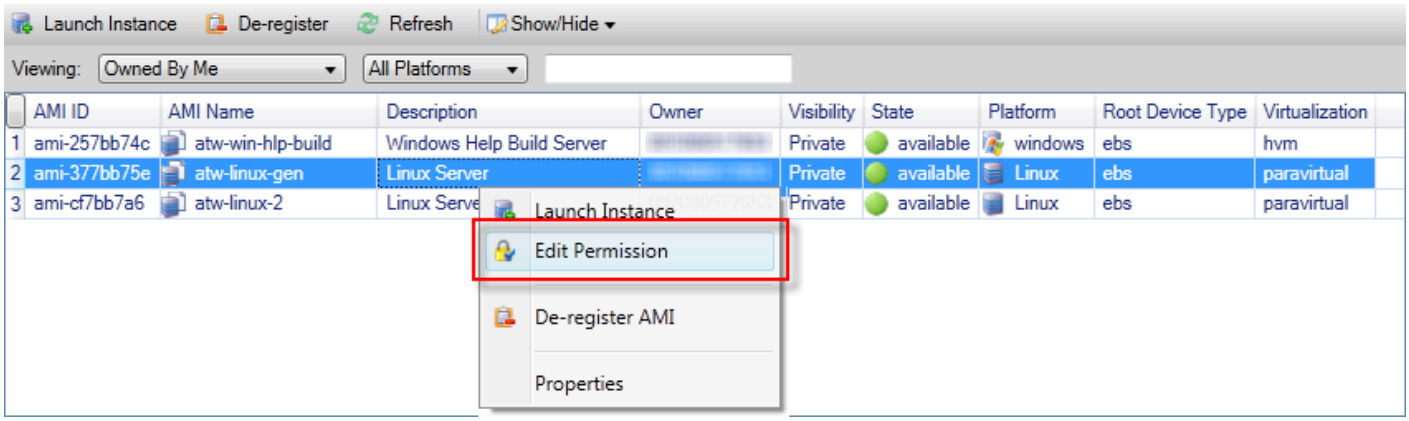
已创建的 AMI 的列表

在 Amazon 系统映像上设置启动许可

您可以通过AMI在中查看AWSExplorer。您可以使用 Set AMI Permissions (设置 AMI 权限) 对话框复制来自 AMI 的权限。

设置 AMI 上的权限

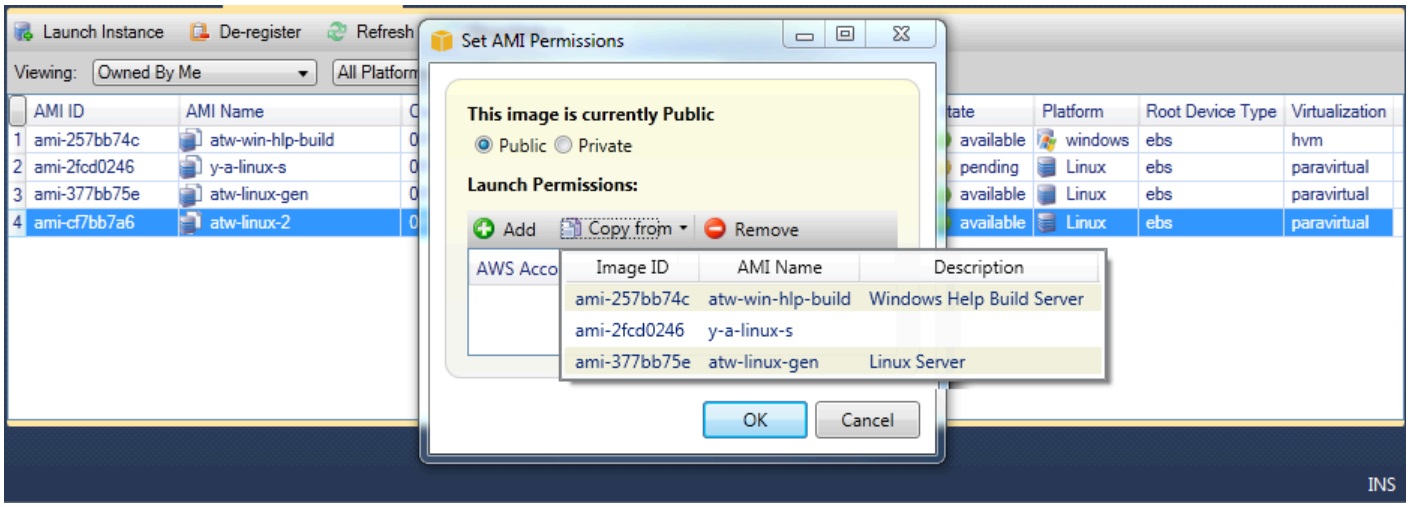
1. 在AMI在中查看AWS在 AMI 上打开上下文菜单 (右键单击)，然后选择编辑权限.



2. Set AMI Permissions (设置 AMI 权限) 对话框中有三个选项：

   - 要授予启动权限，请选择Add，然后键入账号的AWS您向其授予启动权限的用户。
   - 要删除启动许可，请选择AWS要为其删除启动许可的用户，然后选择Remove.
   - 要将许可从一个 AMI 复制到另一个 AMI，请从列表中选择一个 AMI，然后选择 Copy from (复制自)。具有您选择的 AMI 上的启动许可的用户将获得当前 AMI 上的启动许可。您可以利用 Copy-from (复制自) 列表中的其他 AMI 重复此流程，以便将许可从多个 AMI 复制到目标 AMI。

     这些区域有：执行 COPY 的操作列表仅包含由账户拥有的 AMI，该 AMI 在执行AMI视图显示自 AWSExplorer。因此，如果该活动账户不拥有任何其他 AMI，Copy-from (复制自) 列表可能不会显示任何 AMI。



Copy AMI permissions (复制 AMI 权限) 对话框

Amazon Virtual Private Cloud (VPC)

通过 Amazon Virtual Private Cloud (Amazon VPC)，您可以将 Amazon Web Services 资源启动到您定义的虚拟网络中。这个虚拟网络类似于您在数据中心中运行的传统网络，并具有使用可扩展基础设施的优势。AWS. 有关更多信息，请转到 Amazon VPC 用户指南。

利用适用于 Toolkit for Visual Studio，开发人员可以访问 VPC 功能，该功能类似于由AWS Management Console但来自 Visual Studio 开发环境。这些区域有：Amazon VPC节点AWSExplorer包含以下区域的子节点。

- VPC
- Subnets (子网)
- 弹性 IP
- Internet 网关
- 网络 ACL
- 路由表
- 安全组

使用创建公有-私有 VPC 以供部署AWS Elastic Beanstalk

本节介绍如何创建同时包含公有子网和私有子网的 Amazon VPC。公有子网包含一个 Amazon EC2 实例，该实例将执行网络地址转换 (NAT) 以支持私有子网中的实例与公有 Internet 通信。两个子网必须位于同一可用区 (AZ)。

这是在 VPC 中部署 AWS Elastic Beanstalk 环境所需的最低 VPC 配置。在这种情况下，托管您的应用程序的 Amazon EC2 实例位于私有子网中；将传入流量路由到您的应用程序的 Elastic Load Balancing 负载均衡器位于公有子网内。

有关网络地址转换 (NAT) 的更多信息，请转至NAT 实例中的Amazon Virtual Private Cloud 用户指南. 有关如何将部署配置为使用 VPC 的示例，请参阅部署到 Elastic Beanstalk。

创建公有-私有子网 VPC

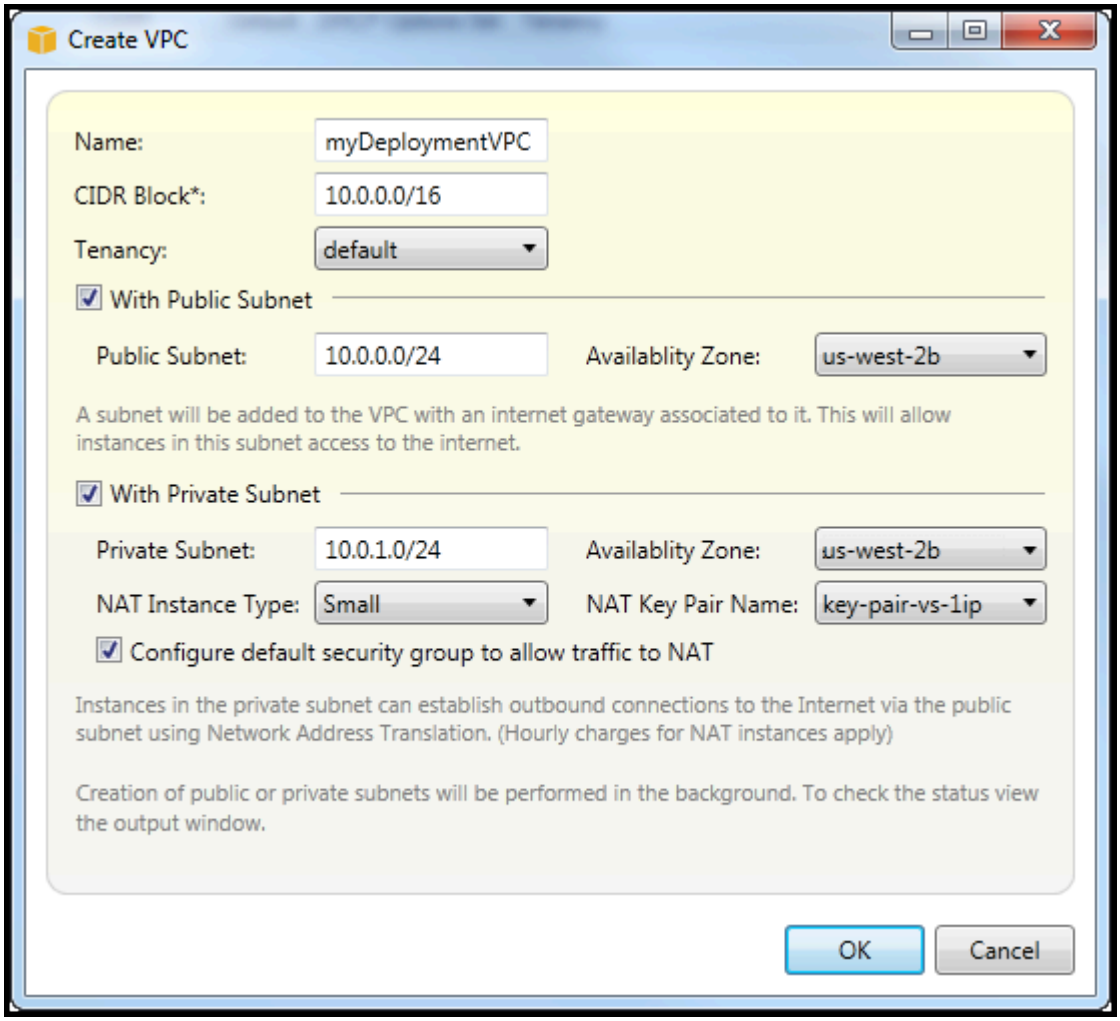
1. 在Amazon VPC在节点AWS资源管理器，打开VPC选择子节点，然后选择创建 VPC.

2. 按下面所示配置 VPC：

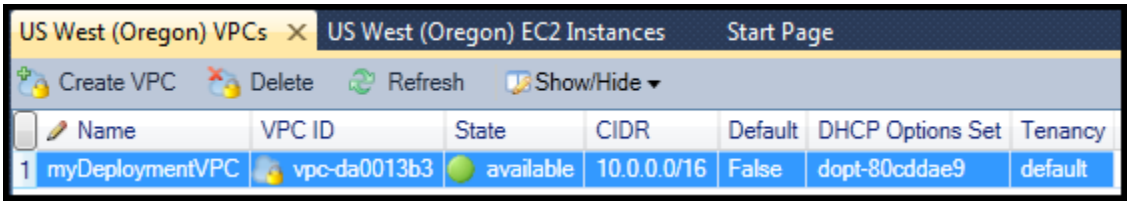
   - 键入 VPC 的名称。
   - 选中 With Public Subnet (使用公有子网) 和 With Private Subnet (使用私有子网) 复选框。
   - 从每个子网的 Availability Zone (可用区) 下拉列表框中，选择一个可用区。确保对两个子网使用相同的 AZ。
   - 对于 NAT Key Pair Name (NAT 密钥对名称) 中的私有子网，提供一个键前缀。此 key pair 用于执行从私有子网到公有 Internet 的网络地址转换的 Amazon EC2 实例。
   - 选中 Configure default security group to allow traffic to NAT (配置默认安全组以允许到 NAT 的流量) 复选框。

   键入 VPC 的名称。选中 With Public Subnet (使用公有子网) 和 With Private Subnet (使用私有子网) 复选框。从每个子网的 Availability Zone (可用区) 下拉列表框中，选择一个可用区。确保对两个子网使用相同的 AZ。对于 NAT Key Pair Name (NAT 密钥对名称) 中的私有子网，提供一个键前缀。此 key pair 用于执行从私有子网到公有 Internet 的网络地址转换的 Amazon EC2 实例。选中 Configure default security group to allow traffic to NAT (配置默认安全组以允许到 NAT 的流量) 复选框。

   选择 OK（确定）。

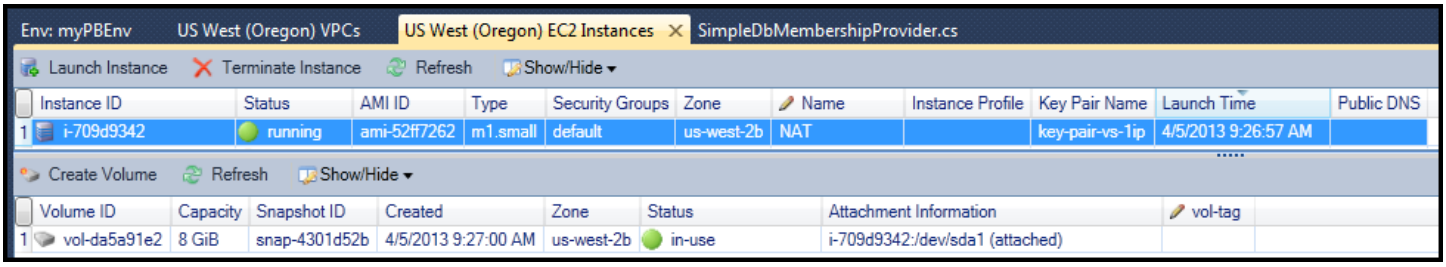


您可 VPC 在VPC选项卡AWSExplorer。



启动 NAT 实例可能需要几分钟。当该实例可用时，您可以通过展开Amazon EC2在节点AWS资源管理器然后打开实例子节点。

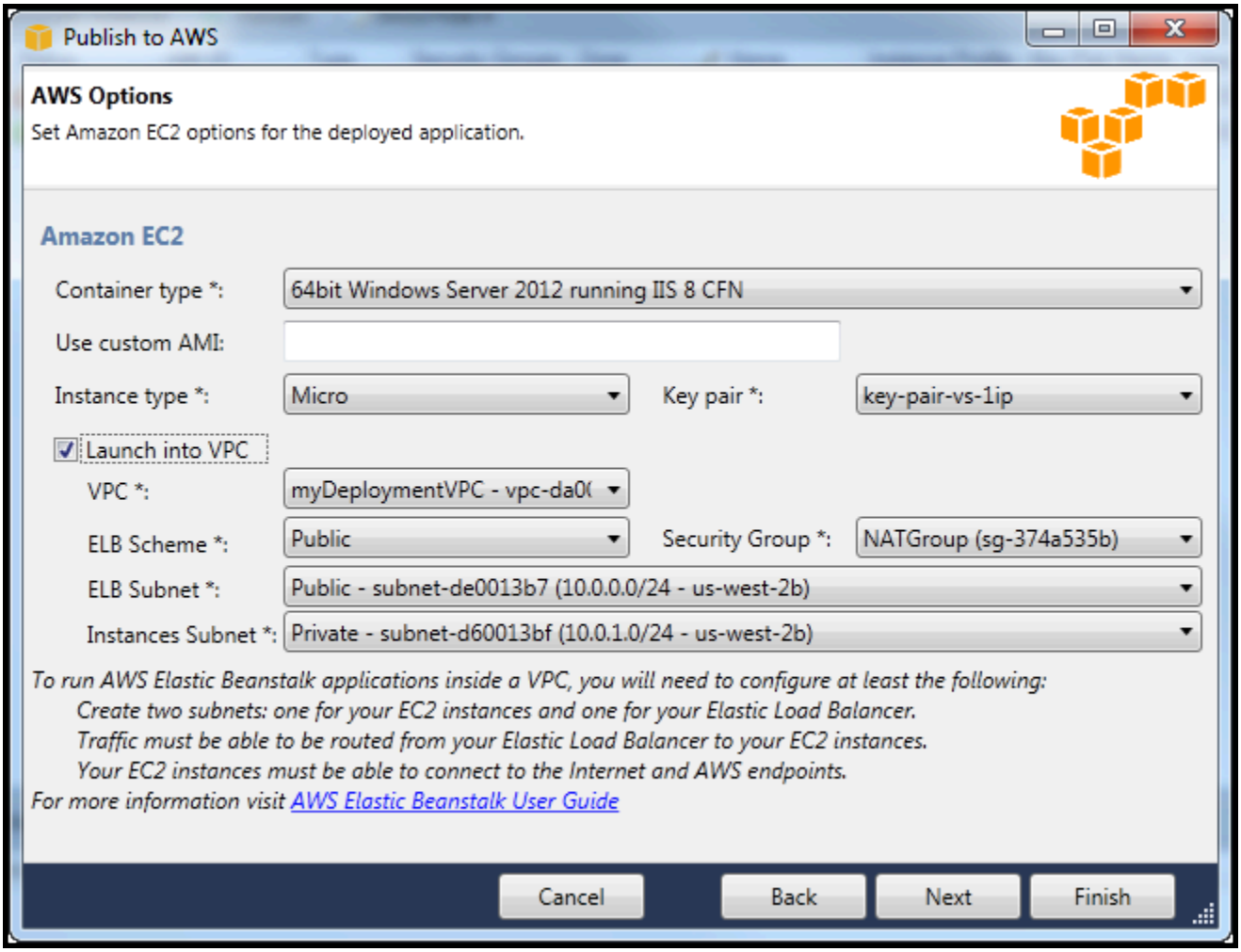
网络 ACL 和安全组都允许 (因此可到达您的实例) 的发起 ping 的AWS Elastic Beanstalk将为 NAT 实例自动创建 (Amazon EBS) 卷。有关 Elastic Beanstalk 的更多信息，请转至AWS Elastic Beanstalk(EBS)中的适用于 Linux 实例的 Amazon EC2 用户指南.



如果您[将应用程序部署到AWS Elastic Beanstalk环境]然后选择在 VPC 中启动环境，工具包将填充Publish to (发布到 CloudWatch)Amazon Web Services对话框，其中包含您的 VPC 的配置信息。

Toolkit 将使用仅来自在 Toolkit 中创建 VPC 的信息填充对话框，而不是来自使用AWS Management Console. 这是因为，当 Toolkit 创建 VPC 时，它将对 VPC 的组件进行标记，以便能够访问这些组件的信息。

来自部署向导的以下屏幕截图显示了一个对话框的示例，该对话框是使用来自在 Toolkit 中创建的 VPC 的值填充的。



删除 VPC

要删除 VPC，您必须先终止其中的任何 Amazon EC2 实例。

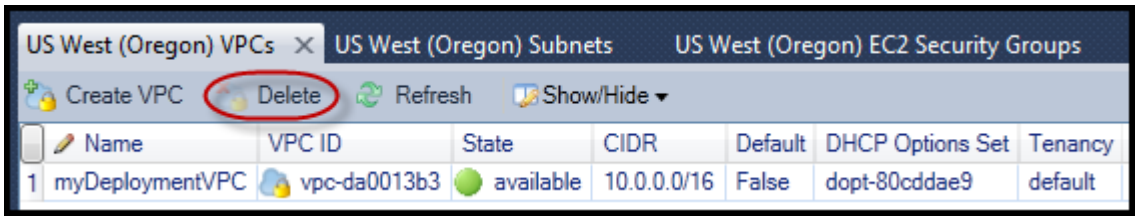
1. 如果您已将应用程序部署到 VPC 中的 AWS Elastic Beanstalk 环境，则删除该环境。这将终止托管您的应用程序的所有 Amazon EC2 实例以及 Elastic 负载均衡器。

   如果您尝试直接终止托管您的应用程序的实例而不删除该环境，Auto Scaling 服务将自动创建新实例来替换删除的实例。有关更多信息，请访问 Auto Scaling 开发人员指南。

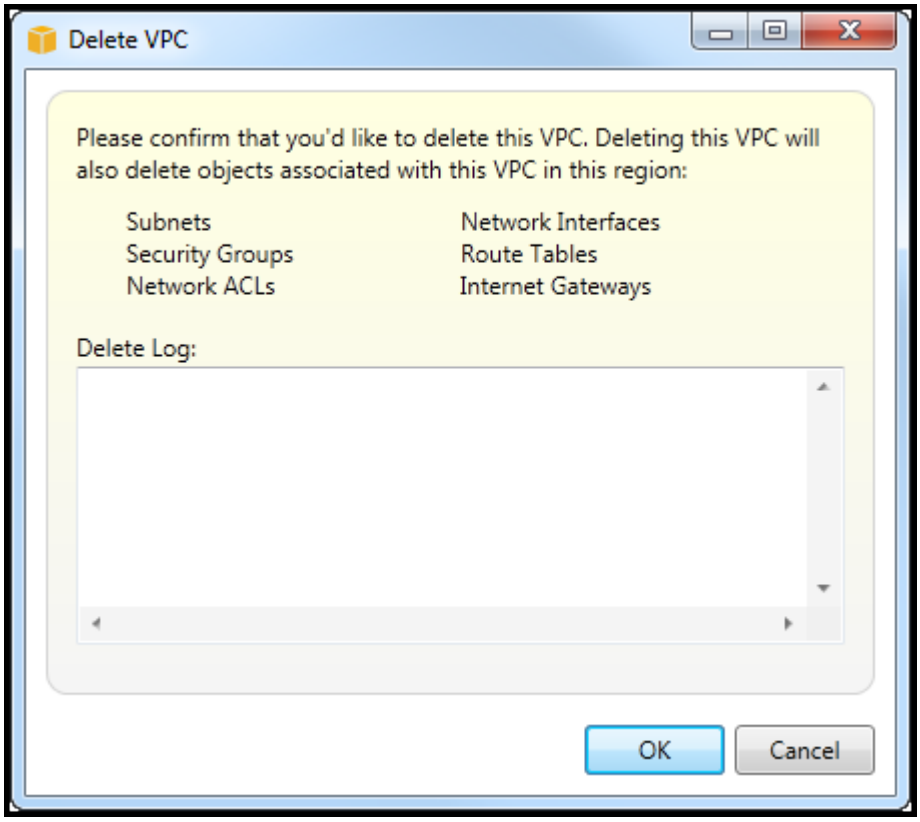
2. 删除 VPC 的 NAT 实例。

   您无需删除与 NAT 实例关联的 Amazon EBS 卷即可删除 VPC。但是，如果您不删除卷，则会继续为其付费，即使您删除了 NAT 实例和 VPC 也是如此。

3. 在 VPC 选项卡上，选择 Delete (删除) 链接以删除 VPC。



4. 在 Delete VPC (删除 VPC) 对话框中，选择 OK (确定)。



使用 Visual Studio 的 AWS CloudFormation 模板编辑器

适用于 Visual Studio 的 Toolkit 包括 AWS CloudFormation 模板编辑器和适用于 Visual Studio 的 AWS CloudFormation 模板项目。支持的功能包括：

- 使用提供的模板项目类型创建新模板（空模板或从现有堆栈或示例 AWS CloudFormation 模板中复制）。
- 利用自动 JSON 验证、自动完成、代码折叠和语法突出显示来编辑模板。
- 模板中字段值的内部函数和资源参考参数的自动建议。
- 用于在 Visual Studio 中对模板执行常见操作的菜单项。

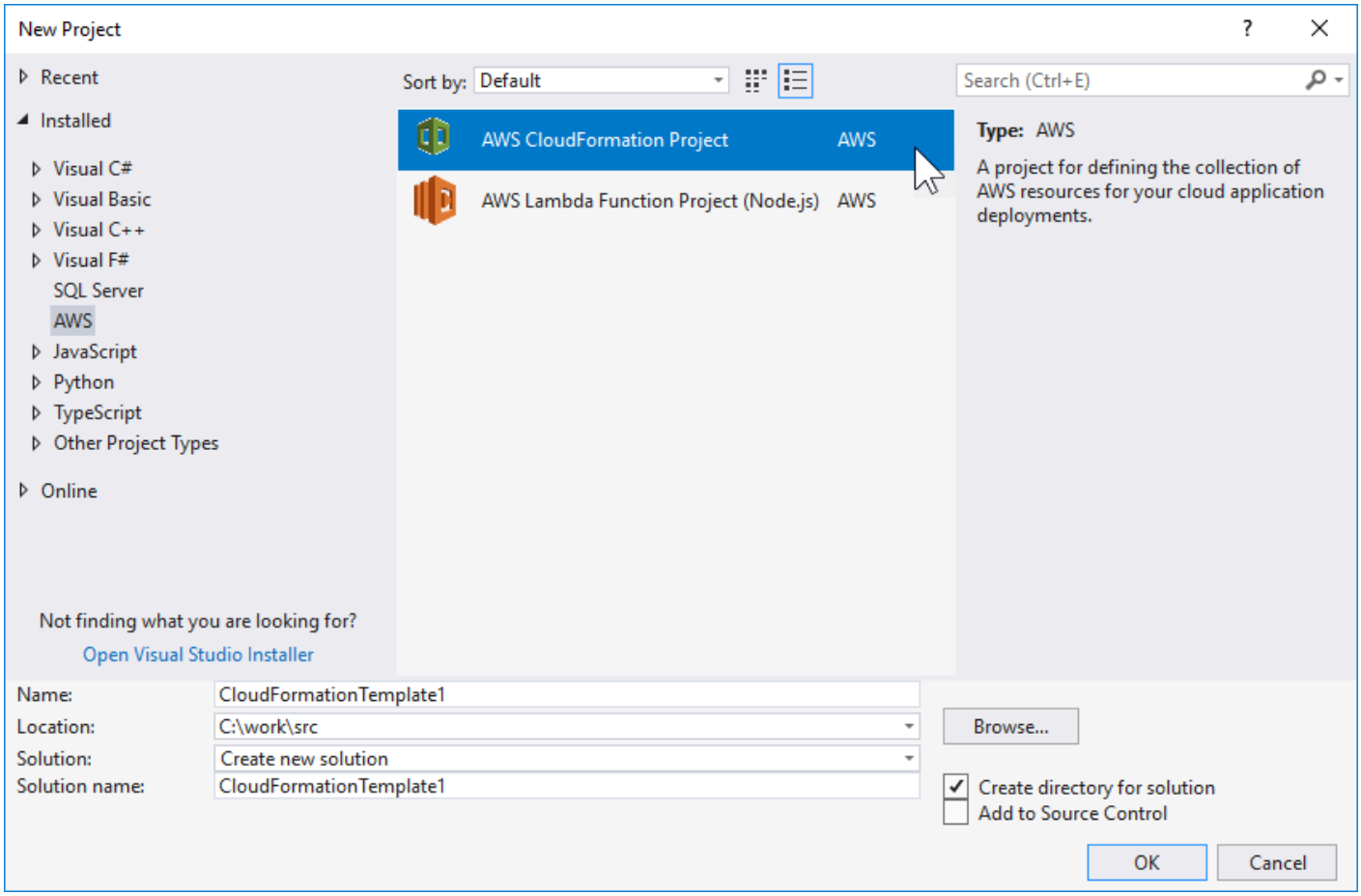
主题

在 Visual Studio 中创建 AWS CloudFormation 模板项目

创建模板项目

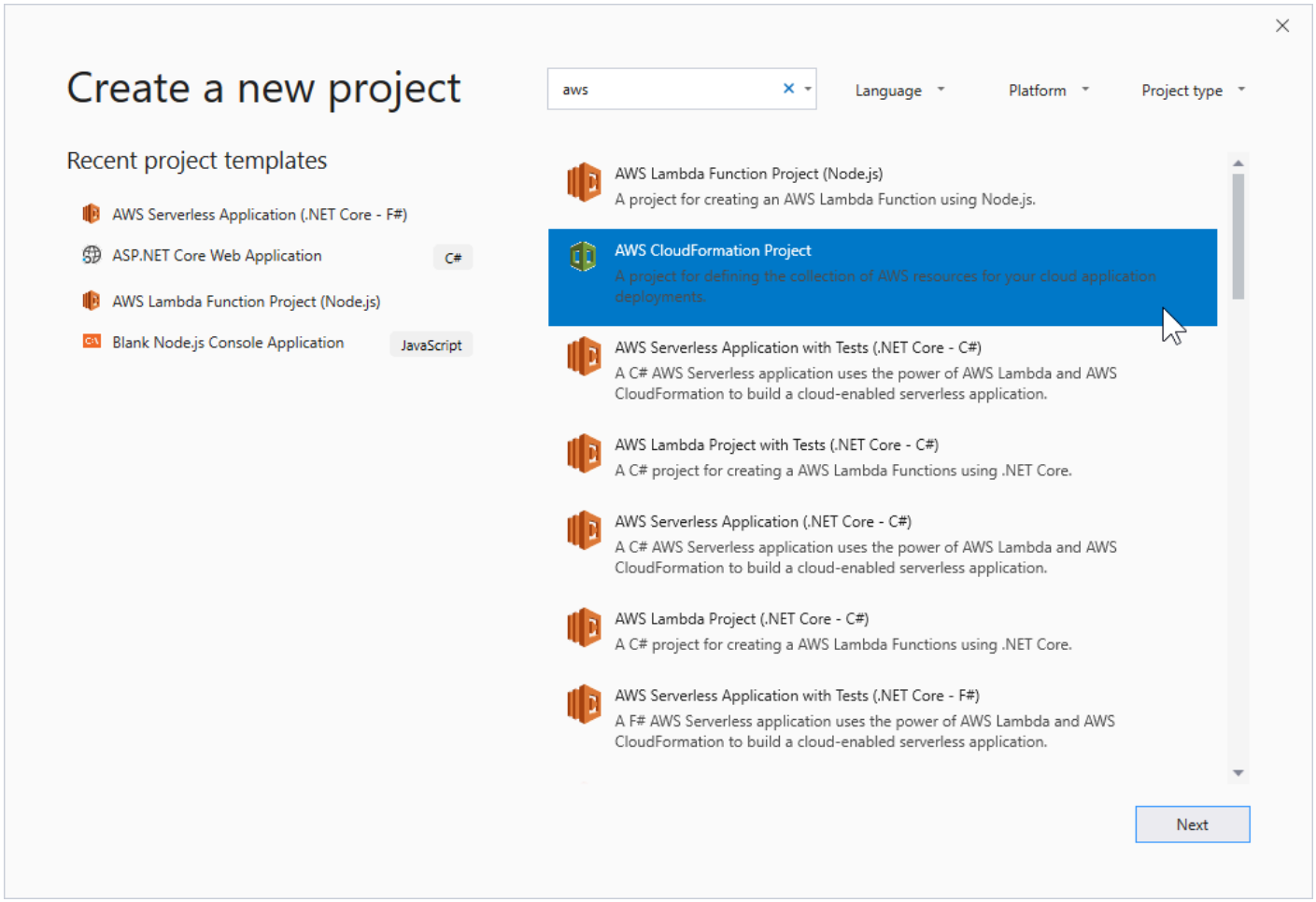
1. 在 Visual Studio 中，依次选择 File (文件)、New (新建) 和 Project (项目)。
2. 对于 Visual Studio 2017：

   在新项目“对话框” 中，展开已安装然后选择AWS.



对于 Visual Studio 2019：

在新项目对话框中，确保语言、平台, 和项目类型下拉框设置为 “All...” 并键入aws中的搜索字段中返回的子位置类型。

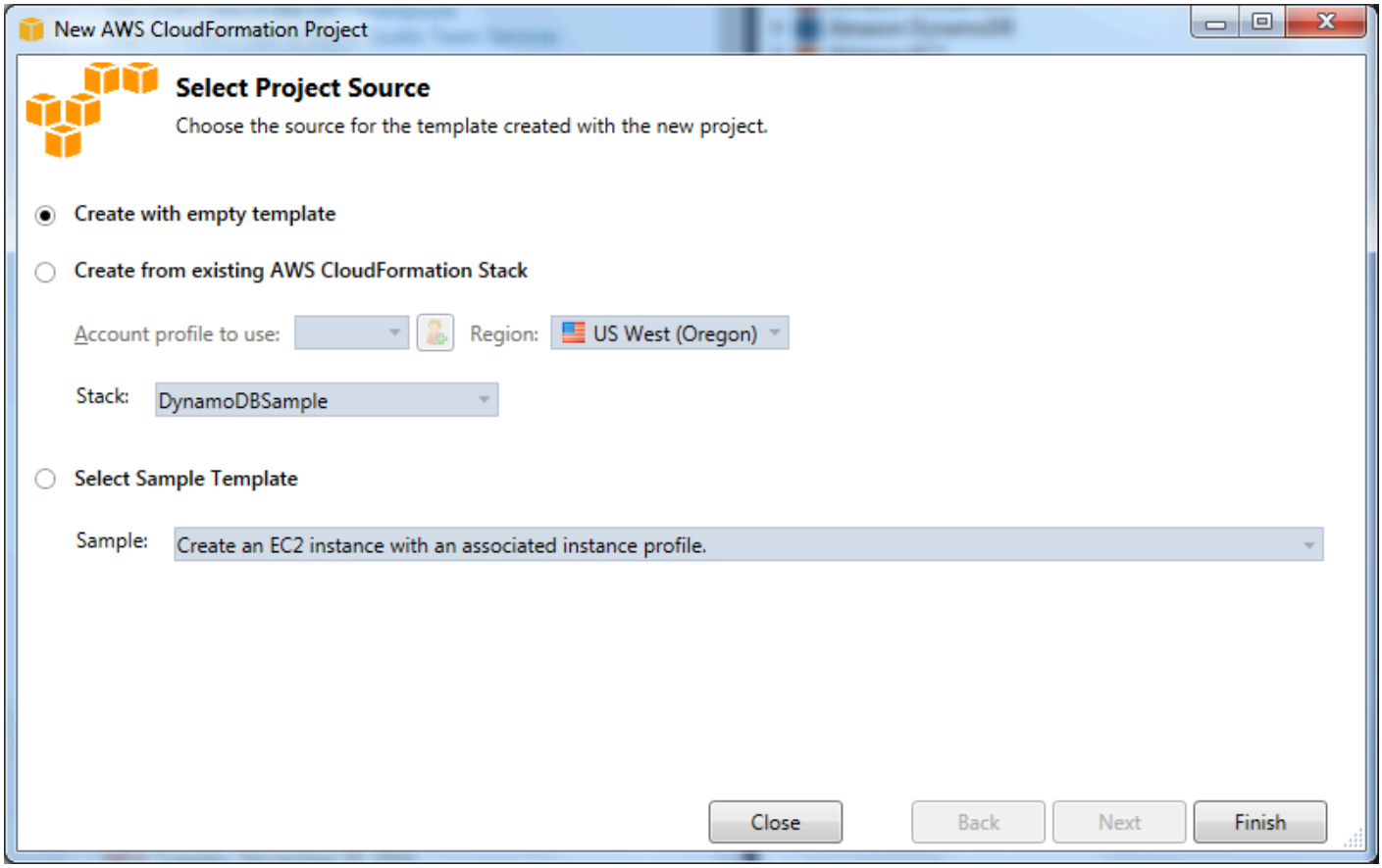


3. SelectAWSCloudFormation 项目模板化。
4. 对于 Visual Studio 2017：

   为您的模板项目输入所需的名称、位置等，然后单击确定。

   对于 Visual Studio 2019：

   单击 Next（下一步）。在下一个对话框中，为您的模板项目输入所需的名称、位置等，然后单击创建。
5. 在 Select Project Source (选择项目源) 页面上，选择您将创建的模板的源：
   - Create with empty template (使用空模板创建) 生成新的空 AWS CloudFormation 模板。
   - 从现有创建AWS|CFN| 堆栈根据您的现有堆栈生成模板AWSaccount.（该堆栈不需要具有状态 `CREATE_COMPLETE`。）
   - Select sample template (选择示例模板) 从 AWS CloudFormation 示例模板之一生成模板。

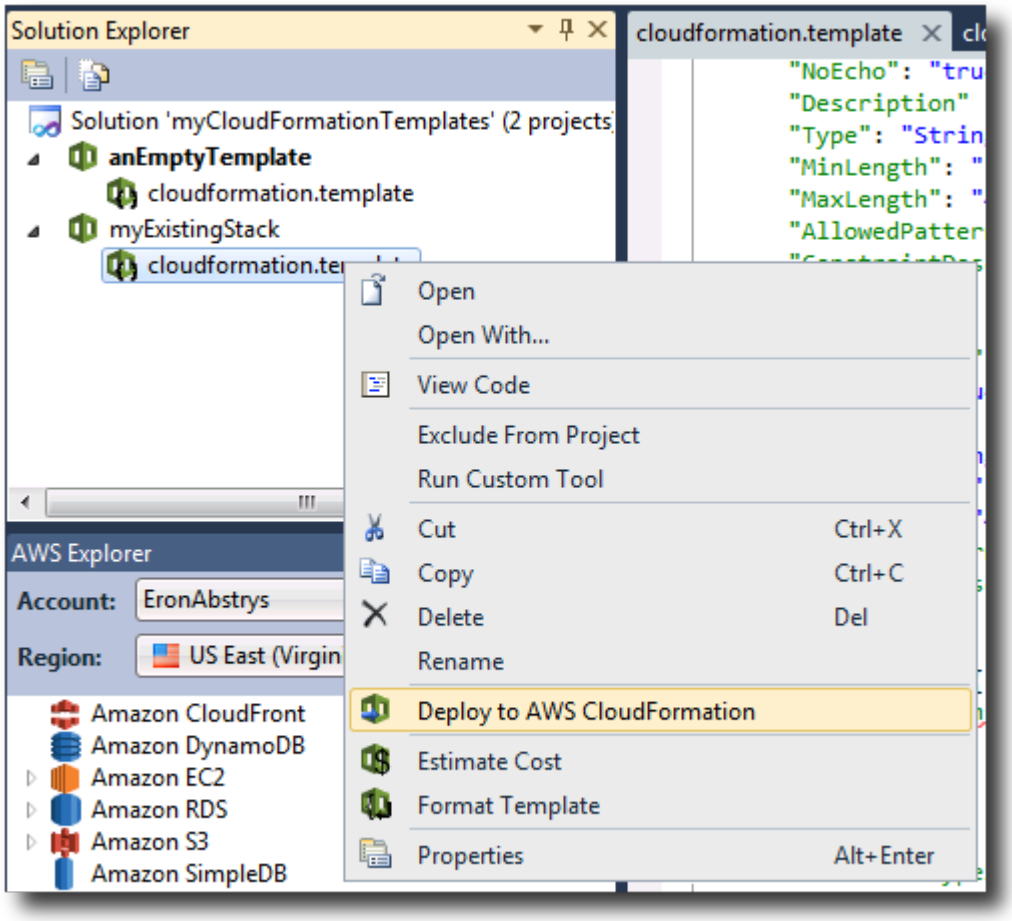


6. 要完成 AWS CloudFormation 模板项目的创建，请选择 Finish (完成)。

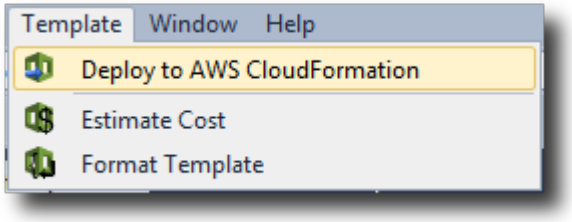
在 Visual Studio 中部署 AWS CloudFormation 模板

部署 CFN 模板

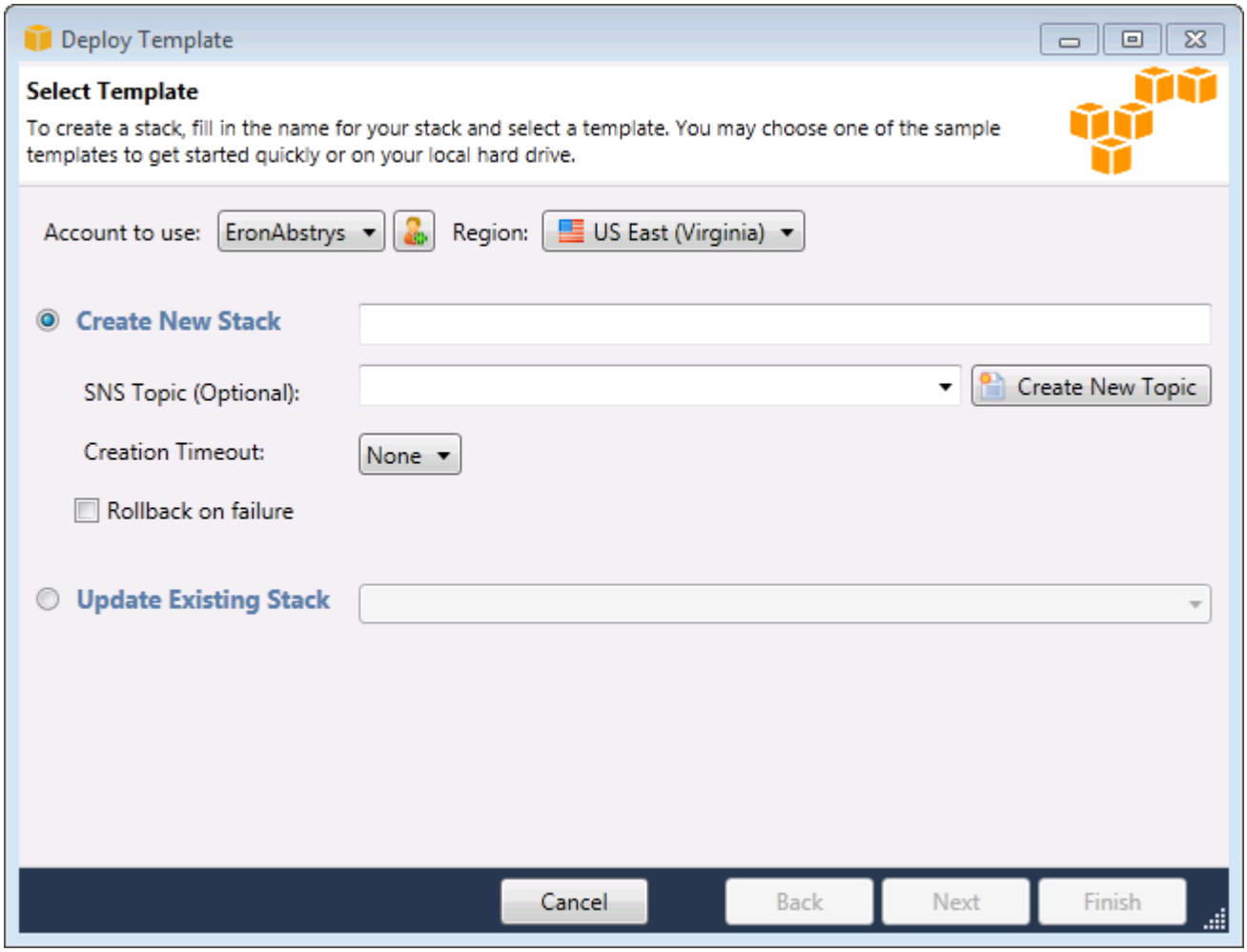
1. 在解决方案资源管理器中，打开要部署的模板的上下文（右键单击）菜单，然后选择部署到AWS CloudFormation.



或者，要部署您当前正在编辑的模板，请从模板在菜单中，选择部署到AWS CloudFormation.



2. 在存储库的部署模板在页面上，选择AWS 账户用于启动堆栈以及将在其中启动该区域。

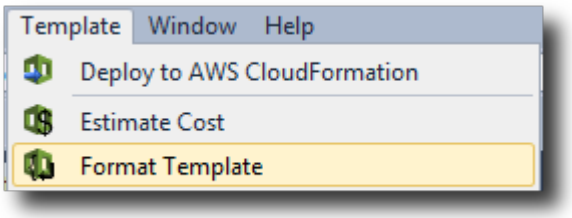


3. 选择 Create New Stack (创建新堆栈) 并为您的堆栈键入名称。
4. 选择以下任一选项（或不选择任何选项）：
   - 要接收有关堆栈的进度的通知，请从 SNS Topic (SNS 主题) 下拉列表中，选择 SNS 主题。您还可以通过选择 Create New Topic (创建新主题) 并在框中键入电子邮件地址来创建 SNS 主题。
   - 使用 Creation Timeout 指定在堆栈被声明失败 (并回滚，除非已清除 AWS CloudFormationRollback on failure 选项) 之前 应留出的创建堆栈的时间量。
   - 如果您希望堆栈在失败时回滚（即，自行删除），请使用 Rollback on failure (失败时回滚)。如果您出于调试目的希望堆栈保持活动状态，请将此选项保持清除状态，即使堆栈未能完成启动。
5. 选择 Finish (完成) 以启动堆栈。

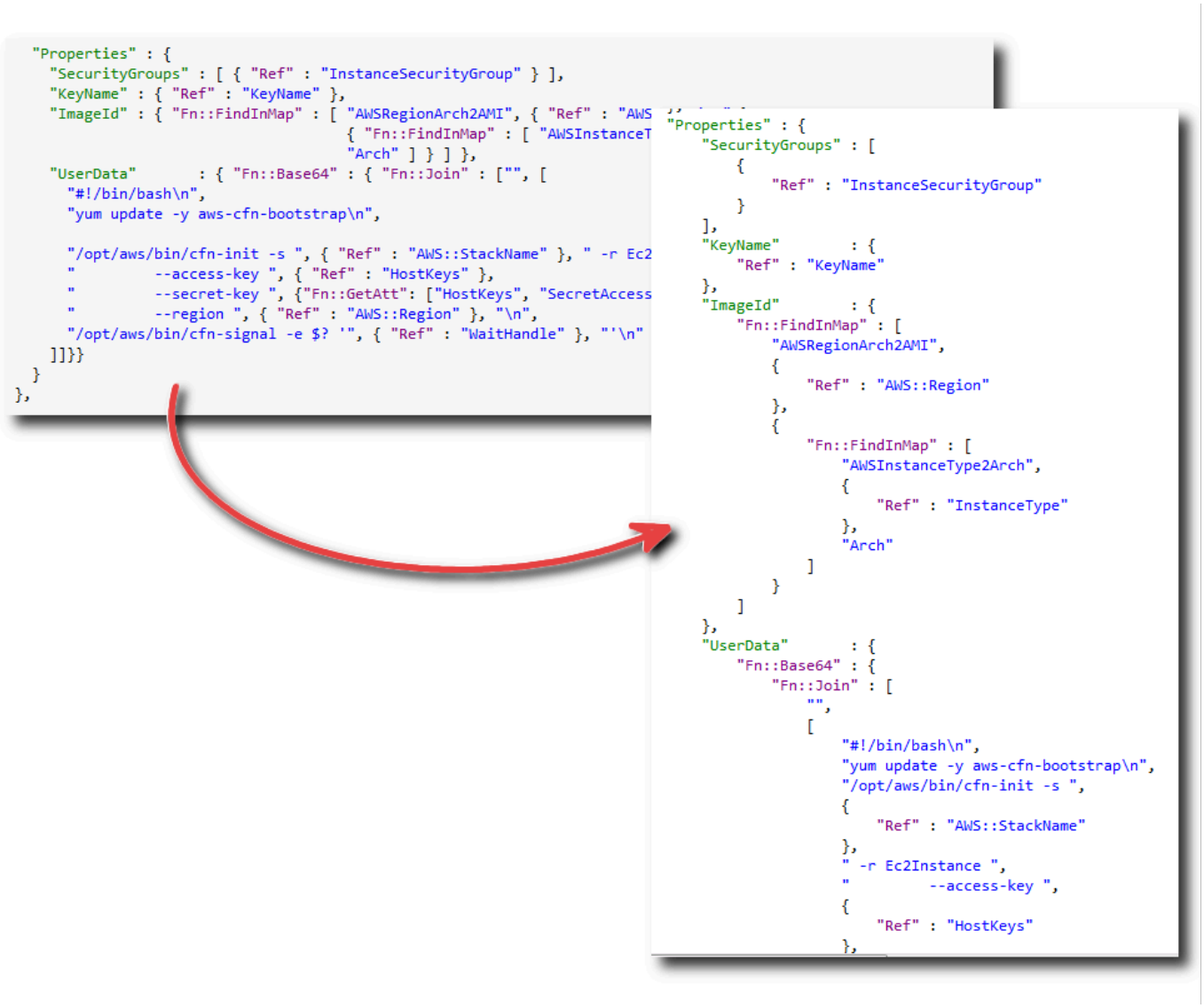
在 Visual Studio 中设置 AWS CloudFormation 模板的格式

- 在解决方案资源管理器中，打开模板的上下文（右键单击）菜单，然后选择 Format Template (格式模板)。

或者，要对您当前编辑的模板进行格式设置，请从 Template (模板) 菜单中，选择 Format Template (格式模板)。



您的 JSON 代码将进行格式设置，以便清晰地呈现其结构。



从使用 Amazon S3AWS探险者

利用 Amazon S3，您可以通过连接到 Internet 来存储和检索数据。您在 Amazon S3 上存储的所有数据与您的账户关联，而且默认情况下，只能由您访问。利 Toolkit for Visual Studio，您可以将数据存储在 Amazon S3 上并查看、管理、检索和分配这些数据。

Amazon S3 使用了存储桶的概念，您可以将此视为类似于文件系统或逻辑驱动器。存储桶可包含文件夹（类似于目录）和对象（类似于文件）。在此部分中，我们将在演练 Toolkit for Visual Studio 公开的 Amazon S3 功能时使用这些概念。

> **Note**
>
> 要使用此工具，您的 IAM 策略必须为`s3:GetBucketAcl`、`s3:GetBucket`, 和`s3:ListBucket`行动。有关更多信息，请参阅 。概述AWSIAM 策略.

创建 Amazon S3 存储桶

存储桶是 Amazon S3 中最基本的存储单位。

创建 S3 存储桶

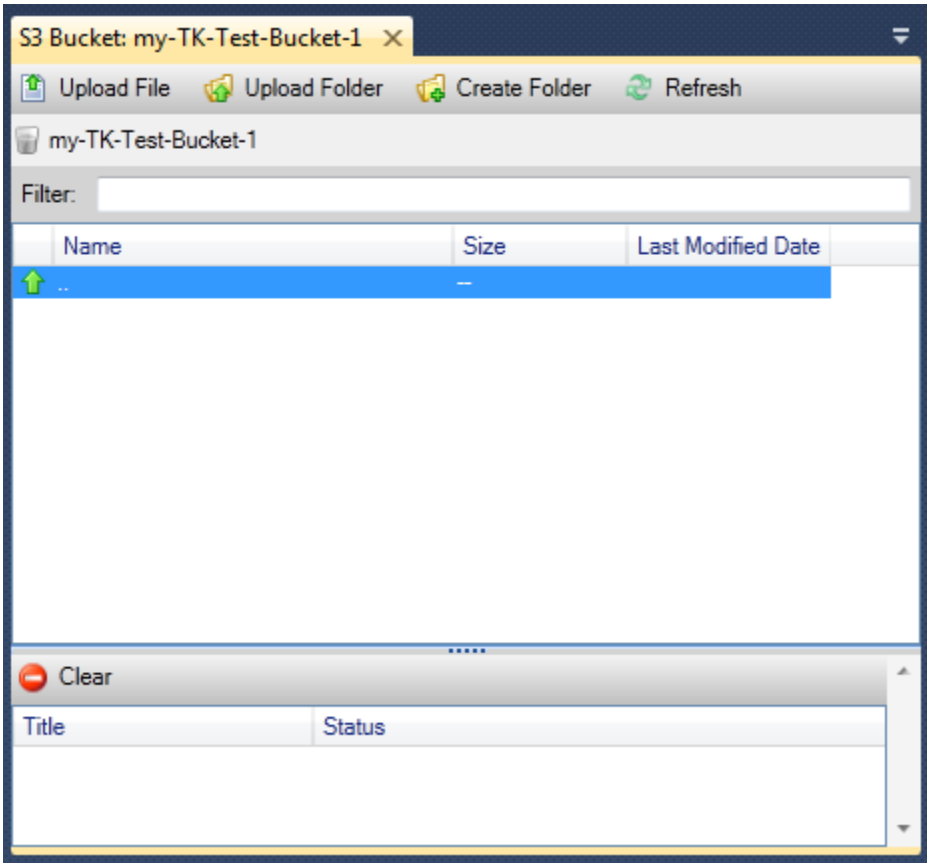
1. InAWS打开资源管理器的上下文 (右键单击) 菜单。Amazon S3节点，然后选择创建存储桶.
2. 在 Create Bucket (创建存储桶) 对话框中，为存储桶键入名称。存储桶名称在中必须唯一AWS. 有关其他约束的信息，请转到 Amazon S3 文档。
3. 选择 OK（确定）。

从管理 Amazon S3 存储桶AWS探险者

InAWS在 Explorer 中，当您打开 Amazon S3 存储桶的上下文（右键单击）菜单时可执行以下操作。

浏览

显示包含在存储桶中的对象的视图。从此处，您可以创建文件夹或者从您的本地计算机上传文件或整个目录和文件夹。下方窗格将显示有关上传过程的状态消息。要清除这些消息，请选择 Clear (清除) 图标。您还可以通过在中双击存储桶名称来访问此存储桶的视图。AWSExplorer。



属性

显示您可在其中执行以下操作的对话框：

- 将 Amazon S3 权限范围设置为：
  - 作为存储桶拥有者的您。
  - 已在上进行身份验证的所有用户AWS.
  - 具有 Internet 访问权限的每个人。
- 开启对存储桶的日志记录。
- 使用 Amazon Simple Notification Service (Amazon SNS) 设置通知，以便在您使用低冗余存储 (RRS) 时，如果发生数据丢失，您将会接到通知。RRS 是 Amazon S3 存储选项，与标准存储相比，它提供的持久性较差，但成本较低。有关更多信息，请参阅 S3 常见问题。
- 使用存储桶中的数据创建静态网站。

策略

使您能够为您的存储桶设置 AWS Identity and Access Management (IAM) 策略。有关更多信息，请转到 IAM 文档及 IAM 和 S3 的使用案例。

创建预签名 URL

使您能够生成一个限时的 URL，您可分配该 URL 以提供对存储桶内容的访问权限。有关更多信息，请参阅如何创建预签名 URL。

查看分段上传

使您能够查看分段上传。Amazon S3 支持将大型对象上传分解为多个部分以使上传过程更高效。有关更多信息，请转到 S3 文档中的分段上传的讨论。

删除

使您能够删除存储桶。您只能删除空存储桶。

将文件和文件夹上传到 Amazon S3

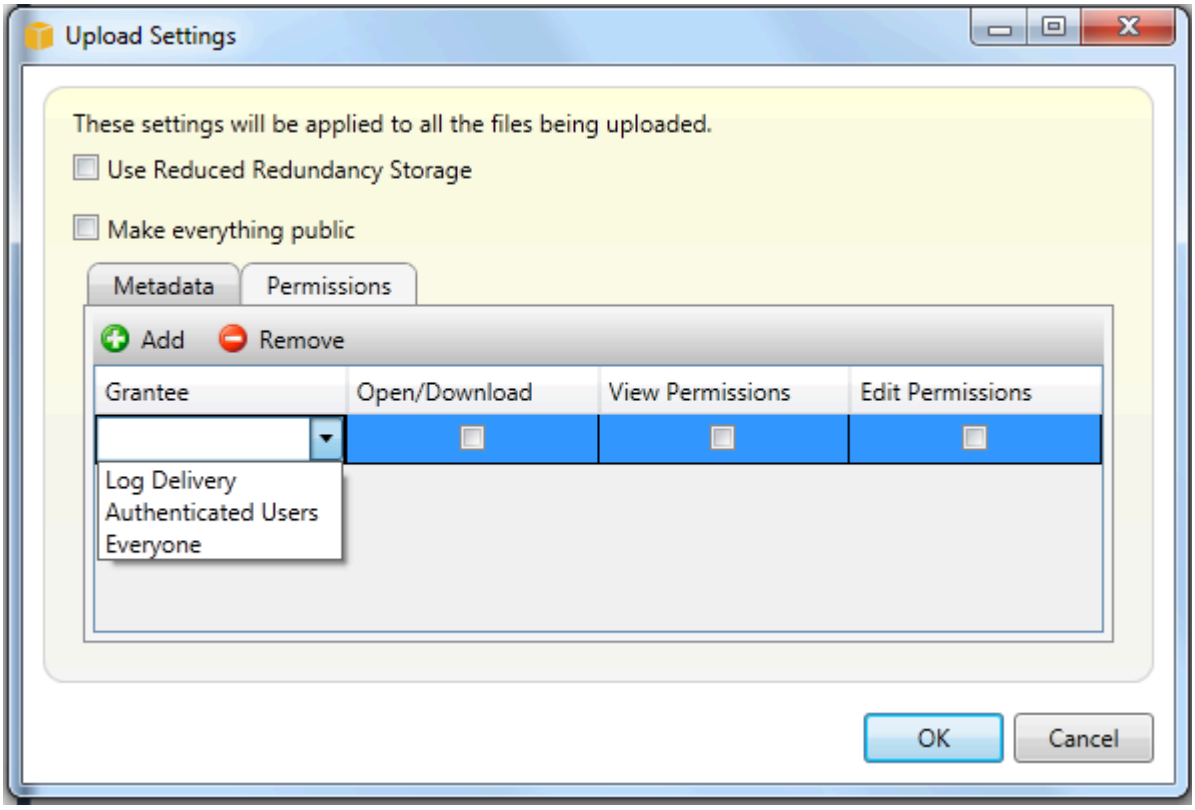
您可以使用AWS将文件或整个文件夹从您的本地计算机传输到任何存储桶。

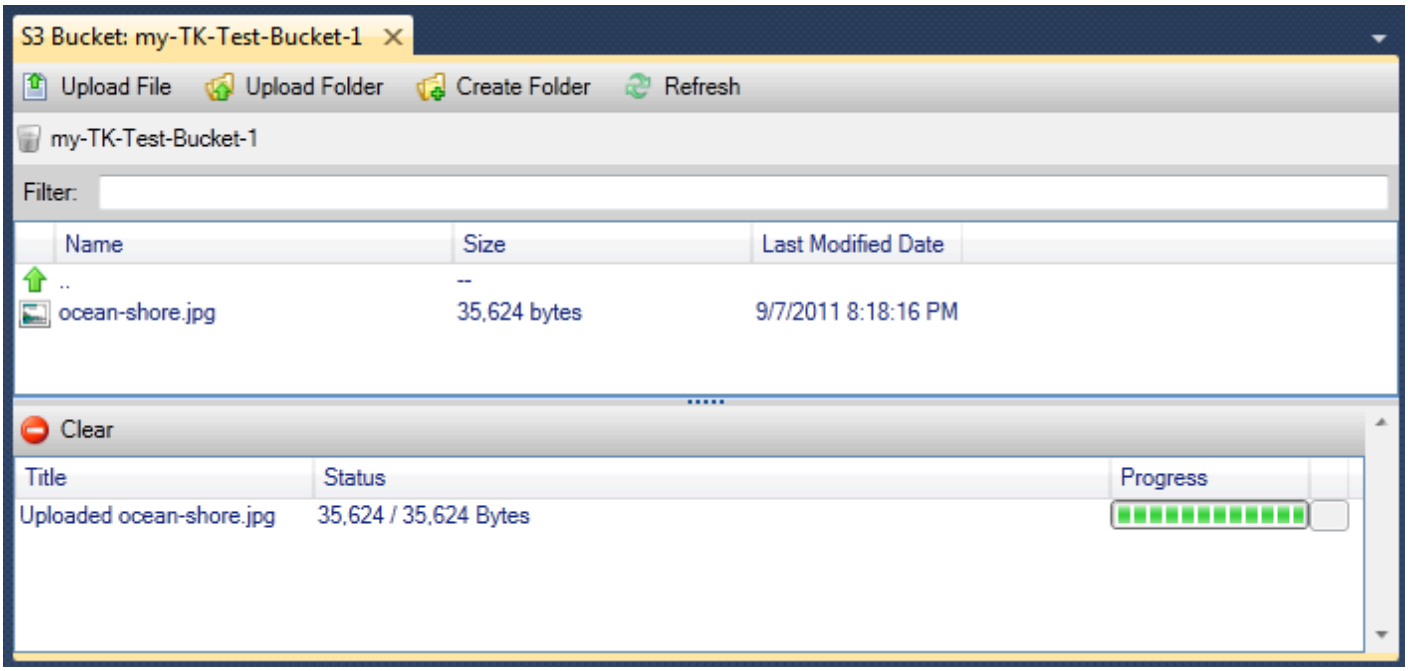
> **Note**
>
> 如果您上传的文件或文件夹与已存在于 Amazon S3 存储桶中的文件或文件夹具有相同的名称，则您上传的文件将覆盖现有文件而不进行提示。

将文件上传到 S3

1. InAWS资源管理器，展开Amazon S3节点，然后双击存储桶或打开存储桶的上下文（右键单击）菜单，然后选择浏览.
2. 在您的存储桶的 Browse (浏览) 视图中，选择 Upload File (上传文件) 或 Upload Folder (上传文件夹)。
3. 在 File-Open (文件-打开) 对话框中，导航到要上传的文件，选择这些文件，然后选择 Open (打开)。如果您要上传文件夹，请导航到并选择该文件夹，然后选择 Open (打开)。

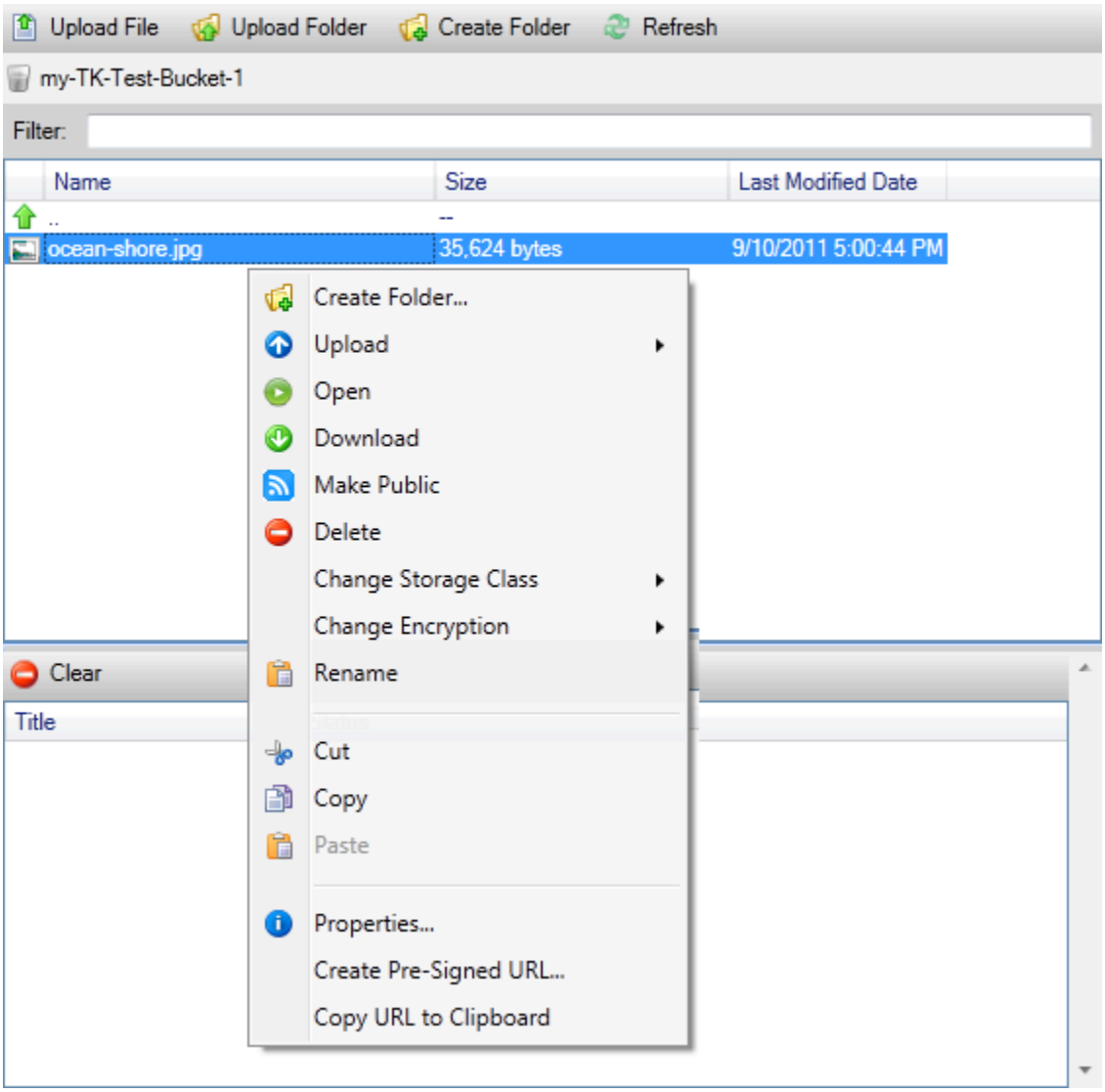
   利用 Upload Settings (上传设置) 对话框，您能够在要上传的文件或文件夹上设置元数据和权限。选中 Make everything public (将所有项公开化) 复选框等同于将 Open/Download (打开/下载) 权限设置为 Everyone (所有人)。您可以为上传的文件选择使用 Reduced Redundancy Storage (减小冗余存储) 的选项。





来自 Amazon S3 文件操作AWSToolkit for Visual Studio

如果您选择 Amazon S3 视图中的一个文件并打开相应的上下文 (右键单击) 菜单，则可对该文件执行各种操作。



Create Folder

使您能够在当前存储桶中创建文件夹。（等同于选择 Create Folder (创建文件夹) 链接。）

上传

使您能够上传文件或文件夹。（等同于选择 Upload File (上传文件) 或 Upload Folder (上传文件夹) 链接。）

Open

尝试在您的默认浏览器中打开所选文件。根据文件的类型和您的默认浏览器的功能，可能不会显示该文件。您的浏览器可能只是下载该文件。

下载

打开 Folder-Tree (文件夹-树) 对话框以使您能够下载所选文件。

公开化

将所选文件的权限设置为 Open/Download (打开/下载) 和 Everyone (所有人)。（等同于选择将所有内容公用” 上的复选框上传设置” 对话框。）

删除

删除所选文件或文件夹。您也可以通过选择文件或文件夹并按 `Delete` 来删除它们。

更改存储类

将存储类设置为“标准”或“低冗余存储 (RRS)”。要查看当前存储类设置，请选择 Properties (属性)。

更改加密

使您能够对文件设置服务器端加密。要查看当前加密设置，请选择 Properties (属性)。

重命名

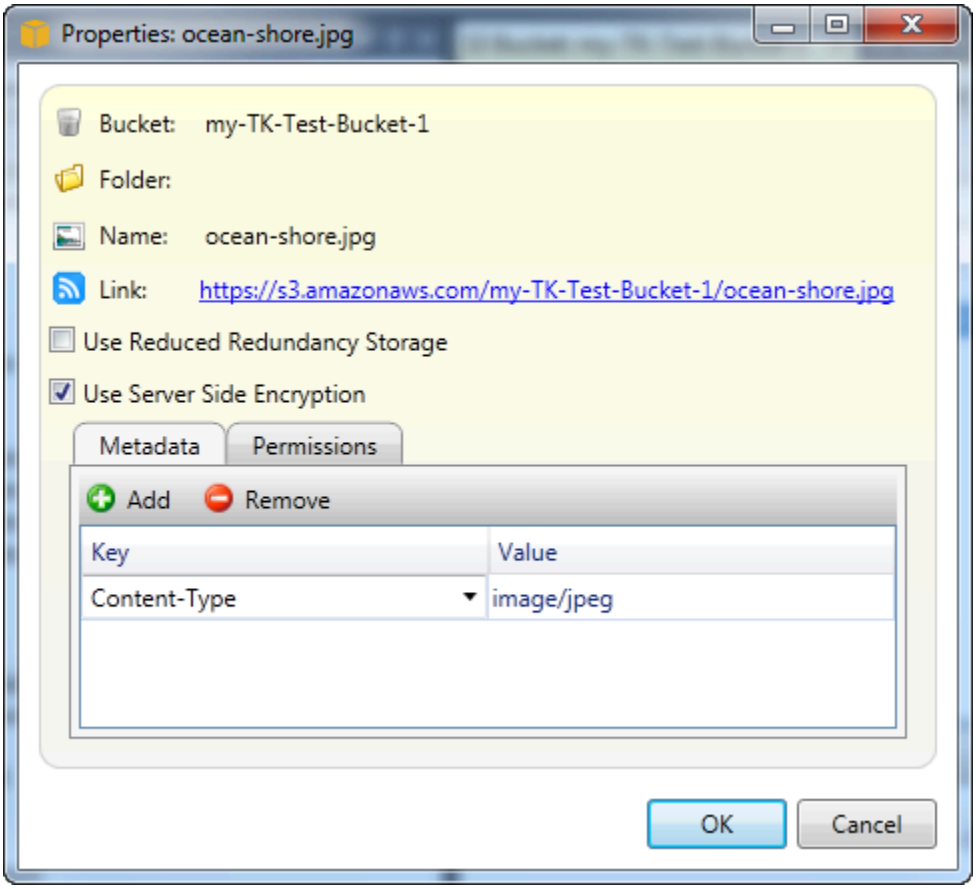
使您能够重命名文件。您无法重命名文件夹。

剪切 | 复制 | 粘贴

使您能够在文件夹或存储桶之间剪切、复制和粘贴文件或文件夹。

属性

显示一个对话框，使您可以在其中设置该文件的元数据和权限，以及在“低冗余存储 (RRS)”和“标准”之间切换文件的存储并为该文件设置服务器端加密。此对话框还显示一个指向该文件的 https 链接。如果选择此链接，Toolkit for Visual Studio 将在您的默认浏览器中打开该文件。如果您将该文件的权限设置为 Open/Download (打开/下载) 和 Everyone (所有人)，则其他人员均将能够通过此链接访问该文件。我们建议您创建并分配预签名 URL 而不是分配此链接。



创建预签名 URL

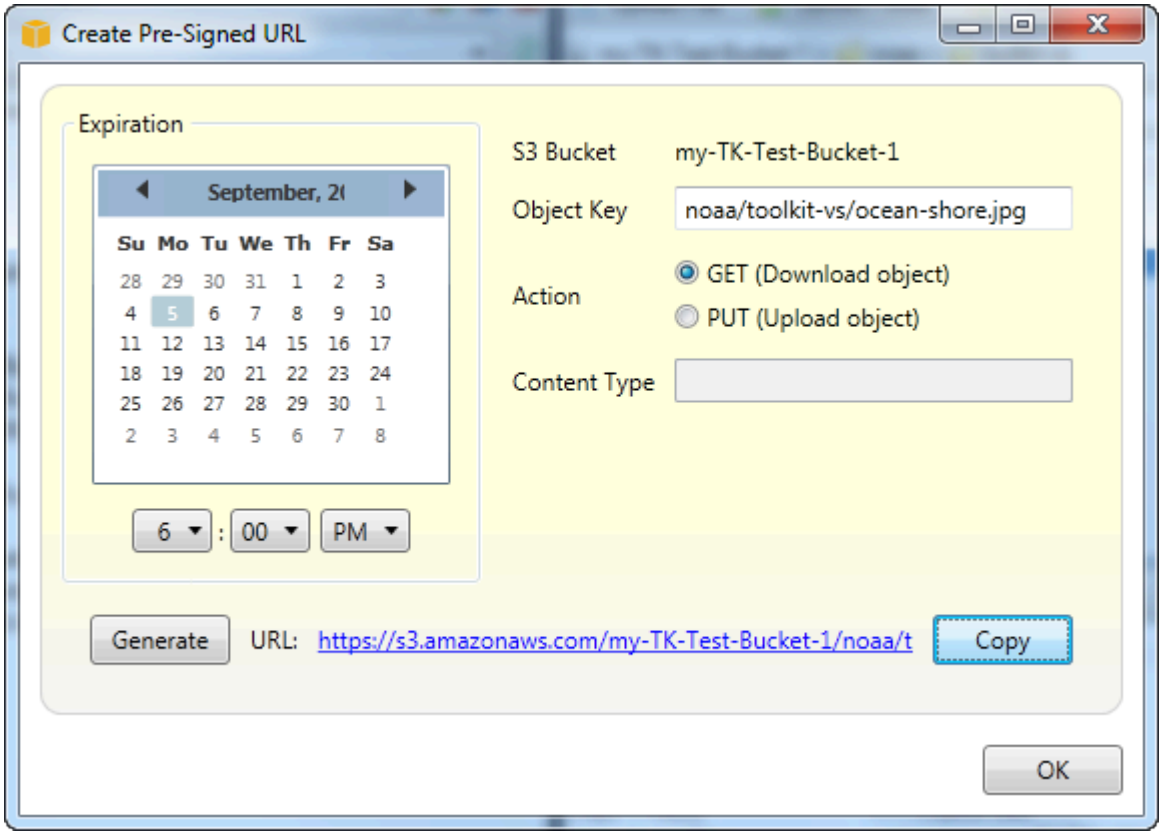
利用，您可以创建一个限时的预签名 URL，您可分配该 URL 以使其他人员能够访问已存储在 Amazon S3 上的内容。

如何创建预签名 URL

您可以为存储桶或存储桶中的文件创建预签名 URL。然后，其他人员可以使用此 URL 访问存储桶或文件。当您在创建此 URL 时指定的时间段过后，此 URL 将会过期。

创建预签名 URL

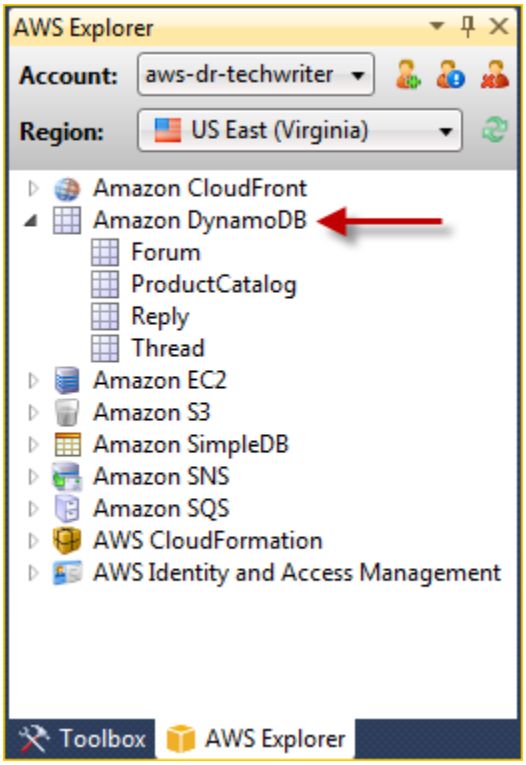
1. 在 Create Pre-Signed URL (创建预签名 URL) 对话框中，设置此 URL 的到期日期和时间。默认设置为从当前时间开始的一个小时。
2. 选择 Generate (生成) 按钮。
3. 要将此 URL 复制到剪贴板，请选择 Copy (复制)。



从使用 DynamoDBAWS探险者

Amazon DynamoDB 是一项快速、高度可扩展、高度可用且经济实惠的非关系数据库服务。DynamoDB 消除了传统上对数据存储可扩展性的限制，同时保留了低延迟性和可预测的性能。Toolkit for Visual Studio 提供了用于在开发上下文中使用 DynamoDB 的功能。有关 DynamoDB 的更多信息，请参阅DynamoDB在 Amazon Web Services 站上。

在 Toolkit for Visual Studio 中，AWSExplorer 显示了与活动项目关联的所有 DynamoDB 表。AWS 账户.



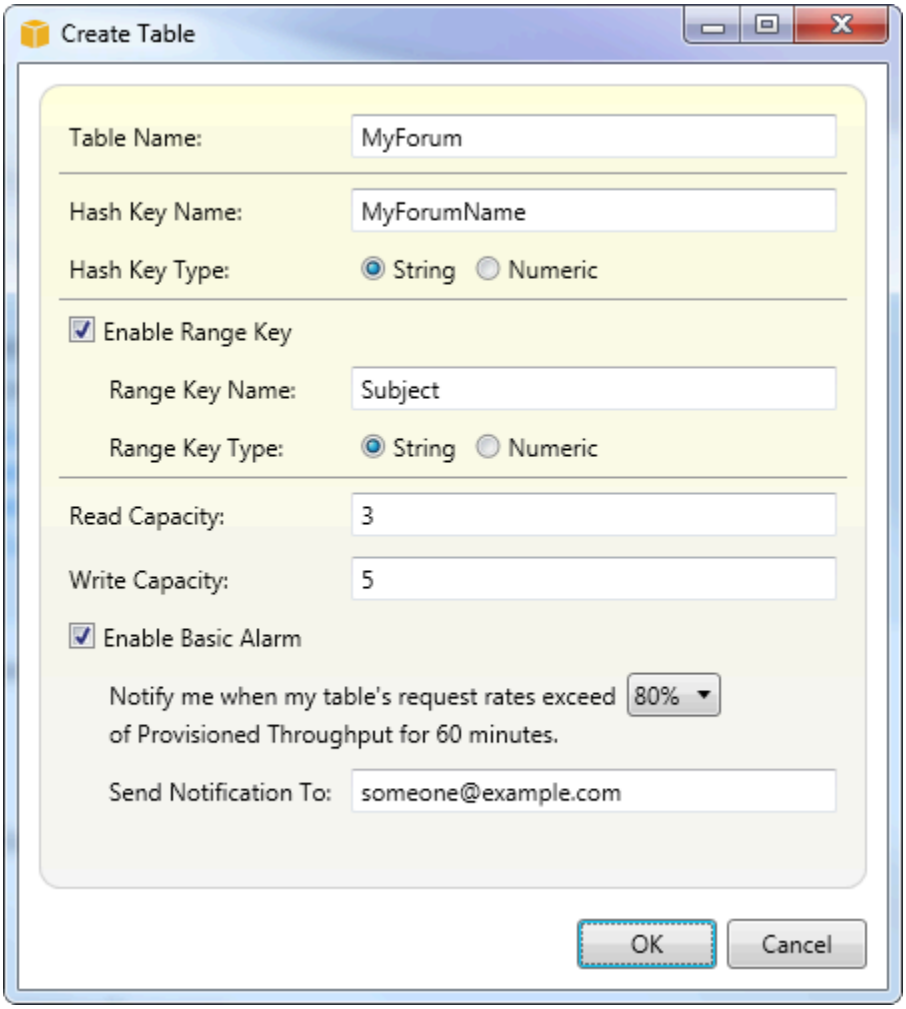
创建 DynamoDB 表

您可以使用 Toolkit for Visual Studio 创建 DynamoDB 表。

要创建表AWS探险者

1. InAWS打开的上下文 (右键单击) 菜单Amazon DynamoDB，然后选择创建表.
2. 在 Create Table (创建表) 向导的 Table Name (表名称) 中，键入表的名称。
3. 在哈希键名称字段中，键入主哈希键属性，然后从哈希键类型按钮，请选择哈希键类型。DynamoDB 使用主键属性构建无序哈希索引，并使用范围主键属性构建可选的有序范围索引。有关主哈希键属性的更多信息，请转至主键在部分中Amazon DynamoDB 开发人员指南.
4. （可选）选择 Enable Range Key (启用范围键)。在 Range Key Name (范围键名称) 字段中，键入范围键属性，然后从 Range Key Type (范围键类型) 按钮中，选择范围键类型。
5. 在 Read Capacity (读取容量) 字段中，键入读取容量单位的数量。在 Write Capacity (写入容量) 字段中，键入写入容量单位的数量。您必须至少指定 3 个读取容量单位和 5 个写入容量单位。有关读取和写入容量单位的更多信息，请转到 DynamoDB 中预配置的吞吐量。
6. （可选）选择 Enable Basic Alarm (启用基本警报) 以在表的请求速率过快时提醒您。选择每 60 分钟的预配置吞吐量的百分比，必须超过此百分比才会发送提醒。在 Send Notifications To (将通知发送到) 中，键入电子邮件地址。

7. 单击 OK (确定) 以创建表。

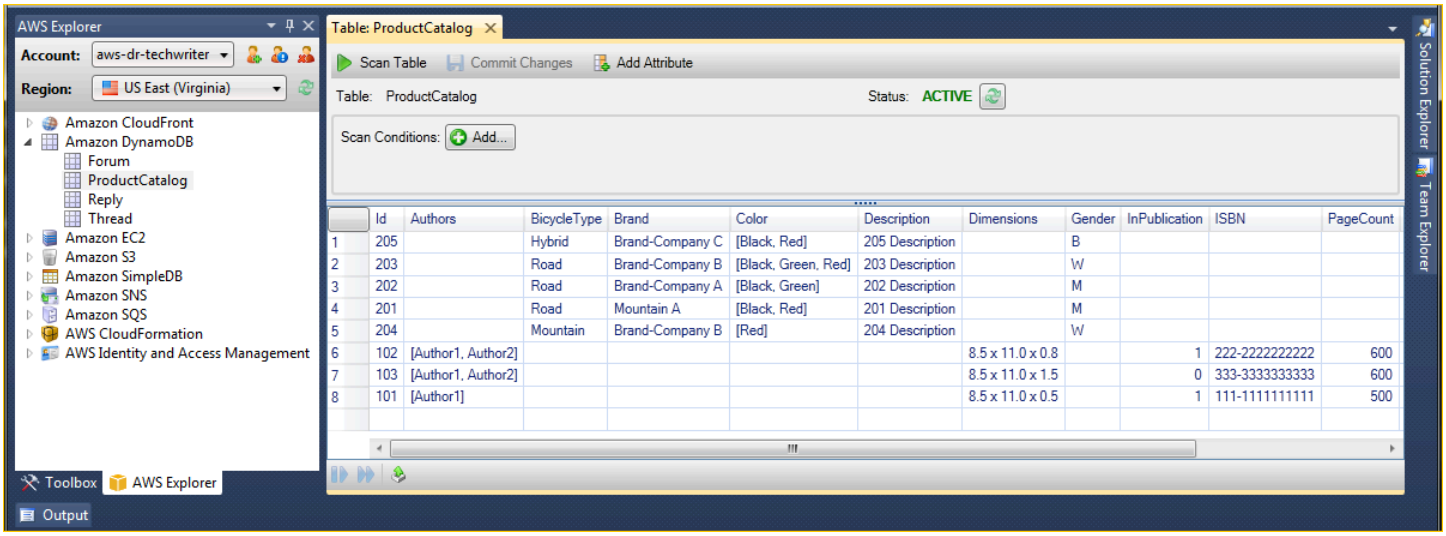


有关 DynamoDB 表的更多信息，请转至数据模型概念-表、项目和属性.

以网格形式查看 DynamoDB 表

要打开 DynamoDB 表的网格视图，请在AWS浏览器，双击与表对应的子节点。从网格视图中，您可以查看存储在表中的项目、属性和值。每个行对应于表中的一个项目。表列与属性对应。表的每个单元格保存与该项目的该属性关联的值。

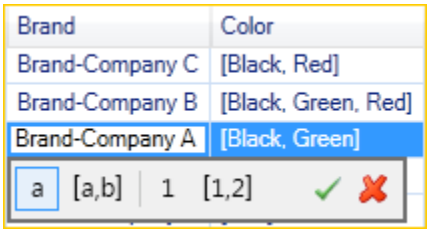
属性可以包含字符串或数字形式的值。某些属性包含由一系列 字符串或数字组成的值。系列值显示为用方括号括起的逗号分隔列表。



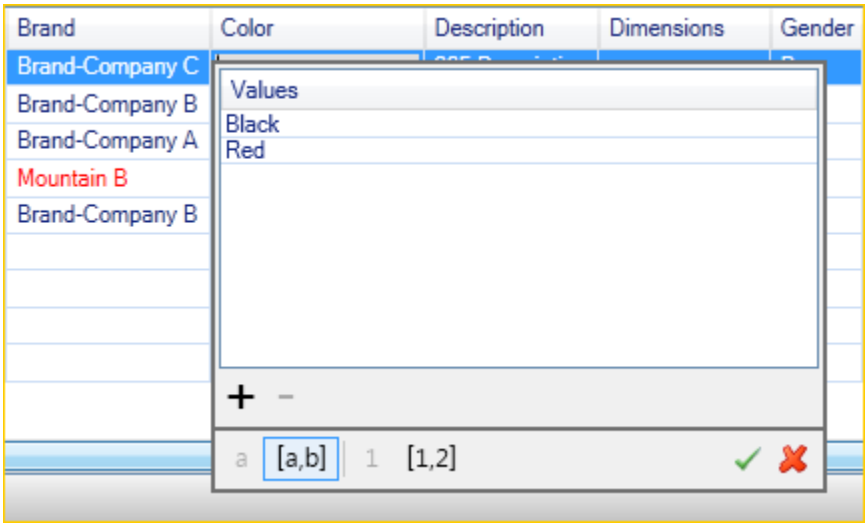
	Id	Authors	BicycleType	Brand	Color	Description	Dimensions	Gender	InPublication	ISBN	PageCount
1	205		Hybrid	Brand-Company C	[Black, Red]	205 Description		B			
2	203		Road	Brand-Company B	[Black, Green, Red]	203 Description		W			
3	202		Road	Brand-Company A	[Black, Green]	202 Description		M			
4	201		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

编辑和添加属性和值

通过双击单元格，您可以编辑项目对应属性的值。对于系列值属性，您还可以在该系列中添加或删除单个值。



除了更改属性的值之外，您还可以更改属性的值的格式（存在一些限制）。例如，任何数字值均可转换为字符串值。如果您有一个字符串值，内容为数字（如 125），那么单元格编辑器可让您将值的格式从字符串转换为数字。您还可以将单一值转换为系列值。但是，您通常无法将系列值转换为单一值；有一个例外情况，即当系列值实际上只包含一个元素时。

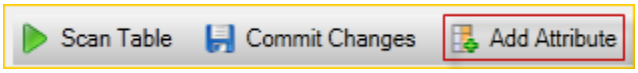


在编辑属性值后，请选择绿色的勾号以确认更改。如果要放弃更改，请选中红色 X。

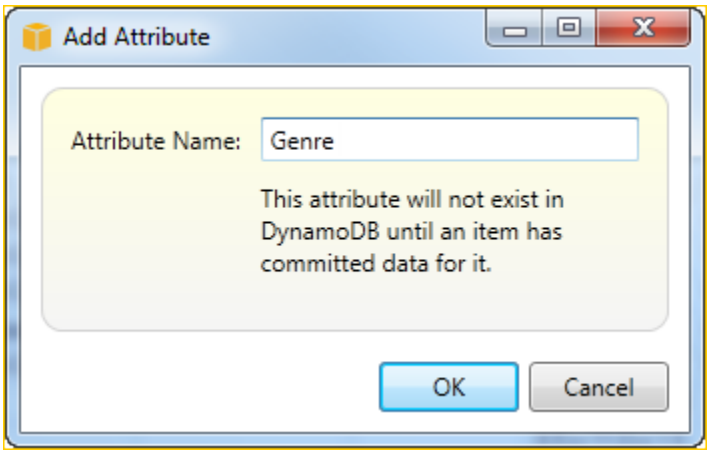
在您确认更改后，属性值将以红色显示。这表示属性已更新，但新值尚未写回到 DynamoDB 数据库。要将更改写回到 DynamoDB，请选择提交更改. 要放弃更改，请选择 Scan Table (扫描表)，当 Toolkit 询问您是否要在扫描之前提交更改时，选择 No (否)。

添加属性

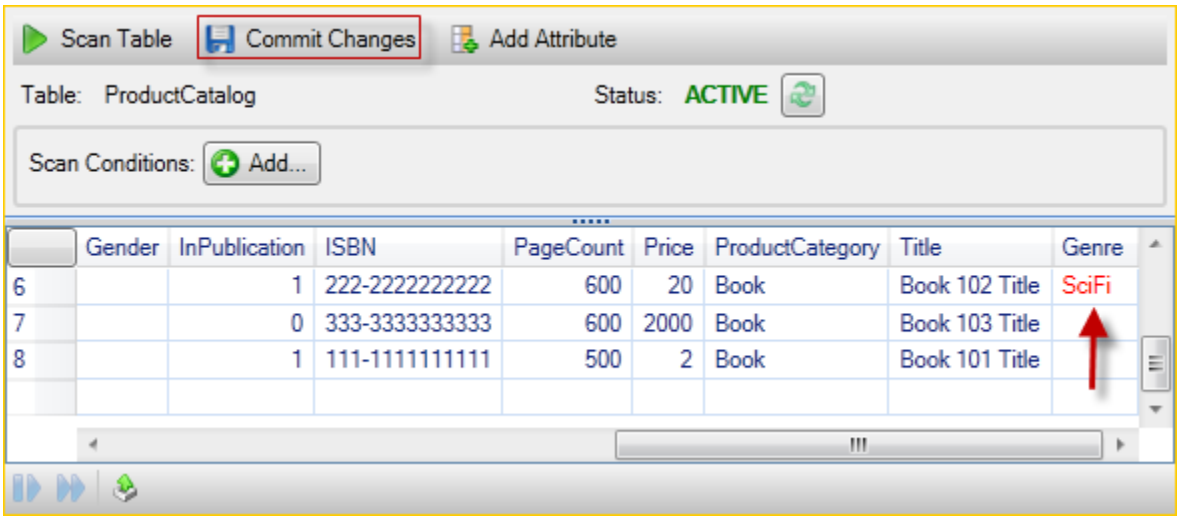
从网格视图中，您还可以将属性添加到表。要添加新属性，请选择 Add Attribute (添加属性)。



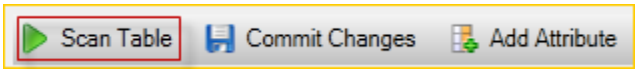
在 Add Attribute (添加属性) 对话框中，键入属性的名称，然后选择 OK (确定)。



要使新属性成为表的一部分，您必须至少为一个项目向新属性添加值，然后选择 Commit Changes (提交更改) 按钮。要放弃新属性，只需关闭表的网格视图，同时不选择 Commit Changes (提交更改) 即可。



扫描 DynamoDB 表

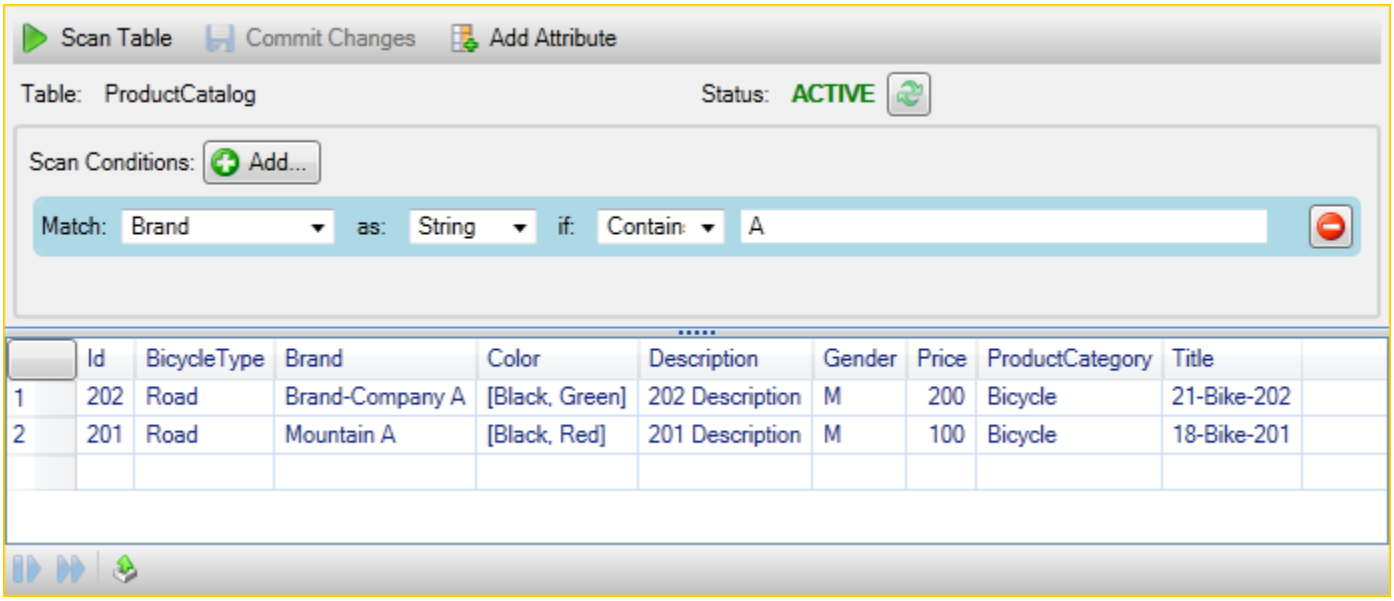


您可以从 Toolkit 对 DynamoDB 表执行扫描。在一次扫描中，您将定义一组条件，扫描将返回表中符合您的条件的所有项目。扫描是代价高昂的操作，应谨慎使用以避免干扰表中优先级更高的生产流量。有关使用扫描操作的更多信息，请转至Amazon DynamoDB 开发人员指南.

从以下方式对 DynamoDB 表执行扫描AWS探险者

1. 在网格视图中，选择 scan conditions: add (扫描条件: 添加) 按钮。
2. 在扫描子句编辑器中，选择要与之匹配的属性，应如何解释属性的值（字符串、数字、系列值）、应如何匹配属性（例如“开头为”或“包含”）以及属性应匹配的文本值。
3. 根据需要为您的搜索添加更多扫描子句。扫描只会返回符合所有扫描子句中的条件的项目。当与字符串值匹配时，扫描将执行区分大小写的比较。
4. 在网格视图顶部的按钮栏上，选择 Scan Table (扫描表)。

要删除扫描子句，请选择每个子句右侧带白线的红色按钮。



要返回到包含所有项目的表的视图，请删除所有扫描子句并再次选择 Scan Table (扫描表)。

为扫描结果分页

视图底部有三个按钮。

前两个蓝色按钮为扫描结果提供分页。第一个按钮将显示另外一页的结果。第二个按钮将显示另外十页的结果。在此上下文中，一个页面等于 1 MB 的内容。

将扫描结果导出到 CSV

第三个按钮将结果从当前扫描导出到 CSV 文件。

使用AWS CodeCommit使用 Visual Studio Team Explorer

您可以使用AWS Identity and Access Management(IAM) 用于创建 Git 凭证并使用它们在 Team Explorer 中创建和克隆存储库。

AWS CodeCommit 的凭证类型

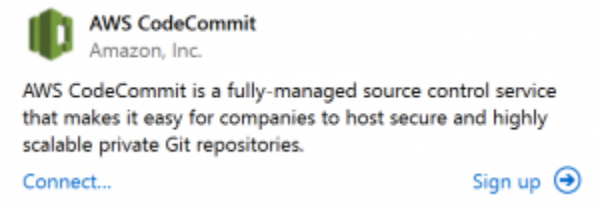
大多数AWS Toolkit for Visual Studio用户知道设置AWS包含访问密钥和私有密钥的凭证配置文件。这些凭证配置文件在 Toolkit for Visual Studio 中用于启用对服务 API 的调用，例如，用于在中列出 Amazon S3 存储桶。AWS或启动 Amazon EC2 实例。AWS CodeCommit 与 Team Explorer 的集成也使用这些凭证配置文件。但是，要使用 Git 本身，您需要额外的凭证，特别是用于 HTTPS 连接的 Git 凭证。您可以阅读有关这些凭证（用户名和密码）适用于使用 Git 凭证的 HTTPS 用户的设置中的AWS CodeCommit用户指南.

您可以创建 Git 凭证AWS CodeCommit仅适用于 IAM 用户账户。您不能为根账户创建这些凭证。您最多可以为服务创建两组这样的凭证，虽然您可以将一组凭证标记为不活动，但仍会计入您的两组凭证的限制中。请注意，您可以随时删除和重新创建凭证。当您使用AWS CodeCommit来自 Visual Studio，你的传统AWS凭证用于处理服务本身，例如，在您创建和列出存储库时。使用托管在 AWS CodeCommit 中的实际 Git 存储库时，您使用 Git 凭证。

作为支持的一部分AWS CodeCommit，Toolkit for Visual Studio 会自动为您创建和管理这些 Git 凭证并将其关联到您的AWS凭证配置文件。您无需担心手头必须有一组合适的凭证在 Team Explorer 中执行 Git 操作。一旦你用你的AWS在您使用 Git 远程时，关联的 Git 凭证将自动使用。

连接到 AWS CodeCommit

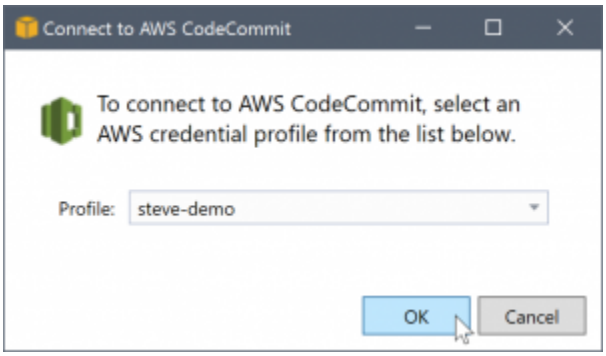
当您在 Visual Studio 2015 或更高版本中打开 Team Explorer 窗口时，您将在“Manage Connections”的“Hosted Service Providers”中看到 AWS CodeCommit 条目。



选择注册在浏览器窗口中打开 Amazon Web Services 主页。选择时会发生什么情况Connect (连接)取决于 Toolkit for Visual Studio 是否可以找到凭证配置文件AWS使其能够拨打电话的访问权限和密钥AWS代表您。在 Visual Studio 的 Toolkit 找不到任何本地存储的凭证时，您可使用在 IDE 中显示的新“Gted Started” 页面设置凭证配置文件。或者您可能一直在使 Toolkit for Visual Studio，AWS Tools for Windows PowerShell，或者AWS CLI而且已经有AWS可供 Visual Studio 使用的 Toolkit 的凭证配置文件。

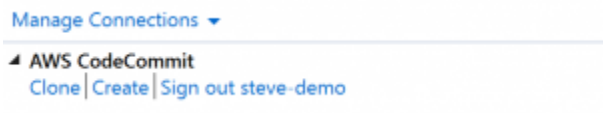
当您选择Connect (连接)，Toolkit for Visual Studio 开始查找要在连接中使用的凭证配置文件的过程。如果 Toolkit for Visual Studio 找不到凭证配置文件，将打开一个对话框，邀请您输入您的访问密钥和私有密钥。AWS 账户. 强烈建议您使用 IAM 用户账户而非使用您的根凭证。此外，如前文所述，只能为 IAM 用户创建您最终需要的 Git 凭证。提供访问密钥和私有密钥并创建凭证配置文件之后，Team Explorer 和 AWS CodeCommit 之间的连接已可供使用。

如果 Toolkit for Visual Studio 找到多个AWS系统将提示您选择要在 Team Explorer 中使用的帐户。



如果您只有一个凭证配置文件，Toolkit for Visual Studio 将绕过配置文件选择对话框，您将立即连接：

在 Team Explorer 与 AWS CodeCommit 之间通过凭证配置文件建立连接时，将关闭邀请对话框并显示连接面板。

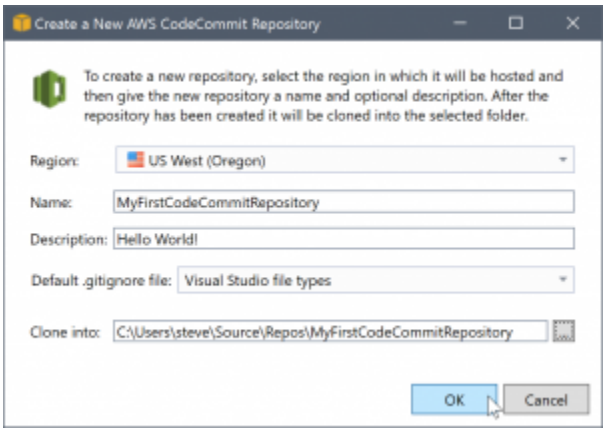


由于您没有本地克隆的存储库，面板只显示您可以执行的操作：Clone、Create, 和注销. 像其他提供商一样，AWS CodeCommit在 Team Explorer 中只能绑定到一个AWS在任何指定时间的凭证配置文件。要切换账户，您可以使用 Sign out (注销) 删除连接，以便您使用不同账户启动新的连接。

现在，您已建立了连接，您可以通过单击 Create (创建) 链接创建存储库。

创建存储库

当你点击Create链接，创建新AWS CodeCommitRepository此时显示对话框。



AWS CodeCommit 存储库按区域排列，因此您可以在 Region (区域) 中选择在哪个区域中托管存储库。列表中有 AWS CodeCommit 支持的所有区域。您为新存储库提供名称 (必需) 和说明 (可选)。

对话框的默认行为是使用存储库名称 (您输入的名称，同时更新文件夹位置) 作为新存储库文件夹位置的后缀。要使用不同的文件夹名称，请在完成输入存储库名称后编辑 Clone into (克隆到) 文件夹路径。

您还可以选择自动为存储库创建初始 `.gitignore` 文件。AWS Toolkit for Visual Studio 为 Visual Studio 文件类型提供了内置的默认值。您还可以选择没有文件，或者使用您希望在各个存储库中重用的自定义现有文件。只需在列表中选择 Use custom (使用自定义) 并导航到要使用的自定义文件。

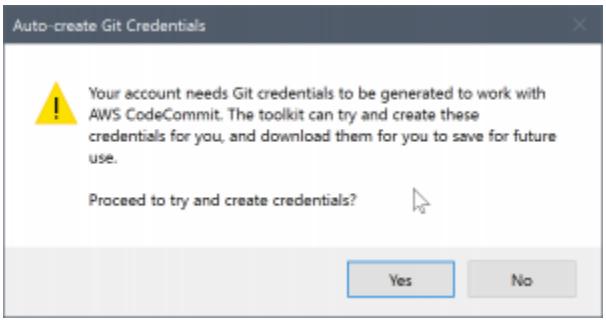
在您有了存储库的名称和位置之后，您可以单击 OK (确定) 并开始创建存储库。Toolkit for Visual Studio 请求服务创建存储库，然后本地克隆新的存储库，为 .gitignore 文件添加初始提交 (如果您要使用该文件)。此时您开始使用 Git remote，因此 Toolkit for Visual Studio 现在需要访问之前所述的 Git 凭证。

设置 Git 凭证

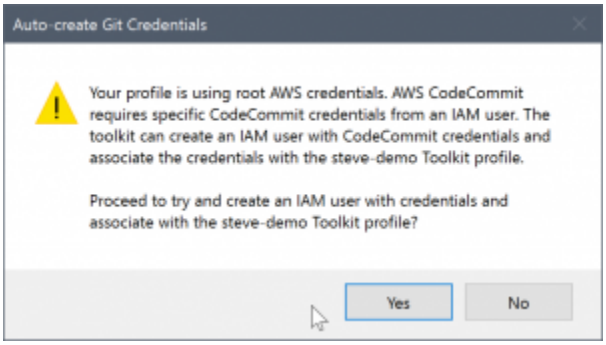
到目前为止你一直在使用AWS请求服务创建您的存储库的访问密钥和私有密钥。现在，您需要使用 Git 本身执行实际克隆操作，Git 不了解AWS访问和秘密密钥。因此您需要向 Git 提供用户名和密码凭证以在 HTTPS 连接上使用 remote。

如设置 Git 凭证中所述，您将使用的 Git 凭证必须与 IAM 用户关联。您不能为根凭证生成它们。您应该始终设置AWS包含 IAM 用户访问密钥和私有密钥，而不是根密钥的凭证配置文件。Toolkit to Visual Studio 可以尝试为AWS CodeCommit为了你，然后将它们与AWS您之前用于在团队资源管理器中连接的凭据配置文件。

当您选择确定中的创建新AWS CodeCommitRepository对话框并成功创建存储库，Toolkit for Visual Studio 会检查AWS在团队资源管理器中连接的凭据配置文件，以确定 Git 凭证是否AWS CodeCommit存在并在本地与配置文件关联。如果是这样的话，Visual Studio 指示 Team Explorer 开始在新存储库上的克隆操作。如果 Git 凭证在本地不可用，Toolkit for Visual Studio 会检查在 Team Explorer 的连接中使用的账户凭证的类型。如果该凭证用于 IAM 用户 (就像我们建议的那样)，则将显示以下消息。

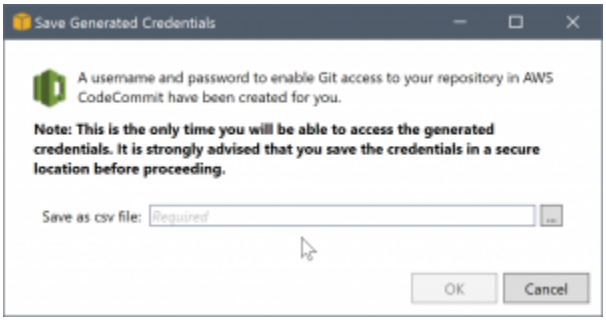


如果凭证是根凭证，将会改为显示以下消息。



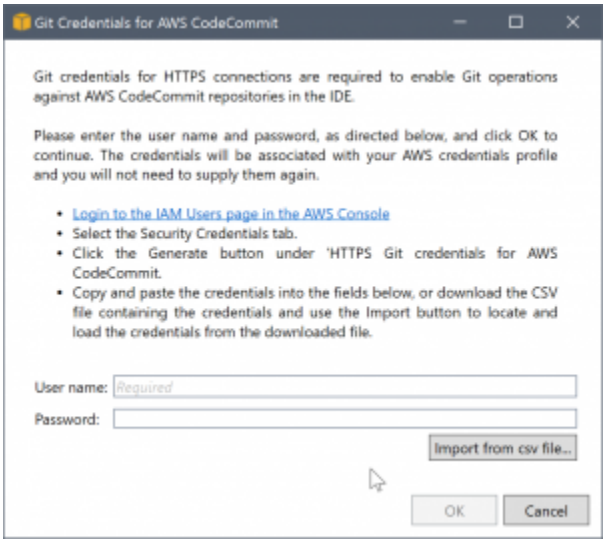
在这两种情况下，Toolkit for Visual Studio 方案尝试完成操作来为您创建必需的 Git 凭证。在第一种情况下，所有需要创建的是 IAM 用户的一组 Git 凭证。当根账户正在使用中时，Toolkit for Visual Studio 首先尝试创建一个 IAM 用户，然后继续为该新用户创建 Git 凭证。如果 Toolkit for Visual Studio 必须创建新用户，它将应用AWS CodeCommit针对该新用户帐户的超级用户托管策略。此策略只允许访问 AWS CodeCommit 并允许使用 AWS CodeCommit 执行除了存储库删除之外的所有操作。

在您创建凭证时，您只能查看一次凭证。因此，Toolkit for Visual Studio 提示您将新创建的凭证另存为`.csv`先文件然后再继续。



这是我们强烈建议的操作，并确保将它们保存到安全位置！

在有些情况下，Toolkit for Visual Studio 可能会无法自动创建凭证。例如，您可能已创建的 Git 凭证的最大数量AWS CodeCommit(两个)，或者您可能没有足够的编程权限让 Toolkit for Visual Studio 为您完成工作（如果您以 IAM 用户身份登录）。在这些情况下，您可以登录AWS Management Console以管理凭证或从管理员处获取凭证。然后，您可以在Git Credentials for AWS CodeCommitToolkit for Visual Studio 将显示该对话框。

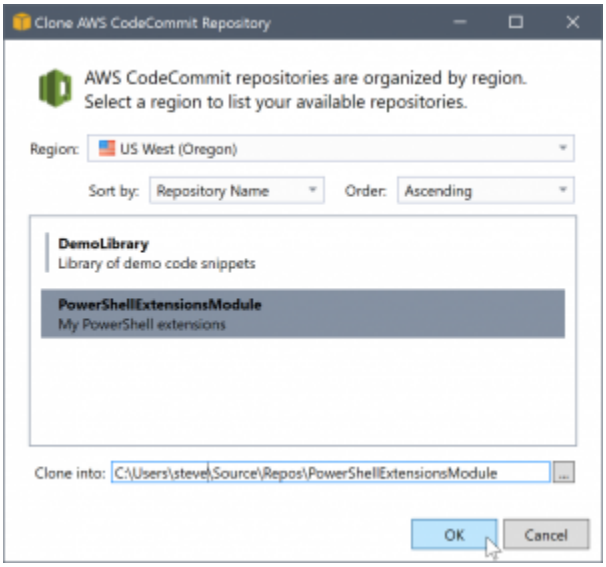


现在 Git 的凭证已可用，新存储库的克隆操作将继续 (请查看 Team Explorer 中操作的进度指示)。如果您选择应用默认 `.gitignore` 文件，则会使用注释“Initial Commit”将其提交到存储库。

这是在 Team Explorer 中设置凭证和创建存储库所需的全部操作。一旦准备好所需的凭证，以后您在创建新存储库时就只会看到创建新AWS CodeCommitRepository对话框本身。

克隆存储库

要克隆现有存储库，请返回到 Team Explorer 中 AWS CodeCommit 的连接面板。单击Clone链接以打开CloneAWS CodeCommitRepository然后选择要克隆的存储库以及您要在磁盘上放置它的位置。



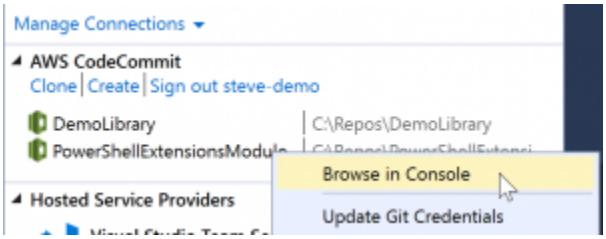
在您选择区域之后，Toolkit for Visual Studio 将查询服务以发现该区域中可用的存储库，并将其显示在对话框的中央列表部分中。还将显示每个存储库的名称和可选说明。您可以重新排序列表，将其按存储库名称或上次修改日期，以升序或降序排序。

在您选择存储库之后，您可以选择要克隆到的位置。这会默认为在 Team Explorer 中其他插件使用的相同存储库位置，不过您可以浏览或输入任何其他位置。默认情况下，存储库名称作为后缀添加到选定的路径。但是，如果您希望使用特定路径，只需在选择文件夹之后编辑文本框。单击 OK (确定) 时文本框中的任何文本将成为文件夹，您可在其中查找克隆的存储库。

选择了存储库和文件夹位置之后，您可以单击 OK (确定) 以继续克隆操作。就像创建存储库一样，您可以在 Team Explorer 中查看报告的克隆操作进度。

使用存储库

当您克隆或创建存储库时，请注意在 Team Explorer 中连接面板的操作链接下列出的本地存储库的连接。这些条目为您提供了一种简便的方法访问存储库以浏览内容。只需右键单击存储库并选择 Browse in Console (在控制台中浏览)。



您也可以使用 Update Git Credentials (更新 Git 凭证) 更新与凭证配置文件关联的已存储 Git 凭证。如果您已轮换凭证，这非常有用。该命令将打开Git Credentials for AWS CodeCommit您可以在其中输入或导入新凭证的对话框。

存储库上的 Git 操作均按您的预期工作。您可以发出本地提交，并在准备好共享时，使用 Team Explorer 中的“Sync”选项。因为 Git 凭证已存储在本地并与我们的已连接关联AWS凭证配置文件，系统不会提示我们再次为AWS CodeCommit远程。

在 Visual Studio 中使用 CodeArtifact

AWS CodeArtifact是一种完全托管的工件存储库服务，使组织能够轻松地存储和共享用于应用程序开发的软件包。您可以将 CodeArtifact 与流行的构建工具和软件包管理器（例如 NuGet 和 .NET Core CLI 和 Visual Studio）结合使用。您还可以配置 CodeArtifact 以从外部的公共存储库中提取软件包，例如Nuget.org.

在 CodeArtifact 中，你的软件包存储在仓库中，然后存储在域中。这些区域有：AWS Toolkit for Visual Studio简化了使用 CodeArtifact 存储库对 Visual Studio 的配置，从而可以轻松地在 Visual Studio 中直接使用 CodeArtifact 和 Nuget.org 中的软件包。

将你的 CodeArtifact 存储库添加为 NuGet 软件包源

要使用 CodeArtifact 中的软件包，您需要将存储库作为可打包源添加到NuGet 软件包管理器在 Visual Studio 中

将仓库添加为软件包源

1. InAWS在资源管理器中导航到存储库AWS CodeArtifact节点。
2. 打开要添加的存储库的上下文菜单 (右键单击)，然后选择复制 NuGet 源终端节点.
3. 导航到包源在下面NuGet 软件包管理器在节点工具 > 选项菜单。
4. In包源中，选择加号 (+)，编辑名称，然后粘贴您之前复制的 NuGet 源终端节点 URL源字段中返回的子位置类型。
5. 选中新添加的软件包源旁边的复选框以启用它。

> **Note**
>
> 我们建议添加外部连接Nuget.org转到你的 CodeArtifact 然后禁用nuget.org在 Visual Studio 中打包源代码。使用外部连接时，所有依赖项都从中拉出Nuget.org存储在 CodeArtifact

中。如果Nuget.org由于任何原因而停机，你需要的包裹仍然可用。有关外部连接的更多信息，请参阅。添加外部连接中的AWS CodeArtifact用户指南.

6. 选择确定关闭菜单。

有关将 CodeArtifact 与 Visual Studio 结合使用的更多信息，请参阅。将 CodeArtifact 与 Visual Studio 结合使用中的AWS CodeArtifact用户指南.

Amazon RDS 来自AWS探险者

Amazon RDS 是一项可让您在云中预配置和管理 SQL 关系数据库系统的服务。Amazon RDS 支持以下三种类型的数据库系统：

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

有关更多信息，请参阅 Amazon RDS 用户指南。

此处讨论的很多功能还可通过AWS管理控制台— 对于 Amazon RDS 来说为

主题

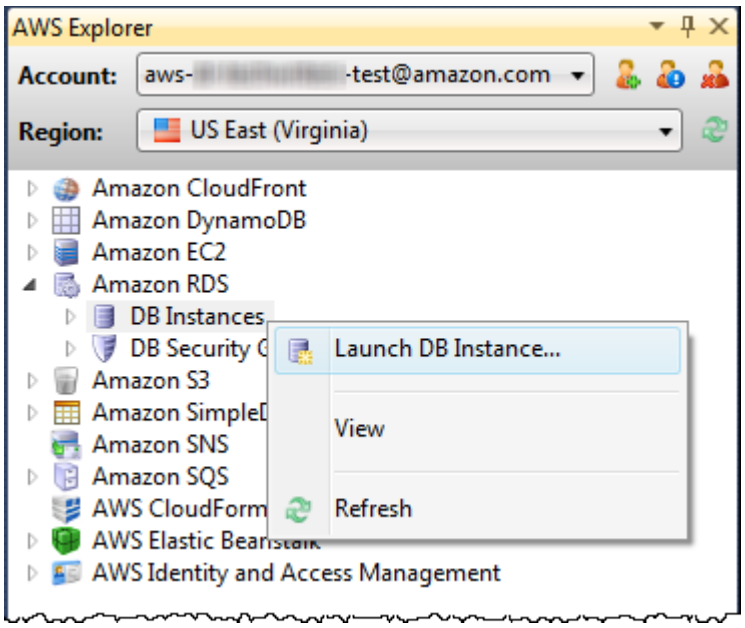
- 启动 Amazon RDS 数据库实例
- 在 RDS 实例中创建 Microsoft SQL Server 数据库
- Amazon RDS 安全组

启动 Amazon RDS 数据库实例

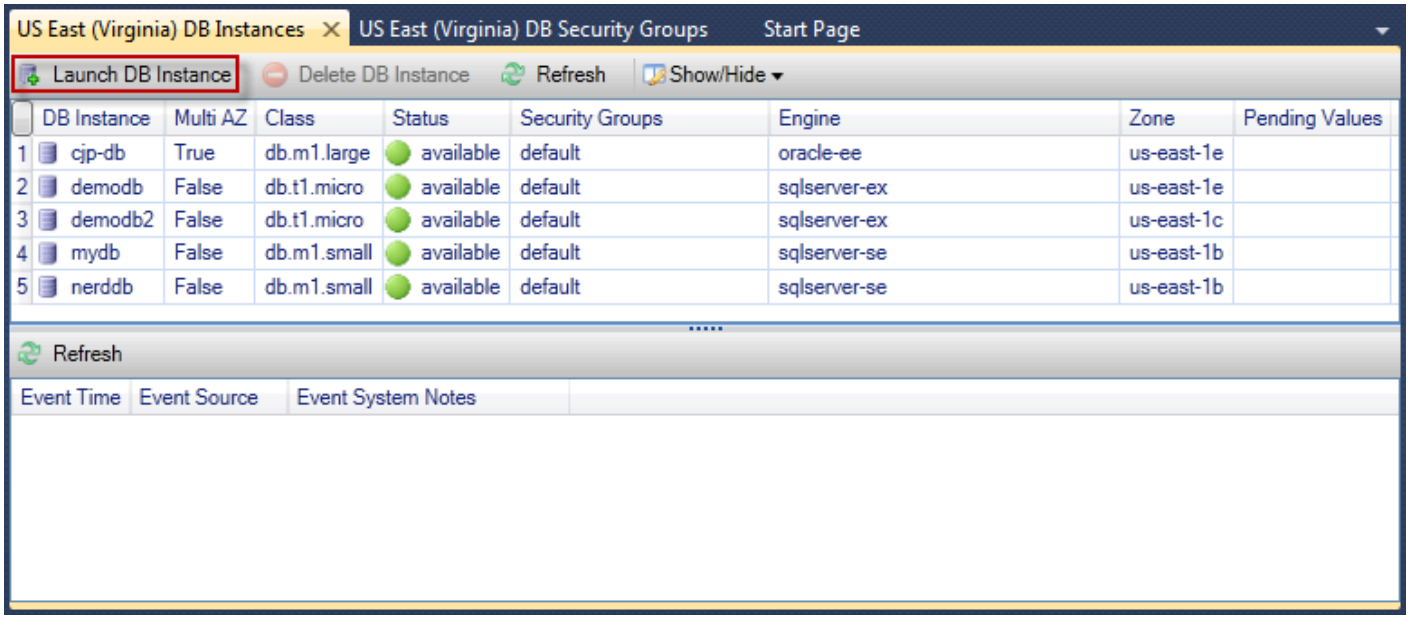
与AWSExplorer，您可以启动 Amazon RDS 支持的任何数据库引擎的实例。以下演练显示启动 Microsoft SQL Server Standard Edition 的实例的用户体验，但此用户体验对于所有支持的引擎均类似。

启动 Amazon RDS 实例

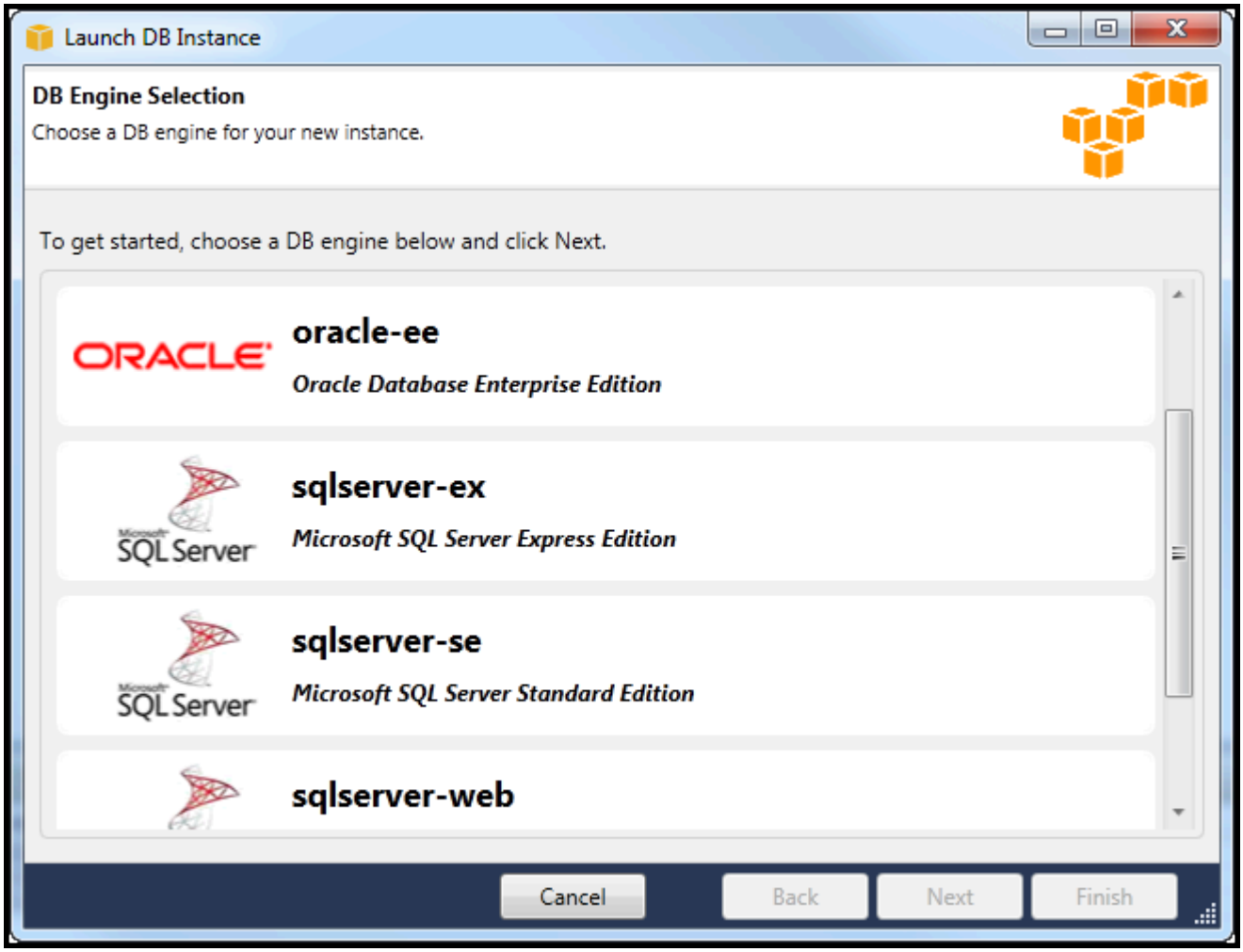
1. InAWS打开 Explorer，打开Amazon RDS节点然后选择启动数据库实例.



或者，在 DB Instances (数据库实例) 选项卡上，选择 Launch DB Instance (启动数据库实例)。



2. 在 DB Engine Selection (数据库引擎选择) 对话框中，选择要启动的数据库引擎的类型。在此演练中，选择 Microsoft SQL Server Standard Edition (sqlserver-se)，然后选择 Next (下一步)。



3. 在 DB Engine Instance Options (数据库引擎实例选项) 对话框中，选择配置选项。

   在 DB Engine Instance Options and Class (数据库引擎实例选项和类) 部分中，您可以指定以下设置。

   许可模式

引擎类型	许可证
Microsoft SQL Server	附带许可
MySql	一般公用许可证
Oracle	自带许可

许可证模型因数据库引擎的类型而异。引擎类型许可证 Microsoft SQL Server 附带许可 MySql general-public-license Oracle bring-your-own-license

DB Instance Version (数据库实例版本)

选择您要使用的数据库引擎的版本。如果仅支持一个版本，则将为您选择该版本。

数据库实例类

选择数据库引擎的实例类。实例类的定价有所差异。有关更多信息，请参阅 Amazon RDS 定价。

Perform a multi AZ deployment (执行多可用区部署)

选择此选项可创建多可用区部署以增强数据持久性和可用性。Amazon RDS 会在不同的可用区中预配置和维护数据库的备用副本，以确保发生计划内或计划外停机时自动执行故障转移。有关多可用区部署的定价信息，请参阅 Amazon RDS 详细信息页面的定价部分。Microsoft SQL Server 不支持此选项。

Upgrade minor versions automatically (自动升级次要版本)

选择此选项可AWS自动为您在 RDS 实例上执行次要版本更新。

在 RDS Database Instance (RDS 数据库实例) 部分中，您可以指定以下设置。

分配的存储空间

Engine	最小 (GB)	最大 (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

最小和最大的分配存储取决于数据库引擎的类型。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

DB Instance Identifier

为数据库实例指定名称。此名称不区分大小写。它将以小写形式显示AWSExplorer。

Master User Name (主用户名

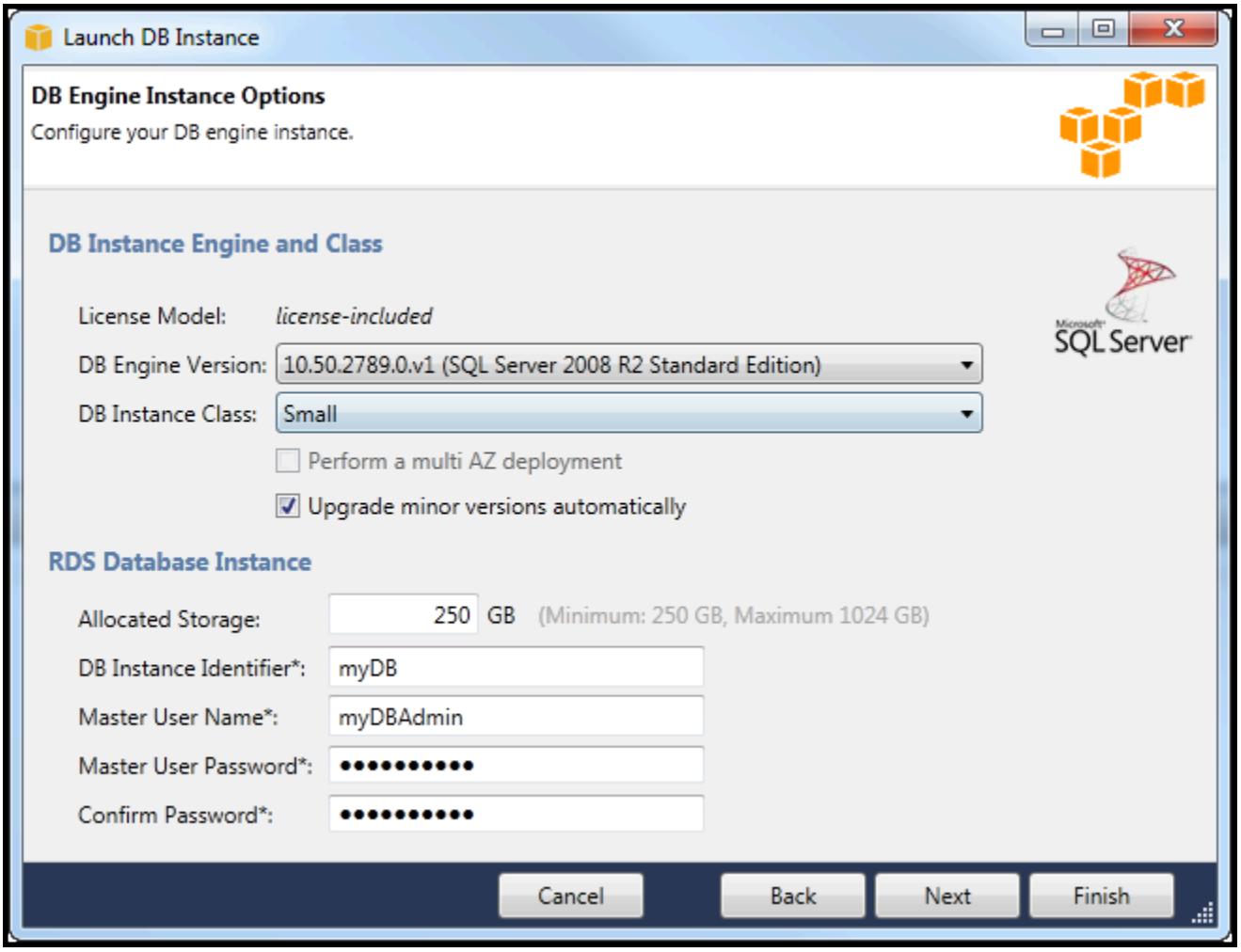
键入数据库实例的管理员的姓名。

主用户密码

键入数据库实例的管理员的密码。

确认密码

再次键入密码以验证其是否正确。



1. 在 Additional Options (其他选项) 对话框中，您可以指定以下设置。

   Database Port

   这是实例将用于在网络上进行通信的 TCP 端口。如果您的计算机通过防火墙访问 Internet，请将此值设置为您的防火墙允许流量通过的端口。

   可用区：

   如果您希望在您区域的某个特定可用区中启动实例，请使用此选项。您指定的数据库实例可能不会在某个给定区域的所有可用区中可用。

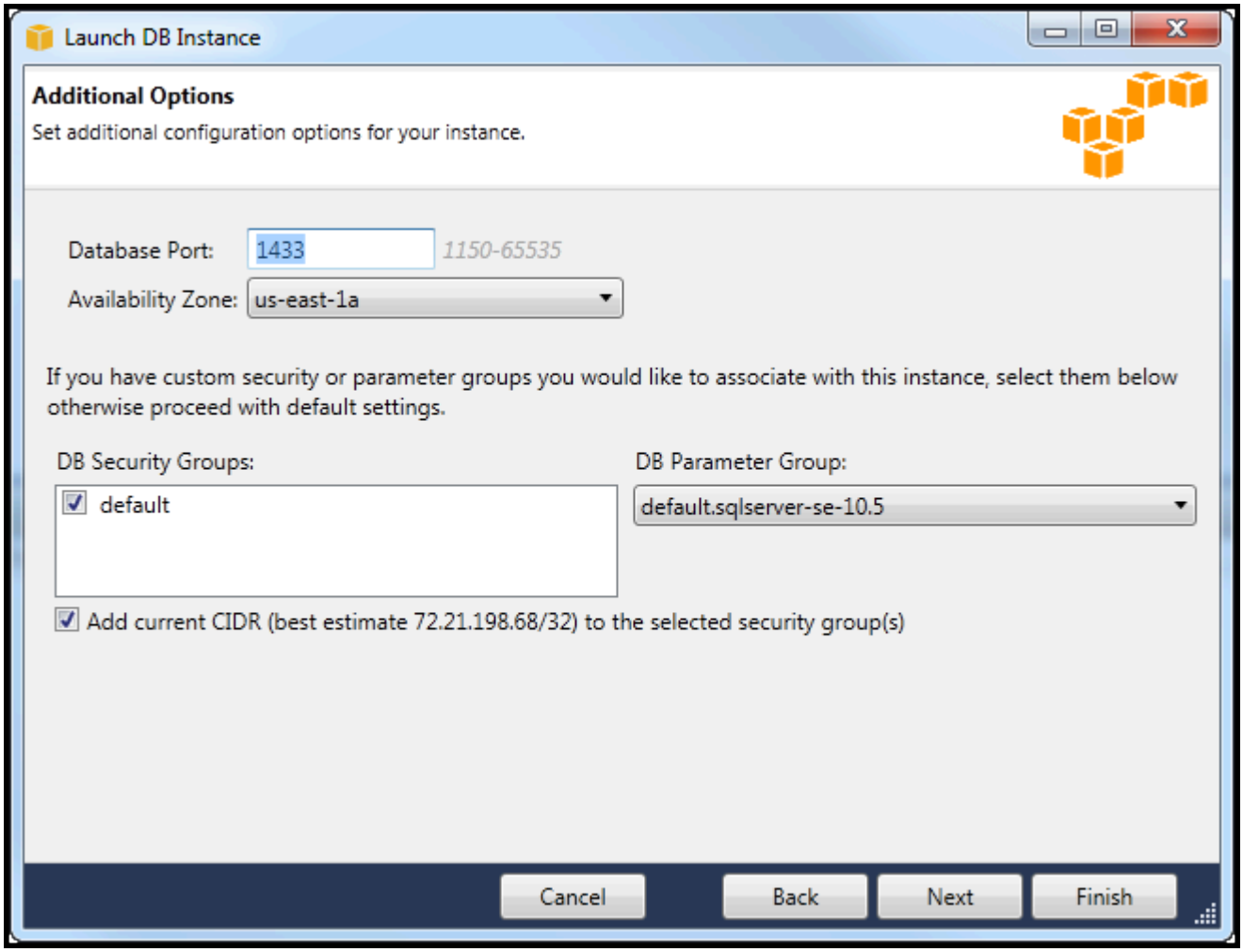
   RDS Security Group (RDS 安全组)

   选择要与您的实例关联的 RDS 安全组。RDS 安全组指定 IP 地址、Amazon EC2 实例和AWS账户允许访问您的实例。有关 RDS 安全组的更多信息，请参阅 Amazon RDS 安全组。利用 Toolkit for Visual Studio 尝试确定您的当前 IP 地址并提供用于将此地址添加到与您的实例关联的安全组的选项。但是，如果您的计算机通过防火墙访问 Internet，则 Toolkit 为您的计算机生成的 IP 地址可能不准确。若要确定要使用的 IP 地址，请咨询您的系统管理员。

   数据库参数组

   （可选）从此下拉列表中，选择要与您的实例关联的数据库参数组。利用数据库参数组，您可以更改实例的默认配置。有关更多信息，请转到 Amazon Relational Database Service 用户指南和本文章。

   您在此对话框上指定设置之后，选择 Next (下一步)。

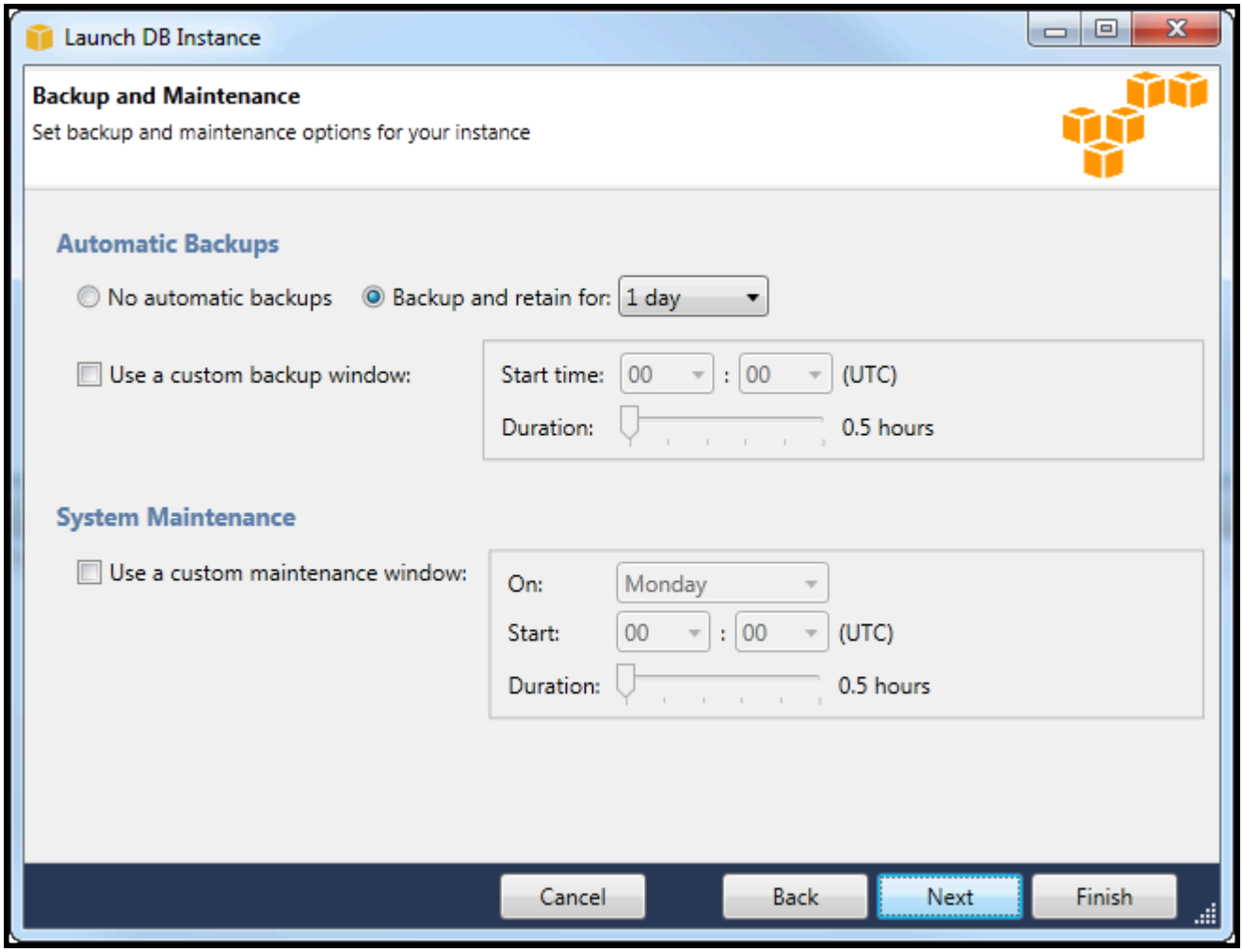


2. 这些区域有：Backup 和维护利用对话框，您可以指定 Amazon RDS 是否应对您的实例进行备份，如果备份，还应指定备份应保留多长时间。您还可以指定备份应发生的时间范围。

   此对话框还使您能够指定是否希望 Amazon RDS 在您的实例上执行系统维护。维护包括常规补丁和次要版本升级。

   您为系统维护指定的时间范围不得与为备份指定的时间范围重叠。

   选择 Next (下一步)。



3. 利用向导中的最后一个对话框，您可以检查实例的设置。如果需要修改设置，请使用 Back (返回) 按钮。如果所有设置均正确，请选择 Launch (启动)。

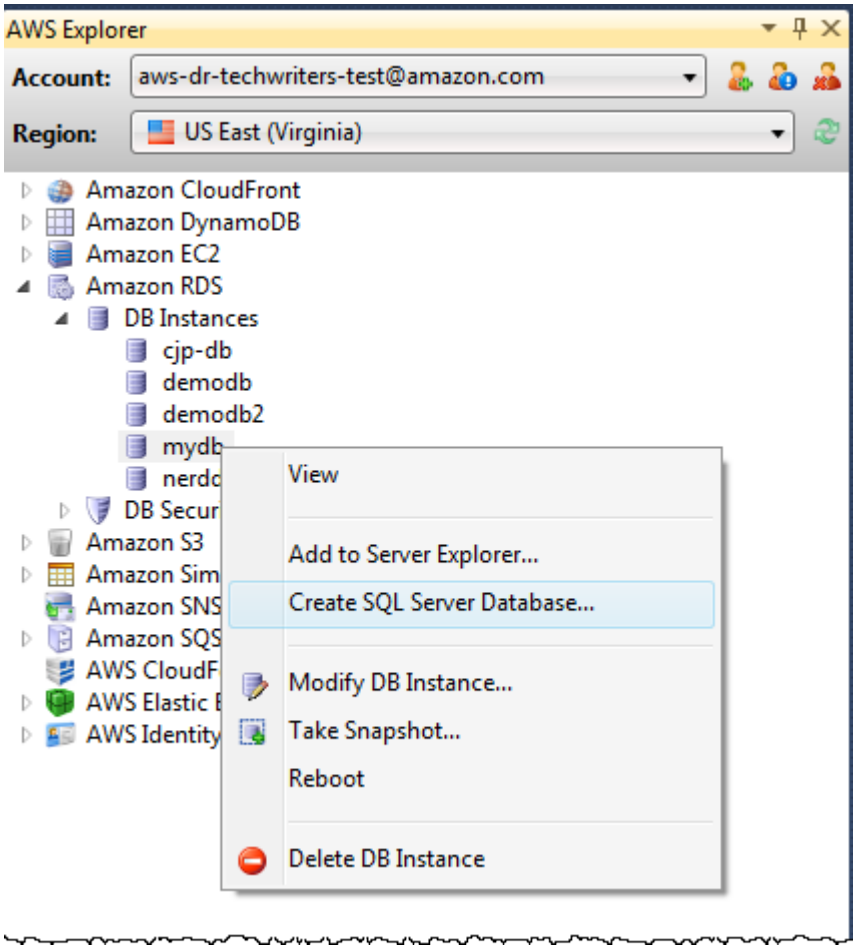
在 RDS 实例中创建 Microsoft SQL Server 数据库

Microsoft SQL Server 的设计方式是在启动 Amazon RDS 实例后，您需要在 RDS 实例中创建 SQL Server 数据库。

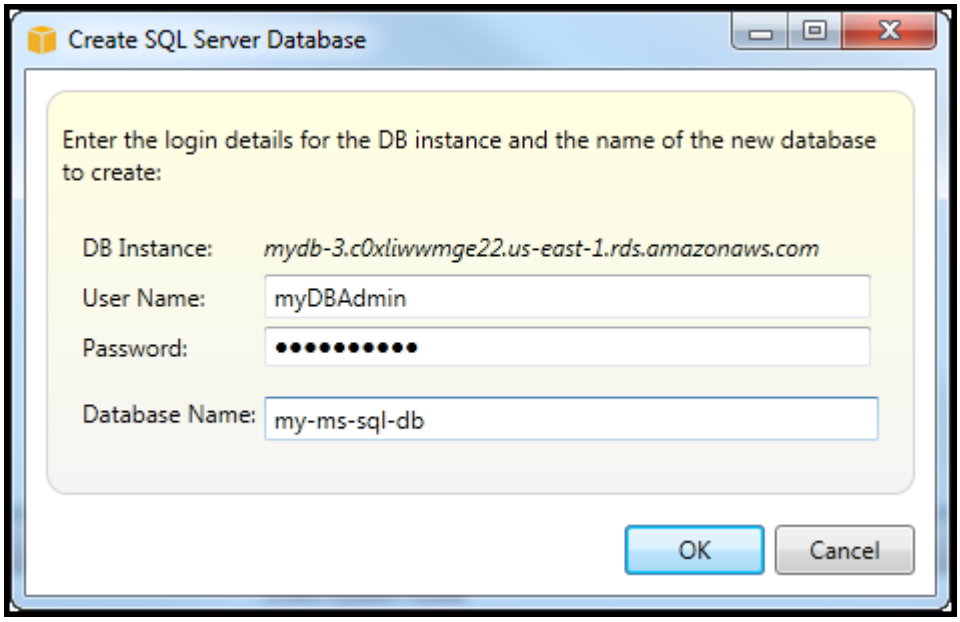
有关如何创建 Amazon RDS 实例的信息，请参阅启动 Amazon RDS 数据库实例.

创建 Microsoft SQL Server 数据库

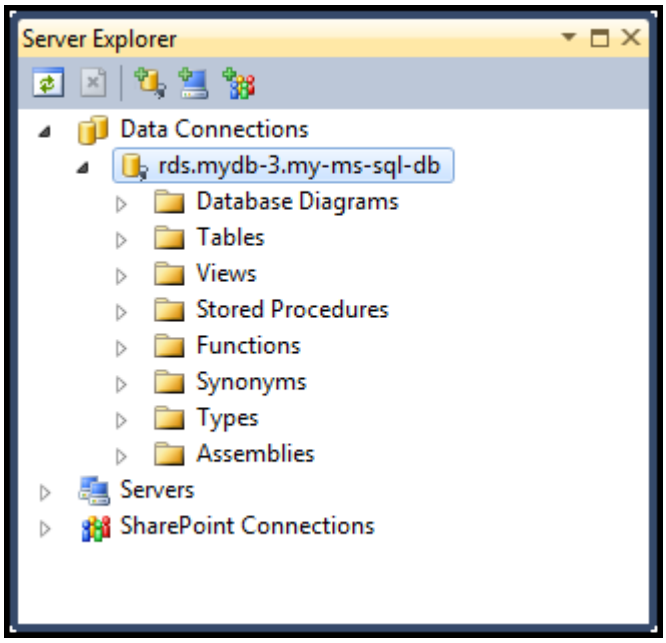
1. InAWSExplorer 中，打开对应于您的 Microsoft SQL Server 的 RDS 实例的节点的上下文（右键单击）菜单，然后选择创建 SQL Server 数据库.



2. 在 Create SQL Server Database (创建 SQL Server 数据库) 对话框中，键入您在创建 RDS 实例时指定的密码，键入 Microsoft SQL Server 数据库的名称，然后选择 OK (确定)。



3. Toolkit for Visual Studio 创建 Microsoft SQL Server 数据库并将其添加到 Visual Studio Server Explorer。



Amazon RDS 安全组

Amazon RDS 安全组使您能够管理对 Amazon RDS 实例的网络访问。利用安全组，您可以使用 CIDR 表示法指定一系列 IP 地址，只有来自这些地址的网络流量才能被您的 Amazon RDS 实例识别。

尽管 Amazon RDS 安全组与 Amazon EC2 安全组的工作方式相似，但它们仍有一些差别。您可将 EC2 安全组添加到 RDS 安全组。作为 EC2 安全组成员的任何 EC2 实例随后都能访问作为 RDS 安全组成员的 RDS 实例。

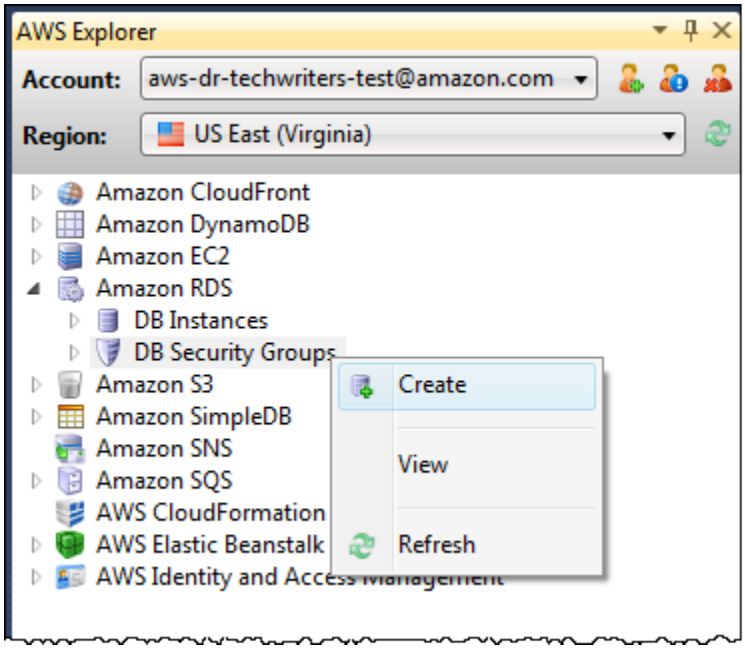
有关 Amazon RDS 安全组的更多信息，请转到RDS 安全组. 有关 Amazon EC2 安全组的更多信息，请转到EC2 用户指南.

创建 Amazon RDS 安全组

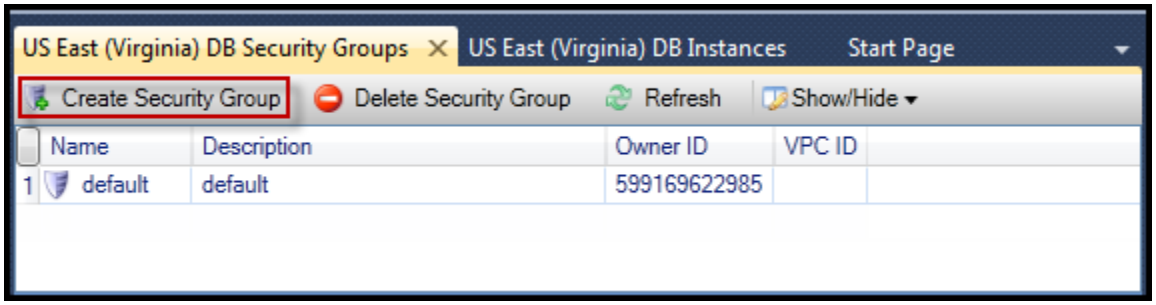
您可以使用适用于 Toolkit for Visual Studio 创建 RDS 安全组。如果您将AWS利用 Toolkit 启动 RDS 实例，该向导将允许您指定要用于您的实例的 RDS 安全组。您可在启动该向导前使用以下步骤创建该安全组。

创建 Amazon RDS 安全组

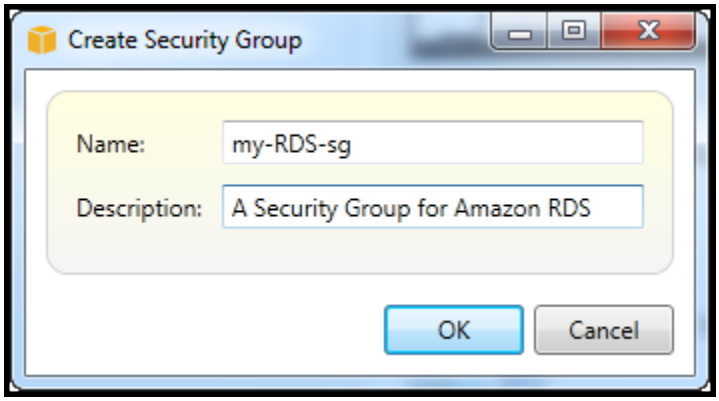
1. InAWS资源管理器，展开Amazon RDS(右键单击) 菜单，打开数据库安全组子节点然后选择Create.



或者，在 Security Groups (安全组) 选项卡上，选择 Create Security Group (创建安全组)。如果此选项卡未显示，则打开 DB Security Groups (数据库安全组) 子节点的上下文（右键单击）菜单，然后选择 View (查看)。



2. 在 Create Security Group (创建安全组) 对话框中，键入该安全组的名称和描述，然后选择 OK (确定)。



设置 Amazon RDS 安全组的访问权限

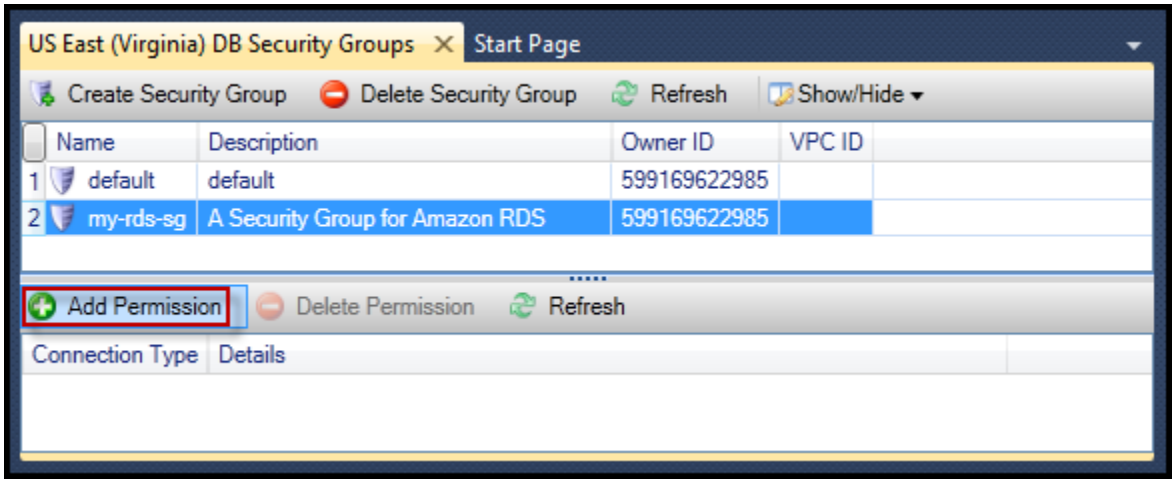
默认情况下，新的 Amazon RDS 安全组不提供网络访问。要启用对使用该安全组的 Amazon RDS 实例的访问，请使用以下步骤设置其访问权限。

设置 Amazon RDS 安全组的访问权限

1. 在 Security Groups (安全组) 选项卡上，从列表视图中选择该安全组。如果您的安全组未显示在列表中，请选择 Refresh (刷新)。如果您的安全组仍未显示在列表中，请验证您正在查看的列表是否正确AWS区域。安全组中的选项卡AWS工具包是特定于区域的。

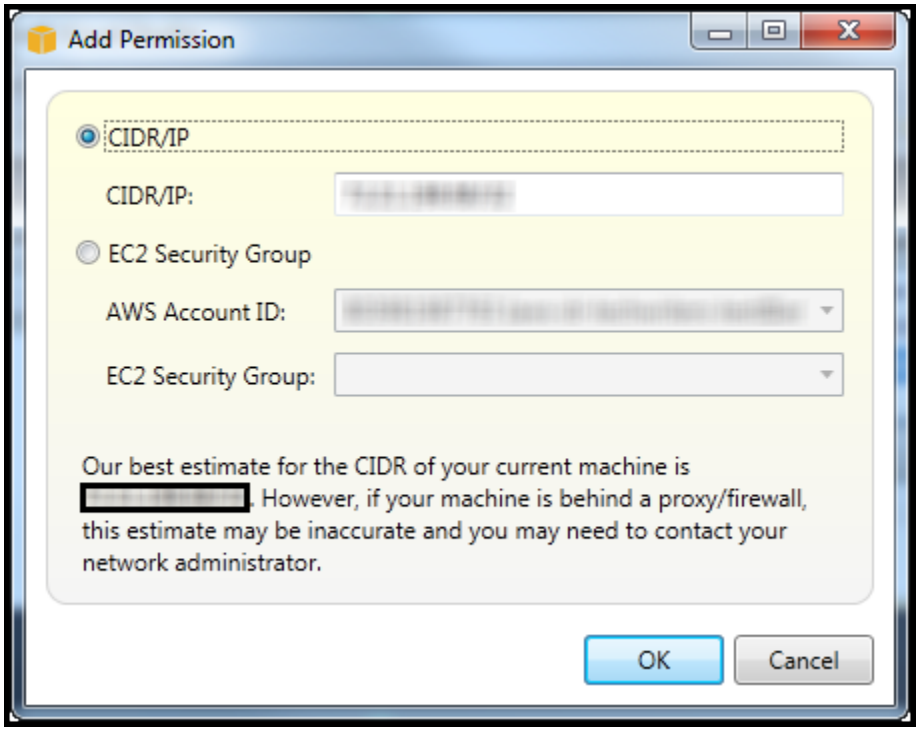
   如果没有安全组出现选项卡，在AWS打开 Location (Explorer) 的上下文 (右键单击) 菜单数据库安全组子节点然后选择查看.

2. 选择 Add Permission。



   添加权限按钮个安全组选项卡

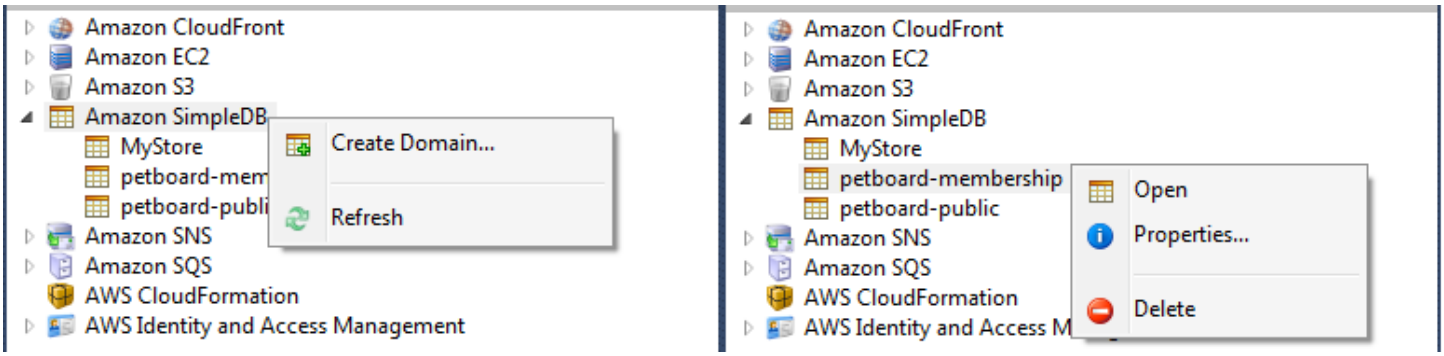
3. 在 Add Permission (添加权限) 对话框中，您可以使用 CIDR 表示法指定可访问您的 RDS 实例的 IP 地址，或者指定可访问您的 RDS 实例的 EC2 安全组。当你选择时EC2 安全组，您可以指定与 AWS 账户您可以访问，或者您可以从下拉列表中选择 EC2 安全组。



这些区域有：AWSToolkit 将尝试确定您的 IP 地址，然后使用相应的 CIDR 规范自动填充对话框。但是，如果您的计算机是通过防火墙访问 Internet 的，则由 Toolkit 确定的 CIDR 可能不准确。

从使用 Amazon SimpleDBAWS探险者

AWSAWS Explorer 显示了与活动项目关联的所有 Amazon SimpleDB 域。AWSaccount. 从AWSAWS Explorer 中，您可以创建或删除 Amazon SimpleDB 域。

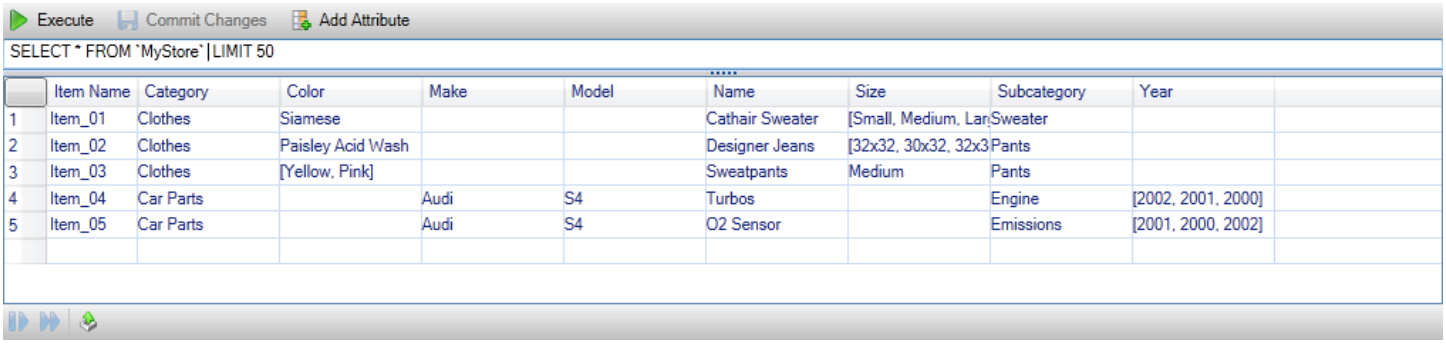


Create, delete, or open Amazon SimpleDB domains associated with your account

执行查询并编辑结果

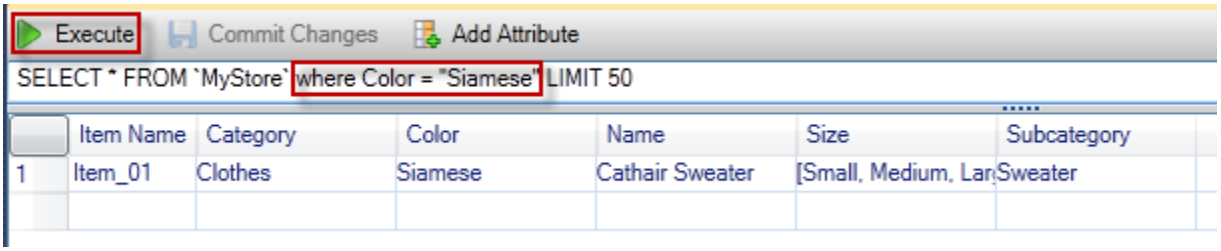
AWSAWS Explorer 还可以显示 Amazon SimpleDB 域的网格视图，您可以从中查看该域中的项目、属性和值。您可以执行查询，以便仅显示该域的一部分项目。通过双击单元格，您可以编辑该项目对应属性的值。您还可以向域添加新属性。

此处显示的域来自于包含的 Amazon SimpleDB 示例。AWS SDK for .NET.



Amazon SimpleDB grid view

要执行查询，请在网格视图顶部的文本框中编辑查询，然后选择 Execute (执行)。将筛选此视图以仅显示与查询匹配的项目。

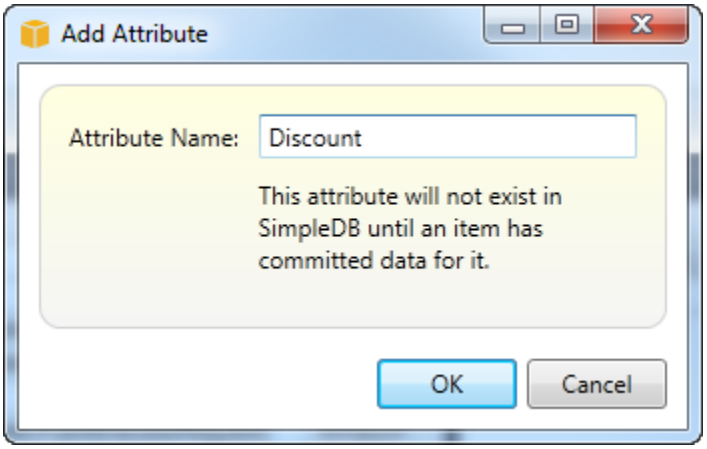


Execute query from AWS Explorer

要编辑与某个属性关联的值，请双击对应的单元格，编辑值，然后选择 Commit Changes (提交更改)。

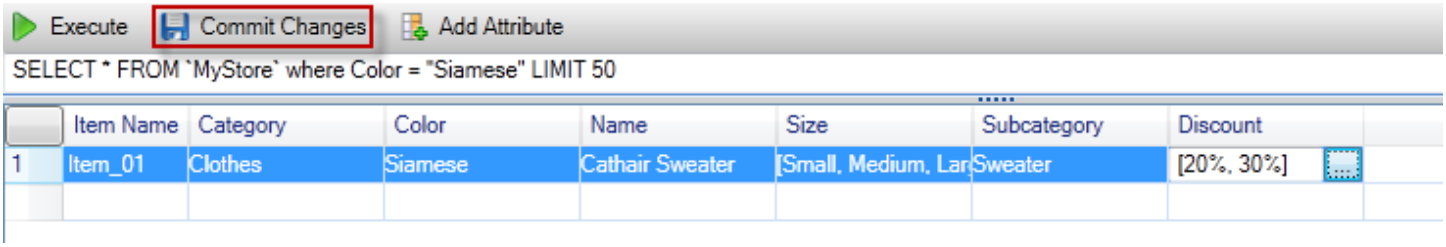
添加属性

要添加属性，请在视图顶部选择 Add Attribute (添加属性)。



Add Attributes dialog box

要使属性成为域的一部分，您必须将属性的一个值添加到至少一个项目，然后选择 Commit Changes (提交更改)。



Commit changes for a new attribute

为查询结果分页

视图底部有三个按钮。

Paginate and export buttons

前两个按钮为查询结果提供分页。要额外显示一页结果，请选择第一个按钮。要额外显示十页结果，请选择第二个按钮。在此上下文中，一个页面等于 100 行或等于 LIMIT 值指定的结果数（如果该值包含在查询中）。

导出到 CSV

最后一个按钮将当前结果导出到 CSV 文件。

从使用 Amazon SQSAWS探险者

Amazon Simple Queue Service (Amazon SQS) 是一项灵活的队列服务，用于实现软件应用程序中的不同执行进程之间的消息传递。Amazon SQS 队列位于AWS基础设施，但传递消息的进程可能位于本地、Amazon EC2 实例上或这些实例的某种组合上。Amazon SQS 非常适合用于协调跨多台计算机的工作分配。

利用 Toolkit for Visual Studio，您可以查看与活动账户关联的 Amazon SQS 队列、创建并删除队列以及通过队列发送消息。（所谓活动账户，我们指的是在中选择的账户。AWSExplorer。）

有关 Amazon SQS 的更多信息，请访问SQS 简介中的AWS文档中）。

创建队列

您可以从创建 Amazon SQS 队列AWSExplorer。队列的 ARN 和 URL 将基于活动账户的账号和您在创建队列时指定的队列名称。

创建队列

1. InAWSExplorer，打开上下文 (右键单击) 菜单，用于Amazon SQS节点，然后选择创建队列.
2. 在 Create Queue (创建队列) 对话框中，指定队列名称、默认可见性超时和默认传递延迟。默认可见性超时和默认传递延迟以秒为单位指定。默认可见性超时是在某个给定进程已获得消息后该消息将对潜在接收进程不可见的时间量。默认传递延迟是从发送消息到消息首次对潜在接收进程可见的时间量。
3. 选择 OK（确定）。新队列将显示为 Amazon SQS 节点下的一个子节点。

删除队列

您可以从中删除现有队列AWSExplorer。如果删除了某个队列，与该队列关联的所有消息都不再可用。

删除队列

1. InAWSExplorer，打开要删除的队列的上下文（右键单击）菜单，然后选择。Delete.

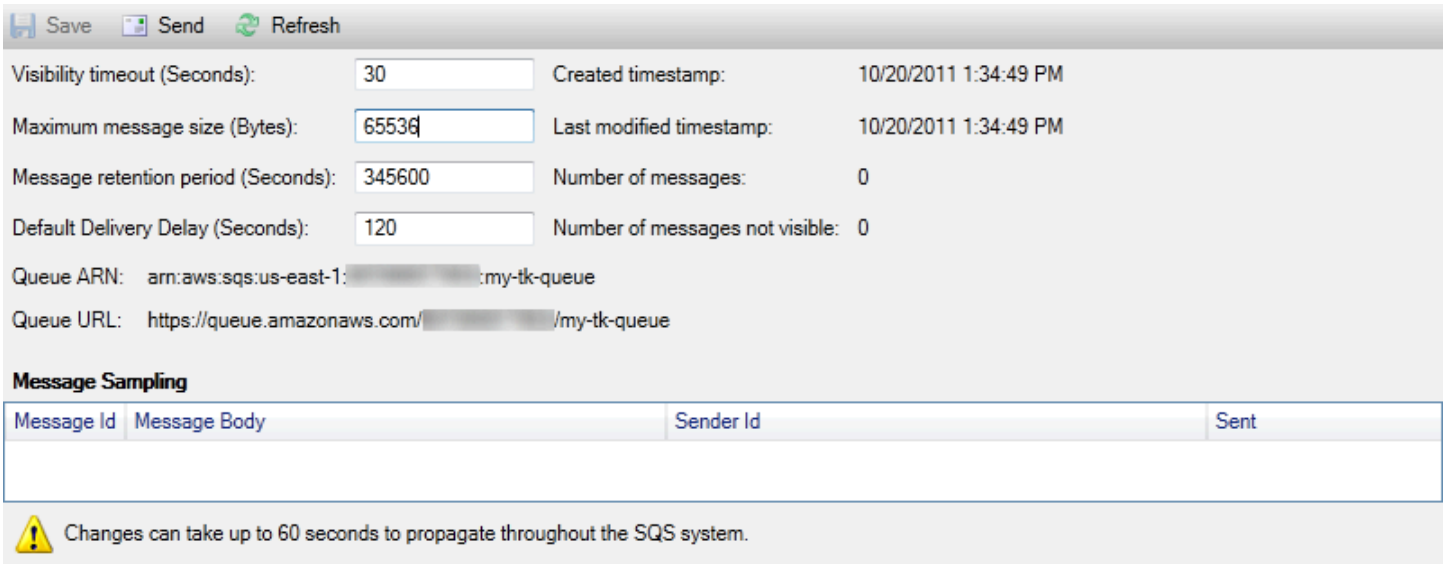
管理队列属性

您可以查看和编辑中显示的任何队列的属性，AWSExplorer。还可以从此属性视图向队列发送消息。

管理队列属性

- InAWSExplorer，打开要管理其属性的队列的上下文（右键单击）菜单，然后选择。查看队列.

  在队列属性视图中，您可以编辑可见性超时、最大消息大小、消息保留期和默认传递延迟。可在发送消息时覆盖默认传递延迟。在以下屏幕截图中，模糊化的文字是队列 ARN 和 URL 的账号组成部分。



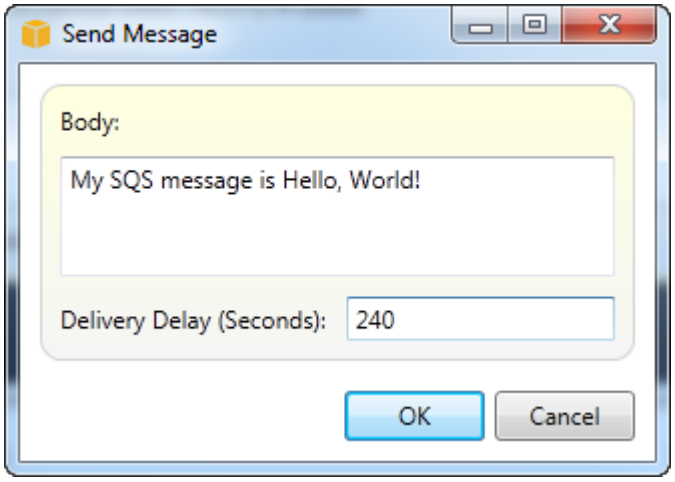
SQS queue properties view

向队列发送消息

在队列属性视图中，您可以向队列发送消息。

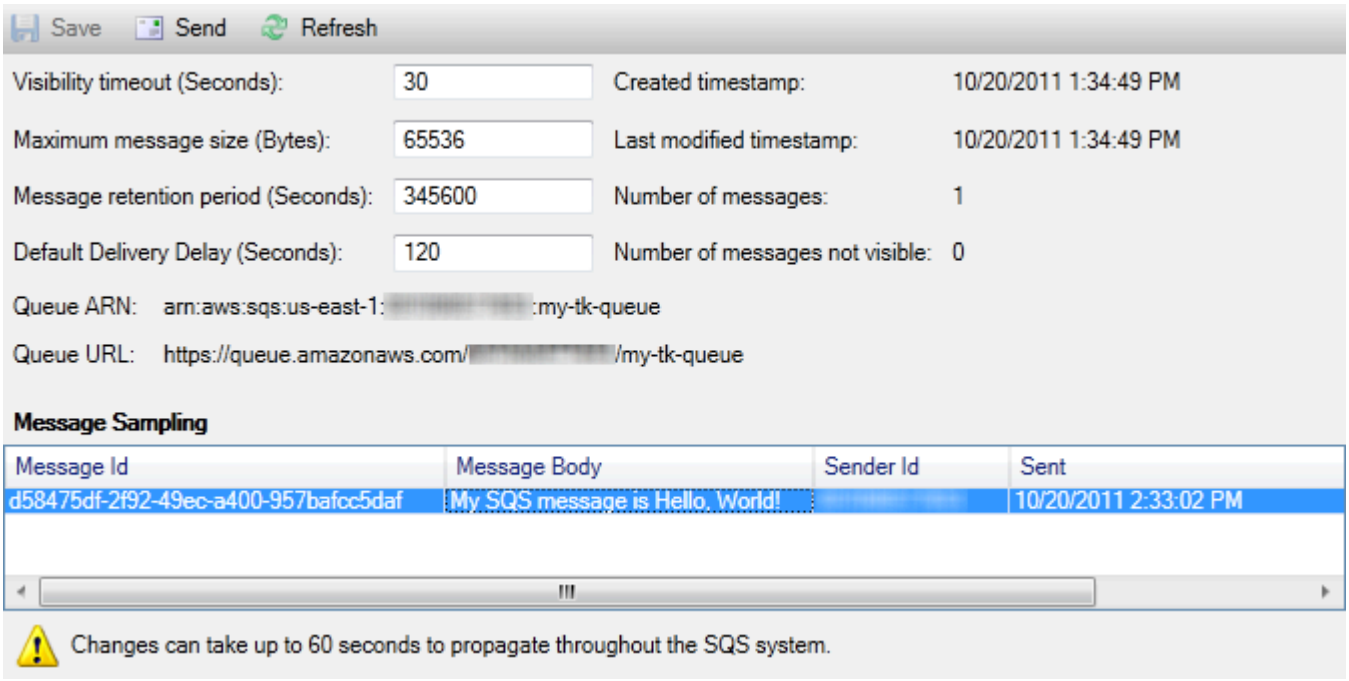
发送邮件

1. 在队列属性视图的顶部，选择 Send (发送) 按钮。
2. 键入消息。（可选）输入将覆盖队列的默认传递延迟的传递延迟。在以下示例中，我们已使用值 240 秒覆盖默认延迟。选择 OK（确定）。



发送消息 dialog box

3. 等待约 240 秒（4 分钟）。消息将显示在队列属性视图的 Message Sampling (消息采样) 部分中。



SQS properties view with sent message

队列属性视图中的时间戳是您选择 Send (发送) 按钮的时间。此时间不包含延迟。因此，消息显示在队列中并对接收方可用的时间可能晚于此时间戳。此时间戳以计算机的本地时间显示。

Identity and Access Management

AWS Identity and Access Management(IAM) 使您能够更安全地管理对您的AWS 账户和资源。使用 IAM，您可以在主用户中创建多个用户 (根) AWS 账户. 这些用户可以有自己的凭证：密码、访问密钥 ID 和私有密钥。但是，所有 IAM 用户都共享单个账号。

您可以通过将 IAM 策略附加到用户来管理每个 IAM 用户的资源访问级别。例如，您可以将策略附加到 IAM 用户，该用户提供了对您的账户中的 Amazon S3 服务和相关资源的访问权限，但未提供对任何其他服务或资源的访问权限。

如要实现更有效的访问管理，您可以创建 IAM 组，即若干用户的集合。当您将一个策略附加到组时，它将影响作为该组成员的所有用户。

除了在用户和组级别管理权限之外，IAM 还支持 IAM 角色的概念。与用户和组一样，您可以将策略附加到 IAM 角色。随后，您可以将 IAM 角色与 Amazon EC2 实例关联。在 EC2 实例上运行的应用程序能够访问。AWS使用 IAM 角色提供的权限。有关将 IAM 角色与 Toolkit 一起使用的更多信息，请参阅创建 IAM 角色。有关 IAM 的更多信息，请转到IAM 用户指南.

创建和配置 IAM 用户

IAM 用户使您可以授予对您的其他访问权限。AWS 账户. 由于您能够将策略附加到 IAM 用户，因此您可以精确地限制 IAM 用户可访问的资源以及他们可对这些资源执行的操作。

作为最佳实践，访问AWS 账户应像 IAM 用户一样执行此操作-甚至账户的所有者也是如此。这确保了在其中一个 IAM 用户的凭证泄露时只需停用这部分凭证。不需要停用或更改账户的根凭证。

在 Toolkit for Visual Studio 中，您可以将策略附加到用户或将用户分配到组，由此向 IAM 用户分配权限。分配到某个组的 IAM 用户将从附加到该组的策略派生其权限。有关更多信息，请参阅创建 IAM 组和将 IAM 用户添加到 IAM 组。

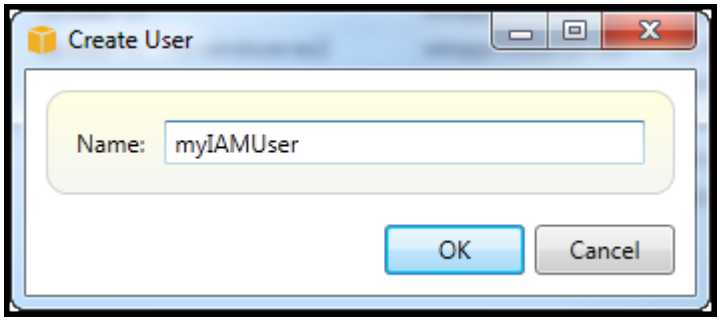
从 Toolkit for Visual Studio 中，您还可以生成AWSIAM 用户的凭证 (访问密钥 ID 和私有密钥)。有关更多信息，请参阅为 IAM 用户生成凭证

Toolkit for Visual Studio 支持 IAM 用户凭证，它们可用于访问服务。AWSExplorer。由于 IAM 用户通常没有对所有 Amazon Web Services 的完全访问权限，因此，中的某些功能AWSExplorer 可能不可用。如果您使用AWS如要在有效账户是 IAM 用户时更改资源，然后将有效账户切换为根账户，则在您刷新中的视图之前，更改可能不可见。AWSExplorer。要刷新该视图，请选择刷新 () 按钮。

有关如何从AWS Management Console，转到使用(在 IAM 用户指南中)。

创建 IAM 用户

1. InAWS资源管理器，展开AWS Identity and Access Management节点，打开的上下文（右键单击）菜单用户然后选择创建用户.
2. 在创建用户在对话框中，键入 IAM 用户的名称并选择确定. 这是 IAM友好的名字. 有关 IAM 用户的名称限制的信息，请转到IAM 用户指南.



Create an IAM user

新用户将显示为下的一个子节点。用户在AWS Identity and Access Management节点。

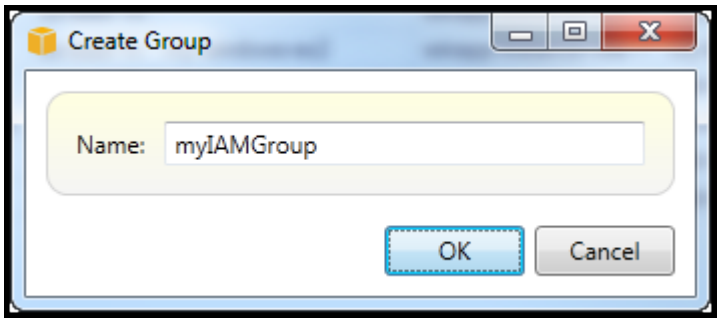
有关如何创建策略并将其附加到用户的信息，请参阅创建 IAM 策略。

创建 IAM 组

组提供了一个将 IAM 策略应用于用户集合的方法。有关如何管理 IAM 用户和组的信息，请转到。使用(在 IAM 用户指南中)。

创建 IAM 组

1. InAWS在 ExplorerIdentity and Access Management打开的上下文（右键单击）菜单Groups然后选择创建组.
2. 在创建组在对话框中，键入 IAM 组的名称并选择确定.



Create IAM group

新的 IAM 组将出现在Groups的子节点Identity and Access Management.

有关创建策略并将其附加到 IAM 组的信息，请参阅。创建 IAM 策略.

将 IAM 用户添加到 IAM 组

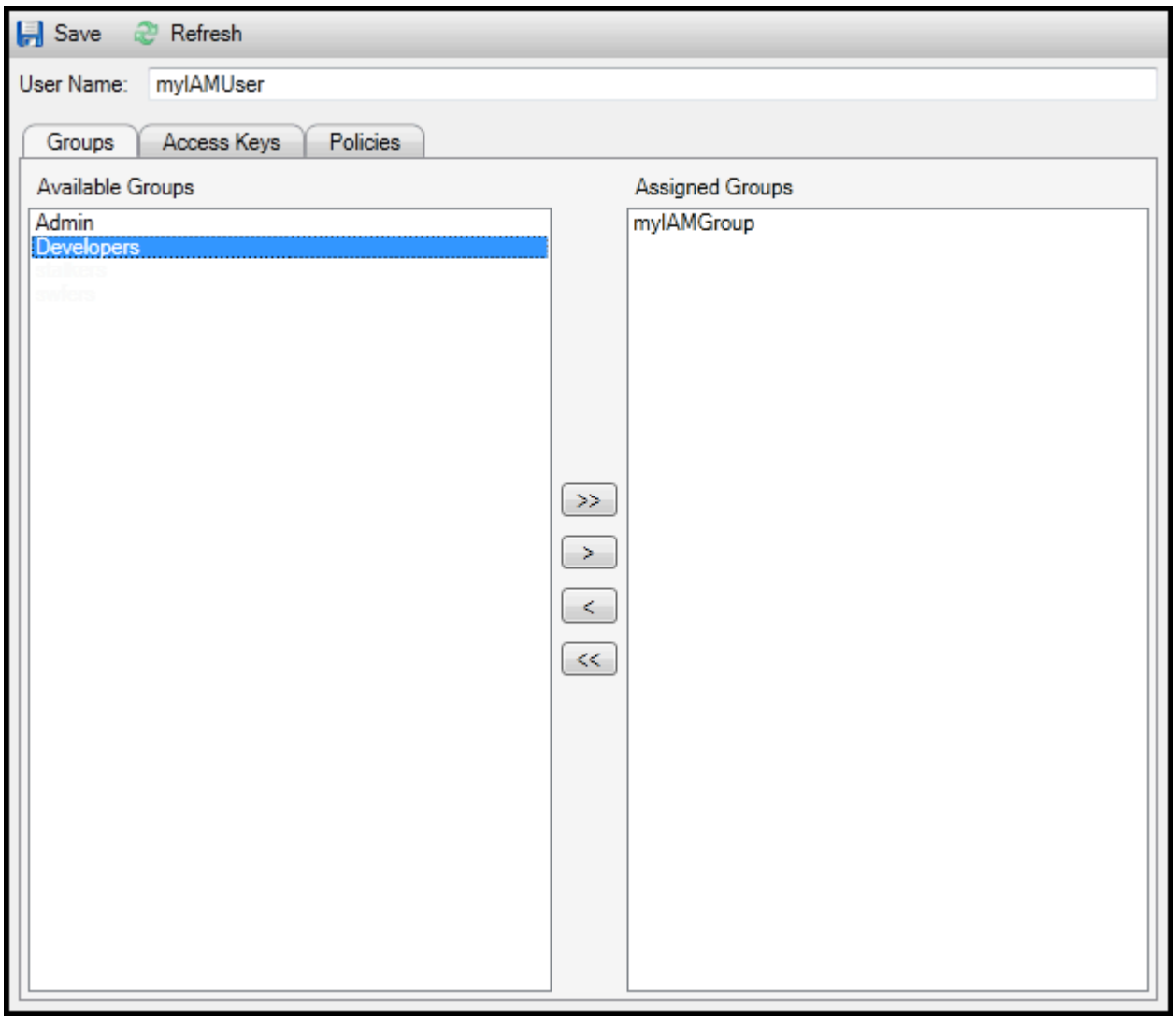
作为 IAM 组的成员的 IAM 用户将从附加到该组的策略派生访问权限。IAM 组的用途是简化对一系列 IAM 用户的权限的管理。

有关附加到某个 IAM 组的策略如何与附加到作为该 IAM 组的成员的 IAM 用户的策略交互的信息，请转到。《IAM 用户指南》中的管理 IAM 策略.

InAWSExplorer，您可以从用户子节点，而不是Groups子节点。

将 IAM 用户添加到 IAM 组

1. InAWS在 ExplorerIdentity and Access Management打开的上下文（右键单击）菜单用户然后选择编辑.



Assign an IAM user to a IAM group

2. 的左侧窗格Groups选项卡显示可用的 IAM 组。右侧窗格显示了已包含指定 IAM 用户的组。

   要将 IAM 用户添加到某个组，请在左侧窗格中，选择 IAM 组，然后选择>按钮。

   要从某个组中删除 IAM 用户，请在右侧窗格中选择该 IAM 组，然后选择<按钮。

   要将 IAM 用户添加到所有 IAM 组，请选择>>按钮。同样，要从所有组中删除 IAM 用户，请选择<<按钮。

   要选择多个组，请按顺序选择它们。您不需要按住 Ctrl 键。要从您的选项中清除某个组，只需再次选择它。

3. 当您完成向 IAM 组分配 IAM 用户时，选择。Save（保存）.

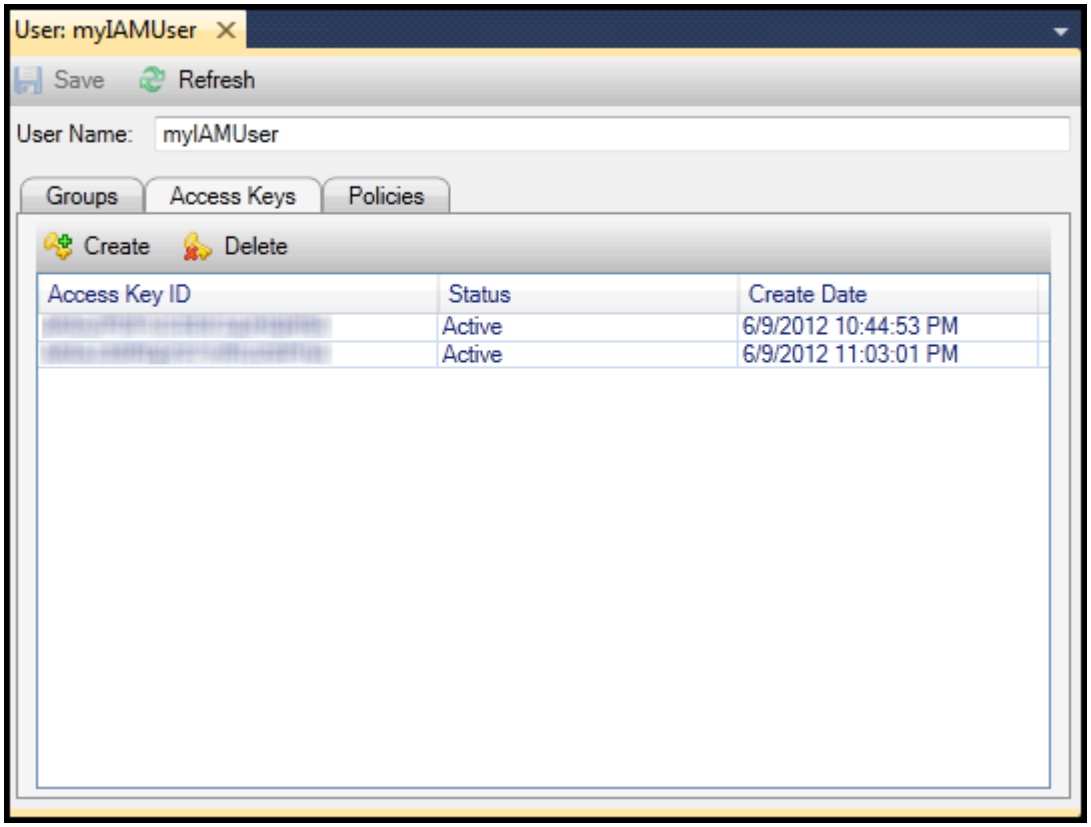
为 IAM 用户生成凭证

利 Toolkit for Visual Studio，您可以生成用于对的访问密钥 ID 和私有密钥。AWS. 还可将这些密钥指定为通过 Toolkit 访问 Amazon Web Services。有关如何指定与 Toolkit 配合使用的凭证的更多信息，请参阅凭证。有关如何安全处理凭证的更多信息，请参阅。管理的最佳实践AWS访问密钥.

Toolkit 无法用于为 IAM 用户生成密码。

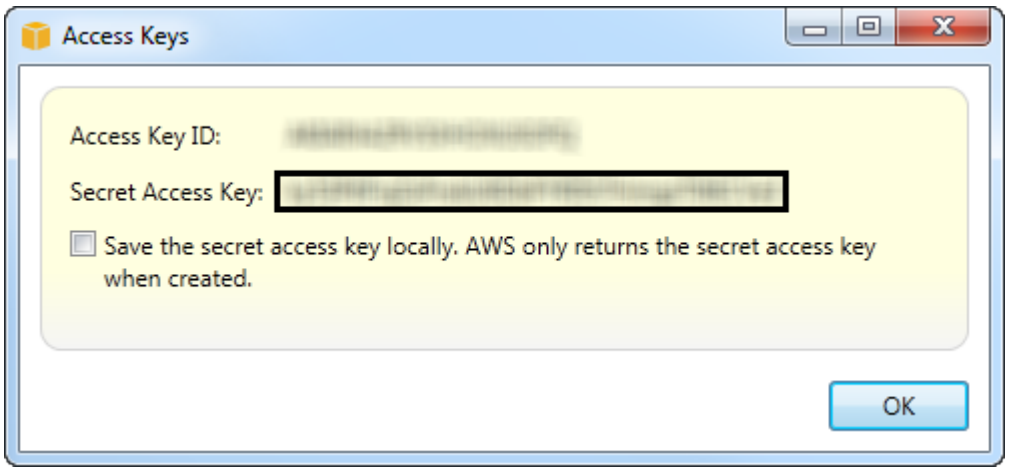
为 IAM 用户生成凭证

1. InAWSExplorer，打开 IAM 用户的上下文 (右键单击) 菜单，然后选择。编辑.



2. 要生成凭证，请在 Access Keys (访问密钥) 选项卡上，选择 Create (创建)。

   您只能为每个 IAM 用户生成两组凭证。如果您已经有两组凭证并需要再创建一组凭证，则必须删除现有的凭证组之一。

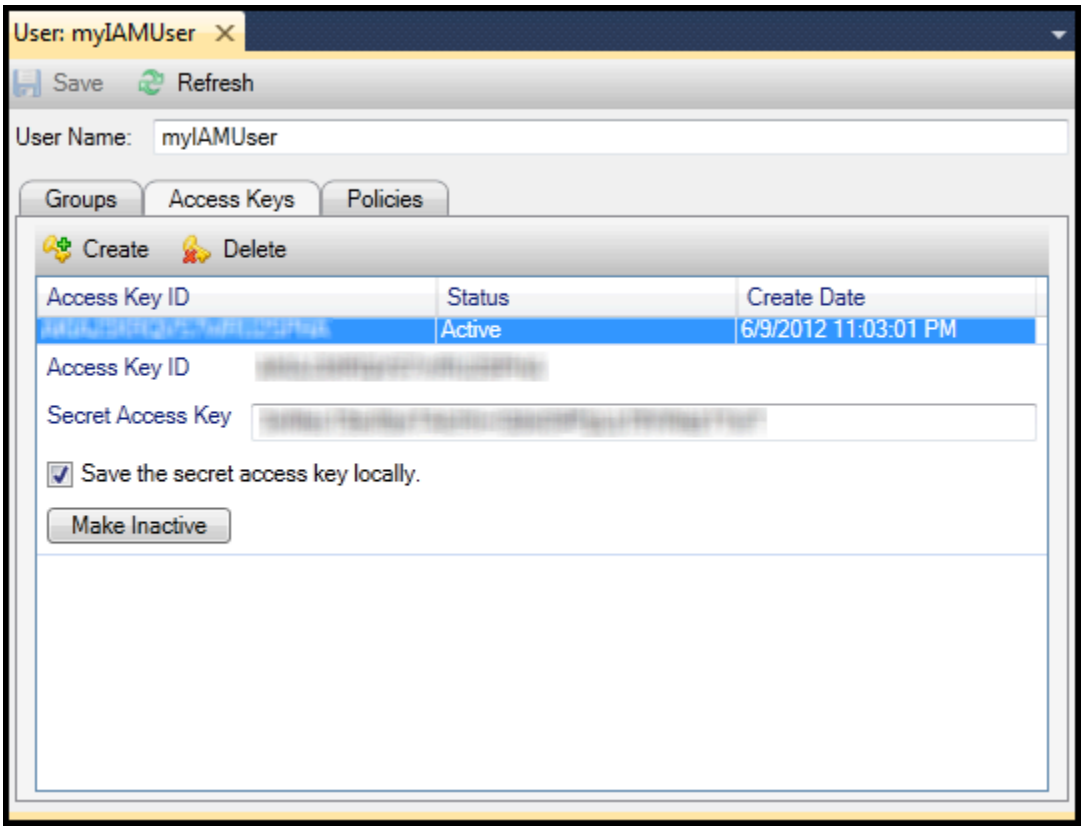


reate credentials for IAM user

如果您希望 Toolkit 将您的秘密访问密钥的加密副本保存到本地硬盘，请选择。在本地保存私有访问密钥。AWS在创建密钥时，才返回私有访问密钥. 您还可以从该对话框中复制秘密访问密钥并将其保存在安全的位置。

3. 选择 OK（确定）。

生成凭证后，您可以从 Access Keys (访问密钥) 选项卡查看它们。如果您选择了让 Toolkit 本地保存私有密钥的选项，该密钥将显示在此处。



Create credentials for IAM user

如果您自行保存了私有密钥并且希望 Toolkit 保存它，请在 Secret Access Key (秘密访问密钥) 框中，键入秘密访问密钥，然后选择 Save the secret access key locally (在本地保存秘密访问密钥)。

要停用凭证，请选择 Make Inactive (转为非活动)。（如果您怀疑凭证已泄露，则可能会这样做。 如果您得到“凭证是安全的”的保证，则可以重新激活它们。）

创建 IAM 角色

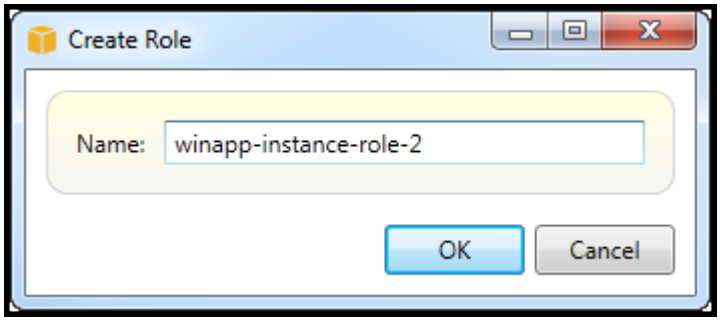
Toolkit of Visual Studio 支持创建和配置 IAM 角色。和使用用用用用户和组一样，您可以将策略附加到 IAM 角色。随后，您可以将 IAM 角色与 Amazon EC2 实例关联。与 EC2 实例的关联通过实例配置文件 来处理，后者是角色的逻辑容器。在 EC2 实例上运行的应用程序将自动获得由与 IAM 角色关联的策略指定的访问级别。即使在应用程序尚未指定其他时也是如此。AWS凭证。

例如，您可以创建一个角色并将仅限制访问 Amazon S3 的策略附加到该角色。将此角色与 EC2 实例关联，随后您可以在该实例上运行一个应用程序，该应用程序将能够访问 Amazon S3，但无法访问任何其他服务或资源。这种方法的优点是您无需关注安全的传输和存储。AWSEC2 实例上的证书。

有关 IAM 角色的更多信息，请转到《IAM 用户指南》中的使用 IAM 角色. 有关访问程序的示例AWS使用与 Amazon EC2 实例关联的 IAM 角色，转到AWS的开发人员指南Java、.NET、PHP和 Ruby (使用 IAM 设置凭证、创建 IAM 角色, 和使用 IAM 策略)。

创建 IAM 角色

1. InAWS在 ExplorerIdentity and Access Management打开的上下文（右键单击）菜单角色然后选择创建角色.
2. 在创建角色在对话框中，键入 IAM 角色的名称并选择确定.



Create IAM role

新的 IAM 角色将出现在角色在Identity and Access Management.

有关如何创建策略并将其附加到角色的信息，请参阅创建 IAM 策略。

创建 IAM 策略

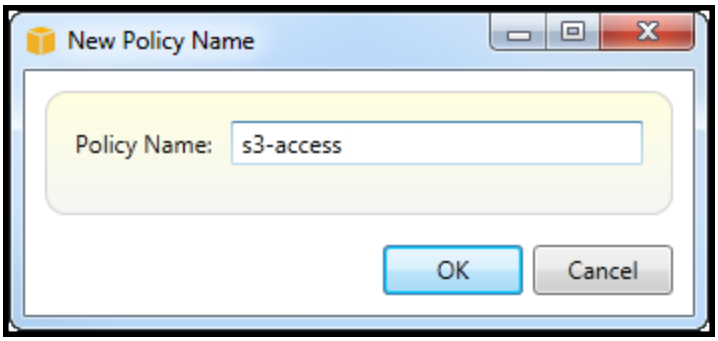
策略是 IAM 的基础。策略可以与 IAM 关联实体例如用户、组或角色。策略指定为用户、组或角色启用的访问级别。

创建 IAM 策略

InAWS资源管理器，展开AWS Identity and Access Management节点，然后展开实体类型的节点 (Groups、角色，或者用户) 你将附加该政策。例如，打开 IAM 角色的上下文菜单并选择编辑.

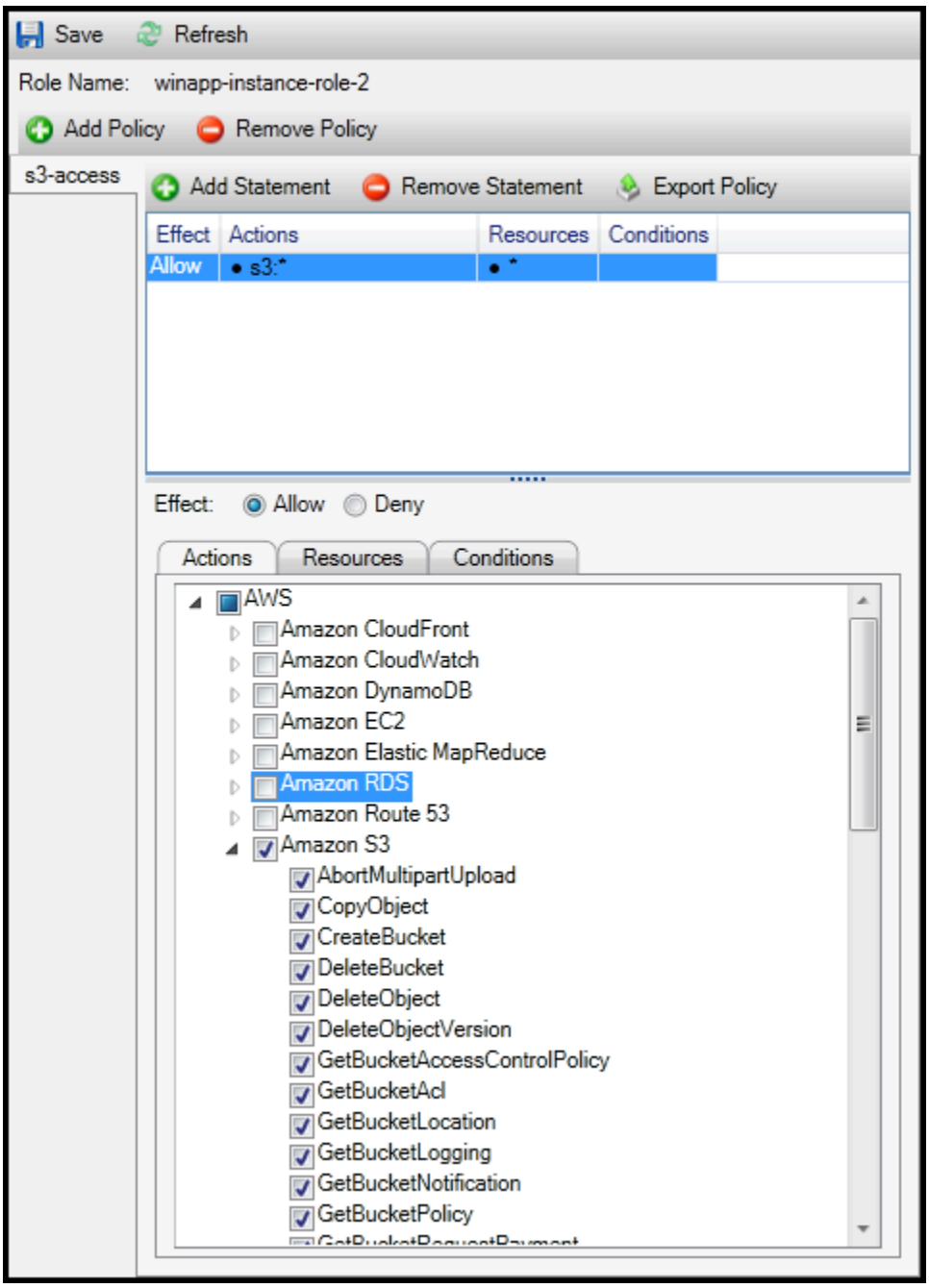
与角色关联的选项卡将出现在AWSExplorer。选择 Add Policy (添加策略) 链接。

在 New Policy Name (新策略名称) 对话框中，键入策略的名称（例如，s3-access）。



New Policy Name dialog box

在策略编辑器中，添加策略声明以指定要提供给角色的访问级别（本例中为与策略关联的 winapp-instance-role-2）。在本示例中，策略提供了对 Amazon S3 的完全访问权限，但未提供对任何其他资源的访问权限。



Specify IAM policy

如要实现更精确的访问控制，您可以在策略编辑器中展开子节点以允许或拒绝与 Amazon Web Services 关联的操作。

当您编辑完策略时，请选择 Save (保存) 链接。

AWS Lambda

使用开发和部署基于.NET 内核的 C# Lambda 函数。 AWS Toolkit for Visual Studio AWS Lambda 是一项计算服务，允许您在不预置或管理服务器的情况下运行代码。Visual Studio 工具包包括 Visual Studio 的 AWS Lambda .NET 核心项目模板。

有关的更多信息 AWS Lambda，请参阅 AWS Lambda 开发人员指南。

有关.NET 核心的更多信息，请参阅 Microsoft .NET 核心指南。有关适用于 Windows、macOS 和 Linux 平台的 .NET 内核先决条件和安装说明，请参阅 .NET 内核下载。

以下主题介绍如何 AWS Lambda 使用适用于 Visual Studio 的工具包。

主题

基础 AWS Lambda 项目

你可以在中使用微软.NET Core 项目模板创建 Lambda 函数。 AWS Toolkit for Visual Studio

创建 Visual Studio .NET Core Lambda 项目

您可以使用 Lambda-Visual Studio 模板和蓝图来帮助加快项目初始化速度。Lambda 蓝图包含预先编写的函数，可简化灵活项目基础的创建。

> **Note**
>
> Lambda 服务对不同的包裹类型有数据限制。有关数据限制的详细信息，请参阅 Lambda 用户指南中的 Lambda AWS 配额主题。

在 Visual Studio 中创建 Lambda 项目

1. 从 Visual Studio 中展开 “文件” 菜单，展开 “新建”，然后选择 “项目”。

2. 在 “新建项目” 对话框中，将 “语言”、“平台” 和 “项目类型” 下拉框设置为 “全部”，然后aws lambda在 “搜索” 字段中键入。选择 AWS Lambda 项目（.NET 核心-C#）模板。
3. 在 “名称” 字段中输入**AWSLambdaSample**，指定所需的文件位置，然后选择 “创建” 继续。
4. 在 “选择蓝图” 页面中，选择空函数蓝图，然后选择 “完成” 创建 Visual Studio 项目。

复查项目文件

有两个项目文件需要复查：aws-lambda-tools-defaults.json 和 Function.cs。

以下示例显示了该aws-lambda-tools-defaults.json文件，该文件是作为项目的一部分自动创建的。您可以使用此文件中的字段来设置生成选项。

> **Note**
>
> Visual Studio 中的项目模板包含许多不同的字段，请注意以下内容：
>
> - 函数处理程序：指定 Lambda 函数运行时运行的方法
> - 在函数处理程序字段中指定值会在 “发布” 向导中预填充该值。
> - 如果重命名函数、类或程序集，则还需要更新aws-lambda-tools-defaults.json文件中的相应字段。

```
{
  "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,
```

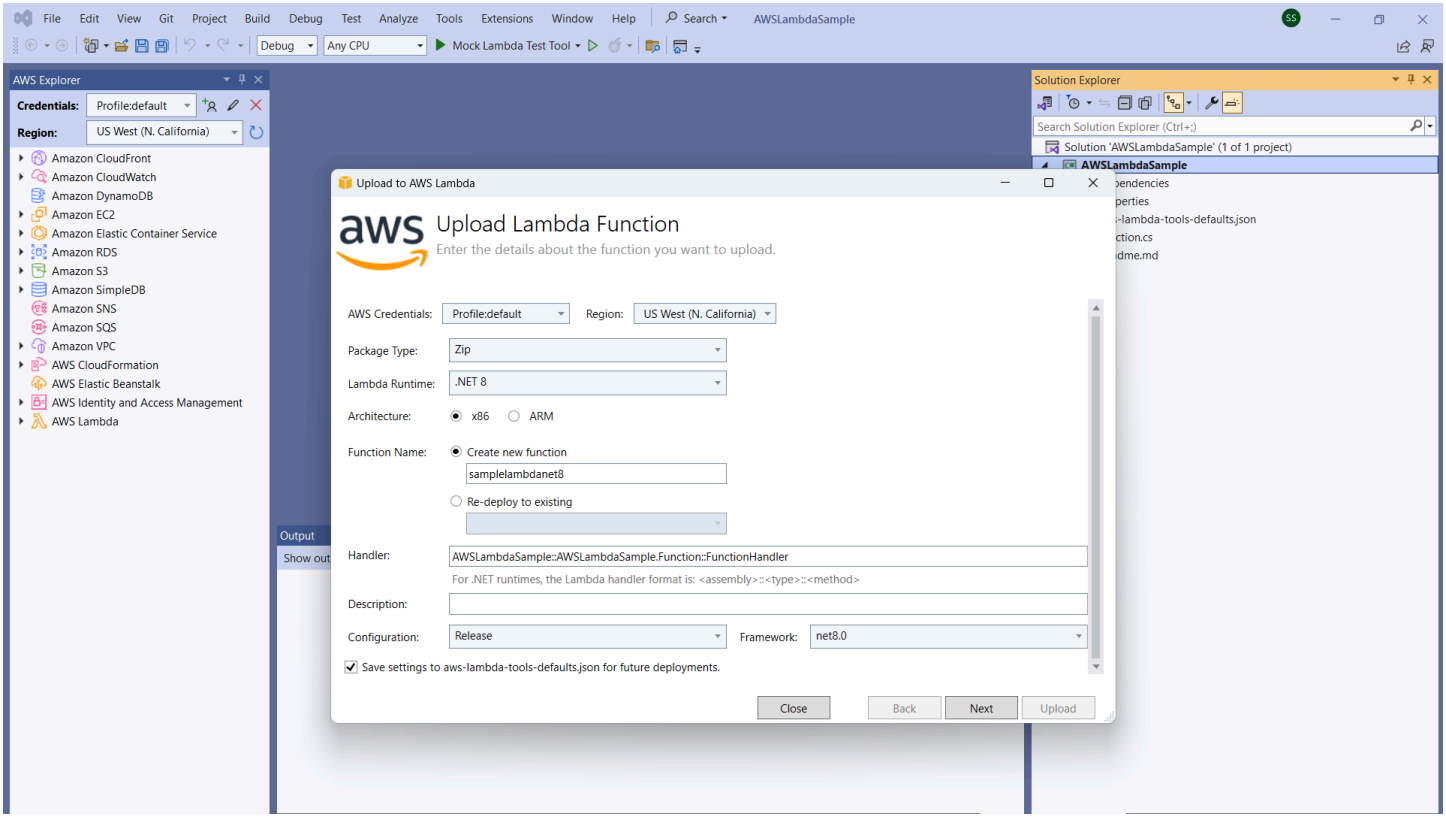
```
  "function-handler": "AWSLambdaSample::AWSLambdaSample.Function::FunctionHandler"
}
```

检查 Function.cs 文件。Function.cs 定义了要作为 Lambda 函数公开的 c# 函数。这里的 FunctionHandler 是在运行 Lambda 函数时运行的 Lambda 功能。在此项目中，定义了一个函数：FunctionHandler，它在输入文本上调用 ToUpper()。

您的项目现在可发布到 Lambda。

发布到 Lambda

以下过程和图像演示了如何使用将您的函数上传到 Lambda。 AWS Toolkit for Visual Studio



将您的函数发布到 Lambda

1. 展开 “视图” 并选择 “ AWS 资源管理器”，即可导航到AWS 资源管理器。
2. 在解决方案资源管理器中，打开（右键单击）要发布的项目的快捷菜单，然后选择发布到 AWS Lambda 以打开上传 Lambd a 函数窗口。
3. 在上传 Lambda 函数窗口中，填写以下字段：

a. Package 类型：选择**Zip**。作为构建过程的结果，将创建一个 ZIP 文件并将其上传到 Lambda。或者，您可以选择 Pack age Type **Image**。教程：创建 Docker 镜像的基本 Lambda 项目描述了如何使用 Package 类型进行发布。 **Image**

b. Lambda 运行时：从下拉菜单中选择您的 Lambda 运行时。

c. 架构：根据您的首选架构选择径向。

d. 函数名称：为 “创建新函数” 选择径向，然后输入 Lambda 实例的显示名称。 AWS 浏览器和 AWS Management Console 显示器都会引用此名称。

e. 处理程序：使用此字段指定函数处理程序。例如：**AWSLambdaSample::AWSLambdaSample.Function::FunctionHandler**。

f. （可选）描述：在中输入要与您的实例一起显示的描述性文本。 AWS Management Console

g. 配置：从下拉菜单中选择您的首选配置。

h. 框架：从下拉菜单中选择您的首选框架。

i. 保存设置：选中此框可将当前设置保存为默认设置，以`aws-lambda-tools-defaults.json`备将来部署时使用。

j. 选择 “下一步” 进入高级函数详细信息窗口。

4. 在 “高级函数详细信息” 窗口中，填写以下字段：

a. 角色名称：选择与您的账户关联的角色。该角色为函数中的代码发出的任何 AWS 服务调用提供临时证书。如果您没有角色，请在下拉选择器中滚动找到 “基于 AWS 托管策略的新角色”，然后选择AWSLambdaBasicExecutionRole。此角色的访问权限最低。

> **Note**
>
> 您的账户必须拥有运行 IAM ListPolicies 操作的权限，否则角色名称列表将为空，您将无法继续。

b. （可选）如果您的 Lambda 函数访问亚马逊 VPC 上的资源，请选择子网和安全组。

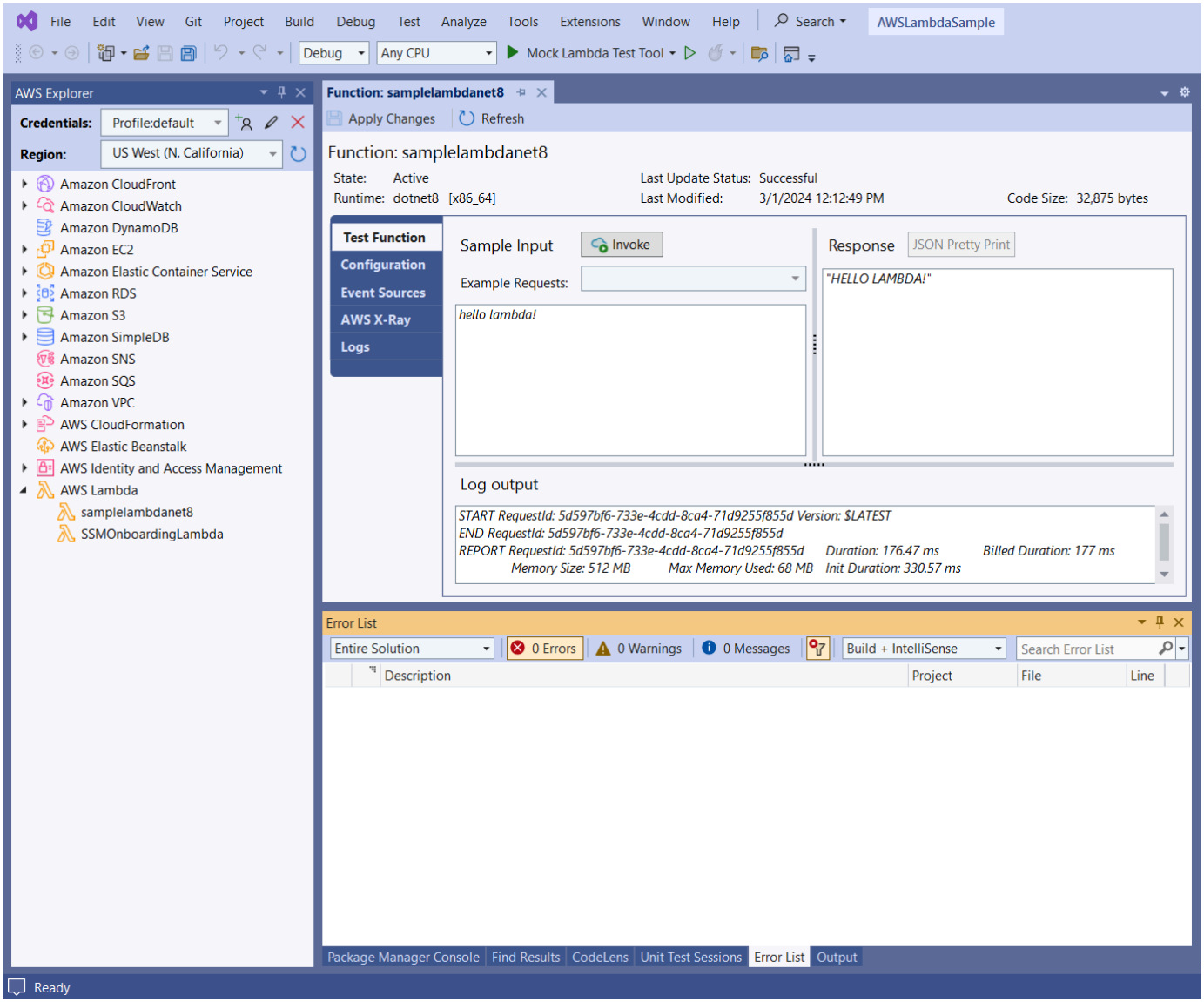
c. （可选）设置 Lambda 函数所需的任何环境变量。这些密钥由免费的默认服务密钥自动加密。或者，您可以指定需要付费的 AWS KMS 密钥。KMS 是一项托管服务，可使用它创建和控制用于对数据进行加密的加密密钥。如果您有 AWS KMS 密钥，则可以从列表中将其选中。

5. 选择 “上传” 以打开 “上传函数” 窗口并开始上传过程。

> **Note**
>
> 函数上传到时，将显示 “上传函数” 页面 AWS。要使向导在上传后保持打开状态以便查看报告，请在上传完成之前取消选中表单底部的在成功完成时自动关闭向导。
> 函数上传后，您的 Lambda 函数就会上线。函数：视图页面将打开并显示您的新 Lambda 函数的配置。

6. 在测试函数选项卡中，在文本输入字段`hello lambda!`中输入，然后选择调用以手动调用您的 Lambda 函数。您的文本将显示在 “回复” 选项卡中，并转换为大写。

> **Note**
>
> 您可以随时重新打开函数：视图，方法是双击位于 AWS 各区服务浏览器中 AWS Lambda 节点下的已部署实例。



7. （可选）要确认您已成功发布您的 Lambda 函数，请登录 AWS Management Console 并选择 Lambda。控制台会显示您发布的所有 Lambda 函数，包括您刚刚创建的函数。

清理

如果您不打算继续使用此示例进行开发，请删除您部署的函数，这样就不会为账户中未使用的资源付费。

> **Note**
>
> Lambda 会自动为您监控 Lambda 函数，并通过亚马逊报告指标。 CloudWatch要监控您的函数并对其进行故障排除，请参阅 AWS Lambda 开发者指南中的使用 Amazon 进行故障排除和监控 AWS Lambda 函数 CloudWatch的主题。

删除函数

1. 从AWS 资源管理器中展开AWS Lambda节点。
2. 右键单击已部署的实例，然后选择删除。

创建 Docker 映像的基本 AWS Lambda 项目

你可以使用 Visual Studio 的 Toolkit for Visual Studio 将你的 AWS Lambda 函数部署为 Docker 镜像。使用 Docker，您可以更好地控制自己的运行时间。例如，您可以选择自定义运行时，例如.NET 8.0。您可以像部署任何其他容器映像一样部署 Docker 映像。本教程与教程：基本 Lambda 项目非常相似，但有两个区别：

- 项目中包含一个 Dockerfile。
- 选择了备用发布配置。

有关 Lambda 容器映像的信息，请参阅《AWS Lambda 开发人员指南》中的 Lambda 部署包。

有关使用 Lambda 的更多信息 AWS Toolkit for Visual Studio，请参阅本用户指南 AWS Toolkit for Visual Studio主题中的使用 AWS Lambda 模板。

创建 Visual Studio .NET Core Lambda 项目

您可以使用 Lambda Visual Studio 模板和蓝图来帮助加快项目初始化的速度。Lambda 蓝图包含预先编写的函数，可简化灵活项目基础的创建。

创建 Visual Studio .NET Core Lambda 项目

1. 从 Visual Studio 中展开 “文件” 菜单，展开 “新建”，然后选择 “项目”。
2. 在 “新建项目” 对话框中，将 “语言”、“平台” 和 “项目类型” 下拉框设置为 “全部”，然后**aws lambda**在 “搜索” 字段中键入。选择 AWS Lambda 项目（.NET 核心-C#）模板。
3. 在 “项目名称” 字段中输入**AWSLambdaDocker**，指定您的文件位置，然后选择 “创建”。

4. 在 “选择蓝图” 页面上，选择 .NET 8（容器映像）蓝图，然后选择 “完成” 创建 Visual Studio 项目。您可以现在复查项目的结构和代码。

查看项目文件

以下各节探讨 .NET 8（容器镜像）蓝图创建的三个项目文件：

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

1. Dockerfile

A `Dockerfile` 执行三个主要操作：

- `FROM`：建立用于此映像的基础映像。此基础映像包含 .NET 运行时系统、Lambda 运行时系统以及为 Lambda .NET 进程提供入口点的 shell 脚本。
- `WORKDIR`：将图像的内部工作目录建立为`/var/task`。
- `COPY`：会将生成过程中生成的文件从其本地位置复制到映像的工作目录中。

以下是您可以指定的可选`Dockerfile`操作：

- `ENTRYPOINT`：基础映像已经包含一个`ENTRYPOINT`，即启动映像时执行的启动过程。如要指定自己的入口点，可以覆盖该基本入口点。
- `CMD`: 指示您要执行 AWS 哪个自定义代码。这要求自定义方法有一个完全限定名称。此行可以直接包含在 Dockerfile 中，也可以在发布过程中指定。

```
# 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（容器镜像）蓝图创建的 Dockerfile 的示例。

```
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

该`aws-lambda-tools-defaults.json`文件用于为 Visual Studio 的 Toolkit for Visual Studio 部署向导和.NET Core CLI 指定默认值。以下列表描述了可在`aws-lambda-tools-defaults.json`文件中设置的字段。

- `profile`: 设置您的 AWS 个人资料。
- `region`：设置您的资源存储 AWS 区域。
- `configuration`：设置用于发布函数的配置。
- `package-type`：将部署包类型设置为容器映像或.zip 文件存档。
- `function-memory-size`：设置函数的内存分配（以 MB 为单位）。
- `function-timeout`: 超时是 Lambda 函数可以运行的最大时间（以秒为单位）。您可以以 1 秒为增量调整此值，最大值为 15 分钟。
- `docker-host-build-output-dir`：设置生成过程的输出目录，该目录与中的指令相关联。`Dockerfile`
- `image-command`: 是您的方法（您希望 Lambda 函数运行的代码）的完全限定名称。语法如下：`{Assembly}::{Namespace}.{ClassName}::{MethodName}`。有关更多信息，请参阅处理程序签名。在此处设置 `image-command` 后，稍后会在 Visual Studio 的“发布”向导中预填充此值。

以下是 aws-lambda-tools-defaults .NET 8（容器镜像）蓝图创建的.json 示例。

```
{
```

```
  "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

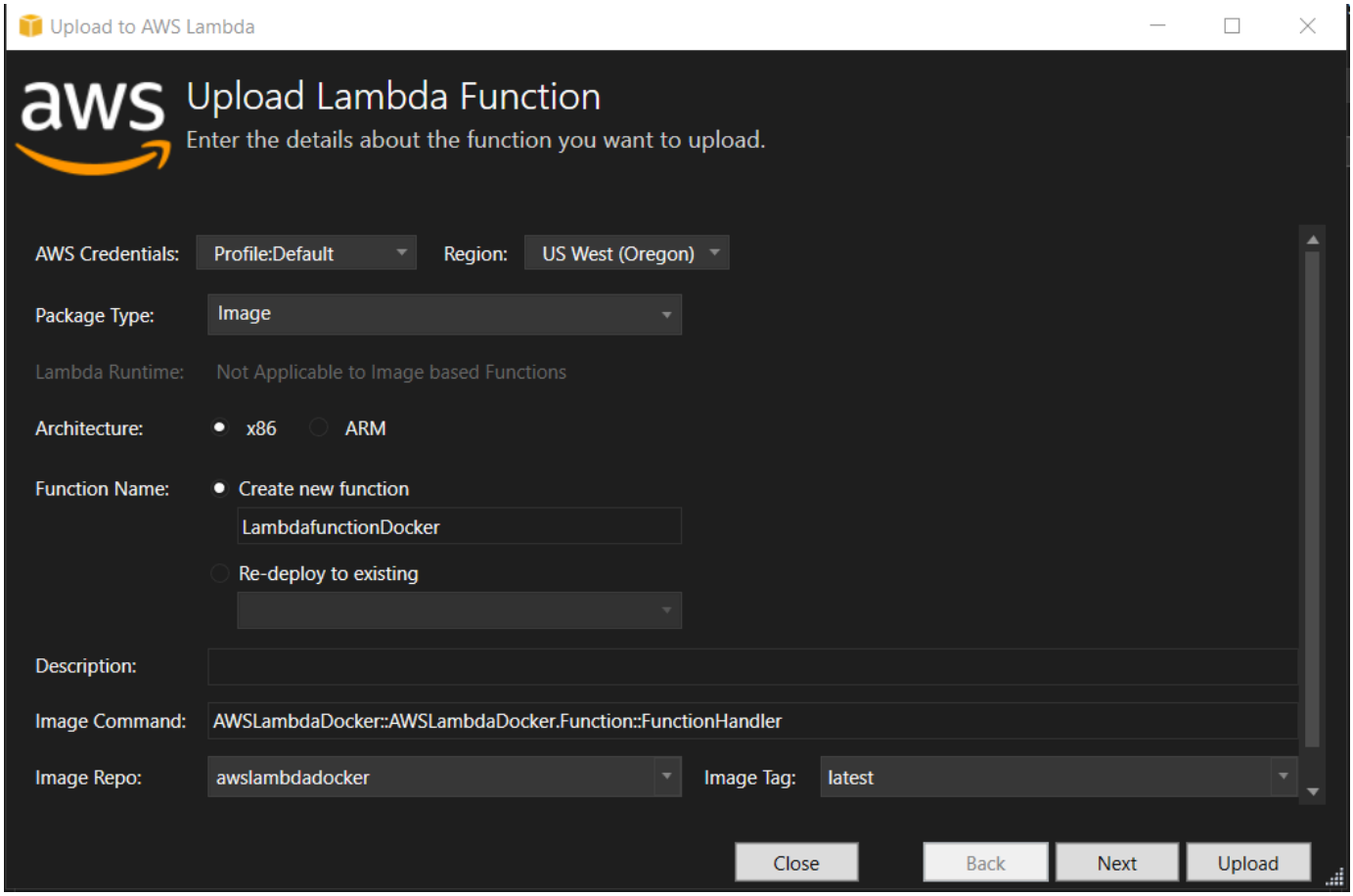
该Function.cs文件定义了要作为 Lambda 函数公开的 c# 函数。FunctionHandler 是在运行 Lambda 函数时运行的 Lambda 功能。在这个项目中，FunctionHandler调用ToUpper()输入的文本。

发布到 Lambda

构建过程中生成的 Docker 映像上传到 Amazon Elastic Container Registry（Amazon ECR）。Amazon ECR 一个完全托管式 Docker 容器映像库，您可以使用该映像库存储、管理和部署 Docker 容器映像。Amazon ECR 托管映像，然后 Lambda 会引用该映像，以便在调用时提供编程的 Lambda 功能。

将函数发布到 Lambda

1. 在解决方案资源管理器中，打开（右键单击）项目的快捷菜单，然后选择 “发布” AWS Lambda 以打开 “上传 Lambda 函数” 窗口。
2. 在上传 Lambda 函数页面上，执行以下操作：



a. 对于包类型，**`Image`** 已被自动选为包类型，因为发布向导在项目中检测到了 `Dockerfile`。

b. 对于函数名称，为 Lambda 实例输入显示名称。此名称是在 Visual Studio 的 AWS 各区服务浏览器中和 AWS Management Console中显示的引用名称。

c. 对于描述，输入要在 AWS Management Console中与您的实例一起显示的文本。

d. 对于映像命令，输入希望 Lambda 函数运行的方法的完全限定路径：**`AWSLambdaDocker::AWSLambdaDocker.Function::FunctionHandler`**

> **Note**
>
> 此处输入的任何方法名称都将覆盖 Dockerfile 中的任何 CMD 指令。只有在 `Dockerfile` 包含用于指示如何启动 Lambda 函数的 `CMD` 时，输入映像命令才是可选的。

e. 对于映像存储库，输入新的或现有 Amazon Elastic Container Registry 的名称。构建过程创建的 Docker 映像将上传到此映像库。要发布的 Lambda 定义将引用该 Amazon ECR 映像。

f. 对于映像标签，输入一个 Docker 标签以与存储库中的映像相关联。

g. 选择下一步。

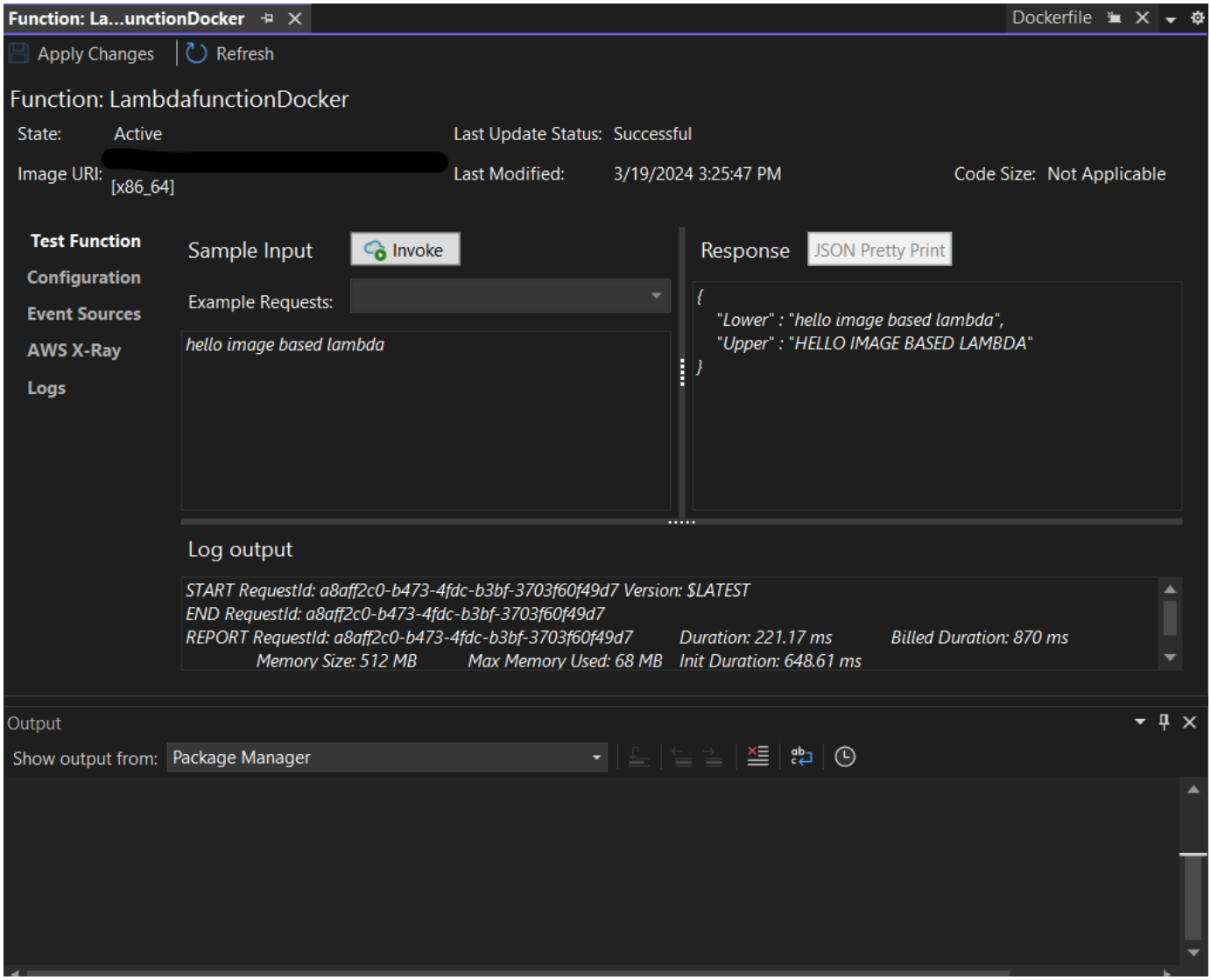
3. 在高级函数详细信息页面的角色名称中，选择与您的账户关联的角色。该角色用于为函数中的代码所发起的任何 Amazon Web Services 调用提供临时凭证。如果您没有角色，请选择 “基于 AWS 托管策略新建角色”，然后选择AWSLambdaBasicExecutionRole。

> **Note**
>
> 您的账户必须拥有运行 IAM ListPolicies 操作的权限，否则角色名称列表将为空。

4. 选择 Up load 开始上传和发布过程。

> **Note**
>
> 上传函数时，将显示正在上传函数页面。然后，发布过程根据配置参数构建映像，必要时创建 Amazon ECR 存储库，将映像上传到存储库，然后创建引用包含该映像的存储库的 Lambda。
>
> 上传函数后，函数页面将打开并显示新 Lambda 函数的配置。

5. 要手动调用 Lambda 函数，请在测试函数选项卡上，在请求的自由文本输入字段输入 `hello image based lambda`，然后选择调用。您的文本将转换为大写并显示在响应中。



6. 要查看存储库，请在 AWS 各区服务浏览器中的 Amazon Elastic Container Service 下选择存储库。

   您可以随时重新打开函数：视图，方法是双击位于 AWS 各区服务浏览器中 AWS Lambda 节点下的已部署实例。

   > **Note**
   >
   > 如果你的 AWS 资源管理器窗口未打开，你可以通过 “视图”-> “AWS 资源管理器” 将其停靠

7. 请注意配置选项卡上其他特定于映像的配置选项。此选项卡提供了一种覆盖可能已在 Dockerfile 中指定的 `ENTRYPOINT`、`CMD`、和 `WORKDIR` 的方法。描述是您在上传/发布期间输入的描述（如果有）。

清理

如果您不打算继续使用此示例进行开发，请记得删除已部署的函数和 ECR 映像，这样就不会为账户中未使用的资源付费。

- 右键单击位于 AWS 各区服务浏览器中 AWS Lambda 节点下的已部署实例，即可删除函数。
- 可以在 AWS 各区服务浏览器中的 Amazon Elastic Container Service -> 存储库下删除存储库。

后续步骤

有关创建和测试 Lambda 映像的信息，请参阅使用 Lambda 容器映像。

有关容器映像部署、权限和覆盖配置设置的信息，请参阅配置函数。

教程：使用以下方法构建和测试无服务器应用程序 AWS Lambda

您可以使用模板构建无服务器 Lambda 应用程序。 AWS Toolkit for Visual Studio Lambda 项目模板包括一个用于AWS 无服务器应用程序的模板，即AWS 无服务器应用程序模型 (S AM) 的 AWS Toolkit for Visual Studio 实现。AWS 使用此项目类型，您可以开发一组 AWS Lambda 函数，并使用任何必要的 AWS 资源将它们作为整个应用程序进行部署， AWS CloudFormation 用于协调部署。

有关设置的先决条件和信息 AWS Toolkit for Visual Studio，请参阅使用 Visual Stu AWS dio AWS 工具包中的 Lambda 模板。

主题

创建一个新的 AWS 无服务器应用程序项目

AWS 无服务器应用程序项目使用无服务器模板创建 Lambda 函数。 AWS CloudFormation AWS CloudFormation 模板使您能够定义其他资源，例如数据库、添加 IAM 角色和一次部署多个函数。这与 AWS Lambda 项目不同，后者侧重于开发和部署单个 Lambda 函数。

以下过程介绍如何创建新的 AWS 无服务器应用程序项目。

1. 从 Visual Studio 中展开 “文件” 菜单，展开 “新建”，然后选择 “项目”。

2. 在 “新建项目” 对话框中，确保将 “语言”、“平台” 和 “项目类型” 下拉框设置为 “全部...”，然后在 “搜索” 字段**aws lambda**中输入。
3. 选择带测试的AWS 无服务器应用程序（.NET Core – C# ）模板。

> **Note**
>
> 带测试的AWS 无服务器应用程序（.NET Core-C#）模板可能不会填充在结果的顶部。

4. 单击 “下一步” 打开 “配置您的新项目” 对话框。
5. 在 “配置您的新项目” 对话框中，输入**ServerlessPowertools**名称，然后根据自己的喜好填写其余字段。选择 “创建” 按钮进入选择蓝图对话框。
6. 从 “选择蓝图” 对话框中为 AWS Lambda蓝图选择 Powertools，然后选择 “完成” 创建 Visual Studio 项目。

查看无服务器应用程序文件

以下各节详细介绍了为您的项目创建的三个无服务器应用程序文件：

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

1. 无服务器.template

`serverless.template`文件是用于声明您的无服务器函数和其他 AWS 资源的 AWS CloudFormation 模板。此项目中包含的文件包含单个 Lambda 函数的声明，该函数将作为一项HTTP *`Get`*操作通过 Amazon API Gateway 公开。您可以编辑此模板以自定义现有函数或添加应用程序所需的更多函数和其他资源。

以下是 `serverless.template` 文件的示例：

```
{
  "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/"
      }
    }
  }
```

```
}
```

请注意，许多...AWS:: Serverless::Function...声明字段与 Lambda 项目部署的字段相似。Powertools 日志、指标和跟踪是通过以下环境变量配置的：

- POWERTOOLS 服务名称= ServerlessGreeting
- PowerTools_log_level=in
- POWERTOOLS_LOGGER_C PascalCase
- PowerTools_tracer_capture_response=T
- PowerTools_tracer_capture_error=True
- POWERTOOLS_METRICS_NAMESPAC ServerlessGreeting

有关环境变量的定义和其他详细信息，请参阅 Powertools 供 AWS Lambda 参考网站。

2. Functions.cs

Functions.cs是一个包含一个 C# 方法的类文件，该方法映射到模板文件中声明的单个函数。Lambda 函数响应 API Gatewa HTTP Get y 中的方法。以下是该Functions.cs文件的示例：

```
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提供了 Visual Studio 内部 AWS 部署向导的默认值以及添加到.NET Core CLI 中的 AWS Lambda 命令。以下是此项目中包含的aws-lambda-tools-defaults.json文件的示例：

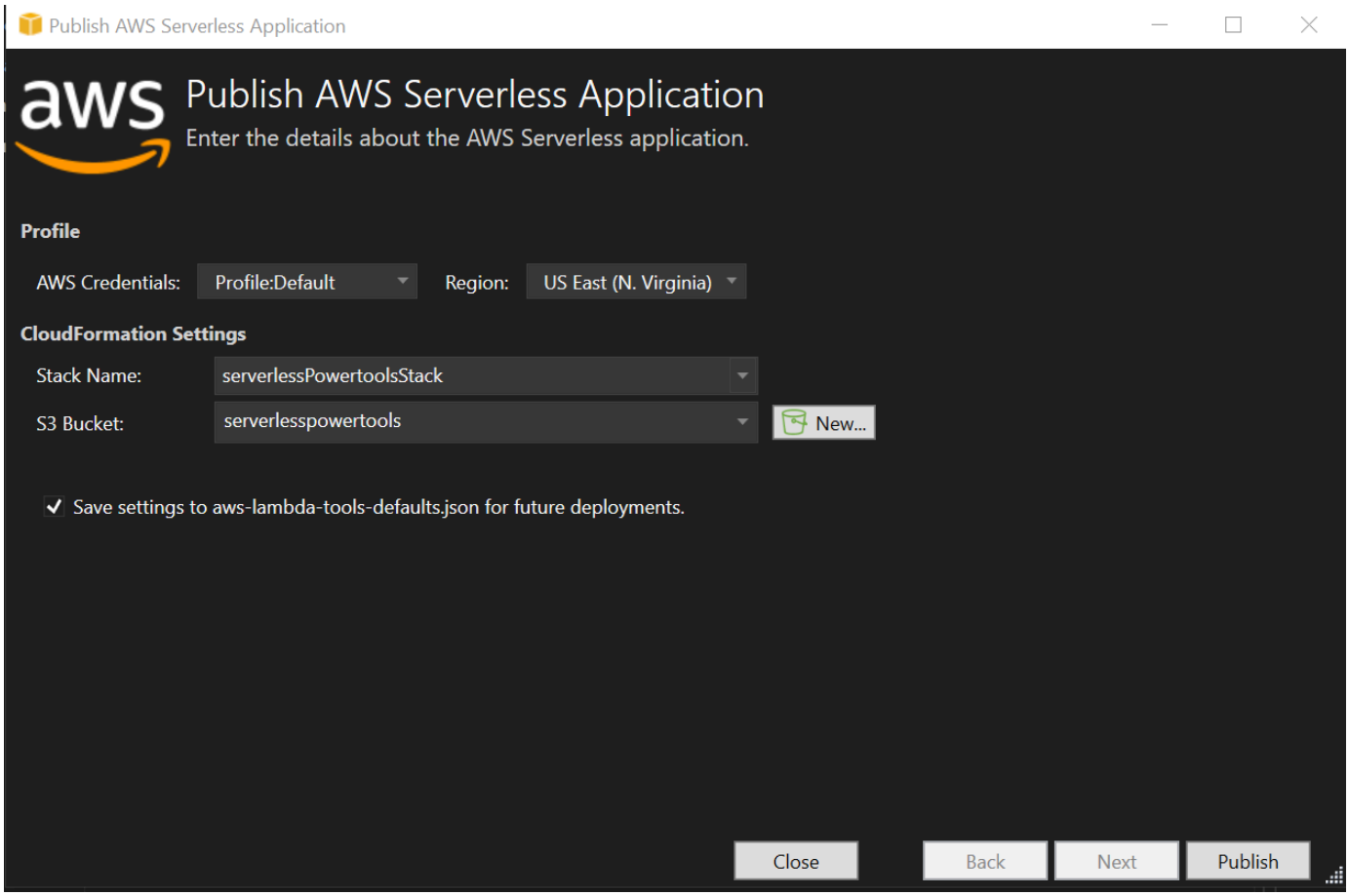
```
{
  "profile": "Default",
  "region": "us-east-1",
  "configuration": "Release",
  "s3-prefix": "ServerlessPowertools/",
  "template": "serverless.template",
  "template-parameters": ""
}
```

部署无服务器应用程序

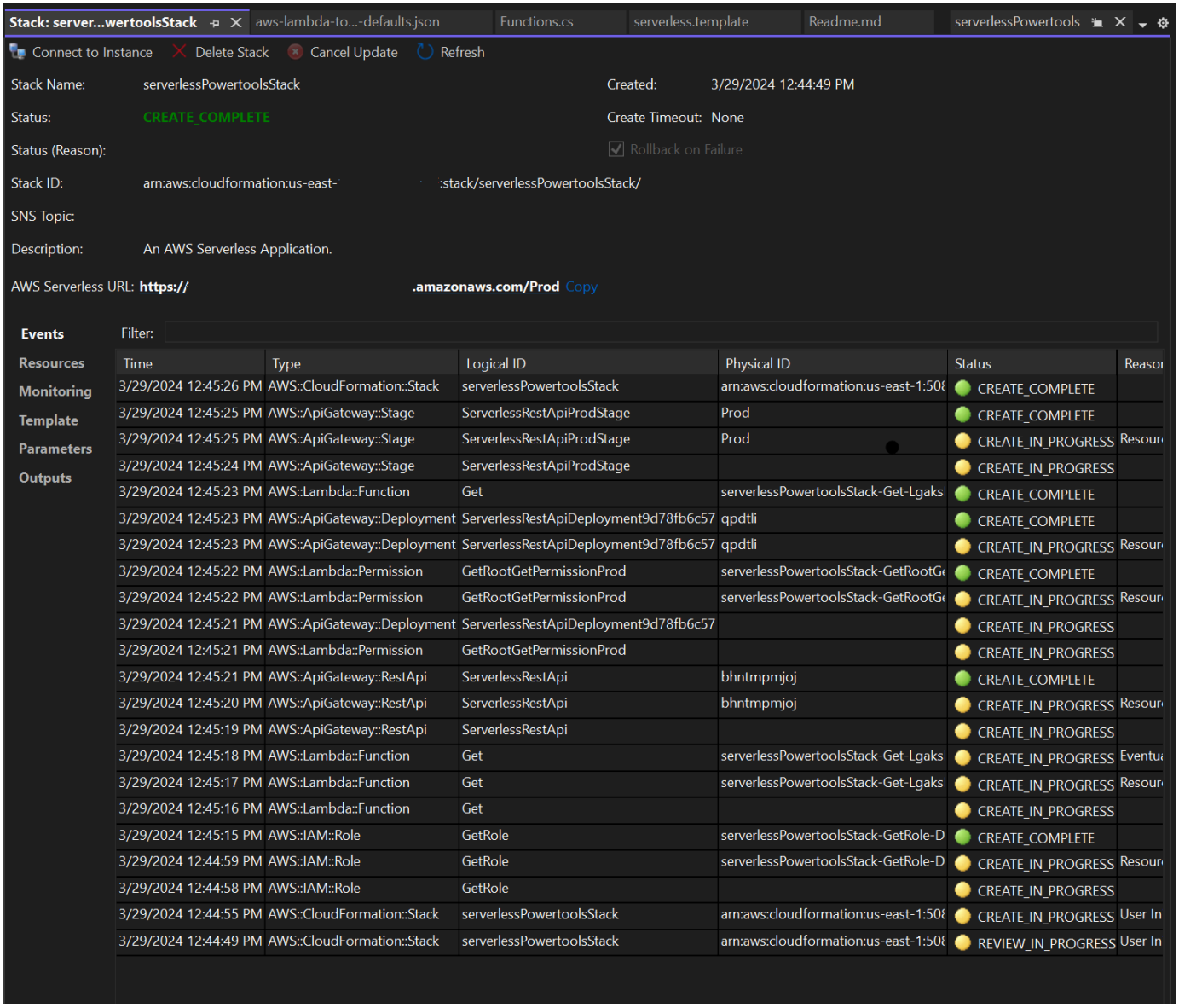
要部署您的无服务器应用程序，请完成以下步骤

1. 在解决方案资源管理器中，打开项目的快捷菜单（右键单击），然后选择 “发布到 AWS Lambda” 以打开 “发布 AWS 无服务器应用程序” 对话框。
2. 在 “发布 AWS 无服务器应用程序” 对话框中，在 AWS CloudFormation 堆栈名称字段中输入堆栈容器的名称。
3. 在 S3 存储桶字段中，选择您的应用程序包将上传到的 Amazon S3 存储桶，或者选择新的... 按钮，然后输入新 Amazon S3 存储桶的名称。然后选择 “发布以发布” 来部署您的应用程序。

> **Note**
>
> 您的 AWS CloudFormation 堆栈和 Amazon S3 存储桶必须位于同一 AWS 区域。项目的其余设置在`serverless.template`文件中定义。



4. 在发布过程中，“堆栈视图” 窗口打开，部署完成后，“状态” 字段将显示：CREATE_COMPLETE。



部署无服务器应用程序

堆栈创建完成后，您可以使用AWS 无服务器 URL 查看您的应用程序。如果您在完成本教程时未添加任何其他函数或参数，则访问您的 AWS 无服务器 URL 会在您的网络浏览器中显示以下短语：`Hello Powertools for AWS Lambda (.NET)`。

教程：创建 Amazon Rekognition Lambda 应用程序

本教程向您说明如何创建 Lambda 应用程序，该应用程序使用 Amazon Rekognition 标记包含检测到的标签的 Amazon S3 对象。

有关设置的先决条件和信息 AWS Toolkit for Visual Studio，请参阅使用 Visual Stu AWS dio AWS 工具包中的 Lambda 模板。

创建 Visual Studio .NET Core Lambda Image Rekognition 项目

以下过程介绍如何通过创建 Amazon Rekognition Lambda 应用程序。 AWS Toolkit for Visual Studio

> **Note**
>
> 创建后，您的应用程序将有一个包含两个项目的解决方案：一个包含要部署到 Lambda 的 Lambda 函数代码的源项目，以及一个使用 xUnit 在本地测试函数的测试项目。
> 有时 Visual Studio 无法找到你的项目的所有 NuGet 参考资料。这是因为蓝图需要必须从中 NuGet检索的依赖关系。创建新项目时，Visual Studio 仅从中提取本地引用，而不会从 NuGet 中提取远程引用。要修复 NuGet 错误，请右键单击您的参考文献，然后选择 “还原包”。

1. 从 Visual Studio 中展开 “文件” 菜单，展开 “新建”，然后选择 “项目”。
2. 在 “新建项目” 对话框中，确保将 “语言”、“平台” 和 “项目类型” 下拉框设置为 “全部...”，然后在 “搜索” 字段**`aws lambda`**中输入。
3. 选择 “AWS Lambda 带测试”（.NET Core-C#）模板。
4. 单击 “下一步” 打开 “配置您的新项目” 对话框。
5. 在 “配置您的新项目” 对话框中，在 “名称 ImageRekognition” 中输入 “”，然后根据自己的喜好填写其余字段。选择创建按钮进入选择蓝图对话框。
6. 从 “选择蓝图” 对话框中，选择 “检测图像标签” 蓝图，然后选择 “完成” 创建 Visual Studio 项目。

> **Note**
>
> 此蓝图提供了用于侦听 Amazon S3 事件的代码，并使用 Amazon Rekognition 检测标签并将其作为标记添加到 S3 对象。

查看项目文件

以下各节将研究这些项目文件：

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

1. Function.cs

在Function.cs文件中，第一段代码是位于文件顶部的汇编属性。默认情况下，Lambda 仅接受输入参数和类型的返回类型。System.IO.Stream必须注册序列化器才能使用类型化类作为输入参数和返回类型。汇编属性注册了 Lambda JSON 序列化器，该序列化器用于将流Newtonsoft.Json转换为类型化类。您可以在程序集或方法级别设置串行器。

以下是程序集属性的示例：

```
// 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
```

该类有两个构造函数。第一个是 Lambda 调用您的函数时使用的默认构造函数。此构造函数创建 Amazon S3 和 Amazon Rekognition 服务客户端。构造函数还会从您在部署函数时分配给该函数的 IAM 角色中检索这些客户端的 AWS 证书。客户端的 AWS 区域设置为运行您的 Lambda 函数的区域。在此蓝图中，只有在 Amazon Rekognition 服务对标签的最低可信度时，您才需要向 Amazon S3 对象添加标签。此构造函数将检查环境变量 MinConfidence 以确定可接受的置信度级别。您可以在部署 Lambda 函数时设置该环境变量。

以下是中第一个类构造函数的示例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}");
    }
}
```

以下示例演示了如何使用第二个构造函数进行测试。测试项目配置自己的 S3 和 Rekognition 客户端，并将它们传入：

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

以下是Function.cs文件中该FunctionHandler方法的示例。

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

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

```
                    Name = record.S3.Object.Key
                }
            }
        });

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

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

`FunctionHandler` 是 Lambda 构建实例后调用的方法。请注意，输入参数的类型是 `S3Event`，而不是 `Stream`。您可以执行此操作，因为您已注册 Lambda JSON 串行器。`S3Event` 包含在 Amazon S3 中触发的事件的所有信息。该函数将遍历组成事件的所有 S3 对象并让 Rekognition 检测标签。在检测标签后，标签将作为标记添加到 S3 对象。

> **Note**
>
> 该代码包含对的调用`Console.WriteLine()`。当该函数在 Lambda 中运行时，所有调用都`Console.WriteLine()`将重定向到 Ama CloudWatch zon 日志。

2。 aws-lambda-tools-defaults.json

该`aws-lambda-tools-defaults.json`文件包含蓝图设置的默认值，用于预填充部署向导中的某些字段。它还有助于设置命令行选项以与.NET Core CLI 集成。

要访问.NET Core CLI 集成，请导航到该函数的项目目录并键入**`dotnet lambda help`**。

> **Note**
>
> 函数处理程序指示 Lambda 要调用什么方法来响应被调用的函数。此字段的格式为:`<assembly-name>::<full-type-name>::<method-name>`. 命名空间必须包含在类型名称中。

部署函数

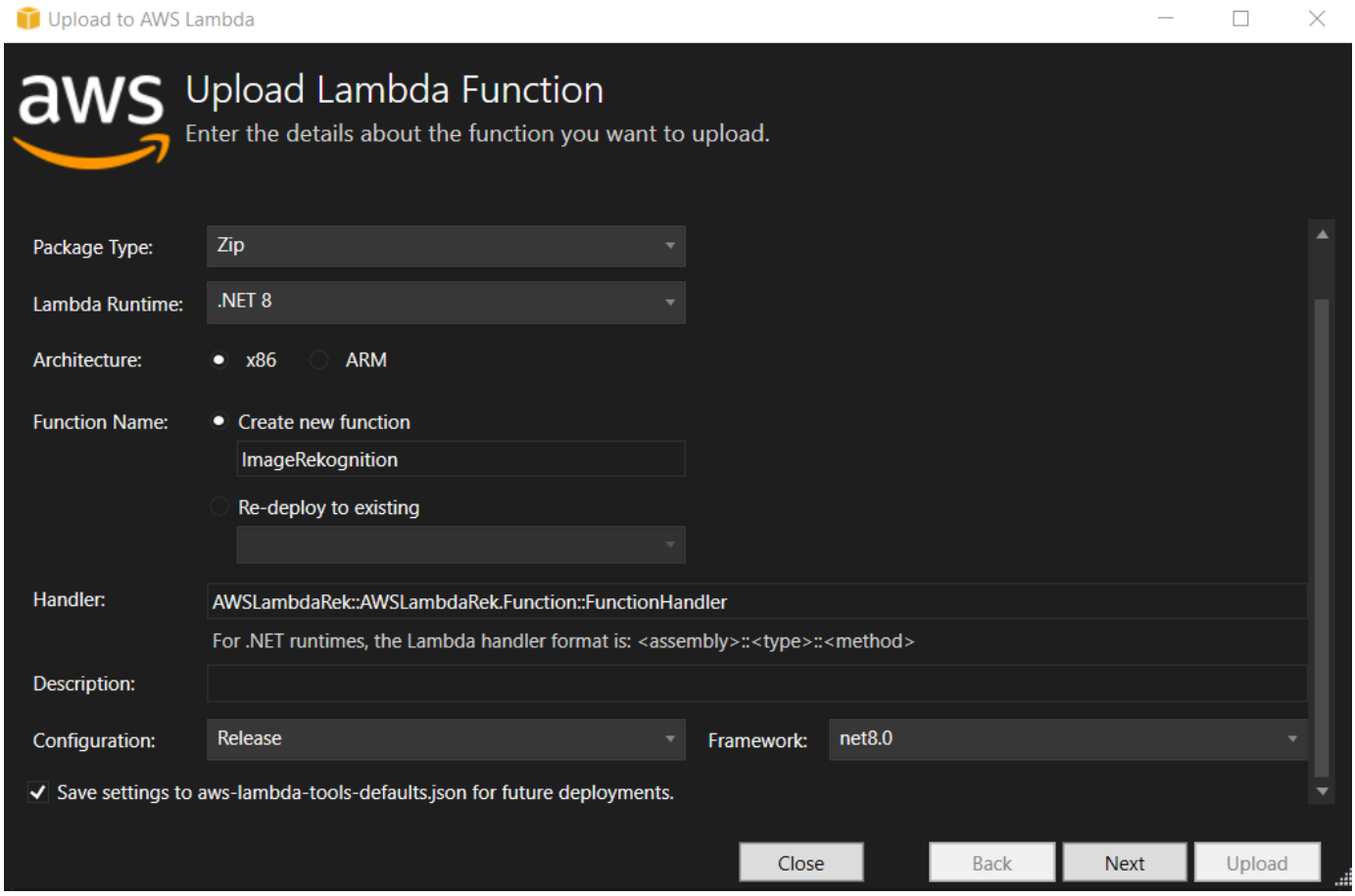
以下过程描述了如何部署 Lambda 函数。

1. 在解决方案资源管理器中，右键单击 Lambda 项目，然后选择 “发布到 Lambda” 以打开 “ AWS 上传到” 窗口。 AWS Lambda

   > **Note**
   >
   > 预设值是从`aws-lambda-tools-defaults.json`文件中检索的。

2. 在 “上传到 AWS Lambda” 窗口中，在 “函数名称” 字段中输入名称，然后选择 “下一步” 按钮前进到 “高级函数详细信息” 窗口。

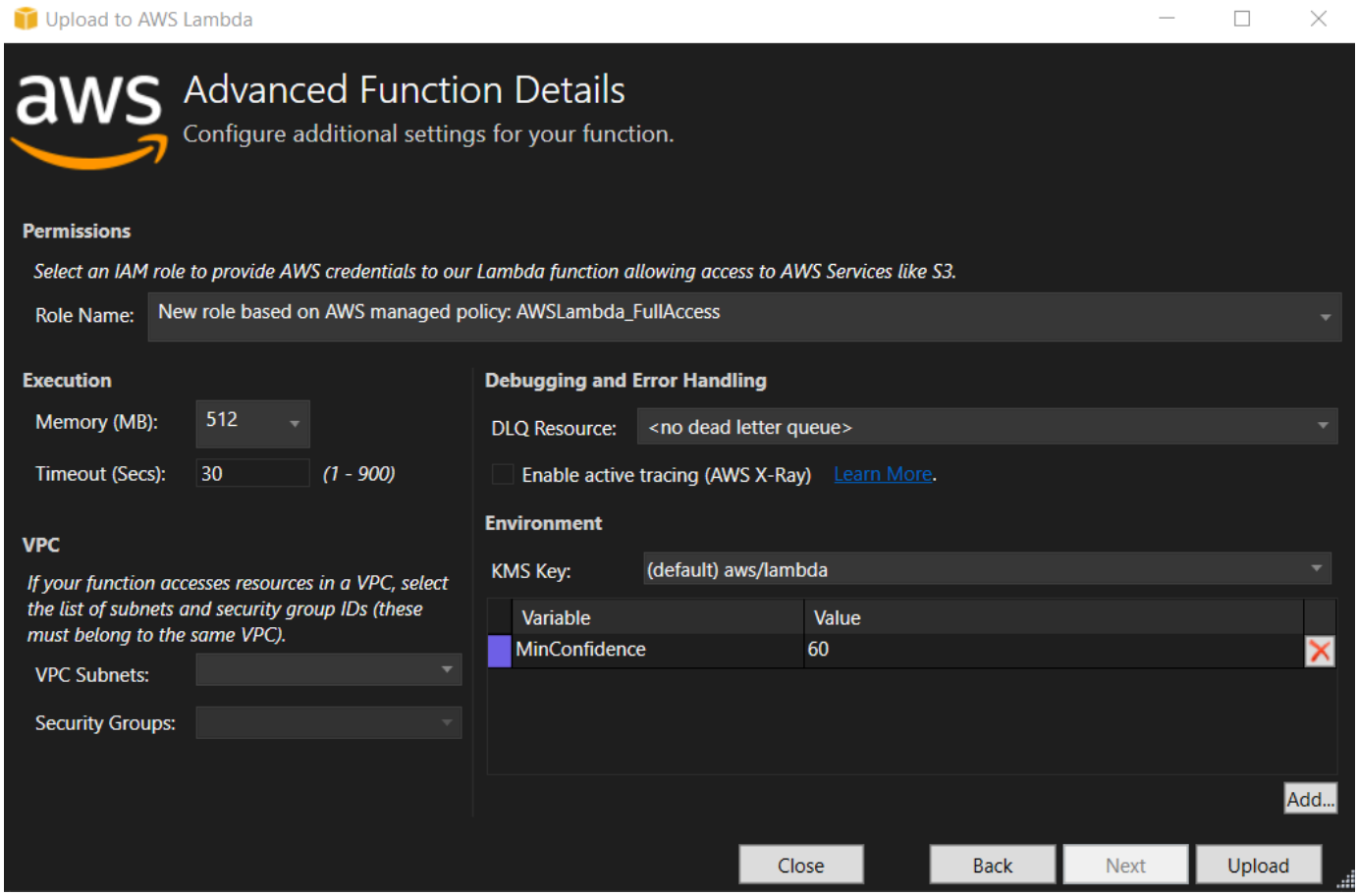
   > **Note**
   >
   > 此示例使用函数名称**`ImageRekognition`**。



3. 在“高级功能详情”窗口中，选择一个 IAM 角色，该角色允许您的代码访问您的 Amazon S3 和 Amazon Rekognition 资源。

> **Note**
>
> 如果您正在关注此示例，请选择`AWSLambda_FullAccess`角色。

4. 将环境变量设置`MinConfidence`为 60，然后选择 U p load 启动部署过程。当“函数”视图显示在AWS 资源管理器中时，发布过程即告完成。



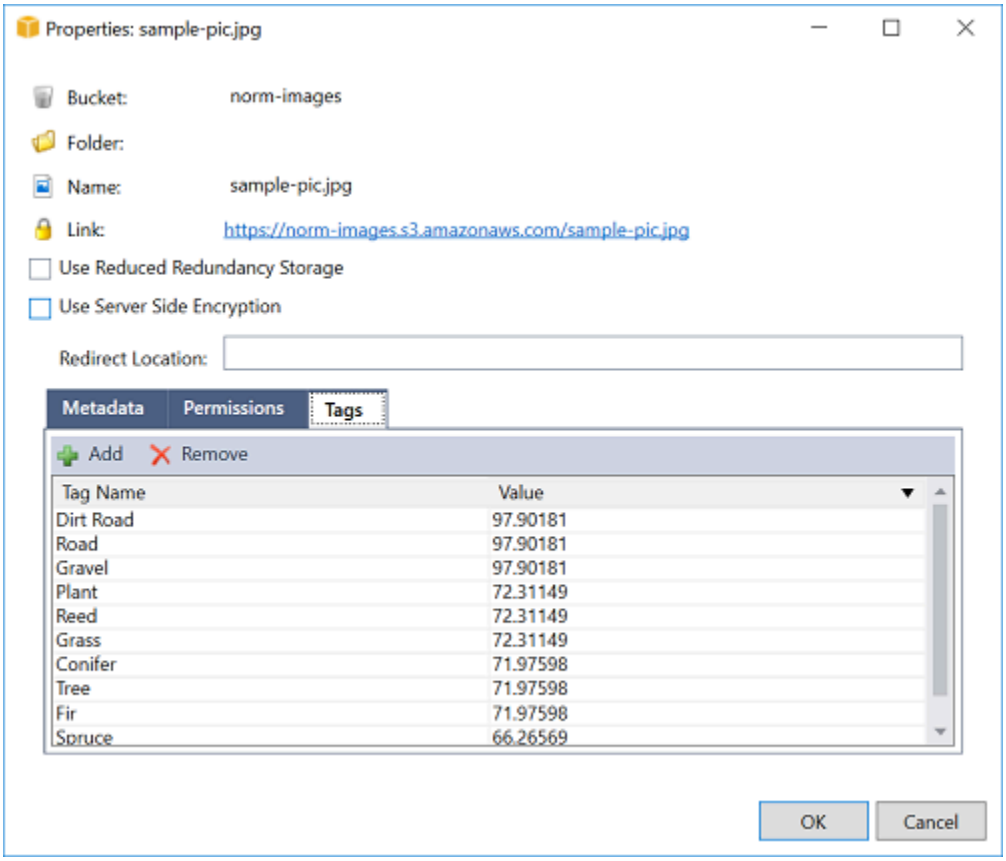
5. 成功部署后，通过导航到 “事件源” 选项卡，将 Amazon S3 配置为将其事件发送到您的新函数。
6. 在事件源选项卡中，选择添加按钮，然后选择要连接您的 Lambda 函数的 Amazon S3 存储桶。

> **Note**
>
> 存储桶必须与您的 Lambda 函数位于同一 AWS 区域。

测试函数。

现在已部署该函数，并将 S3 桶配置为函数的事件源，请从 AWS 各区服务浏览器中为您选定的桶打开 S3 桶浏览器。然后上传一些图像。

上传完成后，您可以通过在函数视图中查看日志来确认您的函数已运行。或者，右键单击存储桶浏览器中的图像，然后选择 Properties (属性)。在 Tags (标签) 选项卡上，您可以查看应用到您的对象的标签。



教程：使用 Amazon 日志框架和 AWS Lambda 创建应用程序日志

您可以使用 Amazon CloudWatch Logs 来监控、存储和访问应用程序的日志。要将日志数据导入 CloudWatch 日志，请使用 S AWS DK 或安装 Log CloudWatch s 代理来监控某些日志文件夹。CloudWatch 日志与几个流行的.NET 日志框架集成，从而简化了工作流程。

要开始使用 CloudWatch 日志和.NET 日志框架，请将相应的 NuGet 包和 CloudWatch 日志输出源添加到您的应用程序中，然后像往常一样使用您的日志库。这使您的应用程序能够使用.NET 框架记录消息，将其发送到 Lo CloudWatch gs，在日志控制台中显示应用程序的 CloudWatch 日志消息。您还可以根据应用程序的日志消息，在 CloudWatch 日志控制台中设置指标和警报。

支持的.NET 日志框架包括：

- nLog：要查看，请参阅 n uget.or g nLog 软件包。
- Log4net：要查看，请参阅 nuget.org Lo g4ne t 软件包。
- ASP.NET Core 日志框架：要查看，请参阅 nug et.org ASP.NET Core 日志框架包。

以下是一个文件示例，该NLog.config文件通过将AWS.Logger.NLog NuGet 软件包和 AWS 目标添加到中来启用日志和控制台作为日志消息的输出NLog.config。 CloudWatch

```
<?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>
```

日志插件都建立在之上，通过类似于 SDK 的流程对您的 AWS 凭据进行身份验证。 AWS SDK for .NET 以下示例详细说明了日志插件凭据访问 CloudWatch 日志所需的权限：

> **Note**
>
> . AWS NET 日志插件是一个开源项目。有关更多信息、示例和说明，请参阅 Lo AWS gging .NET GitHub 存储库中的示例和说明主题。

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

```
    ]
}
```

部署到 AWS

Toolkit for Visual Studio 支持将应用程序部署到AWS Elastic Beanstalk容器或AWS CloudFormation堆栈。

> **Note**
>
> 如果您使用的是 Visual Studio Express Edition：
>
> - 您可以使用 Docker CLI 将应用程序部署到 Amazon ECS 容器。
> - 您可以使用AWS管理控制台将应用程序部署到 Elastic Beanstalk 容器。
>
> 对于 Elastic Beanstalk 部署，必须先创建 Web 部署包。有关更多信息，请参阅如何：在 Visual Studio 中创建 Web 部署程序包。要部署 Amazon ECS，您必须拥有 Docker 镜像。有关更多信息，请参阅适用于 Docker 的 Visual Studio 工具。

主题

使用 PublishAWS在 Visual Stu

Publish to (发布到 CloudWatch)AWS是一种交互式部署体验，可帮助您将 .NET 应用程序发布到AWS部署目标，支持以 .NET Core 3.1 及更高版本为目标的应用程序。使用 PublishAWS通过直接在 IDE 中提供以下部署功能，将工作流程保留在 Visual Studio 中：

- 只需单击一下即可部署应用程序。
- 基于您的应用程序的部署建议。
- 自动创建 Dockerfile，这是部署目标环境（部署目标）的相关和必需的。
- 根据部署目标的要求，优化了用于构建和打包应用程序的设置。

> **Note**
>
> 有关发布 .NET Framework 应用程序的其他信息，请参阅指南在 Elastic Beanstalk 上创建和部署 .NET 应用程序
> 您也可以访问发布到 CloudWatchAWS来自 .NET CLI。有关更多信息，请参阅 。在上部署 .NET 应用程序AWS指南。

主题

先决条件

成功地将 .NET 应用程序发布到AWS服务，请在本地设备上安装以下内容：

- .NET Core 3.1+ (包括 .NET5 和 .NET6): 有关这些产品的更多信息和下载信息，请访问微软下载网站.
- Node.js 14.x 或更高版本：运行 Node.js 是必需的AWS Cloud Development Kit (AWS CDK). 要下载或获取有关 Node.js 的更多信息，请访问Node.js 下载站点.

  > **Note**
  >
  > Publish to (发布到 CloudWatch)AWS利用AWS CDK将您的应用程序及其所有部署基础架构作为一个项目进行部署。有关 的更多信息AWS CDK请参阅Cloud Development Kit指南。

- （可选）在部署到基于容器的服务（如 Amazon ECS）时使用 Docker。有关更多信息和下载 Docker，请参阅Docker 下载site（站点）。

支持的应用程序类型

在发布到新目标或现有目标之前，首先在 Visual Studio 中创建或打开以下项目类型之一：

- ASP.NET 核心应用程序
- .NET 控制台应用程序

- Blazor WebAssembly 应用程序

向发布应用程序AWS目标

发布到新目标时，发布到AWS将通过提出建议和使用常用设置引导您完成该过程。如果您需要发布到之前设置的目标，则您的首选项将被存储并可以进行调整，或者立即可用于一键式部署。

发布到新目标

下面介绍了如何配置 Publish (发布目标)AWS部署首选项，当您发布到新目标时。

1. 从AWS探险者中，展开凭证下拉菜单中，然后选择AWS与该区域相对应的配置文件和AWS您的部署所需的服务。
2. 展开区域下拉菜单中，然后选择AWS包含以下内容的区域AWS您的部署所必需的服务。
3. 从 Visual Studi解决方案管理器窗格中，打开 (右键单击) 项目名称的上下文菜单，然后选择Publish to (发布到 CloudWatch)AWS. 这将打开Publish to (发布到 CloudWatch)AWS.
4. 从Publish to (发布到 CloudWatch)AWS，选择发布到新目标配置新的部署。

> **Note**
>
> 要修改默认部署凭据，请选择或单击编辑链接位于凭证部分，在Publish to (发布到 CloudWatch)AWS.
> 要绕过目标配置过程，请选择发布到现有目标，然后从以前的部署目标列表中选择首选配置。

5. 从发布目标窗格中，选择AWS服务来管理您的应用程序部署。
6. 根据需要做好配置后，选择发布启动部署过程。

> **Note**
>
> 启动部署后，Publish to (发布到 CloudWatch)AWS显示以下状态更新：
>
> - 在部署过程中，Publish to (发布到 CloudWatch)AWS显示有关部署进度的信息。
> - 在部署过程之后，Publish to (发布到 CloudWatch)AWS指示部署是成功了还是失败了。
> - 成功部署后，资源面板提供了有关所创建资源的其他信息。此信息因应用程序类型和部署配置而异，具体取决于应用程序类型和部署配置。

发布到现有目标

下面介绍如何将.NET 应用程序重新发布到现有的AWS目标。

1. 从AWS探险者中，展开凭证下拉菜单中，然后选择AWS与该区域相对应的配置文件和AWS您的部署所需的服务。
2. 展开区域下拉菜单中，然后选择AWS包含以下内容的区域AWS您的部署所必需的服务。
3. 从 Visual Studi解决方案管理器窗格中，右键单击项目名称，然后选择Publish to (发布到 CloudWatch)AWS打开Publish to (发布到 CloudWatch)AWS.
4. 从Publish to (发布到 CloudWatch)AWS，选择发布到现有目标从现有目标列表中选择部署环境，然后从现有目标列表中选择部署环境。

   > **Note**
   >
   > 如果您最近发布了任何应用程序到AWSCloud，则这些应用程序将显示在 “发布到”AWS.

5. 选择要在其中部署应用程序的发布目标，然后单击发布启动部署过程。

部署AWS Lambda使用 .NET 核心 CLI 项目

AWS Toolkit for Visual Studio 包含适用于 Visual Studio 的 AWS Lambda .NET 内核项目模板。您可以使用 .NET 内核命令行界面 (CLI) 部署 Visual Studio 中构建的 Lambda 函数。

主题

先决条件

在使用 .NET 核心 CLI 部署 Lambda 函数之前，您必须满足以下先决条件：

- 请确保安装 Visual Studio 2015 Update 3。
- 安装NET 内核 for Windows.

- 设置 .NET 核心 CLI 使用 Lambda。有关更多信息，请参阅 。.NET 内核 CLI中的AWS Lambda开发人员指南.
- 安装 Toolkit for Visual Studio。有关更多信息，请参阅 正在安装 AWS Toolkit for Visual Studio。

相关主题

在使用 .NET 核心 CLI 部署 Lambda 函数时，以下相关主题可能会有所帮助：

- 有关 Lambda 函数的更多信息，请参阅是什么AWSLambda？中的AWS Lambda开发人员指南.
- 有关在 Visual Studio 中创建 Lambda 函数的信息，请参阅AWS Lambda.
- 有关 Microsoft .NET 内核的更多信息，请参阅.NET 内核在微软的在线文档中。

列出 .NET 核心 CLI 使用的 Lambda 命令

要列出可通过 .NET 核心 CLI 使用的 Lambda 命令，请执行以下操作。

1. 打开命令提示符窗口，然后导航到包含 Visual Studio .NET 核心 Lambda 项目的文件夹。
2. 输入 `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>
```

从 .NET 核心 CLI 发布 .NET 核心 Lambda 项目

以下说明假定您已在 Visual Studio 中创建 AWS Lambda .NET 内核函数。

1. 打开命令提示符窗口，然后导航到包含您的 Visual Studio .NET 核心 Lambda 项目的文件夹。
2. 输入 `dotnet lambda deploy-function`。
3. 当系统提示时，输入要部署的函数的名称。它可以是新名称或现有函数的名称。
4. 当系统提示时，输入AWS区域 (Lambda 函数将部署到的区域)。
5. 当系统提示时，选择或创建 Lambda 将在执行函数时代入的 IAM 角色。

成功完成后，将显示消息 New Lambda function created (新 Lambda 函数已创建)。

```
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
```

如果您部署现有函数，则部署函数仅要求AWS区域。

```
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
```

在部署 Lambda 函数后，便可使用该函数。有关更多信息，请参阅 。如何使用 的示例AWSLambda.

Lambda 会自动替您监控 Lambda 函数，并通过亚马逊报告各项指标 CloudWatch. 要监控 Lambda 函数并排除故障，请参阅诊断和监控AWSLambda 在亚马逊上使用 CloudWatch.

部署到 Elastic Beanstalk

AWS Elastic Beanstalk是简化配置过程的服务AWS您的应用程序的资源。Elastic Beanstalk 提供所有AWS部署应用程序所需的基础设施。此基础设施包括：

- 一些 Amazon EC2 实例，可托管您的应用程序的可执行文件和内容。
- 一个 Auto Scaling 组，可保留适当数量的 Amazon EC2 实例来支持您的应用程序。
- 一个Elastic 负载均衡负载均衡器，可将传入流量路由至具有最大带宽的 Amazon EC2 实例。

Toolkit for Visual Studio 提供了一个通过 Elastic Beanstalk 简化应用程序发布的向导。此向导将在以下部分中介绍。

有关 Elastic Beanstalk 的更多信息，请转到Elastic Beanstalk 文档.

主题

- 将传统的 ASP.NET 应用程序部署到 Elastic Beanstalk
- 将 ASP.NET Core 应用程序部署到 Elastic Beanstalk（传统）
- 如何指定AWS应用程序的安全凭证
- 如何将您的应用程序重新发布到 Elastic Beanstalk 环境（传统）
- 自定义 Elastic Beanstalk 应用程序部署
- 自定义 ASP.NET 内核 Elastic Beanstalk 部署
- 对 .NET 和 Elastic Beanstalk 的多应用程序 Support

将传统的 ASP.NET 应用程序部署到 Elastic Beanstalk

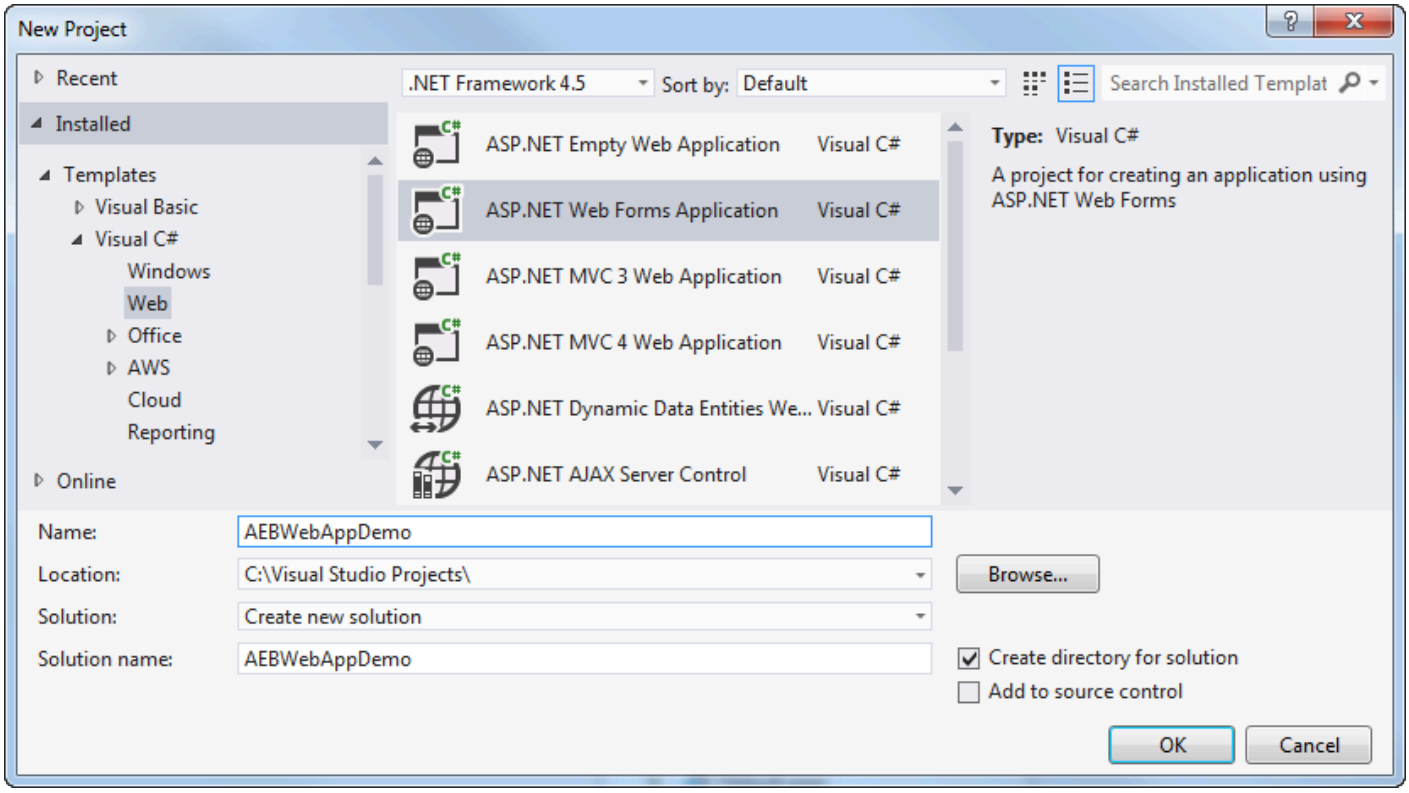
本节介绍如何使用作为 Visual Studio 工具包一部分提供的 “发布到 Elastic Beanstalk” 向导通过 Elastic Beanstalk 部署应用程序。要进行练习，您可使用 Visual Studio 中内置的 Web 应用程序初学者项目的实例，也可使用您自己的项目。

> **Note**
>
> 此向导还支持部署 ASP.NET 内核应用程序。有关 ASP.NET Core 的信息，请参阅 AWS.NET 部署工具指南和更新后的 “部署到AWS目录”。

> **Note**
>
> 必须先下载并安装 Web Deploy，然后才能使用 “发布到 Elastic Beanstalk” 向导。此向导依赖 Web Deploy 将 Web 应用程序和网站部署到 Internet Information Services (IIS) Web 服务器。

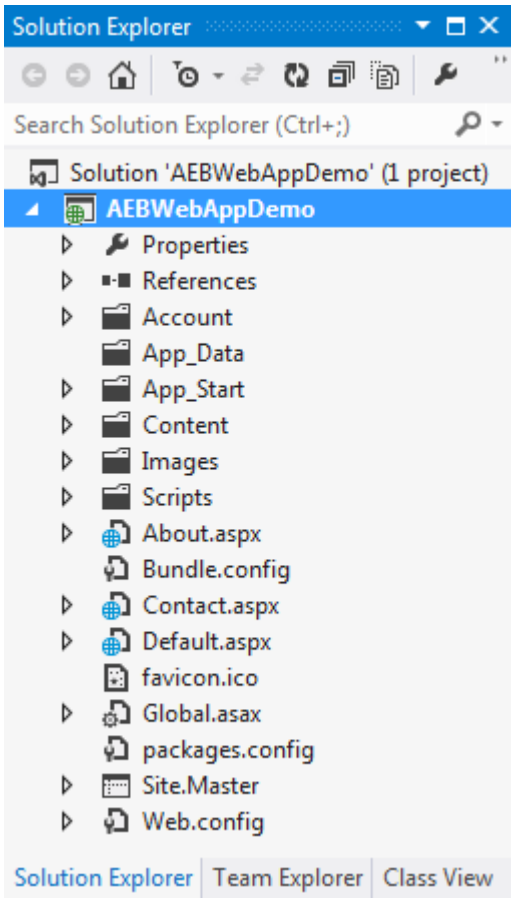
创建示例 Web 应用程序初学者项目

1. 在 Visual Studio 中，从 File (文件) 菜单中，选择 New (新建)，然后选择 Project (项目)。
2. 在 New Project (新建项目) 对话框的导航窗格中，依次展开 Installed (已安装)、Templates (模板) 和 Visual C#，然后选择 Web。
3. 在 Web 项目模板的列表中，选择其说明中包含 `Web` 和 `Application` 字样的任何模板。在本示例中，请选择 ASP.NET Web Forms Application (ASP.NET Web 表单应用程序)。



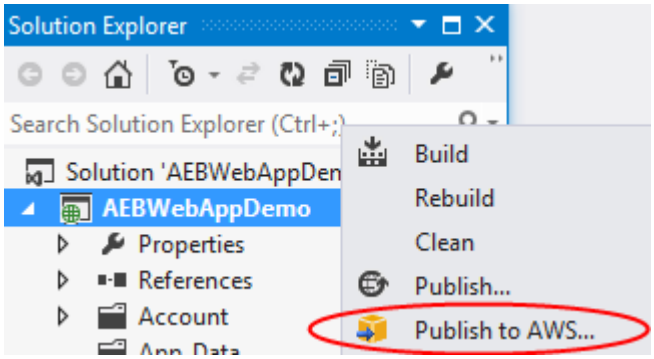
4. 在 Name (名称) 框中，键入 `AEBWebAppDemo`。
5. 在 Location (位置) 框中，键入您的开发计算机上的解决方案文件夹的路径或选择 Browse (浏览)，然后浏览并选择解决方案文件夹，再选择 Select Folder (选择文件夹)。
6. 确认选中了 Create directory for solution (为解决方案创建目录) 框。在 Solution (解决方案) 下拉列表中，确认选择了 Create new solution (创建新解决方案)，然后选择 OK (确定)。Visual Studio 将

基于 ASP.NET Web 表单应用程序项目模板创建解决方案和项目。随后，Visual Studio 将显示解决方案资源管理器，其中将显示新的解决方案和项目。

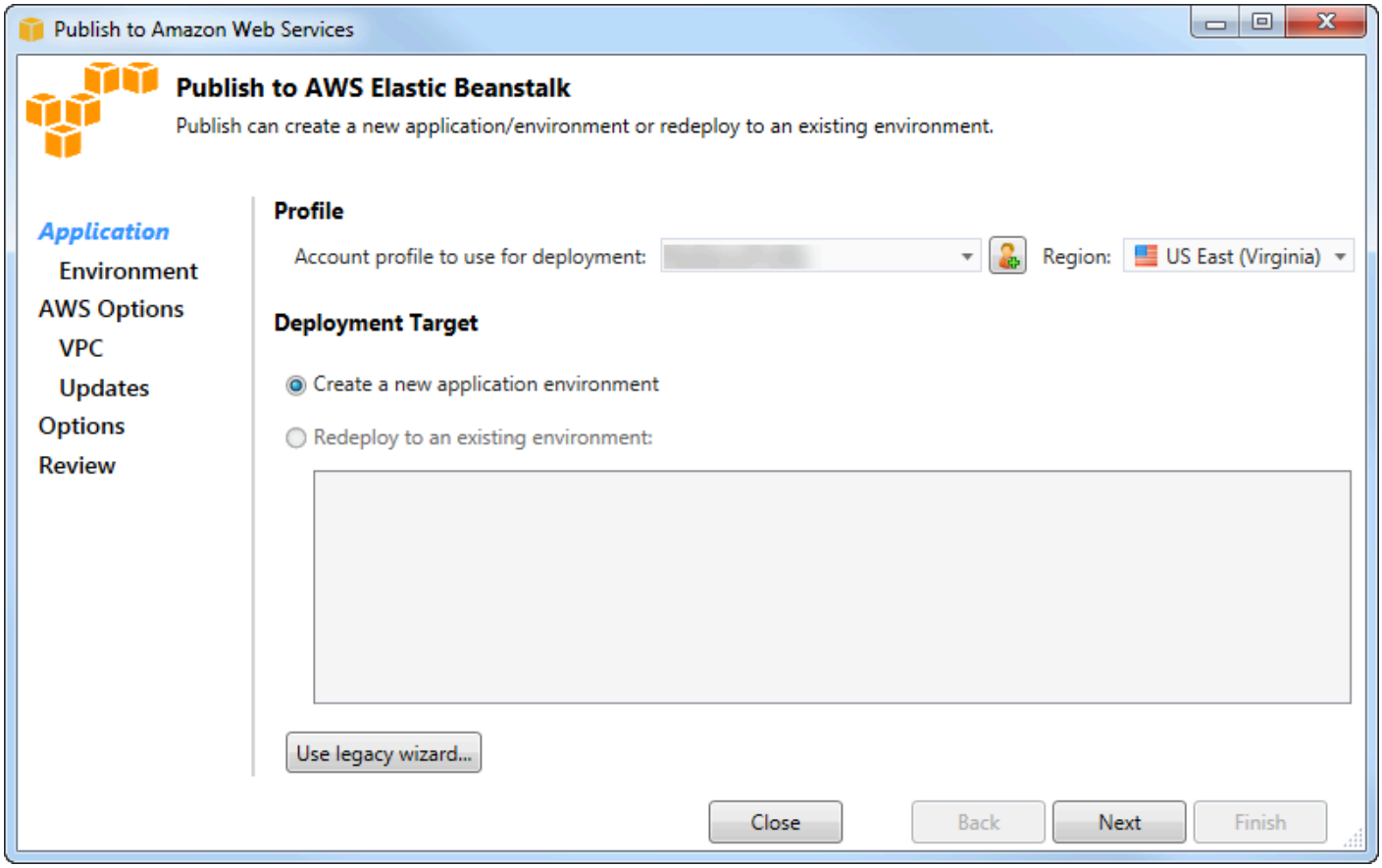


使用 Publish to Elastic Beanstalk 向导部署应用程序

1. 在 “解决方案资源管理器” 中，打开您在上一节中创建的项目的 AEBWebAppDemo 项目文件夹的上下文（右键单击）菜单，或者打开您自己的应用程序的项目文件夹的快捷菜单，然后选择 “发布到 EAWS lastic Beanstalk”。



随即显示 Publish to Elastic Beanstalk (发布到 Elastic Beanstalk) 向导。



2. 在 “配置文件” 中，从 “用于部署的帐户配置文件” 下拉列表中，选择要用于部署的AWS帐户配置文件。

   （可选）如果您有想要使用的AWS账户，但尚未为其创建AWS账户资料，则可以选择带有加号 (+) 的按钮来添加AWS账户资料。

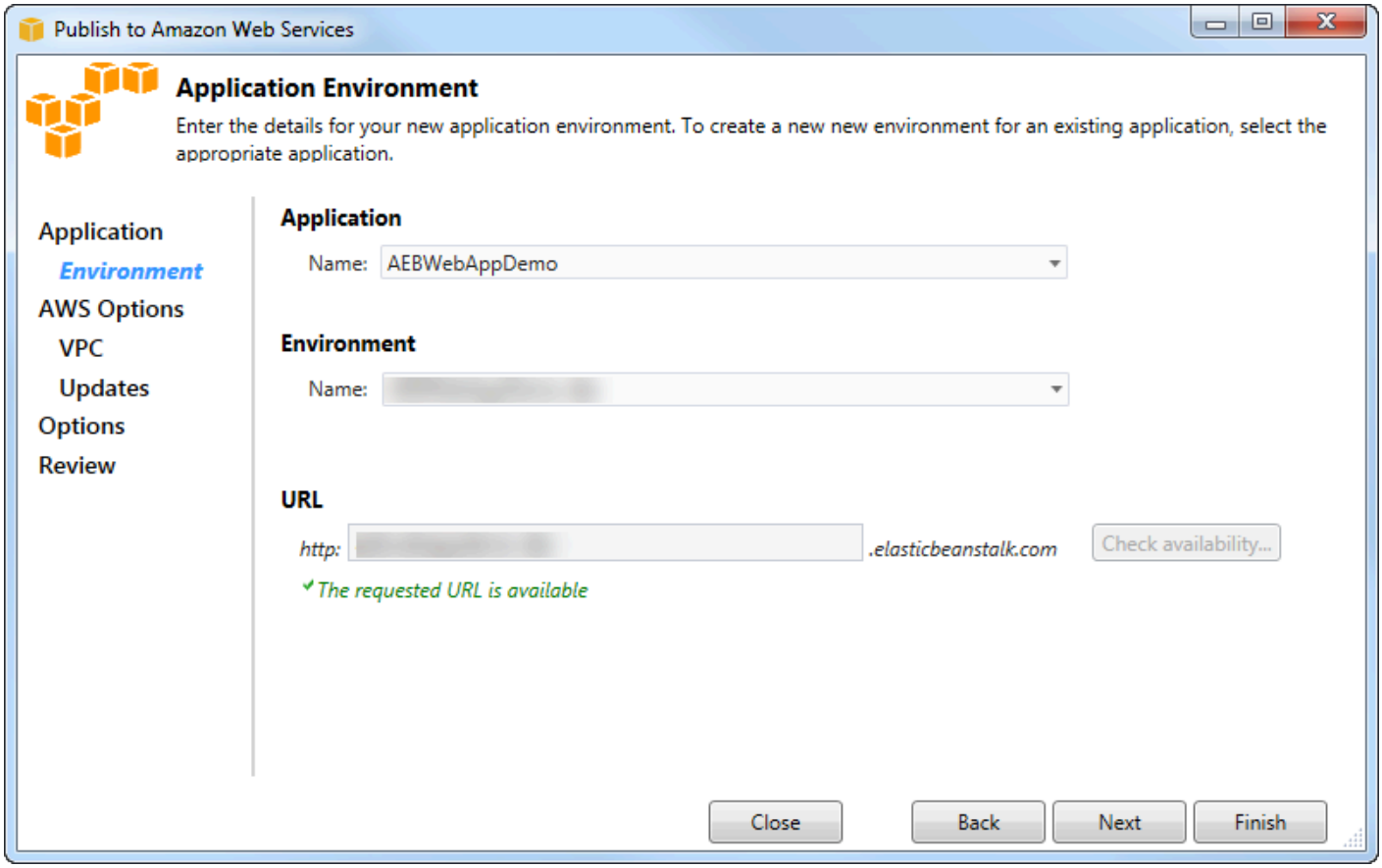
3. 从区域下拉列表中，选择您希望 Elastic Beanstalk 将应用程序部署到的区域。

4. 在 Deployment Target (部署目标) 中，您可选择 Create a new application environment (创建新应用程序环境) 执行应用程序的初始部署或选择 Redeploy to an existing environment (重新部署到现有环境) 重新部署之前已部署的应用程序。（之前的部署可能是使用向导或已过时的独立部署工具执行的。）如果您选择 Redeploy to an existing environment (重新部署到现有环境)，则当向导从当前正在运行的之前的部署中检索信息时可能会出现延迟。

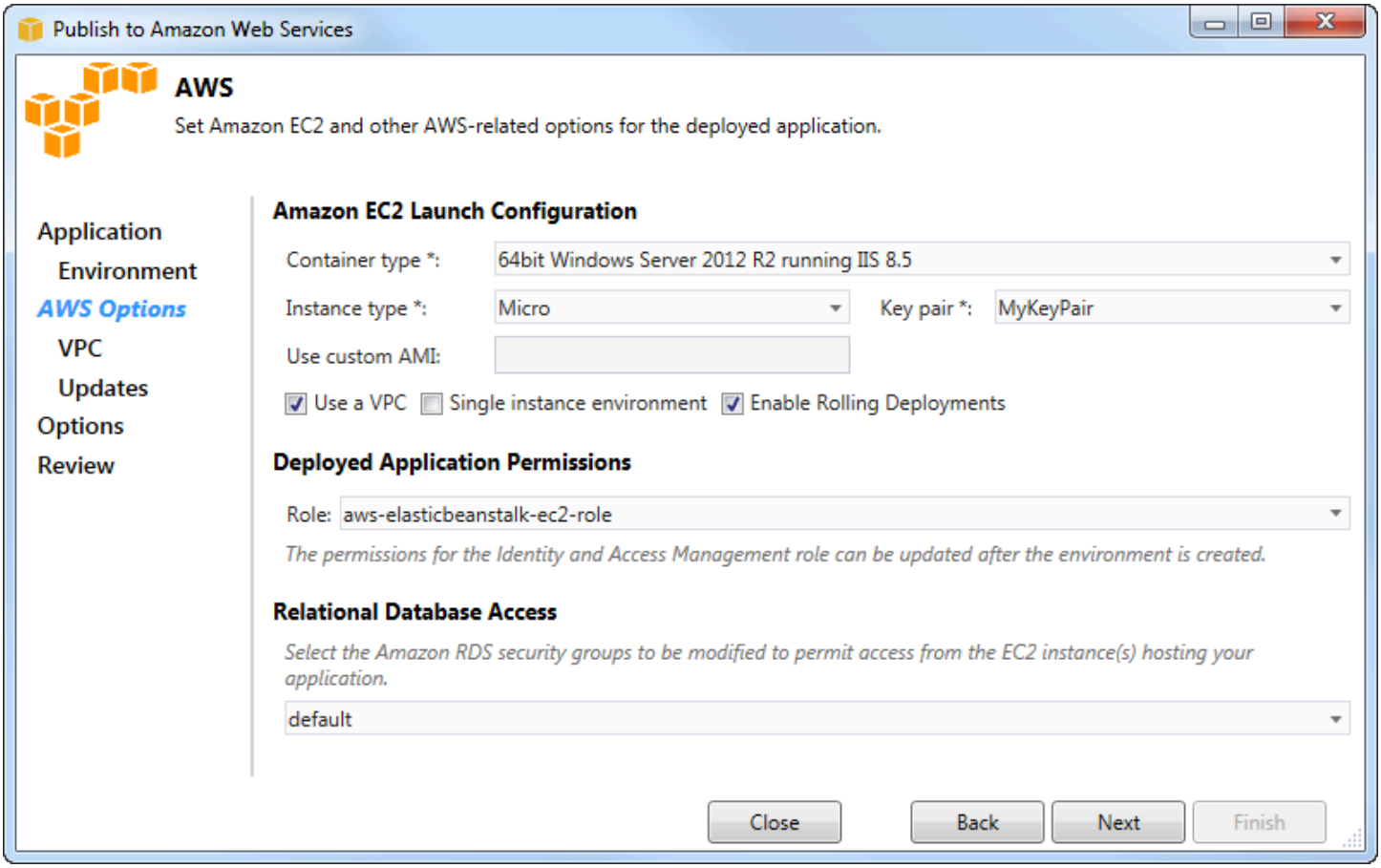
> **Note**
>
> 如果您选择 Redeploy to an existing environment (重新部署到现有环境)，再选择列表中的环境，然后选择 Next (下一步)，则向导会将您定向至 Application Options (应用程序选项)

页面。如果您执行此过程，请向前跳至此部分中后面描述如何使用 Application Options (应用程序选项) 页面的说明。

5. 选择 Next（下一步）。



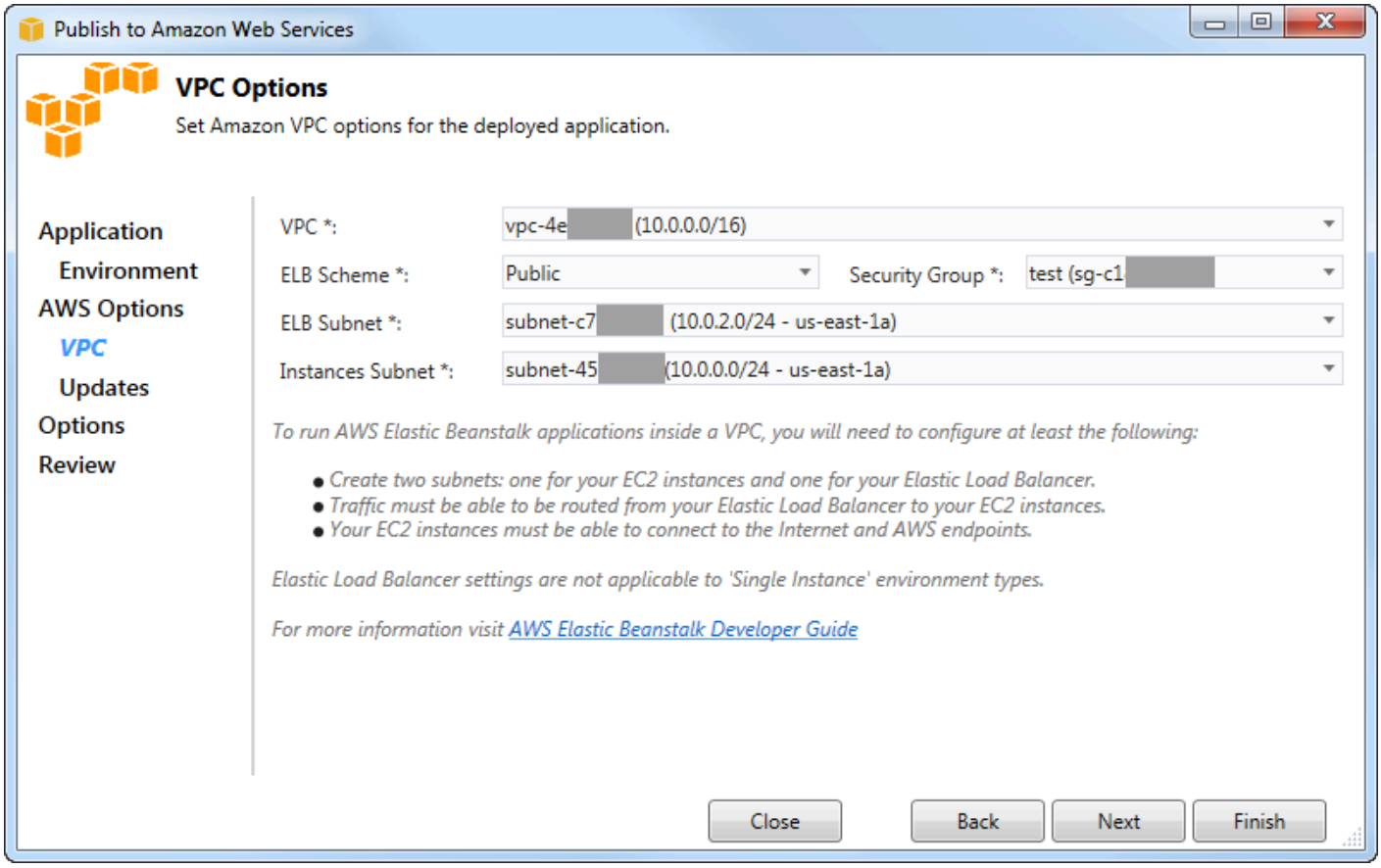
6. 在 Application Environment (应用程序环境) 页面上的 Application (应用程序) 区域中，Name (名称) 下拉列表将为应用程序建议默认名称。您可通过选择此下拉列表中的其他名称来更改默认名称。
7. 在环境区域的名称下拉列表中，键入您的 Elastic Beanstalk 环境的名称。在这种情况下，“环境”一词是指适用于您的应用程序的基础设施 Elastic Beanstalk 规定。此下拉列表中可能已建议默认名称。如果未建议默认名称，您可键入一个名称或从下拉列表中选择一个名称（如果提供了任何其他名称）。环境名称的长度不得超过 23 个字符。
8. 在 URL 区域中，此框会建议将作为您的 Web 应用程序 URL 的默认子域 `.elasticbeanstalk.com`。您可键入新的子域名来更改默认子域。
9. 选择 Check availability (检查可用性) 以确保您的 Web 应用程序 URL 未在使用中。
10. 如果您的 Web 应用程序 URL 可以使用，请选择 Next (下一步)。



1. 在AWS选项页面的 Amazon EC2 启动配置中，从容器类型下拉列表中选择将用于您的应用程序的亚马逊系统映像 (AMI) 类型。
2. 在实例类型下拉列表中，指定要使用的 Amazon EC2 实例类型。在本示例中，我们建议您使用 Micro (微型)。这将最大程度降低相关的实例运行成本。有关 Amazon EC2 Amazon Ser vic2 C
3. 在 key pair 下拉列表中，选择一个 Amazon EC2 实例密钥对，用于登录将用于您的应用程序的实例。
4. （可选）在 Use custom AMI (使用自定义 AMI) 框中，您可指定将覆盖 Container type (容器类型) 下拉列表中指定的 AMI 的自定义 AMI。有关如何创建自定义 AMI 的更多信息，请转到 Elasti AWSc Beanstalk 开发人员指南中的使用自定义 AM I 和从 Amazon EC2 实例创建 AMI。
5. （可选）如果您要在 VPC 中启动实例，请选中 Use a VPC (使用 VPC) 框。
6. （可选）如果您想启动单个 Amazon EC2 实例，然后将应用程序部署到该实例，请选中 “单实例环境” 复选框。

   如果您选中此复选框，Elastic Beanstalk 仍会创建 Auto Scaling 组，但不会对其进行配置。如果您想稍后配置 Auto Scaling 组，则可以使用AWS Management Console。

7. （可选）如果您希望控制将应用程序部署到实例时的条件，请选中 Enable Rolling Deployments (启用滚动部署) 框。只能在未选中 Single instance environment (单个实例环境) 框时选中此框。
8. 如果您的应用程序使用 Amazon S3 和 DynamoDB 等AWS服务，则提供证书的最佳方法是使用 IAM 角色。在 “已部署应用程序权限” 区域中，您可以选择现有的 IAM 角色，也可以创建向导将用于启动环境的角色。使用的应用程序在向AWS服务发出请求时AWS SDK for .NET将自动使用此 IAM 角色提供的证书。
9. 如果您的应用程序访问 Amazon RDS 数据库，请在关系数据库访问区域的下拉列表中，选中向导将更新的任何 Amazon RDS 安全组旁边的复选框，以便您的 Amazon EC2 实例可以访问该数据库。
10. 选择 Next（下一步）。
    - 如果您已选择 Use a VPC (使用 VPC)，则将显示 VPC Options (VPC 选项) 页面。
    - 如果您已选择 Enable Rolling Deployments (启用滚动部署)，但未选择 Use a VPC (使用 VPC)，则将显示 Rolling Deployments (滚动部署) 页面。向前跳至此部分中后面描述如何使用 Rolling Deployments (滚动部署) 页面的说明。
    - 如果您未选择 Use a VPC (使用 VPC) 或 Enable Rolling Deployments (启用滚动部署)，则将显示 Application Options (应用程序选项) 页面。向前跳至此部分中后面描述如何使用 Application Options (应用程序选项) 页面的说明。
11. 如果您已选择 Use a VPC (使用 VPC)，请在 VPC Options (VPC 选项) 页面上指定信息以在 VPC 中启动应用程序。



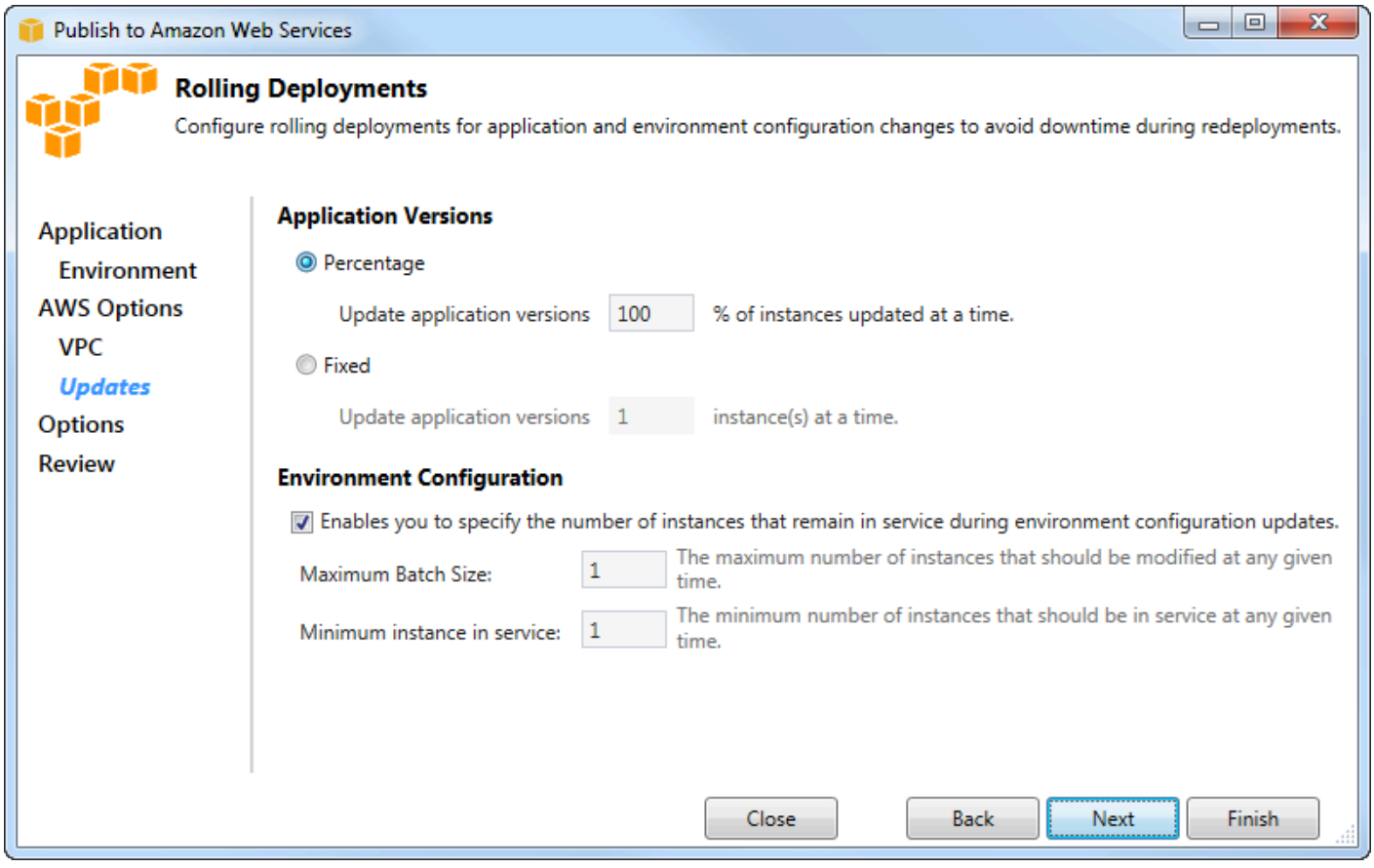
必须已创建 VPC。如果您在 VToolkit for Visual Studio 中创建了 VPC，则 VToolkit for Visual Studio 将为您填充此页面。如果您在AWS管理控制台中创建了 VPC，请在此页面中键入有关您的 VPC 的信息。

针对 VPC 的部署的主要注意事项

- 您的 VPC 需要至少一个公有子网和一个私有子网。
- 在 ELB Subnet (ELB 子网) 下拉列表中，指定公有子网。Toolkit for Visual Studio 将应用程序的弹性负载平衡负载均衡器部署到公有子网。公有子网与具有指向 Internet 网关的入口的路由表关联。您可识别 Internet 网关，因为它具有以 `igw-` 开头的 ID（例如，`igw-83cddaex`）。您使用适用于 Visual Studio 的工具包创建的公用子网具有可将其标识为公共子网的标签值。
- 在 Instances Subnet (实例子网) 下拉列表中，指定私有子网。适用于 Visual Studio 的工具包将您的应用程序的 Amazon EC2 实例部署到私有子网。
- 您的应用程序的 Amazon EC2 实例通过公有子网中执行网络地址转换 (NAT) 的 Amazon EC2 实例从私有子网与互联网通信。要启用此通信，您需要允许流量从私有子网流至 NAT 实例的 VPC 安全组。在 Security Group (安全组) 下拉列表中，指定此 VPC 安全组。

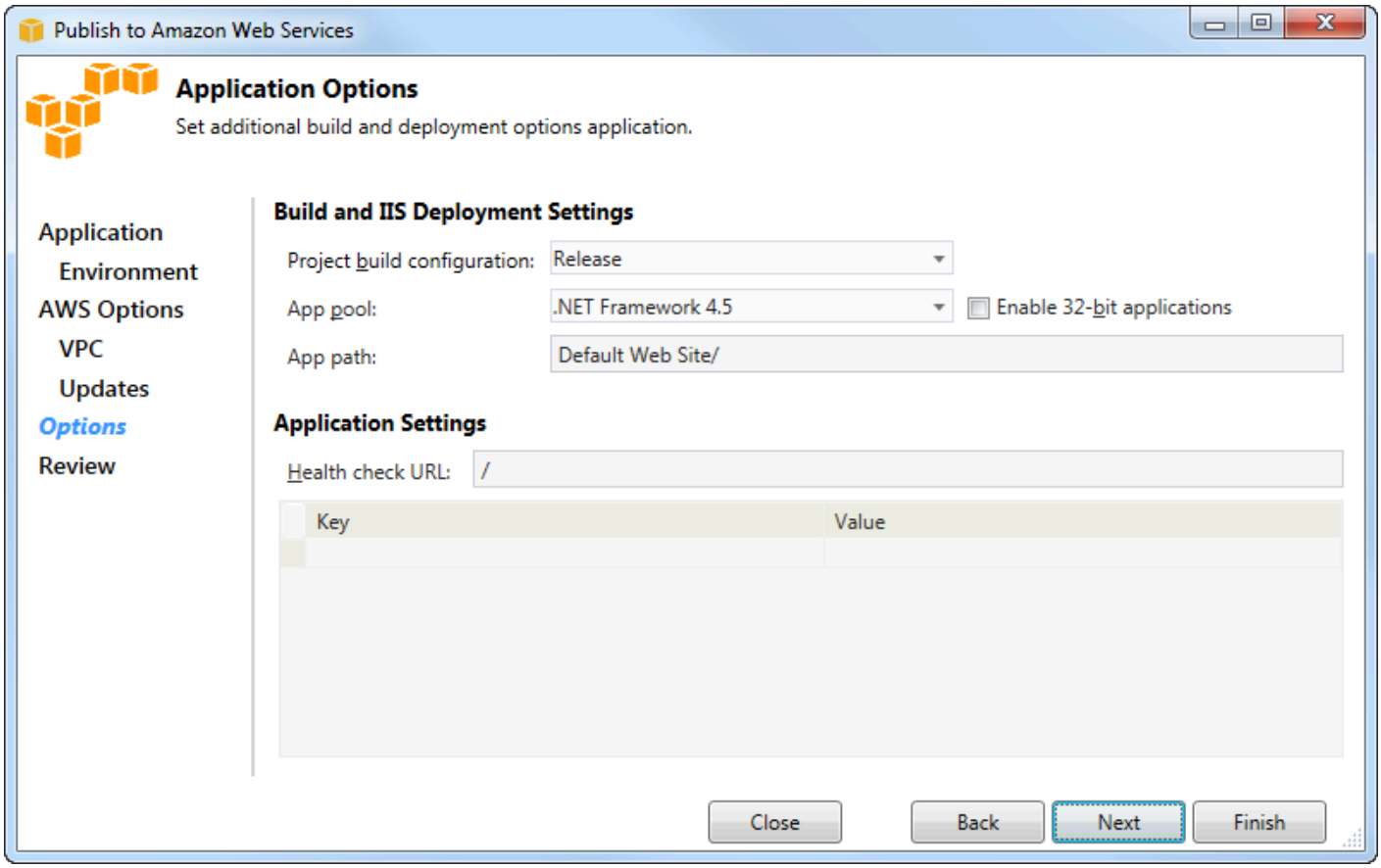
有关如何将 Elastic Beanstalk 应用程序部署到 VPC 的更多信息，请访问 EAWS lastic Beanstalk 开发者指南。

1. 在填充 VPC Options (VPC 选项) 页面上的所有信息后，请选择 Next (下一步)。
   - 如果您已选择 Enable Rolling Deployments (启用滚动部署)，则将显示 Rolling Deployments (滚动部署) 页面。
   - 如果您未选择 Enable Rolling Deployments (启用滚动部署)，则将显示 Application Options (应用程序选项) 页面。向前跳至此部分中后面描述如何使用 Application Options (应用程序选项) 页面的说明。
2. 如果您已选择 Enable Rolling Deployments (启用滚动部署)，请在 Rolling Deployments (滚动部署) 页面上指定信息以配置新版本的应用程序部署到负载均衡环境中的实例的方式。例如，如果您的环境中有 4 个实例，并且您需要更改实例类型，则可将环境配置为一次更改 2 个实例。这可帮助确保您的应用程序在执行更改时仍处于运行状态。



3. 在 Application Versions (应用程序版本) 区域中，选择用于控制一次部署的实例的百分比或数量的选项。指定所需百分比或数量。

4. （可选）在 Environment Configuration (环境配置) 区域中，如果要指定在部署期间保持运行的实例数量，请选中此框。如果选中此框，请指定一次应修改的实例的最大数目和/或一次应保持运行的实例的最小数目。
5. 选择 Next（下一步）。
6. 在 Application Options (应用程序选项) 页面上，指定有关版本、Internet Information Services (IIS) 和应用程序设置的信息。



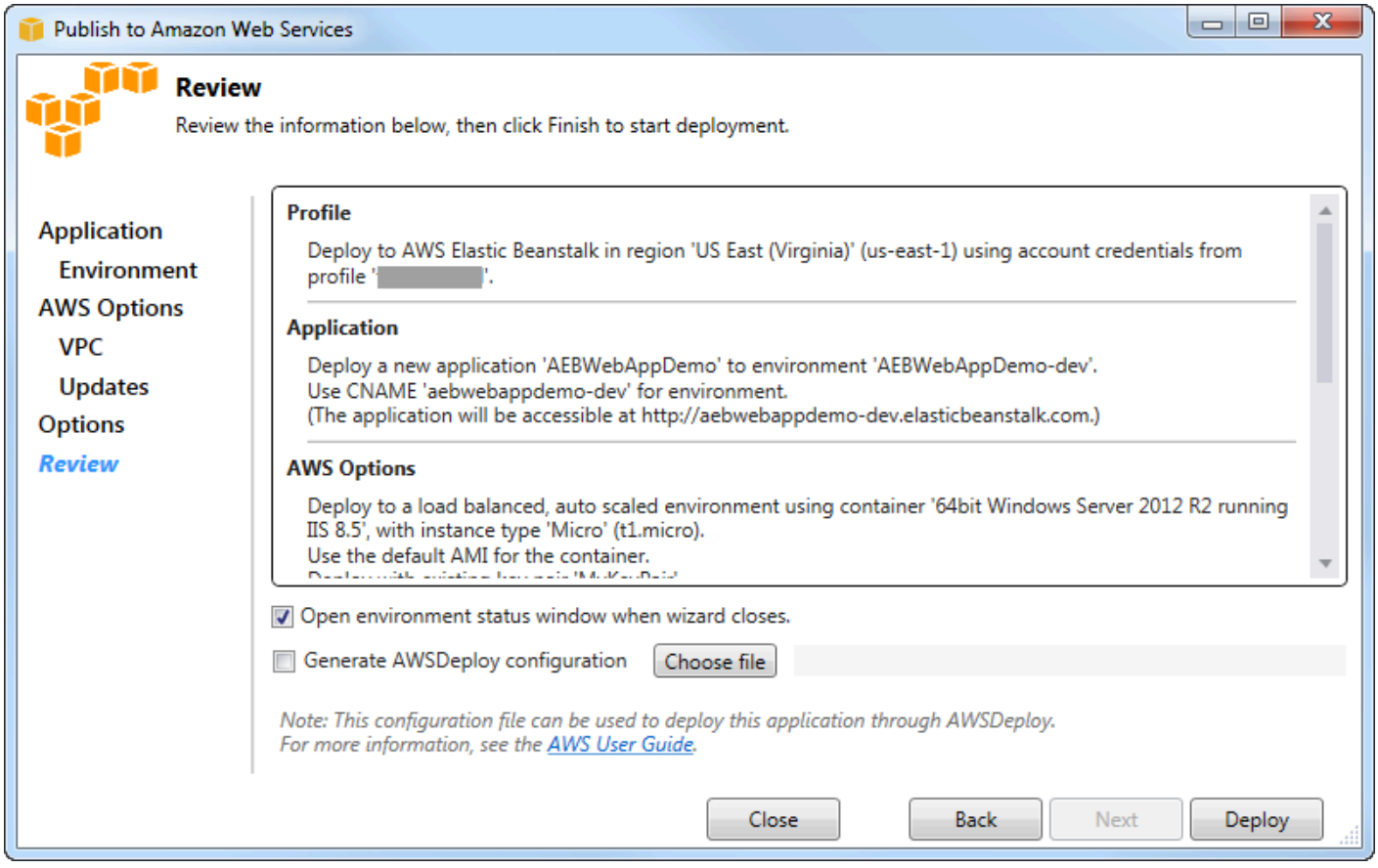
7. 在 Build and IIS Deployment Settings (生成和 IIS 部署设置) 区域的 Project build configuration (项目生成配置) 下拉列表中，选择目标版本配置。如果向导可以找到它，则 Release (发布) 将显示，否则此框中将显示有效配置。
8. 在 App pool (应用程序池) 下拉列表中，选择您的应用程序所需的 .NET Framework 版本。应已显示正确的 .NET Framework 版本。
9. 如果您的应用程序是 32 位的，请选中 Enable 32-bit applications (启用 32 位应用程序) 框。
10. 在 App path (应用程序路径) 框中，指定 IIS 将用来部署应用程序的路径。默认情况下，指定 Default Web Site/，它通常会转换为路径 `c:\inetpub\wwwroot`。如果您指定了 Default Web Site/ 之外的路径，向导将在 Default Web Site/ 路径中放置指向您指定的路径的重定向。

11.在 “应用程序设置” 区域的 “Healt h 检查 URL” 框中，键入 Elastic Beanstalk 的 URL 以检查以确定您的 Web 应用程序是否仍在响应。此 URL 相对于根服务器 URL。默认情况下，已指定根服务器 URL。例如，如果完整 URL 为 `example.com/site-is-up.html`，则键入 `/site-is-up.html`。

12.在 Key (键) 和 Value (值) 的区域中，可指定要添加到应用程序的 `Web.config` 文件的任何密钥和值对。

> **Note**
>
> 尽管不推荐，但您可以使用密钥和值区域来指定应用程序应在哪些AWS证书下运行。首选方法是在AWS选项页面的Id entity and Access Management 角色下拉列表中指定 IAM 角色。但是，如果您必须使用AWS证书而不是 IAM 角色来运行应用程序，请在密钥行中选择AWSAccessKey。在 Value (值) 行中，键入访问密钥。重复这些步骤AWSSecretKey。

13.选择 Next（下一步）。



14.在 Review (查看) 页面上，查看您配置的选项，然后选中 Open environment status window when wizard closes (在向导关闭时打开环境状态窗口) 框。

15.如果一切正常，请选择 Deploy (部署)。

> **Note**
>
> 部署应用程序时，活动账户将为应用程序使用的AWS资源付费。

有关部署的信息将显示在 Visual Studio 状态栏和 Output (输出) 窗口中。该过程可能需要几分钟。部署完成后，Output (输出) 窗口中将显示确认消息。

16要删除部署，在AWS Explorer 中，展开 E lastic Beanstalk Beanstalk 节点，为部署的子节点打开上下文（右键单击）菜单（右键单击）菜单，然后选择 Delete Delete Delete (删除) 此删除过程可能需要几分钟。

将 ASP.NET Core 应用程序部署到 Elastic Beanstalk（传统）

> **Important**
>
> 本文档涉及旧版服务和功能。有关更新的指南和内容，请参阅 AWS.NET 部署工具指南和更新后的 “部署到AWS目录”。

AWS Elastic Beanstalk是一项简化应用程序AWS资源配置过程的服务。 AWS Elastic Beanstalk提供部署应用程序所需的所有AWS基础架构。

Toolkit for Visual Studio 支持AWS使用 Elastic Beanstalk 部署 ASP.NET Core 应用程序。ASP.NET 内核是对 ASP.NET 的重新设计，具有模块化的架构，它最大程度地降低了依赖项开销并简化了应用程序以便在云中运行。

AWS Elastic Beanstalk可以轻松地将各种不同语言的应用程序部署到AWS。Elastic Beanstalk 支持传统的 ASP.NET 应用程序和 ASP.NET 核心应用程序。本主题描述如何部署 ASP.NET 内核应用程序。

使用部署向导

将 ASP.NET Core 应用程序部署到 Elastalk Elastic Beanstalk 的最简单方法是使用 Toolkit for Visual Studio 部署 AS

如果您之前用过此工具包部署传统 ASP。NET 应用程序，您将发现 ASP.NET 内核的体验与之非常相似。在以下步骤中，我们将演练部署体验。

如果您之前从未用过此工具包，则在安装此工具包后首先需要使用此工具包注册您的AWS凭证。有关如何操作的详细信息，请参阅如何为 Visual Studio 应用程序指定AWS安全凭证文档。

要部署 ASP.NET Core Web 应用程序，请在解决方案资源管理器中右键单击该项目，然后选择 “发布到AWS...”。

在 “发布到AWS Elastic Beanstalk部署” 向导的第一页上，选择创建新的 Elastic Beanstalk 应用程序。Elastic Beanstalk 应用程序 是 Elastic Beanstalk 组件的逻辑集合，包括环境、版本 和环境配置。此部署向导将生成一个应用程序，此应用程序将包含应用程序版本和环境的集合。环境包含运行应用程序版本的实际AWS资源。每次部署应用程序时，都会创建一个新的应用程序版本，并且此向导会将环境指向此版本。您可在 Elastic Beanstalk 组件中了解有关这些概念的更多信息。

接下来，为应用程序及其第一个环境设置名称。每个环境均关联一个唯一的别名，可使用此别名在部署完成后访问应用程序。

下一页 “AWS选项” 允许您配置要使用的AWS资源类型。在此示例中，将保留默认值（Key pair (密钥对) 部分除外）。利用密钥对，可以检索 Windows 管理员密码，以便您能登录到计算机。如果您尚未创建密钥对，可能需要选择 Create new key pair (创建新密钥对)。

权限

权限页面用于为运行您的应用程序的 EC2 实例分配AWS证书。如果您的应用程序使用访问其他AWS服务，AWS SDK for .NET这一点很重要。如果您未使用应用程序中的任何其他服务，则可将保留此页面的默认值。

应用程序选项

Application Options (应用程序选项) 页面上的详细信息不同于部署传统 ASP.NET 应用程序时指定的详细信息。在此处，可指定用于打包应用程序的版本配置和框架，还可指定应用程序的 IIS 资源路径。

完成 Application Options (应用程序选项) 页面后，单击 Next (下一步) 查看设置，然后单击 Deploy (部署) 开始部署过程。

检查环境状态

将应用程序打包并上传到后AWS，您可以通过在 Visual Studio 的AWS资源管理器中打开环境状态视图来检查 Elastic Beanstalk 环境的状态。

事件将在环境联机时显示在状态栏中。一切完成后，环境状态将变为正常状态。您可单击 URL 来查看站点。从这里，您还可以将日志从环境或远程桌面提取到作为 Elastic Beanstalk 环境一部分的 Amazon EC2 实例中。

任何应用程序的首次部署都会比后续的重新部署花费更长的时间，因为它会创建新的AWS资源。在开发期间对应用程序执行迭代时，可再次通过向导快速重新部署，或通过在右键单击项目时选择 Republish (重新发布) 选项来快速重新部署。

使用先前运行部署向导的设置重新发布打包应用程序，并将应用程序包上传到现有的 Elastic Beanstalk 环境。

如何指定AWS应用程序的安全凭证

这些区域有：AWS您在中指定的账户发布到 Elastic Beanstalk向导是AWS向导将使用账户部署到 Elastic Beanstalk。

尽管不推荐使用，但您可能还需要指定AWS您的应用程序将用于访问的账户凭证AWS部署后的服务。首选方法是指定一个 IAM 角色。在发布到 Elastic Beanstalk向导，您可以通过IdIdentity and Access Management s下拉列表中)AWS选项页. 在传统发布到 Amazon Web Services向导，您可以通过IAM角色下拉列表中)AWS选项页.

如果您必须使用AWS账户凭证而不是 IAM 角色，您可以指定AWS通过以下方法之一为应用程序的账户凭证：

- 引用对应于的配置文件AWS中的 AccentityappSettings项目的元素Web.config文件。（要创建配置文件，请参阅配置AWS凭证.) 以下示例指定了配置文件名称为 myProfile 的凭证。

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

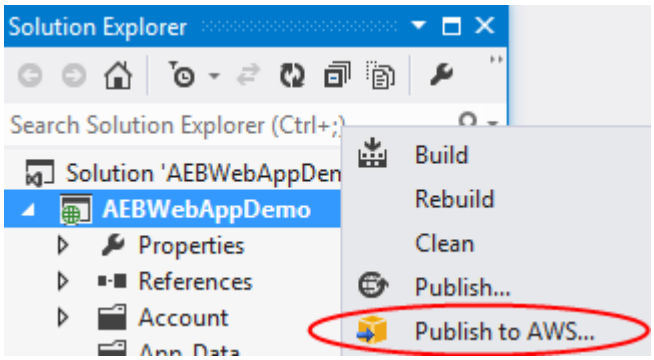
- 如果您使用的是发布到 Elastic Beanstalk向导，在应用程序选项页面中)，在密钥的 row of the密钥和值区域，选择AWSAccessKey. 在 Value (值) 行中，键入访问密钥。对重复这些步骤AWSSecretKey.
- 如果您使用的是旧的 Publish to Amazon Web Services (发布到 Amazon Web Services) 向导，则在 Application Options (应用程序选项) 页面上的 Application Credentials (应用程序凭证) 区域中，选择 Use these credentials (使用这些凭证)，然后将访问密钥和秘密访问密钥键入到 Access Key (访问密钥) 和 Secret Key (私有密钥) 框。

如何将您的应用程序重新发布到 Elastic Beanstalk 环境（传统）

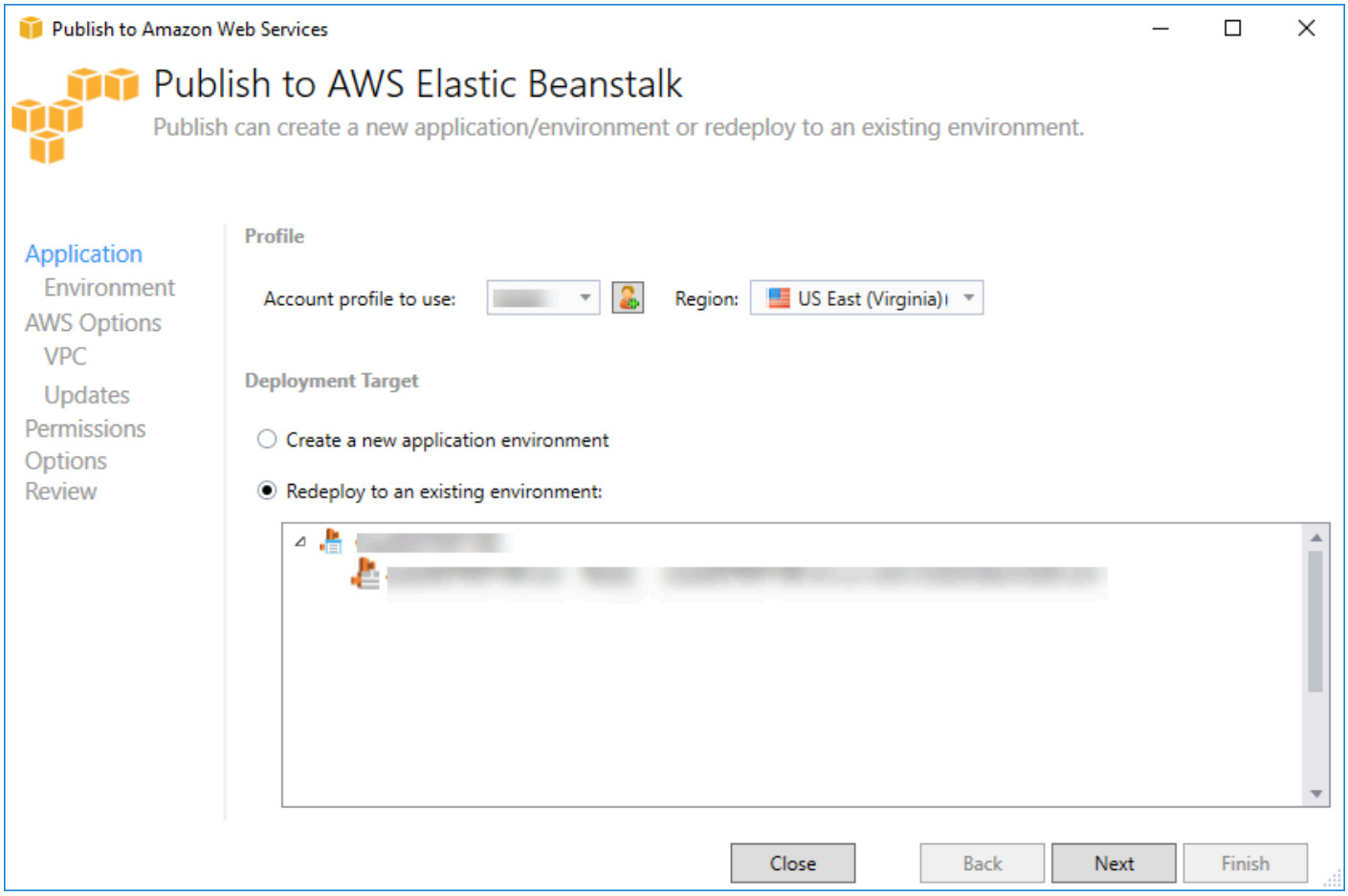
> **Important**
>
> 本文档涉及旧版服务和功能。有关更新的指南和内容，请参阅 AWS.NET 部署工具指南和更新后的 “部署到AWS目录”。

您可以对应用程序进行迭代，方法是进行离散的更改，然后将新版本重新发布到已启动的 Elastic Beanstalk 环境中。

1. 在 Solution Explorer (解决方案资源管理器) 中，打开您在上一节中发布的WebAppDemo项目的 AEB 项目文件夹的上下文（右键单击）菜单，然后选择 Publish to (发布到)AWS Elastic Beanstalk。

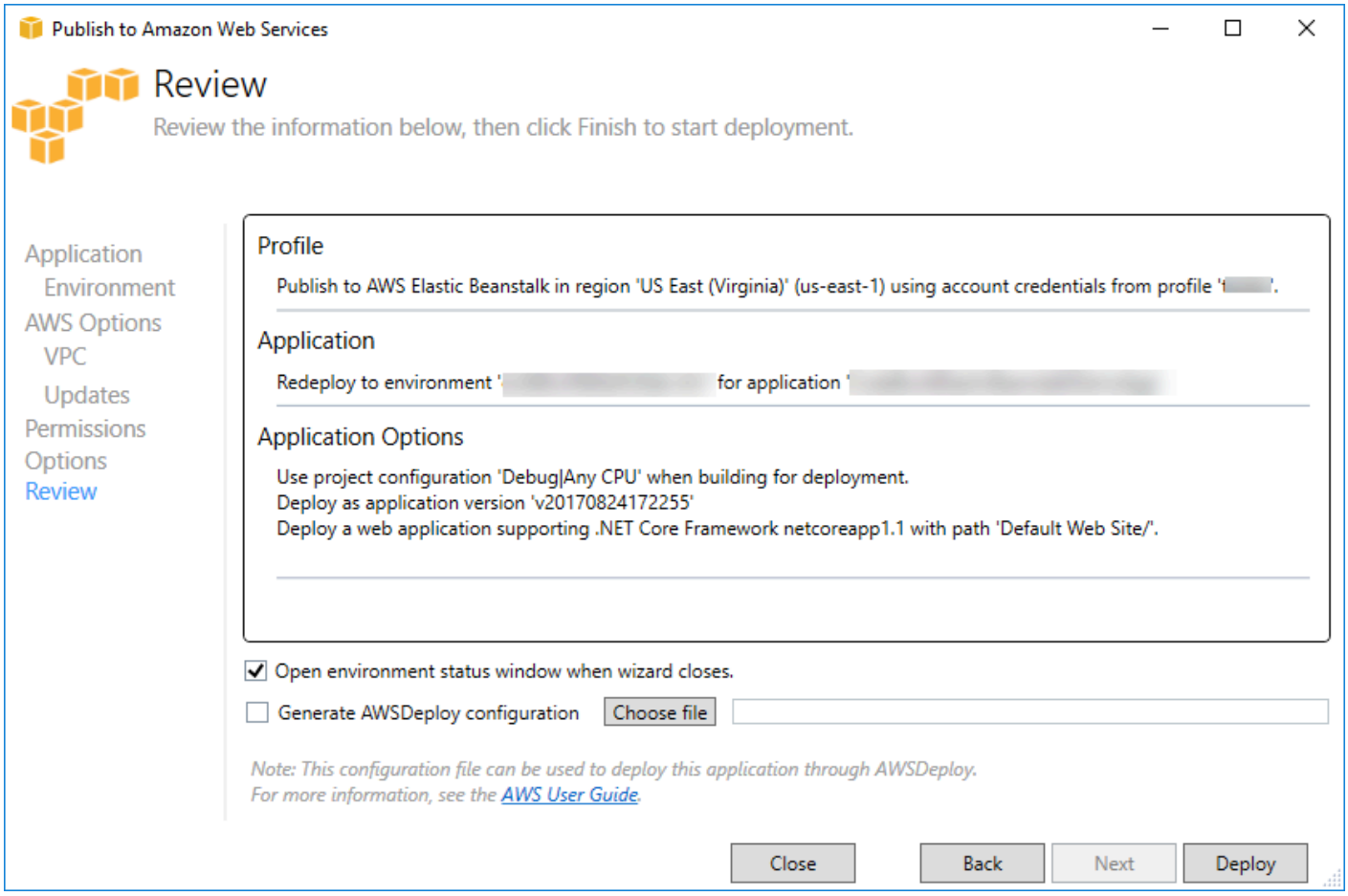


随即显示 Publish to Elastic Beanstalk (发布到 Elastic Beanstalk) 向导。



2. 选择 Redeploy to an existing environment (重新部署到现有环境)，然后选择您之前所发布到的环境。单击 Next（下一步）。

   Review (查看) 向导随即出现。



3. 单击 Deploy (部署)。该应用程序将重新部署到相同的环境。

如果您的应用程序正处于启动或终止过程，则无法重新发布。

自定义 Elastic Beanstalk 应用程序部署

本主题介绍 Elastic Beanstalk 的 Microsoft Windows 容器的部署清单如何支持自定义应用程序部署。

对于想利用 Elastic Beanstalk 的功能来创建和管理其功能的高级用户，自定义应用程序部署是一个强大的功能。AWS资源，但希望完全控制其应用程序的部署方式。对于自定义应用程序部署，您将为 Elastic Beanstalk 执行的三个不同的操作创建 Windows PowerShell 脚本。安装操作在启动部署时使用，重新启动操作在从工具包或 Web 控制台调用 `RestartAppServer` API 时使用，卸载操作在新部署出现时被任何之前的部署调用。

例如，当您的文档团队编写了一个他们希望包含在部署中的静态网站情况下，您可能希望部署一个 ASP.NET 应用程序。您可以按如下方式编写部署清单来执行该操作：

```
{
```

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

为每个操作列出的脚本必须位于与部署清单文件相关的应用程序包中。在本示例中，应用程序包还将包含一个 documentation.zip 文件，该文件包含由您的文档团队创建的静态网站。

`install.ps1` 脚本将提取该 zip 文件并设置 IIS 路径。

```
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}
```

由于您的应用程序在 IIS 中运行，重新启动操作将调用 IIS 重置。

```
iisreset /timeout:1
```

若要卸载脚本，则务必清除在安装阶段使用的所有设置和文件。这样，在新版本的安装阶段，您可以避免与以前的部署之间的所有冲突。在本示例中，您需要删除静态网站的 IIS 应用程序并删除网站文件。

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

由于这些脚本文件以及 documentation.zip 文件包含在您的应用程序包中，该部署将创建 ASP.NET 应用程序，然后部署文档站点。

在本示例中，我们将选择一个部署简单静态网站的简单示例，但通过自定义应用程序部署，您可以部署任何类型的应用程序并让 Elastic Beanstalk 管理AWS它的资源。

自定义 ASP.NET 内核 Elastic Beanstalk 部署

本主题介绍了部署的工作方式，以及在利用 Elastic Beanstalk 和 Visual Studio Toolkit 创建 ASP.NET 核心应用程序时如何自定义部署。

在 Visual Studio Toolkit 中完成部署向导后，Toolkit 将对该应用程序打包并将其发送到 Elastic Beanstalk。创建应用程序包的第一步是借助新的 dotnet CLI 为应用程序做好使用 publish 命令进行发布的准备。框架和配置将从向导中的设置向下传递到 publish 命令。所以如果你选择版本为了`configuration`和netcoreapp1.0(对于)`framework`，Toolkit 将执行以下命令：

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

当 publish 命令完成后，Toolkit 会将新的部署清单写入到发布文件夹。部署清单是名为的 JSON 文件aws-windows-deployment-manifest.jsonElastic Beanstalk Windows 容器（版本 1.2 或更高版本）将读取该文件以确定如何部署应用程序。例如，对于要在 IIS 的根处部署的 ASP.NET 内核应用程序，Toolkit 将生成一个清单文件，如下所示：

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

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

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

appBundle 属性指示了应用程序位与清单文件相关的位置。此属性可指向目录或 ZIP 存档。iisPath 和 iisWebSite 属性指示了 IIS 中要托管应用程序的位置。

自定义清单

如果某个清单文件在发布文件夹中尚不存在，则 Toolkit 仅写入该清单文件。如果该文件已存在，Toolkit 将更新该清单的 aspNetCoreWeb 部分下列出的第一个应用程序中的 appBundle、iisPath 和 iisWebSite 属性。这使您可以将 aws-windows-deployment-manifest.json 添加到您的项目并自定义该清单。要对 Visual Studio 中的 ASP.NET 内核 Web 应用程序执行此操作，请将新的 JSON 文件添加到项目的根并将其命名为 aws-windows-deployment-manifest.json。

该清单必须命名为 aws-windows-deployment-manifest.json 且必须位于项目的根处。Elastic Beanstalk 容器将在根中寻找该清单，如果找到，则会调用部署工具。如果该文件不存在，Elastic Beanstalk 容器将回退到较早的部署工具，该工具假定存档为msdeploy存档。

要确保 dotnet CLI publish 命令包含该清单，请更新 project.json 文件以将该清单文件包含在 publishOptions 中 include 下的 include 部分。

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

既然您已声明该清单以便让它包含在应用程序包中，您可以进一步配置要部署应用程序的方式。您可以自定义除部署向导支持的部署之外的部署。AWS已为aws-windows-deployment-manifest.json 文件，并且在您为 Visual Studio 安装 Toolkit 后，该安装程序为架构注册了 URL。

当您打开 `windows-deployment-manifest.json` 时，您将看到在“Schema (架构)”下拉框中选择的架构 URL。您可以导航到该 URL 以获取可在该清单中设置的内容的完整说明。在已选择该架构的情况下，Visual Studio 将在您编辑该清单时提供 IntelliSense。

您可以执行的一项自定义是配置应用程序将在其下运行的 IIS 应用程序池。以下示例显示了如何定义 IIS 应用程序池（“customPool”），该池每 60 分钟再循环一次流程，并使用 `"appPool": "customPool"` 将流程分配到应用程序。

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

此外，该清单还可以声明 Windows PowerShell 脚本在安装、重启和卸载操作之前和之后运行。例如，以下清单运行 Windows PowerShell 脚本 `PostInstallSetup.ps1` 以在 ASP.NET 内核应用程序部署到 IIS 之后完成进一步设置工作。在添加类似这样的脚本时，请确保将它们添加到 `project.json` 文件中的 publishOptions 下的 include 部分，正如对 `aws-windows-deployment-manifest.json` 文件的处理方式一样。如果没有这样做，这些脚本将不会作为 dotnet CLI publish 命令的一部分包含。

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

ebextensions 怎么样？

Elastic Beanstalk.ebextensions配置文件像所有其他 Elastic Beanstalk 容器一样受支持。要在 ASP.NET 内核应用程序中包含 ebextensions，请将 `.ebextensions` 目录添加到 `project.json` 文件中的 `publishOptions` 下的 `include` 部分。有关 ebextensions 的更多信息，请查阅 Elastic Beanstalk 开发人员指南。

对 .NET 和 Elastic Beanstalk 的多应用程序 Support

通过使用部署清单，您能够将多个应用程序部署到同一 Elastic Beanstalk 环境。

部署清单支持 ASP.NET 内核 Web 应用程序以及用于传统 ASP.NET 应用程序的 msdeploy 存档。设想以下情形：您为前端编写了一个使用 ASP.NET 内核的很棒的新应用程序并为扩展 API 编写了一个 Web API 项目。您还有一个使用传统 ASP.NET 编写的管理应用程序。

工具包的部署向导侧重于部署单一项目。要利用多应用程序部署，您必须手动构造应用程序包。要开始，请写入清单。在本示例中，您将在解决方案的根部写入清单。

清单中的部署部分包含两个子级：要部署的 ASP.NET 内核 Web 应用程序的数组以及要部署的 msdeploy 存档的数组。对于每个应用程序，您应设置应用程序的位相对于清单的 IIS 路径和位置。

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

在写入清单后，您将使用 Windows PowerShell 创建应用程序包并更新现有 Elastic Beanstalk 环境以运行此包。写入脚本时将假定它在包含您的 Visual Studio 解决方案的文件夹中运行。

您在脚本中需要执行的一个操作是设置在其中创建应用程序包的工作区文件夹。

```
$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
}
```

创建此文件夹后，是时候为前端做好准备了。与使用部署向导时一样，应使用 dotnet CLI 来发布应用程序。

```
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
```

请注意，子文件夹“frontend”用于输出文件夹 (与您在清单中设置的文件夹匹配)。现在您需要对 Web API 项目执行相同的操作。

```
Write-Host 'Publish the ASP.NET Core extensibility API'
$publishExtAPIFolder = [System.IO.Path]::Combine($publishWorkspace, "ext-api")
dotnet publish .\src\AmazingExtensibleAPI\project.json -o $publishExtAPIFolder -c
 Release -f netcoreapp1.0
```

管理员站点是传统 ASP.NET 应用程序，因此您无法使用 dotnet CLI。对于管理应用程序，您应使用在生成目标包中传递的 msbuild 来创建 msdeploy 存档。默认情况下，包目标将在 `obj\Release\Package` 文件夹下创建 msdeploy 存档，因此您需要将此存档部署到发布工作区。

```
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
```

要告知 Elastic Beanstalk 环境如何处理所有这些应用程序，请将此清单从您的解决方案复制到发布工作区，然后压缩文件夹。

```
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)
```

现在您已拥有应用程序包，您可转到 Web 控制台并将存档上传到 Elastic Beanstalk 环境。或者，您也可以继续使用AWSPowerShell cmdlet，通过应用程序包更新 Elastic Beanstalk 环境。确保您使用以下方式将当前配置文件和区域设置为包含您的 Elastic Beanstalk 环境的配置文件和区域。`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
```

现在，通过在工具包或 Web 控制台中使用 Elastic Beanstalk 环境状态页来检查更新状态。完成后，您将能够导航到部署于在部署清单中设置的 IIS 路径的所有应用程序。

部署到 Amazon EC2 Container Service

> **Important**
>
> 新Publish to (发布到 CloudWatch)AWS功能旨在简化将 .NET 应用程序发布到AWS. 选择后系统可能会问您是否希望切换到此发布体验。将容器发布到AWS. 有关更多信息，请参阅 使用 PublishAWS在 Visual Stu。

Amazon Elastic Container Service 是一种高度可扩展、高性能的容器管理服务，它支持 Docker 容器并可让您在 Amazon EC2 实例的托管群集上轻松运行应用程序。

要在 Amazon Elastic Container Service 上部署应用程序，必须开发应用程序组件以在 Docker 容器中运行。Docker 容器是一个软件开发的标准化单位，包含您的软件应用程序需要运行的一切：代码、运行时、系统工具、系统库等。

Toolkit for Visual Studio 提供了一个通过 Amazon ECS 简化应用程序发布的向导。此向导将在以下部分中介绍。

有关 Amazon ECS 的更多信息，请转至Elastic Container Service 文. 它概述了 Docker 的基础知识和集群创建。

主题

- 指定AWS适用于 ASP.NET 核心 2 应用程序的凭证
- 将 ASP.NET Core 2.0 应用程序部署到 Amazon ECS（Fargate）（传统）
- 将 ASP.NET 内核 2.0 应用程序部署到 Amazon ECS (EC2)

指定AWS适用于 ASP.NET 核心 2 应用程序的凭证

当您将应用程序部署到 Docker 容器时，有两种类型的凭证在发挥作用：部署凭证和实例凭证。

“发布容器” 使用部署凭证来AWS向导以在 Amazon ECS 中创建环境。这包括任务、服务、IAM 角色、Docker 容器存储库等，如果您选择的话，还包括负载均衡器。

实例（包括您的应用程序）使用实例凭证来访问不同的 AWS 服务。例如，如果您的 ASP.NET Core 2.0 应用程序读取和写入到 Amazon S3 对象，它需要适当的权限。您可以根据环境使用不同的方法提供不同凭证。例如，您的 ASP.NET 内核e 2 应用程序可能面向开发和生产环境。您可以使用本地 Docker 实例和凭证进行开发，而在生产中使用定义的角色。

指定部署凭证

这些区域有：AWS您在将容器发布到AWS向导是AWS向导将用于部署到 Amazon ECS 的账户。账户配置文件必须具有亚马逊弹性计算云、亚马逊弹性容器服务和AWS Identity and Access Management.

如果您注意到下拉列表中缺少某些选项，可能是因为您缺乏权限。例如，如果您为应用程序创建了一个集群但在将容器发布到AWS向导群集页面。如果出现这种情况，请添加所缺的权限并重试该向导。

指定开发实例凭证

对于非生产环境，您可以在 appsettings.<environment>.json 文件中配置凭证。例如，要在 Visual Studio 2017 的 appsettings.Development.json 文件中配置凭证，请执行以下操作：

1. 将 AWSSDK.Extensions.NETCore.Setup NuGet 软件包添加到您的项目中。
2. AddAWS将设置到 appsetings.Devin.json。以下配置将设置 `Profile` 和 `Region`。

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

指定生产实例凭证

对于生产实例，我们建议您使用 IAM 角色来控制应用程序 (和服务) 可以访问的内容。例如，要在中使用 Amazon ECS 将 IAM 角色配置为具有对 Amazon Simple Storage 服务和 Amazon DynamoDB 的权限的服务委托方，请执行以下操作：AWS Management Console：

1. 登录 AWS Management Console，然后通过以下网址打开 IAM 控制台：https://console.aws.amazon.com/iam/。
2. 在 IAM 控制台的导航窗格中，选择 Roles，然后选择 Create role。
3. 选择AWS服务选择角色类型，然后选择EC2 Container Service.
4. 选择 EC2 Container Service Task (EC2 Container Service 任务) 使用案例。使用案例由服务定义以包含服务要求的信任策略。接下来，选择 Next (下一步)：Permissions (下一步：权限)。
5. 选择 AmazonS3FullAccess 和 AmazonDynamoDBFullAccess 权限策略。选中每个策略旁边的复选框，然后选择后续：审核、
6. 对于Role name (角色名称)，键入有助于识别此角色的作用的角色名称或角色名称后缀。角色名称在您的 AWS 账户内必须是唯一的。名称不区分大小写。例如，您无法同时创建名为 PRODROLE 和 prodrole 的角色。由于多个单位可能引用该角色，角色创建完毕后无法编辑角色名称。
7. (可选) 对于 Role description，键入新角色的描述。
8. 检查角色，然后选择 Create role。

你可以使用此角色作为任务角色在ECS 任务定义的页面将容器发布到AWS向导。

有关更多信息，请参阅使用基于服务的角色。

将 ASP.NET Core 2.0 应用程序部署到 Amazon ECS（Fargate）（传统）

> **Important**
>
> 本文档涉及旧版服务和功能。有关更新的指南和内容，请参阅 AWS.NET 部署工具指南和更新后的 “部署到AWS目录”。

本节介绍如何使用作为 VisuToolkit for Visual Studio 一部分提供的 “发布容器到AWS” 向导，使用 Fargate 启动类型通过 Amazon ECS 部署针对 Linux 的容器化 ASP.NET Core 2.0 应用程序。由于 Web 应用程序要持续运行，因此将作为一项服务部署。

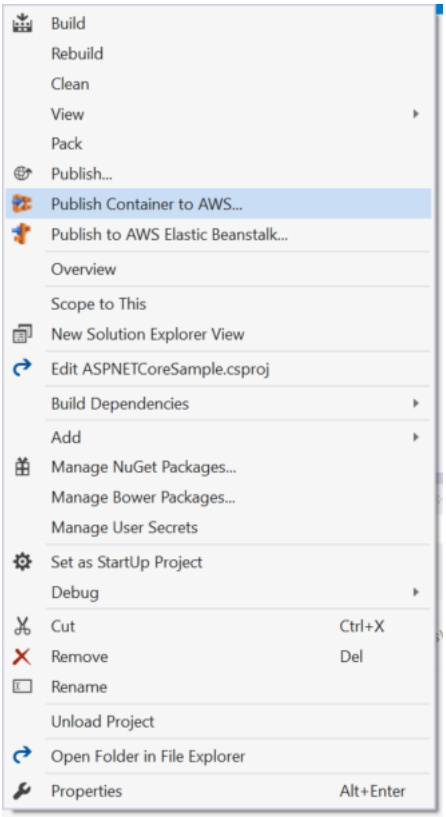
在您发布容器之前

在使用 “发布容器到AWS” 向导部署 ASP.NET Core 2.0 应用程序之前：

- 指定您的AWS凭证并使用 Amazon ECS 进行设置。
- 安装 Docker。有几个不同的安装选项，包括 Docker for Windows。
- 在 Visual Studio 中，为针对 Linux 的 ASP.NET Core 2.0 容器化应用程序创建（或打开）一个项目。

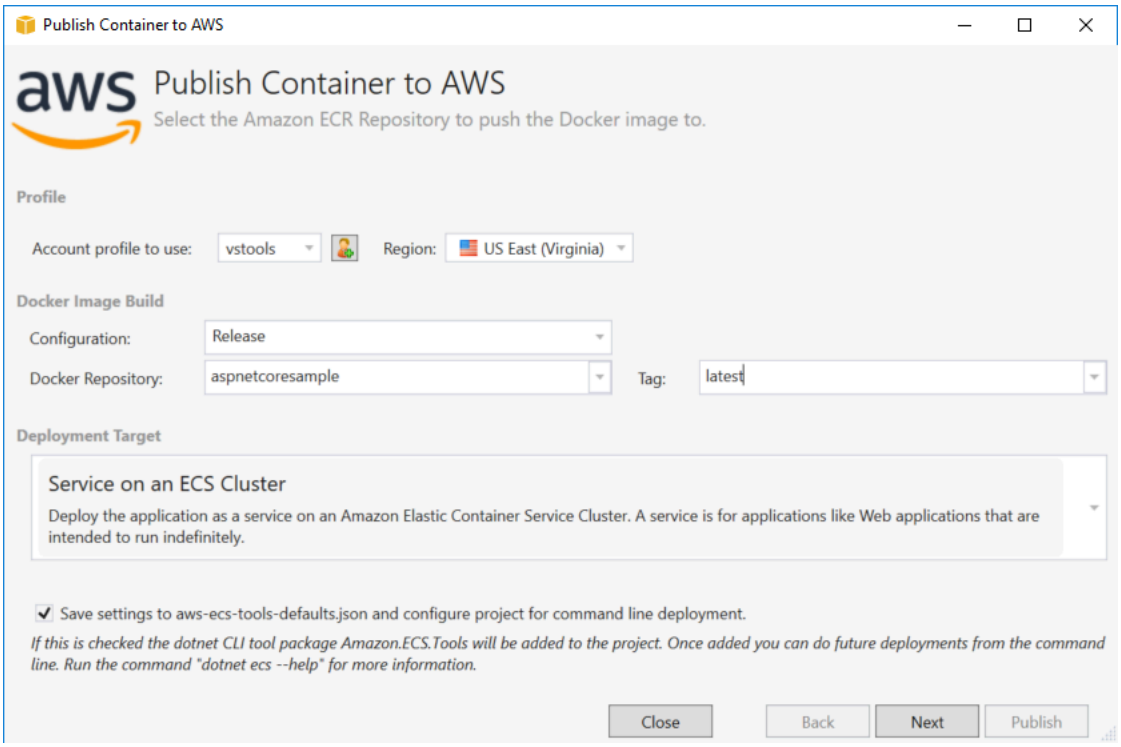
访问 “将容器发布到AWS” 向导

要部署针对 Linux 的 ASP.NET Core 2.0 容器化应用程序，请在解决方案资源管理器中右键单击该项目，然后选择 “将容器发布到”AWS。



也可以在 Visual Studio 的 “构建” 菜单AWS上选择 “将容器发布到”。

将容器发布到AWS向导



Account profile to use (要使用的账户配置文件) - 选择要使用的账户配置文件。

Region (区域) - 选择部署区域。配置文件和区域用于设置您的部署环境资源并选择默认的 Docker 注册表。

Configuration (配置) - 选择 Docker 映像构建配置。

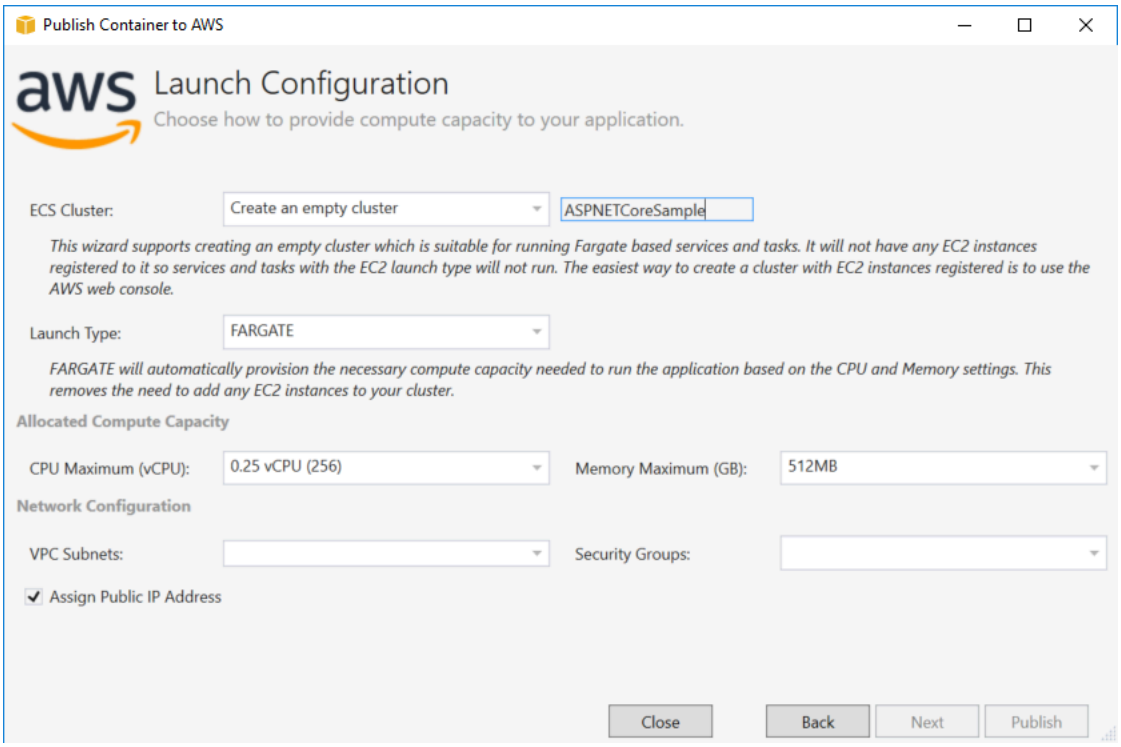
Docker Repository (Docker 存储库) - 选择现有 Docker 存储库，或键入新存储库的名称即可创建新存储库。这是构建容器要推送到的存储库。

Tag (标签) - 选择现有标签或键入新标签的名称。标签可以跟踪重要详细信息，如 Docker 容器的版本、选项或其他唯一配置元素。

Deployment Target (部署目标) – 选择 Service on an ECS Cluster (ECS 集群上的服务)。当您的应用程序（如 ASP.NET Web 应用程序）计划长时间运行时，请使用此部署选项。

将设置保存到 **`aws-docker-tools-defaults.json`** 并为命令行部署配置项目 - 如果您需要从命令行灵活部署，请选中此选项。使用您的项目目录中的 `dotnet ecs deploy` 以部署和 `dotnet ecs publish` 容器。

“Launch Configuration (启动配置)”页面



ECS Cluster (ECS 集群) – 选择将运行 Docker 映像的集群。如果您选择创建空集群，请为您的新集群命名。

Launch Type (启动类型) - 选择 FARGATE。

CPU Maximum (vCPU) (CPU 最大容量(vCPU)) - 选择您的应用程序所需的最大计算容量。要查看 CPU 和内存值的允许范围，请参阅任务大小。

Memory Maximum (GB) (CPU 最大容量(GB)) – 选择您的应用程序可用的最大内存容量。

VPC Subnets (VPC 子网) – 选择单个 VPC 中的一个或多个子网。如果您选择多个子网，则您的任务将分配到这几个子网中。这可以提高可用性。有关更多信息，请参阅默认 VPC 和默认子网。

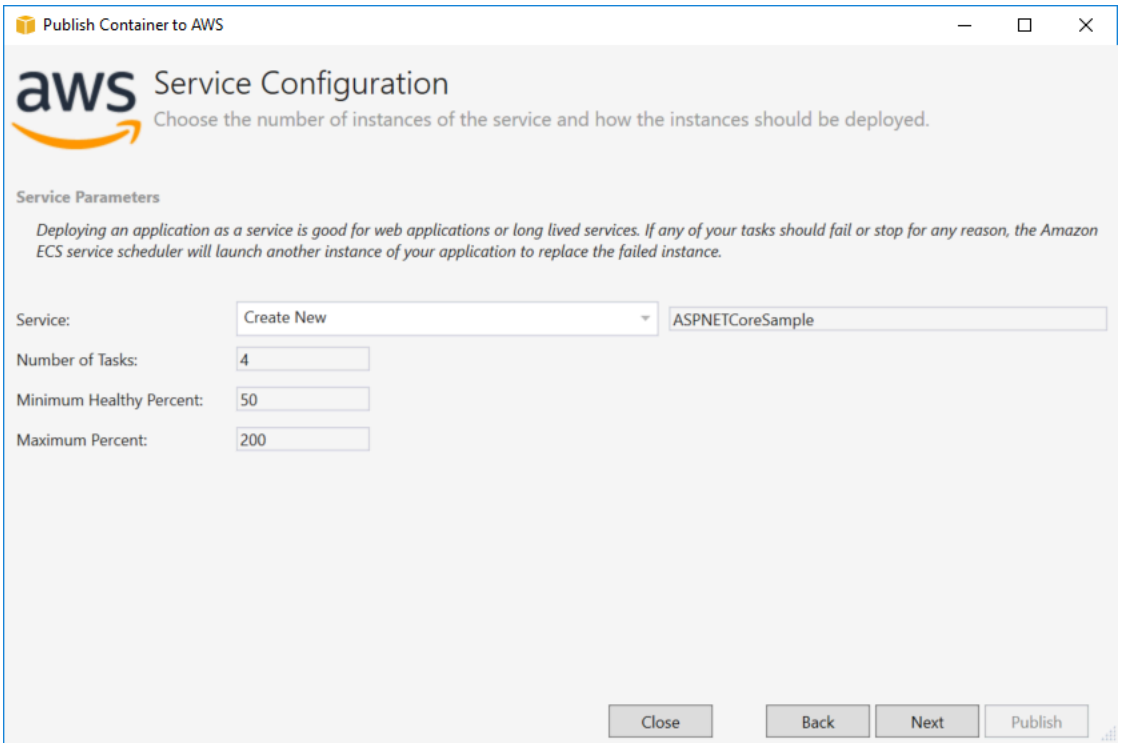
Security Groups (安全组) - 选择一个安全组。

安全组可作为关联 Amazon EC2 实例的防火墙，在实例级别控制入站和出站流量。

默认安全组配置为允许来自分配给同一安全组的多个实例的入站流量和所有出站 IPv4 流量。您需要允许出站，以便服务可以访问容器存储库。

Assign Public IP Address (分配公有 IP 地址) – 选中此复选框以便从 Internet 访问任务。

“Service Configuration (服务配置)”页面



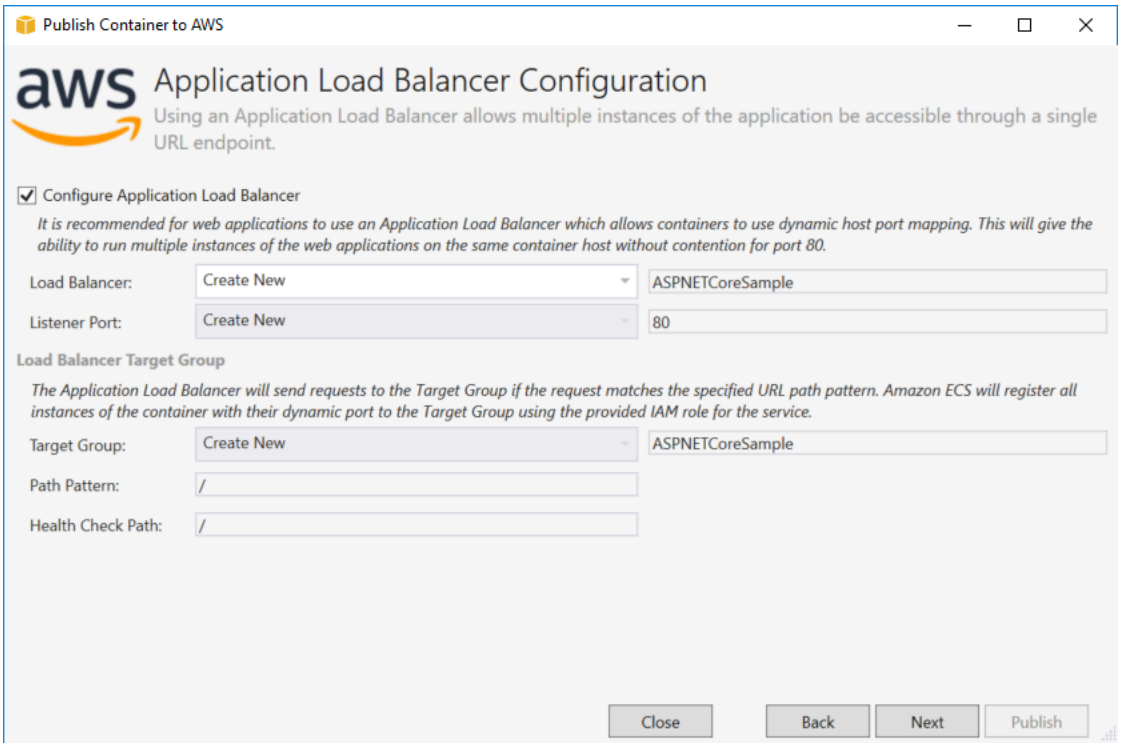
Service (服务) - 从下拉框中选择一项服务，将您的容器部署到该现有服务。或者选择 Create New (新建) 新建一项服务。一个集群中的服务名称必须唯一，但是您可以为一个区域或多个区域中多个集群中的服务提供相似的名称。

Number of Tasks (任务数) - 要在您的集群中部署并保持运行的任务数量。每个任务都是您的容器的一个实例。

Minimum Healthy Percent (最小正常运行状况百分比) - 在部署期间必须处于 RUNNING 状态的任务百分比（四舍五入到最近的整数）。

Maximum Percent (最大百分比) - 在部署期间允许处于 RUNNING 或 PENDING 状态的任务百分比（向下舍入到最近的整数）。

“Application Load Balancer (应用程序负载均衡器)”页面



Configure Application Load Balancer (配置应用程序负载均衡器) - 选中此项可配置应用程序负载均衡器。

Load Balancer (负载均衡器) - 选择一个现有负载均衡器，或者选择 Create New (新建) 并键入新负载均衡器的名称。

Listener Port (侦听器端口) - 选择一个现有侦听器端口，或者选择 Create New (新建) 并键入一个端口号。默认端口 80 适用于大多数 Web 应用程序。

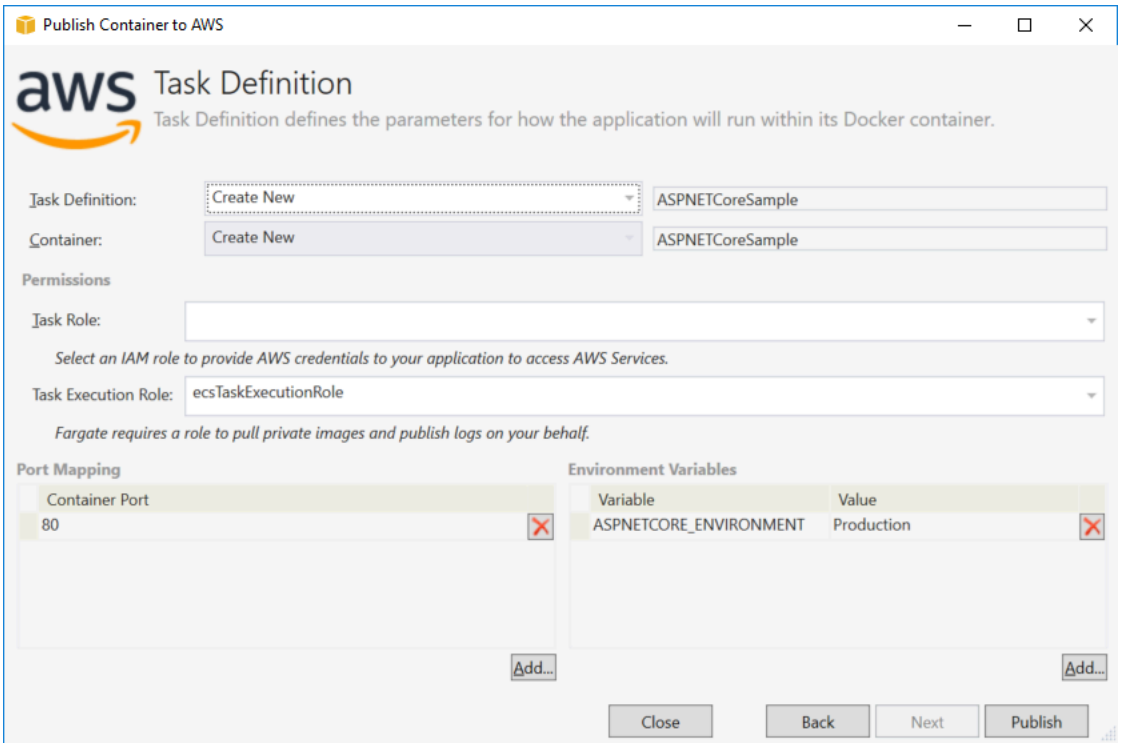
目标组-选择 Amazon ECS 将任务注册到的目标组。

Path Pattern (路径模式) - 负载均衡器将使用基于路径的路由。接受默认 / 或提供一个不同模式。路径模式区分大小写，长度最多为 128 个字符，并且可包含一组选定字符。

Health Check Path (运行状况检查路径) - 进行运行状况检查的目标上的目的地的 Ping 路径。默认为 /。输入不同的路径（如果需要）。如果您输入的路径无效，则运行状况检查将失败，并将视为运行状况不佳。

如果您要部署多个服务，且每个服务都将部署到不同的路径或位置，您需要自定义检查路径。

“Task Definition (任务定义)”页面



Task Definition (任务定义) - 选择一个现有任务定义，或者选择 Create New (新建) 并键入新任务定义的名称。

Container (容器) - 选择一个现有容器，或者选择 Create New (新建) 并键入新容器的名称。

任务角色-选择一个 IAM 角色，该角色拥有您的应用程序访问AWS服务所需的证书。凭证就是通过这种方法传递给您的应用程序的。查看如何为您的应用程序指定AWS安全证书。

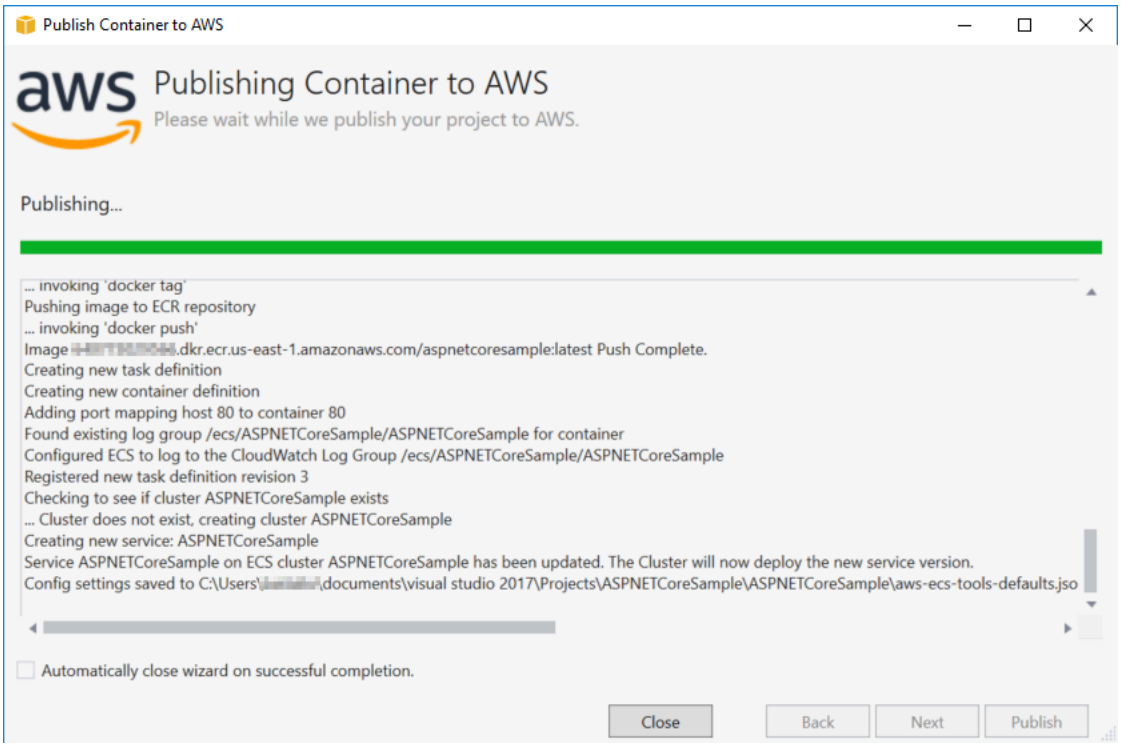
任务执行角色-选择具有提取私有映像和发布日志权限的角色。 AWSFargate 会代表您使用这一事宜。

Port Mapping (端口映射) - 选择绑定到自动分配的主机端口的容器上的端口号。

Environment Variables (环境变量) - 添加、修改或删除容器的环境变量。您可以进行修改以满足部署要求。

如果您对配置满意，请单击 Publish (发布) 开始部署流程。

将容器发布到AWS



在部署过程中会显示事件。成功完成后向导会自动关闭。您可以通过取消选中页面底部的复选框来覆盖该功能。

您可以在AWS资源管理器中找到新实例的 URL。展开 Amazon ECS 和集群，然后单击您的集群。

将 ASP.NET 内核 2.0 应用程序部署到 Amazon ECS (EC2)

本节说明如何将发布容器到AWS向导作为 Toolkit for Visual Studio 的一部分提供，此向导可使用 EC2 启动类型通过 Amazon ECS 部署针对 Linux 的容器化 ASP.NET 内核 2.0 应用程序。由于 Web 应用程序要持续运行，因此将作为一项服务部署。

在您发布容器之前

在使用发布容器到AWS要部署 ASP.NET 内核 2.0 应用程序，请执行以

- 指定您的AWS证书和借助 Amazon ECS 获取设置.
- 安装 Docker。有几个不同的安装选项，包括 Docker for Windows。
- 根据 Web 应用程序的需求创建 Amazon ECS 集群。只需几个步骤即可完成。
- 在 Visual Studio 中，创建（或打开）一个针对 Linux 的 ASP.NET 内核 2.0 容器化应用程序的项目。

访问 Publish Container inAWS巫师

要部署针对 Linux 的 ASP.NET 内核 2.0 容器化应用程序，请右键单击解决方案资源管理器中的项目并选择发布容器到AWS.

您还可以选择发布容器到AWS在 Visual Studio Build 菜单中。

发布容器到AWS向导

Account profile to use (要使用的账户配置文件) - 选择要使用的账户配置文件。

Region (区域) - 选择一个部署区域。配置文件和区域用于设置您的部署环境资源并选择默认的 Docker 注册表。

Configuration (配置) - 选择 Docker 映像构建配置。

Docker Repository (Docker 存储库) - 选择现有 Docker 存储库，或键入新存储库的名称即可创建新存储库。这是构建的容器映像要推送到的存储库。

Tag (标签) - 选择现有标签或键入新标签的名称。标签可以跟踪重要详细信息，如 Docker 容器的版本、选项或其他唯一配置元素。

Deployment (部署) – 选择 Service on an ECS Cluster (ECS 集群上的服务)。当您的应用程序（如 ASP.NET 内核 2.0 Web 应用程序）计划长时间运行时，请使用此部署选项。

将设置保存到 **`aws-docker-tools-defaults.json`** 并为命令行部署配置项目 - 如果您需要从命令行灵活部署，请选中此选项。使用您的项目目录中的 `dotnet ecs deploy` 以部署和 `dotnet ecs publish` 容器。

“Launch Configuration (启动配置)”页面

ECS Cluster (ECS 集群) – 选择将运行 Docker 映像的集群。您可以创建 ECS 集群使用AWS管理控制台。

Launch Type (启动类型) - 选择 EC2。要使用 Fargate 启动类型，请参阅将 ASP.NET 内核 2.0 应用程序部署到 Amazon ECS (Fargate)。

“Service Configuration (服务配置)”页面

Service (服务) - 从下拉框中选择一项服务，将您的容器部署到该现有服务。或者选择 Create New (新建) 新建一项服务。一个集群中的服务名称必须唯一，但是您可以为一个区域或多个区域中多个集群中的服务提供相似的名称。

Number of Tasks (任务数) - 要在您的集群中部署并保持运行的任务数量。每个任务都是您的容器的一个实例。

Minimum Healthy Percent (最小正常运行状况百分比) - 在部署期间必须处于 RUNNING 状态的任务百分比（四舍五入到最近的整数）。

Maximum Percent (最大百分比) - 在部署期间允许处于 RUNNING 或 PENDING 状态的任务百分比（向下舍入到最近的整数）。

Placement Templates (放置模板) – 选择任务放置模板。

如果您在集群中启动任务，Amazon ECS 必须根据任务定义中指定的要求（例如 CPU 和内存）确定将任务放置在何处。同样，如果您缩减任务计数，Amazon ECS 必须确定终止哪些任务。

放置模板用于控制任务如何在集群中启动：

- AZ Balanced Spread (AZ 均衡分散) - 在各个可用区以及每个可用区中的各个容器实例中分配任务。
- AZ Balanced BinPack (AZ 均衡装填) - 在各个可用区以及具有最低可用内存的容器实例中分配任务。
- BinPack (装填) - 根据 CPU 或内存的最低可用量来分配任务。
- One Task Per Host (每个主机一项任务) - 在每个容器实例中最多可放置服务的一个任务。

有关更多信息，请参阅 Amazon ECS 任务放置。

“Application Load Balancer (应用程序负载均衡器)”页面

Configure Application Load Balancer (配置应用程序负载均衡器) - 选中此项可配置应用程序负载均衡器。

Select IAM role for service (为服务选择 IAM 角色) – 选择一个现有角色，或者选择 Create New (新建) 即可创建一个新角色。

Load Balancer (负载均衡器) - 选择一个现有负载均衡器，或者选择 Create New (新建) 并键入新负载均衡器的名称。

Listener Port (侦听器端口) - 选择一个现有侦听器端口，或者选择 Create New (新建) 并键入一个端口号。默认端口 80 适用于大多数 Web 应用程序。

Target Group (目标组) - 默认情况下，负载均衡器使用您为目标组指定的端口和协议将请求发送到已注册目标。在将每个目标注册到目标组时，可以覆盖此端口。

Path Pattern (路径模式) - 负载均衡器将使用基于路径的路由。接受默认 / 或提供一个不同模式。路径模式区分大小写，长度最多为 128 个字符，并且可包含一组选定字符。

Health Check Path (运行状况检查路径) - 进行运行状况检查的目标上的目的地的 Ping 路径。默认为 /，该设置适用于 Web 应用程序。输入不同的路径（如果需要）。如果您输入的路径无效，则运行状况检查将失败，并将视为运行状况不佳。

如果您要部署多个服务，且每个服务都将部署到不同的路径或位置，您可能需要自定义检查路径。

“ECS Task Definition (ECS 任务定义)”页面

Task Definition (任务定义) - 选择一个现有任务定义，或者选择 Create New (新建) 并键入新任务定义的名称。

Container (容器) - 选择一个现有容器，或者选择 Create New (新建) 并键入新容器的名称。

Memory (MiB) (内存(MiB)) - 提供软限制和/或硬限制的值。

要为容器预留的内存量的软限制（以 MiB 为单位）。Docker 尝试将容器内存控制在软限制以下。如果容器需要消耗更多内存，则上限为内存参数指定的硬限制（如果适用）或者容器实例中的全部可用内存，以较低者为准。

要提供给容器的内存的硬限制（以 MiB 为单位）。如果您的容器尝试使用超出此处指定的内存，该容器将被终止。

任务角色-为 IAM 角色选择一个任务角色，此角色可使容器有权调用AWS在关联策略中代表您指定的API。凭证就是通过这种方法传递给您的应用程序的。请参阅如何指定AWS应用程序的安全凭证.

Port Mapping (端口映射) – 添加、修改或删除容器的端口映射。如果负载均衡器为开启状态，主机端口将默认设置为 0，端口分配采用动态方式。

Environment Variables (环境变量) - 添加、修改或删除容器的环境变量。

如果您对配置满意，请单击 Publish (发布) 开始部署流程。

发布容器到AWS

在部署过程中会显示事件。成功完成后向导会自动关闭。您可以通过取消选中页面底部的复选框来覆盖该功能。

您可以在AWSExplorer。展开 Amazon ECS 和集群，然后单击您的集群。

疑难解答 AWS Toolkit for Visual Studio

以下各节包含有关 AWS Toolkit for Visual Studio 和使用该工具包中的 AWS 服务的一般疑难解答信息。

> **Note**
>
> 安装和 set-up-specific 疑难解答信息可在本用户指南中的安装问题疑难解答主题中找到。

主题

问题排查最佳实践

以下是解决 AWS Toolkit for Visual Studio 问题时推荐的最佳做法。

- 在发送报告之前，尝试重现问题或错误。
- 详细记录重现过程中的每个步骤、设置和错误消息。
- 收集 AWS 工具包日志。有关如何查找 AWS Toolkit 日志的详细说明，请参阅本指南主题中的 “如何找到您的 AWS 日志” 过程。
- 查看未解决的请求、已知的解决方案，或者在 AWS Toolkit for Visual Studio GitHub 存储库的 “AWS Toolkit for Visual Studio 问题” 部分报告未解决的问题。

如何找到你的 AWS Toolkit 日志

1. 在 Visual Studio 主菜单中，展开扩展。
2. 选择AWS 工具包以展开 Tool AWS kit 菜单，然后选择查看 Toolkit 日志。
3. 当 AWS Toolkit 日志文件夹在您的操作系统中打开时，按日期对文件进行排序，然后找到任何包含与当前问题相关的信息的日志文件。

Amazon CodeWhisperer 登录和注销已禁用

如果您在使用该 CodeWhisperer 服务时遇到登录和注销菜单项均被禁用的问题，请通过完成以下步骤来解决问题。

1. 从 Windows 文件资源管理器中，导航到位于:的 AWS 工具包缓存文件夹`%LOCALAPPDATA%/aws/toolkits/language-servers/CodeWhisperer`。
2. 清除缓存文件夹中的内容。
3. 关闭并重新打开当前解决方案。

的安全性 AWS Toolkit for Visual Studio

云安全性一直是 Amazon Web Services (AWS) 的重中之重。作为 AWS 客户，您将从专为满足大多数安全敏感型企业的要求而打造的数据中心和网络架构中受益。安全是双方共同承担 AWS 的责任。责任共担模式将其描述为云的安全性和云中的安全性。

云安全 — AWS 负责保护运行 AWS 云中提供的所有服务的基础架构，并为您提供可以安全使用的服务。我们的安全责任是重中之重 AWS，作为AWS 合规计划的一部分，第三方审计师定期测试和验证我们安全的有效性。

云端安全 — 您的责任由您使用的 AWS 服务以及其他因素决定，包括数据的敏感性、组织的要求以及适用的法律和法规。

本 AWS 产品或服务通过其支持的特定 Amazon Web Services (AWS) 服务遵循分担责任模式。有关 AWS 服务安全信息，请参阅AWS 服务安全文档页面和合规计划合 AWS 规工作范围内的AWS 服务。

主题

- 中的数据保护 AWS Toolkit for Visual Studio
- Identity and Access Management
- 此 AWS 产品或服务的合规性验证
- 本 AWS 产品或服务的弹性
- 本 AWS 产品或服务的基础设施安全
- 中的配置和漏洞分析 AWS Toolkit for Visual Studio

中的数据保护 AWS Toolkit for Visual Studio

分担责任模型 AWS 分担责任模型适用于 Visual Studio AWS Toolkit for Visual Studio 中的数据保护。如本模型所述 AWS ，负责保护运行所有内容的全球基础架构 AWS Cloud。您负责维护对托管在此基础设施上的内容的控制。您还负责您所使用的 AWS 服务 的安全配置和管理任务。有关数据隐私的更多信息，请参阅数据隐私常见问题。有关欧洲数据保护的信息，请参阅 AWS 安全性博客 上的 AWS 责任共担模式和 GDPR 博客文章。

出于数据保护目的，我们建议您保护 AWS 账户 凭证并使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 设置个人用户。这样，每个用户只获得履行其工作职责所需的权限。我们还建议您通过以下方式保护数据：

- 对每个账户使用多重身份验证 (MFA)。

- 使用 SSL/TLS 与资源通信。 AWS 我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 使用设置 API 和用户活动日志 AWS CloudTrail。
- 使用 AWS 加密解决方案以及其中的所有默认安全控件 AWS 服务。
- 使用高级托管安全服务（例如 Amazon Macie），它有助于发现和保护存储在 Amazon S3 中的敏感数据。
- 如果您在 AWS 通过命令行界面或 API 进行访问时需要经过 FIPS 140-2 验证的加密模块，请使用 FIPS 端点。有关可用的 FIPS 端点的更多信息，请参阅《美国联邦信息处理标准（FIPS）第 140-2 版》。

我们强烈建议您切勿将机密信息或敏感信息（如您客户的电子邮件地址）放入标签或自由格式文本字段（如名称字段）。这包括使用控制台、API 或 SDK AWS 服务 使用 Toolkit for Visual Stu AWS dio 或其他工具包时。 AWS CLI在用于名称的标签或自由格式文本字段中输入的任何数据都可能会用于计费或诊断日志。如果您向外部服务器提供网址，强烈建议您不要在网址中包含凭证信息来验证对该服务器的请求。

Identity and Access Management

AWS Identity and Access Management (IAM) AWS 服务 可帮助管理员安全地控制对 AWS 资源的访问权限。IAM 管理员控制谁可以进行身份验证（登录）和授权（拥有权限）使用 AWS 资源。您可以使用 IAM AWS 服务 ，无需支付额外费用。

主题

- 受众
- 使用身份进行身份验证
- 使用策略管理访问
- 如何 AWS 服务 使用 IAM
- 对 AWS 身份和访问进行故障排除

受众

您的使用方式 AWS Identity and Access Management (IAM) 会有所不同，具体取决于您所做的工作 AWS。

服务用户-如果您 AWS 服务 曾经完成工作，则您的管理员会为您提供所需的凭证和权限。当你使用更多 AWS 功能来完成工作时，你可能需要额外的权限。了解如何管理访问权限有助于您向管理员请求适

合的权限。如果您无法访问中的功能 AWS，请参阅对 AWS 身份和访问进行故障排除或 AWS 服务 您正在使用的用户指南。

服务管理员-如果您负责公司的 AWS 资源，则可能拥有完全访问权限 AWS。您的工作是确定您的服务用户应访问哪些 AWS 功能和资源。然后，您必须向 IAM 管理员提交请求以更改服务用户的权限。请查看该页面上的信息以了解 IAM 的基本概念。要详细了解您的公司如何使用 IAM AWS，请参阅 AWS 服务 您正在使用的用户指南。

IAM 管理员：如果您是 IAM 管理员，您可能希望了解如何编写策略以管理对 AWS的访问权限的详细信息。要查看您可以在 IAM 中使用的 AWS 基于身份的策略示例，请参阅 AWS 服务 您正在使用的用户指南。

使用身份进行身份验证

身份验证是您 AWS 使用身份凭证登录的方式。您必须以 IAM 用户身份或通过担 AWS 账户根用户任 IAM 角色进行身份验证（登录 AWS）。

您可以使用通过身份源提供的凭据以 AWS 联合身份登录。 AWS IAM Identity Center （IAM Identity Center）用户、贵公司的单点登录身份验证以及您的 Google 或 Facebook 凭据就是联合身份的示例。当您以联合身份登录时，您的管理员以前使用 IAM 角色设置了身份联合验证。当你使用联合访问 AWS 时，你就是在间接扮演一个角色。

根据您的用户类型，您可以登录 AWS Management Console 或 AWS 访问门户。有关登录的更多信息 AWS，请参阅《AWS 登录 用户指南》中的如何登录到您 AWS 账户的。

如果您 AWS 以编程方式访问，则会 AWS 提供软件开发套件 (SDK) 和命令行接口 (CLI)，以便使用您的凭据对请求进行加密签名。如果您不使用 AWS 工具，则必须自己签署请求。有关使用推荐的方法自行签署请求的更多信息，请参阅 IAM 用户指南中的签署 AWS API 请求。

无论使用何种身份验证方法，您可能需要提供其他安全信息。例如， AWS 建议您使用多重身份验证 (MFA) 来提高账户的安全性。要了解更多信息，请参阅《AWS IAM Identity Center 用户指南》中的多重身份验证和《IAM 用户指南》中的在 AWS中使用多重身份验证（MFA）。

AWS 账户 root 用户

创建时 AWS 账户，首先要有一个登录身份，该身份可以完全访问账户中的所有资源 AWS 服务 和资源。此身份被称为 AWS 账户 root 用户，使用您创建账户时使用的电子邮件地址和密码登录即可访问该身份。强烈建议您不要使用根用户执行日常任务。保护好根用户凭证，并使用这些凭证来执行仅根用户可以执行的任务。有关要求您以根用户身份登录的任务的完整列表，请参阅《IAM 用户指南》中的需要根用户凭证的任务。

联合身份

作为最佳实践，要求人类用户（包括需要管理员访问权限的用户）使用与身份提供商的联合身份验证 AWS 服务 通过临时证书进行访问。

联合身份是指您的企业用户目录、Web 身份提供商、Identity C enter 目录中的用户，或者任何使用 AWS 服务 通过身份源提供的凭据进行访问的用户。 AWS Directory Service当联合身份访问时 AWS 账户，他们将扮演角色，角色提供临时证书。

要集中管理访问权限，建议您使用 AWS IAM Identity Center。您可以在 IAM Identity Center 中创建用户和群组，也可以连接并同步到您自己的身份源中的一组用户和群组，以便在您的所有 AWS 账户 和应用程序中使用。有关 IAM Identity Center 的信息，请参阅《AWS IAM Identity Center 用户指南》中的什么是 IAM Identity Center？。

IAM 用户和群组

I AM 用户是您 AWS 账户 内部对个人或应用程序具有特定权限的身份。在可能的情况下，我们建议使用临时凭证，而不是创建具有长期凭证（如密码和访问密钥）的 IAM 用户。但是，如果您有一些特定的使用场景需要长期凭证以及 IAM 用户，建议您轮换访问密钥。有关更多信息，请参阅《IAM 用户指南》中的对于需要长期凭证的使用场景定期轮换访问密钥。

IAM 组是一个指定一组 IAM 用户的身份。您不能使用组的身份登录。您可以使用组来一次性为多个用户指定权限。如果有大量用户，使用组可以更轻松地管理用户权限。例如，您可能具有一个名为 IAMAdmins 的组，并为该组授予权限以管理 IAM 资源。

用户与角色不同。用户唯一地与某个人员或应用程序关联，而角色旨在让需要它的任何人代入。用户具有永久的长期凭证，而角色提供临时凭证。要了解更多信息，请参阅《IAM 用户指南》中的何时创建 IAM 用户（而不是角色）。

IAM 角色

I AM 角色是您内部具有特定权限 AWS 账户 的身份。它类似于 IAM 用户，但与特定人员不关联。您可以 AWS Management Console 通过切换角色在中临时担任 IAM 角色。您可以通过调用 AWS CLI 或 AWS API 操作或使用自定义 URL 来代入角色。有关使用角色的方法的更多信息，请参阅《IAM 用户指南》中的 使用 IAM 角色。

具有临时凭证的 IAM 角色在以下情况下很有用：

- 联合用户访问 – 要向联合身份分配权限，请创建角色并为角色定义权限。当联合身份进行身份验证时，该身份将与角色相关联并被授予由此角色定义的权限。有关联合身份验证的角色的信息，请参阅《IAM 用户指南》中的 为第三方身份提供商创建角色。如果您使用 IAM Identity Center，则需要配

置权限集。为控制您的身份在进行身份验证后可以访问的内容，IAM Identity Center 将权限集与 IAM 中的角色相关联。有关权限集的信息，请参阅《AWS IAM Identity Center 用户指南》中的权限集。

- 临时 IAM 用户权限 – IAM 用户可代入 IAM 用户或角色，以暂时获得针对特定任务的不同权限。
- 跨账户存取 – 您可以使用 IAM 角色以允许不同账户中的某个人（可信主体）访问您的账户中的资源。角色是授予跨账户访问权限的主要方式。但是，对于某些资源 AWS 服务，您可以将策略直接附加到资源（而不是使用角色作为代理）。要了解用于跨账户访问的角色和基于资源的策略之间的差别，请参阅《IAM 用户指南》中的 IAM 角色与基于资源的策略有何不同。
- 跨服务访问 — 有些 AWS 服务 使用其他 AWS 服务服务中的功能。例如，当您在某个服务中进行调用时，该服务通常会在 Amazon EC2 中运行应用程序或在 Amazon S3 中存储对象。服务可能会使用发出调用的主体的权限、使用服务角色或使用服务相关角色来执行此操作。
  - 转发访问会话 (FAS) — 当您使用 IAM 用户或角色在中执行操作时 AWS，您被视为委托人。使用某些服务时，您可能会执行一个操作，然后此操作在其他服务中启动另一个操作。FAS 使用调用委托人的权限以及 AWS 服务 向下游服务发出请求的请求。 AWS 服务只有当服务收到需要与其他 AWS 服务 或资源交互才能完成的请求时，才会发出 FAS 请求。在这种情况下，您必须具有执行这两个操作的权限。有关发出 FAS 请求时的策略详情，请参阅转发访问会话。
  - 服务角色 - 服务角色是服务代表您在您的账户中执行操作而分派的 IAM 角色。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的创建向 AWS 服务委派权限的角色。
  - 服务相关角色-服务相关角色是一种链接到的服务角色。 AWS 服务服务可以代入代表您执行操作的角色。服务相关角色出现在您的中 AWS 账户 ，并且归服务所有。IAM 管理员可以查看但不能编辑服务相关角色的权限。
- 在 A@@ mazon EC2 上运行的应用程序 — 您可以使用 IAM 角色管理在 EC2 实例上运行并发出 AWS CLI 或 AWS API 请求的应用程序的临时证书。这优先于在 EC2 实例中存储访问密钥。要向 EC2 实例分配 AWS 角色并使其可供其所有应用程序使用，您需要创建附加到该实例的实例配置文件。实例配置文件包含角色，并使 EC2 实例上运行的程序能够获得临时凭证。有关更多信息，请参阅《IAM 用户指南》中的 使用 IAM 角色为 Amazon EC2 实例上运行的应用程序授予权限。

要了解是使用 IAM 角色还是 IAM 用户，请参阅《IAM 用户指南》中的何时创建 IAM 角色（而不是用户）。

使用策略管理访问

您可以 AWS 通过创建策略并将其附加到 AWS 身份或资源来控制中的访问权限。策略是其中的一个对象 AWS ，当与身份或资源关联时，它会定义其权限。 AWS 在委托人（用户、root 用户或角色会话）发出请求时评估这些策略。策略中的权限确定是允许还是拒绝请求。大多数策略都以 JSON 文档

的 AWS 形式存储在中。有关 JSON 策略文档的结构和内容的更多信息，请参阅《IAM 用户指南》中的 JSON 策略概览。

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体 可以对什么资源执行操作，以及在什么条件下执行。

默认情况下，用户和角色没有权限。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM 策略。管理员随后可以向角色添加 IAM 策略，用户可以代入角色。

IAM 策略定义操作的权限，无关乎您使用哪种方法执行操作。例如，假设您有一个允许 `iam:GetRole` 操作的策略。拥有该策略的用户可以从 AWS Management Console AWS CLI、或 AWS API 获取角色信息。

基于身份的策略

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的创建 IAM 策略。

基于身份的策略可以进一步归类为内联策略或托管式策略。内联策略直接嵌入单个用户、组或角色中。托管策略是独立的策略，您可以将其附加到中的多个用户、群组和角色 AWS 账户。托管策略包括 AWS 托管策略和客户托管策略。要了解如何在托管式策略和内联策略之间进行选择，请参阅《IAM 用户指南》中的在托管式策略与内联策略之间进行选择。

基于资源的策略

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Simple Storage Service（Amazon S3）存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中指定主体。委托人可以包括账户、用户、角色、联合用户或 AWS 服务。

基于资源的策略是位于该服务中的内联策略。您不能在基于资源的策略中使用 IAM 中的 AWS 托管策略。

访问控制列表 (ACL)

访问控制列表（ACL）控制哪些主体（账户成员、用户或角色）有权访问资源。ACL 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

Amazon S3 和 Amazon VPC 就是支持 ACL 的服务示例。 AWS WAF要了解有关 ACL 的更多信息，请参阅《Amazon Simple Storage Service 开发人员指南》 中的访问控制列表（ACL）概览。

其他策略类型

AWS 支持其他不太常见的策略类型。这些策略类型可以设置更常用的策略类型向您授予的最大权限。

- 权限边界 - 权限边界是一个高级功能，用于设置基于身份的策略可以为 IAM 实体（IAM 用户或角色）授予的最大权限。您可为实体设置权限边界。这些结果权限是实体基于身份的策略及其权限边界的交集。在 `Principal` 中指定用户或角色的基于资源的策略不受权限边界限制。任一项策略中的显式拒绝将覆盖允许。有关权限边界的更多信息，请参阅《IAM 用户指南》中的 IAM 实体的权限边界。
- 服务控制策略 (SCP)-SCP 是 JSON 策略，用于指定组织或组织单位 (OU) 的最大权限。 AWS Organizations AWS Organizations 是一项用于对您的企业拥有的多 AWS 账户 项进行分组和集中管理的服务。如果在组织内启用了所有功能，则可对任意或全部账户应用服务控制策略 (SCP)。SCP 限制成员账户中的实体（包括每个 AWS 账户根用户实体）的权限。有关 Organizations 和 SCP 的更多信息，请参阅《AWS Organizations 用户指南》中的 SCP 的工作原理。
- 会话策略 – 会话策略是当您以编程方式为角色或联合用户创建临时会话时作为参数传递的高级策略。结果会话的权限是用户或角色的基于身份的策略和会话策略的交集。权限也可以来自基于资源的策略。任一项策略中的显式拒绝将覆盖允许。有关更多信息，请参阅《IAM 用户指南》中的会话策略。

多个策略类型

当多个类型的策略应用于一个请求时，生成的权限更加复杂和难以理解。要了解在涉及多种策略类型时如何 AWS 确定是否允许请求，请参阅 IAM 用户指南中的策略评估逻辑。

如何 AWS 服务 使用 IAM

要全面了解如何 AWS 服务 使用大多数 IAM 功能，请参阅 IAM 用户指南中的与 IAM 配合使用的AWS 服务。

要了解如何在 IAM 中 AWS 服务 使用特定的，请参阅相关服务的用户指南的安全部分。

对 AWS 身份和访问进行故障排除

使用以下信息来帮助您诊断和修复在使用 AWS 和 IAM 时可能遇到的常见问题。

主题

- 我无权在以下位置执行操作 AWS
- 我无权执行 iam：PassRole

- 我想允许我以外的人 AWS 账户 访问我的 AWS 资源

我无权在以下位置执行操作 AWS

如果您收到错误提示，表明您无权执行某个操作，则您必须更新策略以允许执行该操作。

当 mateojackson IAM 用户尝试使用控制台查看有关虚构 *my-example-widget* 资源的详细信息，但不拥有虚构 awes:*GetWidget* 权限时，会发生以下示例错误。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 awes:GetWidget on resource: my-example-widget
```

在此情况下，必须更新 mateojackson 用户的策略，以允许使用 awes:*GetWidget* 操作访问 *my-example-widget* 资源。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

我无权执行 iam：PassRole

如果您收到一个错误，表明您无权执行 iam:PassRole 操作，则必须更新策略以允许您将角色传递给 AWS。

有些 AWS 服务 允许您将现有角色传递给该服务，而不是创建新的服务角色或服务相关角色。为此，您必须具有将角色传递到服务的权限。

当名为 marymajor 的 IAM 用户尝试使用控制台在 AWS中执行操作时，会发生以下示例错误。但是，服务必须具有服务角色所授予的权限才可执行此操作。Mary 不具有将角色传递到服务的权限。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

在这种情况下，必须更新 Mary 的策略以允许她执行 iam:PassRole 操作。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

我想允许我以外的人 AWS 账户 访问我的 AWS 资源

您可以创建一个角色，以便其他账户中的用户或您组织外的人员可以使用该角色来访问您的资源。您可以指定谁值得信赖，可以担任角色。对于支持基于资源的策略或访问控制列表（ACL）的服务，您可以使用这些策略向人员授予对您的资源的访问权。

要了解更多信息，请参阅以下内容：

- 要了解是否 AWS 支持这些功能，请参阅如何 AWS 服务 使用 IAM。
- 要了解如何提供对您拥有的资源的访问权限 AWS 账户 ，请参阅 IAM 用户指南中的向您拥有 AWS 账户 的另一个 IAM 用户提供访问权限。
- 要了解如何向第三方提供对您的资源的访问权限 AWS 账户，请参阅 IAM 用户指南中的向第三方提供访问权限。 AWS 账户
- 要了解如何通过身份联合验证提供访问权限，请参阅《IAM 用户指南》中的为经过外部身份验证的用户（身份联合验证）提供访问权限。
- 要了解使用角色和基于资源的策略进行跨账户存取之间的差别，请参阅《IAM 用户指南》中的 IAM 角色与基于资源的策略有何不同。

此 AWS 产品或服务的合规性验证

要了解是否属于特定合规计划的范围，请参阅AWS 服务 “按合规计划划分的范围” ”，然后选择您感兴趣的合规计划。 AWS 服务 有关一般信息，请参阅AWS 合规计划AWS。

您可以使用下载第三方审计报告 AWS Artifact。有关更多信息，请参阅中的 “下载报告” 中的 “ AWS Artifact。

您在使用 AWS 服务 时的合规责任取决于您的数据的敏感性、贵公司的合规目标以及适用的法律和法规。 AWS 提供了以下资源来帮助实现合规性：

- 安全与合规性快速入门指南 — 这些部署指南讨论了架构注意事项，并提供了部署以安全性和合规性为重点 AWS 的基准环境的步骤。
- 在 A@@ mazon Web Services 上构建 HIPAA 安全与合规架构 — 本白皮书描述了各公司如何使用 AWS 来创建符合 HIPAA 资格的应用程序。

> **Note**
>
> 并非所有 AWS 服务 人都符合 HIPAA 资格。有关更多信息，请参阅符合 HIPAA 要求的服务参考。

- AWS 合规资源AWS — 此工作簿和指南集可能适用于您所在的行业和所在地区。
- AWS 客户合规指南 — 从合规角度了解责任共担模式。这些指南总结了保护的最佳实践， AWS 服务 并将指南映射到跨多个框架（包括美国国家标准与技术研究院 (NIST)、支付卡行业安全标准委员会 (PCI) 和国际标准化组织 (ISO)）的安全控制。

- 使用AWS Config 开发人员指南中的规则评估资源 — 该 AWS Config 服务评估您的资源配置在多大程度上符合内部实践、行业准则和法规。
- AWS Security Hub— 这 AWS 服务 可以全面了解您的安全状态 AWS。Security Hub 通过安全控件评估您的 AWS 资源并检查其是否符合安全行业标准和最佳实践。有关受支持服务及控件的列表，请参阅 Security Hub 控件参考。
- Amazon GuardDuty — 它通过监控您的 AWS 账户环境中是否存在可疑和恶意活动，来 AWS 服务检测您的工作负载、容器和数据面临的潜在威胁。 GuardDuty 通过满足某些合规性框架规定的入侵检测要求，可以帮助您满足各种合规性要求，例如 PCI DSS。
- AWS Audit Manager— 这 AWS 服务 可以帮助您持续审计 AWS 使用情况，从而简化风险管理以及对法规和行业标准的合规性。

本 AWS 产品或服务通过其支持的特定 Amazon Web Services (AWS) 服务遵循分担责任模式。有关 AWS 服务安全信息，请参阅AWS 服务安全文档页面和合规计划合 AWS 规工作范围内的AWS 服务。

本 AWS 产品或服务的弹性

AWS 全球基础设施是围绕 AWS 区域 可用区构建的。

AWS 区域 提供多个物理分隔和隔离的可用区，这些可用区通过低延迟、高吞吐量和高度冗余的网络连接。

利用可用区，您可以设计和操作在可用区之间无中断地自动实现失效转移的应用程序和数据库。与传统的单个或多个数据中心基础设施相比，可用区具有更高的可用性、容错性和可扩展性。

有关 AWS 区域和可用区的更多信息，请参阅AWS 全球基础设施。

本 AWS 产品或服务通过其支持的特定 Amazon Web Services (AWS) 服务遵循分担责任模式。有关 AWS 服务安全信息，请参阅AWS 服务安全文档页面和合规计划合 AWS 规工作范围内的AWS 服务。

本 AWS 产品或服务的基础设施安全

本 AWS 产品或服务使用托管服务，因此受到 AWS 全球网络安全的保护。有关 AWS 安全服务以及如何 AWS 保护基础设施的信息，请参阅AWS 云安全。要使用基础设施安全的最佳实践来设计您的 AWS 环境，请参阅 S AWS ecurity Pillar Well-Architected Fram ework 中的基础设施保护。

您可以使用 AWS 已发布的 API 调用通过网络访问此 AWS 产品或服务。客户端必须支持以下内容：

- 传输层安全性协议 (TLS)。我们要求使用 TLS 1.2，建议使用 TLS 1.3。

- 具有完全向前保密 (PFS) 的密码套件，例如 DHE（临时 Diffie-Hellman）或 ECDHE（临时椭圆曲线 Diffie-Hellman）。大多数现代系统（如 Java 7 及更高版本）都支持这些模式。

此外，必须使用访问密钥 ID 和与 IAM 委托人关联的秘密访问密钥来对请求进行签名。或者，您可以使用 AWS Security Token Service（AWS STS）生成临时安全凭证来对请求进行签名。

本 AWS 产品或服务通过其支持的特定 Amazon Web Services (AWS) 服务遵循分担责任模式。有关 AWS 服务安全信息，请参阅AWS 服务安全文档页面和合规计划合 AWS 规工作范围内的AWS 服务。

中的配置和漏洞分析 AWS Toolkit for Visual Studio

随着新功能或修补程序的开发，Toolkit for Visual Studio 会发布到 Visual Studio Marketplace。这些更新有时包含安全更新，因此让 Toolkit for Visual Studio 保持最新状态非常重要。

验证是否已为扩展启用自动更新

1. 通过选择工具、扩展和更新（Visual Studio 2017）或扩展、管理扩展（Visual Studio 2019）打开扩展管理器。
2. 选择更改扩展和更新设置（Visual Studio 2017）或更改扩展的设置（Visual Studio 2019）。
3. 调整环境的设置。

如果选择为扩展禁用自动更新，请务必按适合您环境的间隔检查 Toolkit for Visual Studio 的更新。

《AWS Toolkit for Visual Studio 用户指南》的文档历史记录

最近文档更新时间：2021 年 4 月 21 日

文档历史记录

下表描述了《AWS Toolkit for Visual Studio 用户指南》的最新重要更改。如需获取对此文档的更新的通知，您可以订阅 RSS 源。

变更	说明	日期
内容更新和维护	更新内容以了解用户界面和 AWS 样式指南的更改。	2024 年 3 月 6 日
内容更新和维护	更新内容以了解用户界面和 AWS 样式指南的更改。	2024 年 3 月 6 日
内容更新和维护	更新内容以了解用户界面和 AWS 样式指南的更改。	2024 年 3 月 6 日
内容更新和维护	更新内容以了解用户界面和 AWS 样式指南的更改。	2024 年 3 月 6 日
内容更新和维护	更新内容以了解用户界面和 AWS 样式指南的更改。	2024 年 3 月 6 日
设置和身份验证的更新	设置和身份验证主题已更新，以改善安全性和 Toolkit 入门体验。要查看更改，请参阅入门和身份验证和访问主题目录。	2023 年 6 月 22 日
身份验证和访问	现在，提供 AWS 凭据是身份验证和访问。重构目录和副主题以满足 AWS 风格和安全要求。	2023 年 5 月 4 日

新的常规问题排查主题	故障排除主题包含 AWS Toolkit for Visual Studio 和相关服务的一般故障排除信息。	2023 年 4 月 30 日
设置部分和主题的更新	本用户指南中的设置 AWS Toolkit for Visual Studio 部分和主题已更新，以改善 AWS Toolkit for Visual Studio入门体验。	2023 年 1 月 30 日
设置部分和主题的更新	本用户指南中的设置 AWS Toolkit for Visual Studio 部分和主题已更新，以改善 AWS Toolkit for Visual Studio入门体验。	2023 年 1 月 30 日
添加了 2022 年的 AWS Toolkit for Visual Studio 信息	对 Visual Studio 2022 的支持已添加到 AWS Toolkit for Visual Studio。	2022 年 12 月 20 日
发布到 AWS 指南的更新	文档更新反映了对 GA 发布服务所做的更改。	2022 年 7 月 6 日
标题更新和位置调整	为了更好地反映内容，对标题进行了细微更改。指南现在位于 “发布 AWS 指南” 中。	2022 年 7 月 6 日
部署到 AWS：标题和内容更新	正式标题为：使用 AWS 工具包进行部署的指南部分包含更新的目录 (TOC)，现在的标题是：部署到 AWS。以下指南已完成弃用，无法再访问：部署到 Elastic Beanstalk（旧版）和部署到（旧版）。AWS CloudFormation 有关部署到 Elastic Beanstalk 和 CloudFormation 的更新内容可从本指南更新的目录中找到。	2022 年 7 月 6 日

“部署 ASP.NET Core 2.0 应用程序（Fargate）”现已属于旧版指南	本文档涉及旧版服务和功能。有关更新的指南和内容，请参阅 AWS .NET deployment tool 指南和更新的部署到 AWS 目录。	2022 年 7 月 6 日
“部署 ASP.NET 应用程序”现已属于旧版指南	本文档涉及旧版服务和功能。有关更新的指南和内容，请参阅 AWS .NET deployment tool 指南和更新的部署到 AWS 目录。	2022 年 7 月 6 日
“部署 ASP.NET 应用程序”现已属于旧版指南	本文档涉及旧版服务和功能。有关更新的指南和内容，请参阅 AWS .NET deployment tool 指南和更新的部署到 AWS 目录。	2022 年 7 月 6 日
新指南主题：在 Visual Studio 中使用 CloudWatch 日志	为 Visual Studio 中的 Amazon CloudWatch 日志集成指南创建了新的概述主题。	2022 年 6 月 29 日
新指南主题：为 Visual Studio 设置 CloudWatch 日志集成	在 Visual Studio 指南中为亚马逊 CloudWatch 日志集成创建了新的设置部分。	2022 年 6 月 29 日
CloudWatch 针对 Visual Studio	为在 Visual Studio 中集成亚马逊 CloudWatch 日志创建了新的指南，包括指南主题：为 Visual Studio 设置 CloudWatch 日志和在 Visual Studio 中使用 CloudWatch 日志。	2022 年 6 月 29 日
发布到 AWS	“发布到” 不再 AWS 处于预览状态。更新反映了用户界面的更改和发布建议的改进。	2022 年 6 月 1 日

全新 “发布” AWS 可供预览	增强的部署体验，可指导您选择哪种 AWS 服务适合您的应用程序。	2021 年 10 月 21 日
凭证支持 SSO 和 MFA AWS	更新以记录对 AWS 单点登录（IAM 身份中心）和凭证中的 AWS 多因素身份验证的新支持。	2021 年 4 月 21 日
创建 Docker 镜像的基本 AWS Lambda 项目	添加了对 Lambda 容器映像的支持。	2020 年 12 月 1 日
安全性内容	增加了安全性内容。	2020 年 2 月 6 日
提供 AWS 凭证	更新了有关在共享的 AWS 凭证文件中创建凭证配置文件的信息。	2019 年 6 月 20 日
在 Visual Studi AWS o AWS 工具包中使用 Lambda 项目	Visual Studio 2019 的支持已添加到 Visual Studio 的 AWS 工具包中。	2019 年 3 月 28 日
教程：创建 Amazon Rekogniti on Lambda 应用程序	Visual Studio 2019 的支持已添加到 Visual Studio 的 AWS 工具包中。	2019 年 3 月 28 日
教程：使用 AWS Lambda 构建和测试无服务器应用程序	Visual Studio 2019 的支持已添加到 Visual Studio 的 AWS 工具包中。	2019 年 3 月 28 日
设置 AWS Toolkit for Visual Studio	已将对 Visual Studio 2019 的支持添加到 AWS Toolkit for Visual Studio。	2019 年 3 月 28 日
部署 ASP.NET Core 2.0 应用程序（Fargate）	Visual Studio 2019 的支持已添加到 Visual Studio 的 AWS 工具包中。	2019 年 3 月 28 日

部署 ASP.NET Core 2.0 应用程序（EC2）	Visual Studio 2019 的支持已添加到 Visual Studio 的 AWS 工具包中。	2019 年 3 月 28 日
在 Visual Studio 中创建 AWS CloudFormation 模板项目	Visual Studio 2019 的支持已添加到 Visual Studio 的 AWS 工具包中。	2019 年 3 月 28 日
Container Service 的详细视图	添加了有关 E AWS xplorer 提供的亚马逊弹性容器服务集群和容器存储库的详细视图的信息。	2018 年 2 月 16 日
部署到 Amazon EC2 Container Service	添加了有关部署到 Amazon EC2 Container Service 的信息。	2018 年 2 月 16 日
使用 Fargate 部署 Container Service	添加了有关如何使用 Fargate 启动类型通过 Amazon ECS 部署针对 Linux 的容器化 ASP.NET 内核 2.0 应用程序的信息。	2018 年 2 月 16 日
使用 EC2 部署 Container Service	添加了有关如何使用 EC2 启动类型通过 Amazon ECS 部署针对 Linux 的容器化 ASP.NET 内核 2.0 应用程序的信息。	2018 年 2 月 16 日
用于部署到 Amazon EC2 Container Service 的凭证	添加了有关如何在部署到 Amazon EC2 Container Service 时指定凭证的信息。	2018 年 2 月 16 日

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。