사용 설명서

AWS Amazon Q를 사용한 도구 키트

AWS Amazon Q를 사용한 도구 키트: 사용 설명서

Table of Contents

AWS Amazon Q를 사용한 도구 키트

이 설명서는 AWS Toolkit for Visual Studio with Amazon Q에 대한 사용 설명서입니다. AWS Toolkit for VS Code를 찾고 있다면 AWS Toolkit for Visual Studio Code용 사용 설명서를 참조하세요.

Amazon Q를 사용하는 AWS Toolkit for Visual Studio란?

AWS Toolkit for Visual Studio with Amazon Q는 Amazon Web Services를 사용하는 .NET 애플리케이션을 더 쉽게 개발, 디버깅 및 배포할 수 있는 Visual Studio IDE의 확장입니다. AWS Toolkit with Amazon Q는 Visual Studio 버전 2022 이상에서 지원됩니다. 툴킷을 다운로드하고 설치하는 방법에 대한 자세한 내용은 이 사용 설명서의 설치 및 설정 주제를 참조하세요.

> **Note**
>
> Toolkit for Visual Studio는 Visual Studio 2008, 2010, 2012, 2013, 2015, 2017 및 2019 버전용으로도 출시되었습니다. 그러나 이러한 버전은 더 이상 지원되지 않습니다. 자세한 내용은 이 사용 설명서의 설치 및 설정 주제를 참조하세요.

AWS Toolkit with Amazon Q에는 개발 경험을 개선하기 위한 다음과 같은 기능이 포함되어 있습니다.

AWS 탐색기

AWS 탐색기 도구 창은 IDE의 보기 메뉴에서 액세스할 수 있으며 Visual Studio의 AWS 서비스와 상호 작용할 수 있습니다. 지원되는 AWS 서비스 및 기능 목록은이 사용 설명서의 AWS 서비스 작업 주제를 참조하세요.

Amazon Q

Visual Studio의 Amazon Q Developer와 채팅하여에서 빌드하는 방법에 대해 질문 AWS 하고 소프트웨어 개발에 대한 지원을 받으세요. Amazon Q는 코딩 개념 및 코드 조각을 설명하고, 코드 및 유닛 테스트를 생성하며, 디버깅 또는 리팩터링을 통해 코드를 개선할 수 있습니다.

Toolkit for Visual Studio용 Amazon Q를 설치하고 설정하려면 이 사용 설명서의 시작하기 주제를 참조하세요. Amazon Q Developer 작업에 대한 자세한 내용은 Amazon Q Developer 사용 설명서의 IDE의 Amazon Q Developer 주제를 참조하세요. Amazon Q 요금제 및 가격에 대한 자세한 내용은 Amazon Q 요금 안내서를 참조하세요.

관련 정보

문제를 게시하거나 현재 게시된 문제를 보려면 https://github.com/aws/aws-toolkit-visual-studio/issues를 방문하세요.

Visual Studio에 대한 자세한 내용은 https://visualstudio.microsoft.com/vs/를 참조하십시오.

Amazon Q

Amazon Q란 무엇인가요?

2024년 4월 30일부터 Amazon CodeWhisperer는 Amazon Q Developer에 통합되며, 인라인 코드 제안 및 보안 스캔도 여기에 포함됩니다.

AWS Toolkit for Visual Studio에서 Amazon Q Developer 작업에 대해 자세히 알아보려면 Amazon Q Developer 사용 설명서의 IDE의 Amazon Q Developer 주제를 참조하세요. Amazon Q 요금제 및 가격에 대한 자세한 내용은 Amazon Q 요금 안내서를 참조하세요.

Toolkit for Visual Studio 다운로드

IDE의 Visual Studio Marketplace를 통해 Toolkit for Visual Studio를 다운로드, 설치 및 설정할 수 있습니다. 자세한 지침은 이 사용 설명서의 시작하기 항목에서 AWS Toolkit for Visual Studio 설치 섹션을 참조하세요.

Visual Studio Marketplace에서 툴킷 다운로드

웹 브라우저에서 AWS Visual Studio 다운로드 사이트로 이동하여 Toolkit for Visual Studio 설치 파일을 다운로드하세요.

AWS의 추가 IDE 툴킷

Toolkit for Visual Studio 외에도 AWS는 VS Code 및 JetBrains를 위한 IDE 툴킷도 제공합니다.

AWS Toolkit for Visual Studio Code 링크

- 이 링크를 따라 VS Code Marketplace에서 AWS Toolkit for Visual Studio Code을 다운로드하세요.
- AWS Toolkit for Visual Studio Code에 대한 자세한 정보는 AWS Toolkit for Visual Studio Code 사용 설명서를 참조하세요.

AWS Toolkit for JetBrains 링크

- 이 링크를 따라 JetBrains Marketplace에서 AWS Toolkit for JetBrains을 다운로드하세요.
- AWS Toolkit for JetBrains에 대한 자세한 정보는 AWS Toolkit for JetBrains 사용 설명서를 참조하세요.

시작하기

AWS Toolkit for Visual Studio를 통해 Visual Studio 통합 개발 환경(IDE)에서 AWS 서비스와 리소스를 사용할 수 있습니다.

시작하는 데 도움이 되도록 다음 항목에서는 AWS Toolkit for Visual Studio 설치, 설정 및 구성 방법을 설명합니다.

주제

설치 및 설정 AWS Toolkit for Visual Studio

다음 주제에서는 AWS Toolkit for Visual Studio를 다운로드, 설치, 설정 및 제거하는 방법에 대해 설명합니다.

주제

사전 조건

다음은 지원되는 AWS Toolkit for Visual Studio버전을 설정하기 위한 사전 요구 사항입니다.

- Visual Studio 19 이상 릴리스
- Windows 10 이상 Windows 릴리스
- Windows 및 Visual Studio에 대한 관리자 액세스
- 활성 AWS IAM 자격 증명

> **Note**
>
> 지원되지 않는 버전의 AWS Toolkit for Visual Studio 는 Visual Studio 2008, 2010, 2012, 2013, 2015 및 2017에서 사용할 수 있습니다. 지원되지 않는 버전을 다운로드하려면 AWS Toolkit for Visual Studio 랜딩 페이지로 이동한 다음 다운로드 링크 목록에서 원하는 버전을 선택하세요. IAM 보안 인증 정보에 대해 자세히 알아보거나 계정에 가입하려면 AWS 콘솔 게이트웨이를 방문하세요.

설치 AWS Toolkit for Visual Studio

를 설치하려면 다음 절차에서 Visual Studio 버전을 AWS Toolkit for Visual Studio찾고 필요한 단계를 완료합니다. 의 모든 버전에 대한 다운로드 링크는 AWS Toolkit for Visual Studio랜딩 페이지에서 찾을 AWS Toolkit for Visual Studio 수 있습니다.

> **Note**
>
> 설치 중에 문제가 발생하면이 가이드의 설치 문제 해결 주제를 AWS Toolkit for Visual Studio 참조하세요.

AWS Toolkit for Visual Studio for Visual Studio 2022 설치

Visual Studio에서 AWS Toolkit for Visual Studio 2022를 설치하려면 다음 단계를 완료합니다.

1. 기본 메뉴에서 확장으로 이동한 다음 확장 관리를 선택하세요.
2. 검색 상자에서 AWS를 검색하세요.
3. 관련 버전의 Visual Studio 2022 다운로드 버튼을 선택하고 설치 안내를 따릅니다.

   > **Note**
   >
   > 설치 프로세스를 완료하려면 Visual Studio를 수동으로 닫고 다시 시작해야 할 수 있습니다.

4. 다운로드 및 설치가 완료되면 보기 메뉴에서 AWS 탐색기를 AWS Toolkit for Visual Studio 선택하여를 열 수 있습니다.

AWS Toolkit for Visual Studio for Visual Studio 2019 설치

Visual Studio에서 AWS Toolkit for Visual Studio 2019를 설치하려면 다음 단계를 완료합니다.

1. 기본 메뉴에서 확장으로 이동한 다음 확장 관리를 선택하세요.
2. 검색 상자에서 AWS를 검색하세요.
3. Visual Studio 2017 및 2019의 다운로드 버튼을 선택하고 안내를 따릅니다.

> **Note**
>
> 설치 프로세스를 완료하려면 Visual Studio를 수동으로 닫고 다시 시작해야 할 수 있습니다.

4. 다운로드 및 설치가 완료되면 보기 메뉴에서 AWS 탐색기를 AWS Toolkit for Visual Studio 선택하여를 열 수 있습니다.

제거 AWS Toolkit for Visual Studio

를 제거하려면 다음 절차에서 Visual Studio 버전을 AWS Toolkit for Visual Studio찾아 필요한 단계를 완료합니다.

AWS Toolkit for Visual Studio for Visual Studio 2022 제거

Visual Studio에서 AWS Toolkit for Visual Studio 2022를 제거하려면 다음 단계를 완료합니다.

1. 기본 메뉴에서 확장으로 이동한 다음 확장 관리를 선택하세요.
2. 확장 관리 탐색 메뉴에서 설치됨 제목을 확장하세요.
3. AWS Toolkit for Visual Studio 2022 확장 프로그램을 찾아 제거 버튼을 선택하세요.

> **Note**
>
> 탐색 메뉴의 설치됨 섹션에서가 표시되지 AWS Toolkit for Visual Studio 않으면 Visual Studio를 다시 시작해야 할 수 있습니다.

4. 화면 프롬프트에 따라 제거 프로세스를 완료하세요.

AWS Toolkit for Visual Studio for Visual Studio 2019 제거

Visual Studio에서 AWS Toolkit for Visual Studio 2019를 제거하려면 다음 단계를 완료하세요.

1. 기본 메뉴에서 도구로 이동한 다음 확장 관리를 선택하세요.
2. 확장 관리 탐색 메뉴에서 설치됨 제목을 확장하세요.
3. AWS Toolkit for Visual Studio 2019 확장 프로그램을 찾아 제거 버튼을 선택하세요.
4. 화면 프롬프트에 따라 제거 프로세스를 완료하세요.

AWS Toolkit for Visual Studio for Visual Studio 2017 제거

Visual Studio에서 AWS Toolkit for Visual Studio 2017을 제거하려면 다음 단계를 완료하세요.

1. 기본 메뉴에서 도구로 이동한 다음 확장 및 업데이트를 선택하세요.
2. 확장 및 업데이트 탐색 메뉴에서 설치됨 제목을 확장하세요.
3. AWS Toolkit for Visual Studio 2017 확장 프로그램을 찾아 제거 버튼을 선택하세요.
4. 화면 프롬프트에 따라 제거 프로세스를 완료하세요.

AWS Toolkit for Visual Studio for Visual Studio 2013 또는 2015 제거

AWS Toolkit for Visual Studio 2013 또는 2015를 제거하려면 다음 단계를 완료합니다.

1. Windows 제어판에서 프로그램 및 기능을 엽니다.

   > **Note**
   >
   > Windows 명령 프롬프트 또는 Windows 실행 대화 상자에서 `appwiz.cpl`을 실행하여 프로그램 및 기능을 즉시 열 수 있습니다.

2. 설치된 프로그램 목록에서 Windows용AWS 도구 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 엽니다.
3. 제거를 선택하고 지시에 따라 제거 프로세스를 완료하세요.

> **Note**
>
> 제거 프로세스 중에는 샘플 디렉터리가 삭제되지 않습니다. 이 디렉터리는 샘플을 수정한 경우 유지됩니다. 이 디렉터리는 수동으로 제거해야 합니다.

에 연결 AWS

다음 섹션에서는 Amazon Q를 사용하여 AWS Toolkit for Visual Studio를 시작하는 방법을 설명합니다. 확장 프로그램을 설치한 후 Visual Studio를 처음 시작하면 편집기 창에 시작하기가 표시됩니다. 시작하기 탭에서 다음 작업을 완료할 수 있습니다.

- Amazon Q 및 AWS 도구 키트를 활성화 또는 비활성화합니다.
- 새 자격 증명을 추가하고 인증합니다.
- 기존 자격 증명으로 인증합니다.
- Amazon Q 및 AWS 툴킷 작업을 시작하는 데 도움이 되는 설명서 및 자습서에 액세스합니다.

사전 조건

Amazon Q 및 AWS 도구 키트 작업을 시작하려면 자격 AWS 증명으로 인증해야 합니다. 이전에 다른 AWS 도구 또는 서비스(예: AWS Command Line Interface)를 통해 AWS 계정 및 인증을 설정한 경우 AWS 도구 키트는 자격 증명을 자동으로 감지합니다. 계정을 처음 AWS 사용하거나 생성하지 않은 경우 가입 AWS 포털에서 AWS 계정에 가입할 수 있습니다. 새 AWS 계정 설정에 대한 자세한 내용은 AWS 설정 사용 설명서의 개요 주제를 참조하세요.

도구 키트 AWS 에서에 연결

AWS 도구 키트에서 AWS 계정에 연결하려면 언제든지 다음을 완료하여 시작하기 탭을 엽니다.

Visual Studio에서 시작하기 탭 열기

1. Visual Studio 주 메뉴에서 확장을 펼친 다음 AWS 도구 키트 하위 메뉴를 확장합니다.
2. 시작하기를 선택합니다.
3. Visual Studio 편집기 창에서 시작하기 탭이 열립니다.

시작하기 탭에는 두 가지 기본 섹션이 있습니다.

- 기능:이 섹션에서는 Amazon Q 및 AWS 도구 키트와 같은 기능을 활성화하거나 비활성화할 수 있습니다.
- 설명서 및 자습서: 활성화된 기능에 대한 참조 모음입니다.

> **Note**
>
> 설명서 및 자습서 섹션은 하나 이상의 기능이 활성화된 경우에만 표시됩니다.

Amazon Q Developer

시작하기 탭의 Amazon Q 섹션에서 Amazon Q를 활성화 또는 비활성화하거나, 새 연결을 추가하거나, 다른 AWS 연결로 전환할 수 있습니다. 이러한 작업을 보거나 액세스하려면 Amazon Q를 활성화해야 합니다. Amazon Q를 활성화하려면 활성화 버튼을 클릭합니다.

Amazon Q가 비활성화되면 모든 Amazon Q 기능이 Visual Studio에서 완전히 제거됩니다. Amazon Q를 활성화하면 시작하기 탭에서 Amazon Q에 대한 설정 인증이 자동으로 열립니다. 계속하려면 자격 AWS IAM Identity Center 증명으로 인증하여 프로페셔널 티어에 액세스하거나 AWS Builder ID로 인증하여 프리 티어에 액세스해야 합니다. 각 티어 옵션에 대한 자세한 내용은 Amazon Q Developer 사용 설명서의 Amazon Q Developer의 서비스 티어 이해 주제를 참조하세요.

계속하려면 다음 절차 중 하나를 완료합니다.

IAM Identity Center를 사용한 프로페셔널 티어 인증

> **Note**
>
> 프로페셔널 티어로 인증하는 데 필요한 프로필 이름, 시작 URL, 프로필 리전 또는 SSO 리전 필드는 일반적으로 회사 또는 조직의 관리자가 제공합니다. IAM Identity Center 자격 증명에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 IAM Identity Center란 무엇인가요? 주제를 참조하세요.

1. Amazon Q로 시작하기: AWS 도구 키트 화면에서 Amazon Q 타일의 로그인 버튼을 선택하여 Amazon Q에 대한 인증 설정 화면으로 이동합니다.
2. Amazon Q에 대한 인증 설정 화면에서 프로페셔널 티어 섹션으로 이동하여 필수 필드를 입력하고 연결 버튼을 선택합니다.
3. 기본 웹 브라우저에서 요청 AWS 권한 부여 포털을 열 것인지 확인합니다.

4. AWS 권한 부여 요청 포털에 필요한 단계를 완료하면 브라우저를 닫고 Visual Studio로 돌아가도 안전하다는 알림을 받게 됩니다.
5. 프로세스가 완료되면 시작하기 탭에 Amazon Q가 IAM Identity Center와 연결되었음이 표시됩니다.

AWS Builder ID를 사용한 프리 티어 인증

> **Note**
>
> AWS Builder ID에 대한 자세한 내용은 로그인 사용 설명서의 AWS Builder ID로 AWS 로그인 주제를 참조하세요.

1. Amazon Q로 시작하기: AWS 도구 키트 화면에서 Amazon Q 타일의 로그인 버튼을 선택하여 Amazon Q에 대한 인증 설정 화면으로 이동합니다.
2. Amazon Q에 대한 인증 설정 화면에서 프리 티어 섹션으로 이동하여 가입 또는 로그인 버튼을 선택합니다.
3. 기본 웹 브라우저에서 요청 AWS 권한 부여 포털을 열 것인지 확인합니다.
4. AWS 권한 부여 요청 포털에 필요한 단계를 완료하면 브라우저를 닫고 Visual Studio로 돌아가도 안전하다는 알림을 받게 됩니다.
5. 시작하기 탭에서 프로세스가 완료되면 Amazon Q가 업데이트되어 AWS Builder ID에 연결되었음을 표시합니다.

IAM Identity Center 또는 AWS Builder ID 자격 증명으로 인증한 후 Visual Studio의 Amazon Q에 액세스할 수 있습니다. 또한 시작하기 탭에서 다음 작업을 수행할 수 있습니다.

- 로그아웃: 모든 Amazon Q 함수에서 현재 자격 증명 연결을 해제합니다. Amazon Q는 활성화된 상태로 유지되지만 대부분의 기능은 작동하지 않습니다.
- Amazon Q 비활성화: Visual Studio에서 모든 Amazon Q 기능을 완전히 비활성화합니다.

AWS 도구 키트

AWS 도구 키트 시작하기 탭의 AWS 도구 키트 섹션에서 AWS 도구 키트를 활성화 또는 비활성화하거나, 새 연결을 추가하거나, 다른 AWS 연결로 전환할 수 있습니다. 이러한 작업을 보거나 액세스하려면

먼저 AWS 도구 키트를 활성화해야 합니다. AWS 도구 키트를 활성화하려면 활성화 버튼을 클릭합니다.

AWS 도구 키트가 활성화되면 도구 키트에 대한 AWS 설정 인증이 도구 키트 시작하기 AWS 탭에서 자동으로 로드됩니다. 계속하려면 AWS IAM Identity Center 자격 증명 또는 IAM 사용자 역할 자격 증명으로 인증해야 합니다.

> **Note**
>
> IAM Identity Center 자격 증명에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 IAM Identity Center란 무엇인가요? 주제를 참조하세요. IAM 사용자 역할 자격 증명에 대한 자세한 내용은 AWS SDK 및 도구 참조 가이드의 AWS 액세스 키: 장기 자격 증명 주제를 참조하세요.

IAM Identity Center로 인증 및 연결

1. Amazon Q로 시작하기: AWS 도구 키트 화면에서 AWS 도구 키트 타일의 로그인 버튼을 선택하여 AWS 도구 키트 인증 설정 화면으로 이동합니다.
2. AWS 도구 키트에 대한 인증 설정 화면의 프로파일 유형 드롭다운 메뉴에서 IAM Identity Center(Single Sign-On의 후속)를 선택합니다.
3. 기존 프로필에서 선택 또는 새 프로필 추가 드롭다운 메뉴에서 기존 프로필을 선택하거나 새 프로필 추가를 선택하여 새 프로필 정보를 추가합니다.

   > **Note**
   >
   > 기존 프로필을 선택하는 경우 7단계로 이동합니다.

4. 프로필 이름 필드에 인증하려는 IAM Identity Center 계정과 연결된 **`profile name`**을 입력합니다.
5. 시작 URL 텍스트 필드에서 IAM Identity Center 자격 증명에 연결된 **`Start URL`**을 입력합니다.
6. 프로필 리전(기본값은 us-east-1) 드롭다운 메뉴에서 인증하려는 IAM Identity Center 사용자 프로필에 정의된 프로필 리전을 선택합니다.
7. SSO 리전(기본값은 us-east-1) 드롭다운 메뉴에서 IAM Identity Center 자격 증명으로 정의된 SSO 리전을 선택합니다.
8. 연결 버튼을 선택하여 기본 웹 브라우저에서 AWS 권한 부여 요청 사이트를 엽니다.

9. 기본 웹 브라우저의 메시지에 따라 인증 프로세스가 완료되면 알림을 받게 되며, 브라우저를 닫고 Visual Studio로 돌아가도 안전합니다.
10. 프로세스가 완료되면 시작하기 탭에 AWS 툴킷 섹션에 IAM Identity Center와 연결되었음이 표시됩니다.

IAM 사용자 역할 자격 증명으로 인증 및 연결

1. Amazon Q로 시작하기: AWS 도구 키트 화면에서 AWS 도구 키트 타일의 로그인 버튼을 선택하여 AWS 도구 키트 인증 설정 화면으로 이동합니다.
2. AWS 도구 키트에 대한 인증 설정 화면의 프로필 유형 드롭다운 메뉴에서 IAM 사용자 역할을 선택합니다.
3. 기존 프로필에서 선택 또는 새 프로필 추가 드롭다운 메뉴에서 **`Add new profile`**을 선택합니다.

> **Note**
>
> 목록에서 기존 프로필 이름을 선택하는 경우 8단계로 건너뜁니다.

4. 프로필 이름 텍스트 필드에 새 프로필의 이름을 입력합니다.
5. 액세스 키 ID 텍스트 필드에 인증하려는 프로필의 **`Access Key ID`**를 입력합니다.
6. 보안 키 텍스트 필드에 인증하려는 프로필의 **`Secret Key`**를 입력합니다.
7. 스토리지 위치(기본값은 공유 자격 증명 파일) 드롭다운 메뉴에서 자격 증명을 공유 자격 증명 파일 또는 .NET 암호화 스토어와 함께 저장할지 여부를 지정합니다.
8. 프로필 리전(기본값은 us-east-1) 드롭다운 메뉴에서 인증하려는 프로필에 연결된 파티션 및 프로필 리전을 선택합니다.
9. 연결 버튼을 선택하여이 프로필을 AWS 스토리지 위치에 추가하거나 인증합니다 AWS.
10. 프로세스가 완료되면 시작하기 탭에서 AWS 툴킷 섹션에 IAM 사용자 역할 자격 증명에 연결되었음을 표시합니다.

IAM Identity Center 또는 IAM 사용자 역할 자격 증명으로 인증한 후 Toolkit for Visual Studio에서 AWS 탐색기에 액세스할 수 있습니다. 또한 시작하기 탭에서 로그아웃하고 AWS Toolkit for Visual Studio with Amazon Q를 비활성화할 수 있습니다.

설명서 및 자습서

설명서 및 자습서 섹션은 AWS 서비스 및 기능 기본 설정에 따라 설명서 및 자습서 제안으로 자동으로 업데이트됩니다. 이러한 참조는 하나 이상의 기능이 활성화된 경우에만 표시됩니다.

AWS Toolkit for Visual Studio의 설치 문제 해결

다음 정보는 AWS Toolkit for Visual Studio 설정 중에 발생하는 일반적인 설치 문제를 해결하는 것으로 알려져 있습니다.

AWS Toolkit for Visual Studio 설치 중에 오류가 발생하거나 설치가 완료되었는지 여부가 확실하지 않은 경우 다음 각 섹션의 정보를 검토합니다.

Visual Studio 관리자 권한

AWS Toolkit for Visual Studio 확장 프로그램을 사용하려면 모든 AWS 서비스와 기능에 액세스할 수 있도록 관리자 권한이 필요합니다.

로컬 관리자 권한이 있는 경우 관리자 권한이 Visual Studio 인스턴스로 직접 확장되지 않을 수 있습니다.

로컬에서 관리자 권한으로 Visual Studio를 시작하려면 다음을 수행하세요.

1. Windows에서 Visual Studio 애플리케이션 시작 관리자(아이콘)를 찾습니다.
2. Visual Studio 아이콘에 대한 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 엽니다.
3. 컨텍스트 메뉴에서 관리자 권한으로 실행을 선택하세요.

관리자 권한으로 Visual Studio를 원격으로 시작하려면 다음을 수행하세요.

1. Windows에서 Visual Studio의 원격 인스턴스에 연결하는 데 사용하는 애플리케이션의 애플리케이션 시작 관리자를 찾습니다.
2. 애플리케이션에 대한 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 엽니다.
3. 컨텍스트 메뉴에서 관리자 권한으로 실행을 선택하세요.

> **Note**
>
> 프로그램을 로컬에서 시작하든 원격으로 연결하든 관계없이 Windows에서 관리 보안 인증 정보를 확인하라는 메시지가 표시될 수 있습니다.

설치 로그 가져오기

위에 있는 이전 관리자 권한 섹션의 단계를 완료하고 관리자 권한으로 Visual Studio를 실행 중이거나 연결하고 있는 것으로 확인된 경우 설치 로그 파일을 확보하면 다른 문제를 진단하는 데 도움이 될 수 있습니다.

`.vsix` 파일에서 AWS Toolkit for Visual Studio를 수동으로 설치하고 설치 로그 파일을 생성하려면 다음 단계를 완료하세요.

1. AWS Toolkit for Visual Studio 랜딩 페이지에서 다운로드 링크를 따라 설치하려는 AWS Toolkit for Visual Studio 버전의 `.vsix` 파일을 저장하세요.
2. Visual Studio 기본 메뉴에서 도구 헤더를 확장하고 명령줄 하위 메뉴를 확장한 다음 Visual Studio 개발자 명령 프롬프트를 선택하세요.
3. Visual Studio 개발자 명령 프롬프트에서 다음 형식으로 `vsixinstaller` 명령을 입력하세요.

   ```
   vsixinstaller /logFile:[file path to log file] [file path to Toolkit installation file]
   ```
4. 설치 로그를 만들려는 디렉터리의 파일 이름과 전체 파일 경로로 `[file path to log file]`을 바꿉니다. 지정한 파일 경로와 파일 이름을 사용하는 `vsixinstaller` 명령의 예제는 다음과 같습니다.

   ```
   vsixinstaller /logFile:C:\Users\Documents\install-log.txt [file path to AWSToolkitPackage.vsix]
   ```
5. `AWSToolkitPackage.vsix`가 위치한 디렉터리의 전체 파일 경로로 `[file path to Toolkit installation file]`을 바꿉니다.

   툴킷 설치 파일의 전체 파일 경로를 포함하는 `vsixinstaller` 명령의 예제는 다음과 비슷해야 합니다.

   ```
   vsixinstaller /logFile:[file path to log file] C:\Users\Downloads\AWSToolkitPackage.vsix
   ```
6. 파일 이름과 경로가 올바른지 확인한 다음 `vsixinstaller` 명령을 실행하세요.

전체 `vsixinstaller` 명령의 예제는 다음과 비슷합니다.

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

다양한 Visual Studio 확장 프로그램 설치

설치 로그 파일을 확보했는데도 여전히 설치 프로세스가 실패한 이유를 확인할 수 없는 경우 다른 Visual Studio 확장 프로그램을 설치할 수 있는지 확인합니다. 다양한 Visual Studio 확장 프로그램을 설치하면 설치 문제를 더 자세히 파악할 수 있습니다. Visual Studio 확장 프로그램을 설치할 수 없는 경우 AWS Toolkit for Visual Studio 대신 Visual Studio를 사용하여 문제를 해결해야 할 수도 있습니다.

지원에 문의

이 안내서에 포함된 모든 섹션을 검토했고 추가 리소스 또는 지원이 필요한 경우 AWS Toolkit for Visual StudioGithub 문제 사이트에서 지난 문제를 보거나 새 문제를 열 수 있습니다.

문제를 신속하게 해결하는 데 도움이 되려면 다음을 수행하세요.

- 과거 및 현재 문제를 확인하여 다른 사람들도 비슷한 상황을 겪었는지 확인하세요.
- 문제를 해결하기 위해 수행한 각 단계를 자세히 기록해 둡니다.
- AWS Toolkit for Visual Studio 또는 다른 프로그램을 설치하여 얻은 로그 파일을 모두 저장하세요.
- 새 문제에 AWS Toolkit for Visual Studio 설치 로그 파일을 첨부하세요.

프로필 및 Window 바인딩

Toolkit for Visual Studio를 위한 프로필 및 Window 바인딩

Toolkit for Visual Studio의 게시 도구, 마법사 및 기타 기능을 사용할 때는 다음 사항에 유의합니다.

- AWS 탐색기 창은 단일 프로필 및 리전에 한 번에 바인딩됩니다. AWS 탐색기에서 기본적으로 해당 바인딩된 프로필 및 리전으로 Windows가 열립니다.
- 새 창이 열리면 해당 AWS 탐색기 인스턴스를 사용하여 다른 프로필 또는 리전으로 전환할 수 있습니다.
- Toolkit for Visual Studio 게시 도구 및 기능은 AWS 탐색기에 설정된 프로필 및 리전을 기본값으로 자동으로 사용합니다.

- 게시 도구, 마법사 또는 기능에 새 프로필 또는 리전을 지정한 경우 이후에 생성되는 모든 리소스는 계속해서 새 프로필 및 리전 설정을 사용합니다.
- Visual Studio의 여러 인스턴스가 열려 있는 경우에는 각 인스턴스를 서로 다른 프로필 및 리전으로 바인딩할 수 있습니다.
- AWS 탐색기는 마지막으로 지정한 프로필 및 리전을 저장하며 가장 최근에 닫은 Visual Studio 인스턴스의 값은 그대로 유지됩니다.

인증 및 액세스

Amazon Q에서 AWS Toolkit for Visual Studio 작업을 시작하기 AWS 위해 로 인증할 필요는 없습니다. 그러나 대부분의 AWS 리소스는 AWS 계정을 통해 관리됩니다. Amazon Q 서비스 및 기능을 사용하여 AWS Toolkit for Visual Studio에 모두 액세스하려면 최소 2가지 유형의 계정 인증이 필요합니다.

1. AWS 계정에 대한 AWS Identity and Access Management (IAM) 또는 AWS IAM Identity Center 인증입니다. 대부분의 AWS 서비스와 리소스는 IAM 및 IAM Identity Center를 통해 관리됩니다.
2. AWS Builder ID는 다른 특정 AWS 서비스의 경우 선택 사항입니다.

다음 항목에 각 보안 인증 유형과 인증 방법에 대한 추가 세부 정보와 설정 지침이 포함되어 있습니다.

주제

- AWS 의 IAM Identity Center 자격 증명 AWS Toolkit for Visual Studio
- AWS IAM 자격 증명
- AWS 빌더 ID
- Toolkit for Visual Studio의 다중 인증(MFA)
- 외부 보안 인증 정보 설정
- 액세스를 허용하도록 방화벽 및 게이트웨이 업데이트

AWS 의 IAM Identity Center 자격 증명 AWS Toolkit for Visual Studio

AWS IAM Identity Center 는 AWS 계정 인증을 관리하는 데 권장되는 모범 사례입니다.

소프트웨어 개발 키트(SDKs) 및 용 IAM Identity Center를 설정하는 방법에 대한 자세한 지침은 SDK 및 도구 참조 안내서의 AWS Toolkit for Visual StudioIAM Identity Center 인증 섹션을 참조하세요. AWS SDKs

에서 IAM Identity Center로 인증 AWS Toolkit for Visual Studio

`credentials` 또는 `config` 파일에 IAM Identity Center 프로파일을 AWS Toolkit for Visual Studio 추가하여에서 IAM Identity Center로 인증하려면 다음 단계를 완료하세요.

1. 원하는 텍스트 편집기에서 <hone-directory>\.aws\credentials 파일에 저장된 AWS 자격 증명 정보를 엽니다.
2. 섹션 [default] 아래의 credentials file에서 이름이 지정된 IAM Identity Center 프로필의 템플릿을 추가하세요. 다음은 예제 템플릿입니다.

> **Important**
>
> credential 파일에 항목을 생성할 때 프로필이라는 단어를 사용하면 credential 파일 명명 규칙과 충돌이 발생할 수 있으니 사용하지 마세요.
> config 파일에서 명명된 프로파일을 구성하는 경우에만 profile_ 접두사를 포함합니다.

```
[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**: 조직의 IAM Identity Center 사용자 포털을 가리키는 URL입니다.
- **sso_region**: IAM Identity Center 포털 호스트가 포함된 AWS 리전입니다. 이는 나중에 기본 region 파라미터에 지정된 AWS 리전과 다를 수 있습니다.
- **sso_account_id**:이 IAM Identity Center 사용자에게 부여하려는 권한이 있는 IAM 역할이 포함된 AWS 계정 ID입니다.
- **sso_role_name**: IAM Identity Center를 통해 보안 인증 정보를 받기 위해 이 프로필을 사용할 때 사용자 권한을 정의하는 IAM 역할의 이름입니다.
- **region**:이 IAM Identity Center 사용자가 로그인하는 기본 AWS 리전입니다.

> **Note**
>
> aws configure sso 명령을 실행 AWS CLI 하여에 IAM Identity Center 지원 프로필을 추가할 수도 있습니다. 이 명령을 실행한 후 IAM Identity Center 디렉터리를 호스팅하는 IAM Identity Center 시작 URL(sso_start_url) 및 AWS 리전(region)의 값을 제공합니다.

자세한 내용은 AWS Command Line Interface 사용 설명서의 AWS Single Sign-On을 사용하도록 AWS CLI 구성을 참조하세요.

IAM Identity Center로 로그인

IAM Identity Center 프로필로 로그인하면 기본 브라우저가 사용자 `credential file`에서 지정한 `sso_start_url`로 실행됩니다. 에서 AWS 리소스에 액세스하려면 먼저 IAM Identity Center 로그인을 확인해야 합니다 AWS Toolkit for Visual Studio. 보안 인증 정보가 만료되면 연결 프로세스를 반복하여 새 임시 보안 인증 정보를 받아야 합니다.

AWS IAM 자격 증명

AWS IAM 자격 증명은 로컬에 저장된 액세스 키를 통해 AWS 계정으로 인증됩니다.

다음 섹션에서는에서 AWS 계정으로 인증하도록 IAM 자격 증명을 설정하는 방법을 설명합니다 AWS Toolkit for Visual Studio.

Important

AWS 계정으로 인증하도록 IAM 자격 증명을 설정하기 전에 다음 사항에 유의하세요.

- 다른 AWS 서비스(예: AWS CLI)를 통해 IAM 자격 증명을 이미 설정한 경우는 해당 자격 증명을 AWS Toolkit for Visual Studio 자동으로 감지합니다.
- AWS 에서는 AWS IAM Identity Center 인증을 사용할 것을 권장합니다. AWS IAM 모범 사례에 대한 자세한 내용은 AWS Identity and Access Management 사용 설명서의 IAM의 보안 모범 사례 섹션을 참조하세요.
- 보안 위험을 방지하려면 목적별 소프트웨어를 개발하거나 실제 데이터로 작업할 때 IAM 사용자를 인증에 사용하지 마세요. 대신와 같은 자격 증명 공급자와의 페더레이션을 사용합니다 AWS IAM Identity Center. 자세한 정보는 AWS IAM Identity Center 사용 설명서의 IAM Identity Center란 무엇인가요? 섹션을 참조하세요.

IAM 사용자 생성

AWS 계정으로 인증 AWS Toolkit for Visual Studio 하도록를 설정하려면 먼저 SDK 및 도구 참조 안내서의 장기 자격 증명을 사용하여 인증 주제에서 1단계: IAM 사용자 생성 및 2단계: 액세스 키 가져오기를 완료해야 합니다. AWS SDKs

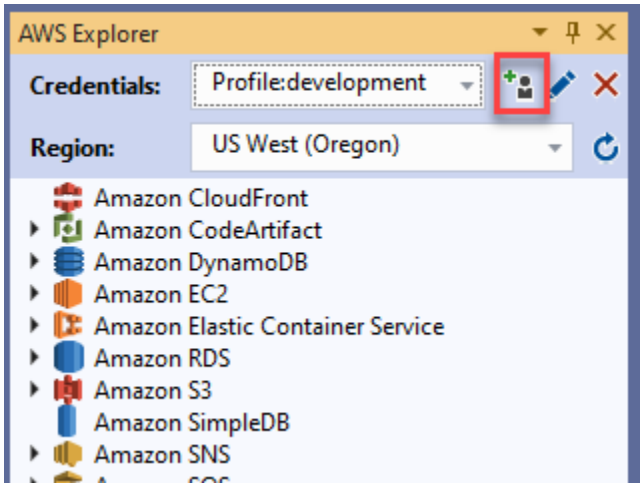
> **Note**
>
> 3단계: 공유 보안 인증 정보 업데이트는 선택 사항입니다.
> 3단계를 완료하면가에서 자격 증명을 AWS Toolkit for Visual Studio 자동으로 감지합니다.credentials file
> 3단계를 완료하지 않은 경우 아래에 있는 섹션에서 자격 증명 파일 생성에 설명된 credentials file 대로를 생성하는 프로세스를 AWS Toolkit for Visual Studio 안내합니다.
> AWS Toolkit for Visual Studio

보안 인증 정보 파일 생성

AWS Toolkit for Visual Studio에 사용자를 추가하거나 credentials file을 생성하려면 다음을 수행하세요.

> **Note**
>
> 툴킷에서 새 사용자 프로필을 추가하는 경우 다음을 수행하세요.
>
> - credentials file이 이미 존재하는 경우 새 사용자 정보가 기존 파일에 추가됩니다.
> - credentials file이 존재하지 않는 경우 새 파일이 생성됩니다.

1. AWS 탐색기에서 새 계정 프로필 아이콘을 선택하여 새 계정 프로필 대화 상자를 엽니다.



2. 새 계정 프로필 대화 상자의 필수 필드를 작성하고 확인 버튼을 선택하여 IAM 사용자를 생성하세요.

툴킷에서 IAM 사용자 보안 인증 정보 편집

툴킷에서 IAM 사용자 보안 인증 정보을 편집하려면 다음 단계를 완료하세요.

1. AWS 탐색기의 자격 증명 드롭다운에서 편집하려는 IAM 사용자 자격 증명을 선택합니다.
2. 프로필 편집 아이콘을 선택하여 프로필 편집 대화 상자를 엽니다.
3. 프로필 편집 대화 상자에서 업데이트를 완료하고 확인 버튼을 선택하여 변경 사항을 저장하세요.

툴킷에서 IAM 사용자 보안 인증 정보를 삭제하려면 다음 단계를 완료하세요.

1. AWS 탐색기의 자격 증명 드롭다운에서 삭제할 IAM 사용자 자격 증명을 선택합니다.
2. 프로필 삭제 아이콘을 선택하여 프로필 삭제 프롬프트를 엽니다.
3. 프로필을 삭제하여 `Credentials file`에서 제거할지 확인합니다.

> **Important**
>
> 프로필 편집 대화 상자의 IAM Identity Center 또는 다중 인증(MFA)과 같은 고급 액세스 기능을 지원하는 프로필은 AWS Toolkit for Visual Studio에서 편집할 수 없습니다. 이러한 유형의 프로필을 변경하려면 텍스트 편집기를 사용하여 `credentials file`을 편집해야 합니다.

텍스트 편집기에서 IAM 사용자 보안 인증 정보 편집

를 사용하여 IAM 사용자를 관리하는 것 외에도 원하는 텍스트 편집기`credential files`에서 편집할 AWS Toolkit for Visual Studio수 있습니다. Windows에서 `credential file`의 기본 위치는 `C:\Users\`*`USERNAME`*`\.aws\credentials`입니다.

`credential files`의 위치 및 구조에 대한 자세한 내용은 AWS SDK 및 도구 참조 가이드의 공유 구성 및 보안 인증 정보 파일 섹션을 참조하세요.

AWS Command Line Interface (AWS CLI)에서 IAM 사용자 생성

AWS CLI 는 명령을 `credentials file`사용하여에서 IAM 사용자를 생성하는 데 사용할 수 있는 또 다른 도구입니다`aws configure`.

에서 IAM 사용자를 생성하는 방법에 대한 자세한 내용은 AWS CLI 사용 설명서의 주제 구성을 AWS CLI AWS CLI 참조하세요.

Toolkit for Visual Studio는 다음과 같은 구성 속성을 지원합니다.

```
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 빌더 ID

AWS Builder ID는 Amazon CodeCatalyst를 사용하여 타사 리포지토리를 복제하는 등 특정 서비스 또는 기능을 사용하는 데 필요할 수 있는 추가 AWS 인증 방법입니다.

AWS Builder ID 인증 방법에 대한 자세한 내용은 로그인 사용 설명서의 AWS Builder ID로 AWS 로그인 주제를 참조하세요.

CodeCatalyst용 리포지토리를 복제하는 방법에 대한 자세한 내용은이 사용 설명서의 Amazon CodeCatalyst 작업 주제를 AWS Toolkit for Visual Studio참조하세요.

Toolkit for Visual Studio의 다중 인증(MFA)

다중 인증(MFA)은 AWS 계정에 대한 추가 보안입니다. MFA를 사용하려면 사용자가 AWS 웹 사이트 또는 서비스에 액세스할 때 AWS 지원되는 MFA 메커니즘에서 로그인 자격 증명과 고유한 인증을 제공해야 합니다.

AWS 는 MFA 인증을 위해 다양한 가상 및 하드웨어 디바이스를 지원합니다. 다음은 스마트폰 애플리케이션을 통해 활성화된 가상 MFA 디바이스의 예제입니다. MFA 디바이스 옵션에 대한 자세한 정보는 IAM 사용 설명서의 AWS에서 다중 인증(MFA) 사용을 참조하세요.

1단계: IAM 사용자에게 액세스 권한을 위임하기 위한 IAM 역할 생성

다음 절차에서는 IAM 사용자에게 권한을 할당하기 위한 역할 위임 설정 방법을 설명합니다. 역할 위임에 대한 자세한 정보는 AWS Identity and Access Management 사용 설명서의 IAM 사용자에게 권한을 위임하기 위한 역할 생성을 참조하세요.

1. https://console.aws.amazon.com/iam에서 IAM 콘솔로 이동하세요.
2. 탐색 창에서 역할을 선택한 후 역할 생성을 선택하세요.
3. 역할 생성 페이지에서 다른 AWS 계정을 선택하세요.
4. 필요한 계정 ID를 입력하고 MFA 필요 확인란을 선택하세요.

> **Note**
>
> 12자리 계정 번호(ID)를 찾으려면 콘솔의 탐색 표시줄에서 지원을 선택한 후 지원 센터를 선택하세요.

5. 다음: 권한을 선택합니다.
6. 기존 정책을 역할에 연결하거나 역할에 대한 새 정책을 생성하세요. 이 페이지에서 선택하는 정책에 따라 IAM 사용자가 도구 키트를 사용하여 액세스할 수 있는 AWS 서비스가 결정됩니다.
7. 정책을 연결한 후 역할에 IAM 태그를 추가하는 옵션으로 다음: 태그를 선택하세요. 계속하려면 다음: 검토를 선택하세요.
8. 검토 페이지에서 필수 역할 이름(예: toolkit-role)을 입력하세요. 역할 설명 옵션을 추가할 수도 있습니다.
9. 역할 생성을 선택합니다.
10. 확인 메시지(예: “toolkit-role 역할이 생성됨”)가 표시되면 메시지에서 역할 이름을 선택하세요.
11. 요약 페이지에서 복사 아이콘을 선택한 후 역할 ARN을 복사하여 파일에 붙여넣습니다. (이 ARN은 IAM 사용자가 역할을 위임하도록 구성할 때 필요합니다.)

2단계: 역할 권한을 위임하는 IAM 사용자 생성

이 단계에서는 권한이 없는 IAM 사용자를 생성하여 인라인 정책을 추가할 수 있습니다.

1. https://console.aws.amazon.com/iam에서 IAM 콘솔로 이동하세요.
2. 탐색 창에서 사용자 및 사용자 추가를 차례로 선택하세요.

3. 사용자 추가 페이지에서 필수 사용자 이름(예: toolkit-user)을 입력하고 프로그래밍 액세스 확인란을 선택합니다.
4. 다음: 권한, 다음: 태그, 다음: 검토를 선택하여 다음 페이지로 이동하세요. 사용자가 해당 역할의 권한을 위임하게 되므로 이 단계에서는 권한을 추가하지 않습니다.
5. 검토 페이지에 이 사용자는 권한이 없음 메시지가 표시됩니다. 사용자 생성을 선택합니다.
6. 성공 페이지에서 .csv 다운로드를 선택하여 액세스 키 ID 및 비밀 액세스 키가 포함된 파일을 다운로드하세요. (보안 인증 정보 파일에 사용자 프로필을 정의하는 경우에는 둘 다 필요합니다.)
7. 닫기를 선택하세요.

3단계: IAM 사용자가 역할을 위임할 수 있도록 허용하는 정책 추가

다음 절차에서는 사용자에게 역할 (및 해당 역할의 권한)을 위임하게 하는 인라인 정책을 생성합니다.

1. IAM 콘솔의 사용자 페이지에서 방금 생성한 IAM 사용자(예: toolkit-user)를 선택하세요.
2. 요약 페이지의 권한 탭에서 인라인 정책 추가를 선택하세요.
3. 정책 생성 페이지에서 서비스 선택을 선택하고 서비스 찾기에 STS를 입력한 다음 결과에서 STS를 선택합니다.
4. 작업에 Assumerole이라는 용어를 입력하기 시작합니다. AssumeRole 확인란이 나타나면 해당 확인란을 선택하세요.
5. 리소스 섹션에서 특정 항목이 선택되어 있는지 확인하고 ARN 추가를 클릭하여 액세스를 제한하세요.
6. ARN 추가 대화 상자의 역할에 대한 ARN 지정에 1단계에서 생성한 역할의 ARN을 추가하세요.

   역할의 ARN을 추가하면 해당 역할과 관련된 신뢰할 수 있는 계정 및 역할 이름이 계정 및 경로가 있는 역할 이름에 표시됩니다.
7. 추가를 선택합니다.
8. 정책 생성 페이지로 돌아가서 요청 조건 지정(선택 사항)을 선택하고 MFA 필요 확인란을 선택한 다음 닫기를 선택하여 확인합니다.
9. [정책 검토(Review policy)]를 선택합니다.
10. 정책 검토 페이지에서 정책의 이름을 입력한 후 정책 생성을 선택하세요.

    권한 탭에는 IAM 사용자에게 직접 연결된 새 인라인 정책이 표시됩니다.

4단계: IAM 사용자에 대한 가상 MFA 디바이스 관리

1. 스마트폰에 가상 MFA 애플리케이션을 다운로드하고 설치하세요.

   지원되는 애플리케이션 목록은 다단계 인증 리소스 페이지를 참조하세요.
2. IAM 콘솔의 탐색 표시줄에서 사용자를 선택한 다음 역할을 위임할 사용자(이 경우 toolkit-user)를 선택하세요.
3. 요약 페이지에서 보안 인증 정보 탭을 선택하고 할당된 MFA 디바이스에 대해 관리를 선택하세요다.
4. MFA 디바이스 관리 창에서 가상 MFA 디바이스를 선택한 후 계속을 선택하세요.
5. 가상 MFA 디바이스 설정 창에서 QR 코드 보기를 선택한 다음 스마트폰에 설치한 가상 MFA 애플리케이션을 사용하여 코드를 스캔하세요.
6. QR 코드를 스캔한 후 가상 MFA 애플리케이션은 일회용 MFA 코드를 생성합니다. MFA 코드 1과 MFA 코드 2에 연속된 두 개의 MFA 코드를 입력하세요.
7. Assign MFA(MFA 할당)을 선택합니다.
8. 사용자의 보안 인증 정보 탭으로 돌아가서 새 할당된 MFA 디바이스의 ARN을 복사하세요.

   ARN에는 12자리 계정 ID가 포함되며 형식은 다음과 비슷합니다.
   `arn:aws:iam::123456789012:mfa/toolkit-user` 다음 단계에서 MFA 프로필을 정의할 때 이 ARN을 사용합니다.

5단계: MFA를 허용하는 프로필 생성

다음 절차에서는 Toolkit for Visual Studio에서 AWS 서비스에 액세스할 때 MFA를 허용하는 프로파일을 생성합니다.

만든 프로필에는 이전 단계에서 복사하여 저장한 세 가지 정보가 포함됩니다.

- IAM 사용자의 액세스 키(액세스 키 ID 및 비밀 액세스 키)
- IAM 사용자에게 권한을 위임하는 역할의 ARN
- IAM 사용자에게 할당된 가상 MFA 디바이스의 ARN

자격 증명이 포함된 AWS 공유 AWS 자격 증명 파일 또는 SDK 스토어에서 다음 항목을 추가합니다.

```
[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
```

제공된 예제에는 두 개의 프로필이 정의되어 있습니다.

- [toolkit-user] 프로필에는 2단계에서 IAM 사용자를 생성할 때 생성되어 저장된 액세스 키와 비밀 액세스 키가 포함됩니다.
- [mfa] 프로필은 다중 인증이 지원되는 방식을 정의합니다. 다음과 같은 세 가지 항목이 있습니다.

  ◦ source_profile: 이 프로필에서 role_arn 설정에 지정된 역할을 위임하는 데 사용되는 보안 인증 정보가 있는 프로필을 지정합니다. 이 경우에는 toolkit-user 프로필입니다.

  ◦role_arn: 이 프로필을 사용하여 요청된 작업을 수행하는 데 사용할 IAM 역할의 Amazon 리소스 이름(ARN)을 지정합니다. 이 경우에는 1단계에서 생성한 역할의 ARN입니다.

  ◦mfa_serial: 사용자가 역할을 위임할 때 사용해야 하는 MFA 디바이스의 ID 또는 일련 번호를 지정합니다. 이 경우에는 3단계에서 설정한 가상 디바이스의 ARN입니다.

외부 보안 인증 정보 설정

AWS에서 직접 지원되지 않는 보안 인증 정보를 생성 또는 조회할 수 있는 방법이 있는 경우에는 credential_process이 포함된 프로필을 공유 보안 인증 정보 파일에 추가할 수 있습니다. 이 설정은 사용할 인증 보안 인증 정보를 생성 또는 검색하기 위해 실행하는 외부 명령을 지정합니다. 예를 들어 config 파일에 다음과 유사한 항목을 포함시킬 수 있습니다.

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

외부 보안 인증 정보 사용 및 관련 보안 위험에 대한 자세한 정보는 AWS Command Line Interface 사용 설명서의 외부 프로세스를 통한 보안 인증 정보 소싱을 참조하세요.

액세스를 허용하도록 방화벽 및 게이트웨이 업데이트

웹 콘텐츠 필터링 솔루션을 사용하여 특정 AWS 도메인 또는 URL 엔드포인트에 대한 액세스를 필터링하는 경우 AWS Toolkit for Visual Studio 및 Amazon Q를 통해 사용할 수 있는 모든 서비스와 기능에

액세스하려면 다음 엔드포인트를 허용해야 합니다. Amazon Q를 사용하는 AWS 도구 키트의 방화벽 및 프록시 설정 문제를 해결하는 방법에 대한 자세한 단계는이 사용 설명서의 문제 해결 주제의 방화벽 및 프록시 설정 섹션을 참조하세요. Amazon Q용 회사 프록시 구성에 대한 자세한 내용은 Amazon Q Developer 사용 설명서의 Amazon Q에서 회사 프록시 구성 주제를 참조하세요.

AWS Toolkit for Visual Studio 엔드포인트

다음은 허용해야 하는 AWS Toolkit for Visual Studio 특정 엔드포인트 및 참조 목록입니다.

엔드포인트

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

Amazon Q 플러그인 엔드포인트

다음은 허용해야 하는 Amazon Q 플러그인별 엔드포인트 및 참조 목록입니다.

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

Amazon Q Developer 엔드포인트

다음은 허용해야 하는 Amazon Q Developer별 엔드포인트 및 참조 목록입니다.

```
https://codewhisperer.us-east-1.amazonaws.com (Inline,Chat, QSDA,...)
https://q.us-east-1.amazonaws.com (Inline,Chat, QSDA....)
https://desktop-release.codewhisperer.us-east-1.amazonaws.com/ (Download URL for CLI.)
```

```
https://specs.q.us-east-1.amazonaws.com (URL for auto-complete specs used by CLI)
* aws-language-servers.us-east-1.amazonaws.com (Local Workspace context)
```

Amazon Q 코드 변환 엔드포인트

다음은 허용해야 하는 Amazon Q 코드 변환별 엔드포인트 및 참조 목록입니다.

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

인증 엔드포인트

다음은 허용해야 하는 인증 엔드포인트 및 참조 목록입니다.

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

자격 증명 엔드포인트

다음 목록에는 AWS IAM Identity Center 및 AWS Builder ID와 같이 자격 증명과 관련된 엔드포인트가 포함되어 있습니다.

AWS IAM Identity Center

IAM Identity Center에 필요한 엔드포인트에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 IAM Identity Center 활성화 주제를 참조하세요.

엔터프라이즈 IAM Identity Center

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

AWS 빌더 ID

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

원격 측정

다음은 허용해야 하는 원격 측정별 엔드포인트입니다.

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

참조

다음은 엔드포인트 참조 목록입니다.

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

```
assets.sso-portal.eu-west-1.amazonaws.com
oidc.eu-west-1.amazonaws.com
aws-toolkit-language-servers.amazonaws.com
aws-language-servers.us-east-1.amazonaws.com
idetoolkits.amazonwebservices.com
```

AWS 서비스 작업

다음 주제에서는 AWS Toolkit for Visual Studio with Amazon Q에서 AWS 서비스 작업을 시작하는 방법을 설명합니다.

주제

Amazon Q를 사용하는 AWS Toolkit for Visual Studio용 Amazon CodeCatalyst

Amazon CodeCatalyst란?

Amazon CodeCatalyst는 소프트웨어 개발 팀을 위한 클라우드 기반 협업 공간입니다. Amazon Q와 함께 AWS Toolkit for Visual Studio를 사용하면 Amazon Q와 함께 AWS Toolkit for Visual Studio에서

직접 CodeCatalyst 리소스를 보고 관리할 수 있습니다. CodeCatalyst에 대한 자세한 내용은 Amazon CodeCatalyst 사용 설명서를 참조하세요.

다음 주제에서는 AWS Toolkit for Visual Studio를 CodeCatalyst의 Amazon Q와 연결하는 방법과 AWS Toolkit for Visual Studio와 Amazon Q를 통해 CodeCatalyst로 작업하는 방법을 설명합니다.

주제

Amazon Q를 사용하여 Amazon CodeCatalyst 및 AWS Toolkit for Visual Studio 시작하기

AWS Toolkit for Visual Studio with Amazon Q에서 Amazon CodeCatalyst 작업을 시작하려면 다음을 완료하세요.

주제

Amazon Q를 사용하여 AWS Toolkit for Visual Studio 설치

AWS Toolkit for Visual Studio를 Amazon Q와 CodeCatalyst 계정과 통합하기 전에 Amazon Q와 함께 최신 버전의 AWS Toolkit for Visual Studio를 사용하고 있는지 확인합니다. Amazon Q와 함께 최신 버전의 AWS Toolkit for Visual Studio를 설치하고 설정하는 방법에 대한 자세한 내용은이 사용 설명서의 Amazon Q와 함께 AWS Toolkit for Visual Studio 설정 섹션을 참조하세요.

CodeCatalyst 계정 및 AWS Builder ID 생성

Amazon Q와 함께 최신 버전의 AWS Toolkit for Visual Studio를 설치하는 것 외에도 Amazon Q와 함께 AWS Toolkit for Visual Studio에 연결하려면 활성 AWS Builder ID 및 CodeCatalyst 계정이 있어야 합니다. 활성 AWS Builder ID 또는 CodeCatalyst 계정이 없는 경우 CodeCatalyst 사용 설명서의 CodeCatalyst로 설정 섹션을 참조하세요. CodeCatalyst

> **Note**
>
> AWS Builder ID는 자격 AWS 증명과 다릅니다. AWS Builder ID를 사용하여 가입하고 인증하는 방법에 대한 지침은이 사용 설명서의 Authentication and access: AWS Builder ID 주제를 참조하세요.
>
> AWS Builder IDs 대한 자세한 내용은 AWS 일반 참조 사용 설명서의 AWS Builder ID 주제를 참조하세요.

CodeCatalyst를 사용하여 Amazon Q와 AWS Toolkit for Visual Studio 연결

AWS Toolkit for Visual Studio를 CodeCatalyst 계정으로 Amazon Q와 연결하려면 다음 단계를 완료하세요.

1. Visual Studio의 Git 메뉴 항목에서 리포지토리 복제...를 선택하세요.
2. 리포지토리 찾아보기 섹션에서 Amazon CodeCatalyst를 공급자로 선택하세요.
3. 연결 섹션에서 AWS Builder ID로 연결을 선택하여 원하는 웹 브라우저에서 CodeCatalyst 콘솔을 엽니다.
4. 브라우저에서 제공된 필드에 AWS Builder ID를 입력하고 지침에 따라 계속합니다.
5. 메시지가 표시되면 허용을 선택하여 AWS Toolkit for Visual Studio와 Amazon Q 및 CodeCatalyst 계정 간의 연결을 확인합니다. 연결 프로세스가 완료되면 브라우저를 닫아도 안전하다는 확인 메시지가 표시됩니다.

Amazon Q를 사용하여 AWS Toolkit for Visual Studio에서 Amazon CodeCatalyst 리소스 작업

다음 섹션에서는 AWS Toolkit for Visual Studio with Amazon Q에 사용할 수 있는 Amazon CodeCatalyst 리소스 관리 기능에 대한 개요를 제공합니다.

주제

- 리포지토리 복제

리포지토리 복제

CodeCatalyst는 CodeCatalyst 프로젝트에서 작업하려면 클라우드에 연결되어 있어야 하는 클라우드 기반 서비스입니다. 로컬에서 프로젝트를 작업하려면 CodeCatalyst 리포지토리를 로컬 시스템에 복제하고 다음에 클라우드에 연결할 때 CodeCatalyst 프로젝트와 동기화할 수 있습니다.

리포지토리를 로컬 시스템에 복제하려면 다음 단계를 완료하세요.

1. Visual Studio의 Git 메뉴 항목에서 리포지토리 복제...를 선택하세요.
2. 리포지토리 찾아보기 섹션에서 Amazon CodeCatalyst를 공급자로 선택하세요.

> **Note**
>
> 연결 섹션에 `Not Connected` 메시지가 표시되면 계속하기 전에이 사용 설명서의 인증 및 액세스: AWS 빌더 ID 섹션의 단계를 완료합니다.

3. 리포지토리를 복제하려는 스페이스 및 프로젝트를 선택하세요.
4. 리포지토리 페이지에서 복제할 리포지토리를 선택하세요.
5. 경로 섹션에서 리포지토리를 복제할 폴더를 선택하세요.

> **Note**
>
> 복제에 성공하려면 처음에 이 폴더가 비어 있어야 합니다.

6. 복제를 선택하여 리포지토리 복제를 시작하세요.
7. 리포지토리가 복제되면 Visual Studio에서 복제된 솔루션을 로드하세요.

> **Note**
>
> Visual Studio가 복제된 리포지토리에서 솔루션을 열지 않는 경우 소스 컨트롤 메뉴의 Git 글로벌 설정에 있는 Git 리포지토리를 열 때 솔루션을 자동으로 로드 설정에서 Visual Studio 옵션을 조정할 수 있습니다.

문제 해결

다음은 CodeCatalyst AWS Toolkit for Visual Studio에서 Amazon Q로 작업할 때 알려진 문제를 해결하기 위한 문제 해결 주제입니다.

주제

자격 증명

CodeCatalyst에서 git 기반 리포지토리를 복제하려고 할 때 보안 인증 정보를 요구하는 대화 상자가 표시되면 AWS CodeCommit 보안 인증 도우미가 전체적으로 구성되어 CodeCatalyst와 간섭이 발생할 수 있습니다. AWS CodeCommit 보안 인증 도우미에 대한 자세한 내용은 AWS CodeCommit 사용 설명서의 AWS CLI 보안 인증 도우미를 사용하여 Windows에서 CodeCommit 리포지토리에 대한 HTTPS 연결 단계 설정을 참조하세요. AWS CodeCommit

AWS CodeCommit 보안 인증 도우미가 CodeCommit URL만 처리하도록 제한하려면 다음 단계를 완료하세요.

1. `%userprofile%\.gitconfig`에서 글로벌 git 구성 파일을 엽니다.
2. 다음 섹션을 파일에 추가하세요.

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

3. 해당 섹션을 다음으로 변경하세요.

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

4. 변경 내용을 저장한 다음 단계를 완료하여 리포지토리를 복제하세요.

Visual Studio용 Amazon CloudWatch Logs 통합

AWS Toolkit for Visual Studio with Amazon Q에서 Amazon CloudWatch Logs 통합을 사용하면 IDE를 벗어나지 않고도 CloudWatch Logs 리소스를 모니터링, 저장 및 액세스할 수 있습니다. CloudWatch 서비스를 설정하고 CloudWatch Logs 작업을 수행하는 방법에 대해 자세히 알아보려면 다음 주제 중에서 선택하세요.

주제

Visual Studio용 CloudWatch Logs 통합 설정

Amazon CloudWatch Logs 통합을 AWS Toolkit with Amazon Q와 사용하려면 먼저 AWS 계정이 있어야 합니다. AWS 로그인 사이트에서 새 AWS 계정을 생성할 수 있습니다. AWS Toolkit with Amazon Q에서 사용할 수 있는 대부분의 CloudWatch Logs 기능은 활성 AWS 자격 증명을 사용하여 액세스할 수 있습니다. 특정 기능에 추가 구성이 필요한 경우 요구 사항은 CloudWatch Logs 작업 안내서의 관련 섹션에 포함되어 있습니다.

CloudWatch Logs 설정에 대한 추가 정보 및 옵션은 Amazon CloudWatch Logs 안내서의 설정 섹션을 참조하세요.

Visual Studio에서 CloudWatch Logs 작업

Amazon CloudWatch Logs 통합을 사용하면 Amazon Q를 사용하여 AWS Toolkit for Visual Studio에서 CloudWatch Logs를 모니터링, 저장 및 액세스할 수 있습니다. IDE를 벗어나지 않고도 CloudWatch Logs 기능에 액세스할 수 있으므로 CloudWatch Logs 개발 프로세스를 간소화하고 워크플로 중단을 줄여 효율성이 향상됩니다. 다음 주제에서는 CloudWatch Logs 통합의 기본 기능 및 함수를 사용하는 방법을 설명합니다.

주제

CloudWatch 로그 그룹

`log group`은 동일한 보존 기간, 모니터링 및 액세스 제어 설정을 공유하는 `log streams`의 그룹입니다. 하나의 로그 그룹이 가질 수 있는 로그 스트림의 수는 제한이 없습니다.

로그 그룹 보기

`View Log Groups` 기능은 CloudWatch 로그 그룹 탐색기에 로그 그룹 목록을 표시합니다.

로그 그룹 보기 기능에 액세스하고 CloudWatch 로그 그룹 탐색기를 열려면 다음 단계를 완료하세요.

1. AWS 탐색기에서 Amazon CloudWatch를 확장합니다.
2. 로그 그룹을 두 번 클릭하거나 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 열고 보기를 선택하여 CloudWatch 로그 그룹 탐색기를 엽니다.

> **Note**
>
> CloudWatch 로그 그룹 탐색기는 솔루션 탐색기와 동일한 창 위치에 열립니다.

로그 그룹 필터링

개인 계정에는 수천 개의 서로 다른 로그 그룹이 포함될 수 있습니다. 특정 그룹에 대한 검색을 단순화하려면 아래 설명된 `filtering` 기능을 사용하세요.

1. CloudWatch 로그 그룹 탐색기에서 창 상단에 있는 검색 창에 커서를 놓습니다.
2. 찾고 있는 로그 그룹과 관련된 접두사를 입력하기 시작합니다.
3. CloudWatch 로그 그룹 탐색기가 자동으로 업데이트되어 이전 단계에서 지정한 검색어와 일치하는 결과를 표시합니다.

로그 그룹 삭제

특정 로그 그룹을 삭제하려면 다음 절차를 참조하세요.

1. CloudWatch 로그 그룹 탐색기에서 삭제하려는 로그 그룹을 마우스 오른쪽 버튼으로 클릭하세요.
2. 메시지가 나타나면 현재 선택한 로그 그룹을 삭제할지를 확인합니다.
3. 예 버튼을 선택하면 선택한 로그 그룹이 삭제되고 CloudWatch 로그 그룹 탐색기가 새로 고쳐집니다.

로그 그룹 새로 고침

CloudWatch 로그 그룹 탐색기에 표시된 현재 로그 그룹 목록을 새로 고치려면 도구 모음에 있는 새로 고침 아이콘 버튼을 선택하세요.

로그 그룹 ARN 복사

특정 로그 그룹의 ARN을 복사하려면 아래 설명된 단계를 완료하세요.

1. CloudWatch 로그 그룹 탐색기에서 ARN을 복사할 로그 그룹을 마우스 오른쪽 버튼으로 클릭하세요.
2. 메뉴에서 ARN 복사 옵션을 선택하세요.
3. 이제 ARN이 로컬 클립보드에 복사되어 붙여넣을 준비가 되었습니다.

CloudWatch 로그 스트림

로그 스트림은 동일한 소스를 공유하는 로그 이벤트 시퀀스입니다.

> **Note**
>
> 로그 스트림을 볼 때는 다음 속성에 유의하세요.
>
> - 기본적으로 로그 스트림은 가장 최근의 이벤트 타임스탬프를 기준으로 정렬됩니다.
> - 열 헤더에 있는 캐럿을 토글하여 로그 스트림과 관련된 열을 오름차순 또는 내림차순으로 정렬할 수 있습니다.
> - 필터링된 항목은 로그 스트림 이름으로만 정렬할 수 있습니다.

로그 스트림 보기

1. CloudWatch 로그 그룹 탐색기에서 로그 그룹을 두 번 클릭하거나 로그 그룹을 마우스 오른쪽 버튼으로 클릭한 다음 컨텍스트 메뉴에서 로그 스트림 보기를 선택하세요.
2. 문서 창에 로그 그룹과 관련된 로그 스트림 목록이 포함된 새 탭이 열립니다.

로그 스트림 필터링

1. 문서 창의 로그 스트림 탭에서 검색 창에 커서를 놓습니다.
2. 찾고 있는 로그 스트림과 관련된 접두사를 입력하기 시작합니다.
3. 입력하면 입력에 따라 현재 디스플레이가 자동으로 업데이트되어 로그 스트림을 필터링하세요.

로그 스트림 새로 고침

문서 창에 표시된 현재 로그 스트림 목록을 새로 고치려면 도구 모음의 검색 창 옆에 있는 새로 고침 아이콘 버튼을 선택하세요.

로그 스트림 복사 ARN

특정 로그 그룹의 ARN을 복사하려면 아래 설명된 단계를 완료하세요.

1. 로그 스트림 탭의 문서 창에서 ARN을 복사하려는 로그 스트림을 마우스 오른쪽 버튼으로 클릭하세요.
2. 메뉴에서 ARN 복사 옵션을 선택하세요.
3. 이제 ARN이 로컬 클립보드에 복사되어 붙여넣을 준비가 되었습니다.

로그 스트림 다운로드

로그 스트림 내보내기 기능은 선택한 로그 스트림을 로컬에 다운로드하고 저장하며, 추가 처리를 위해 사용자 지정 도구 및 소프트웨어로 액세스할 수 있습니다.

1. 로그 스트림 탭의 문서 창에서 다운로드하려는 로그 스트림을 마우스 오른쪽 버튼으로 클릭하세요.
2. 로그 스트림 내보내기를 선택하여 텍스트 파일로 내보내기 대화 상자를 엽니다.
3. 파일을 로컬에 저장할 위치를 선택하고 제공된 텍스트 필드에 이름을 지정하세요.
4. 확인을 선택하여 다운로드를 확인하세요. 다운로드 상태는 Visual Studio 작업 상태 센터에 표시됩니다.

CloudWatch 로그 이벤트

로그 이벤트는 CloudWatch로 모니터링 중인 애플리케이션 또는 리소스에 의해 기록된 활동의 기록입니다.

로그 이벤트 작업

로그 이벤트는 테이블로 표시됩니다. 기본적으로 이벤트는 가장 오래된 이벤트부터 가장 최근 이벤트까지 정렬됩니다.

Visual Studio의 로그 이벤트와 관련된 작업은 다음과 같습니다.

- 줄 바꿈 텍스트 모드: 이벤트를 클릭하여 줄 바꿈된 텍스트를 전환할 수 있습니다.
- 텍스트 줄 바꿈 버튼: `document window` **`toolbar`**에 있는 이 버튼을 사용하면 모든 항목에 대해 텍스트 줄 바꿈을 켜거나 끌 수 있습니다.
- 클립보드에 메시지 복사: 복사하려는 메시지를 선택한 다음 선택 항목을 마우스 오른쪽 버튼으로 클릭하고 복사(키보드 단축키 `Ctrl + C`)를 선택하세요.

로그 이벤트 보기

1. 문서 창에서 로그 스트림 목록이 포함된 탭을 선택하세요.
2. 로그 스트림을 두 번 클릭하거나, 로그 스트림을 마우스 오른쪽 버튼으로 클릭한 다음 메뉴에서 로그 스트림 보기를 선택하세요.
3. 선택한 로그 스트림과 관련된 로그 이벤트 테이블이 포함된 새 로그 이벤트 탭이 문서 창에 열립니다.

로그 이벤트 필터링

로그 이벤트를 필터링하는 방법에는 내용 기준, 시간 범위 기준 또는 둘 함께 사용 등의 세 가지가 있습니다. 내용과 시간 범위를 기준으로 로그 이벤트를 필터링하려면 먼저 내용 또는 시간 범위를 기준으로 메시지를 필터링한 다음 다른 방법으로 결과를 필터링하세요.

내용 기준으로 로그 이벤트를 필터링하려면 다음을 수행하세요.

1. 문서 창의 로그 이벤트 탭에서 창 상단에 있는 검색 표시줄에 커서를 놓습니다.
2. 검색 중인 로그 이벤트와 관련된 용어 또는 문구를 입력하세요.
3. 입력하면 현재 디스플레이에서 자동으로 로그 이벤트를 필터링하기 시작합니다.

> **Note**
>
> 필터 패턴은 대/소문자를 구분합니다. 영숫자가 아닌 문자가 포함된 정확한 용어 및 구문을 큰따옴표 (*"""*) 로 묶으면 검색 결과를 개선할 수 있습니다. 필터 패턴에 대한 자세한 정보는 Amazon CloudWatch 안내서의 필터 및 패턴 구문 주제를 참조하세요.

특정 시간 범위 동안 생성된 로그 이벤트를 보려면 다음을 수행하세요.

1. 문서 창의 로그 이벤트 탭에서 도구 모음에 있는 캘린더 아이콘 버튼을 선택하세요.
2. 제공된 필드를 사용하여 검색하려는 시간 범위를 지정하세요.
3. 날짜 및 시간 제한을 지정하면 필터링된 결과가 자동으로 업데이트됩니다.

> **Note**
>
> 필터 지우기 옵션은 현재 날짜 및 시간 필터 선택 항목을 모두 지웁니다.

로그 이벤트 새로 고침

로그 이벤트 탭에 표시된 현재 로그 이벤트 목록을 새로 고치려면 도구 모음에 있는 새로 고침 아이콘 버튼을 선택하세요.

CloudWatch Logs에 대한 추가 액세스

Visual Studio의 AWS 도구 키트에서 직접 다른 AWS 서비스 및 리소스와 연결된 CloudWatch Logs에 액세스할 수 있습니다.

Lambda

Lambda 함수와 연결된 로그 스트림을 보려면 다음을 수행하세요.

> **Note**
>
> Lambda 실행 역할에는 CloudWatch Logs에 로그를 전송할 수 있는 적절한 권한이 있어야 합니다. CloudWatch Logs로 데이터를 전송하는 데 필요한 권한에 대한 자세한 정보는 https://docs.aws.amazon.com/lambda/latest/dg/monitoring-cloudwatchlogs.html#monitoring-cloudwatchlogs-prereqs 섹션을 참조하세요.

1. AWS Toolkit Explorer에서 Lambda를 확장합니다.
2. 보려는 함수를 마우스 오른쪽 버튼으로 클릭한 다음 로그 보기를 선택하여 문서 창에서 관련 로그 스트림을 엽니다.

Lambda 통합 `function view`를 사용하여 로그 스트림을 보려면 다음을 수행하세요.

1. AWS Toolkit Explorer에서 Lambda를 확장합니다.
2. 보려는 함수를 마우스 오른쪽 버튼으로 클릭한 다음 함수 보기를 선택하여 문서 창에서 함수 보기를 엽니다.
3. `function view`에서 Logs 탭으로 전환하면 선택한 Lambda 함수와 관련된 로그 스트림이 표시됩니다.

ECS

ECS 작업 컨테이너와 연결된 로그 리소스를 보려면 다음 절차를 완료하세요.

> **Note**
>
> Amazon ECS 서비스가 CloudWatch에 로그를 전송하려면 해당 Amazon ECS 작업의 각 컨테이너가 필수 구성을 충족해야 합니다. 필요한 설정 및 구성에 대한 자세한 내용은 AWS 로그 로그 드라이버 사용 설명서를 참조하세요.

1. AWS Toolkit Explorer에서 Amazon ECS를 확장합니다.
2. 보려는 Amazon ECS 클러스터를 선택하여 문서 창에서 새 ECS 클러스터 탭을 엽니다.
3. ECS 클러스터 탭의 왼쪽에 있는 탐색 메뉴에서 작업을 선택하여 클러스터와 관련된 모든 작업을 나열하세요.
4. 작업 디스플레이에서 작업을 선택하고 왼쪽 하단에 있는 로그 보기 링크를 선택하세요.

   > **Note**
   >
   > 이 디스플레이에는 클러스터에 포함된 모든 작업이 나열되며, `View Logs` 링크는 필수 로그 구성을 충족하는 각 작업에 대해서만 표시됩니다.
   >
   > - 작업이 단일 컨테이너에만 연결된 경우 로그 보기 링크를 클릭하면 해당 컨테이너의 로그 스트림이 열립니다.
   > - 작업이 여러 컨테이너와 연결된 경우 로그 보기 링크를 클릭하면 ECS 작업용 CloudWatch Logs 보기 대화 상자가 열리고 컨테이너: 드롭다운 메뉴를 사용하여 로그를 보려는 컨테이너를 선택한 다음 확인을 선택하세요.

5. 문서 창에 컨테이너 선택과 관련된 로그 스트림을 표시하는 새 탭이 열립니다.

Amazon EC2 인스턴스 관리

AWS Explorer는 Amazon Machine Image(AMI) 및 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에 대한 세부 보기를 제공합니다. 이러한 보기에서 AMI의 Amazon EC2 인스턴스를 시작하고, 해당 인스턴스에 연결하며, 인스턴스를 중지하거나 종료할 수 있습니다. 모두 Visual Studio 개발 환경 내에서 수행됩니다. 인스턴스 보기를 사용하여 인스턴스에서 AMI를 생성할 수 있습니다. 자세한 내용은 Amazon EC2 인스턴스에서 AMI 생성을 참조하십시오.

Amazon 머신 이미지 및 Amazon EC2 보기

AWS Explorer에서 Amazon Machine Image(AMIs) 및 Amazon EC2 인스턴스의 보기를 표시할 수 있습니다. AWS 탐색기에서 Amazon EC2 노드를 확장합니다.

첫 번째 하위 노드 AMI에서 AMI 보기를 표시하려면 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴를 연 다음 보기를 선택합니다.

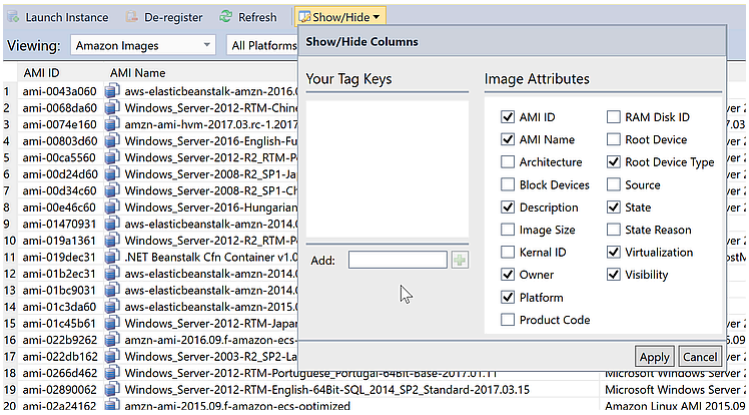
인스턴스 노드에서 Amazon EC2 인스턴스를 표시하려면 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴를 연 다음 보기를 선택합니다.

적합한 노드를 두 번 클릭하여 보기를 표시할 수도 있습니다.

- 뷰의 범위는 AWS 탐색기에 지정된 리전(예: 미국 서부(캘리포니아 북부) 리전)으로 지정됩니다.
- 클릭하고 끌어 열을 다시 정렬할 수 있습니다. 열에서 값을 정렬하려면 열 제목을 클릭합니다.
- 보기에서 드롭다운 목록 및 필터 상자를 사용하여 보기를 구성할 수 있습니다. 초기 보기에는 AWS 탐색기에 지정된 계정이 소유한 모든 플랫폼 유형(Windows 또는 Linux)의 AMIs가 표시됩니다.

열 표시/숨기기

또한 보기 상단에서 표시/숨기기 드롭다운 목록을 선택하여 표시할 열을 구성할 수 있습니다. 보기를 닫고 다시 열어도 선택한 열이 그대로 유지됩니다.



AMI 및 인스턴스 보기의 열 표시/숨기기 UI

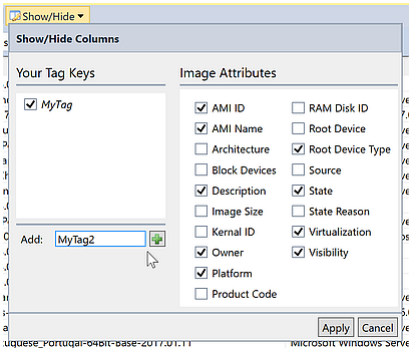
AMI, 인스턴스 및 볼륨 태그 지정

또한 표시/숨기기 드롭다운 목록을 사용하여 소유한 AMI, Amazon EC2 인스턴스 또는 볼륨에 태그를 추가할 수 있습니다. 태그는 AMI, 인스턴스 및 볼륨에 메타데이터를 연결해 주는 이름 값 쌍입니다. 태그 이름은 계정에 모두 지정되거나 AMI 및 인스턴스에 따로 지정됩니다. 예를 들어, AMI 및 인스턴스에 동일한 태그 이름을 사용해도 충돌되지 않습니다. 태그 이름은 대/소문자를 구분하지 않습니다.

태그에 대한 자세한 정보는 Linux 인스턴스용 Amazon EC2 사용 설명서의 태그 사용을 참조하세요.

태그를 추가하려면

1. 추가 상자에 태그 이름을 입력합니다. 더하기 기호(+)가 있는 녹색 버튼을 선택한 다음 적용을 선택합니다.



   AMI 또는 Amazon EC2 인스턴스에 태그 추가

   새 태그는 기울임꼴로 표시되며, 해당 태그와 연결된 값이 아직 없음을 나타냅니다.

   목록 보기에서 태그 이름이 새 열로 나타납니다. 하나 이상의 값이 태그와 연결되면 태그가 AWS Management Console 콘솔에 표시됩니다.
2. 태그에 값을 추가하려면 해당 태그 열에서 셀을 두 번 클릭하고 값을 입력합니다. 태그 값을 삭제하려면 셀을 두 번 클릭하고 텍스트를 삭제합니다.

   표시/숨기기 드롭다운 목록에서 태그를 선택 해제하면 해당 열이 보기에서 사라집니다. 태그는 AMI, 인스턴스 또는 볼륨과 연결된 태그 값과 함께 유지됩니다.

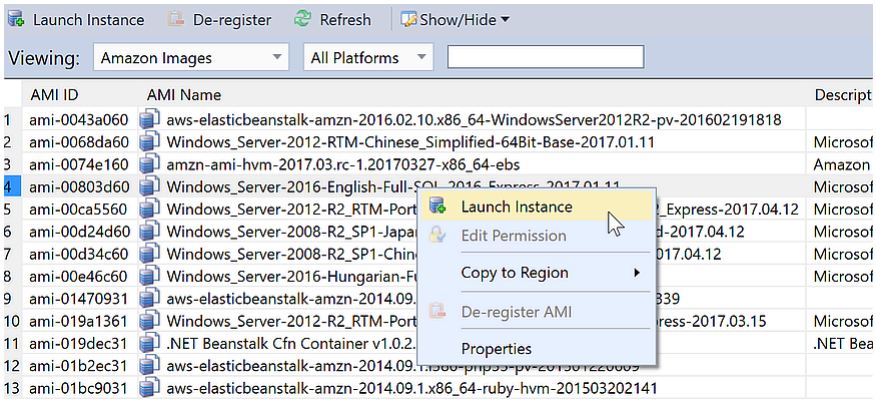
> **Note**
>
> 연결된 값이 없는 표시/숨기기 드롭다운 목록에서 태그를 지우면 AWS 도구 키트가 태그를 완전히 삭제합니다. 더 이상 목록 보기 또는 표시/숨기기 드롭다운 목록에 나타나지 않습니다. 해당 태그를 다시 사용하려면 표시/숨기기 대화 상자를 사용하여 태그를 다시 생성하십시오.

Amazon EC2 인스턴스 실행

AWS Explorer는 Amazon EC2 인스턴스를 시작하는 데 필요한 모든 기능을 제공합니다. 이 섹션에서는 Amazon 머신 이미지(AMI)를 선택하고 구성한 다음 Amazon EC2 인스턴스로 시작해 보겠습니다.

Windows Server Amazon EC2 인스턴스를 실행하려면

1. AMI 보기 상단에 있는 왼쪽 드롭다운 목록에서 Amazon Images(Amazon 이미지)를 선택합니다. 오른쪽 드롭다운 목록에서 Windows를 선택합니다. 필터 상자에서 Elastic Block Storage를 나타내는 ebs를 입력합니다. 보기를 새로 고치는 데 다소 시간이 걸릴 수 있습니다.
2. 목록에서 AMI를 선택하고 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴를 연 다음 인스턴스 실행를 선택합니다.



AMI 목록

3. Launch New Amazon EC2 Instance(새 Amazon EC2 인스턴스 시작) 대화 상자에서 애플리케이션에 대한 AMI를 구성합니다.

   인스턴스 유형

   실행할 EC2 인스턴스 유형을 선택합니다. EC2 요금 페이지에서 인스턴스 유형의 목록과 요금 정보를 확인할 수 있습니다.

   이름

   인스턴스 이름을 입력합니다. 이 이름은 256자를 초과할 수 없습니다.

   키 페어

   키 페어는 RDP(Remote Desktop Protocol)를 사용하는 EC2 인스턴스에 로그인하는 데 필요한 Windows 암호를 얻는 데 사용됩니다. 프라이빗 키 액세스 권한이 있는 키 페어를 선택하거나 키 페어를 생성하는 옵션을 선택합니다. 도구 키트에서 키 페어를 만든 경우 도구 키트가 프라이빗 키를 저장할 수 있습니다.

   ToolKit에 저장된 키 페어는 암호화됩니다. 암호화된 키는 `%LOCALAPPDATA%\AWSToolkit\keypairs`(일반적으로 `C:\Users\<user>\AppData\Local\AWSToolkit\keypairs`)에 있습니다. 암호화된 키 페어를 `.pem` 파일로 내보낼 수 있습니다.

   a. Visual Studio에서 보기를 선택하고 AWS 탐색기를 클릭하세요.

b. Amazon EC2를 클릭하고 Key Pairs(키 페어)를 선택합니다.

c. 키 페어가 나열되고 도구 키트가 생성하고 관리하는 항목은 Stored in AWSToolkit(AWSToolkit에 저장됨)으로 표시됩니다.

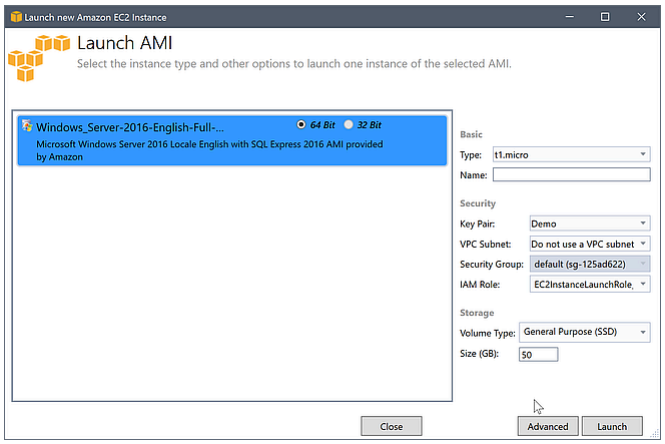
d. 생성된 키 페어를 마우스 오른쪽 버튼으로 클릭하고 Export Private Key(프라이빗 키 내보내기)를 선택합니다. 프라이빗 키는 암호화되지 않고 지정한 위치에 저장됩니다.

보안 그룹

보안 그룹은 EC2 인스턴스가 허용하는 네트워크 트래픽의 유형을 제어합니다. 포트 3389(RDP에서 사용하는 포트)에서 수신 트래픽을 허용하는 보안 그룹을 선택하면 EC2 인스턴스에 연결할 수 있습니다. 도구 키트를 사용하여 보안 그룹을 생성하는 방법에 대한 자세한 내용은 AWS 탐색기에서 보안 그룹 관리를 참조하세요.

인스턴스 프로파일

인스턴스 프로파일은 IAM 역할에 대한 논리 컨테이너입니다. 인스턴스 프로파일을 선택하면 해당하는 IAM 역할과 EC2 인스턴스를 연결합니다. IAM 역할은 Amazon Web Services 및 계정 리소스에 대한 액세스를 지정하는 정책으로 구성됩니다. EC2 인스턴스가 IAM 역할과 연결되면 인스턴스에서 실행 중인 애플리케이션 소프트웨어가 IAM 역할에서 지정한 권한으로 실행됩니다. 이렇게 하면 자체 자격 AWS 증명을 지정하지 않고도 애플리케이션 소프트웨어를 실행할 수 있으므로 소프트웨어가 더 안전해집니다. IAM 역할에 대한 자세한 내용은 IAM 사용 설명서를 참조하십시오.

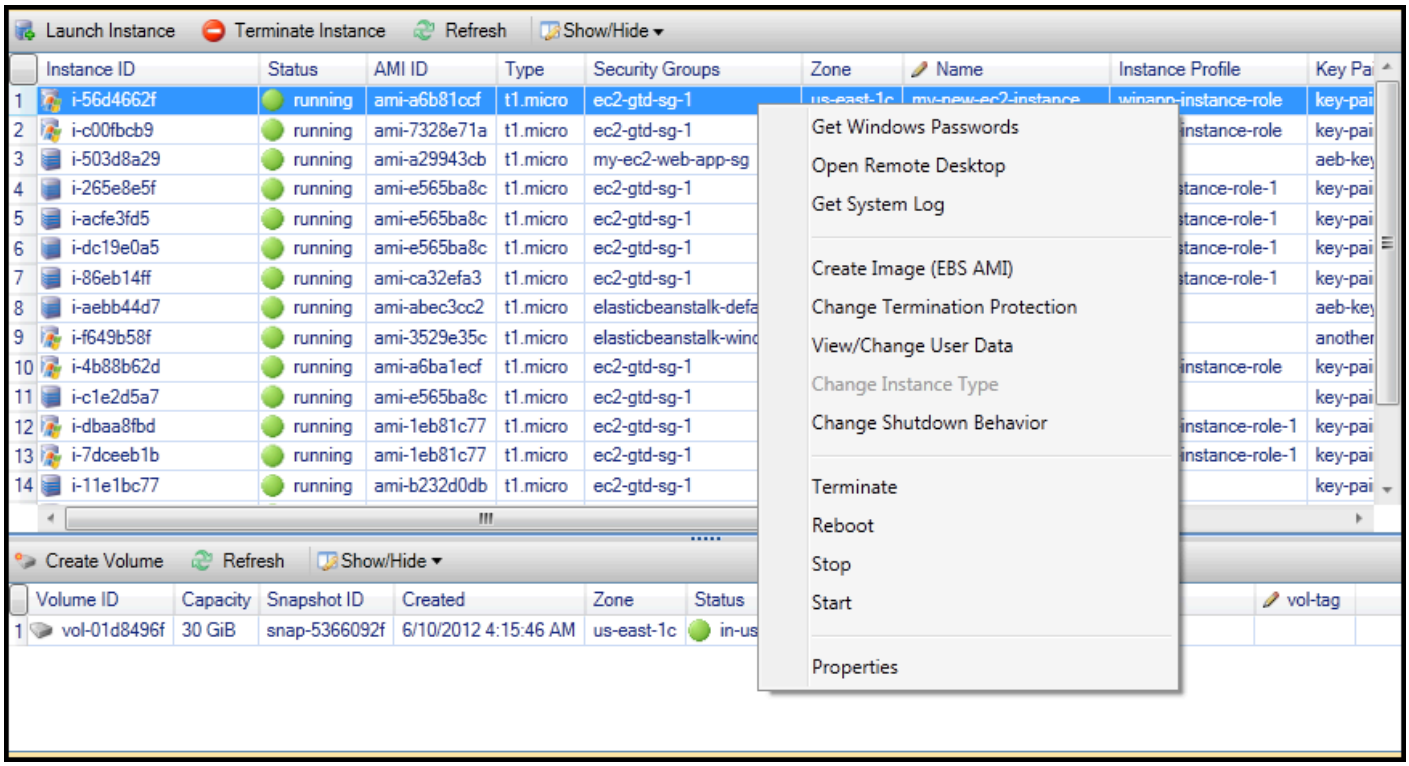


EC2 AMI 시작 대화 상자

4. 시작을 선택합니다.

AWS 탐색기의 Amazon EC2 인스턴스 하위 노드에서 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴를 연 다음 보기를 선택합니다. AWS 도구 키트에는 활성 계정과 연결된 Amazon EC2 인스턴스 목록

이 표시됩니다. 새 인스턴스를 보려면 새로 고침을 선택해야 합니다. 인스턴스가 먼저 나타나면 대기 중 상태일 수도 있지만 시간이 지나면 실행 중 상태로 전환됩니다.



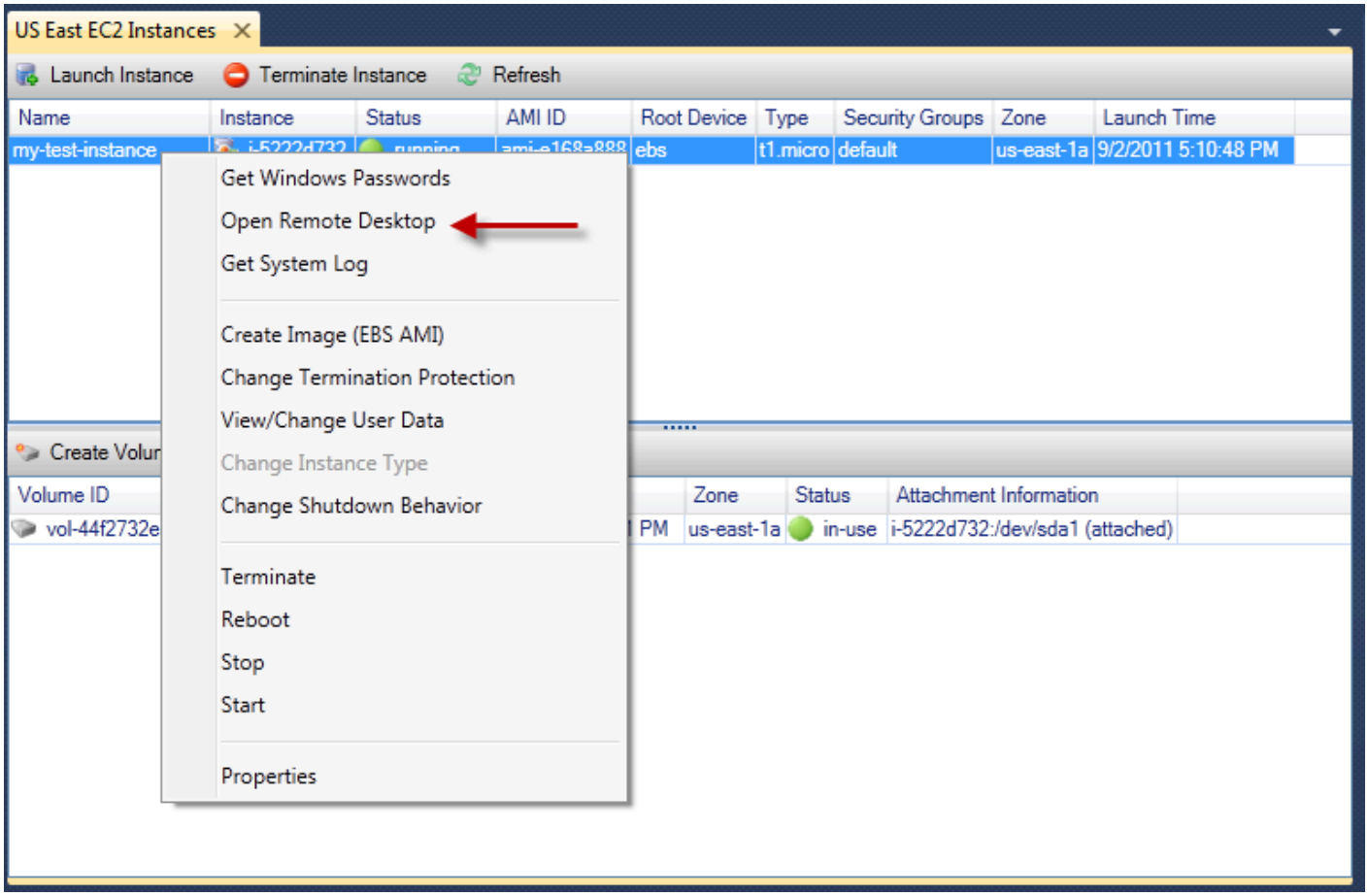
Amazon EC2 인스턴스에 연결

Windows Remote Desktop을 사용하여 Windows Server 인스턴스에 연결할 수 있습니다. 인증을 위해 AWS 도구 키트를 사용하면 인스턴스의 관리자 암호를 검색하거나 인스턴스와 연결된 저장된 키 페어를 사용할 수 있습니다. 다음 절차에서는 저장된 키 페어를 사용하겠습니다.

Windows Remote Desktop을 사용하여 Windows Server 인스턴스에 연결하려면

1. EC2 인스턴스 목록에서 연결하려는 Windows Server 인스턴스를 마우스 오른쪽 버튼으로 클릭합니다. 컨텍스트 메뉴에서 Open Remote Desktop(원격 데스크톱 열기)을 선택합니다.

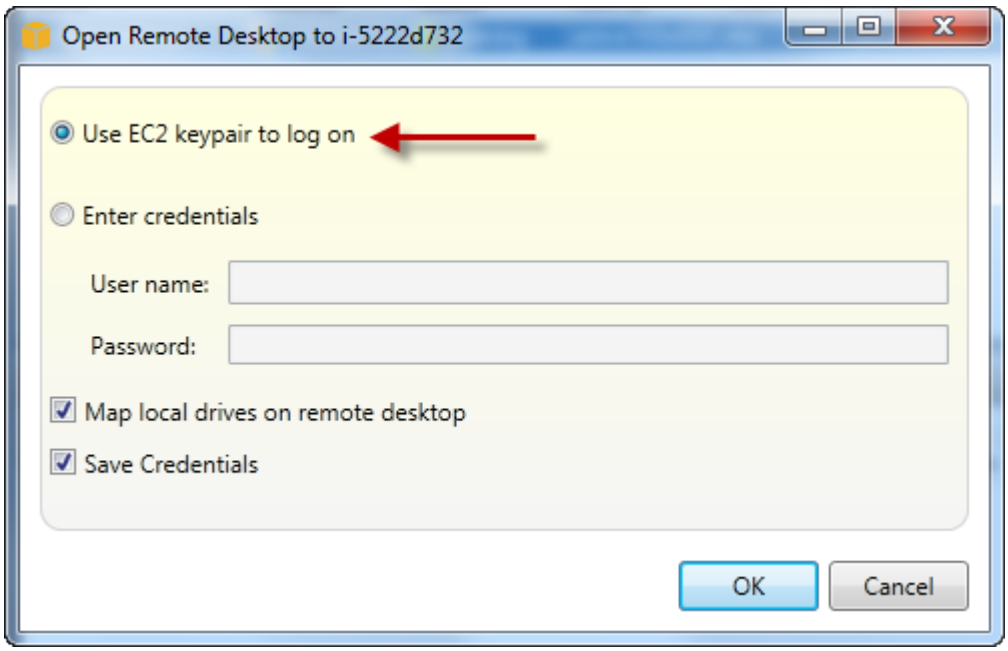
   관리자 암호를 사용하여 인증하려면 Get Windows Passwords(Windows 암호 가져오기)를 선택합니다.



EC2 인스턴스 컨텍스트 메뉴

2. Open Remote Desktop(원격 데스크톱 열기) 대화 상자에서 Use EC2 keypair to log on(EC2 키 페어를 사용하여 로그인)을 선택한 다음 확인을 선택합니다.

   AWS 도구 키트로 키 페어를 저장하지 않은 경우 프라이빗 키가 포함된 PEM 파일을 지정합니다.

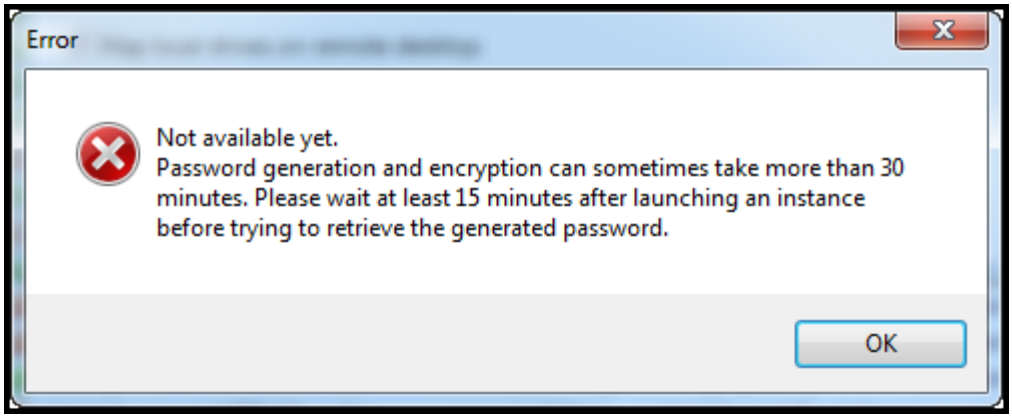


Open Remote Desktop(원격 데스크톱 열기) 대화 상자

3. Remote Desktop(원격 데스크톱) 창이 열립니다. 키 페어로 인증이 이루어지므로 로그인할 필요가 없습니다. Amazon EC2 인스턴스에서 관리자로 실행 중입니다.

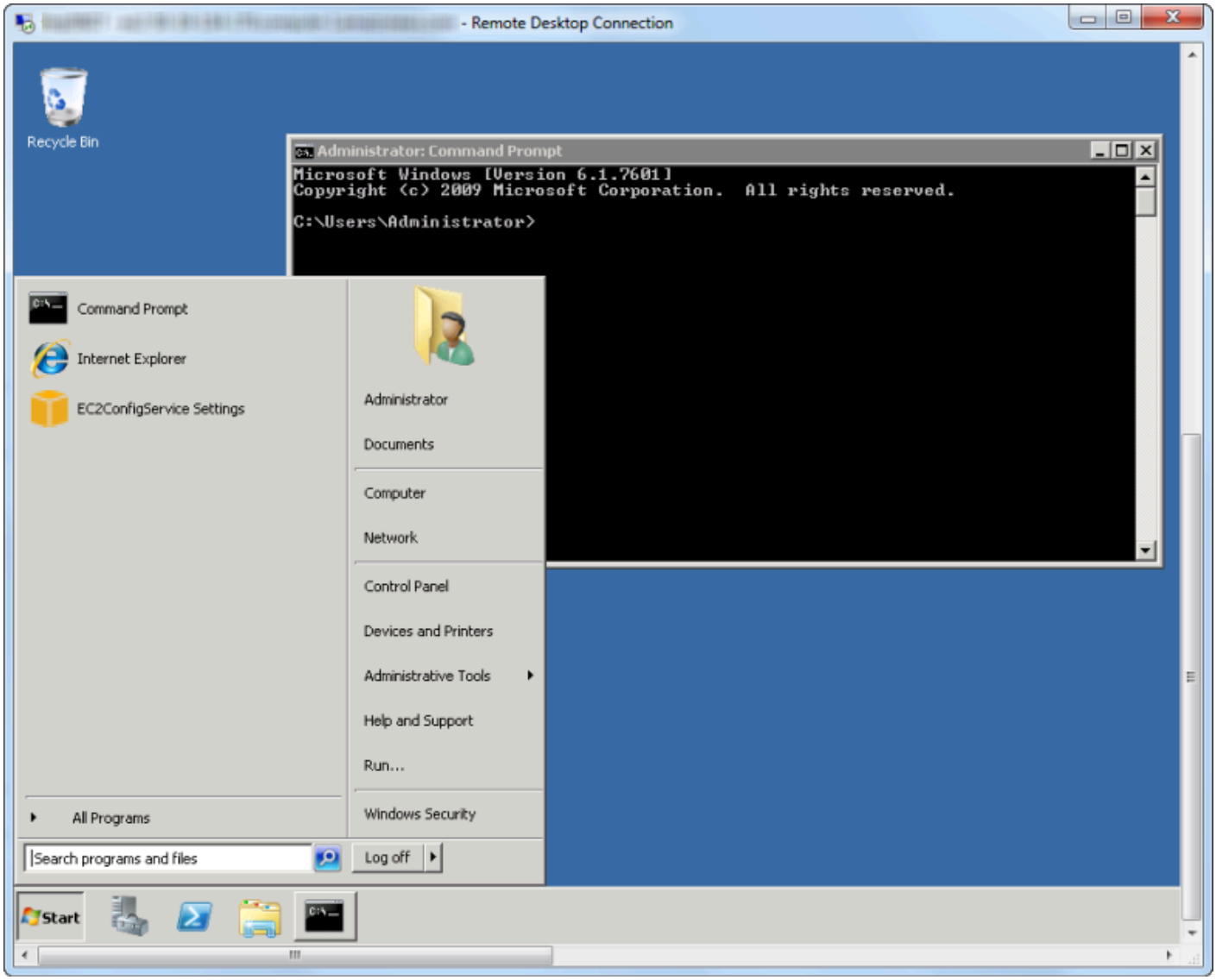
   EC2 인스턴스가 최근에 시작된 경우 두 가지 이유로 인해 연결하지 못할 수 있습니다.

   - 원격 데스크톱 서비스가 아직 실행되지 않았을 수 있습니다. 몇 분 기다린 후 다시 시도하십시오.
   - 암호 정보가 아직 인스턴스에 전송되지 않았을 수 있습니다. 이런 경우 다음과 유사한 메시지 상자가 표시됩니다.



암호를 아직 사용할 수 없음

다음 스크린샷은 원격 데스크톱을 통해 관리자로 연결된 사용자를 표시합니다.



원격 데스크톱

Amazon EC2 인스턴스 종료

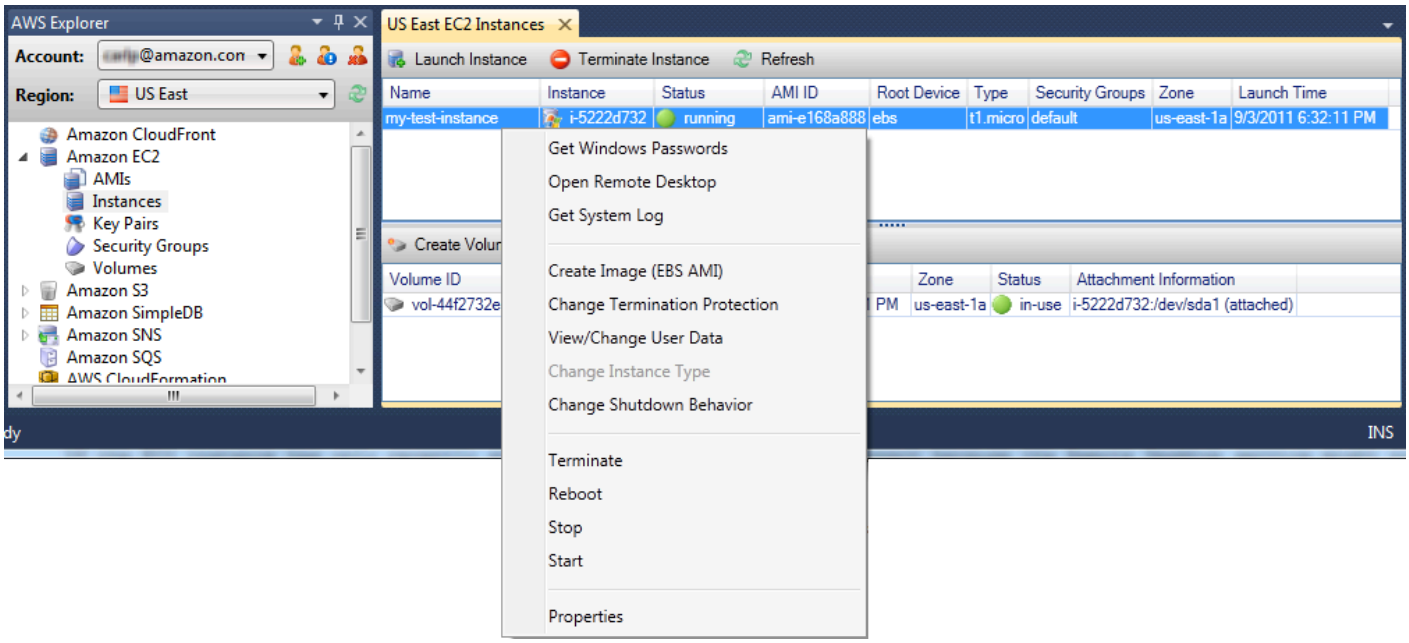
AWS 도구 키트를 사용하면 Visual Studio에서 실행 중인 Amazon EC2 인스턴스를 중지하거나 종료할 수 있습니다. 인스턴스를 중지하려면 EC2 인스턴스가 Amazon EBS 볼륨을 사용하고 있어야 합니다. EC2 인스턴스가 Amazon EBS 볼륨을 사용하지 않으면 인스턴스 종료만 가능합니다.

인스턴스를 중지하면 EBS 볼륨에 저장된 데이터가 보관됩니다. 인스턴스를 종료하면 인스턴스의 로컬 스토리지 장치에 저장된 모든 데이터가 손실됩니다. 중지하거나 종료하면 EC2 인스턴스의 비용은 부과되지 않습니다. 그러나 인스턴스를 중지한 경우 해당 인스턴스가 중지된 후에도 유지되는 EBS 스토리지의 비용은 계속 부과됩니다.

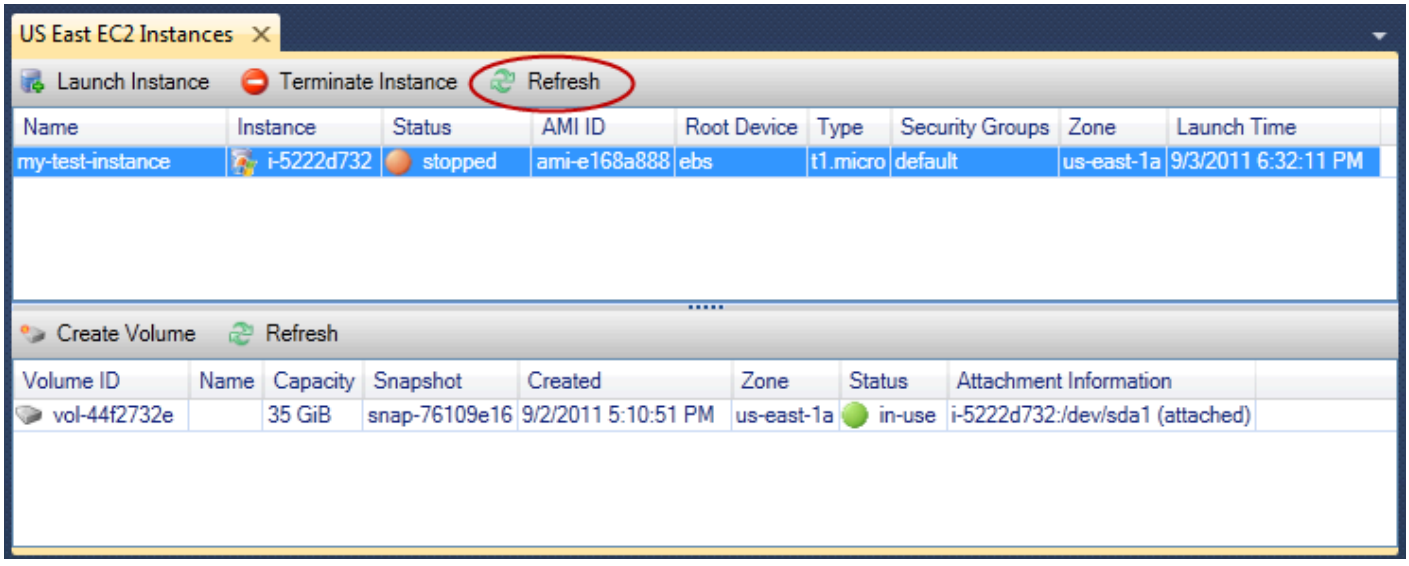
인스턴스를 종료하는 다른 방법은 원격 데스크톱을 사용하여 인스턴스에 연결한 다음 Windows 시작 메뉴에서 종료를 사용하는 것입니다. 이 시나리오에서는 인스턴스를 중지하거나 종료하도록 구성할 수 있습니다.

Amazon EC2 인스턴스를 중지하려면

1. AWS 탐색기에서 Amazon EC2 노드를 확장하고 인스턴스의 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴를 연 다음 보기를 선택합니다. 인스턴스 목록에서 중지하려는 인스턴스를 마우스 오른쪽 버튼으로 클릭하고 컨텍스트 메뉴에서 중지를 선택합니다. 예를 선택하여 인스턴스 중지를 확인합니다.

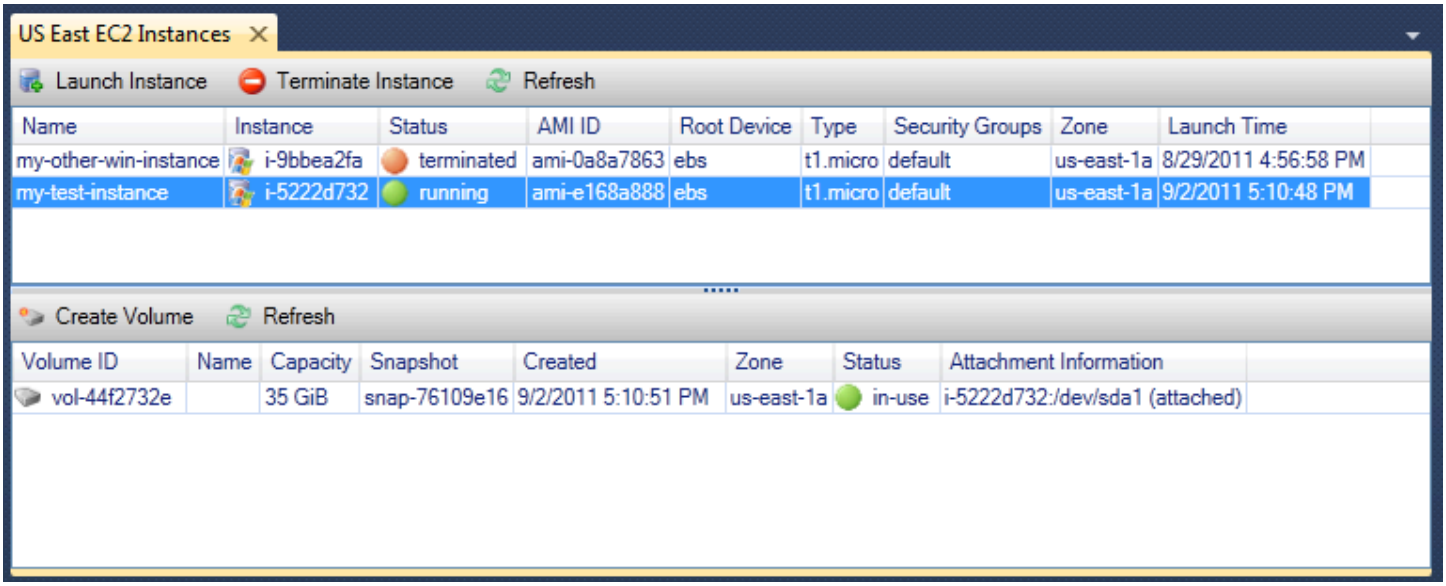


2. 인스턴스 목록 상단에서 새로 고침을 선택하여 Amazon EC2 인스턴스의 상태 변경을 확인합니다. 인스턴스를 종료하지 않고 중지했으므로 인스턴스와 연결된 EBS 볼륨은 계속 활성 상태입니다.



종료된 인스턴스가 계속 표시됨

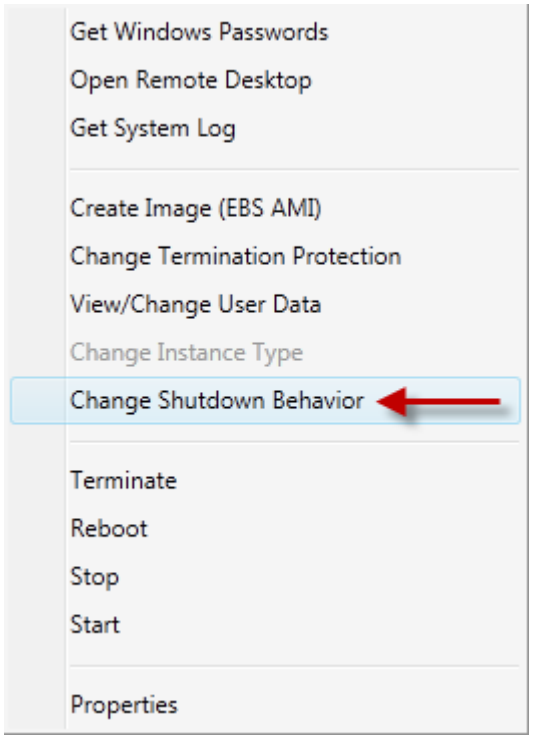
인스턴스를 종료하면 실행 중 또는 중지된 인스턴스와 함께 인스턴스 목록에 계속 나타납니다. 결국는 이러한 인스턴스를 AWS 회수하고 목록에서 사라집니다. 종료된 상태에서는 인스턴스의 비용이 부과되지 않습니다.



종료 시 EC2 인스턴스 동작을 지정하려면

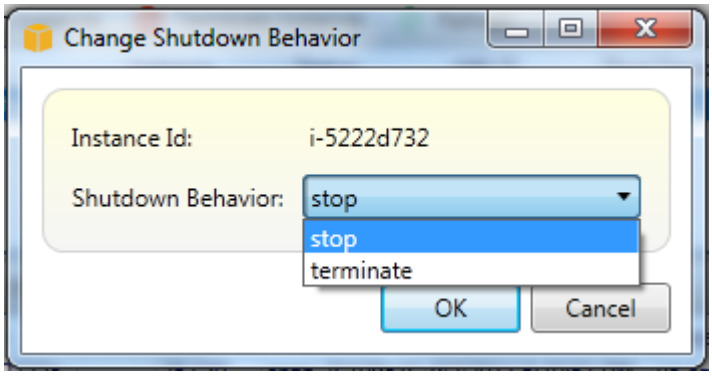
도구 AWS 키트를 사용하면 시작 메뉴에서 종료를 선택한 경우 Amazon EC2 인스턴스를 중지 또는 종료할지 여부를 지정할 수 있습니다.

1. 인스턴스 목록에서 Amazon EC2 인스턴스를 마우스 오른쪽 버튼으로 클릭한 다음 종료 방식 변경을 선택합니다.



종료 방식 변경 메뉴 항목

2. 종료 방식 변경 대화 상자의 종료 방식 드롭다운 목록에서 중지 또는 종료를 선택합니다.



Amazon ECS 인스턴스 관리

AWS 탐색기는 Amazon Elastic Container Service(Amazon ECS) 클러스터 및 컨테이너 리포지토리에 대한 세부 보기를 제공합니다. Visual Studio 개발 환경 내에서 클러스터 및 컨테이너 세부 정보를 생성, 삭제 및 관리할 수 있습니다.

서비스 속성 수정

클러스터 보기에서 서비스 세부 정보, 서비스 이벤트 및 서비스 속성을 확인할 수 있습니다.

1. AWS 탐색기에서 관리할 클러스터의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 연 다음 보기를 선택합니다.
2. ECS 클러스터 보기에서 왼쪽의 서비스를 클릭한 다음, 상세 정보 보기의 상세 정보 탭을 클릭합니다. 이벤트를 클릭하여 이벤트 메시지를 확인하고 배포를 클릭하여 배포 상태를 확인할 수 있습니다.
3. 편집을 클릭합니다. 원하는 작업 수와 최소 및 최대 정상 상태 백분율을 변경할 수 있습니다.
4. 변경 내용을 수락하려면 저장을 클릭하고 기존 값으로 되돌리려면 취소를 클릭합니다.

작업 중지

작업의 헌재 상태를 확인하고 클러스터 보기에서 하나 이상의 작업을 중지할 수 있습니다.

작업을 중지하려면

1. AWS 탐색기에서 중지하려는 작업이 있는 클러스터의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 연 다음 보기를 선택합니다.
2. ECS 클러스터 보기에서 왼쪽의 작업을 클릭합니다.
3. Desired Task Status(원하는 작업 상태) 가 `Running`으로 설정되어 있는지 확인합니다. 중지하려면 개별 작업들을 선택하고 중지 또는 모두 중지를 클릭하여 실행 중인 모든 작업을 선택하여 중지시킵니다.
4. Stop Tasks(작업 중지) 대화 상자에서 예를 선택합니다.

서비스 삭제

클러스터 보기에서 클러스터로부터 서비스를 삭제할 수 있습니다.

클러스터 서비스를 삭제하려면

1. AWS 탐색기에서 삭제하려는 서비스가 있는 클러스터의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 연 다음 보기를 선택합니다.
2. ECS 클러스터 보기에서 왼쪽의 서비스를 클릭한 다음, 삭제를 클릭합니다.
3. 클러스터 삭제 대화 상자에서 클러스터에 로드 밸런서와 대상 그룹이 있는 경우에 클러스터에서 이들을 삭제할 수 있습니다. 삭제된 서비스는 더 이상 사용되지 않습니다.
4. 클러스터 삭제 대화 상자에서 확인을 선택합니다. 삭제된 클러스터는 AWS 탐색기에서 제거됩니다.

클러스터 삭제

AWS 탐색기에서 Amazon Elastic Container Service 클러스터를 삭제할 수 있습니다.

클러스터 삭제

1. AWS 탐색기에서 Amazon ECS의 클러스터 노드에서 삭제할 클러스터의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 연 다음 삭제를 선택합니다.
2. 클러스터 삭제 대화 상자에서 확인을 선택합니다. 삭제된 클러스터는 AWS 탐색기에서 제거됩니다.

리포지토리 생성

AWS Explorer에서 Amazon Elastic Container Registry 리포지토리를 생성할 수 있습니다.

리포지토리 생성

1. AWS 탐색기에서 Amazon ECS 아래에 있는 리포지토리 노드의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 연 다음 리포지토리 생성을 선택합니다.
2. 리포지토리 생성 대화 상자에서 리포지토리 이름을 입력하고 확인을 선택합니다.

리포지토리 삭제

AWS 탐색기에서 Amazon Elastic Container Registry 리포지토리를 삭제할 수 있습니다.

리포지토리 삭제

1. AWS 탐색기에서 Amazon ECS 아래에 있는 리포지토리 노드의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 연 다음 리포지토리 삭제를 선택합니다.
2. 리포지토리 삭제 대화 상자에서 이미지가 포함된 경우라도 리포지토리를 삭제하겠다고 선택할 수 있습니다. 그렇지 않으면 비어 있는 경우만 삭제가 됩니다. 예를 클릭합니다.

AWS Explorer에서 보안 그룹 관리

Toolkit for Visual Studio를 사용하면 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스 및 CloudFormation과 함께 사용할 보안 그룹을 생성하고 구성할 수 있습니다. Amazon EC2 인스턴스를 시작하거나 애플리케이션을 배포할 때 Amazon EC2 인스턴스와 연결할 보안 그룹을 CloudFormation 지정합니다. (에 배포하면 Amazon EC2 인스턴스가 CloudFormation 생성됩니다.)

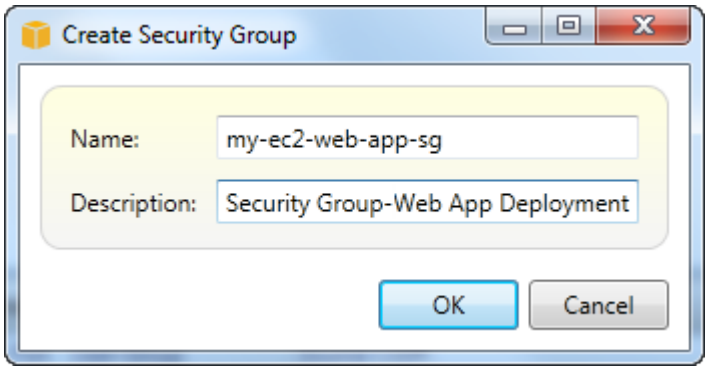
보안 그룹은 수신되는 네트워크 트래픽에 대한 방화벽의 역할을 하며, 보안 그룹은 Amazon EC2 인스턴스에 허용되는 네트워크 트래픽의 유형을 지정합니다. 또한 특정 IP 주소 또는 지정된 사용자나 다른 보안 그룹에서만 수신 트래픽이 수락되도록 지정할 수 있습니다.

보안 그룹 생성

이 단원에서는 보안 그룹을 생성합니다. 보안 그룹이 생성된 후에는 해당 보안 그룹에 권한이 구성되어 있지 않습니다. 권한 구성은 추가 작업을 통해 처리됩니다.

보안 그룹을 생성하는 방법

1. AWS 탐색기의 Amazon EC2 노드에서 보안 그룹 노드의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 연 다음 보기를 선택합니다.
2. EC2 보안 그룹 탭에서 보안 그룹 생성을 선택합니다.
3. 보안 그룹 생성 대화 상자에서 보안 그룹의 이름과 설명을 입력한 다음 확인을 선택합니다.

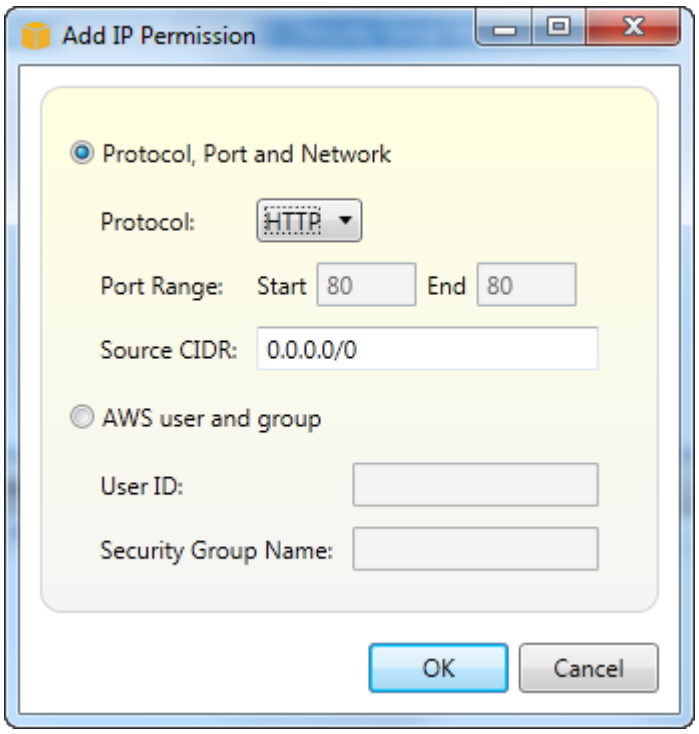


보안 그룹에 권한 추가

이 단원에서는 보안 그룹에 HTTP 및 HTTPS 프로토콜을 통한 웹 트래픽을 허용하는 권한을 추가합니다. 또한 다른 컴퓨터가 Windows RDP(Remote Desktop Protocol)를 사용하여 연결하는 것도 허용합니다.

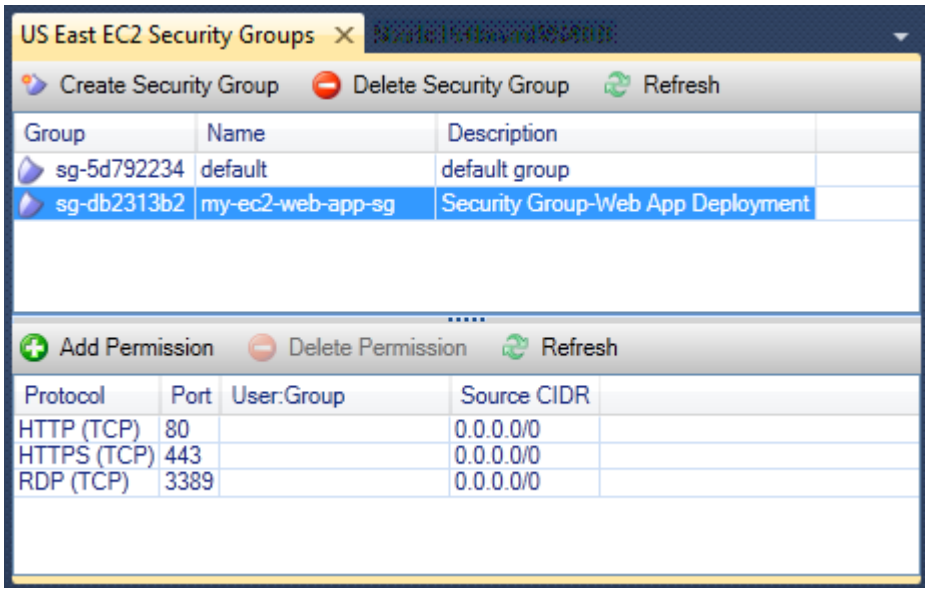
보안 그룹에 권한을 추가하려면,

1. EC2 보안 그룹 탭에서 보안 그룹을 선택한 다음 권한 추가 버튼을 선택합니다.
2. Add IP Permission(IP 권한 추가) 대화 상자에서 Protocol, Port and Network(프로토콜, 포트 및 네트워크) 라디오 버튼을 선택한 다음 프로토콜 드롭다운 목록에서 HTTP를 선택합니다. 포트 범위는 HTTP에 대한 기본 포트인 포트 80으로 자동으로 조정됩니다. Source CIDR(소스 CIDR) 필드의 기본값은 0.0.0.0/0입니다. 이 값은 HTTP 네트워크 트래픽이 모든 외부 IP 주소에서 수락되도록 지정합니다. 확인을 선택합니다.



이 보안 그룹에 대해 포트 80(HTTP) 열기

3. HTTPS 및 RDP에 대해 이 절차를 반복합니다. 보안 그룹 권한이 이제 다음과 같아야 합니다.



사용자 ID 및 보안 그룹 이름을 지정하여 보안 그룹의 권한을 설정할 수도 있습니다. 이 경우 이 보안 그룹의 Amazon EC2 인스턴스는 지정된 보안 그룹의 Amazon EC2 인스턴스에서 수신되는 모든 네트워크 트래픽을 허용합니다. 또한 보안 그룹 이름을 명확하게 구분하는 방법으로 사용자 ID를 지정해야 합

니다. 보안 그룹 이름은 모두 고유할 필요는 없습니다 AWS. 보안 그룹에 대한 자세한 내용은 EC2 설명서를 참조하십시오.

Amazon EC2 인스턴스에서 AMI 생성

AWS Toolkit for Visual Studio를 사용하여 Amazon Machine Image(AMI)를 생성할 수 있습니다. AMI에 대한 자세한 정보는 Windows 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 Amazon Machine Image(AMI) 주제를 참조하세요.

기존 Amazon EC2 인스턴스에서 AMI를 생성하려면 다음 절차를 완료합니다.

기존 Amazon EC2 인스턴스에서 AMI 생성

1. AWS Toolkit Explorer에서 Amazon EC2를 확장하고 인스턴스를 선택하여 기존 인스턴스 목록을 봅니다.
2. AMI의 기반으로 사용할 인스턴스를 마우스 오른쪽 버튼으로 클릭하고 이미지 생성(ABS AMI)을 선택하여 이미지 생성 대화 상자를 엽니다.
3. 이미지 생성 대화 상자 창에서 제공된 필드에 이미지의 이름과 설명을 추가한 다음 확인 버튼을 선택하여 계속합니다.
4. 이미지가 생성되면 Visual Studio에서 이미지 생성됨 확인 창이 열리고 확인 버튼을 선택하여 계속합니다.

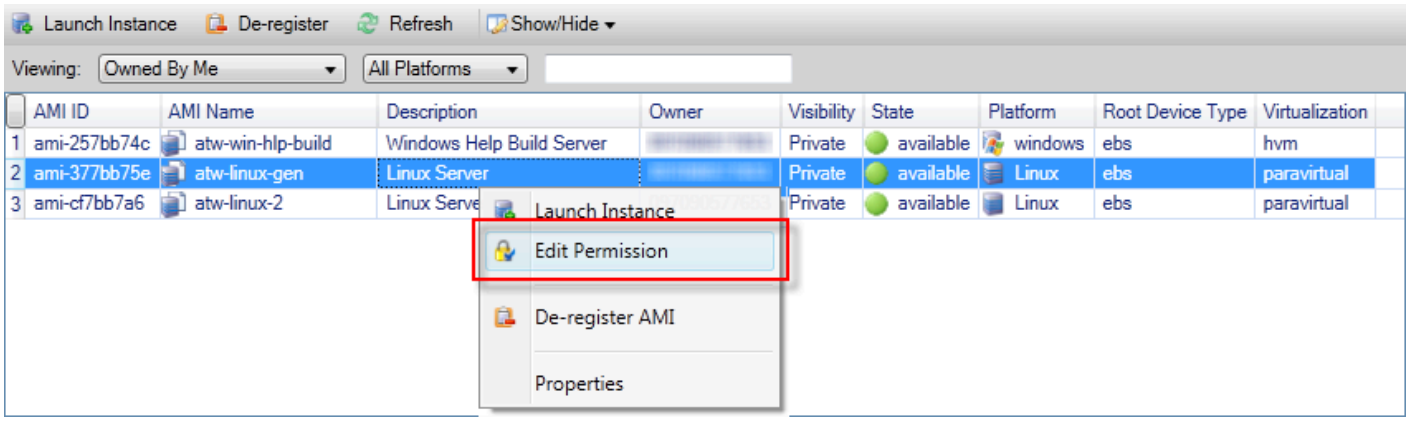
AWS 도구 키트를 사용하여 새 AMI를 보려면 Amazon EC2를 확장하고 AMIs 두 번 클릭하여 Visual Studio Editor Payne에서 기존 AMIs 목록을 표시하는 창을 엽니다. 목록에 새 AMI가 표시되지 않으면 AMI 창 상단에 있는 새로 고침 버튼을 선택합니다.

Amazon Machine Image의 시작 권한 설정

AWS Explorer의 AMIs)에 대한 시작 권한을 설정할 수 있습니다. AMIs Set AMI Permissions(AMI 권한 설정) 대화 상자를 사용하여 AMI에서 권한을 복사할 수 있습니다.

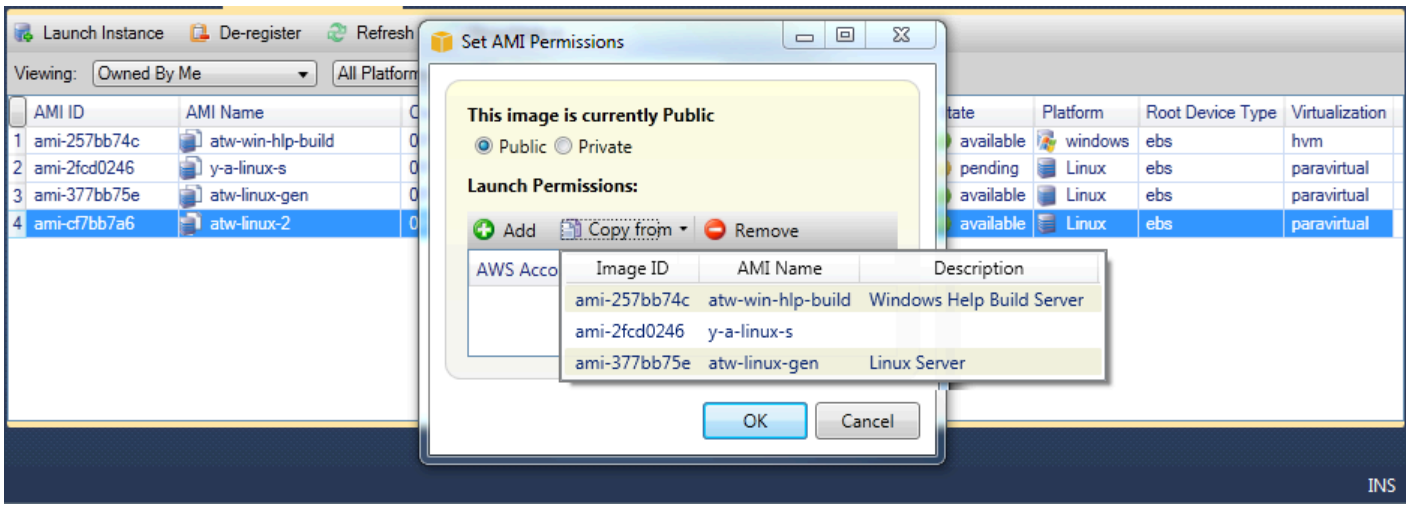
AMI에서 권한을 설정하려면

1. AWS 탐색AMIs 보기에서 AMI의 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴를 연 다음 권한 편집을 선택합니다.



2. Set AMI Permissions(AMI 권한 편집) 대화 상자에서 3개의 옵션을 사용할 수 있습니다.
   - 시작 권한을 부여하려면 추가를 선택하고 시작 권한을 부여할 AWS 사용자의 계정 번호를 입력합니다.
   - 시작 권한을 제거하려면 시작 권한을 제거할 AWS 사용자의 계정 번호를 선택하고 제거를 선택합니다.
   - AMI의 권한을 다른 AMI에 복사하려면 목록에서 AMI를 선택하고 Copy from(다음으로부터 복사)을 선택합니다. 선택한 AMI에 대한 시작 권한이 있는 사용자에게 현재 AMI에 대한 시작 권한이 부여됩니다. Copy-from(다음으로부터 복사) 목록에 있는 다른 AMI로 이 프로세스를 반복하여 여러 AMI의 권한을 대상 AMI로 복사할 수 있습니다.

     다음으로부터 복사 목록에는 AWS 탐색기에 AMI 보기가 표시될 때 활성 상태의 계정이 소유한 AMI만 포함됩니다. 따라서 활성 계정이 소유한 AMI가 없으면 Copy-from(다음으로부터 복사) 목록에 아무 것도 표시되지 않을 수 있습니다.



Copy AMI permissions(AMI 권한 복사) 대화 상자

Amazon Virtual Private Cloud(VPC)

Amazon Virtual Private Cloud(Amazon VPC)를 사용하면 사용자가 정의한 가상 네트워크로 Amazon Web Services 리소스를 시작할 수 있습니다. 이 가상 네트워크는 AWS의 확장 가능한 인프라를 사용한다는 이점과 함께 고객의 자체 데이터 센터에서 운영하는 기존 네트워크와 유사합니다. 자세한 내용은 Amazon VPC 사용 설명서를 참조하십시오.

개발자는 Toolkit for Visual Studio를 사용하여 AWS Management Console에서 공개한 기능과 유사한 VPC 기능에 액세스할 수 있지만, Visual Studio 개발 환경의 기능에는 액세스할 수 없습니다. AWS Explorer의 Amazon VPC 노드에는 다음 영역에 대한 하위 노드가 포함되어 있습니다.

- VPC
- 서브넷
- 엘라스틱 IP 주소
- 인터넷 게이트웨이
- 네트워크 ACL
- 라우팅 테이블
- 보안 그룹

를 사용하여 배포를 위한 퍼블릭-프라이빗 VPC 생성 AWS Elastic Beanstalk

이 섹션에서는 퍼블릭 및 프라이빗 서브넷이 모두 포함되어 있는 Amazon VPC를 생성하는 방법을 설명합니다. 퍼블릭 서브넷에는 프라이빗 서브넷의 인스턴스가 퍼블릭 인터넷과 통신하도록 Network Address Translation(NAT)을 수행하는 Amazon EC2 인스턴스가 포함되어 있습니다. 두 서브넷은 동일한 AZ(가용 영역)에 있어야 합니다.

이는 VPC에 AWS Elastic Beanstalk 환경을 배포하는 데 필요한 최소 VPC 구성입니다. 이 시나리오에서 애플리케이션을 호스팅하는 Amazon EC2 인스턴스는 프라이빗 서브넷에 있으며, 수신 트래픽을 애플리케이션에 라우팅하는 Elastic Load Balancing 로드 밸런서는 퍼블릭 서브넷에 있습니다.

Network Address Translation(NAT)에 대한 자세한 정보는 Amazon Virtual Private Cloud 사용 설명서의 NAT 인스턴스를 참조하세요. VPC를 사용하도록 배포를 구성하는 방법에 대한 예제는 Elastic Beanstalk에 배포를 참조하십시오.

퍼블릭-프라이빗 서브넷 VPC를 생성하려면

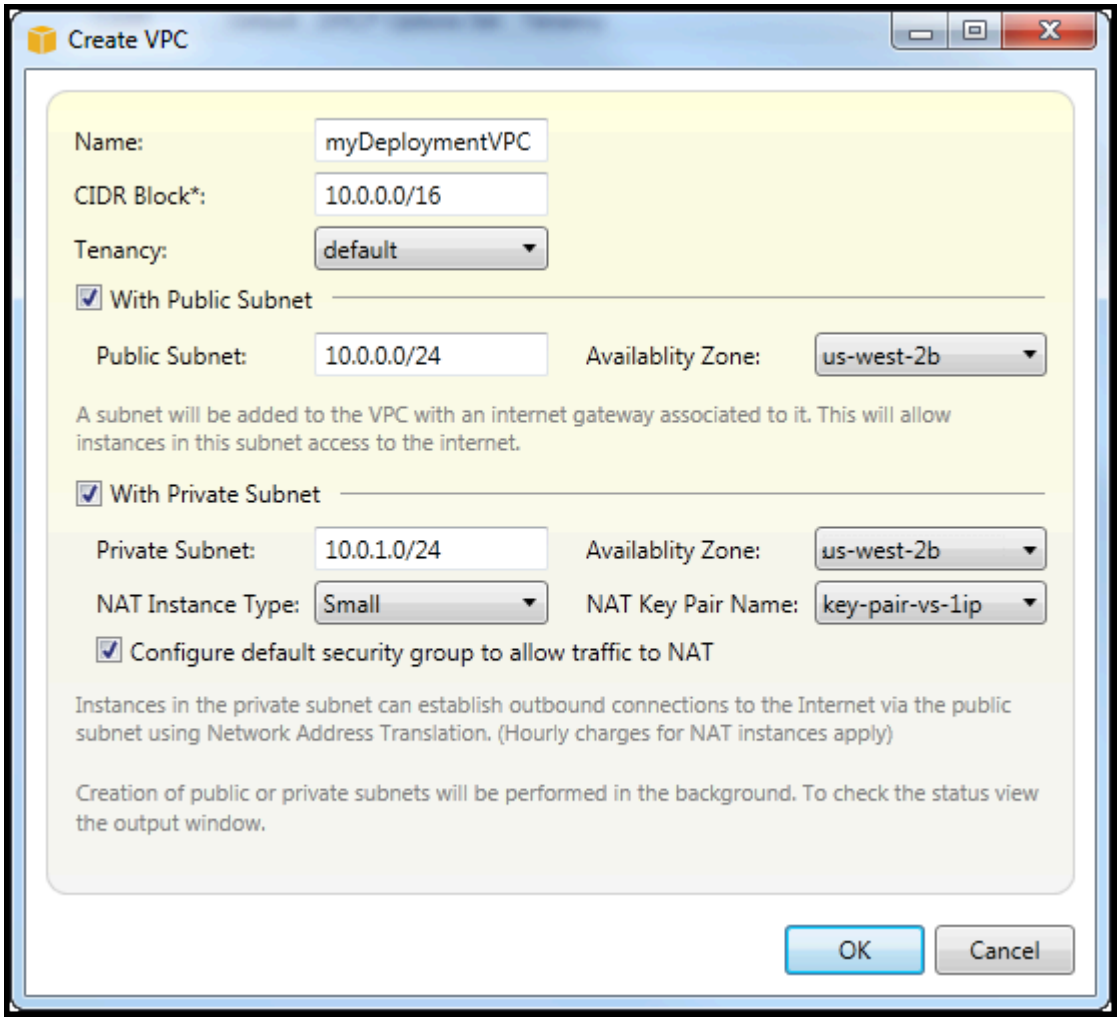
1. AWS 탐색기의 Amazon VPC 노드에서 VPCs 하위 노드를 연 다음 VPC 생성을 선택합니다.

2. 다음과 같이 VPC를 구성합니다.

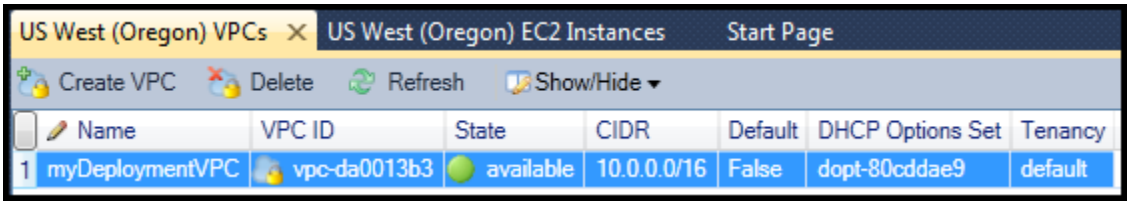
   - VPC 이름을 입력합니다.
   - With Public Subnet(퍼블릭 서브넷 사용) 및 With Private Subnet(프라이빗 서브넷 사용) 확인란을 선택합니다.
   - 각 서브넷에 대한 Availability Zone(가용 영역) 드롭다운 목록에서 가용 영역을 선택합니다. 두 서브넷에 대해 동일한 AZ를 사용해야 합니다.
   - 프라이빗 서브넷의 경우 NAT Key Pair Name(NAT 키 페어 이름)에 키 페어를 제공합니다. 이 키 페어는 프라이빗 서브넷에서 퍼블릭 인터넷으로 네트워크 주소 변환을 수행하는 Amazon EC2 인스턴스에 대해 사용됩니다.
   - Configure default security group to allow traffic to NAT(기본 보안 그룹을 구성하여 NAT에 대한 트래픽 허용) 확인란을 선택합니다.

   VPC 이름을 입력합니다. With Public Subnet(퍼블릭 서브넷 사용) 및 With Private Subnet(프라이빗 서브넷 사용) 확인란을 선택합니다. 각 서브넷에 대한 Availability Zone(가용 영역) 드롭다운 목록에서 가용 영역을 선택합니다. 두 서브넷에 대해 동일한 AZ를 사용해야 합니다. 프라이빗 서브넷의 경우 NAT Key Pair Name(NAT 키 페어 이름)에 키 페어를 제공합니다. 이 키 페어는 프라이빗 서브넷에서 퍼블릭 인터넷으로 네트워크 주소 변환을 수행하는 Amazon EC2 인스턴스에 대해 사용됩니다. Configure default security group to allow traffic to NAT(기본 보안 그룹을 구성하여 NAT에 대한 트래픽 허용) 확인란을 선택합니다.

   확인을 선택합니다.

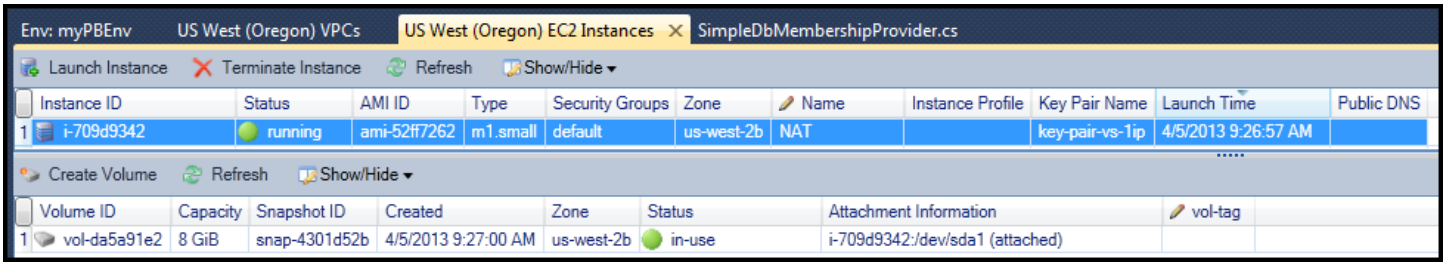


AWS 탐색기의 VPC 탭에서 새 VPCs를 볼 수 있습니다.



NAT 인스턴스를 시작하는 데 몇 분 정도 걸릴 수 있습니다. 사용 가능한 경우 AWS 탐색기에서 Amazon EC2 노드를 확장한 다음 인스턴스 하위 노드를 열어 볼 수 있습니다.

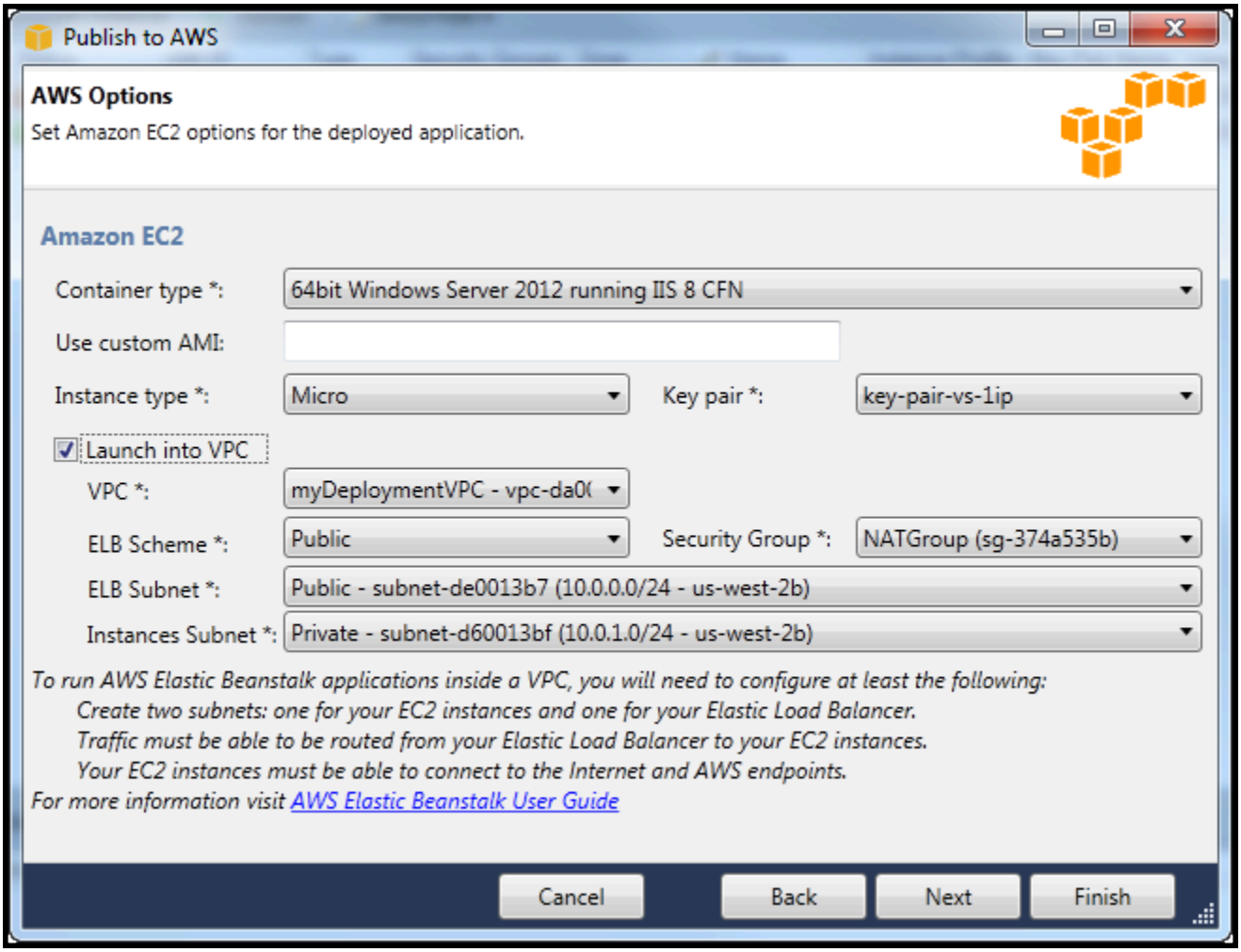
NAT 인스턴스에 대해 Amazon Elastic Block Store(Amazon EBS) 볼륨이 자동으로 생성됩니다. Amazon EBS에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서에서 Amazon Elastic Block Store(Amazon EBS)를 참조하세요.



AWS Elastic Beanstalk 환경에 애플리케이션을 배포하고 VPC에서 환경을 시작하도록 선택하면 Toolkit은 VPC의 구성 정보로에 게시 Amazon Web Services 대화 상자를 채웁니다.

툴킷은 AWS Management Console을 사용하여 생성된 VPC의 정보가 아닌 툴킷에서 생성된 VPC의 정보만으로 대화 상자를 채웁니다. 이는 도구 키트가 VPC를 생성할 때 정보에 액세스할 수 있도록 VPC의 구성 요소에 태그를 지정하기 때문입니다.

배포 마법사의 다음 스크린샷은 도구 키트에서 생성된 VPC의 값으로 채워진 대화 상자의 예를 표시합니다.



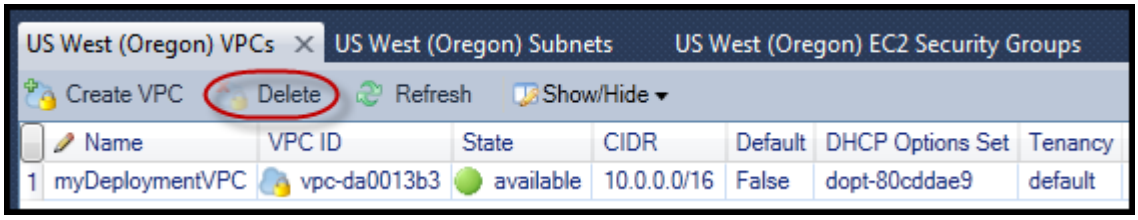
VPC를 삭제하는 방법

VPC를 삭제하려면 먼저 VPC에서 모든 Amazon EC2 인스턴스를 종료해야 합니다.

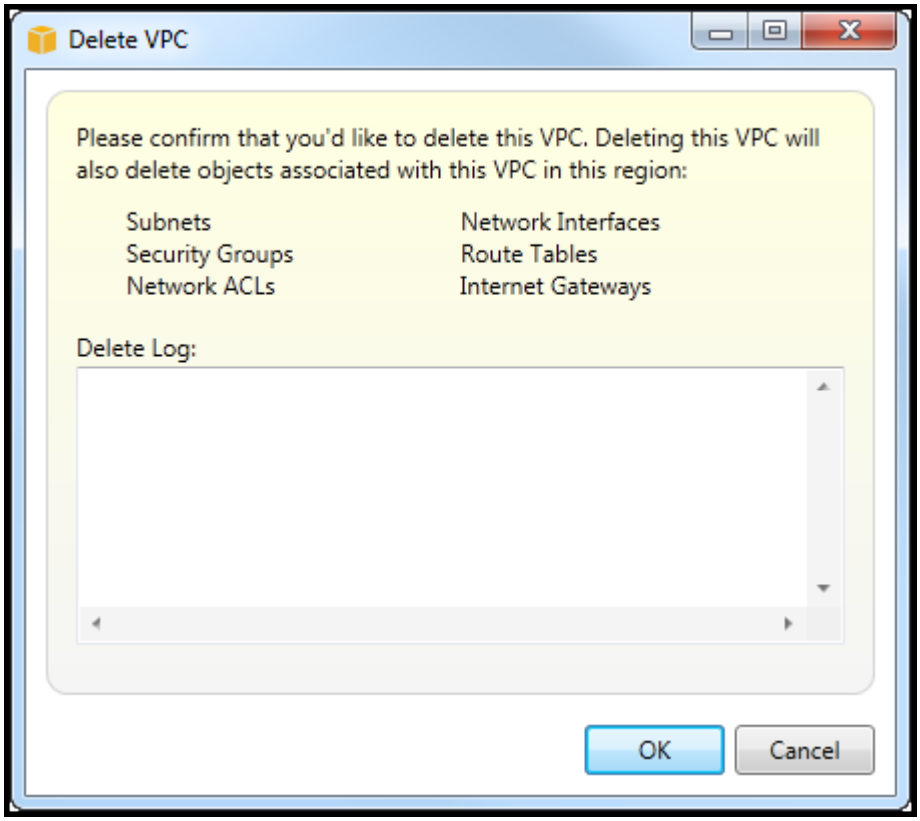
1. VPC의 AWS Elastic Beanstalk 환경에 애플리케이션을 배포한 경우 환경을 삭제합니다. 이렇게 하면 Elastic Load Balancing 로드 밸런서와 함께 애플리케이션을 호스팅한 모든 Amazon EC2 인스턴스가 종료됩니다.

   환경을 삭제하지 않고 애플리케이션을 호스팅하는 인스턴스를 직접 종료하려고 시도하면 Auto Scaling 서비스가 자동으로 새 인스턴스를 생성하여 삭제된 인스턴스를 대체합니다. 자세한 내용은 Auto Scaling 개발자 안내서를 참조하십시오.
2. VPC에 대한 NAT 인스턴스를 삭제합니다.

   VPC를 삭제하기 위해 NAT 인스턴스와 연결된 Amazon EBS 볼륨을 삭제할 필요가 없습니다. 그러나 볼륨을 삭제하지 않으면 NAT 인스턴스와 VPC를 삭제한 경우에도 해당 볼륨에 대해 계속 요금이 부과됩니다.
3. VPC 탭에서 삭제 링크를 선택하여 VPC를 삭제합니다.



4. VPC 삭제 대화 상자에서확인을 선택합니다.



Visual Studio용 CloudFormation 템플릿 편집기 사용

Toolkit for Visual Studio에는 Visual Studio용 CloudFormation 템플릿 편집기 및 CloudFormation 템플릿 프로젝트가 포함되어 있습니다. 지원되는 기능은 다음과 같습니다.

- 제공된 CloudFormation 템플릿 프로젝트 유형을 사용하여 새로운 템플릿(비어 있거나 기존 스택 또는 샘플 템플릿에서 복사) 생성
- 자동 JSON 확인, 자동 완성, 코드 폴딩 및 구문 강조로 템플릿 편집
- 템플릿의 필드 값의 내장 함수 및 리소스 참조 파라미터 자동 제안
- Visual Studio에서 템플릿에 대한 일반적인 작업을 수행할 메뉴 항목입니다.

주제

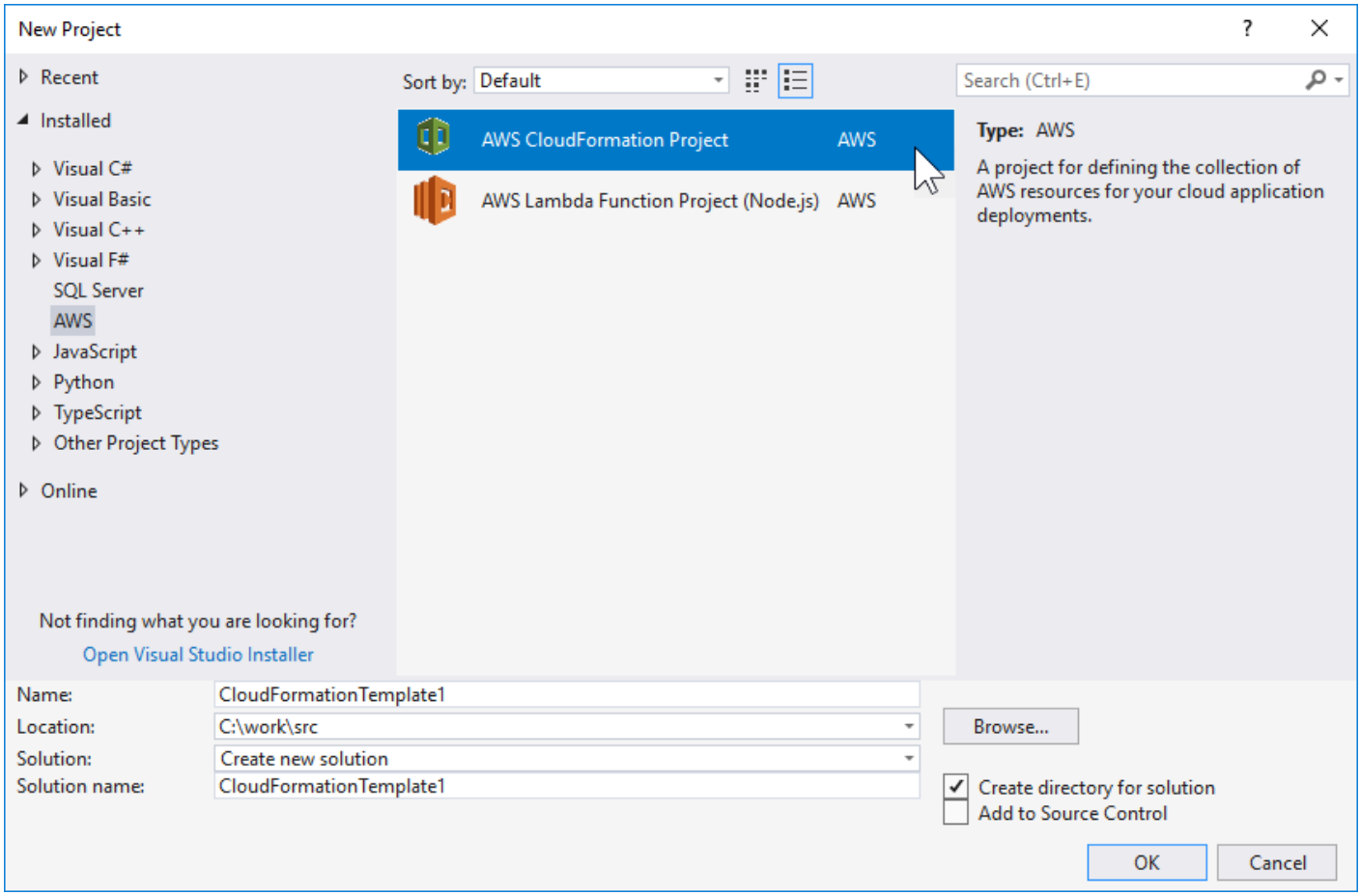
- Visual Studio에서 CloudFormation 템플릿 프로젝트 생성
- Visual Studio에 CloudFormation 템플릿 배포
- Visual Studio에서 CloudFormation 템플릿 형식 지정

Visual Studio에서 CloudFormation 템플릿 프로젝트 생성

템플릿 프로젝트를 생성하려면

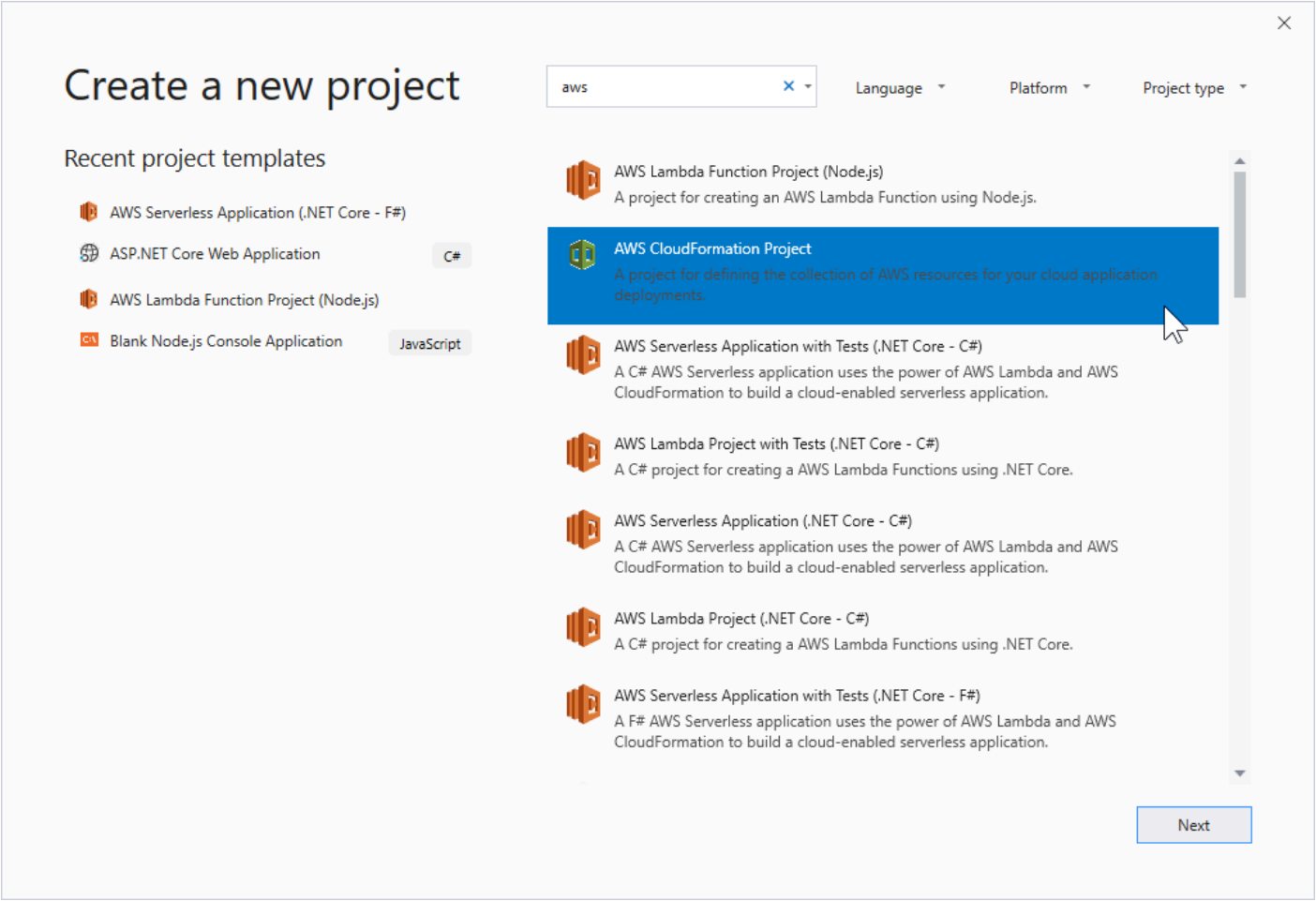
1. Visual Studio의 파일 메뉴에서 새로 만들기를 선택한 다음 프로젝트를 선택합니다.
2. Visual Studio 2017:

   새 프로젝트 대화 상자에서 설치됨을 확장하고 AWS를 선택하세요.



   Visual Studio 2019:

   새 프로젝트 대화 상자에서 언어, 플랫폼 및 프로젝트 유형 드롭다운 상자가 “모두...”로 설정되어 있는지 확인하고 검색 필드에 aws를 입력하세요.

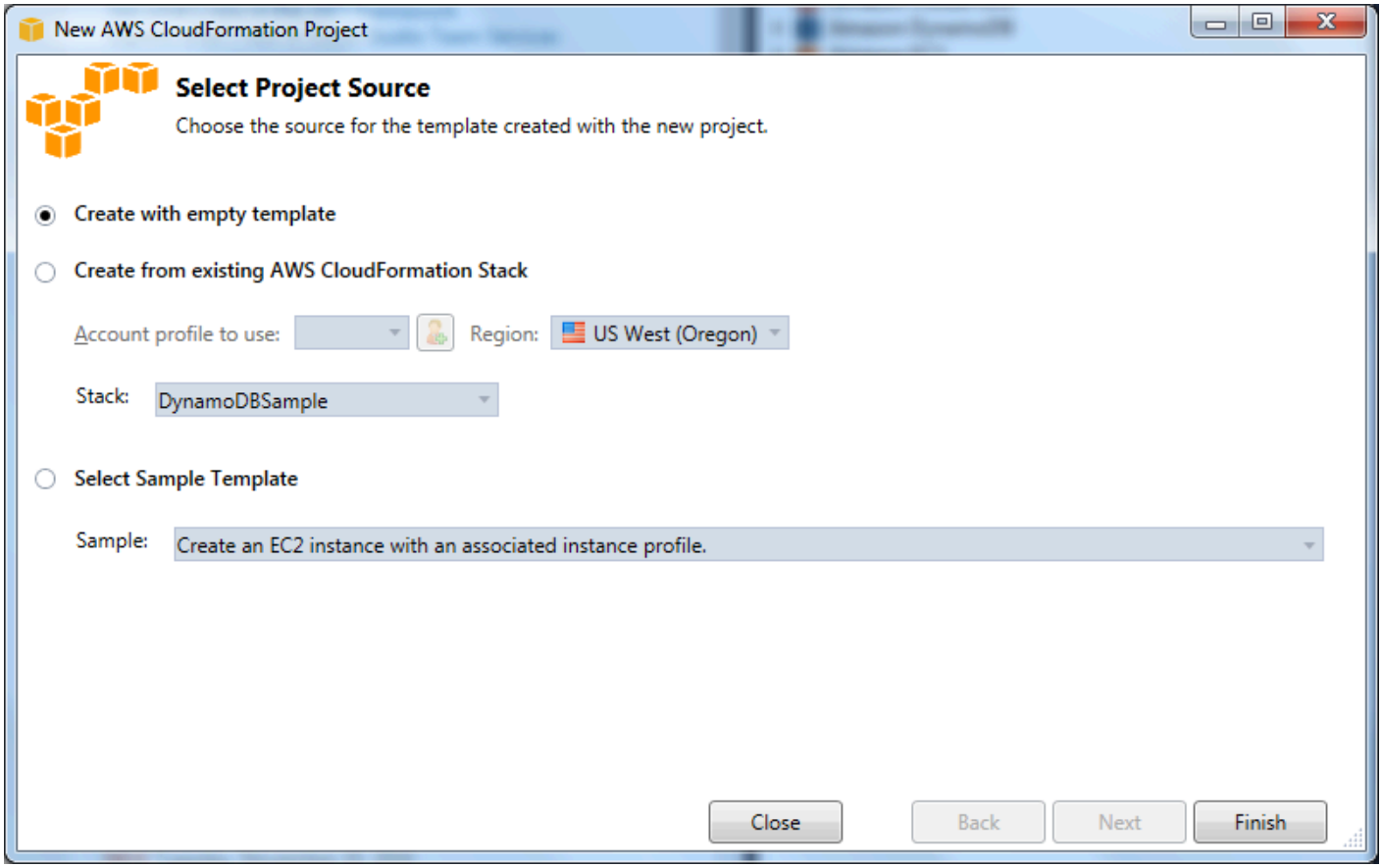


3. AWS CloudFormation 프로젝트 템플릿을 선택하세요.
4. Visual Studio 2017:

   템플릿 프로젝트에 대해 원하는 이름, 위치 등을 입력한 다음 확인을 클릭합니다.

   Visual Studio 2019:

   다음을 클릭합니다. 다음 대화 상자에서 템플릿 프로젝트에 대해 원하는 이름, 위치 등을 입력한 다음 만들기를 클릭합니다.
5. Select Project Source(프로젝트 소스 선택) 페이지에서 생성할 템플릿의 소스를 선택합니다.
   - Create with empty template(빈 템플릿으로 생성)은 새로운 빈 CloudFormation 템플릿을 생성합니다.
   - 기존 AWS |CFN| 스택에서 생성하면 AWS 계정의 기존 스택에서 템플릿이 생성됩니다. (스택은 `CREATE_COMPLETE` 상태일 필요가 없습니다.)
   - Select sample template(샘플 템플릿 선택)은 CloudFormation 샘플 템플릿 중 하나에서 템플릿을 생성합니다.

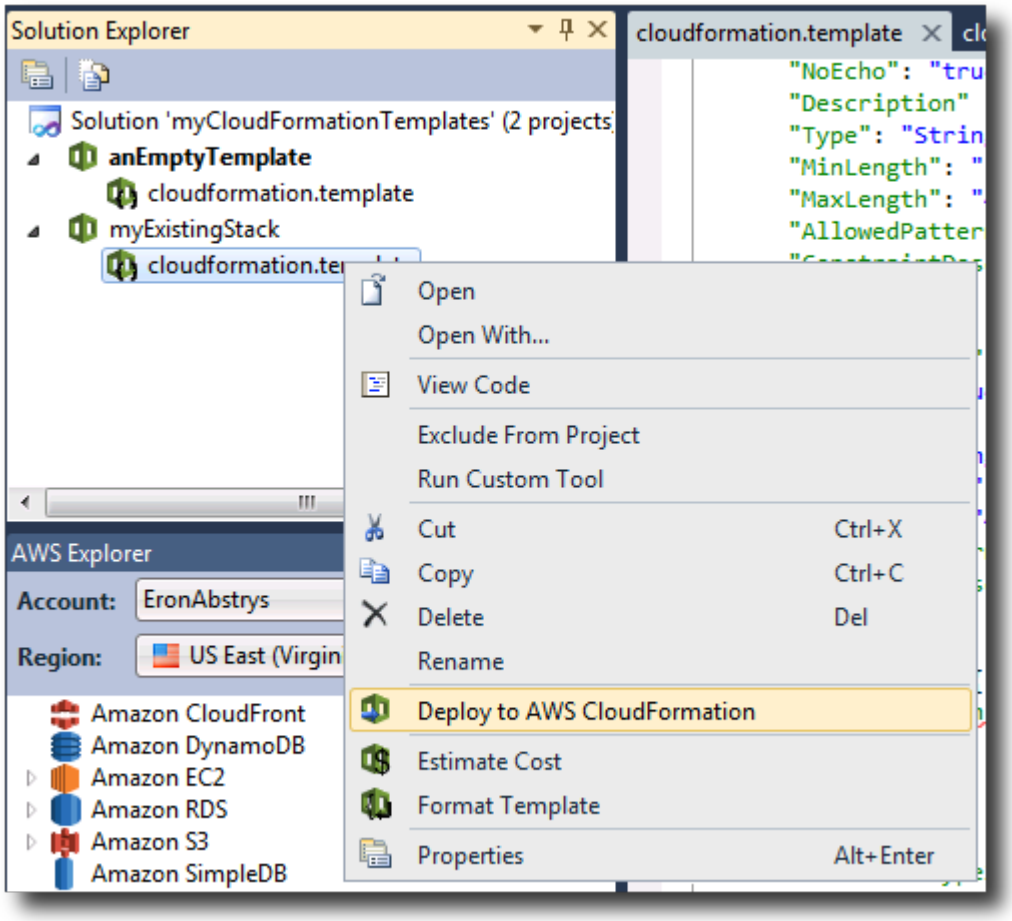


6. CloudFormation 템플릿 프로젝트 생성을 완료하려면 완료를 선택합니다.

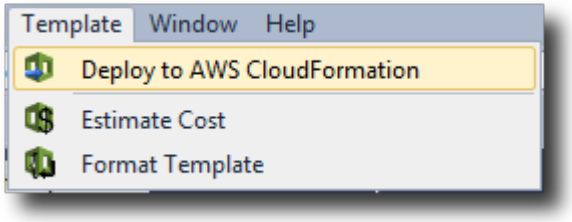
Visual Studio에 CloudFormation 템플릿 배포

CFN 템플릿을 배포하려면

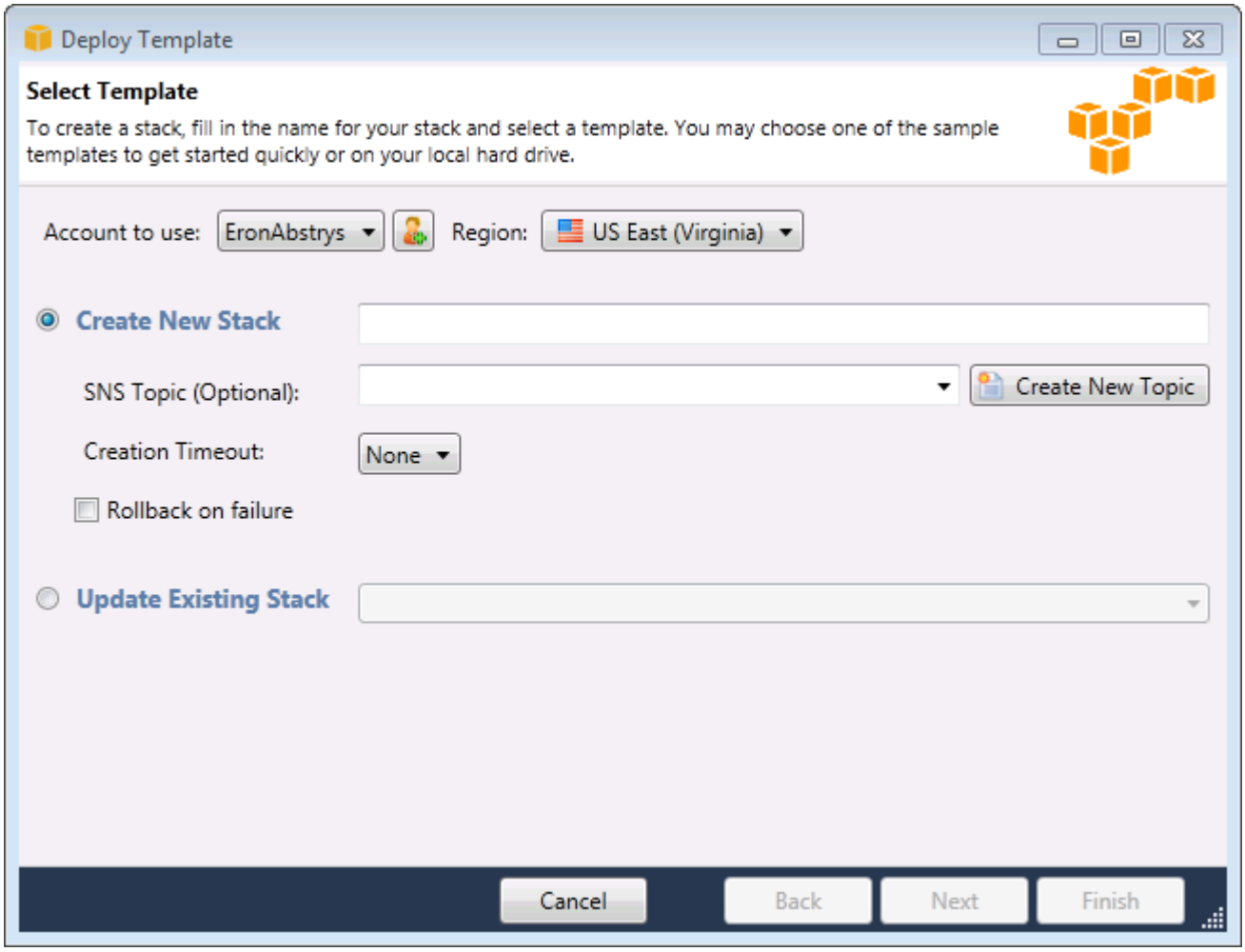
1. 솔루션 탐색기에서 배포할 템플릿의 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴를 열고 AWS CloudFormation에 배포를 선택하세요.



현재 편집 중인 템플릿을 배포하려면 템플릿 메뉴에서 AWS CloudFormation에 배포를 선택하세요.



2. 템플릿 배포 페이지에서 스택 AWS 계정 을 시작하는 데 사용할와 스택이 시작될 리전을 선택합니다.

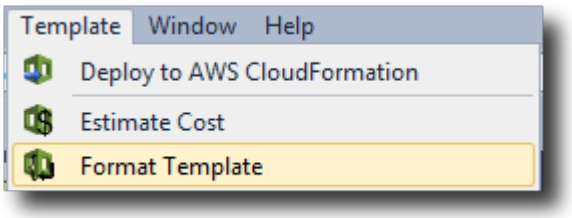


3. Create New Stack(스택 새로 만들기)을 선택하고 스택 이름을 입력합니다.
4. 다음 옵션을 선택합니다(아무 것도 선택하지 않아도 됨).
   - 스택 진행 상황에 대한 알림을 받으려면 SNS 주제 드롭다운 목록에서 SNS 주제를 선택합니다. 새 주제 생성을 선택하고 상자에 이메일 주소를 입력하여 SNS 주제를 생성할 수도 있습니다.
   - 생성 제한 시간을 사용하여 스택이 실패로 선언되기 전에에서 스택을 생성할 수 CloudFormation 있는 시간을 지정합니다(실패 시 롤백 옵션을 선택 취소하지 않는 한 롤백).
   - 실패 시 롤백을 사용하여 실패 시 스택을 롤백(자체 삭제)합니다. 실행을 완료하지 못해도 디버깅을 목적으로 스택을 계속 활성 상태로 두려면 이 옵션의 해제 상태를 유지합니다.
5. 완료를 선택하여 스택을 시작합니다.

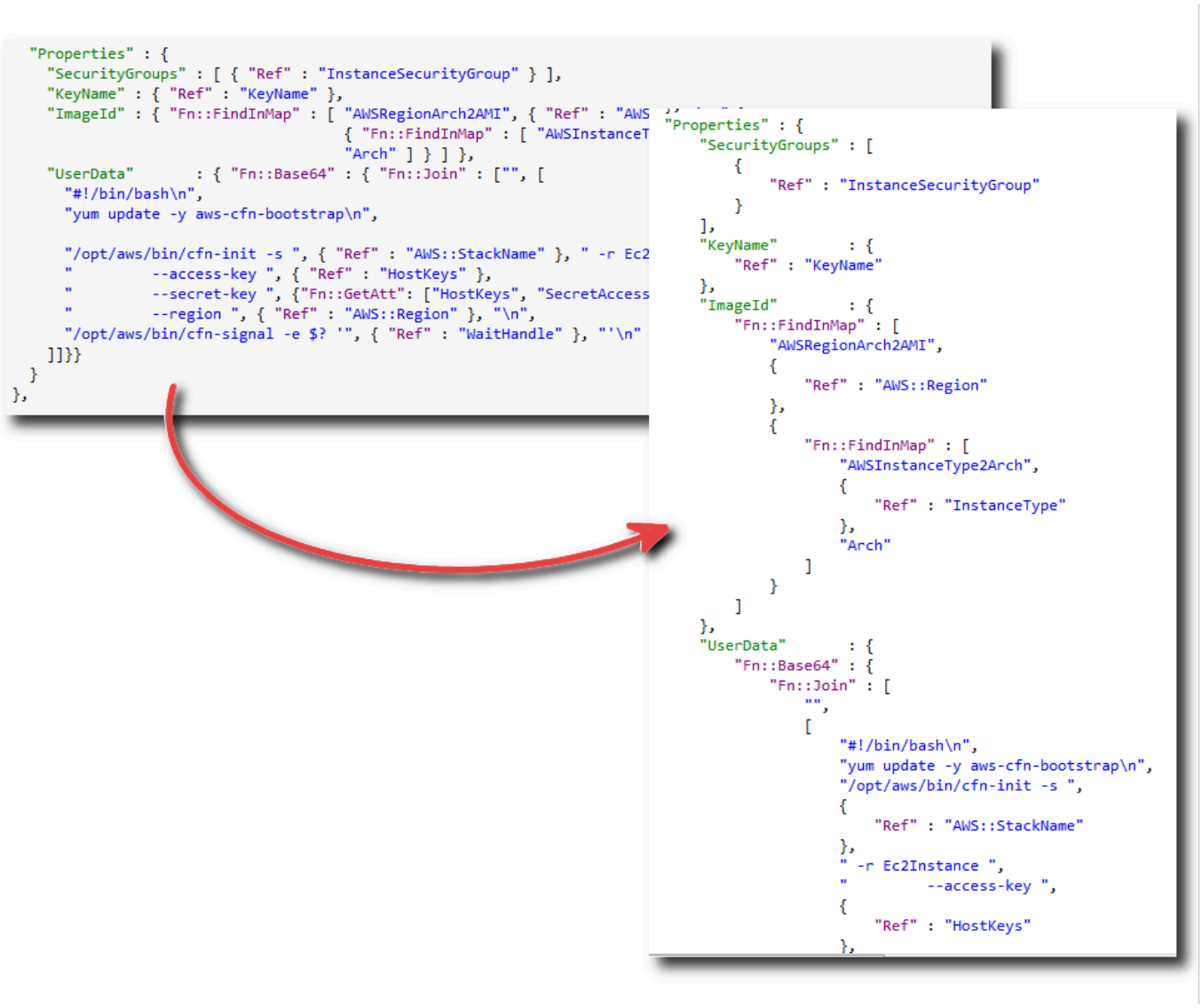
Visual Studio에서 CloudFormation 템플릿 형식 지정

- Solution Explorer에서 템플릿에 대한 컨텍스트(오른쪽 클릭) 메뉴를 열고 Format Template(형식 템플릿)을 선택합니다.

또는 현재 편집 중인 템플릿의 형식을 지정하기 위해 템플릿 메뉴에서 Format Template(형식 템플릿)을 선택합니다.



JSON 코드의 구조가 명시적으로 나타나도록 해당 코드의 형식이 지정됩니다.



AWS Explorer에서 Amazon S3 사용

Amazon Simple Storage Service(S3)를 사용하면 인터넷에 연결된 장소에서 데이터를 저장하고 검색할 수 있습니다. Amazon S3에 저장한 모든 데이터는 계정과 연결되며, 기본적으로 사용자만 액세스할 수 있습니다. Toolkit for Visual Studio를 사용하면 Amazon S3에 데이터를 저장하고 해당 데이터를 보고, 관리하고, 검색하고, 배포할 수 있습니다.

Amazon S3는 파일 시스템이나 논리 드라이브와 유사한 것으로 생각할 수 있는 버킷 개념을 사용합니다. 버킷에는 디렉터리와 비슷한 폴더 및 파일과 비슷한 객체가 포함될 수 있습니다. 이 섹션에서는 Toolkit for Visual Studio가 공개한 Amazon S3 기능에 대해 알아볼 때 이러한 개념을 사용합니다.

> **Note**
>
> 이 도구를 사용하려면 IAM 정책이 `s3:GetBucketAcl`, `s3:GetBucket`, `s3:ListBucket` 작업에 대한 권한을 부여해야 합니다. 자세한 내용은 AWS IAM 정책 개요를 참조하세요.

Amazon S3 버킷 생성

버킷은 Amazon S3에서 가장 기본적인 저장 단위입니다.

S3 버킷을 생성하려면,

1. AWS 탐색기에서 Amazon S3 노드의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 연 다음 버킷 생성을 선택합니다.
2. 버킷 만들기 대화 상자에서 버킷에 대한 이름을 입력합니다. 버킷 이름은 AWS에서 고유해야 합니다. 다른 제약 조건에 대한 자세한 내용은 Amazon S3 설명서를 참조하십시오.
3. 확인을 선택합니다.

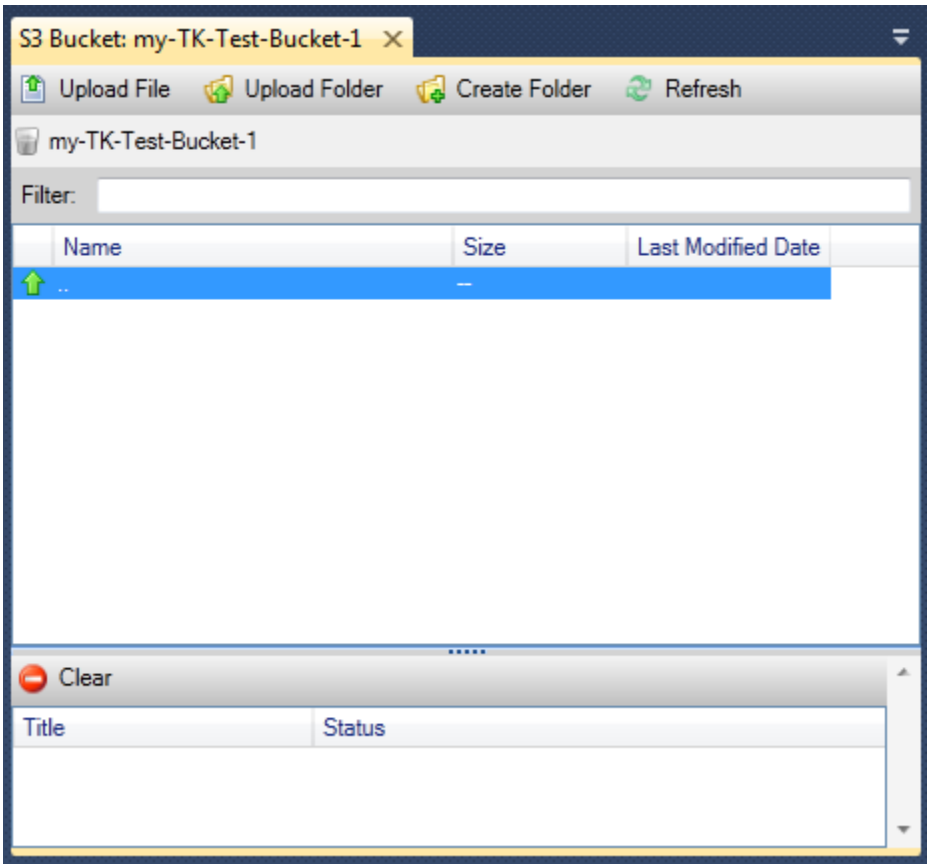
AWS Explorer에서 Amazon S3 버킷 관리

AWS 탐색기에서 Amazon S3 버킷에 대한 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴를 열 때 다음 작업을 사용할 수 있습니다.

찾아보기

버킷에 포함된 객체의 보기를 표시합니다. 여기에서 폴더를 생성하거나, 로컬 컴퓨터에서 파일이나 전체 디렉터리 및 폴더를 업로드할 수 있습니다. 하단 창에는 업로드 프로세스에 대한 상태 메시지가 표

시됩니다. 이러한 메시지를 지우려면 지우기 아이콘을 선택합니다. AWS 탐색기에서 버킷 이름을 두 번 클릭하여 버킷의이 보기에 액세스할 수도 있습니다.



속성

다음 작업을 수행할 수 있는 대화 상자를 표시합니다.

- 다음과 같은 범위의 Amazon S3 권한을 설정합니다.
  - 버킷 소유자
  - AWS에서 인증된 모든 사용자
  - 인터넷으로 액세스할 수 있는 모든 사용자
- 버킷에 대한 로깅을 설정합니다.
- 중복 감소 스토리지(RRS)를 사용하는 경우 데이터 손실이 발생하면 알림을 받을 수 있도록 Amazon Simple Notification Service(SNS)를 사용하여 알림을 설정합니다. RRS는 표준 스토리지보다 내구성이 낮지만 비용이 감소된 Amazon S3 스토리지 옵션입니다. 자세한 내용은 S3 FAQ를 참조하십시오.
- 버킷에서 데이터를 사용하여 정적 웹 사이트를 생성합니다.

정책

버킷에 대한 AWS Identity and Access Management (IAM) 정책을 설정할 수 있습니다. 자세한 내용은 IAM 설명서와 IAM 및 S3에 대한 사용 사례를 참조하십시오.

미리 서명된 URL 생성

버킷의 내용에 대한 액세스를 제공하기 위해 배포할 수 있는 기간이 제한된 URL을 생성할 수 있습니다. 자세한 내용은 미리 서명된 URL 생성 방법을 참조하십시오.

멀티파트 업로드 보기

멀티파트 업로드를 볼 수 있습니다. Amazon S3는 더 효율적인 업로드 프로세스를 위해 대량의 객체 업로드를 작게 나누는 것을 지원합니다. 자세한 내용은 S3 설명서의 멀티파트 업로드에 대한 설명을 참조하십시오.

삭제

버킷을 삭제할 수 있습니다. 빈 버킷만 삭제할 수 있습니다.

Amazon S3에 파일 및 폴더 업로드

AWS Explorer를 사용하여 로컬 컴퓨터에서 버킷으로 파일 또는 전체 폴더를 전송할 수 있습니다.

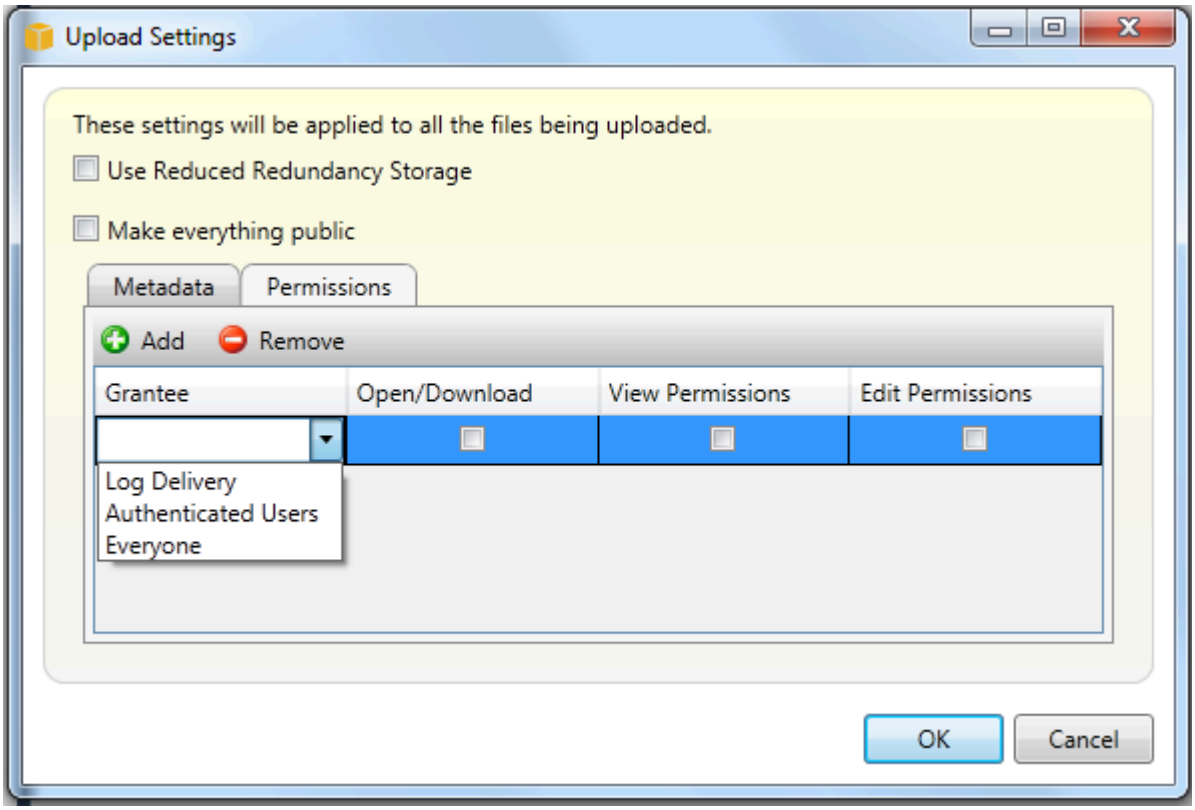
> **Note**
>
> 이미 Amazon S3 버킷에 있는 파일이나 폴더의 이름과 동일한 이름의 파일 또는 폴더를 업로드하면 경고 없이 업로드된 파일이 기존 파일을 덮어씁니다.

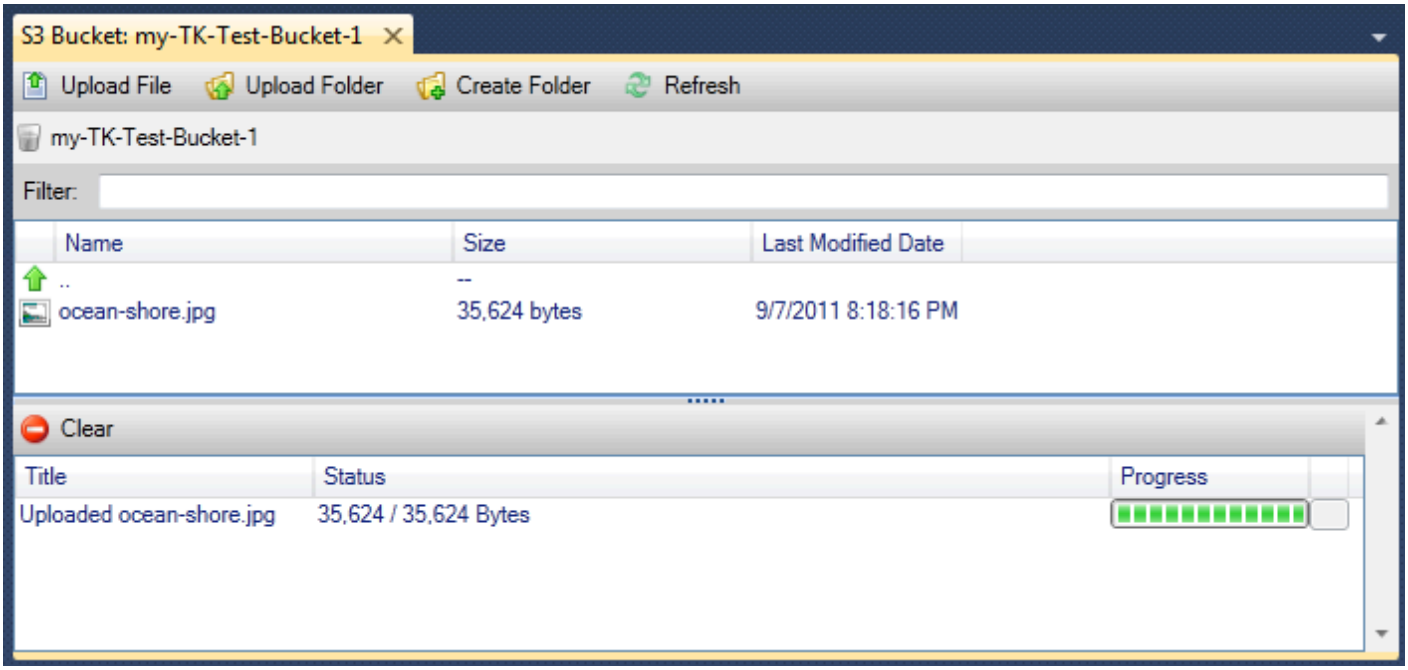
S3로 파일을 업로드하려면

1. AWS 탐색기에서 Amazon S3 노드를 확장하고 버킷을 두 번 클릭하거나 버킷의 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴를 열고 찾아보기를 선택합니다.
2. 버킷의 찾아보기 보기에서 파일 업로드 또는 Upload Folder(폴더 업로드)를 선택합니다.
3. File-Open(파일-열기) 대화 상자에서 업로드할 파일을 탐색하여 선택한 다음 열기를 선택합니다. 폴더를 업로드하는 경우 해당 폴더를 탐색하여 선택한 다음 열기를 선택합니다.

   Upload Settings(업로드 설정) 대화 상자를 사용하면 업로드하는 파일이나 폴더에 대한 메타데이터 및 권한을 설정할 수 있습니다. 모두 공개하기 확인란을 선택하는 것은 열기/다운로드 권한을 모

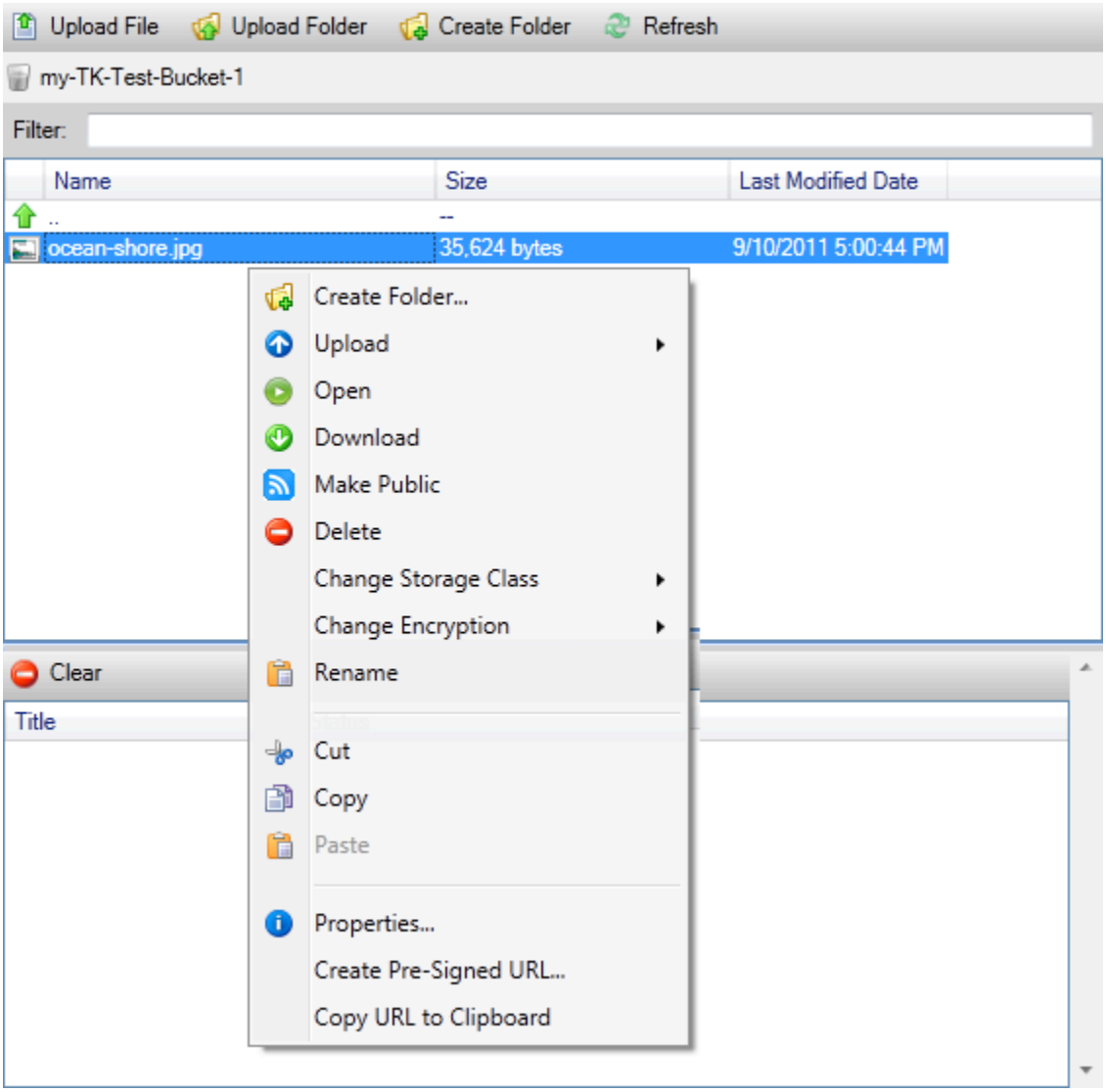
두로 설정하는 것과 같습니다. 업로드된 파일에 대해 Reduced Redundancy Storage를 사용하는 옵션을 선택할 수 있습니다.





AWS Toolkit for Visual Studio의 Amazon S3 파일 작업

Amazon S3 보기에서 파일을 선택하고 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴를 열면 해당 파일에 대한 다양한 작업을 수행할 수 있습니다.



폴더 만들기

현재 버킷에서 폴더를 생성할 수 있습니다. (폴더 생성 링크 선택과 동일)

업로드

파일이나 폴더를 업로드할 수 있습니다. (파일 업로드 또는 Upload Folder(폴더 업로드) 링크 선택과 동일)

Open

기본 브라우저에서 선택한 파일을 열려고 시도합니다. 파일 유형과 기본 브라우저의 기능에 따라 파일이 표시되지 않을 수도 있습니다. 대신 브라우저에서 간단하게 다운로드할 수 있습니다.

다운로드

선택한 파일을 다운로드할 수 있는 Folder-Tree(폴더-트리) 대화 상자를 엽니다.

퍼블릭으로 설정

선택한 파일에 대한 권한을 열기/다운로드 및 모든 사람로 설정합니다. (Upload Settings(업로드 설정) 대화 상자에서 Make everything public(전부 퍼블릭으로 설정) 확인란을 선택하는 것과 동일)

삭제

선택한 파일 또는 폴더를 삭제합니다. 파일이나 폴더를 선택하고 `Delete`를 눌러 삭제할 수도 있습니다.

스토리지 클래스 변경

스토리지 클래스를 표준 또는 RRS(Reduced Redundancy Storage)로 설정합니다. 현재 스토리지 클래스 설정을 보려면 속성을 선택합니다.

암호화 변경

파일에 대해 서버 측 암호화를 설정할 수 있습니다. 현재 암호화 설정을 보려면 속성을 선택합니다.

이름 바꾸기

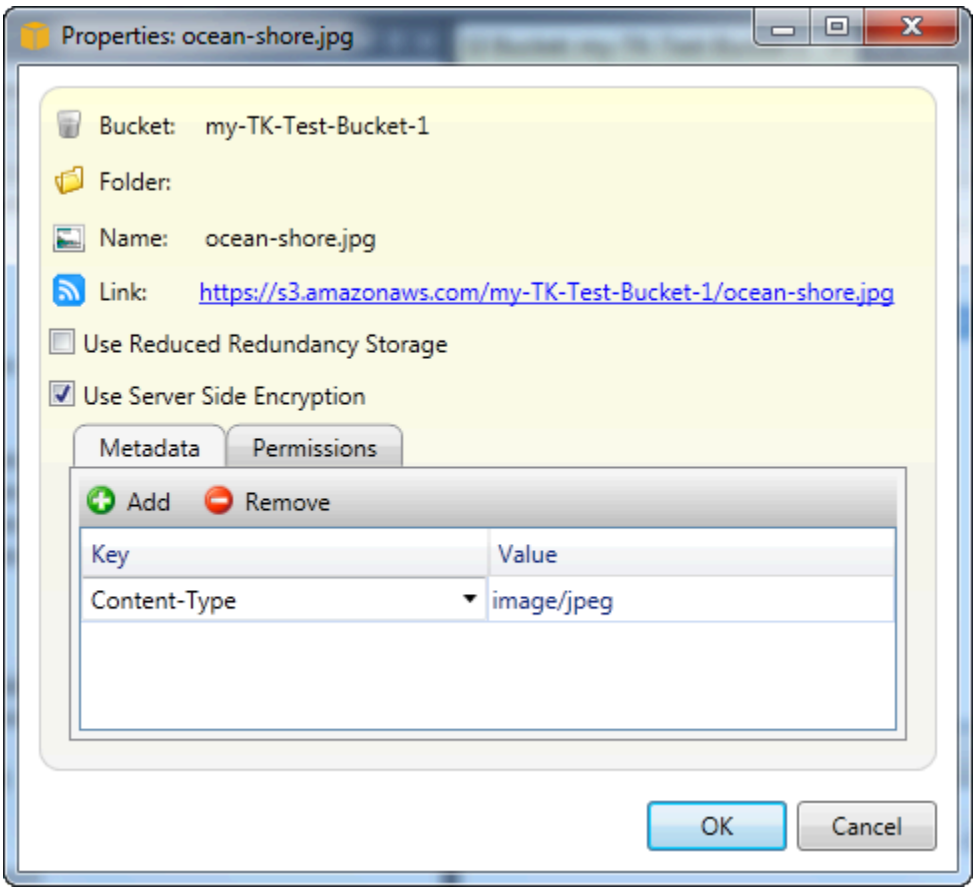
파일의 이름을 변경할 수 있습니다. 폴더의 이름을 바꿀 수 없습니다.

잘라내기 | 복사 | 붙여넣기

폴더 간 또는 버킷 간에 파일이나 폴더에 대해 잘라내기, 복사 및 붙여넣기를 수행할 수 있습니다.

속성

파일에 대한 메타데이터 및 권한을 설정하고, RRS(Reduced Redundancy Storage) 및 표준 간 파일에 대한 스토리지를 전환하고, 파일에 대해 서버 측 암호화를 설정할 수 있는 대화 상자를 표시합니다. 이 대화 상자에는 파일에 대한 https 링크도 표시됩니다. 이 링크를 선택하면 Toolkit for Visual Studio가 기본 브라우저에서 파일을 엽니다. 파일에 대해 열기/다운로드 및 모든 사람으로 설정된 권한이 있는 경우, 다른 사용자는 이 링크를 통해 파일에 액세스할 수 있습니다. 이 링크를 배포하는 대신 미리 서명된 URL을 생성하고 배포하는 것이 좋습니다.



미리 서명된 URL 생성

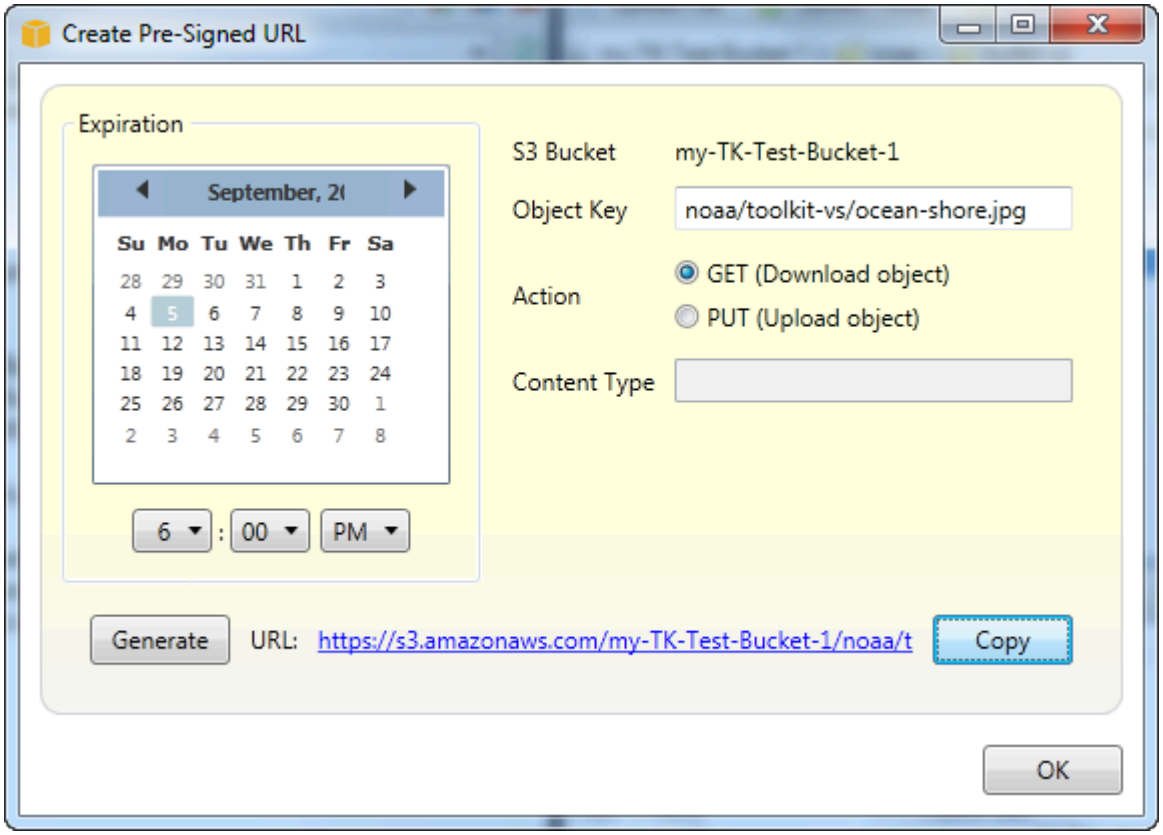
Amazon S3에 저장한 내용에 다른 사용자가 액세스하는 것을 허용하기 위해 배포할 수 있는 기간이 제한된 미리 서명된 URL을 생성할 수 있습니다.

미리 서명된 URL 생성 방법

버킷에 대해 또는 버킷의 파일에 대해 미리 서명된 URL을 생성할 수 있습니다. 그러면 다른 사용자가 이 URL을 사용하여 버킷이나 파일에 액세스할 수 있습니다. URL은 URL을 생성할 때 지정한 기간 이후에 만료됩니다.

미리 서명된 URL을 생성하려면

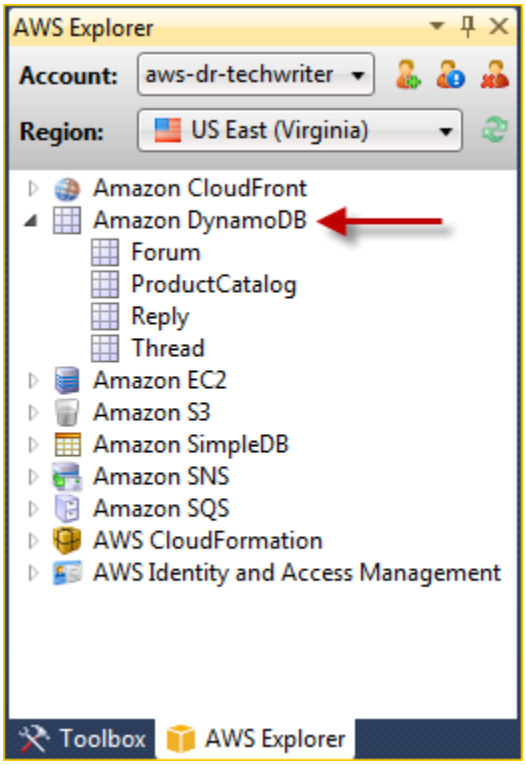
1. Create Pre-Signed URL(미리 서명된 URL 생성) 대화 상자에서 URL에 대한 만료 날짜와 시간을 설정합니다. 기본 설정은 현재 시간으로부터 한 시간입니다.
2. 생성 버튼을 선택합니다.
3. URL을 클립보드에 복사하려면 복사를 선택합니다.



AWS 탐색기에서 DynamoDB 사용

Amazon DynamoDB는 속도가 빠르고 확장성이 뛰어나며 비용 효과적인 비 관계형 데이터베이스 서비스입니다. DynamoDB는 기존 데이터 스토리지의 확장성 제한을 없애면서도 낮은 지연 시간과 예측 가능한 성능을 유지합니다. Toolkit for Visual Studio는 개발 컨텍스트에서 DynamoDB를 사용하기 위한 기능을 제공합니다. DynamoDB에 대한 자세한 정보는 Amazon Web Services 웹 사이트의 DynamoDB를 참조하세요.

Toolkit for Visual Studio에서 AWS Explorer는 활성와 연결된 모든 DynamoDB 테이블을 표시합니다 AWS 계정.



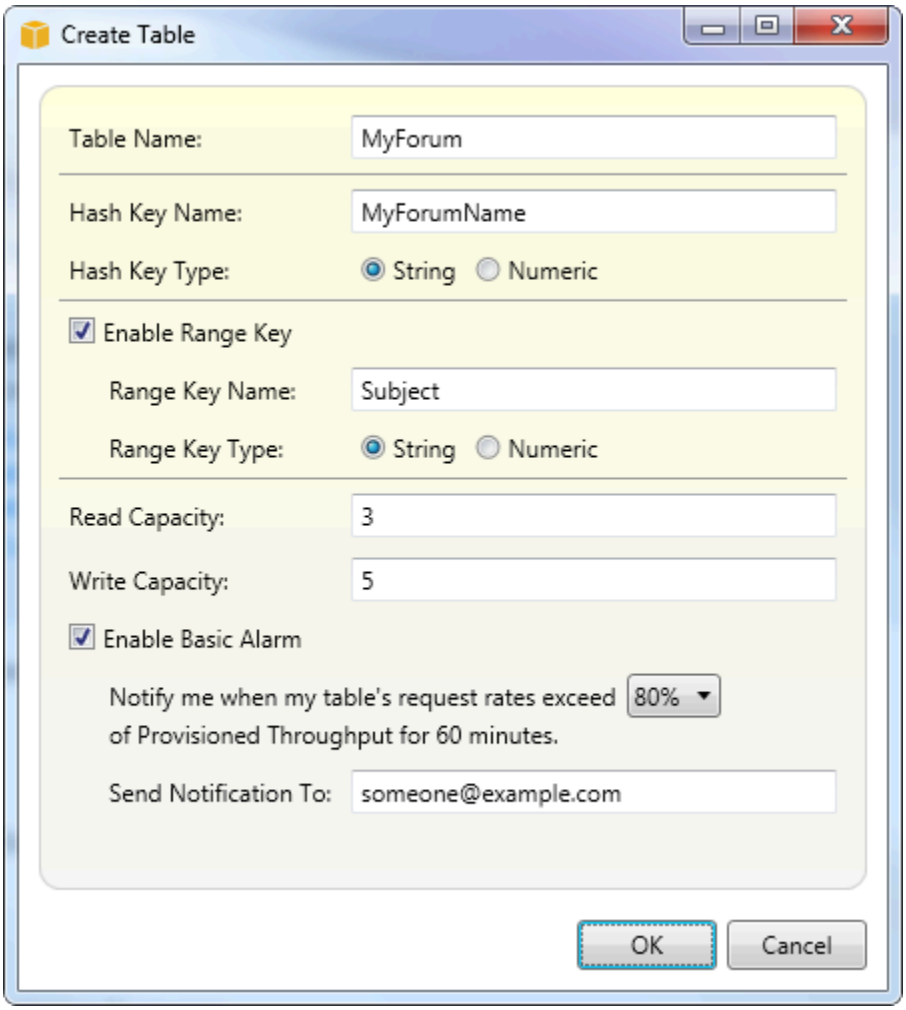
DynamoDB 테이블 생성

Toolkit for Visual Studio를 사용하여 DynamoDB 테이블을 생성할 수 있습니다.

AWS 탐색기에서 테이블을 생성하려면

1. AWS 탐색기에서 Amazon DynamoDB의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 연 다음 테이블 생성을 선택합니다.
2. 테이블 만들기 마법사의 테이블 이름에 테이블 이름을 입력합니다.
3. 해시 키 이름 필드에 기본 해시 키 속성을 입력하고 해시 키 유형 버튼에서 해시 키 유형을 선택하세요. DynamoDB는 해시 기본 키 속성을 사용하여 정렬되지 않은 해시 인덱스를 빌드하고, 범위 기본 키 속성을 사용하여 정렬된 범위 인덱스를 빌드합니다. 기본 해시 키 속성에 대한 자세한 정보는 Amazon DynamoDB 개발자 안내서의 기본 키 섹션을 참조하세요.
4. (선택 사항) Enable Range Key(범위 키 활성화)를 선택합니다. Range Key Name(범위 키 이름) 필드에 범위 키 속성을 입력한 다음 Range Key Type(범위 키 유형) 버튼에서 범위 키 유형을 선택합니다.
5. Read Capacity(읽기 용량) 필드에 읽기 용량 단위 수를 입력합니다. Write Capacity(쓰기 용량) 필드에 쓰기 용량 단위 수를 입력합니다. 최소 3개의 읽기 용량 단위와 5개의 쓰기 용량 단위를 지정해야 합니다. 읽기 및 쓰기 용량 단위에 대한 자세한 내용은 DynamoDB에서 프로비저닝된 처리량을 참조하십시오.

6. (선택 사항) Enable Basic Alarm(기본 알림 활성화)을 선택하여 테이블의 요청 비율이 너무 높으면 사용자에게 알림을 보내도록 합니다. 알림을 보내기 전까지 초과되어야 하는 60분당 프로비저닝된 처리율을 선택하십시오. 알림 받을 대상에 이메일 주소를 입력합니다.
7. 확인을 클릭하여 테이블을 만듭니다.

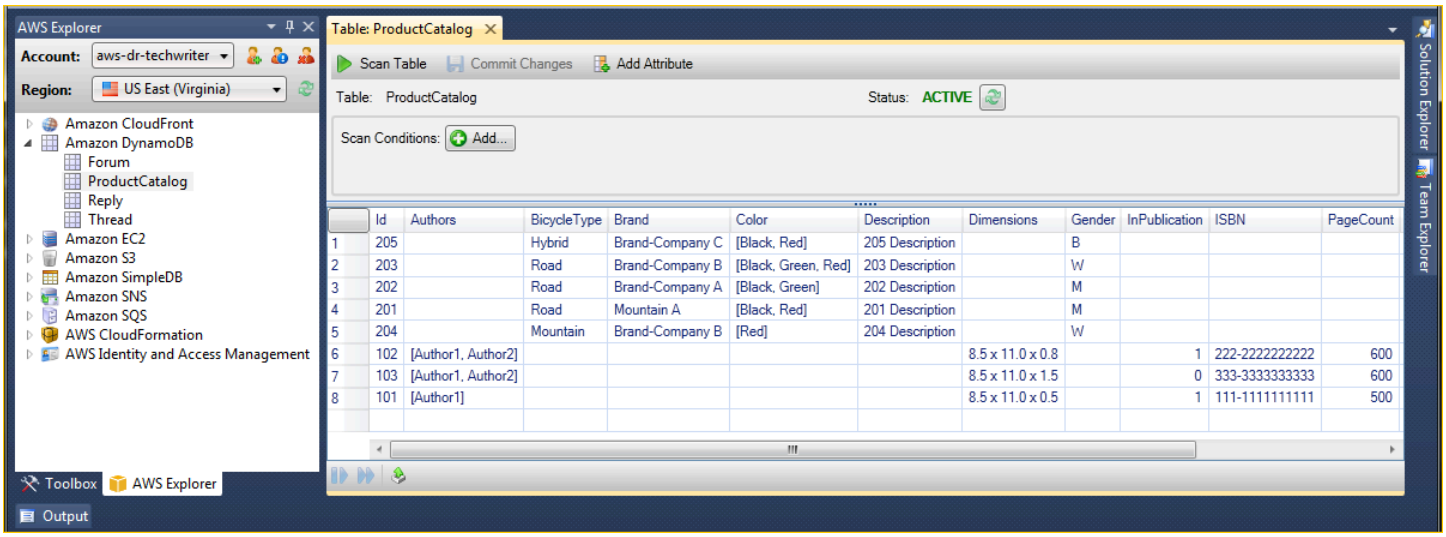


DynamoDB 테이블에 대한 자세한 정보는 데이터 모델 개념 - 테이블, 항목 및 속성을 참조하세요.

DynamoDB 테이블을 그리드로 보기

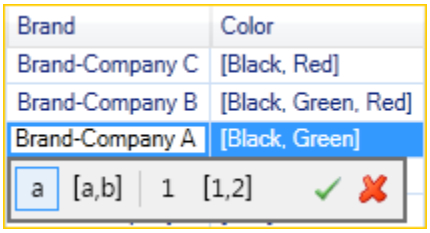
DynamoDB 테이블 중 하나의 그리드 보기를 열려면 AWS 탐색기에서 테이블에 해당하는 하위 노드를 두 번 클릭합니다. 그리드 보기에서 테이블에 저장된 항목, 속성 및 값을 볼 수 있습니다. 각 열은 테이블의 항목에 해당합니다. 테이블 열은 속성에 해당합니다. 테이블의 각 셀에는 항목에 대한 속성과 연결된 값이 있습니다.

속성에는 문자열 값 또는 숫자 값이 있을 수 있습니다. 일부 속성에는 문자열 또는 숫자의 집합으로 구성된 값이 있습니다. 집합 값은 대괄호로 묶여 있으며 쉼표로 구분된 목록으로 표시됩니다.

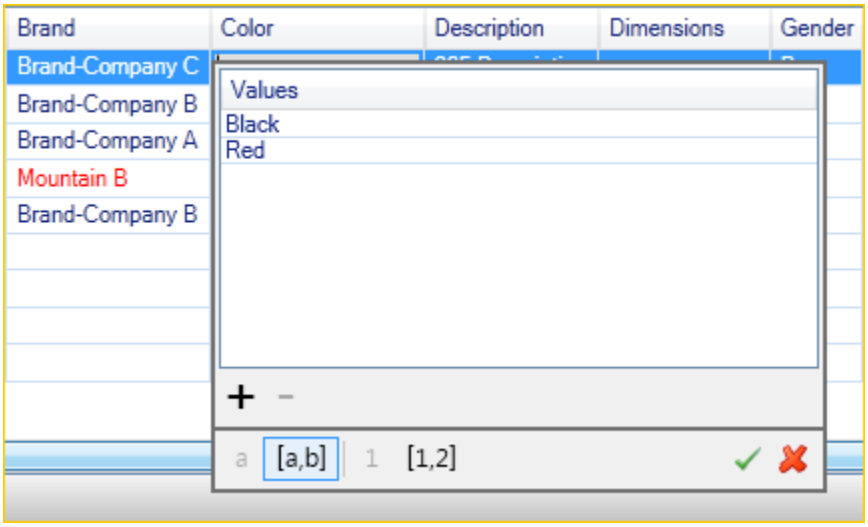


속성 및 값 편집/추가

셀을 두 번 클릭하여 항목의 해당 속성에 대한 값을 편집할 수 있습니다. 집합 값 속성의 경우 집합에서 개별 값을 추가하거나 삭제할 수도 있습니다.



일부 제한이 있기는 하지만 속성 값을 변경할 때 속성에 대한 값 형식도 변경할 수 있습니다. 예를 들어, 모든 숫자 값을 문자열 값으로 전환할 수 있습니다. 125와 같이 콘텐츠가 숫자인 문자열 값을 사용하는 경우 셀 편집기로 값 형식을 문자열에서 숫자로 변환할 수 있습니다. 또한 단일 값을 설정 값으로 변환할 수 있습니다. 그러나 일반적으로 집합 값을 단일 값으로 변환할 수 없습니다. 단, 집합 값의 집합에 실제로 요소가 하나만 있는 경우는 제외됩니다.

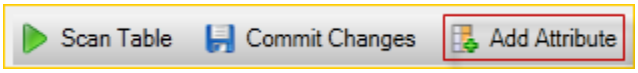


속성 값을 편집한 후 녹색 확인 표시를 선택하여 변경 사항을 확인하십시오. 변경 사항을 취소하려면 빨간색 X를 선택하십시오.

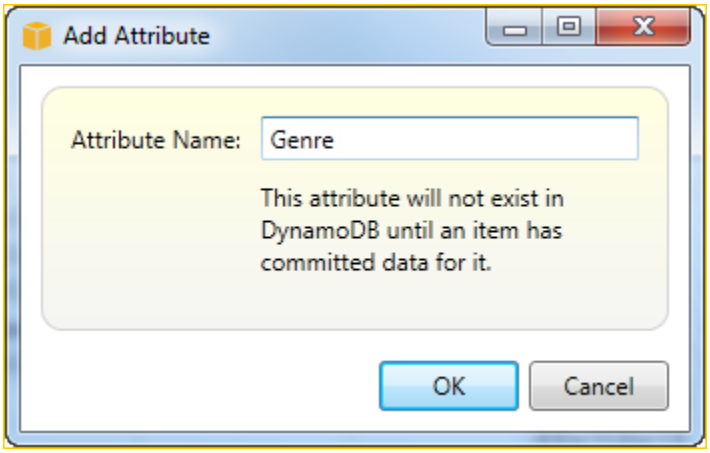
변경 사항을 확인한 후 속성 값이 빨간색으로 표시됩니다. 이는 속성이 업데이트되었지만 아직 새 값이 DynamoDB 데이터베이스에 작성되지 않았음을 나타냅니다. DynamoDB에 변경 사항을 다시 작성하려면 변경 사항 커밋을 선택하세요. 변경 사항을 취소하려면 Scan Table(테이블 스캔)을 선택하고 도구 키트가 스캔 전에 변경 사항을 커밋할지 여부를 묻는 경우 아니요를 선택합니다.

속성 추가

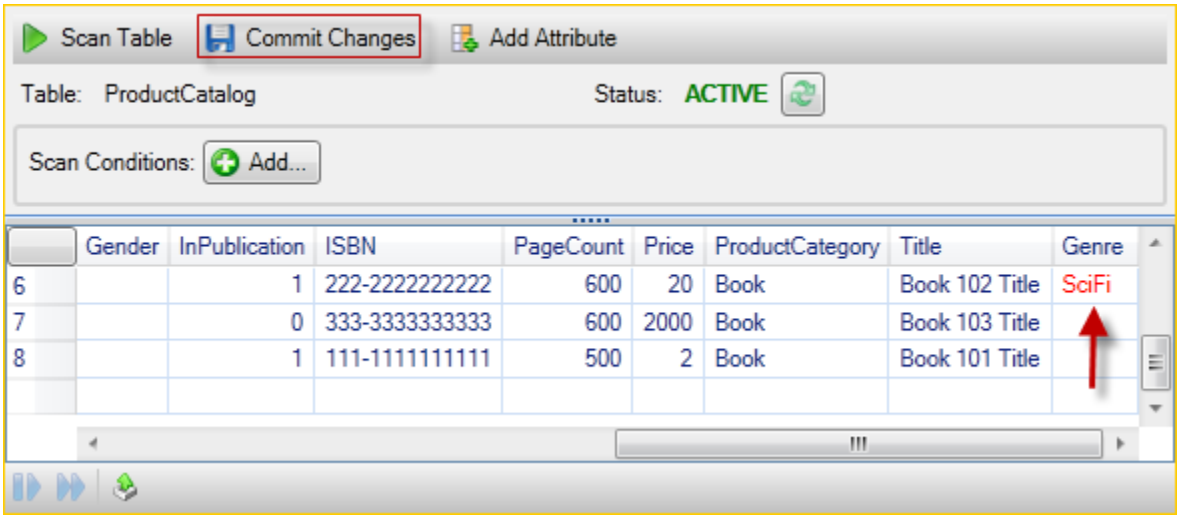
또한 그리드 보기에서 속성을 테이블에 추가할 수 있습니다. 새 속성을 추가하려면 속성 추가를 선택합니다.



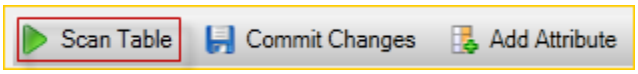
속성 추가 대화 상자에서 속성 이름을 입력한 다음 확인을 선택합니다.



새 속성을 테이블의 일부로 만들려면 하나 이상의 항목에 값을 추가한 다음 변경 사항 커밋 버튼을 선택해야 합니다. 새 속성을 취소하려면 변경 사항 커밋을 선택하지 않고 테이블의 그리드 보기를 닫으면 됩니다.



DynamoDB 테이블 스캔

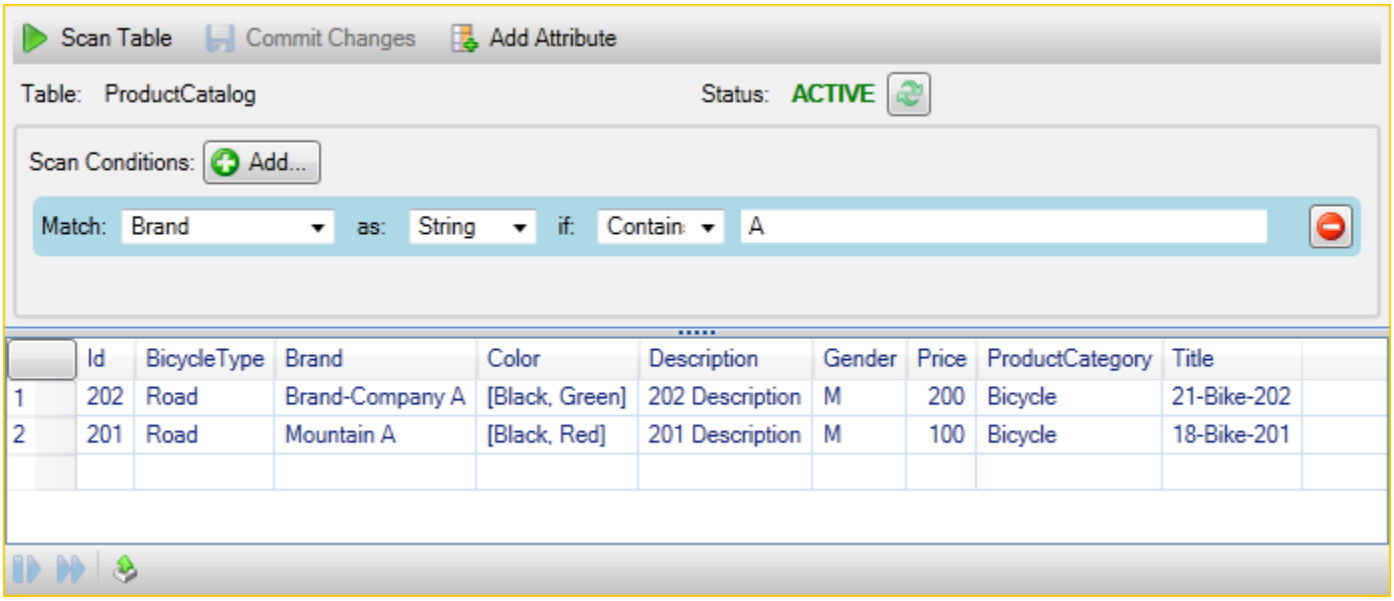


툴킷의 DynamoDB 테이블에서 스캔을 수행할 수 있습니다. 스캔에서 기준 집합을 정의하면 스캔이 테이블에서 기준과 일치하는 모든 항목을 반환합니다. 스캔은 비용이 많이 드는 작업이므로 테이블에서 우선 순위가 높은 프로덕션 트래픽을 방해하지 않도록 주의해야 합니다. 스캔 작업 사용에 대한 자세한 정보는 Amazon DynamoDB 개발자 안내서를 참조하세요.

AWS 탐색기에서 DynamoDB 테이블에서 스캔을 수행하려면

1. 그리드 보기에서 Scan Conditions: Add(스캔 조건: 추가) 버튼을 선택합니다.
2. Scan 절 편집기에서 일치시킬 속성, 속성 값 해석 방법(문자열, 숫자, 설정 값), 속성 값 일치 방법(예: Begins With 또는 Contains) 및 일치해야 할 리터럴 값을 선택합니다.
3. 필요한 경우 검색에 사용할 스캔 절을 추가합니다. 스캔에서는 모든 스캔 절의 기준과 일치하는 항목만 반환됩니다. 스캔은 문자열 값과 일치하는 경우 대/소문자 구분 비교를 수행합니다.
4. 그리드 보기 상단의 버튼 모음에서 Scan Table(테이블 스캔)을 선택합니다.

스캔 절을 제거하려면 각 절의 오른쪽에 흰색 줄이 있는 빨간색 버튼을 선택합니다.



모든 항목이 포함된 테이블 보기로 돌아가려면 모든 스캔 절을 제거하고 Scan Table(테이블 스캔)을 다시 선택합니다.

스캔 결과 페이지 매김

보기 아래쪽에 버튼 3개가 있습니다.

처음 두 개의 파란색 버튼은 스캔 결과에 대한 페이지 매김을 제공합니다. 첫 번째 버튼은 결과를 1페이지씩 표시합니다. 두 번째 버튼은 결과를 10페이지씩 표시합니다. 이 컨텍스트에서 페이지는 1MB의 콘텐츠와 동일합니다.

스캔 결과를 CSV로 내보내기

세 번째 버튼은 현재 스캔의 결과를 CSV 파일로 내보냅니다.

Visual Studio Team Explorer에서 AWS CodeCommit 사용

AWS Identity and Access Management(IAM) 사용자 계정을 사용하여 Git 보안 인증 정보를 생성하고, 해당 보안 인증 정보를 사용하여 Team Explorer 내에서 리포지토리를 생성하고 복제할 수 있습니다.

AWS CodeCommit에 대한 자격 증명 유형

대부분의 AWS Toolkit for Visual Studio 사용자는 액세스 키 및 비밀 키가 포함된 AWS 보안 인증 정보 프로필 설정에 대해 잘 파악하고 있습니다. 이러한 보안 인증 정보 프로필은 AWS 탐색기에서 Amazon

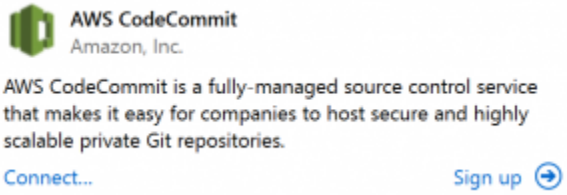
S3 버킷을 나열하거나 Amazon EC2 인스턴스를 시작하는 등 Toolkit for Visual Studio에서 서비스 API에 대한 호출을 활성화하는 데 사용됩니다. Team Explorer와 AWS CodeCommit의 통합에서도 이러한 자격 증명 프로파일을 사용합니다. 그러나 Git 자체로 작업하려면 추가 자격 증명이 필요합니다. 특히 HTTPS 연결에 대해서는 Git 자격 증명이 필요합니다. AWS CodeCommit 사용 설명서의 Git 보안 인증 정보를 사용하는 HTTPS 사용자 설정에서 이러한 보안 인증 정보(사용자 이름 및 암호)에 대한 내용을 읽을 수 있습니다.

IAM 사용자 계정에 대해서만 AWS CodeCommit용 Git 보안 인증 정보를 생성할 수 있습니다. 루트 계정에 대해서는 해당 자격 증명을 생성할 수 없습니다. 서비스에 대해 이러한 자격 증명 집합을 최대 2개까지 생성할 수 있습니다. 자격 증명 집합을 비활성 상태로 표시하는 경우에도 비활성 집합이 두 개의 제한에 가산됩니다. 언제든지 자격 증명을 삭제하고 다시 생성할 수 있습니다. Visual Studio 내에서 AWS CodeCommit을 사용할 경우 리포지토리를 생성하고 나열하는 등 서비스 자체에 대한 작업에 기존 AWS 보안 인증 정보가 사용됩니다. AWS CodeCommit에 호스팅된 실제 Git 리포지토리로 작업할 경우 Git 자격 증명을 사용합니다.

AWS CodeCommit에 대한 지원의 일환으로 Toolkit for Visual Studio는 이러한 Git 보안 인증 정보를 자동으로 생성 및 관리하며, 해당 보안 인증 정보를 AWS 보안 인증 정보 프로필에 연결합니다. Team Explorer 내에서 Git 작업을 수행하기 위해 현재 올바른 자격 증명 집합을 보유했는지에 대해 걱정할 필요가 없습니다. AWS 보안 인증 정보 프로필을 통해 Team Explorer에 연결하면 Git 원격 작업을 수행할 때마다 연결된 Git 보안 인증 정보가 자동으로 사용됩니다.

에 연결AWS CodeCommit

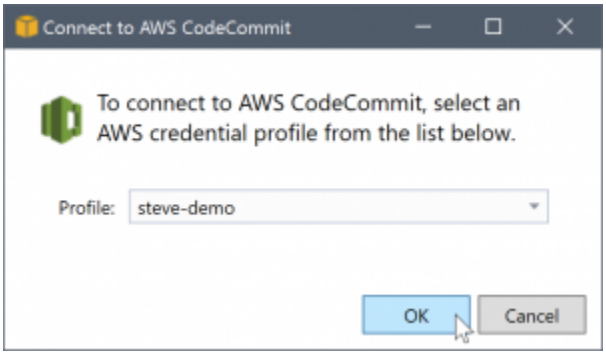
Visual Studio 2015 이상에서 Team Explorer 창을 열면 Manage Connections(연결 관리)의 Hosted Service Providers(호스팅된 서비스 공급업체)에 AWS CodeCommit 항목이 표시됩니다.



가입을 선택하면 브라우저 창에 Amazon Web Services 홈 페이지가 열립니다. 연결을 선택할 때 발생하는 사항은 Amazon Web Services가 AWS 액세스 키 및 비밀 키가 포함된 보안 인증 정보 프로필을 찾고 활성화하여 사용자 대신 AWS에 대한 호출을 수행할 수 있는지 여부에 따라 달라집니다. Toolkit for Visual Studio가 로컬로 저장된 보안 인증 정보를 찾을 수 없는 경우 IDE에 표시되는 새 시작하기 페이지를 사용하여 보안 인증 정보 프로필을 설정할 수도 있습니다. 또는 AWS Tools for Windows PowerShell, AWS CLI를 사용하고, 사용할 Toolkit for Visual Studio에 사용 가능한 AWS 보안 인증 정보 프로필을 사용할 수도 있습니다.

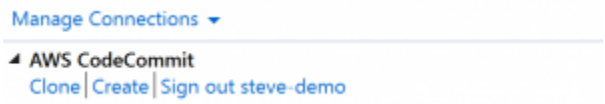
연결을 선택하면 Toolkit for Visual Studio가 연결에서 사용할 보안 인증 정보 프로필 찾기 프로세스를 시작합니다. Toolkit for Visual Studio가 보안 인증 정보 프로필을 찾을 수 없는 경우 AWS 계정 계정에 대한 액세스 키 및 비밀 키를 입력하도록 사용자를 초대하는 대화 상자가 열립니다. 루트 자격 증명이 아닌 IAM 사용자 계정을 사용하는 것이 좋습니다. 또한 이전에 언급한 대로 최종적으로 필요한 Git 자격 증명은 IAM 사용자에 대해서만 생성될 수 있습니다. 액세스 키 및 보안 키가 제공되고 자격 증명 프로파일이 생성되면 Team Explorer와 AWS CodeCommit 간의 연결을 사용할 준비가 되었습니다.

Toolkit for Visual Studio가 둘 이상의 AWS 보안 인증 정보 프로필을 찾은 경우 Team Explorer 내에서 사용할 계정을 선택하라는 메시지가 나타납니다.



보안 인증 정보 프로필이 하나만 있는 경우 Toolkit for Visual Studio는 프로필 선택 대화 상자를 우회하고 즉시 연결됩니다.

자격 증명 프로파일을 통해 Team Explorer와 AWS CodeCommit 간의 연결이 설정되면 초대 대화 상자가 닫히고 연결 패널이 표시됩니다.

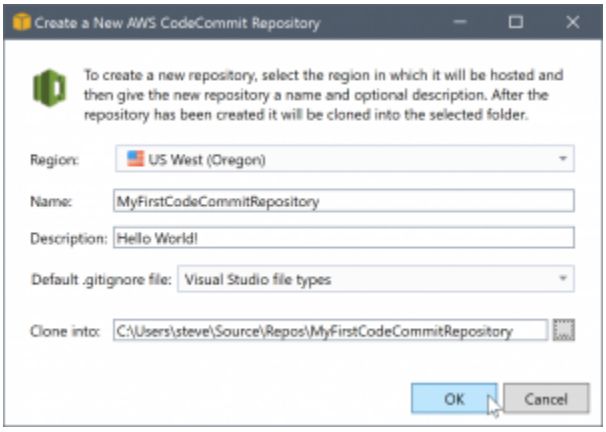


로컬로 복제된 리포지토리가 없으므로 패널에는 수행할 수 있는 작업인 Clone(복제), 생성 및 로그아웃만 표시됩니다. 다른 공급자와 마찬가지로 Team Explorer의 AWS CodeCommit은 특정 시점에서 단일 AWS 보안 인증 정보 프로필에만 바인딩될 수 있습니다. 계정을 전환하려면 다른 연결을 사용하여 새 연결을 시작할 수 있도록 로그아웃을 사용하여 연결을 제거합니다.

이제 연결을 설정했으므로 생성 링크를 클릭하여 리포지토리를 생성할 수 있습니다.

리포지토리 생성

생성 링크를 클릭하면 새 AWS CodeCommit 리포지토리 생성 대화 상자가 열립니다.



AWS CodeCommit 리포지토리는 리전에 따라 구성되므로 리전에서 리포지토리를 호스팅할 리전을 선택할 수 있습니다. 목록에는 AWS CodeCommit이 지원되는 모든 리전이 있습니다. 새 리포지토리에 대해 이름(필수) 및 설명(선택 사항)을 제공합니다.

대화 상자의 기본 동작은 새 리포지토리의 폴더 위치에 리포지토리 이름으로 접미사를 지정하는 것입니다(이름을 입력하면 폴더 위치도 업데이트됨). 다른 폴더 이름을 사용하려면 리포지토리 이름 입력을 완료한 후 Clone into(다음으로 복제) 폴더 경로를 편집합니다.

리포지토리에 대한 초기 `.gitignore` 파일을 자동으로 생성하도록 선택할 수도 있습니다. AWS Toolkit for Visual Studio는 Visual Studio 파일 유형에 대해 내장된 기본값을 제공합니다. 파일이 없거나 리포지토리에 재사용할 사용자 지정 기존 파일을 사용하도록 선택할 수도 있습니다. 목록에서 Use custom(사용자 지정 사용)을 선택하고 사용할 사용자 지정 파일을 탐색합니다.

리포지토리 이름과 위치가 제공된 경우 확인을 클릭하여 리포지토리 생성을 시작할 준비가 되었습니다. Toolkit for Visual Studio에서는 서비스가 리포지토리를 생성한 다음 새 리포지토리를 로컬로 복제하고, 해당 리포지토리를 사용할 경우 .gitignore 파일에 대한 초기 커밋을 추가합니다. 이 시점에서 Git 원격 작업을 시작하므로 이제 Toolkit for Visual Studio는 이전에 설명한 Git 보안 인증 정보에 액세스해야 합니다.

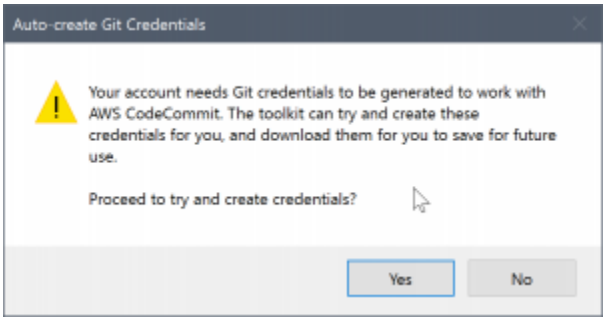
Git 자격 증명 설정

지금까지는 AWS 액세스 키 및 비밀 키를 사용하여 서비스가 리포지토리를 생성하도록 요청했습니다. 이제 Git 자체로 작업하여 실제 복제 작업을 수행해야 합니다. Git는 AWS 액세스 키와 비밀 키를 이해하지 못합니다. 대신 원격 작업과 HTTPS 연결에 사용할 사용자 이름과 암호 자격 증명을 Git에 제공해야 합니다.

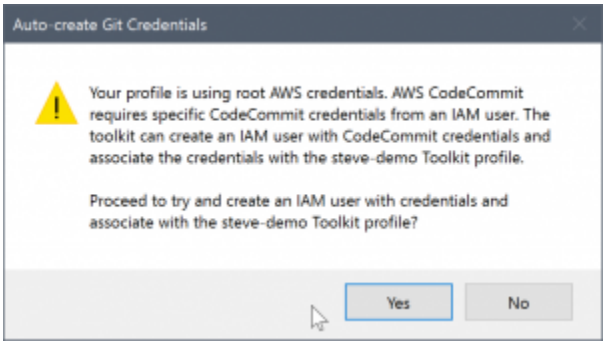
Git 자격 증명 설정에 언급된 대로 사용할 Git 자격 증명은 IAM 사용자와 연결되어야 합니다. 루트 자격 증명에 대해서는 해당 자격 증명을 생성할 수 없습니다. 항상 IAM 사용자 액세스 키 및 비밀 키를 포함하고 루트 키를 포함하지 않도록 AWS 보안 인증 정보 프로필을 설정해야 합니다. Toolkit for Visual

Studio는 AWS CodeCommit에 대한 Git 보안 인증 정보를 설정하고 해당 보안 인증 정보을 이전에 Team Explorer에 연결하는 데 사용한 AWS 보안 인증 정보 프로필과 연결하려고 시도할 수 있습니다.

새 AWS CodeCommit 리포지토리 생성 대화 상자에서 확인을 선택하고 성공적으로 리포지토리를 생성하면, Toolkit for Visual Studio에서는 Team Explorer에서 연결된 AWS 보안 인증 정보 프로필을 확인하여 AWS CodeCommit에 대한 Git 보안 인증 정보가 있는지 및 프로필과 로컬로 연결되었는지를 확인합니다. 그러한 경우 Toolkit for Visual Studio는 새 리포지토리에 대해 복제 작업을 시작하도록 Team Explorer에 지시합니다. Git 보안 인증 정보를 로컬로 사용할 수 없는 경우 Toolkit for Visual Studio는 Team Explorer에서 연결에 사용된 계정 보안 인증 정보의 유형을 확인합니다. 자격 증명이 권장하는 대로 IAM 사용자를 위한 것일 경우 다음 메시지가 표시됩니다.

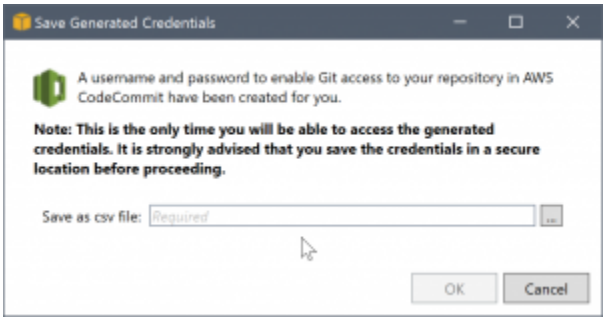


자격 증명이 루트 자격 증명인 경우 다음 메시지가 대신 표시됩니다.



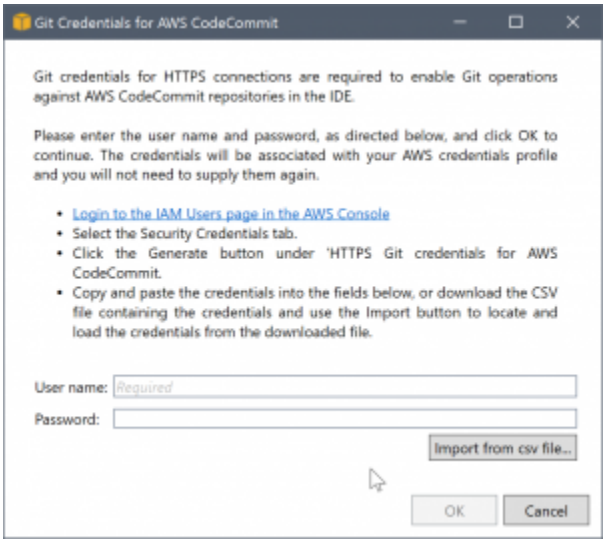
두 경우 모두 Toolkit for Visual Studio가 필요한 Git 보안 인증 정보를 생성하는 작업을 수행합니다. 첫 번째 시나리오에서 생성에 필요한 사항은 IAM 사용자에 대한 Git 자격 증명 집합입니다. 루트 계정을 사용하고 있는 경우 Toolkit for Visual Studio는 먼저 IAM 사용자를 생성한 다음 해당 새 사용자에 대한 Git 보안 인증 정보 생성을 진행합니다. Toolkit for Visual Studio가 새 사용자를 생성해야 하는 경우 해당 새 사용자 계정에 AWS CodeCommit Power User 관리형 정책이 적용됩니다. 이 정책은 AWS CodeCommit에 대해서만 액세스를 허용하며 AWS CodeCommit을 통해 리포지토리 삭제를 제외한 모든 작업을 수행합니다.

자격 증명을 생성 중인 경우 한 번만 볼 수 있습니다. 따라서, Toolkit for Visual Studio에는 계속 진행하기 전에 새로 생성된 보안 인증 정보를 `.csv` 파일로 저장하라는 메시지가 표시됩니다.



이는 매우 권장되는 사항으로 안전한 위치에 저장해야 합니다!

Toolkit for Visual Studio가 보안 인증 정보를 자동으로 생성할 수 없는 경우가 있을 수 있습니다. 예를 들어, 이미 AWS CodeCommit에 대한 Git 보안 인증 정보 집합의 최대 수(2개)를 생성했거나 Toolkit for Visual Studio가 작업을 수행할 수 있는 프로그래밍 방식의 권한이 충분하지 않을 수 있습니다(IAM 사용자로 로그인한 경우). 이러한 경우 AWS Management Console에 로그인하여 보안 인증 정보를 관리하거나, 관리자로부터 보안 인증 정보를 받을 수 있습니다. 그런 다음 해당 정보를 Toolkit for Visual Studio에 표시되는 AWS CodeCommit에 대한 Git 보안 인증 정보 대화 상자에 입력할 수 있습니다.

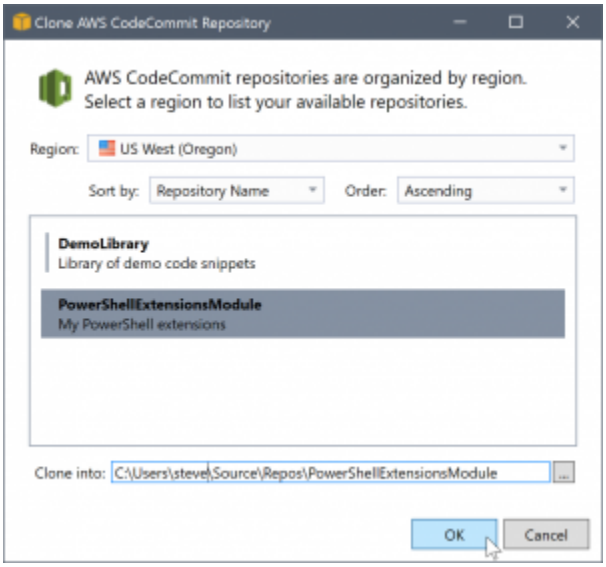


이제 Git에 대한 자격 증명을 사용할 수 있으므로 새 리포지토리에 대한 복제 작업이 진행됩니다(Team Explorer 내의 작업에 대한 진행 표시 참조). 기본 `.gitignore` 파일을 적용하도록 선택한 경우 해당 파일이 '초기 커밋'이라는 설명과 함께 리포지토리에 커밋됩니다.

이걸로 끝이며 Team Explorer 내에서 자격 증명이 설정되고 리포지토리가 생성됩니다. 필수 보안 인증 정보가 마련되면 향후 새 리포지토리를 생성할 때 새 AWS CodeCommit 리포지토리 생성 대화 상자 자체만 표시됩니다.

리포지토리 복제

기존 리포지토리를 복제하려면 Team Explorer의 AWS CodeCommit에 대한 연결 패널로 돌아갑니다. 복제 링크를 클릭하여 AWS CodeCommit 리포지토리 복제 대화 상자를 연 다음 복제할 리포지토리와 디스크에서 리포지토리를 배치할 위치를 선택하세요.



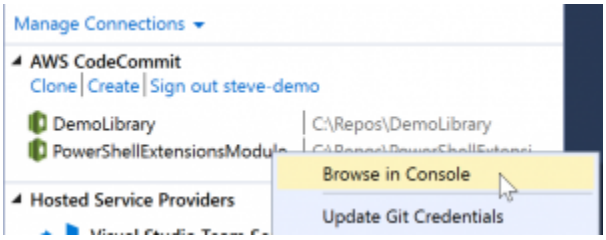
리전을 선택하면 Toolkit for Visual Studio에서는 해당 리전에서 사용할 수 있는 리포지토리를 검색하고 대화 상자의 가운데 목록 부분에 표시하도록 서비스에 쿼리합니다. 각 리포지토리의 이름과 설명(선택 사항)도 표시됩니다. 리포지토리 이름이나 마지막 수정 날짜를 기준으로 정렬하고 오름차순 또는 내림차순으로 정렬하도록 목록을 재정렬할 수 있습니다.

리포지토리를 선택하면 리포지토리를 복제할 위치를 선택할 수 있습니다. 이 기본값은 Team Explorer에 대해 다른 플러그인에서 사용된 동일한 리포지토리 위치이지만 다른 위치를 찾거나 입력할 수 있습니다. 기본적으로 리포지토리 이름이 선택한 경로에 접미사로 추가됩니다. 그러나 특정 경로를 원하는 경우 폴더를 선택한 다음 텍스트를 편집하면 됩니다. 상자에 있는 텍스트가 무엇이든지 간에 확인을 클릭하면 복제된 리포지토리가 있는 폴더가 됩니다.

리포지토리와 폴더 위치를 선택한 다음 확인을 클릭하여 복제 작업으로 진행합니다. 리포지토리 생성과 마찬가지로 Team Explorer에 보고된 복제 작업의 진행을 볼 수 있습니다.

리포지토리 작업

리포지토리를 복제하거나 생성할 때 연결에 대한 로컬 리포지토리가 Team Explorer의 연결 패널에 있는 작업 링크 아래에 나열됩니다. 이러한 항목은 콘텐츠를 찾아보기 위해 리포지토리에 편리하게 액세스할 수 있는 방법을 제공합니다. 리포지토리를 마우스 오른쪽 버튼으로 클릭하고 Browse in Console(콘솔에서 찾아보기)을 클릭하면 됩니다.



Update Git Credentials(Git 자격 증명 업데이트)를 사용하여 자격 증명 프로파일과 연결된 저장된 Git 자격 증명을 업데이트할 수도 있습니다. 이는 자격 증명을 교체한 경우 유용합니다. 이 명령은 새 보안 인증 정보를 입력하거나 가져올 수 있는 AWS CodeCommit에 대한 Git 보안 인증 정보 대화 상자를 엽니다.

리포지토리에 대한 Git 작업은 예상대로 작동합니다. 로컬 커밋을 만들고 공유할 준비가 된 경우 Team Explorer에서 [Sync] 옵션을 사용할 수 있습니다. Git 보안 인증 정보가 이미 로컬로 저장되고 연결된 AWS 보안 인증 정보 프로필과 연결되었으므로 AWS CodeCommit 원격 작업에 대해 보안 인증 정보를 다시 제공하라는 메시지가 표시되지 않습니다.

Visual Studio에서 CodeArtifact 사용

AWS CodeArtifact는 완전관리형 아티팩트 리포지토리 서비스로서, 이를 통해 조직은 애플리케이션 개발에 사용되는 소프트웨어 패키지를 안전하게 저장하고 공유할 수 있습니다. CodeArtifact는 NuGet 및 .NET Core CLI 및 Visual Studio와 같은 인기 있는 빌드 도구 및 패키지 관리자와 함께 사용할 수 있습니다. 또한 NuGet.org와 같은 외부 공개 리포지토리에서 패키지를 가져오도록 CodeArtifact를 구성할 수 있습니다.

CodeArtifact에서는 패키지가 리포지토리에 저장되고 도메인 내에 저장됩니다. AWS Toolkit for Visual Studio는 CodeArtifact 리포지토리를 사용하여 Visual Studio를 간단하게 구성할 수 있으므로 CodeArtifact에서 직접 그리고 NuGet.org 모두에서 Visual Studio의 패키지를 쉽게 사용할 수 있습니다.

CodeArtifact 리포지토리를 NuGet 패키지 소스로 추가

CodeArtifact의 패키지를 사용하려면 Visual Studio의 NuGet 패키지 관리자에서 리포지토리를 패키지 소스로 추가해야 합니다.

리포지토리를 패키지 소스로 추가

1. AWS 탐색기에서 AWS CodeArtifact 노드의 해당 리포지토리로 이동하세요.
2. 추가하려는 리포지토리의 컨텍스트 메뉴(마우스 오른쪽 버튼으로 클릭)를 연 다음, NuGet 소스 엔드포인트 복사를 선택하세요.

3. 도구 > 옵션 메뉴의 NuGet 패키지 관리자 노드 아래에 있는 패키지 소스로 이동하세요.
4. 패키지 소스에서 더하기 기호(+)를 선택하고 이름을 편집한 다음 이전에 복사한 NuGet 소스 엔드포인트 URL을 소스 필드에 붙여넣습니다.
5. 새로 추가한 패키지 소스 옆의 확인란을 선택하여 활성화하세요.

> **Note**
>
> CodeArtifact에 Nuget.org에 대한 외부 연결을 추가하고 Visual Studio에서 nuget.org 패키지 소스를 비활성화하는 것이 좋습니다. 외부 연결을 사용하는 경우 Nuget.org에서 가져온 모든 종속성이 CodeArtifact에 저장됩니다. 어떤 이유로든 NuGet.org가 다운되더라도 필요한 패키지는 계속 사용할 수 있습니다. 외부 연결에 대한 자세한 정보는 AWS CodeArtifact 사용 설명서의 외부 연결 추가를 참조하세요.

6. 확인을 선택하여 메뉴를 닫습니다.

Visual Studio에서 CodeArtifact를 사용하는 방법에 대한 자세한 정보는 AWS CodeArtifact 사용 설명서의 Visual Studio에서 CodeArtifact 사용 참조하세요.

AWS 탐색기에서 Amazon RDS 사용

Amazon Relational Database Service(RDS)를 사용하면 클라우드에서 SQL 관계형 데이터베이스 시스템을 프로비저닝하고 관리할 수 있습니다. Amazon RDS는 다음 세 가지 유형의 데이터베이스 시스템을 지원합니다.

- MySQL Community Edition
- Oracle Database Enterprise Edition
- Microsoft SQL Server(Express, Standard 또는 Web Edition)

자세한 내용은 Amazon RDS 사용 설명서를 참조하세요.

여기에 설명된 많은 기능은 Amazon RDS용 AWS 관리 콘솔을 통해서도 사용할 수 있습니다.

주제

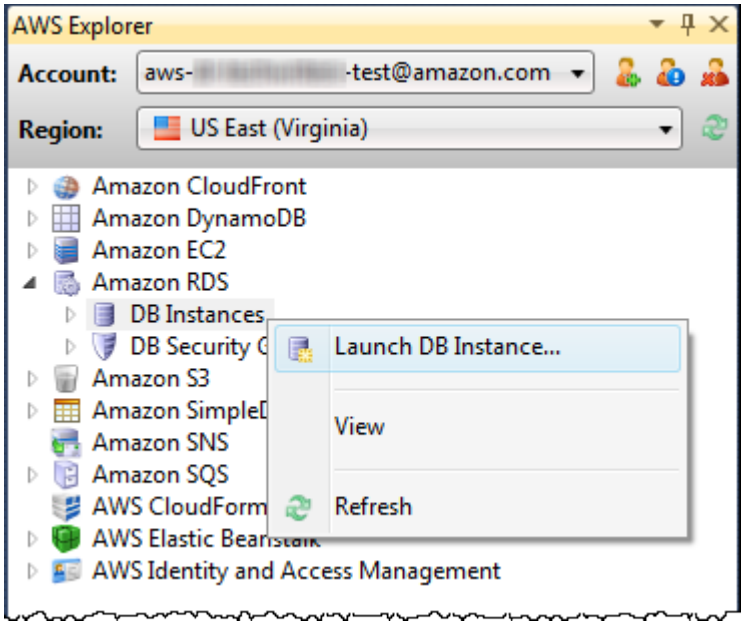
- Amazon RDS 데이터베이스 인스턴스 시작
- RDS 인스턴스에서 Microsoft SQL 서버 데이터베이스 생성
- Amazon RDS 보안 그룹

Amazon RDS 데이터베이스 인스턴스 시작

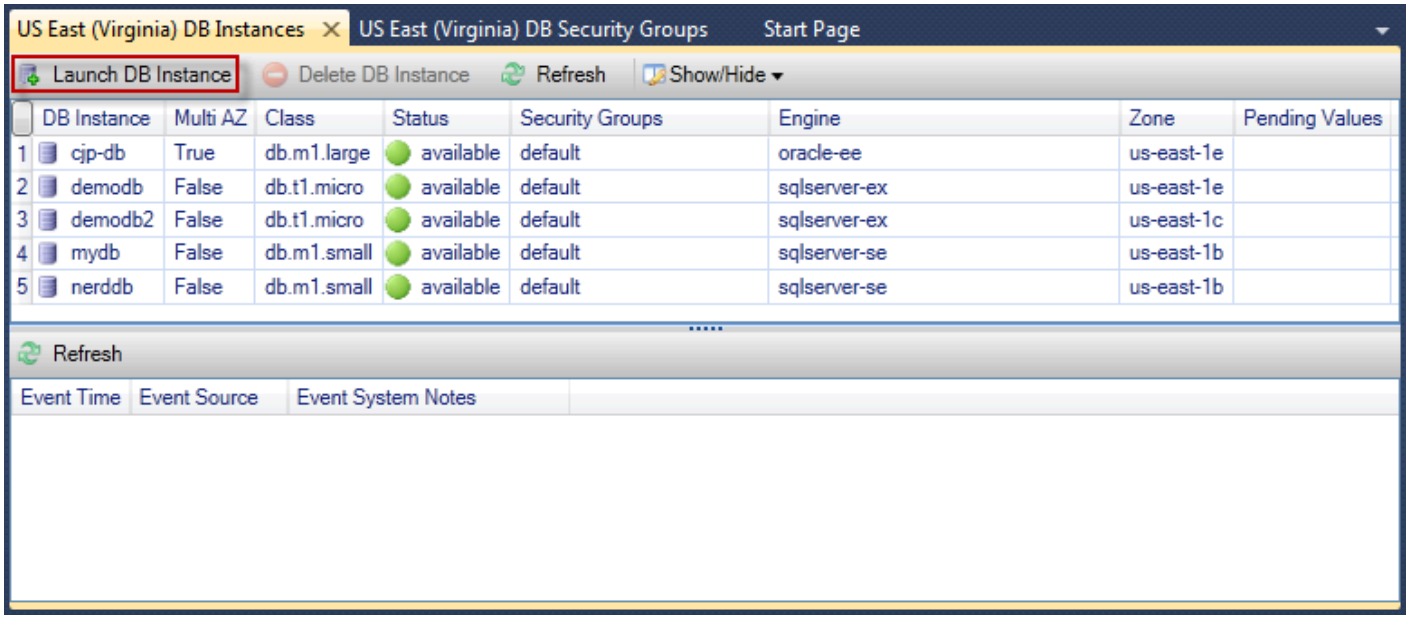
AWS Explorer를 사용하면 Amazon RDS에서 지원하는 데이터베이스 엔진의 인스턴스를 시작할 수 있습니다. 다음 연습은 Microsoft SQL Server Standard Edition의 인스턴스를 시작하기 위한 사용자 환경을 보여줍니다. 그러나 사용자 환경은 지원되는 모든 엔진에 대해 비슷합니다.

Amazon RDS 인스턴스를 시작하려면

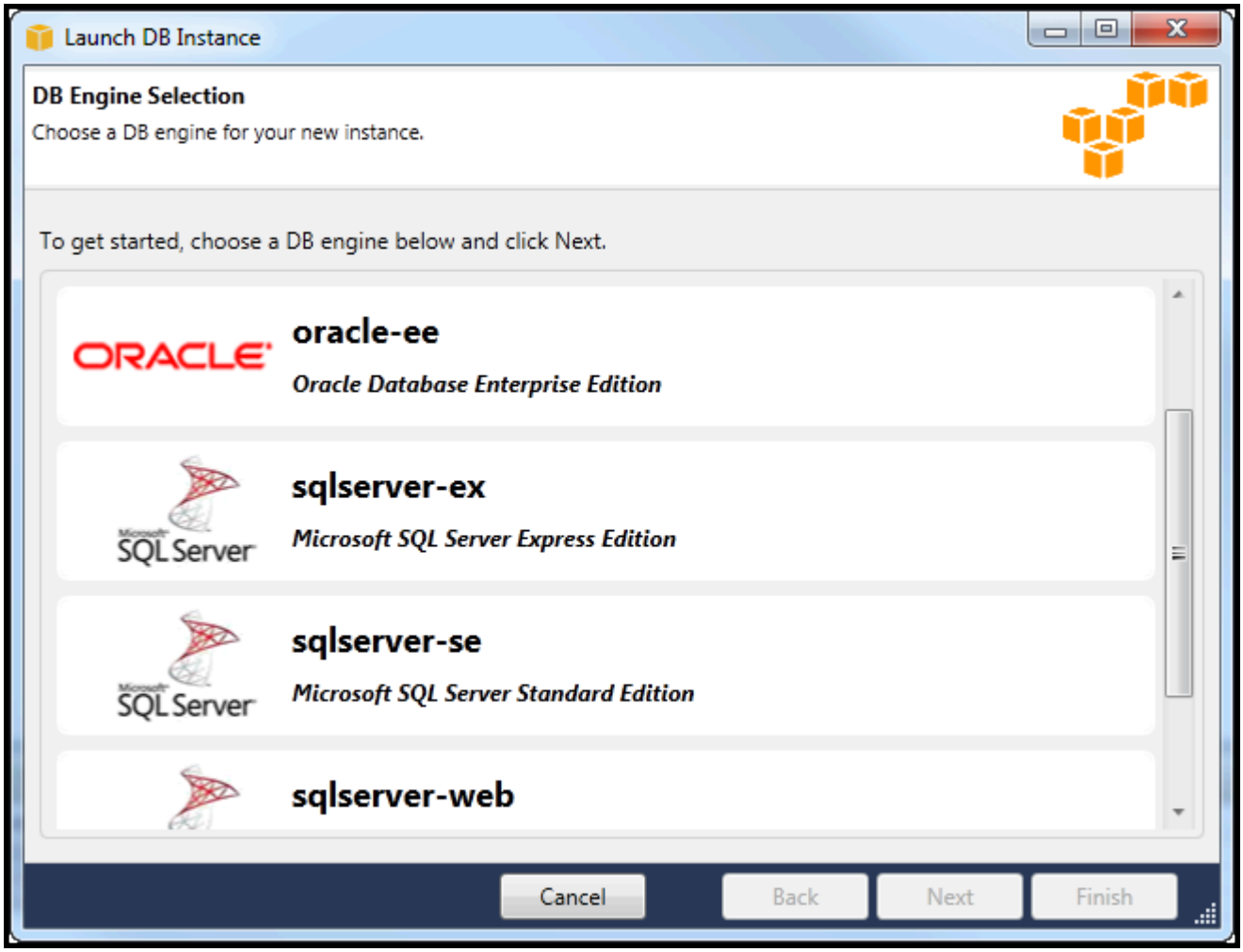
1. AWS 탐색기에서 Amazon RDS 노드의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 열고 DB 인스턴스 시작을 선택합니다.



   또는 DB 인스턴스 탭에서 DB 인스턴스 실행을 선택합니다.



2. DB Engine Selection(DB 엔진 선택) 대화 상자에서 시작할 데이터베이스 엔진 유형을 선택합니다. 이 연습의 경우 Microsoft SQL Server Standard Edition(sqlserver-se)을 선택한 다음 다음을 선택합니다.



3. DB Engine Instance Options(DB 엔진 인스턴스 옵션) 대화 상자에서 구성 옵션을 선택합니다.

   DB Engine Instance Options and Class(DB 엔진 인스턴스 옵션 및 클래스) 섹션에서 다음 설정을 지정할 수 있습니다.

   라이선스 모델

엔진 유형	라이선스
Microsoft SQL Server	라이선스 포함
MySql	general-public-license
Oracle	BYOL(bring-your-own-license, 기존 보유 라이선스 사용)

라이선스 모델은 데이터베이스 엔진 유형에 따라 달라집니다. 엔진 유형 라이선스 Microsoft SQL Server 라이선스 포함 MySql general-public-license Oracle 기존 보유 라이선스 사용

DB 인스턴스 버전

사용할 데이터베이스 엔진의 버전을 선택합니다. 버전이 하나만 지원되는 경우 해당 버전을 선택합니다.

DB 인스턴스 클래스

데이터베이스 엔진에 대한 인스턴스 클래스를 선택합니다. 인스턴스 클래스에 대한 요금은 다양합니다. 자세한 내용은 Amazon RDS 요금을 참조하십시오.

다중 AZ 배포 수행

이 옵션을 선택하면 향상된 데이터 내구성과 가용성을 위한 다중 AZ 배포를 생성할 수 있습니다. Amazon RDS는 예정된 중단이나 예상치 못한 중단이 발생하는 경우 자동 장애 조치를 위해 다른 가용 영역에 데이터베이스의 예비 복사본을 프로비저닝 및 유지 관리합니다. 다중 AZ 배포 요금에 대한 자세한 내용은 Amazon RDS 세부 정보 페이지의 요금 섹션을 참조하십시오. 이 옵션은 Microsoft SQL Server에 대해 지원되지 않습니다.

마이너 버전 자동 업그레이드

RDS 인스턴스에서 마이너 버전 업데이트를 AWS 자동으로 수행하려면이 옵션을 선택합니다.

RDS Database Instance(RDS 데이터베이스 인스턴스) 섹션에서 다음 설정을 지정할 수 있습니다.

할당된 스토리지

Engine	최소(GB)	최대(GB)
MySQL	5	1024
Oracle Enterprise Edition	10	1024
Microsoft SQL Server Express Edition	30	1024
Microsoft SQL Server Standard Edition	250	1024

Engine	최소(GB)	최대(GB)
Microsoft SQL Server Web Edition	30	1024

허용되는 최소 및 최대 스토리지는 데이터베이스 엔진 유형에 따라 달라집니다. 엔진 최소(GB) 최대(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

데이터베이스 인스턴스에 대한 이름을 지정합니다. 이 이름은 대/소문자를 구분하지 않습니다. AWS Explorer에 소문자 형식으로 표시됩니다.

마스터 사용자 이름(Master User Name)

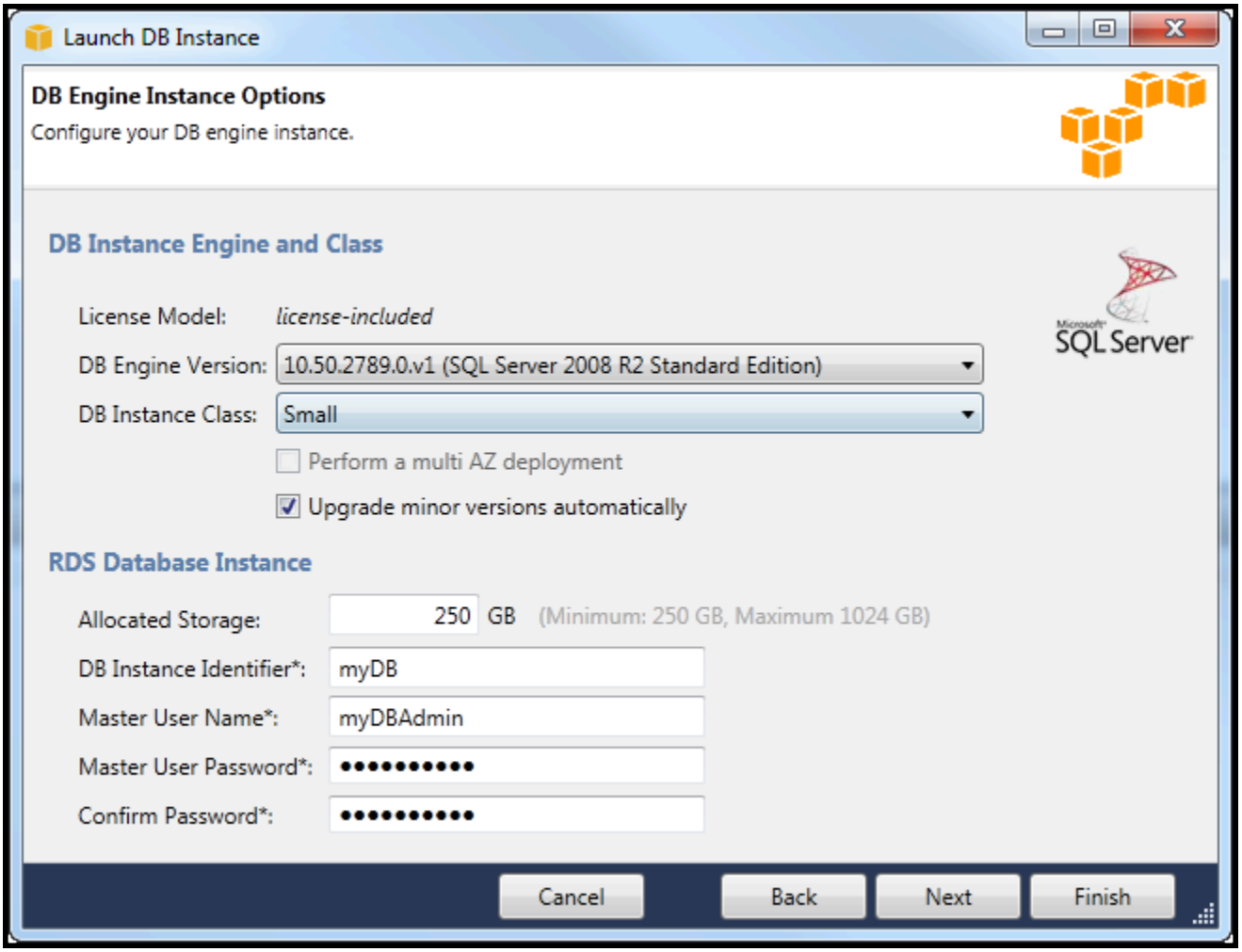
데이터베이스 인스턴스의 관리자에 대한 이름을 입력합니다.

마스터 사용자 암호

데이터베이스 인스턴스의 관리자에 대한 암호를 입력합니다.

[Confirm Password]

암호를 다시 입력하여 정확한지 확인합니다.



1. 추가 옵션 대화 상자에서 다음 설정을 지정할 수 있습니다.

   데이터베이스 포트

   인스턴스가 네트워크에서 통신하기 위해 사용할 TCP 포트입니다. 컴퓨터가 방화벽을 통해 인터넷에 액세스하는 경우 이 값을 방화벽이 트래픽을 허용하는 포트로 설정하십시오.

   가용 영역

   리전의 특정 가용 영역에서 인스턴스가 시작되도록 하려면 이 옵션을 사용합니다. 지정한 데이터베이스 인스턴스는 지정된 리전의 일부 가용 영역에서 사용할 수 없을 수도 있습니다.

   RDS 보안 그룹

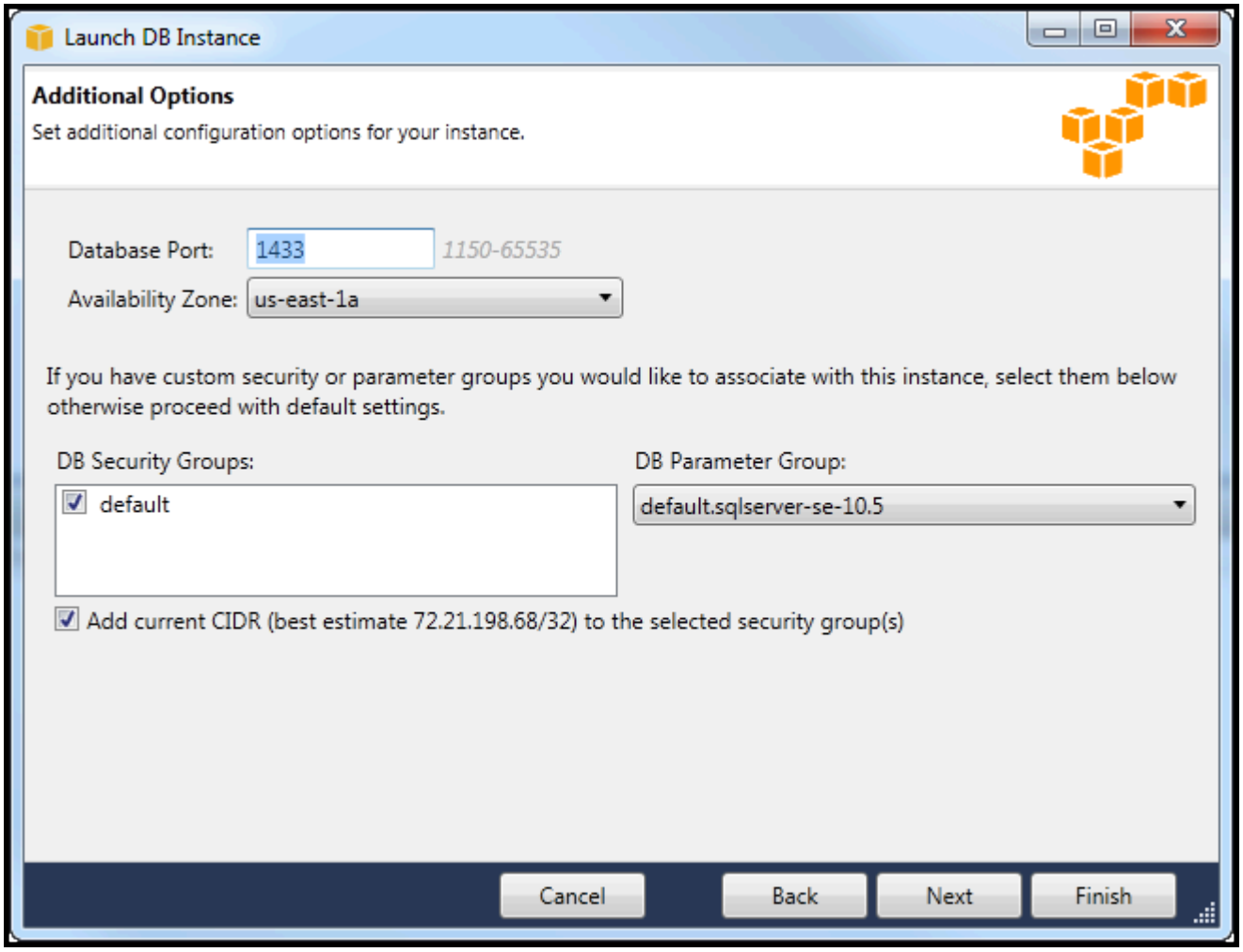
   인스턴스와 연결할 RDS 보안 그룹을 선택합니다. RDS 보안 그룹은 인스턴스에 액세스할 수 AWS 계정 있는 IP 주소, Amazon EC2 인스턴스 및를 지정합니다. RDS 보안 그룹에 대한 자세한 내용은 Amazon RDS 보안 그룹을 참조하십시오. Toolkit for Visual Studio는 현재 IP 주소를 확인하고, 이 주소를 인스턴스와 연결된 보안 그룹에 추가할 수 있는 옵션을 제공합니다. 그러나

컴퓨터가 방화벽을 통해 인터넷에 액세스하는 경우 컴퓨터에 대해 도구 키트에서 생성한 IP 주소는 정확하지 않을 수 있습니다. 사용할 IP 주소를 결정하려면 시스템 관리자에 문의하십시오.

DB 파라미터 그룹

(선택 사항) 이 드롭다운 목록에서 인스턴스와 연결할 DB 파라미터 그룹을 선택합니다. DB 파라미터 그룹을 사용하면 인스턴스에 대한 기본 구성을 변경할 수 있습니다. 자세한 내용은 Amazon 관계형 데이터베이스 서비스 사용 설명서 및 본 문서를 참조하십시오.

이 대화 상자에서 설정을 지정했으면 다음을 선택합니다.

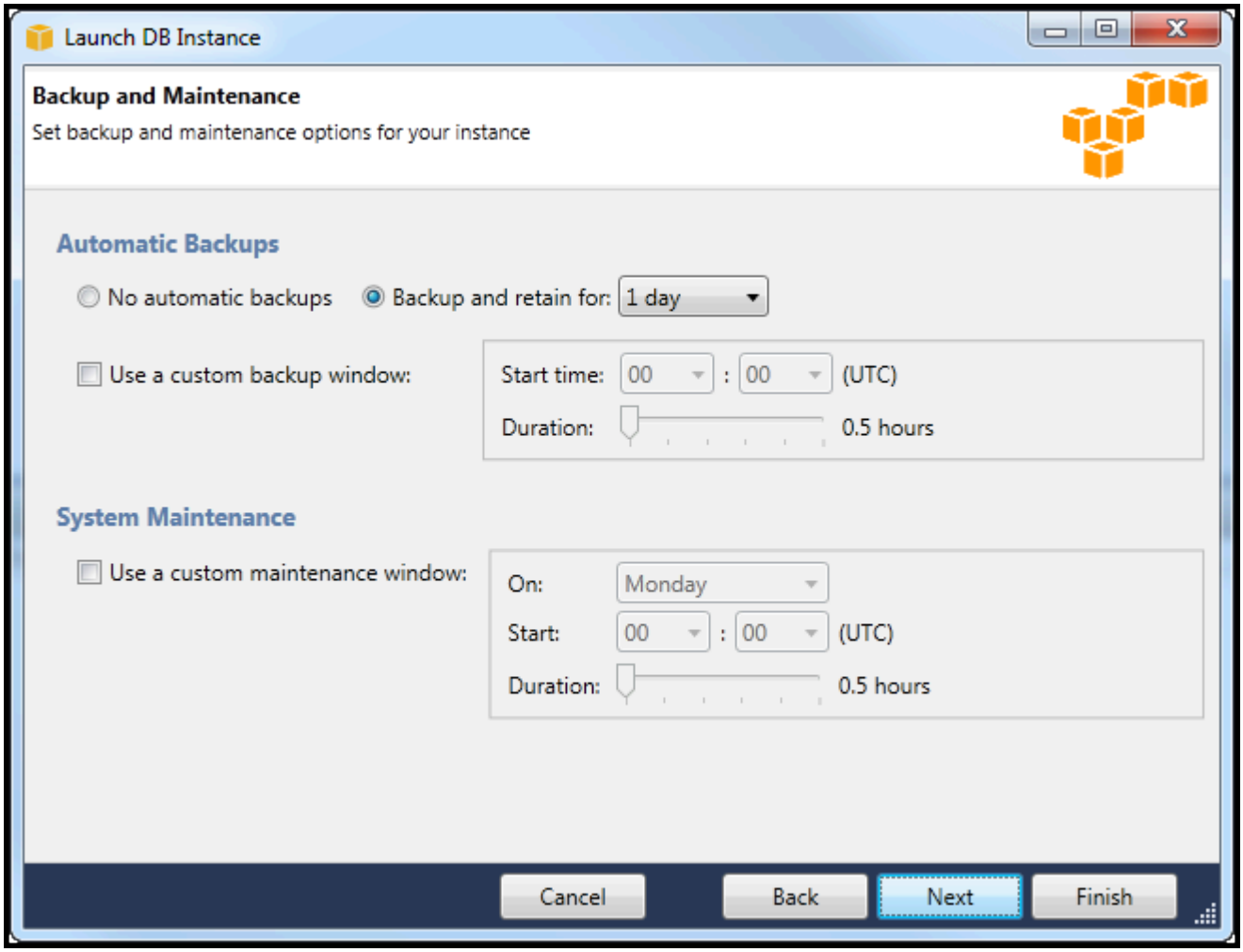


2. 백업 및 유지 관리 대화 상자에서는 Amazon RDS가 인스턴스를 백업해야 하는지 여부 및 백업해야 할 경우 백업을 보존해야 할 기간을 지정할 수 있습니다. 또한 백업이 수행되는 기간도 지정할 수 있습니다.

   이 대화 상자를 사용하여 Amazon RDS가 인스턴스에 대해 시스템 유지 관리를 수행할지 여부를 지정할 수도 있습니다. 유지 관리에는 루틴 패치와 마이너 버전 업그레이드가 포함됩니다.

시스템 유지 관리를 위해 지정한 기간은 백업을 위해 지정한 기간과 겹칠 수 없습니다.

다음을 선택합니다.



3. 마법사의 마지막 대화 상자에서는 인스턴스에 대한 설정을 검토할 수 있습니다. 설정을 수정해야 할 경우 뒤로 버튼을 사용합니다. 모든 설정이 올바르면 시작을 선택합니다.

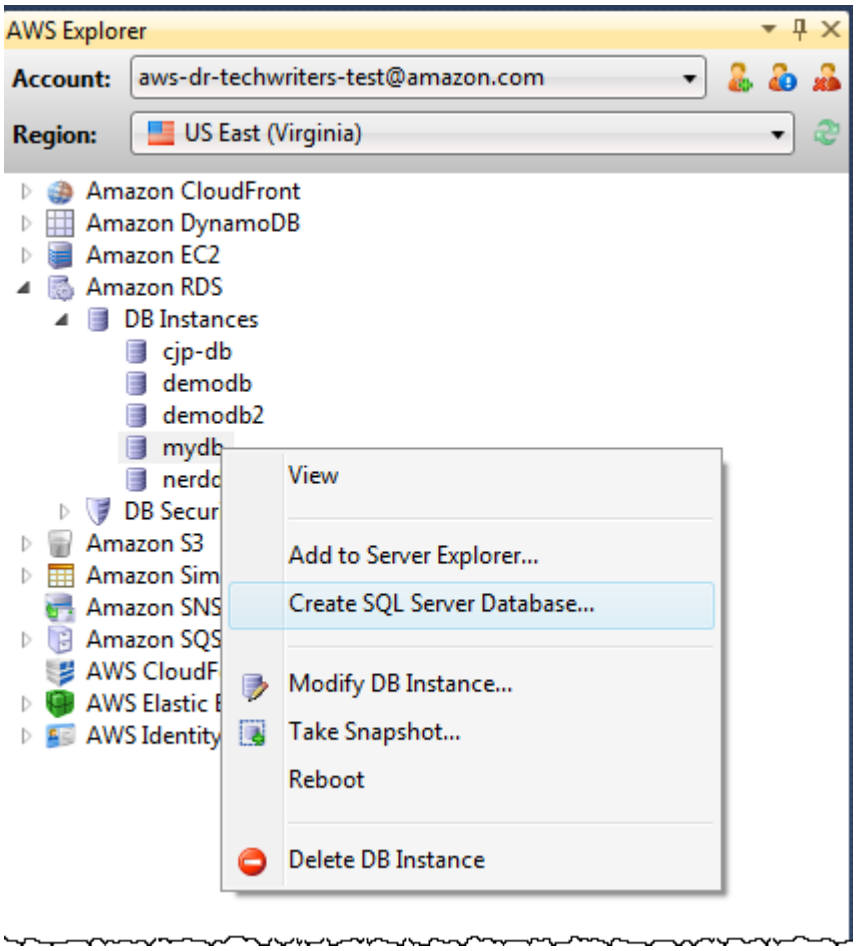
RDS 인스턴스에서 Microsoft SQL 서버 데이터베이스 생성

Microsoft SQL Server는 Amazon RDS 인스턴스를 시작한 후 RDS 인스턴스에서 SQL Server 데이터베이스를 생성해야 하는 방식으로 설계되었습니다.

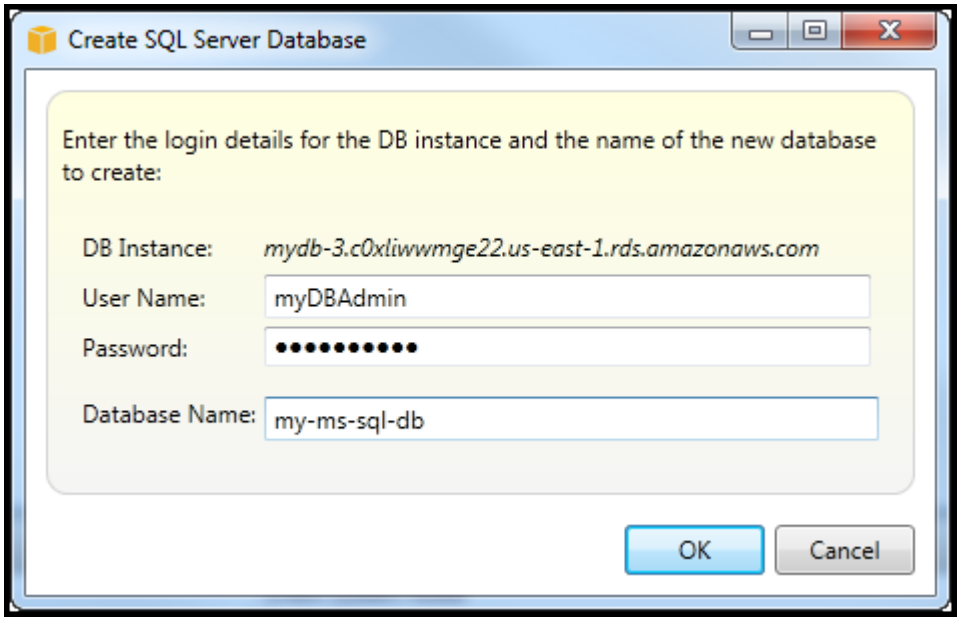
Amazon RDS 인스턴스 생성 방법에 대한 자세한 정보는 Amazon RDS 데이터베이스 인스턴스 시작을 참조하세요.

Microsoft SQL Server 데이터베이스를 생성하려면

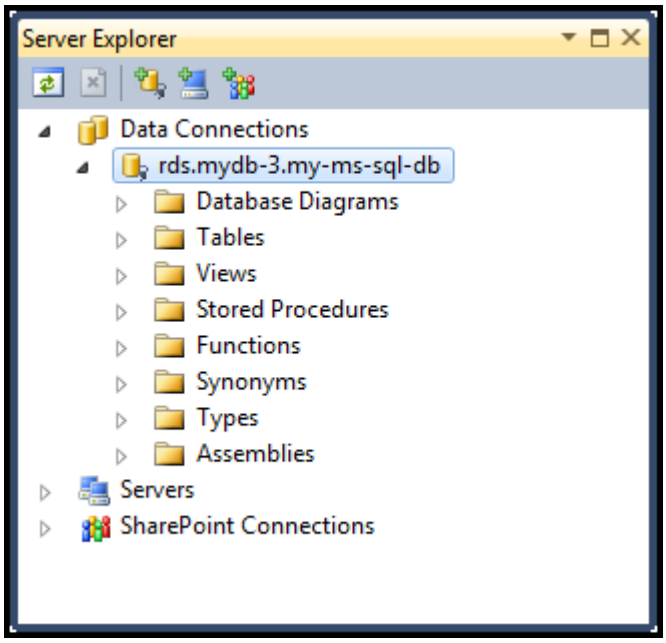
1. AWS 탐색기에서 Microsoft SQL Server용 RDS 인스턴스에 해당하는 노드의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 열고 SQL Server 데이터베이스 생성을 선택합니다.



2. Create SQL Server Database(SQL Server 데이터베이스 생성) 대화 상자에서 RDS 인스턴스를 생성할 때 지정한 암호를 입력하고, Microsoft SQL Server 데이터베이스에 대한 이름을 입력한 다음 확인을 선택합니다.



3. Toolkit for Visual Studio가 Microsoft SQL Server 데이터베이스를 생성하고 해당 데이터베이스를 Visual Studio 서버 탐색기에 추가하세요.



Amazon RDS 보안 그룹

Amazon RDS 보안 그룹을 사용하면 Amazon RDS 인스턴스에 대한 네트워크 액세스를 관리할 수 있습니다. 보안 그룹과 함께 CIDR 표기법을 사용하여 IP 주소 모음을 지정하면 이 주소에서 시작하는 네트워크 트래픽만 Amazon RDS 인스턴스에서 인식합니다.

기능은 비슷하지만 Amazon RDS 보안 그룹은 Amazon EC2 보안 그룹과 다릅니다. RDS 보안 그룹에는 EC2 보안 그룹을 추가할 수 있습니다. 그러면 EC2 보안 그룹의 멤버인 EC2 인스턴스가 RDS 보안 그룹의 멤버인 RDS 인스턴스에 액세스할 수 있습니다.

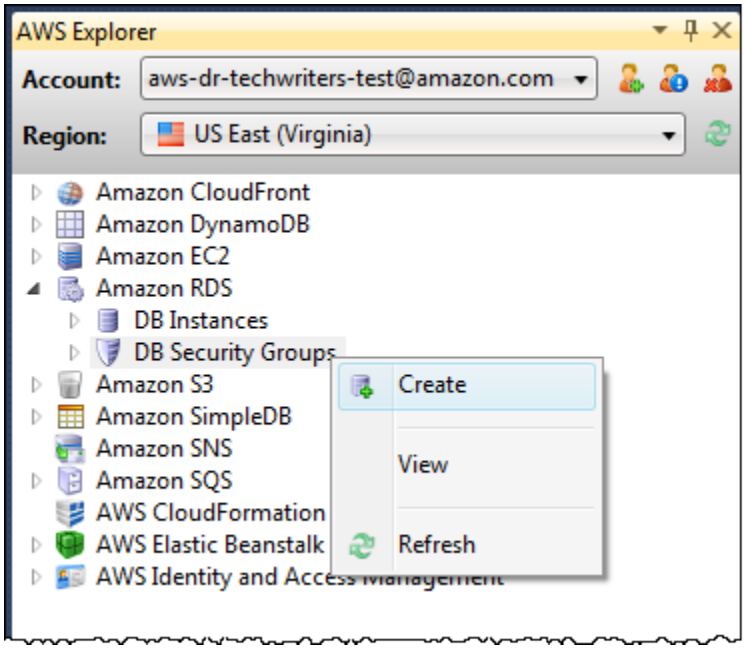
Amazon RDS 보안 그룹에 대한 자세한 정보는 RDS 보안 그룹을 참조하세요. Amazon EC2 보안 그룹에 대한 자세한 정보는 EC2 사용 설명서를 참조하세요.

Amazon RDS 보안 그룹 생성

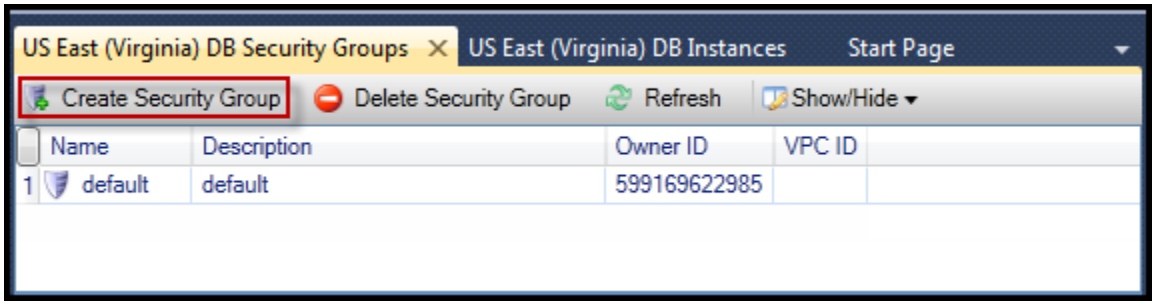
Toolkit for Visual Studio를 사용하여 RDS 보안 그룹을 생성할 수 있습니다. AWS 도구 키트를 사용하여 RDS 인스턴스를 시작하는 경우 마법사를 사용하여 인스턴스에 사용할 RDS 보안 그룹을 지정할 수 있습니다. 마법사를 시작하기 전에 다음 절차를 사용하여 보안 그룹을 생성할 수 있습니다.

Amazon RDS 보안 그룹을 생성하려면

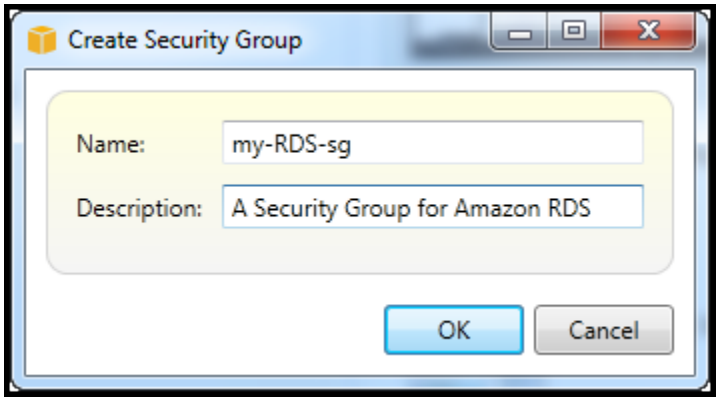
1. AWS 탐색기에서 Amazon RDS 노드를 확장하고 DB 보안 그룹 하위 노드의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 연 다음 생성을 선택합니다.



   또는 보안 그룹 탭에서 보안 그룹 생성을 선택합니다. 이 탭이 표시되지 않으면 DB 보안 그룹 하위 노드의 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴를 연 다음 보기를 선택합니다.



2. 보안 그룹 생성 대화 상자에서 보안 그룹의 이름과 설명을 입력한 다음 확인을 선택합니다.



Amazon RDS 보안 그룹에 대한 액세스 권한 설정

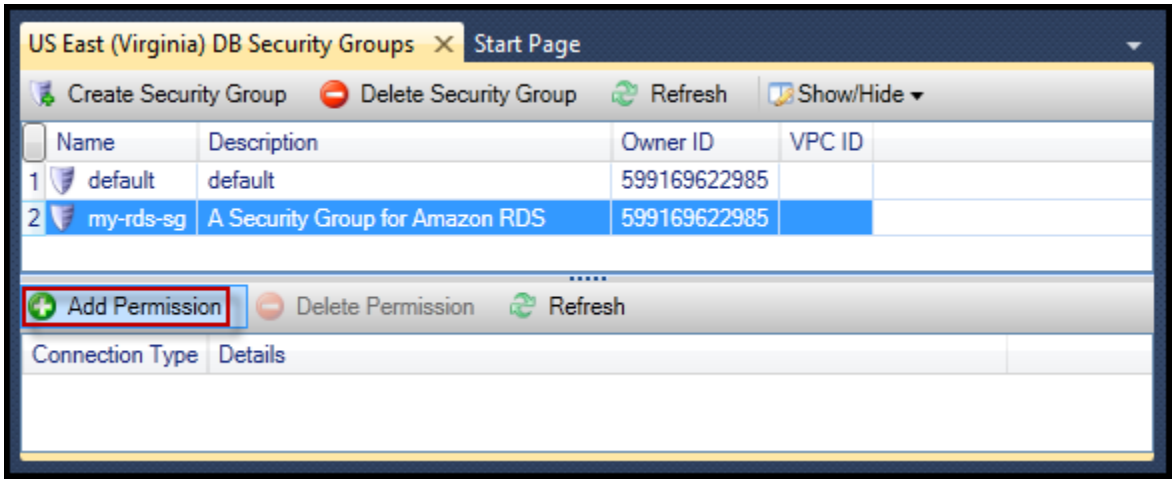
기본적으로 새 Amazon RDS 보안 그룹은 네트워크 액세스를 제공하지 않습니다. 보안 그룹을 사용하는 Amazon RDS 인스턴스 액세스를 활성화하려면 다음 절차를 사용하여 액세스 권한을 설정하세요.

Amazon RDS 보안 그룹에 대한 액세스를 설정하려면

1. 보안 그룹 탭의 목록 보기에서 보안 그룹을 선택합니다. 보안 그룹이 목록에 나타나지 않으면 새로 고침을 선택합니다. 보안 그룹이 여전히 목록에 표시되지 않는 경우 올바른 AWS 리전의 목록을 보고 있는지 확인합니다. AWS 도구 키트의 보안 그룹 탭은 리전별로 다릅니다.

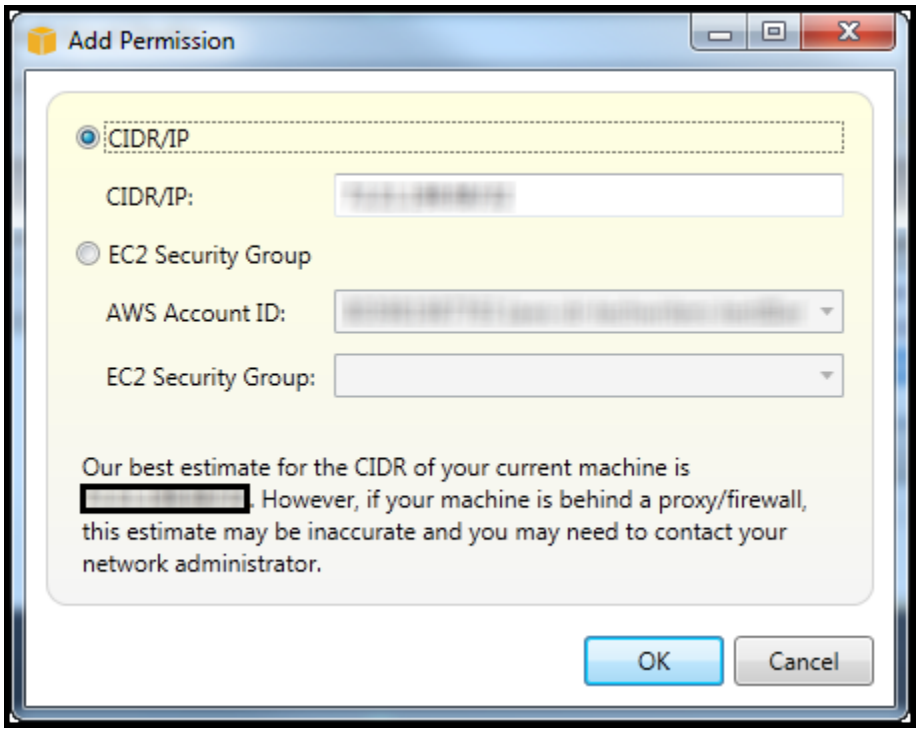
   보안 그룹 탭이 표시되지 않으면 AWS 탐색기에서 DB 보안 그룹 하위 노드의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 열고 보기를 선택합니다.

2. [Add permission]을 선택합니다.



보안 그룹 탭에 있는 권한 추가 버튼

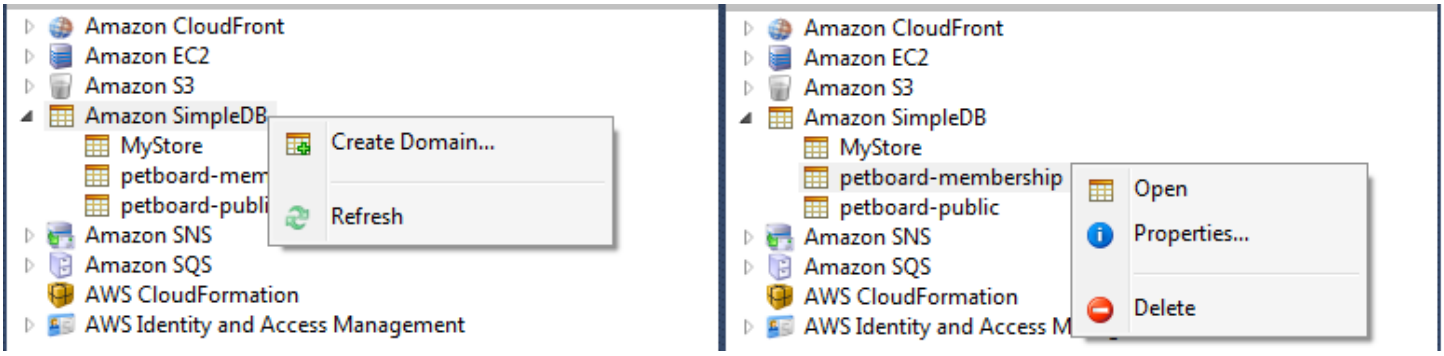
3. 권한 추가 대화 상자에서 CIDR 표기법을 사용하여 RDS 인스턴스에 액세스할 수 있는 IP 주소를 지정하거나 RDS 인스턴스에 액세스할 수 있는 EC2 보안 그룹을 지정할 수 있습니다. EC2 보안 그룹을 선택하면 액세스 권한이 있는와 연결된 모든 EC2 인스턴스에 대한 AWS 계정 액세스를 지정하거나 드롭다운 목록에서 EC2 보안 그룹을 선택할 수 있습니다.



AWS 도구 키트는 IP 주소를 확인하고 대화 상자를 적절한 CIDR 사양으로 자동 채우려고 시도합니다. 그러나 컴퓨터가 방화벽을 통해 인터넷에 액세스하는 경우 도구 키트로 결정된 CIDR이 정확하지 않을 수 있습니다.

AWS Explorer에서 Amazon SimpleDB 사용

AWS 탐색기는 활성 AWS 계정과 연결된 모든 Amazon SimpleDB 도메인을 표시합니다. AWS 탐색기에서 Amazon SimpleDB 도메인을 생성하거나 삭제할 수 있습니다.

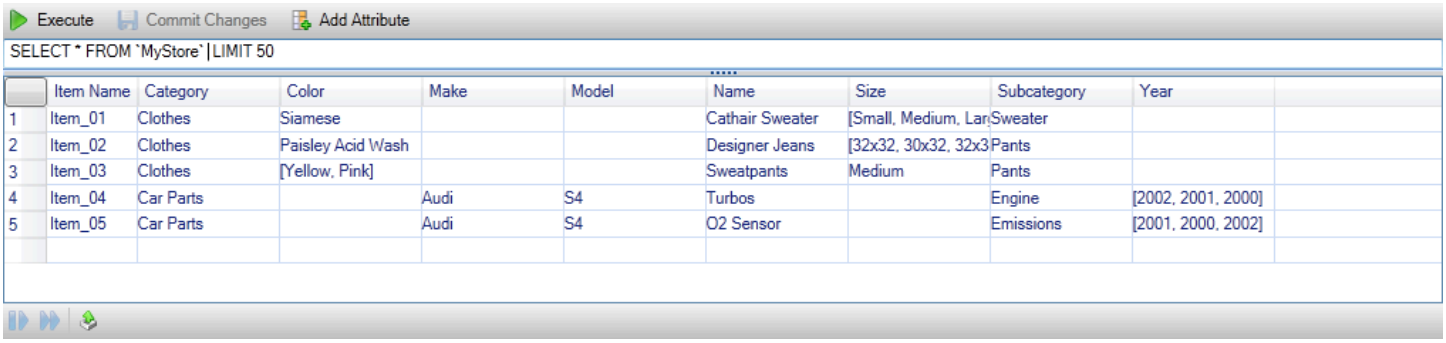


Create, delete, or open Amazon SimpleDB domains associated with your account

쿼리 실행 및 결과 편집

AWS 탐색기는 해당 도메인의 항목, 속성 및 값을 볼 수 있는 Amazon SimpleDB 도메인의 그리드 보기를 표시할 수도 있습니다. 도메인 항목의 하위 세트만 표시되도록 쿼리를 실행할 수 있습니다. 셀을 두 번 클릭하여 항목의 해당 속성에 대한 값을 편집할 수 있습니다. 도메인에 새로운 속성을 추가할 수도 있습니다.

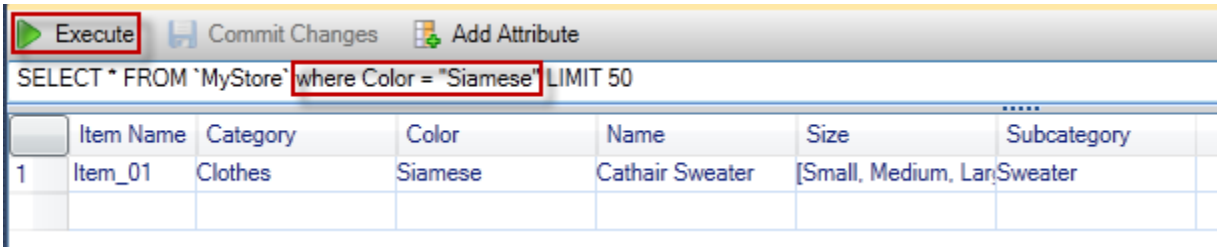
여기에 표시된 도메인은 AWS SDK for .NET과 함께 포함된 Amazon SimpleDB 샘플에서 가져온 것입니다.



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

Amazon SimpleDB grid view

쿼리를 실행하려면 표 보기 상단의 텍스트 상자에서 쿼리를 편집한 다음 실행을 선택합니다. 쿼리와 일치하는 항목만 표시하도록 보기가 필터링됩니다.

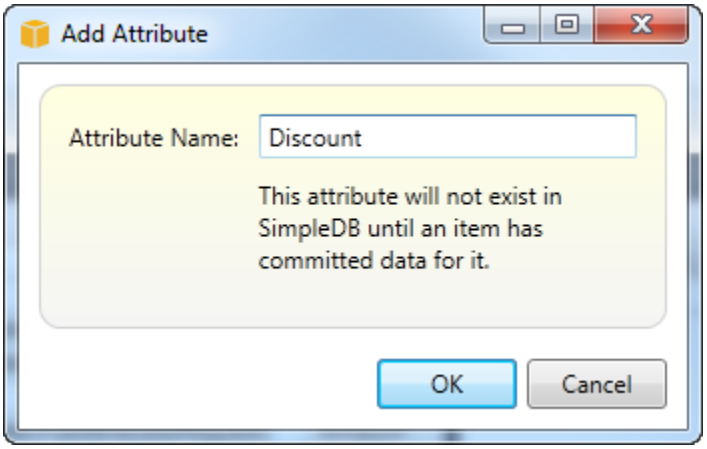


Execute query from AWS Explorer

속성과 연결된 값을 편집하려면 해당 셀을 두 번 클릭하여 값을 편집한 다음 변경 사항 커밋을 선택합니다.

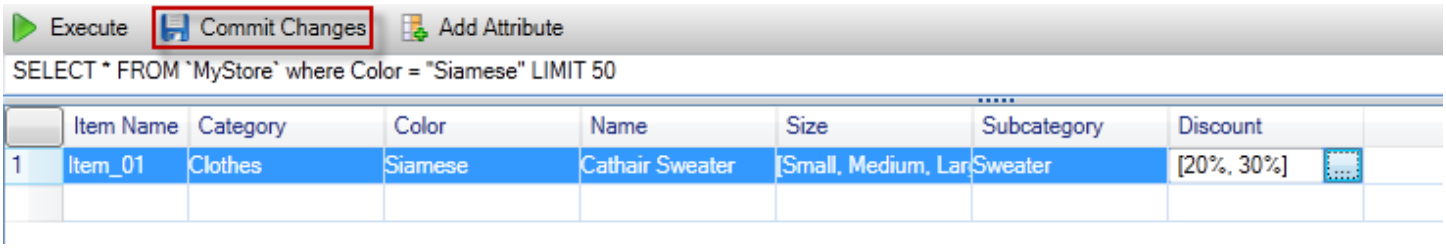
속성 추가

속성을 추가하려면 보기 상단에서 속성 추가를 선택합니다.



속성 추가 dialog box

속성을 도메인 일부로 만들려면 하나 이상의 항목에 값을 추가한 다음 변경 사항 커밋을 선택합니다.



Commit changes for a new attribute

쿼리 결과 페이지 매김

보기 하단에 버튼 3개가 있습니다.

Paginate and export buttons

처음 2개의 버튼은 쿼리 결과 페이지를 매깁니다. 결과의 추가 페이지 하나를 표시하려면 첫 번째 버튼을 선택하고 결과의 추가 페이지 결과 10개를 표시하려면 두 번째 버튼을 선택하십시오. 이 컨텍스트에서 한 페이지는 열 100개 또는 LIMIT 값으로 지정한 결과의 수(쿼리에 포함된 경우)와 같습니다.

CSV로 내보내기

마지막 버튼은 현재 결과를 CSV 파일로 내보냅니다.

AWS Explorer에서 Amazon SQS 사용

Amazon Simple Queue Service(Amazon SQS)는 소프트웨어 애플리케이션에서 서로 다른 실행 프로세스 간에 메시지를 전달할 수 있는 유연한 대기열 서비스입니다. Amazon SQS 대기열은 AWS 인프라에 있지만 메시지를 전달하는 프로세스는 로컬, Amazon EC2 인스턴스 또는 이들의 일부 조합에 위치할 수 있습니다. Amazon SQS는 여러 컴퓨터에 대한 작업 분배를 조정하는 데 이상적입니다.

Toolkit for Visual Studio를 사용하면 활성 계정과 연결된 Amazon SQS 대기열을 보고, 대기열을 생성하거나 삭제하며, 대기열을 통해 메시지를 보낼 수 있습니다. (활성 계정은 AWS 탐색기에서 선택한 계정입니다.)

Amazon SQS에 대한 자세한 내용은 AWS 설명서의 SQS 소개를 참조하세요.

대기열 만들기

AWS 탐색기에서 Amazon SQS 대기열을 생성할 수 있습니다. 대기열의 ARN 및 URL은 활성 계정의 계정 번호와 생성 시 지정한 대기열 이름을 기반으로 합니다.

대기열 생성

1. AWS 탐색기에서 Amazon SQS 노드의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 연 다음 대기열 생성을 선택합니다.
2. 대기열 생성 대화 상자에서 대기열 이름, 기본 제한 시간 초과 및 기본 전송 지연 시간을 지정합니다. 기본 제한 시간 초과 및 기본 전송 지연 시간은 초 단위로 지정됩니다. 기본 제한 시간 초과는 지정된 프로세스가 메시지를 받은 후 잠재적인 수신 프로세스가 메시지를 볼 수 없게 되는 시간입니다. 기본 전송 지연 시간은 메시지가 전송된 순간부터 잠재적인 수신 프로세스에 처음 표시될 때까지의 시간입니다.
3. 확인을 선택합니다. 새 대기열이 Amazon SQS 노드에 하위 노드로 나타납니다.

대기열 삭제

AWS Explorer에서 기존 대기열을 삭제할 수 있습니다. 대기열을 삭제하면 대기열과 연결된 모든 메시지를 더 이상 사용할 수 없습니다.

대기열 삭제

1. AWS 탐색기에서 삭제할 대기열의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 연 다음 삭제를 선택합니다.

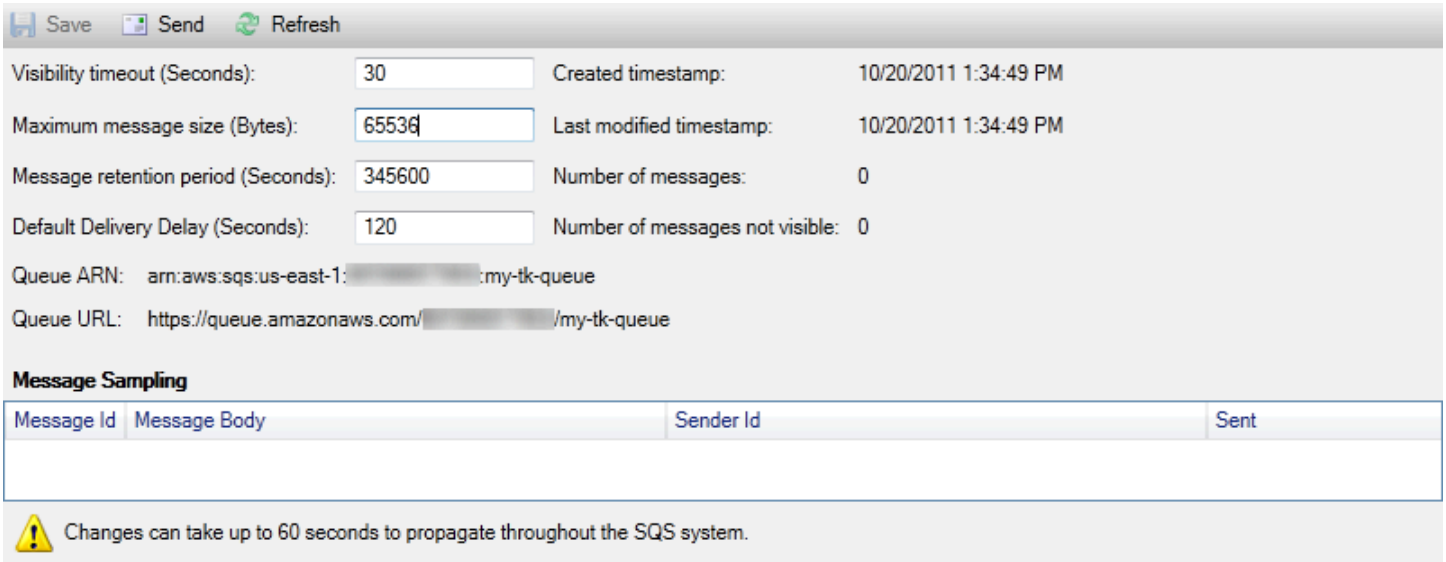
대기열 속성 관리

AWS 탐색기에 표시된 대기열의 속성을 보고 편집할 수 있습니다. 또한 이 속성 보기에서 대기열로 메시지를 전송할 수 있습니다.

대기열 속성을 관리하려면

- AWS 탐색기에서 속성을 관리하려는 대기열의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 연 다음 대기열 보기를 선택합니다.

  대기열 속성 보기에서 제한 시간 초과, 최대 메시지 크기, 메시지 보존 기간 및 기본 전송 지연 시간을 편집할 수 있습니다. 메시지를 전송할 때 기본 전송 지연 시간을 재정의할 수 있습니다. 다음 스크린샷에서 숨겨진 텍스트는 대기열 ARN 및 URL의 계정 번호 구성 요소입니다.



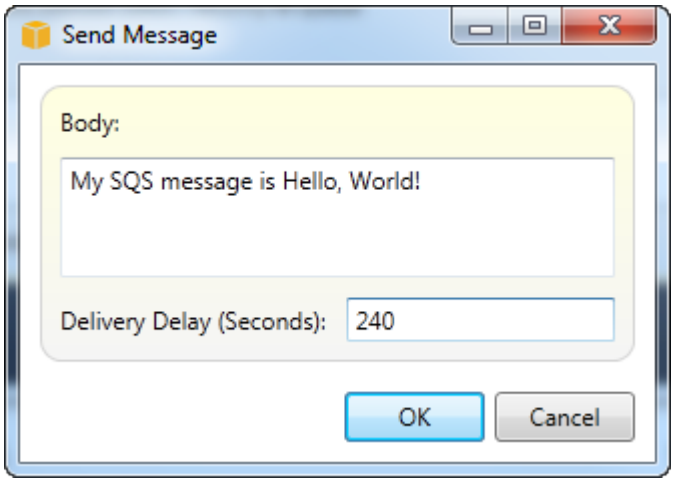
SQS queue properties view

대기열로 메시지 전송

대기열 속성 보기에서 대기열로 메시지를 전송할 수 있습니다.

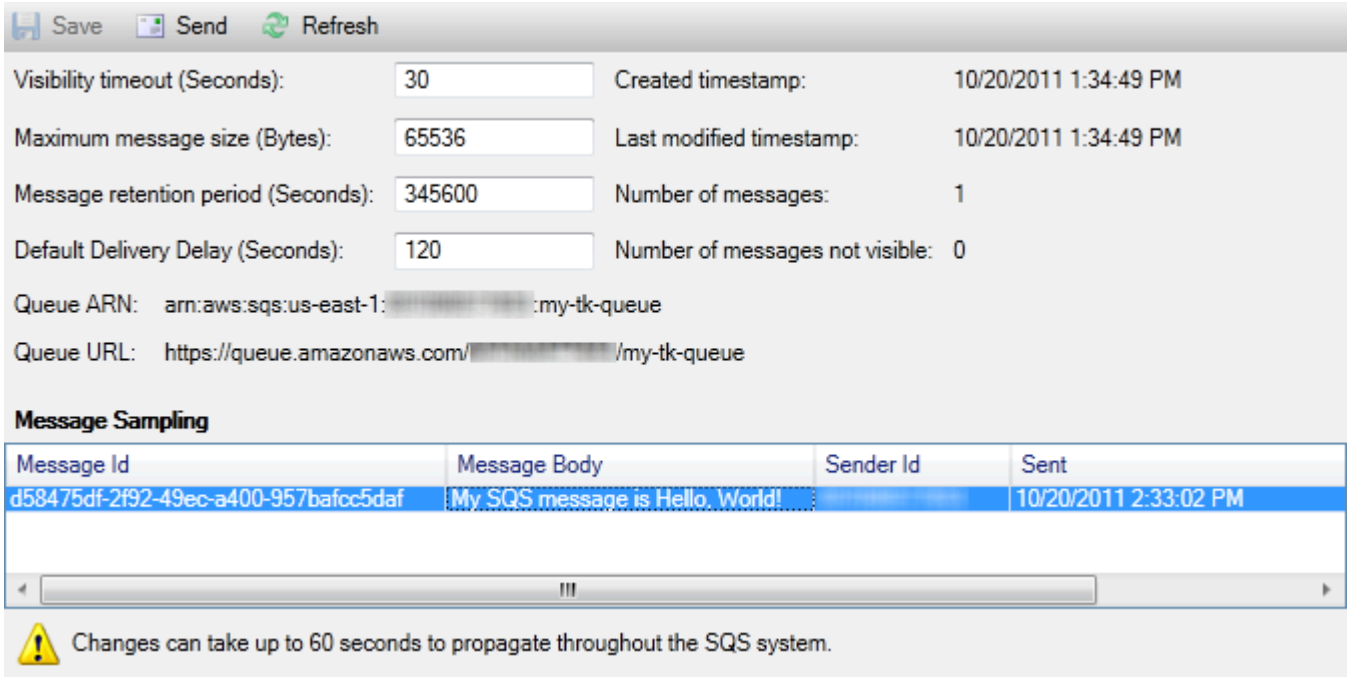
메시지 전송

1. 대기열 속성 보기의 상단에서 전송 버튼을 선택합니다.
2. 메시지를 입력합니다. (선택 사항) 대기열의 기본 전송 지연 시간을 대신할 전송 지연 시간을 입력합니다. 다음 예에서는 지연 시간을 240초로 재정의했습니다. 확인을 선택합니다.



메시지 전송 dialog box

3. 약 240초(4분)을 기다립니다. 메시지가 대기열 속성 보기의 Message Sampling(메시지 샘플링) 섹션에 나타납니다.



SQS properties view with sent message

대기열 속성 보기의 타임스탬프는 전송 버튼을 선택한 시간입니다. 지연 시간에는 포함되지 않습니다. 따라서 메시지가 대기열에 나타나고 수신자가 사용할 수 있는 시간이 이 타임스탬프보다 늦을 수 있습니다. 타임스탬프는 컴퓨터의 현지 시간으로 표시됩니다.

자격 증명 및 액세스 관리

AWS Identity and Access Management (IAM)를 사용하면 AWS 계정 및 리소스에 대한 액세스를 보다 안전하게 관리할 수 있습니다. IAM을 사용하면 기본(루트)에서 여러 사용자를 생성할 수 있습니다 AWS 계정. 이러한 사용자는 자체 보안 인증 정보(암호, 액세스 키 ID, 비밀 키)를 가질 수 있지만 모든 IAM 사용자는 하나의 계정 번호를 공유합니다.

사용자에게 IAM 정책을 연결하여 각 IAM 사용자의 리소스 액세스 수준을 관리할 수 있습니다. 예를 들어, 계정에서 Amazon S3 서비스 및 관련 리소스에 대해 사용자 액세스를 제공한 IAM 사용자와 다른 서비스 또는 리소스에 대해 액세스를 제공하지 않은 IAM 사용자에게 정책을 연결할 수 있습니다.

더욱 효율적인 액세스 관리를 위해 사용자를 모아 IAM 그룹을 생성할 수 있습니다. 그룹에 정책을 연결하면 그룹 멤버인 모든 사용자에게 영향을 줍니다.

IAM은 사용자 및 그룹 수준에서 권한 관리 외에 IAM 역할의 개념도 제공합니다. 사용자 및 그룹과 같이 IAM 역할에 정책을 연결할 수 있습니다. 그런 다음 IAM 역할을 Amazon EC2 인스턴스와 연결할 수 있습니다. EC2 인스턴스에서 실행되는 애플리케이션은 IAM 역할에서 제공하는 권한을 AWS 사용하여에 액세스할 수 있습니다. 도구 키트로 IAM 역할 사용에 대한 자세한 내용은 IAM 역할 생성을 참조하십시오. IAM에 대한 자세한 정보는 IAM 사용 설명서를 참조하세요.

IAM 사용자 생성 및 구성

IAM 사용자를 사용하면 다른 사용자에게에 대한 액세스 권한을 부여할 수 있습니다 AWS 계정. IAM 사용자에게 정책을 연결할 수 있으므로 IAM 사용자가 액세스할 수 있는 리소스와 해당 리소스에서 수행할 수 있는 작업을 세부적으로 제한할 수 있습니다.

에 액세스하는 모든 사용자는 계정 소유자를 포함하여 IAM 사용자로 AWS 계정 액세스하는 것이 가장 좋습니다. 이렇게 하면 IAM 사용자 중 한 명의 보안 인증 정보가 손상되는 경우 해당 보안 인증 정보만 비활성화할 수 있습니다. 계정에 대한 루트 자격 증명을 비활성화하거나 변경할 필요가 없습니다.

Toolkit for Visual Studio에서 사용자에게 IAM 정책을 연결하거나 그룹에 사용자를 할당하여 IAM 사용자에게 권한을 할당할 수 있습니다. 그룹에 할당된 IAM 사용자는 그룹에 연결된 정책에서 권한을 가져옵니다. 자세한 내용은 IAM 그룹 생성 및 IAM 그룹에 IAM 사용자 추가를 참조하십시오.

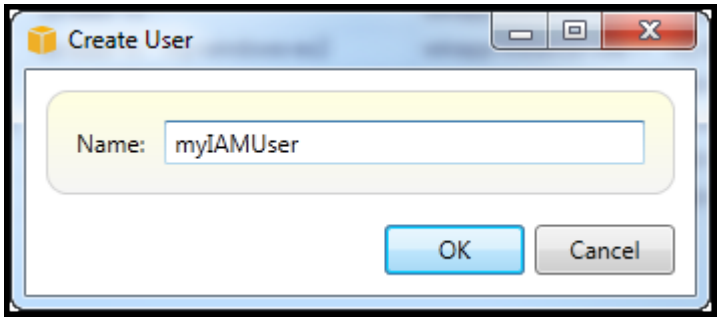
Toolkit for Visual Studio에서 IAM 사용자에 대한 AWS 자격 증명(액세스 키 ID 및 보안 키)을 생성할 수도 있습니다. 자세한 내용은 IAM 사용자의 자격 증명 생성을 참조하십시오.

Toolkit for Visual Studio는 AWS Explorer를 통해 서비스에 액세스하기 위한 IAM 사용자 자격 증명 지정을 지원합니다. IAM 사용자는 일반적으로 모든 Amazon Web Services에 대한 전체 액세스 권한을 가지고 있지 않으므로 AWS Explorer의 일부 기능을 사용하지 못할 수 있습니다. 활성 계정이 IAM 사용자인 동안 AWS Explorer를 사용하여 리소스를 변경한 다음 활성 계정을 루트 계정으로 전환하면 AWS Explorer에서 뷰를 새로 고칠 때까지 변경 사항이 표시되지 않을 수 있습니다. 보기를 새로 고치려면 새로 고침 () 버튼을 선택합니다.

에서 IAM 사용자를 구성하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 사용자 및 그룹 작업을 AWS Management Console참조하세요. https://docs.aws.amazon.com/IAM/latest/UserGuide/Using_WorkingWithGroupsAndUsers.html

IAM 사용자를 생성하려면

1. AWS 탐색기에서 AWS Identity and Access Management 노드를 확장하고 사용자의 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴를 연 다음 사용자 생성을 선택합니다.
2. 사용자 생성 대화 상자에서 IAM 사용자에 대한 이름을 입력하고 확인을 선택하세요. 이는 IAM 표시 이름입니다. IAM 사용자 이름 제한에 대한 정보는 IAM 사용 설명서를 참조하세요.



Create an IAM user

새 사용자는 AWS Identity and Access Management 노드 아래의 사용자 아래에 하위 노드로 표시됩니다.

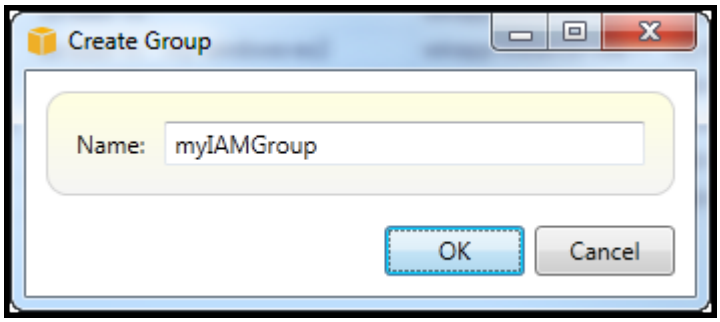
정책을 생성하고 사용자에게 연결하는 방법에 대한 자세한 내용은 IAM 정책 생성을 참조하십시오.

IAM 그룹 생성

그룹은 사용자 모음에 IAM 정책을 적용하는 방법을 제공합니다. IAM 사용자 및 그룹을 관리하는 방법에 대한 정보는 IAM 사용 설명서에서 사용자 및 그룹 작업 섹션을 참조하세요.

IAM 그룹을 생성하려면

1. AWS 탐색기의 Identity and Access Management에서 그룹의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 열고 그룹 생성을 선택합니다.
2. 그룹 생성 대화 상자에서 IAM 그룹에 대한 이름을 입력하고 확인을 선택하세요.



Create IAM group

새 IAM 그룹은 Identity and Access Management의 그룹 하위 노드 아래에 표시됩니다.

정책 생성 및 IAM 그룹 연결에 대한 정보는 IAM 정책 생성을 참조하세요.

IAM 그룹에 IAM 사용자 추가

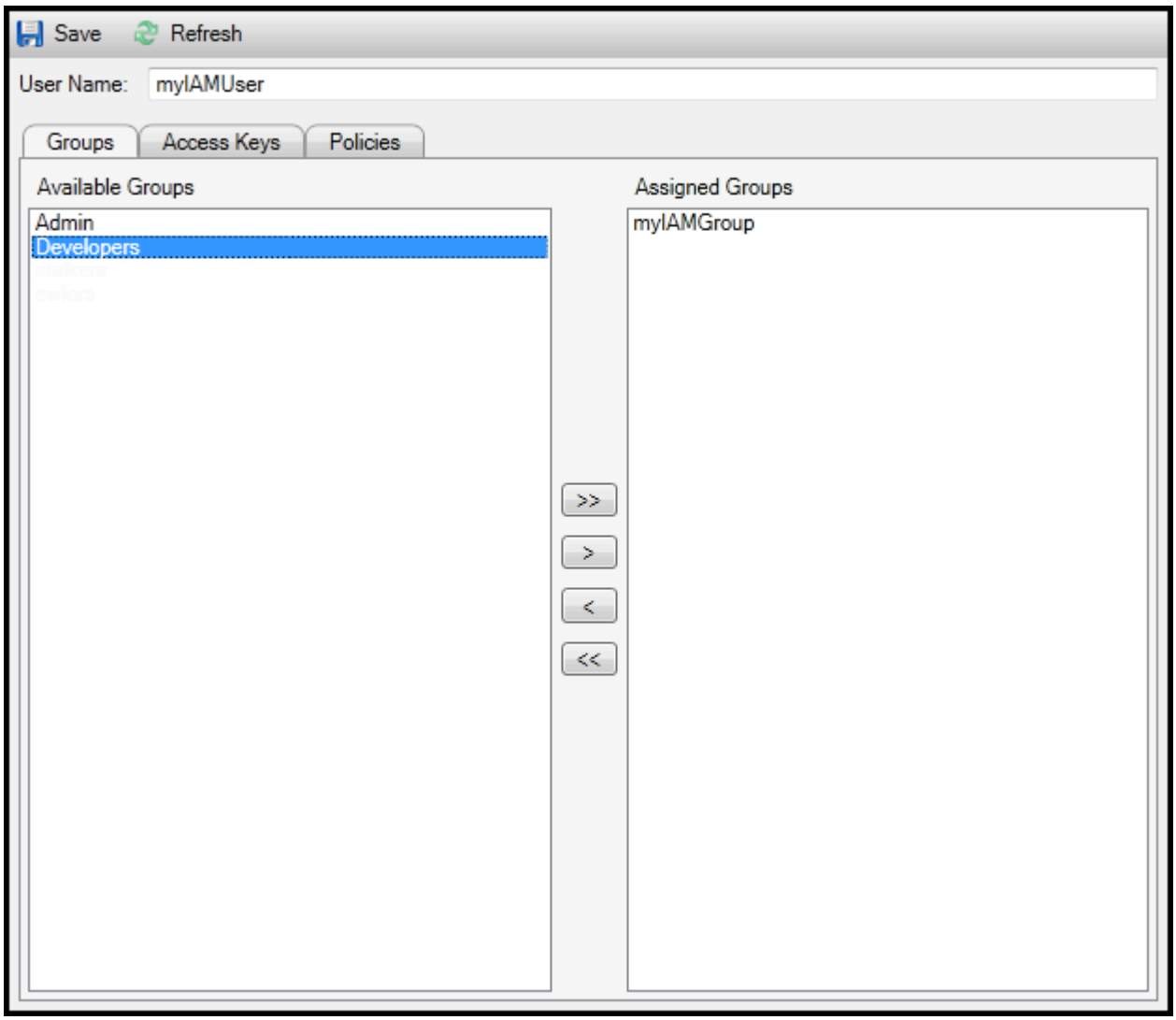
IAM 그룹의 멤버인 IAM 사용자는 해당 그룹에 연결된 정책에서 액세스 권한을 가져옵니다. IAM 그룹의 목적은 IAM 사용자를 모아서 권한을 쉽게 관리하는 것입니다.

IAM 그룹에 연결된 정책이 해당 IAM 그룹의 멤버인 IAM 사용자에게 연결된 정책과 상호 작용하는 방법에 대한 정보는 IAM 사용 설명서에서 IAM 정책 관리 섹션을 참조하세요.

AWS 탐색기에서는 그룹 하위 노드가 아닌 사용자 하위 노드에서 IAM 사용자를 IAM 그룹에 추가합니다.

IAM 그룹에 IAM 사용자를 추가하려면

1. AWS 탐색기의 Identity and Access Management에서 사용자의 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴를 열고 편집을 선택합니다.



Assign an IAM user to a IAM group

2. 그룹 탭의 왼쪽 창에는 사용 가능한 IAM 그룹이 표시됩니다. 오른쪽 창에는 지정한 IAM 사용자가 이미 멤버인 그룹이 표시됩니다.

   왼쪽 창에서 그룹에 IAM 사용자를 추가하려면 IAM 그룹을 선택한 다음 > 버튼을 선택하세요.

   오른쪽 창의 그룹에서 IAM 사용자를 제거하려면 IAM 그룹을 선택한 다음 < 버튼을 선택하세요.

   모든 IAM 그룹에 IAM 사용자를 추가하려면 >> 버튼을 선택하세요. 또한 모든 그룹에서 IAM 사용자를 제거하려면 << 버튼을 선택하세요.

   여러 그룹을 선택하려면 순서대로 선택합니다. Ctrl 키를 계속 누를 필요가 없습니다. 선택한 그룹을 선택 해제하려면 다시 한 번 선택하면 됩니다.

3. IAM 그룹에 IAM 사용자 할당을 마쳤으면 저장을 선택하세요.

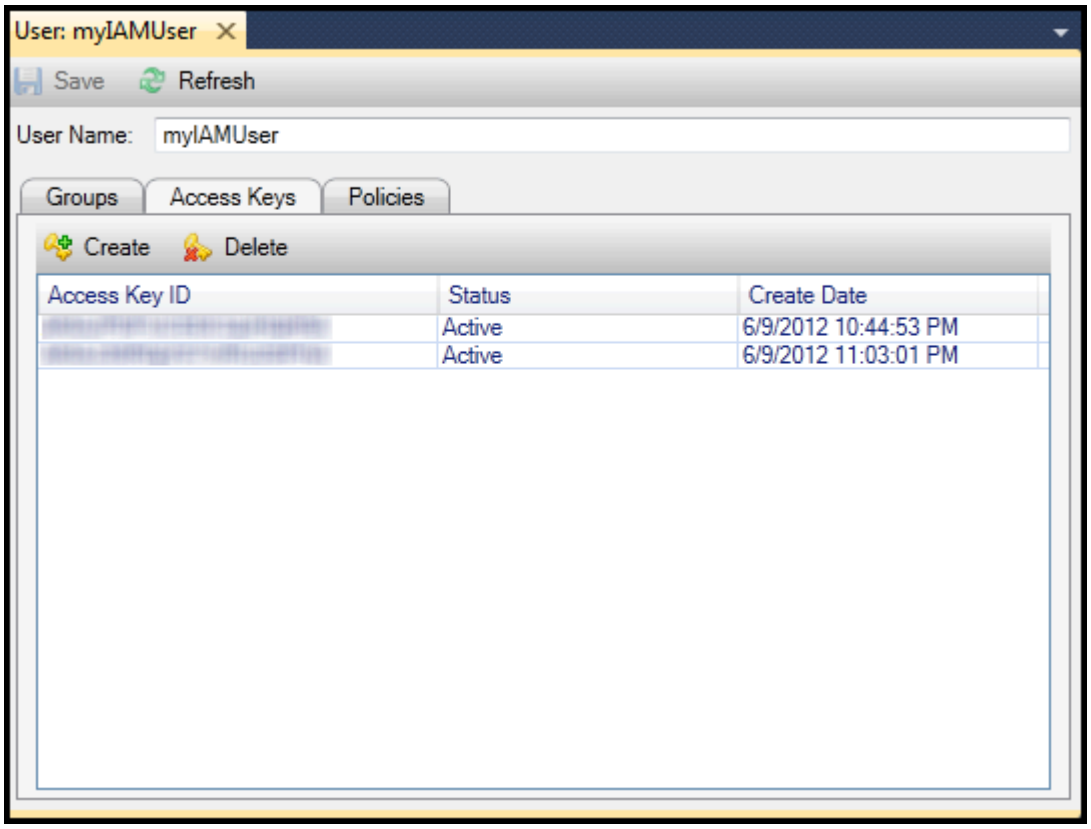
IAM 사용자의 보안 인증 정보 생성

Toolkit for Visual Studio를 사용하면 AWS로 호출하는 API를 만드는 데 사용되는 액세스 키 ID 및 비밀 키를 생성할 수 있습니다. 또한 이러한 키는 툴킷을 통해 Amazon Web Services에 액세스되도록 지정할 수 있습니다. 도구 키트에서 사용할 자격 증명을 지정하는 방법에 대한 자세한 내용은 자격 증명을 참조하십시오. 자격 증명을 안전하게 처리하는 방법에 대한 자세한 내용은 AWS 액세스 키 관리 모범 사례를 참조하세요.

이 툴킷은 IAM 사용자의 암호를 생성하는 데 사용할 수 없습니다.

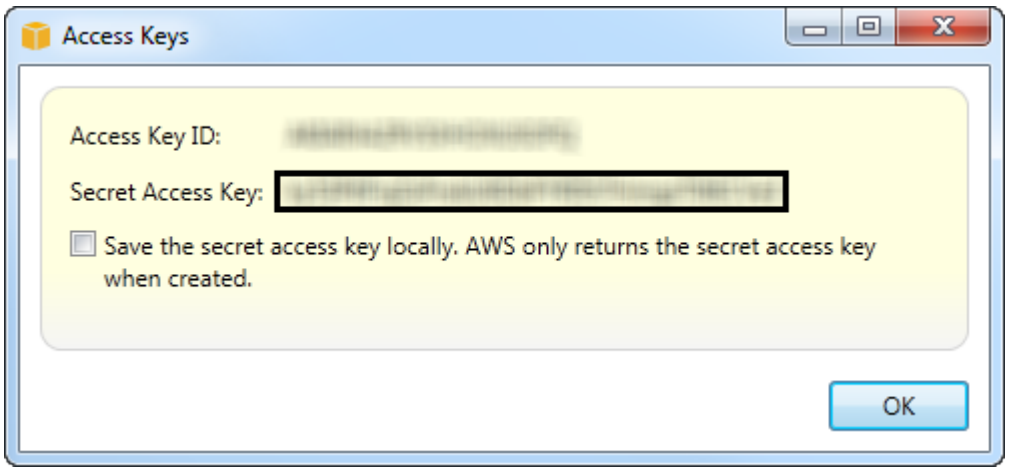
IAM 사용자의 자격 증명을 생성하려면

1. AWS 탐색기에서 IAM 사용자의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 열고 편집을 선택합니다.



2. 액세스 키 탭에서 자격 증명을 생성하려면 생성을 선택합니다.

   IAM 사용자당 두 세트의 자격 증명만 생성할 수 있습니다. 이미 두 세트의 자격 증명이 있지만 추가로 한 세트를 더 생성해야 하는 경우 기존 세트 중 하나를 삭제해야 합니다.

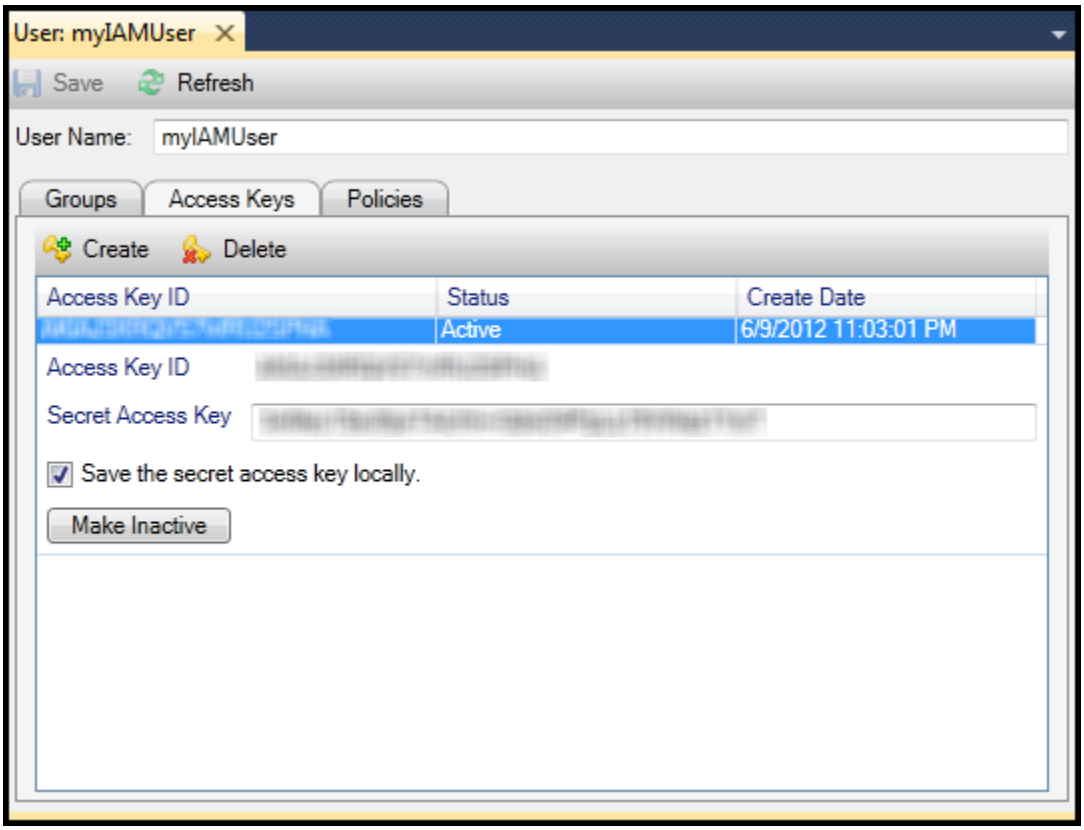


reate credentials for IAM user

도구 키트가 보안 액세스 키의 암호화된 사본을 로컬 드라이브에 저장하도록 하려면 보안 액세스 키 로컬에 저장을 선택합니다.는 생성 시 보안 액세스 키 AWS 만 반환합니다. 또한 대화 상자에서 보안 액세스 키를 복사해 안전한 위치에 저장할 수 있습니다.

3. 확인을 선택합니다.

자격 증명을 생성하면 액세스 키 탭에서 볼 수 있습니다. 도구 키트에서 보안 키를 로컬로 저장하는 옵션을 선택했으면 여기에 표시됩니다.



Create credentials for IAM user

보안 키를 직접 저장하고 도구 키트에서도 저장되도록 하려면 Secret Access Key(보안 액세스 키) 상자에서 보안 액세스 키를 입력한 다음 Save the secret access key locally(로컬 방식으로 보안 액세스 키 저장)를 선택합니다.

자격 증명을 비활성화하려면 Make Inactive(비활성화)를 선택합니다. (보안 인증 정보가 손상된 것으로 의심되는 경우 이 방법을 사용할 수 있습니다. 보안 인증 정보가 안전하다는 보장을 받으면 보안 인증 정보를 다시 활성화할 수 있습니다.)

IAM 역할 생성

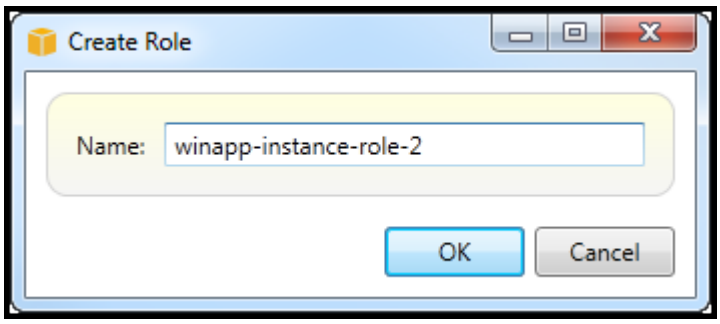
Toolkit for Visual Studio는 IAM 역할의 생성 및 구성을 지원합니다. 사용자 및 그룹과 같이 IAM 역할에 정책을 연결할 수 있습니다. 그런 다음 IAM 역할을 Amazon EC2 인스턴스와 연결할 수 있습니다. EC2 인스턴스와의 연결은 역할에 대한 논리 컨테이너인 인스턴스 프로파일을 통해 처리됩니다. EC2 인스턴스에서 실행되는 애플리케이션은 IAM 역할과 연결된 정책이 지정하는 액세스 수준을 자동으로 부여합니다. 애플리케이션이 다른 AWS 자격 증명을 지정하지 않은 경우에도 마찬가지입니다.

예를 들어, 역할을 생성하고 Amazon S3 액세스만 제한하는 해당 역할에 정책을 연결할 수 있습니다. 이 역할을 EC2 인스턴스와 연결하면 해당 인스턴스에서 애플리케이션을 실행할 수 있으며, 애플리케이션은 Amazon S3에는 액세스하지만 다른 서비스 또는 리소스에는 액세스하지 않습니다. 이 접근 방식의 장점은 EC2 인스턴스에서 AWS 자격 증명을 안전하게 전송하고 저장하는 데 걱정할 필요가 없다는 것입니다.

IAM 역할에 대한 자세한 정보는 IAM 사용 설명서에서 IAM 역할 작업 섹션을 참조하세요. Amazon EC2 인스턴스와 연결된 IAM 역할을 AWS 사용하여에 액세스하는 프로그램의 예는 Java, .NET, PHP 및 Ruby용 AWS 개발자 안내서(IAM을 사용하여 자격 증명 설정, IAM 역할 생성 및 IAM 정책 작업)를 참조하세요.

IAM 역할을 생성하려면

1. AWS Explorer의 Identity and Access Management에서 역할의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 연 다음 역할 생성을 선택합니다.
2. 역할 생성 대화 상자에서 IAM 역할에 대한 이름을 입력하고 확인을 선택하세요.



Create IAM role

새 IAM 역할은 Identity and Access Management의 역할 아래에 표시됩니다.

정책을 생성하고 역할에 연결하는 방법에 대한 자세한 내용은 IAM 정책 생성을 참조하십시오.

IAM 정책 생성

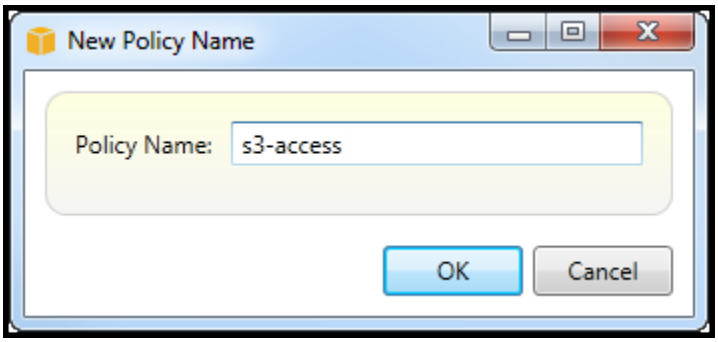
정책은 IAM의 기초입니다. 정책은 사용자, 그룹 또는 역할과 같은 IAM 엔터티와 연결할 수 있습니다. 정책은 사용자, 그룹 또는 역할에 대해 활성화된 액세스 수준을 지정합니다.

IAM 정책을 만들려면

AWS 탐색기에서 AWS Identity and Access Management 노드를 확장한 다음 정책을 연결할 엔터티 유형(그룹, 역할 또는 사용자)에 맞게 노드를 확장합니다. 예를 들어, IAM 역할의 컨텍스트 메뉴를 열고 편집을 선택합니다.

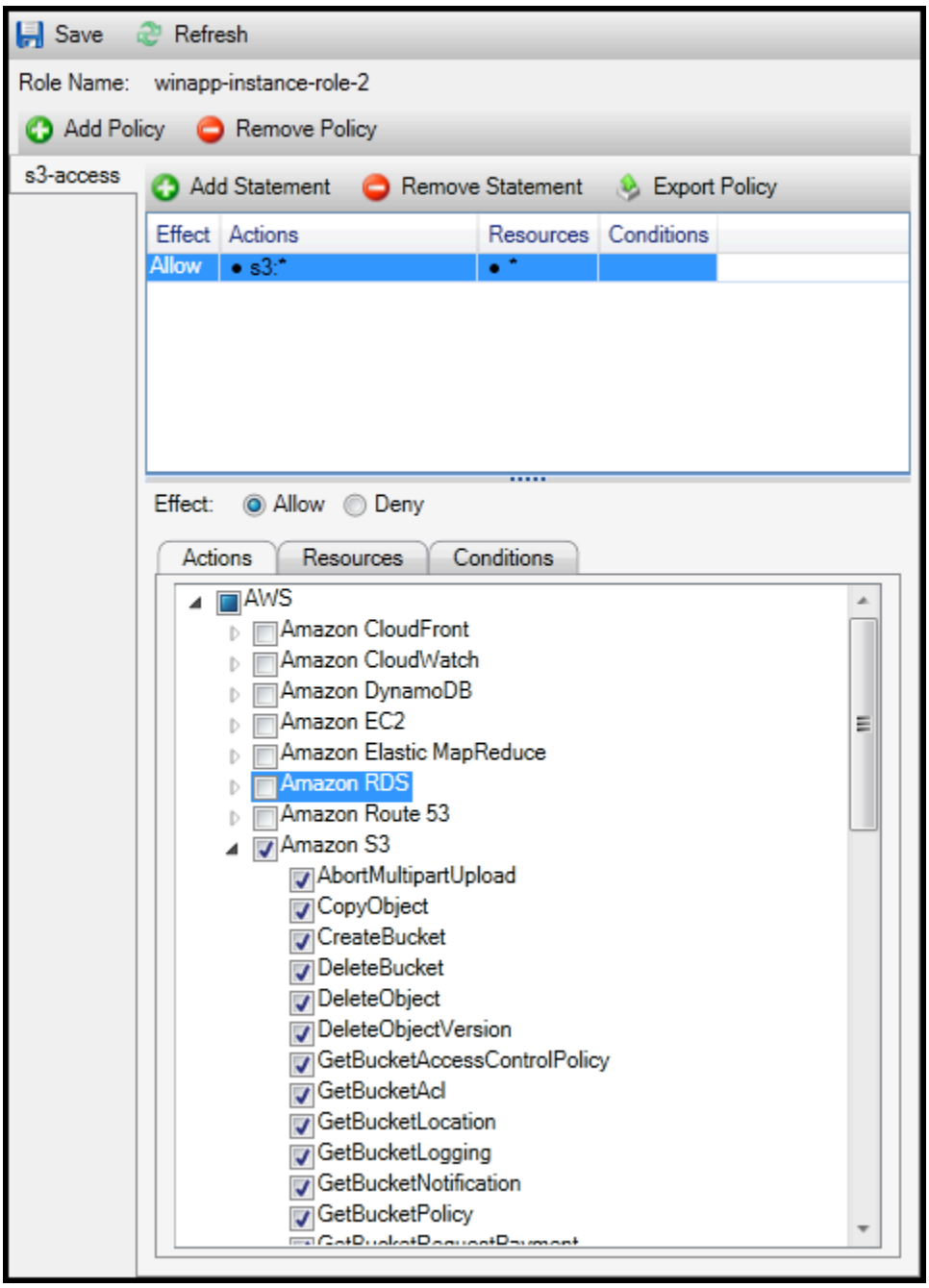
역할과 연결된 탭이 AWS 탐색기에 나타납니다. 정책 추가 링크를 선택합니다.

New Policy Name(새 정책 이름) 대화 상자에서 정책에 대한 이름(예: s3-access)을 입력합니다.



New Policy Name dialog box

정책 편집기에서 역할에 제공할 액세스 수준을 지정하는 정책 명령문을 추가합니다(이 예에서는 정책과 연결된 winapp-instance-role-2). 이 예제에서 정책은 Amazon S3에 대한 전체 액세스를 제공하지만 다른 리소스에 대한 액세스는 제공하지 않습니다.



Specify IAM policy

보다 세밀하게 액세스를 제어하려면 정책 편집기에서 하위 노드를 확장하여 Amazon Web Services와 연결된 작업을 허용 또는 금지할 수 있습니다.

정책을 편집했으면 저장 링크를 선택합니다.

AWS Lambda

를 사용하여 .NET Core 기반 C# Lambda 함수를 개발하고 배포합니다 AWS Toolkit for Visual Studio. AWS Lambda 는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있는 컴퓨팅 서비스입니다. Toolkit for Visual Studio에는 Visual Studio용 AWS Lambda .NET Core 프로젝트 템플릿이 포함되어 있습니다.

에 대한 자세한 내용은 AWS Lambda 개발자 안내서를 AWS Lambda참조하세요.

.NET Core에 대한 자세한 내용은 Microsoft .NET Core 안내서를 참조하세요. Windows, macOS 및 Linux 플랫폼에 대한 .NET Core 사전 조건 및 설치 지침은 .NET Core 다운로드를 참조하십시오.

다음 주제에서는 Toolkit for Visual Studio를 AWS Lambda 사용하여를 사용하는 방법을 설명합니다.

주제

- 기본 AWS Lambda 프로젝트
- 기본 AWS Lambda 프로젝트 생성 Docker 이미지
- 자습서:를 사용하여 서버리스 애플리케이션 빌드 및 테스트 AWS Lambda
- 자습서: Amazon Rekognition Lambda 애플리케이션 생성
- 자습서:에서 Amazon Logging Frameworks AWS Lambda 를 사용하여 애플리케이션 로그 생성

기본 AWS Lambda 프로젝트

AWS Toolkit for Visual Studio에서 Microsoft .NET Core 프로젝트 템플릿을 사용하여 Lambda 함수를 생성할 수 있습니다.

Visual Studio .NET Core Lambda 프로젝트 생성

Lambda-Visual Studio 템플릿과 블루프린트를 사용하여 프로젝트 초기화 속도를 높일 수 있습니다. Lambda 블루프린트에는 미리 작성된 함수가 포함되어 있어 유연한 프로젝트 기반 생성을 간소화해 줍니다.

> **Note**
>
> Lambda 서비스에는 다양한 패키지 유형에 대한 데이터 제한이 있습니다. 데이터 제한에 대한 자세한 내용은 AWS Lambda 사용 설명서의 Lambda 할당량 주제를 참조하세요.

Visual Studio에서 Lambda 프로젝트를 생성하려면

1. Visual Studio에서 파일 메뉴와 새로 만들기를 확장한 다음 프로젝트를 선택합니다.
2. 새 프로젝트 대화 상자에서 언어, 플랫폼 및 프로젝트 유형 드롭다운 상자가 '모두'로 설정되어 있는지 확인하고 검색 필드에 aws lambda를 입력합니다. AWS Lambda 프로젝트(.NET Core - C#) 템플릿을 선택합니다.
3. 이름 필드에 **AWSLambdaSample**을 입력하고 원하는 파일 위치를 지정한 다음 생성을 선택하여 계속 진행합니다.
4. 블루프린트 선택 페이지에서 빈 함수 블루프린트를 선택한 후 완료를 선택하여 Visual Studio 프로젝트를 만듭니다.

프로젝트 파일 검토

검토할 프로젝트 파일은 `aws-lambda-tools-defaults.json` 및 `Function.cs`입니다.

다음 예제는 프로젝트의 일부로 자동으로 생성된 `aws-lambda-tools-defaults.json` 파일을 보여줍니다. 이 파일의 필드를 사용하여 빌드 옵션을 설정할 수 있습니다.

> **Note**
>
> Visual Studio의 프로젝트 템플릿에는 다음과 같은 다양한 필드가 포함되어 있습니다.
>
> - function-handler: Lambda 함수가 실행될 때 실행되는 메서드를 지정합니다.
> - function-handler 필드에 값을 지정하면 게시 마법사에서 해당 값이 미리 채워집니다.
> - 그러나 함수, 클래스 또는 어셈블리의 이름을 바꾸면 `aws-lambda-tools-defaults.json` 파일의 해당 필드를 업데이트해야 합니다.

```
{
  "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."
  ],
```

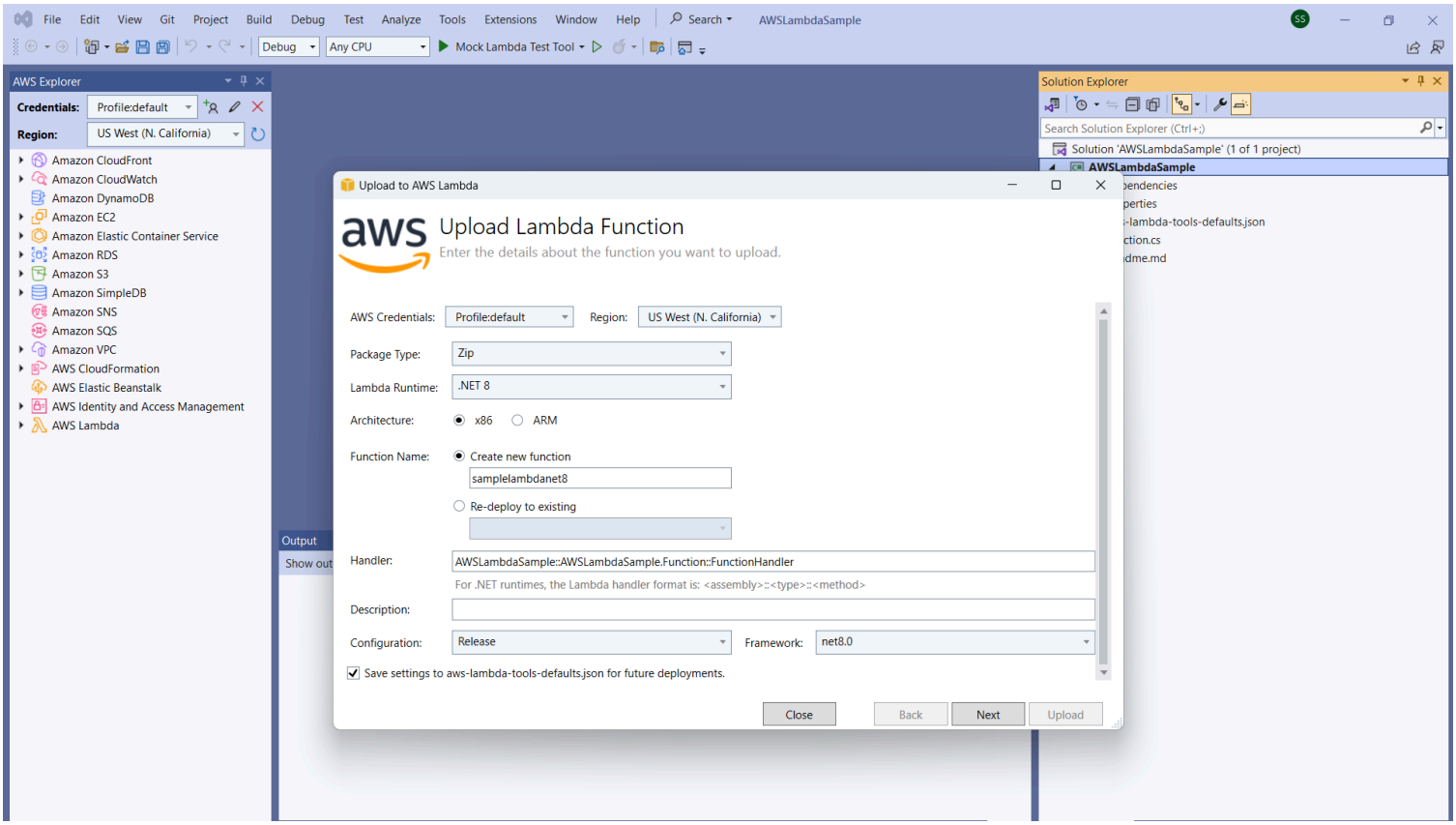
```
  "profile": "default",
  "region": "us-west-2",
  "configuration": "Release",
  "function-architecture": "x86_64",
  "function-runtime": "dotnet8",
  "function-memory-size": 512,
  "function-timeout": 30,
  "function-handler": "AWSLambdaSample::AWSLambdaSample.Function::FunctionHandler"
}
```

Function.cs 파일을 검사합니다. Function.cs는 Lambda 함수로 노출할 c# 함수를 정의합니다. FunctionHandler는 Lambda 함수가 실행될 때 실행되는 Lambda 기능입니다. 이 프로젝트에는 입력 텍스트에서 ToUpper()를 호출하는 FunctionHandler 함수가 하나 정의되어 있습니다.

이제 프로젝트에서 Lambda에 게시할 준비가 되었습니다.

Lambda에 게시

다음 절차와 이미지는 AWS Toolkit for Visual Studio를 사용하여 함수를 Lambda에 업로드하는 방법을 보여줍니다.



Lambda에 함수 게시

1. 보기를 확장하고 AWS 탐색기를 선택하여 AWS 탐색기로 이동합니다.
2. 솔루션 탐색기에서 게시하려는 프로젝트의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 연 다음 AWS Lambda에 게시를 선택하여 Lambda 함수 업로드 창을 엽니다.
3. Lambda 함수 업로드 창에서 다음 필드를 작성합니다.

   a. 패키지 유형: **`Zip`**을 선택합니다. 빌드 프로세스의 결과로 ZIP 파일이 생성되고 Lambda에 업로드됩니다. 또는 패키지 유형 **`Image`**를 선택할 수 있습니다. 자습서: 기본 Lambda 프로젝트 Docker 이미지 생성에서는 패키지 유형 **`Image`**를 사용하여 게시하는 방법을 설명합니다.

   b. Lambda 런타임: 드롭다운 메뉴에서 Lambda 런타임을 선택합니다.

   c. 아키텍처: 원하는 아키텍처의 방사형을 선택합니다.

   d. 함수 이름: 새 함수 생성의 방사형을 선택한 다음 Lambda 인스턴스의 표시 이름을 입력합니다. 이 이름은 AWS 탐색기와 AWS Management Console 디스플레이 모두에서 참조됩니다.

   e. 핸들러: 이 필드를 사용하여 함수 핸들러를 지정합니다. 예를 들어 **`AWSLambdaSample::AWSLambdaSample.Function::FunctionHandler`**입니다.

   f. (선택 사항) 설명: AWS Management Console내의 인스턴스와 함께 표시할 텍스트를 입력합니다.

   g. 구성: 드롭다운 메뉴에서 원하는 구성을 선택합니다.

   h. 프레임워크: 드롭다운 메뉴에서 원하는 프레임워크를 선택합니다.

   i. 설정 저장: 현재 설정을 향후 배포의 기본값으로 `aws-lambda-tools-defaults.json`에 저장하려면 이 상자를 선택합니다.

   j. 다음을 선택하여 고급 함수 세부 정보 창으로 이동합니다.

4. 고급 함수 세부 정보 창에서 다음 필드를 작성합니다.

   a. 역할 이름: 계정과 관련된 역할을 선택합니다. 역할은 함수의 코드에 의해 수행된 모든 AWS 서비스 호출에 대한 임시 자격 증명을 제공합니다. 역할이 없는 경우 드롭다운 선택기에서 스크롤하여 AWS 관리형 정책을 기반으로 새 역할을 찾은 다음 AWSLambdaBasicExecutionRole을 선택합니다. 이 역할에는 최소한의 액세스 권한이 있습니다.

   > **Note**
   >
   > 계정에는 IAM ListPolicies 작업을 실행할 수 있는 권한이 있어야 합니다. 그렇지 않으면 역할 이름 목록이 비게 되어 계속 진행할 수 없습니다.

b. (선택 사항) Lambda 함수가 Amazon VPC의 리소스에 액세스할 경우 서브넷과 보안 그룹을 선택합니다.

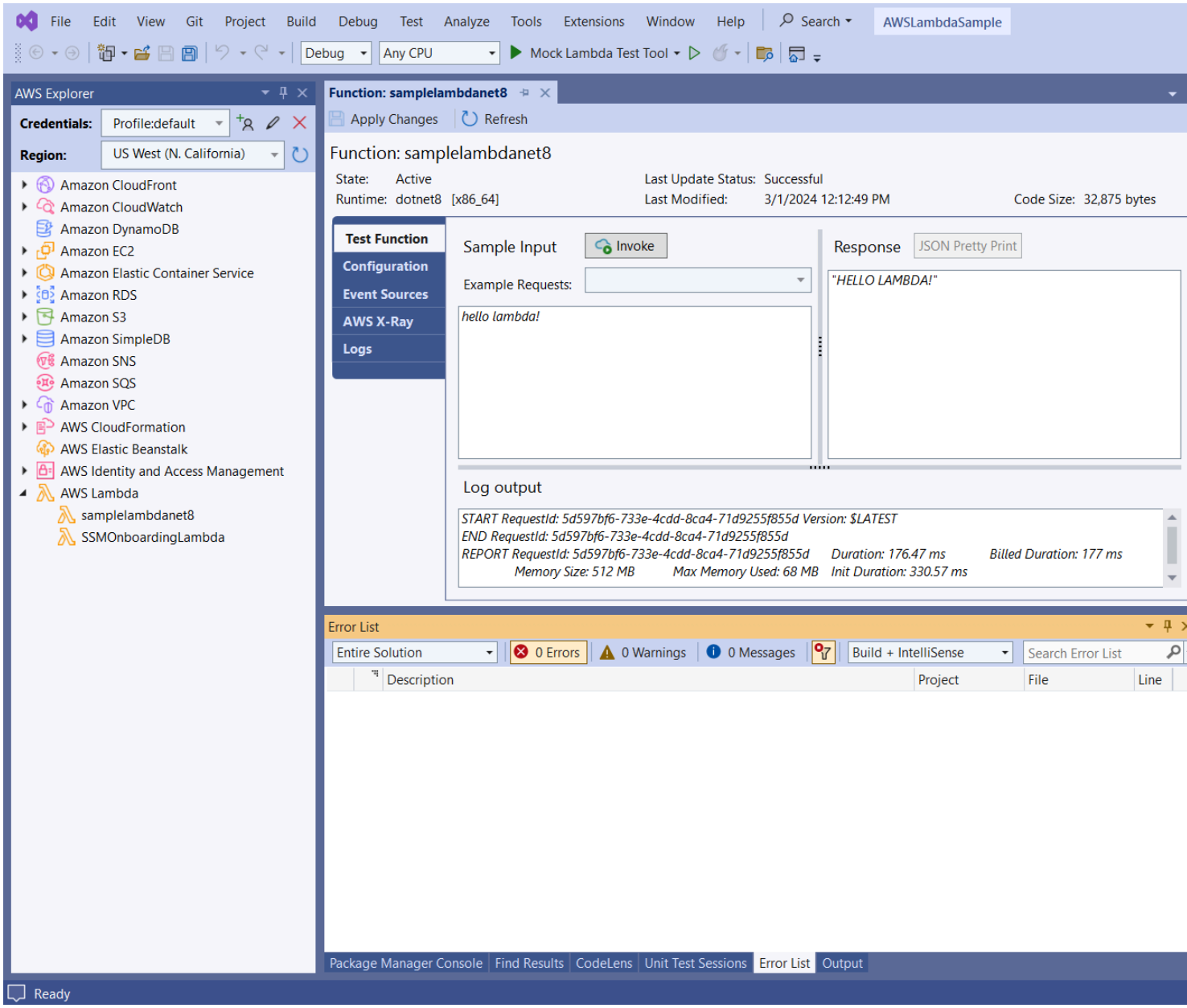
c. (선택 사항) Lambda 함수에 필요한 환경 변수를 설정합니다. 키는 무료 기본 서비스 키로 자동으로 암호화됩니다. 또는 요금이 부과되는 AWS KMS 키를 지정할 수 있습니다. KMS는 데이터 암호화에 사용하는 암호화 키를 생성하고 제어하는 데 사용할 수 있는 관리형 서비스입니다. AWS KMS 키가 있는 경우 목록에서 키를 선택할 수 있습니다.

5. 업로드를 선택하여 함수 업로드 창을 열고 업로드 프로세스를 시작합니다.

> **Note**
>
> 함수가 업로드되는 동안 함수 업로드 페이지가 표시됩니다 AWS. 보고서를 볼 수 있도록 마법사를 열어 두려면 업로드가 완료되기 전에 양식 하단에 있는 완료 후 자동으로 마법사 닫기를 지웁니다.
> 함수가 업로드되고 나면 Lambda 함수가 활성화됩니다. 함수: 보기 페이지가 열리고 새 Lambda 함수의 구성이 표시됩니다.

6. 테스트 함수 탭에서 텍스트 입력 필드에 `hello lambda!`를 입력한 다음 간접 호출을 선택하여 Lambda 함수를 수동으로 간접 호출합니다. 텍스트가 대문자로 변환되어 응답 탭에 표시됩니다.

> **Note**
>
> AWS 탐색기의 AWS Lambda 노드 아래에 있는 배포된 인스턴스를 두 번 클릭하여 언제든지 함수: 보기를 다시 열 수 있습니다.



7. (선택 사항) Lambda 함수를 성공적으로 게시했는지 확인하려면에 로그인 AWS Management Console 한 다음 Lambda를 선택합니다. 콘솔에는 방금 생성한 함수를 포함하여 게시된 Lambda 함수가 모두 표시됩니다.

정리

이 예제를 사용하여 개발을 계속하지 않을 경우 계정에서 사용하지 않은 리소스에 대한 요금이 청구되지 않도록 배포된 함수를 삭제합니다.

> **Note**
>
> Lambda는 자동으로 Lambda 함수를 모니터링하고 Amazon CloudWatch를 통해 지표를 보고합니다. 함수를 모니터링하고 문제를 해결하려면 AWS Lambda 개발자 안내서의 Amazon CloudWatch를 사용한 AWS Lambda 함수 문제 해결 및 모니터링을 참조하세요.

함수 삭제

1. AWS 탐색기에서 AWS Lambda 노드를 확장합니다.
2. 배포된 인스턴스를 마우스 오른쪽 버튼으로 클릭한 다음 삭제를 선택합니다.

기본 AWS Lambda 프로젝트 생성 Docker 이미지

Toolkit for Visual Studio를 사용하여 AWS Lambda 함수를 도커 이미지로 배포할 수 있습니다. Docker를 사용하면 런타임을 더 잘 제어할 수 있습니다. 예를 들어 .NET 8.0과 같은 사용자 지정 런타임을 선택할 수 있습니다. 다른 컨테이너 이미지와 동일한 방식으로 도커 이미지를 배포합니다. 이 자습서는 자습서: 기본 Lambda 프로젝트와 비슷하지만 두 가지 차이점이 있습니다.

- Dockerfile은 프로젝트에 포함되어 있습니다.
- 대체 게시 구성이 선택됩니다.

Lambda 컨테이너 이미지에 대한 자세한 정보는 AWS Lambda 개발자 안내서의 Lambda 배포 패키지를 참조하세요.

Lambda 작업에 대한 자세한 내용은이 사용 설명서의 주제에서 템플릿 사용을 AWS Toolkit for Visual Studio참조하세요. AWS LambdaAWS Toolkit for Visual Studio

Visual Studio .NET Core Lambda 프로젝트 생성

Lambda Visual Studio 템플릿과 블루프린트를 사용하여 프로젝트 초기화 속도를 높일 수 있습니다. Lambda 블루프린트에는 미리 작성된 함수가 포함되어 있어 유연한 프로젝트 기반 생성을 간소화해 줍니다.

Visual Studio .NET Core Lambda 프로젝트 생성

1. Visual Studio에서 파일 메뉴와 새로 만들기를 확장한 다음 프로젝트를 선택합니다.

2. 새 프로젝트 대화 상자에서 언어, 플랫폼 및 프로젝트 유형 드롭다운 상자가 '모두'로 설정되어 있는지 확인하고 검색 필드에 **aws lambda**를 입력합니다. AWS Lambda 프로젝트(.NET Core - C#) 템플릿을 선택합니다.
3. 프로젝트 이름 필드에 **AWSLambdaDocker**를 입력하고 파일 위치를 지정한 다음 생성을 선택합니다.
4. 블루프린트 선택 페이지에서 .NET 8(컨테이너 이미지) 블루프린트를 선택한 다음 완료를 선택하여 Visual Studio 프로젝트를 생성합니다. 이제 프로젝트의 구조와 코드를 검토할 수 있습니다.

프로젝트 파일 검토

다음 섹션에서는 .NET 8(컨테이너 이미지) 블루프린트에서 생성된 세 가지 프로젝트 파일을 살펴봅니다.

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

1. Dockerfile

Dockerfile은 세 가지 기본 작업을 수행합니다.

- FROM: 이 이미지에 활용할 기본 이미지를 설정합니다. 이 기본 이미지는.NET 런타임, Lambda 런타임 및 Lambda .NET 프로세스의 진입점을 제공하는 쉘 스크립트를 제공합니다.
- WORKDIR: 이미지의 내부 작업 디렉터리를 로 설정합니다/var/task.
- COPY: 빌드 프로세스에서 생성된 파일을 로컬 위치에서 이미지의 작업 디렉터리로 복사합니다.

다음은 지정할 수 있는 선택적 Dockerfile 작업입니다.

- ENTRYPOINT: 이 기본 이미지에는 이미지가 시작될 때 실행되는 시작 프로세스인 ENTRYPOINT가 이미 포함되어 있습니다. 직접 지정하려는 경우 해당 기본 진입점을 재정의해야 합니다.
- CMD: 실행할 AWS 사용자 지정 코드를 지시합니다. 사용자 지정 메서드에는 완전한 이름이 필요합니다. 이 행은 Dockerfile에 직접 포함하거나 게시 프로세스 중에 지정할 수 있습니다.

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

다음은 .NET 8(컨테이너 이미지) 블루프린트에 의해 생성된 Dockerfile의 예입니다.

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

WORKDIR /var/task

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

2. aws-lambda-tools-defaults.json

`aws-lambda-tools-defaults.json` 파일은 Toolkit for Visual Studio 배포 마법사 및 .NET Core CLI의 기본값을 지정하는 데 사용됩니다. 다음 목록은 `aws-lambda-tools-defaults.json` 파일에서 설정할 수 있는 필드를 설명합니다.

- `profile`: AWS 프로필을 설정합니다.
- `region`: 리소스가 저장되는 AWS 리전을 설정합니다.
- `configuration`: 함수를 게시하는 데 사용되는 구성을 설정합니다.
- `package-type`: 배포 패키지 유형을 컨테이너 이미지 또는 .zip 파일 아카이브로 설정합니다.
- `function-memory-size`: 함수의 메모리 할당을 MB 단위로 설정합니다.
- `function-timeout`: 제한 시간은 Lambda 함수를 실행할 수 있는 최대 시간(초)입니다. 이 값을 1초 단위로 최대 15분까지 조정할 수 있습니다.
- `docker-host-build-output-dir`: `Dockerfile`의 지침과 관련된 빌드 프로세스의 출력 디렉터리를 설정합니다.
- `image-command`: Lambda 함수를 실행하려는 메서드의 정규화된 이름, 즉 Lambda 함수를 실행하려는 코드입니다. 구문은 다음과 같습니다. `{Assembly}::{Namespace}.{ClassName}::`

{MethodName} 자세한 정보는 핸들러 서명을 참조하세요. 여기서 image-command를 설정하면 나중에 Visual Studio의 게시 마법사에 이 값이 미리 채워집니다.

다음은 .NET 8(컨테이너 이미지) 블루프린트에 의해 생성된 aws-lambda-tools-defaults.json의 예입니다.

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

3. Function.cs

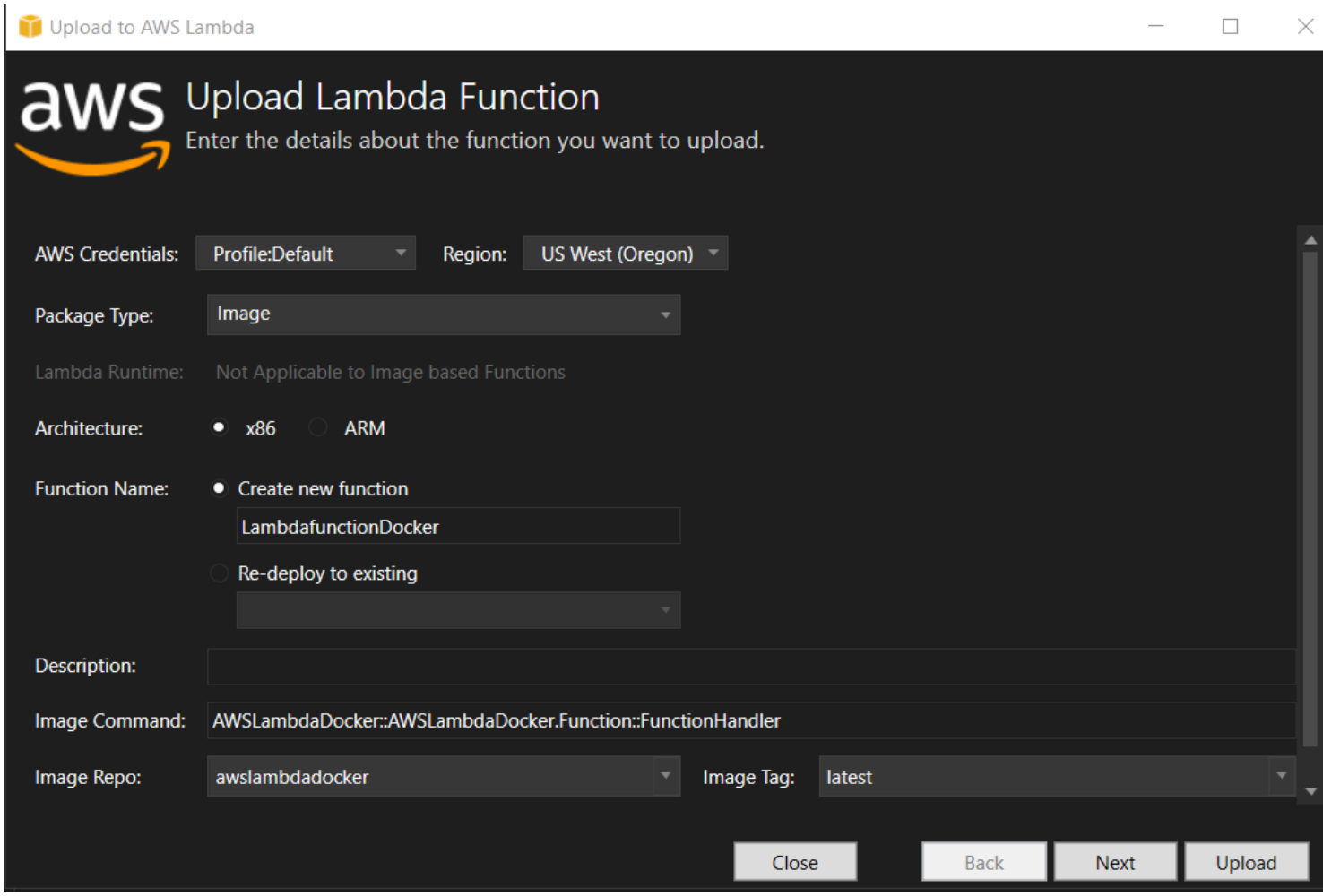
Function.cs 파일은 Lambda 함수로 노출되는 c# 함수를 정의합니다. FunctionHandler는 Lambda 함수가 실행될 때 실행되는 Lambda 기능입니다. 이 프로젝트에서 FunctionHandler는 입력 텍스트의 ToUpper()를 직접적으로 호출합니다.

Lambda에 게시

빌드 프로세스에서 생성되는 도커 이미지는 Amazon Elastic Container Registry(Amazon ECR)에 업로드됩니다. Amazon ECR은 개발자가 도커 컨테이너 이미지를 저장, 관리 및 배포할 수 있게 해주는 완벽한 관리형 도커 컨테이너 레지스트리입니다. Amazon ECR은 이미지를 호스팅하며, Lambda는 호출 시 이를 참조하여 프로그래밍된 Lambda 함수를 제공합니다.

Lambda에 함수 게시

1. 솔루션 탐색기에서 프로젝트의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 연 다음 AWS Lambda에 게시를 선택하여 Lambda 함수 업로드 창을 엽니다.
2. Lambda 함수 업로드 페이지에서 다음을 수행합니다.



   a. 패지지 유형의 경우 게시 마법사가 프로젝트 내에서 `Dockerfile`을 발견했기 때문에 **`Image`**가 패키지 유형으로 자동 선택되었습니다.

   b. 함수 이름에 Lambda 인스턴스의 표시 이름을 입력하세요. 이 이름은 Visual Studio의 AWS 탐색기 및 AWS Management Console에서 모두 표시되는 참조 이름입니다.

   c. 설명에는 AWS Management Console의 인스턴스와 함께 표시할 텍스트를 입력하세요.

   d. 이미지 명령에는 Lambda 함수를 실행하려는 메서드의 정규화된 경로를 입력하세요. **`AWSLambdaDocker::AWSLambdaDocker.Function::FunctionHandler`**

> **Note**
>
> 여기에 입력된 모든 메서드 이름은 Dockerfile 내의 모든 CMD 명령을 무시합니다. `Dockerfile`에 Lambda 함수를 시작하는 방법을 지시하는 `CMD`가 포함된 경우에만 이미지 명령 입력은 선택 사항입니다.

e. 이미지 리포지토리에는 새로운 또는 기존의 Amazon Elastic Container Registry 이름을 입력하세요. 빌드 프로세스에서 생성되는 도커 이미지는 이 레지스트리에 업로드됩니다. 게시되는 Lambda 정의는 해당 Amazon ECR 이미지를 참조합니다.

f. 이미지 태그의 경우 리포지토리의 이미지와 연결할 도커 태그를 입력하세요.

g. 다음을 선택합니다.

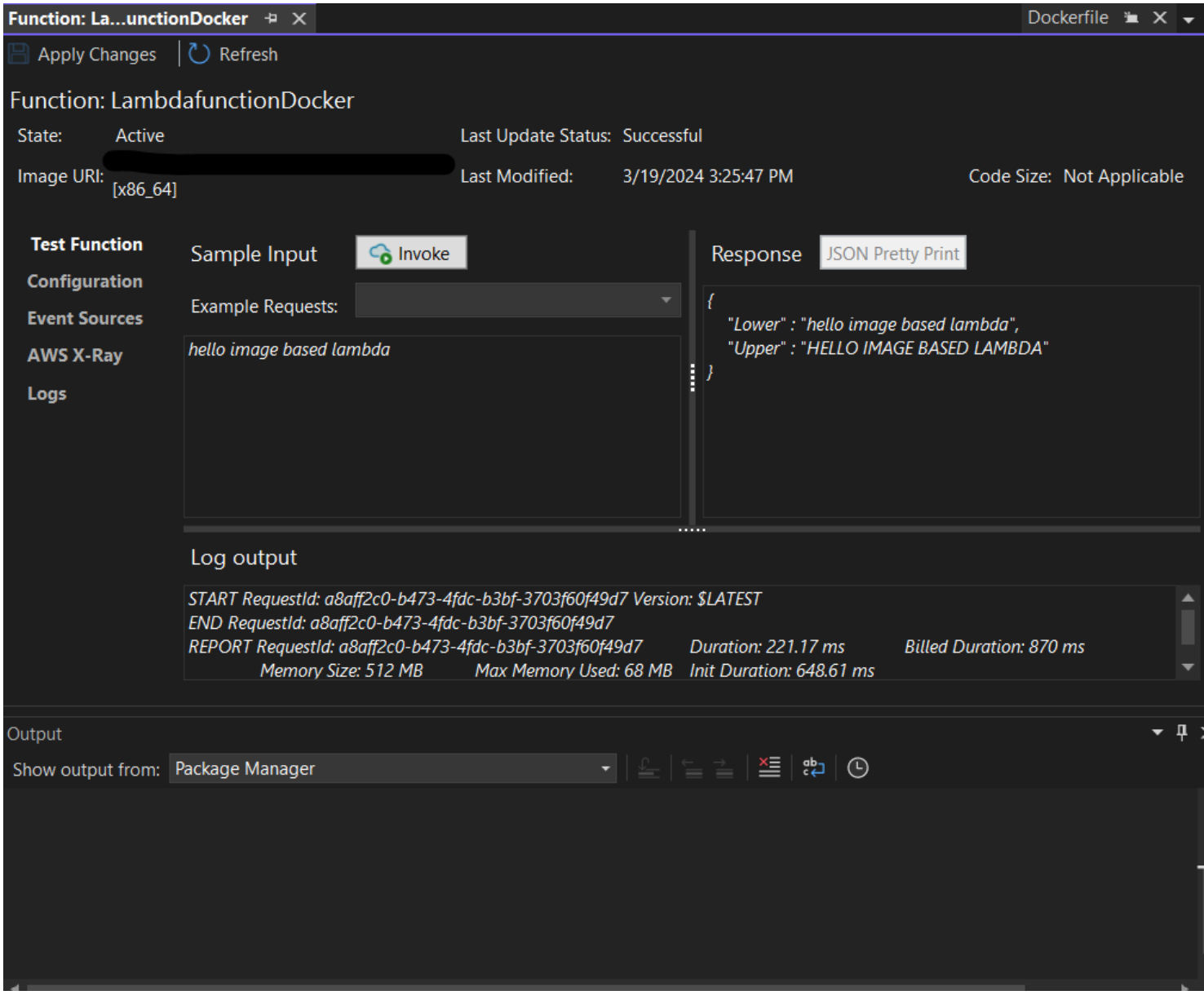
3. 고급 함수 세부 정보 페이지의 역할 이름에서 계정과 관련된 역할을 선택하세요. 역할은 함수의 코드에 의해 수행된 Amazon Web Services 호출에 대한 임시 보안 인증 정보를 제공하는 데 사용됩니다. 역할이 없는 경우 AWS 관리형 정책을 기반으로 새 역할을 선택한 다음 AWSLambdaBasicExecutionRole을 선택합니다.

> **Note**
>
> 계정에는 IAM ListPolicies 작업을 실행할 수 있는 권한이 있어야 합니다. 그렇지 않으면 역할 이름 목록이 비게 됩니다.

4. 업로드를 선택하여 업로드 및 게시 프로세스를 시작합니다.

> **Note**
>
> 함수를 업로드하는 동안 함수 업로드 페이지가 표시됩니다. 그런 다음 게시 프로세스는 구성 파라미터를 기반으로 이미지를 빌드하고, 필요한 경우 Amazon ECR 리포지토리를 생성하고, 이미지를 리포지토리에 업로드하고, 해당 이미지와 함께 해당 리포지토리를 참조하는 Lambda를 생성하세요.
> 함수가 업로드되면 함수 페이지가 열리고 새 Lambda 함수의 구성이 표시됩니다.

5. Lambda 함수를 수동으로 호출하려면 테스트 함수 탭에서 요청 자유 텍스트 입력 필드에 `hello image based lambda`를 입력한 다음 호출을 선택하세요. 대문자로 변환된 텍스트는 응답에 표시됩니다.



6. 리포지토리를 보려면 AWS 탐색기의 Amazon Elastic Container Service에서 리포지토리를 선택하세요.

   AWS 탐색기의 AWS Lambda 노드 아래에 있는 배포된 인스턴스를 두 번 클릭하여 언제든지 함수: 보기를 다시 열 수 있습니다.

> **Note**
>
> AWS 탐색기 창이 열려 있지 않은 경우 보기 -> AWS 탐색기를 통해 도킹할 수 있습니다.

7. 구성 탭에 있는 추가 이미지별 구성 옵션을 참고하세요. 이 탭은 Dockerfile 내에 지정되었을 수 `ENTRYPOINT`, `CMD`, `WORKDIR`을 재정의하는 방법을 제공합니다. 설명은 업로드/게시 중에 입력한 설명(있는 경우)입니다.

정리

이 예제를 사용하여 개발을 계속하지 않을 경우 계정에서 사용하지 않은 리소스에 대한 요금이 청구되지 않도록 배포된 함수와 ECR 이미지를 삭제해야 합니다.

- AWS 탐색기의 AWS Lambda 노드 아래에 있는 배포된 인스턴스를 마우스 오른쪽 버튼으로 클릭하여 함수를 삭제할 수 있습니다.
- 리포지토리는 AWS 탐색기의 Amazon Elastic Container Service -> 리포지토리에서 삭제할 수 있습니다.

다음 단계

Lambda 이미지 생성 및 테스트에 대한 정보는 Lambda에서 컨테이너 이미지 사용을 참조하세요.

컨테이너 이미지 배포, 권한 및 구성 설정 재정의에 대한 정보는 함수 구성을 참조하세요.

자습서:를 사용하여 서버리스 애플리케이션 빌드 및 테스트 AWS Lambda

AWS Toolkit for Visual Studio 템플릿을 사용하여 서버리스 Lambda 애플리케이션을 빌드할 수 있습니다. Lambda 프로젝트 템플릿에는 AWS 서버리스 애플리케이션 모델(SAM)의 AWS Toolkit for Visual Studio 구현인 서버리스 애플리케이션용 템플릿이 포함되어 있습니다. AWSAWS 이 프로젝트 유형을 사용하면 AWS Lambda 함수 모음을 개발하고 필요한 AWS 리소스와 함께 전체 애플리케이션으로 배포할 수 있습니다.를 사용하여 배포를 오케스트레이션 AWS CloudFormation 할 수 있습니다.

설정에 대한 사전 조건 및 정보는 Toolkit for Visual Studio의 Lambda 템플릿 사용을 AWS Toolkit for Visual Studio참조하세요. AWSAWS

주제

- 새 AWS 서버리스 애플리케이션 프로젝트 생성
- 서버리스 애플리케이션 파일 검토
- 서버리스 애플리케이션 배포
- 서버리스 애플리케이션 테스트

새 AWS 서버리스 애플리케이션 프로젝트 생성

AWS 서버리스 애플리케이션 프로젝트는 서버리스 CloudFormation 템플릿으로 Lambda 함수를 생성합니다. CloudFormation templates를 사용하면 데이터베이스와 같은 추가 리소스를 정의하고, IAM 역할을 추가하고, 한 번에 여러 함수를 배포할 수 있습니다. 이는 단일 AWS Lambda 함수를 개발하고 배포하는 데 중점을 둔 Lambda 프로젝트와 다릅니다.

다음 절차에서는 새로운 AWS 서버리스 애플리케이션 프로젝트를 생성하는 방법을 설명합니다.

1. Visual Studio에서 파일 메뉴와 새로 만들기를 확장한 다음 프로젝트를 선택합니다.
2. 새 프로젝트 대화 상자에서 언어, 플랫폼 및 프로젝트 유형 드롭다운 상자가 '모두...'로 설정되어 있는지 확인하고 검색 필드에 **`aws lambda`**를 입력합니다.
3. 테스트가 포함된AWS 서버리스 애플리케이션(.NET Core - C#) 템플릿을 선택하세요.

> **Note**
>
> 테스트가 포함된AWS 서버리스 애플리케이션(.NET Core - C#) 템플릿이 결과 상단에 채워지지 않을 수 있습니다.

4. 다음을 클릭하여 새 프로젝트 구성 대화 상자를 엽니다.
5. 새 프로젝트 구성 대화 상자에서 이름에 **`ServerlessPowertools`**를 입력한 다음 나머지 필드를 원하는 대로 작성합니다. 생성 버튼을 선택하여 블루프린트 선택 대화 상자로 이동합니다.
6. 블루프린트 선택 대화 상자에서 AWS Lambda용 Powertools 블루프린트를 선택한 후 완료를 선택하여 Visual Studio 프로젝트를 만듭니다.

서버리스 애플리케이션 파일 검토

다음 섹션에서는 프로젝트에 대해 생성된 3개의 서버리스 애플리케이션 파일을 자세히 살펴봅니다.

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

1. serverless.template

`serverless.template` 파일은 Serverless 함수 및 기타 AWS 리소스를 선언하기 위한 AWS CloudFormation 템플릿입니다. 이 프로젝트에 포함된 파일에는 Amazon API Gateway를 통해 HTTP

Get 작업으로 노출되는 단일 Lambda 함수에 대한 선언이 포함되어 있습니다. 이 템플릿을 편집하여 기존 함수를 사용자 지정하거나 애플리케이션에 필요한 함수 및 기타 리소스를 더 추가할 수 있습니다.

다음은 serverless.template 파일의 예제입니다.

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

많은 `...AWS:: Serverless::Function...` 선언 필드가 Lambda 프로젝트 배포의 필드와 유사합니다. Powertools 로깅, 지표 및 추적은 다음 환경 변수를 통해 구성됩니다.

- 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

환경 변수에 대한 정의 및 추가 세부 정보는 Powertools for AWS Lambda references 웹 사이트를 참조하세요.

2. Functions.cs

`Functions.cs`는 템플릿 파일에 선언된 단일 함수에 매핑된 C# 메서드가 포함된 클래스 파일입니다. Lambda 함수는 API Gateway의 `HTTP Get` 메서드에 응답합니다. 다음은 `Functions.cs` 파일의 예입니다.

```
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`는 Visual Studio 내의 AWS 배포 마법사 기본값과 .NET Core CLI에 추가된 AWS Lambda 명령을 제공합니다. 다음은 이 프로젝트에 포함된 `aws-lambda-tools-defaults.json` 파일의 예입니다.

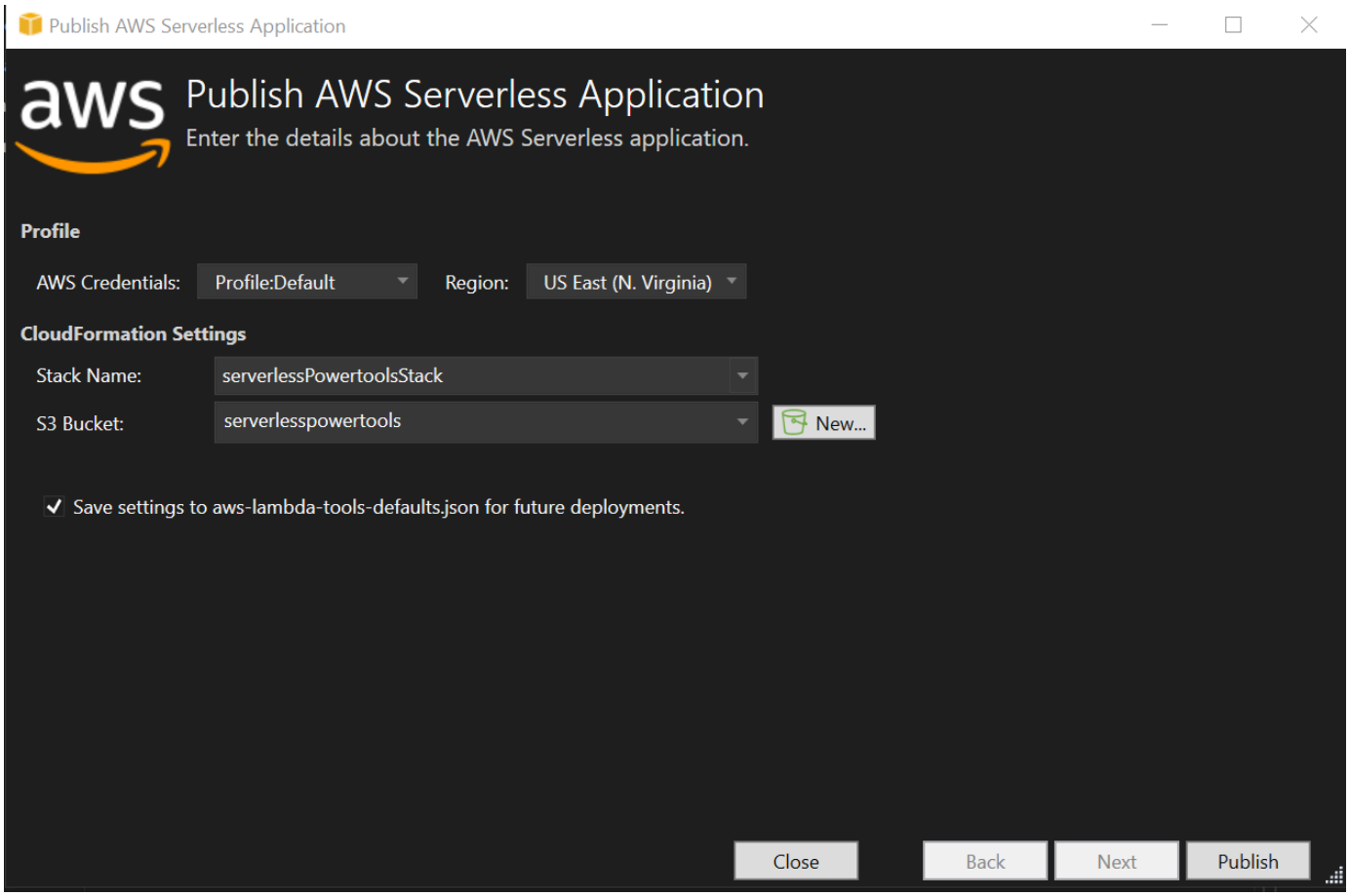
```
{
  "profile": "Default",
  "region": "us-east-1",
  "configuration": "Release",
  "s3-prefix": "ServerlessPowertools/",
  "template": "serverless.template",
  "template-parameters": ""
}
```

서버리스 애플리케이션 배포

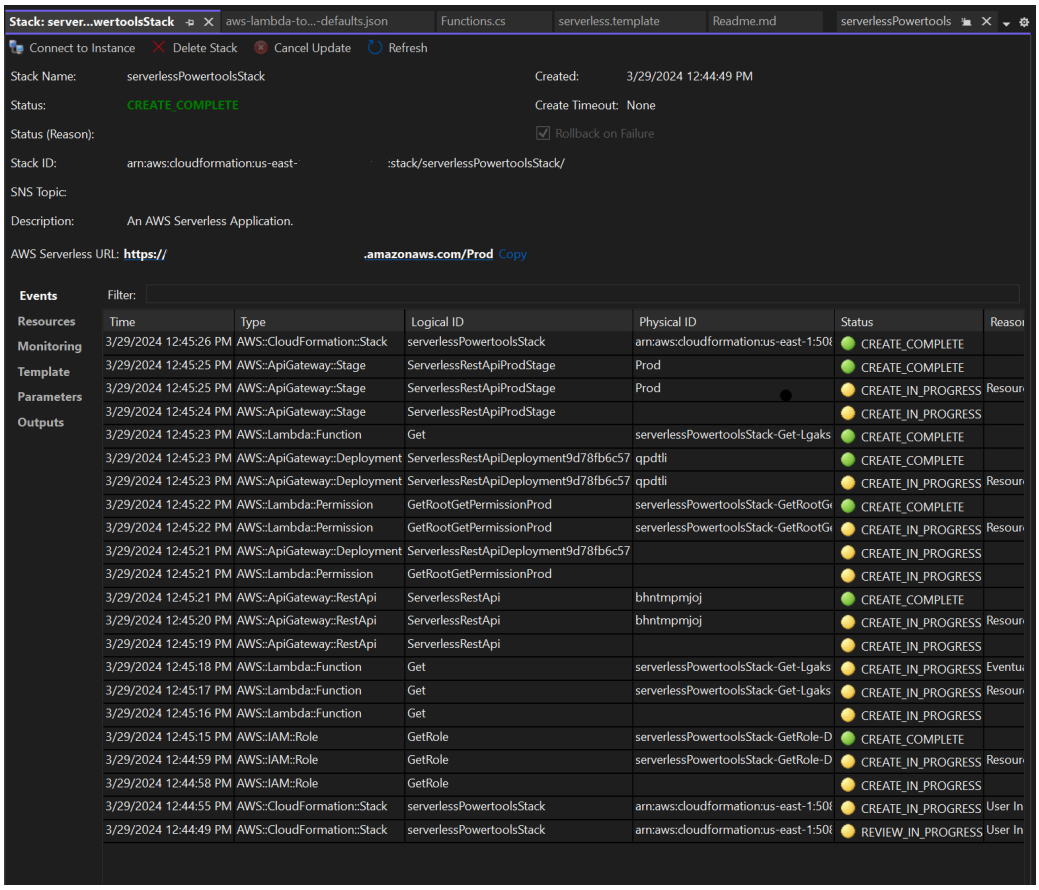
서버리스 애플리케이션을 배포하려면 다음 단계를 완료합니다.

1. 솔루션 탐색기에서 프로젝트의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 열고 AWS Lambda에 게시를 선택하여 AWS 서버리스 애플리케이션 게시 대화 상자를 엽니다.
2. AWS 서버리스 애플리케이션 게시 대화 상자에서 CloudFormation 스택 이름 필드에 스택 컨테이너의 이름을 입력합니다.
3. S3 버킷 필드에서 애플리케이션 번들이 업로드할 Amazon S3 버킷을 선택하거나 새로 만들기... 버튼을 선택하고 새 Amazon S3 버킷의 이름을 입력합니다. 그런 다음 게시를 선택하여 애플리케이션 배포를 게시합니다.

> **Note**
>
> CloudFormation 스택과 Amazon S3 버킷이 동일한 AWS 리전에 있어야 합니다. 프로젝트의 나머지 설정은 `serverless.template` 파일에 정의되어 있습니다.



4. 게시 프로세스 중에 스택 보기 창이 열리고 배포가 완료되면 상태 필드에 CREATE_COMPLETE가 표시됩니다.



서버리스 애플리케이션 테스트

스택 생성이 완료되면 AWS 서버리스 URL을 사용하여 애플리케이션을 볼 수 있습니다. 함수나 파라미터를 추가하지 않고이 자습서를 완료한 경우 AWS 서버리스 URL에 액세스하면 웹 브라우저에 라는 문구가 표시됩니다`Hello Powertools for AWS Lambda (.NET)`.

자습서: Amazon Rekognition Lambda 애플리케이션 생성

이 자습서는 Amazon Rekognition을 사용하여 감지된 레이블이 있는 Amazon S3 객체에 태그를 지정하는 Lambda 애플리케이션을 생성하는 방법을 보여줍니다.

설정에 대한 사전 조건 및 정보는 Toolkit for Visual Studio의 Lambda 템플릿 사용을 AWS Toolkit for Visual Studio참조하세요. AWSAWS

Visual Studio .NET Core 이미지 인식 프로젝트 생성

다음 절차에서는 AWS Toolkit for Visual Studio에서 Amazon Rekognition Lambda 애플리케이션을 생성하는 방법을 설명합니다.

> **Note**
>
> 애플리케이션을 생성하면 두 가지 프로젝트가 포함된 솔루션이 생성됩니다. 이 프로젝트는 Lambda에 배포하기 위한 Lambda 함수 코드가 포함된 소스 프로젝트와 함수를 로컬로 테스트하기 위해 xUnit을 사용하는 테스트 프로젝트입니다.
> 경우에 따라 Visual Studio가 프로젝트에 대한 모든 NuGet 참조를 찾지 못할 수 있습니다. 이는 블루프린트에 NuGet에서 검색해야 하는 종속성이 필요하기 때문입니다. 새 프로젝트가 생성되면 Visual Studio에서는 NuGet에서 로컬 참조만 가져오고 원격 참조는 가져오지 않습니다. NuGet 오류를 해결하려면 참조를 마우스 오른쪽 버튼으로 클릭하고 패키지 복원을 선택합니다.

1. Visual Studio에서 파일 메뉴와 새로 만들기를 확장한 다음 프로젝트를 선택합니다.
2. 새 프로젝트 대화 상자에서 언어, 플랫폼 및 프로젝트 유형 드롭다운 상자가 '모두'로 설정되어 있는지 확인하고 검색 필드에 **`aws lambda`**를 입력합니다.
3. AWS Lambda with Tests(.NET Core - C#) 템플릿을 선택합니다.
4. 다음을 클릭하여 새 프로젝트 구성 대화 상자를 엽니다.
5. 새 프로젝트 구성 대화 상자에서 이름에 'ImageRekognition'을 입력한 다음 나머지 필드를 원하는 대로 작성합니다. 생성 버튼을 선택하여 블루프린트 선택 대화 상자로 이동합니다.
6. 블루프린트 선택 대화 상자에서 이미지 레이블 감지 블루프린트를 선택한 후 완료를 선택하여 Visual Studio 프로젝트를 만듭니다.

> **Note**
>
> 이 청사진은 Amazon S3 이벤트를 수신하기 위한 코드를 제공하며, Amazon Rekognition을 사용하여 레이블을 감지하고 감지된 레이블을 S3 객체에 태그로 추가합니다.

프로젝트 파일 검토

다음 섹션에서는 이러한 프로젝트 파일을 검토합니다.

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

1. Function.cs

Function.cs 파일 내에서 코드의 첫 번째 세그먼트는 파일 상단에 있는 어셈블리 속성입니다. 기본적으로 Lambda는 System.IO.Stream 유형의 반환 유형과 입력 파라미터만 허용합니다. 입력 파라미터와 반환 유형에 대해 입력한 클래스를 사용하려면 직렬 변환기를 등록해야 합니다. 어셈블리 속성은 Lambda JSON 직렬 변환기를 등록합니다. 이 직렬 변환기는 Newtonsoft.Json을 사용하여 스트림을 입력된 클래스로 변환합니다. 어셈블리 수준 또는 메서드 수준에서 직렬 변환기를 설정할 수 있습니다.

다음은 어셈블리 속성에 대한 예시입니다.

```
// 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
```

클래스에는 두 개의 생성자가 있습니다. 첫 번째는 기본 생성자로 Lambda가 함수를 호출할 때 사용됩니다. 이 생성자는 Amazon S3 및 Amazon Rekognition 서비스 클라이언트를 생성합니다. 생성자는 또한 함수를 배포할 때 함수에 할당하는 IAM 역할에서 이러한 클라이언트의 AWS 자격 증명을 검색합니다. 클라이언트의 AWS 리전은 Lambda 함수가 실행 중인 리전으로 설정됩니다. 이 블루프린트에서 Amazon Rekognition 서비스에 레벨에 대한 최소 수준의 신뢰도가 있는 경우 Amazon S3 객체에만 태그를 추가합니다. 이 생성자는 MinConfidence 환경 변수를 확인하여 허용되는 신뢰도 수준을 결정합니다. Lambda 함수를 배포할 때 이 환경 변수를 설정할 수 있습니다.

다음은 Function.cs의 첫 번째 클래스 생성자의 예입니다.

```
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}");
    }
}
```

다음 예제에서는 두 번째 생성자를 테스트에 활용하는 방법을 보여줍니다. 테스트 프로젝트는 고유한 S3 및 Rekognition 클라이언트를 구성하여 전달합니다.

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

다음은 `Function.cs` 파일 내 `FunctionHandler` 메서드의 예입니다.

```
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;
}
```

FunctionHandler는 인스턴스를 구성한 후 Lambda가 호출하는 메서드입니다. 입력 파라미터는 S3Event이 아닌 Stream 유형이어야 합니다. 등록된 Lambda JSON 직렬 변환기로 인해 이를 수행할 수 있습니다. S3Event에는 Amazon S3에서 트리거된 이벤트에 대한 모든 정보가 포함되어 있습니다. 함수는 이벤트의 일부인 모든 S3 객체를 통해 반복하며 Rekognition에 레이블을 감지하라고 알려줍니다. 레이블이 감지되면 태그로 S3 객체에 추가됩니다.

> **Note**
>
> 코드에는 Console.WriteLine()에 대한 직접 호출이 포함됩니다. Lambda에서 함수가 실행되면 Console.WriteLine()에 대한 모든 호출이 Amazon CloudWatch Logs로 리디렉션됩니다.

2. aws-lambda-tools-defaults.json

aws-lambda-tools-defaults.json 파일에는 배포 마법사의 일부 필드를 채우도록 블루프린트가 설정한 기본값이 포함되어 있습니다. 이는 .NET Core CLI와의 통합을 통해 명령줄 옵션을 설정할 때에도 유용합니다.

.NET Core CLI 통합에 액세스하려면 함수의 프로젝트 디렉터리로 이동하여 **dotnet lambda help**를 입력합니다.

> **Note**
>
> 함수 핸들러는 간접 호출된 함수에 대한 응답으로 Lambda가 직접 호출할 메서드를 나타냅니다. 이 필드의 형식은 <assembly-name>::<full-type-name>::<method-name>입니다. 유형 이름과 함께 네임스페이스를 포함해야 합니다.

함수 배포

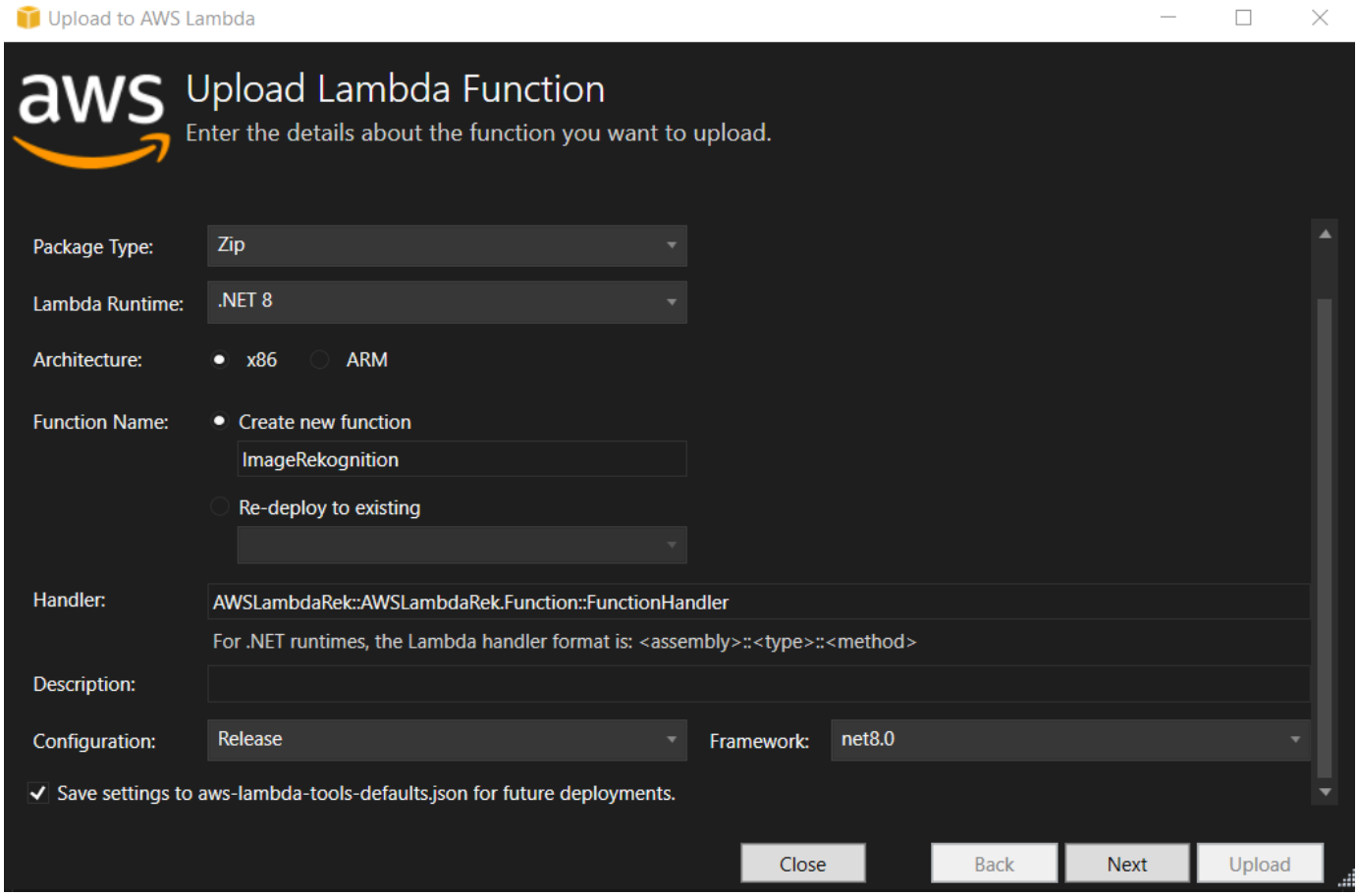
다음 절차에서는 Lambda 함수를 배포하는 방법을 설명합니다.

1. 솔루션 탐색기에서 Lambda 프로젝트를 마우스 오른쪽 버튼으로 클릭하고 AWS Lambda에 게시를 선택하여 업로드 AWS Lambda 대상 창을 엽니다.

   > **Note**
   >
   > 사전 설정 값은 aws-lambda-tools-defaults.json 파일에서 검색됩니다.

2. AWS Lambda에 업로드 창에서 함수 이름 필드에 이름을 입력한 후 다음 버튼을 선택하여 고급 함수 세부 정보 창으로 이동합니다.

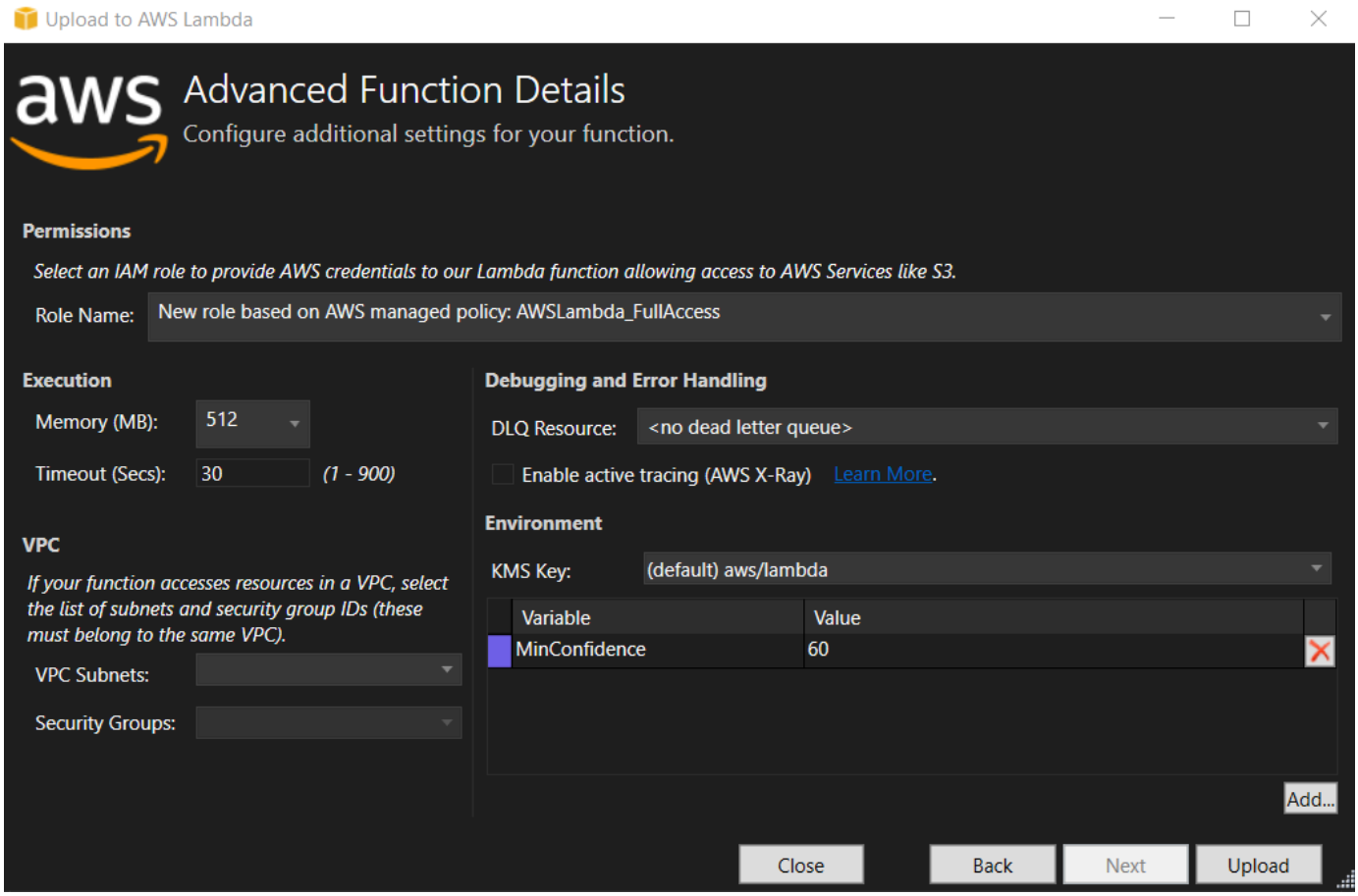
> **Note**
>
> 이 예에서는 함수 이름 **`ImageRekognition`**을 사용합니다.



3. 고급 함수 세부 정보 창에서 Amazon S3 및 Amazon Rekognition 리소스에 액세스할 수 있는 권한을 코드에 부여하는 IAM 역할을 선택합니다.

> **Note**
>
> 이 예제와 함께 따르는 경우 `AWSLambda_FullAccess` 역할을 선택합니다.

4. 환경 변수 `MinConfidence`를 60으로 설정한 다음 업로드를 선택하여 배포 프로세스를 시작합니다. 함수 보기가 AWS 탐색기에 표시되면 게시 프로세스가 완료된 것입니다.



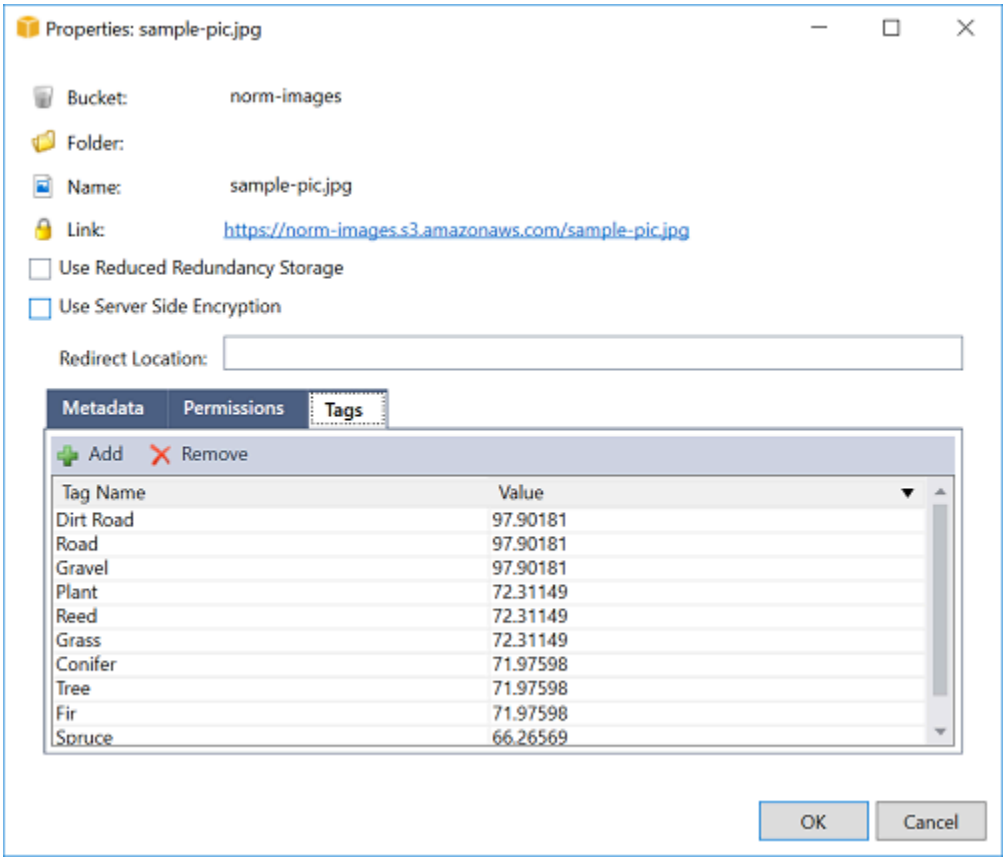
5. 배포가 성공하면 이벤트 소스 탭으로 이동하여 새 함수로 이벤트를 보내도록 Amazon S3를 구성합니다.
6. 이벤트 소스 탭에서 추가 버튼을 선택한 다음 Lambda 함수에 연결할 Amazon S3 버킷을 선택합니다.

> **Note**
>
> 버킷은 Lambda 함수와 동일한 AWS 리전에 있어야 합니다.

함수 테스트

함수가 배포되고 S3 버킷이 함수에 대한 이벤트 소스로 구성되면 선택한 버킷에 대해 AWS 탐색기에서 S3 버킷 브라우저를 엽니다. 그런 다음 일부 이미지를 업로드합니다.

업로드가 완료되면 함수 보기에서 로그를 살펴 함수가 실행되었는지를 확인할 수 있습니다. 또는 버킷 브라우저의 이미지를 마우스 오른쪽 버튼으로 클릭하고 속성을 선택합니다. 태그 탭에서는 객체에 적용된 태그를 볼 수 있습니다.



자습서:에서 Amazon Logging Frameworks AWS Lambda 를 사용하여 애플리케이션 로그 생성

Amazon CloudWatch Logs를 사용하여 애플리케이션의 로그를 모니터링, 저장 및 액세스할 수 있습니다. CloudWatch Logs로 로그 데이터를 가져오려면 AWS SDK를 사용하거나 CloudWatch Logs 에이전트를 설치하여 특정 로그 폴더를 모니터링합니다. CloudWatch Logs는 널리 사용되는 여러 .NET 로깅 프레임워크와 통합되어 워크플로를 간소화합니다.

CloudWatch Logs 및 .NET 로깅 프레임워크 작업을 시작하려면 애플리케이션에 적절한 NuGet 패키지 및 CloudWatch Logs 출력 소스를 추가한 다음 평소와 같이 로깅 라이브러리를 사용합니다. 이렇게 하면 애플리케이션이 .NET 프레임워크로 메시지를 로깅하여 CloudWatch Logs로 전송하고 CloudWatch Logs 콘솔에 애플리케이션의 로그 메시지를 표시할 수 있습니다. 또한 CloudWatch Logs 콘솔에서 애플리케이션의 로그 메시지에 따라 지표 및 경보도 설정할 수 있습니다.

지원되는 .NET 로깅 프레임워크에는 다음이 포함됩니다.

- NLog: 보려면 nuget.org NLog 패키지를 참조합니다.
- Log4net: 보려면 nuget.org Log4net 패키지를 참조합니다.

- ASP.NET Core 로깅 프레임워크: 보려면 nuget.org ASP.NET Core 로깅 프레임워크 패키지를 참조합니다.

다음은 AWS.Logger.NLog NuGet 패키지 및 AWS 대상을에 추가하여 CloudWatch Logs와 콘솔을 모두 로그 메시지의 출력으로 활성화하는 NLog.config 파일의 예입니다NLog.config.

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

로깅 플러그인은 모두를 기반으로 빌드되며 SDK AWS SDK for .NET 와 유사한 프로세스에서 자격 AWS 증명을 인증합니다. 다음 예제에서는 CloudWatch Logs에 액세스하기 위해 로깅 플러그인 자격 증명에 필요한 권한을 자세히 설명합니다.

> **Note**
>
> AWS .NET 로깅 플러그인은 오픈 소스 프로젝트입니다. 추가 정보, 샘플 및 지침은 AWS 로깅 .NET GitHub 리포지토리의 샘플 및 지침 주제를 참조하세요.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
```

```
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogGroups"
      ],
      "Resource": [
        "arn:aws:logs:*:*:*"
      ]
    }
  ]
}
```

AWS에 배포

Toolkit for Visual Studio는 AWS Elastic Beanstalk 컨테이너 또는 CloudFormation 스택에 대한 애플리케이션 배포를 지원합니다.

> **Note**
>
> Visual Studio Express Edition을 사용할 경우,
>
> - 도커 CLI를 사용하여 Amazon ECS 컨테이너에 애플리케이션을 배포할 수 있습니다.
> - AWS 관리 콘솔을 사용하여 Elastic Beanstalk 컨테이너에 애플리케이션을 배포할 수 있습니다.
>
> Elastic Beanstalk 배포의 경우, 먼저 웹 배포 패키지를 생성해야 합니다. 자세한 내용은 Visual Studio에서 웹 배포 패키지를 생성하는 방법을 참조하십시오. Amazon ECS 배포의 경우에는 도커 이미지가 있어야 합니다. 자세한 내용은 Docker용 Visual Studio 도구를 참조하십시오.

주제

Visual Studio에서 AWS에 게시 작업

AWS에 게시는 .NET 애플리케이션을 AWS 배포 대상에 게시하는 데 도움이 되는 대화형 배포 환경으로, .NET Core 3.1 이상을 대상으로 하는 애플리케이션을 지원합니다. AWS에 게시 작업을 통해 다음과 같은 배포 기능을 IDE에서 직접 사용할 수 있으므로 Visual Studio 내에서 작업 흐름을 유지할 수 있습니다.

- 클릭 한 번으로 애플리케이션을 배포할 수 있습니다.
- 애플리케이션을 기반으로 한 배포 권장 사항.
- 배포 대상 환경(배포 대상)과 관련이 있고 요구하는 대로 자동 Dockerfile 생성.

- 배포 대상에서 요구하는 대로 애플리케이션을 빌드하고 패키징하기 위한 설정을 최적화했습니다.

> **Note**
>
> .NET Framework 애플리케이션 게시에 대한 자세한 정보는 Elastic Beanstalk에서.NET 애플리케이션 생성 및 배포 안내서를 참조하세요.
> .NET CLI에서 AWS에 게시에 액세스할 수도 있습니다. 자세한 정보는 AWS에서 .NET 애플리케이션 배포 안내서를 참조하세요.

주제

사전 조건

.NET 애플리케이션을 AWS 서비스에 성공적으로 게시하려면 로컬 디바이스에 다음을 설치하세요.

- .NET Core 3.1+(.NET5 및 .NET6 포함): 이러한 제품에 대한 추가 정보와 다운로드 정보를 보려면 Microsoft 다운로드 사이트를 방문하세요.
- Node.js 14.x 이상 버전: Node.js는 AWS Cloud Development Kit (AWS CDK)를 실행해야 합니다. Node.js 에 대한 자세한 정보를 다운로드하거나 가져오려면 Node.js 다운로드 사이트를 방문하세요.

  > **Note**
  >
  > AWS에 게시는 AWS CDK를 활용하여 애플리케이션과 모든 배포 인프라를 단일 프로젝트로 배포합니다. AWS CDK에 대한 자세한 정보는 Cloud Development Kit 안내서를 참조하세요.

- (선택 사항) 도커는 Amazon ECS와 같은 컨테이너 기반 서비스에 배포할 때 사용됩니다. 자세한 정보를 확인하고 도커를 다운로드하려면 도커 다운로드 사이트를 참조하세요.

지원되는 애플리케이션 유형

새 대상이나 기존 대상에 게시하기 전에 먼저 Visual Studio에서 다음 프로젝트 유형 중 하나를 생성하거나 엽니다.

- ASP.NET Core 애플리케이션
- .NET 콘솔 애플리케이션
- Blazor 웹어셈블리 애플리케이션

AWS 대상에 애플리케이션 게시

새 대상에 게시할 때 AWS에 게시는 권장 사항을 제공하고 공통 설정을 사용하여 프로세스를 안내합니다. 이전에 설정된 대상에 게시해야 하는 경우 기본 설정이 저장되어 조정하거나 원클릭 배포로 즉시 사용할 수 있습니다.

> **Note**
>
> .NET CLI Server와의 툴킷 통합:
> 게시는 localhost에서 .NET 서버 프로세스를 시작하여 게시 프로세스를 수행합니다.

새 대상에 게시

다음은 새 대상에 게시할 때 AWS 배포에 게시 기본 설정을 구성하는 방법에 대한 설명입니다.

1. AWS 탐색기에서 자격 증명 드롭다운 메뉴를 확장한 다음 배포에 필요한 리전 및 AWS 서비스에 해당하는 AWS 프로필을 선택합니다.
2. 리전 드롭다운 메뉴를 확장한 다음 배포에 필요한 AWS 서비스가 포함된 AWS 리전을 선택하세요.
3. Visual Studio 솔루션 탐색기에서 프로젝트 이름의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 연 다음 AWS에 게시를 선택하세요. 그러면 AWS에 게시가 열립니다.
4. AWS에 게시에서 새 대상에 게시를 선택하여 새 배포를 구성하세요.

   > **Note**
   >
   > 기본 배포 보안 인증 정보를 수정하려면 AWS에 게시의 보안 인증 정보 섹션 옆에 있는 편집 링크를 선택하거나 클릭하세요.
   > 대상 구성 프로세스를 우회하려면 기존 대상에 게시를 선택한 다음 이전 배포 대상 목록에서 원하는 구성을 선택하세요.

5. 게시 대상 창에서 애플리케이션 배포를 관리할 AWS 서비스를 선택하세요.
6. 구성에 만족하면 게시를 클릭하여 배포 프로세스를 시작하세요.

> **Note**
>
> 배포를 시작하면 AWS에 게시에 다음과 같은 상태 업데이트가 표시됩니다.
>
> - 배포 프로세스 중에는 AWS에 게시에 배포 진행 상황에 대한 정보가 표시됩니다.
> - 배포 프로세스가 끝나면 AWS에 게시에 배포의 성공 또는 실패 여부를 표시하세요.
> - 배포에 성공하면 리소스 패널에 생성된 리소스에 대한 추가 정보가 제공됩니다. 이 정보는 애플리케이션 유형 및 배포 구성에 따라 달라집니다.

기존 대상에 게시

다음은.NET 애플리케이션을 기존 AWS 대상에 다시 게시하는 방법을 설명합니다.

1. AWS 탐색기에서 자격 증명 드롭다운 메뉴를 확장한 다음 배포에 필요한 리전 및 AWS 서비스에 해당하는 AWS 프로필을 선택합니다.
2. 리전 드롭다운 메뉴를 확장한 다음 배포에 필요한 AWS 서비스가 포함된 AWS 리전을 선택하세요.
3. Visual Studio 솔루션 탐색기 창에서 프로젝트 이름을 마우스 오른쪽 단추로 클릭하고 AWS에 게시시를 선택하여 AWS에 게시를 엽니다.
4. AWS에 게시에서 기존 대상에 게시를 선택하여 기존 대상 목록에서 배포 환경을 선택하세요.

   > **Note**
   >
   > 최근에 AWS 클라우드에 애플리케이션을 게시한 경우 해당 애플리케이션이 AWS에 게시에 표시됩니다.

5. 애플리케이션을 배포할 게시 대상을 선택한 다음 게시를 클릭하여 배포 프로세스를 시작하세요.

.NET Core CLI를 사용하여 AWS Lambda 프로젝트 배포

에는 Visual Studio용 AWS Toolkit for Visual Studio include AWS Lambda .NET Core 프로젝트 템플릿이 포함되어 있습니다. .NET Core 명령줄 인터페이스(CLI)를 사용하여 Visual Studio에 내장된 Lambda 함수를 배포할 수 있습니다.

주제

사전 조건

.NET Core CLI를 사용하여 Lambda 함수를 배포하려면 먼저 다음 사전 조건을 충족해야 합니다.

- Visual Studio 2015 업데이트 3이 설치되어 있는지 확인하세요.
- NET Core for Windows를 설치하세요.
- .NET Core CLI가 Lambda와 함께 작동하도록 설정하세요. 자세한 정보는 AWS Lambda 개발자 안내서의 .NET Core CLI 섹션을 참조하세요.
- Toolkit for Visual Studio를 설치하세요. 자세한 내용은 설치 AWS Toolkit for Visual Studio 단원을 참조하십시오.

관련 주제

.NET Core CLI를 사용하여 Lambda 함수를 배포할 때 유용할 수 있는 정보는 다음과 같습니다.

- Lambda 함수에 대한 자세한 내용은 AWS Lambda 개발자 안내서의 AWS Lambda란 무엇입니까?를 참조하세요.
- Visual Studio에서 Lambda 함수 생성에 대한 자세한 정보는 AWS Lambda 섹션을 참조하세요.
- Microsoft .NET Core에 대한 자세한 정보는 Microsoft 온라인 설명서의 .NET Core를 참조하세요.

.NET Core CLI를 통해 사용 가능한 Lambda 명령 나열

.NET Core CLI를 통해 사용할 수 있는 Lambda 명령을 나열하려면 다음을 수행하세요.

1. 명령 프롬프트 창을 열고 Visual Studio .NET Core Lambda 프로젝트가 포함된 폴더로 이동하세요.
2. `dotnet lambda --help`을 입력합니다.

```
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>
```

.NET Core CLI에서 .NET Core Lambda 프로젝트 게시

다음 지침에서는 Visual Studio에서 AWS Lambda .NET Core 함수를 생성했다고 가정합니다.

1. 명령 프롬프트 창을 열고 Visual Studio .NET Core Lambda 프로젝트가 포함된 폴더로 이동하세요.
2. `dotnet lambda deploy-function`을 입력합니다.
3. 메시지가 나타나면 배포할 함수의 이름을 입력하세요. 새로운 이름이거나 기존 함수의 이름일 수 있습니다.
4. 메시지가 표시되면 AWS 리전(Lambda 함수가 배포될 리전)을 입력합니다.
5. 메시지가 나타나면 함수를 실행할 때 Lambda가 맡을 IAM 역할을 선택하거나 생성하세요.

성공적으로 완료되면 New Lambda function created(새 Lambda 함수 생성됨) 메시지가 표시됩니다.

```
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
```

기존 함수를 배포할 경우 배포 함수는 AWS 리전에 대해서만 요청합니다.

```
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
```

Lambda 함수가 배포되면 해당 함수를 사용할 준비가 되었습니다. 자세한 정보는 AWS Lambda 사용 예제를 참조하세요.

Lambda는 자동으로 Lambda 함수를 모니터링하고 Amazon CloudWatch를 통해 지표를 보고합니다. Lambda 함수를 모니터링하고 문제를 해결하려면 Amazon CloudWatch를 사용한 AWS Lambda 함수 문제 해결 및 모니터링을 참조하세요.

Amazon Q와 함께 AWS Toolkit for Visual Studio를 사용하여 Visual Studio AWS Elastic Beanstalk 의에 배포

AWS Elastic Beanstalk 는 애플리케이션의 AWS 리소스 프로비저닝 프로세스를 간소화하는 서비스입니다. Elastic Beanstalk는 애플리케이션을 배포하는 데 필요한 모든 AWS 인프라를 제공합니다. 이 인프라에는 다음 사항이 포함됩니다.

- 애플리케이션에 대한 실행 파일 및 내용을 호스팅하는 Amazon EC2 인스턴스.
- 애플리케이션을 지원하기 위해 적절한 Amazon EC2 인스턴스 수를 유지하는 오토 스케일링.
- 대부분의 대역폭에서 Amazon EC2 인스턴스로 수신 트래픽을 라우팅하는 Elastic Load Balancing 로드 밸런서.

이 사용 설명서 주제에서는 AWS Toolkit with Amazon Q에서 Elastic Beanstalk 마법사를 사용하는 방법을 설명합니다. Elastic Beanstalk에 대한 자세한 내용은 AWS Elastic Beanstalk 개발자 안내서를 참조하세요. AWS Toolkit with Amazon Q에 대한 Elastic Beanstalk 마법사는 다음 주제 섹션에 설명되어 있습니다.

주제

- Elastic Beanstalk에 기존 ASP.NET 애플리케이션 배포
- Elastic Beanstalk에 ASP.NET Core 애플리케이션 배포(레거시)
- 애플리케이션의 AWS 보안 자격 증명을 지정하는 방법

- 애플리케이션을 Elastic Beanstalk 환경에 재게시하는 방법(레거시)
- 사용자 지정 Elastic Beanstalk 애플리케이션 배포
- 사용자 지정 ASP.NET Core Elastic Beanstalk 배포
- .NET 및 Elastic Beanstalk에 대한 다중 애플리케이션 지원

Elastic Beanstalk에 기존 ASP.NET 애플리케이션 배포

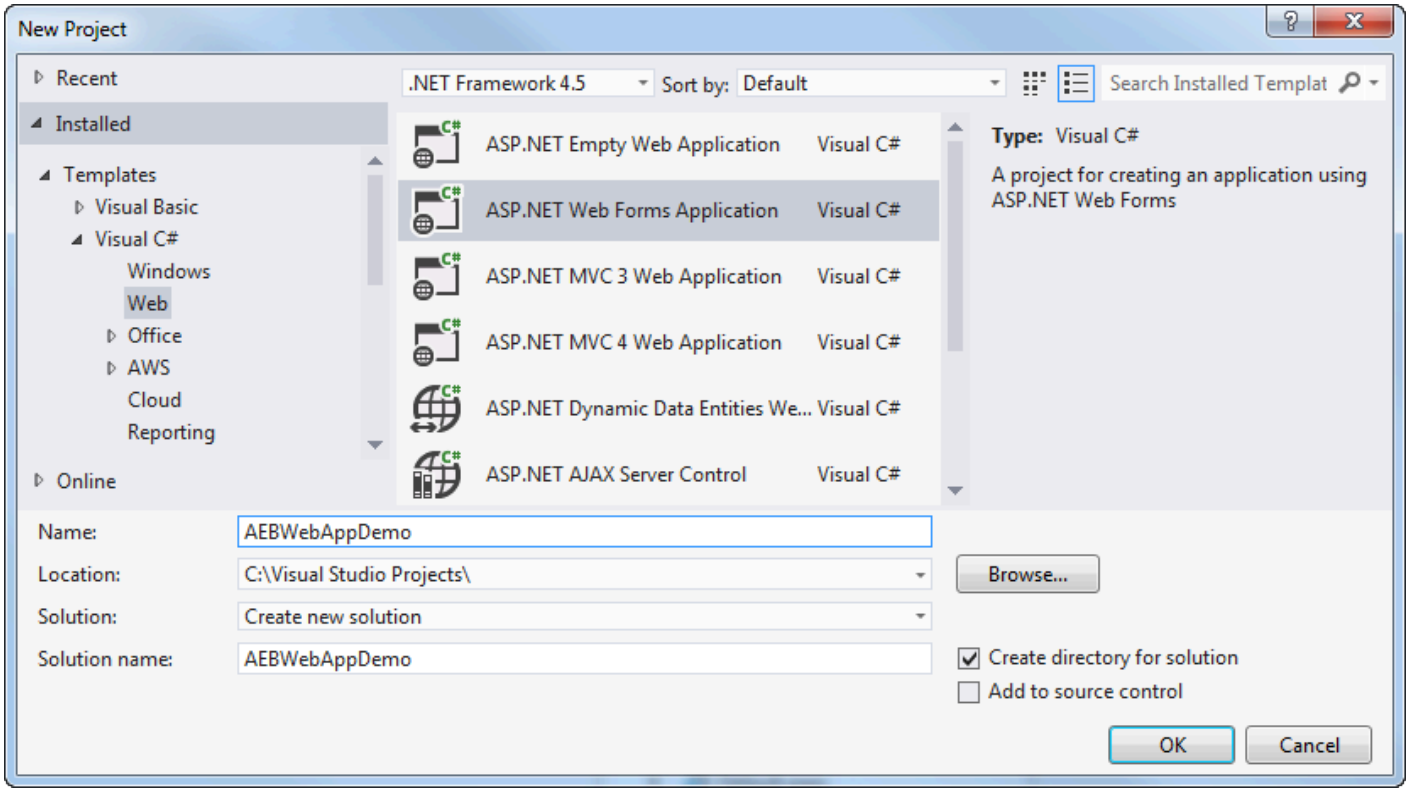
이 섹션에서는 Toolkit for Visual Studio의 일부로 제공되는 Elastic Beanstalk에 게시 마법사를 사용하여 Elastic Beanstalk를 통해 애플리케이션을 배포하는 방법을 설명합니다. 연습을 위해 Visual Studio에 내장되어 있는 웹 애플리케이션 스타터 프로젝트의 인스턴스를 사용하거나 자체의 프로젝트를 사용할 수 있습니다.

> **Note**
>
> ASP.NET Core 애플리케이션 배포도 마법사에서 지원됩니다. ASP.NET Core에 대한 자세한 정보는 AWS .NET 배포 도구 가이드 및 업데이트된 AWS에 배포 목차를 참조하세요.

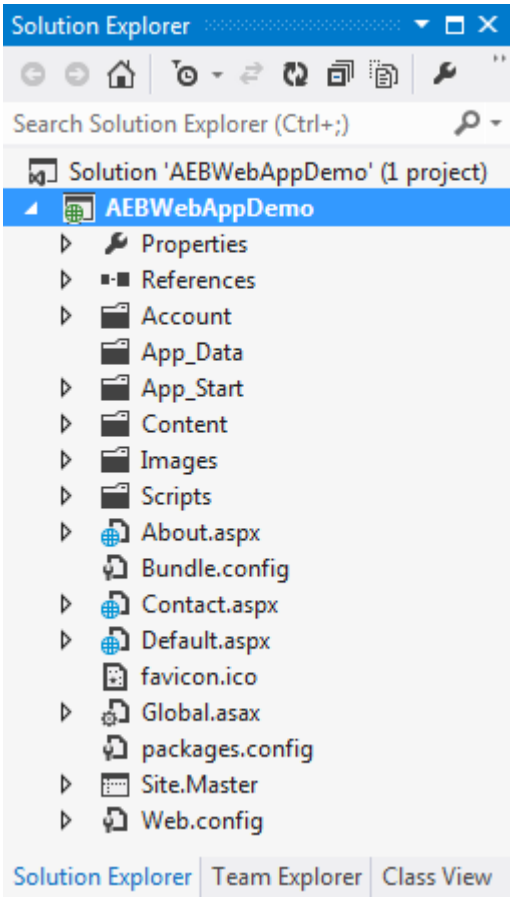
> **Note**
>
> Publish to Elastic Beanstalk 마법사를 사용하려면 먼저 웹 배포를 다운로드하여 설치해야 합니다. 이 마법사는 웹 배포를 통해 IIS(인터넷 정보 서비스) 웹 서버에 웹 애플리케이션과 웹 사이트를 배포합니다.

샘플 웹 애플리케이션 스타터 프로젝트 만들기

1. Visual Studio의 파일 메뉴에서 새로 만들기, 프로젝트를 차례대로 선택합니다.
2. New Project(새 프로젝트) 대화 상자의 탐색 창에서 Installed(설치됨), Templates(템플릿), Visual C#을 차례대로 확장한 다음 Web(웹)을 선택합니다.
3. 프로젝트 템플릿 목록에서 `Web` 및 `Application` 단어가 설명에 포함된 템플릿을 선택합니다. 이 예제에서는 ASP.NET Web Forms Application을 선택합니다.

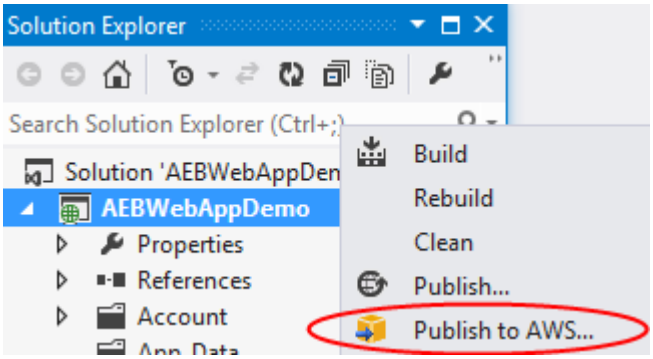


4. 이름 상자에 AEBWebAppDemo를 입력합니다.
5. 위치 상자에서 개발 시스템의 솔루션 폴더에 대한 경로를 입력하거나 찾아보기를 선택한 다음, 솔루션 폴더를 찾아 선택하고, Select Folder(폴더 선택)를 선택합니다.
6. Create directory for solution(솔루션에 대해 디렉터리 생성) 상자가 선택되어 있는지 확인합니다. Solution(솔루션) 드롭다운 목록에서 Create new solution(새 솔루션 생성)이 선택되어 있는지 확인한 다음 확인을 선택합니다. Visual Studio가 ASP.NET Web Forms Application 프로젝트 템플릿에 따라 솔루션과 프로젝트를 생성합니다. 그런 다음 솔루션과 프로젝트가 나타나는 솔루션 탐색기를 Visual Studio에서 표시합니다.

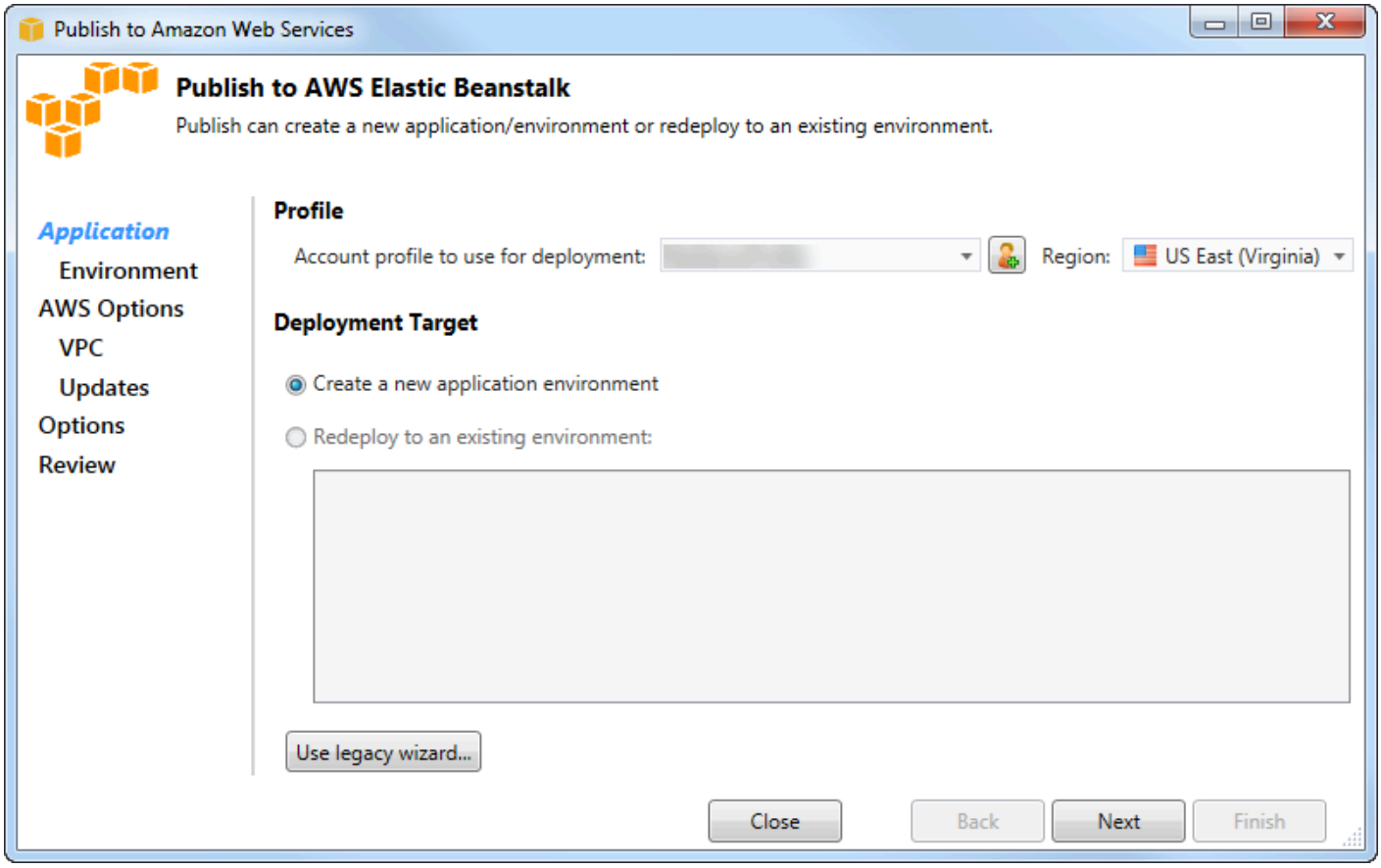


Publish to Elastic Beanstalk 마법사를 사용하여 애플리케이션을 배포하려면

1. 솔루션 탐색기에서 이전 섹션에서 생성한 프로젝트의 AEBWebAppDemo 프로젝트 폴더에 대한 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 열거나 자체 애플리케이션의 프로젝트 폴더에 대한 컨텍스트 메뉴를 열고 AWS Elastic Beanstalk에 게시를 선택합니다.



   Publish to Elastic Beanstalk(Elastic Beanstalk에 게시) 마법사가 표시됩니다.



2. 프로필의 배포에 사용할 계정 프로필 드롭다운 목록에서 배포에 사용할 AWS 계정 프로필을 선택합니다.

   선택적으로 사용하려는 AWS 계정이 있지만 아직 AWS 계정 프로필을 생성하지 않은 경우 더하기 기호(+)가 있는 버튼을 선택하여 AWS 계정 프로필을 추가할 수 있습니다.

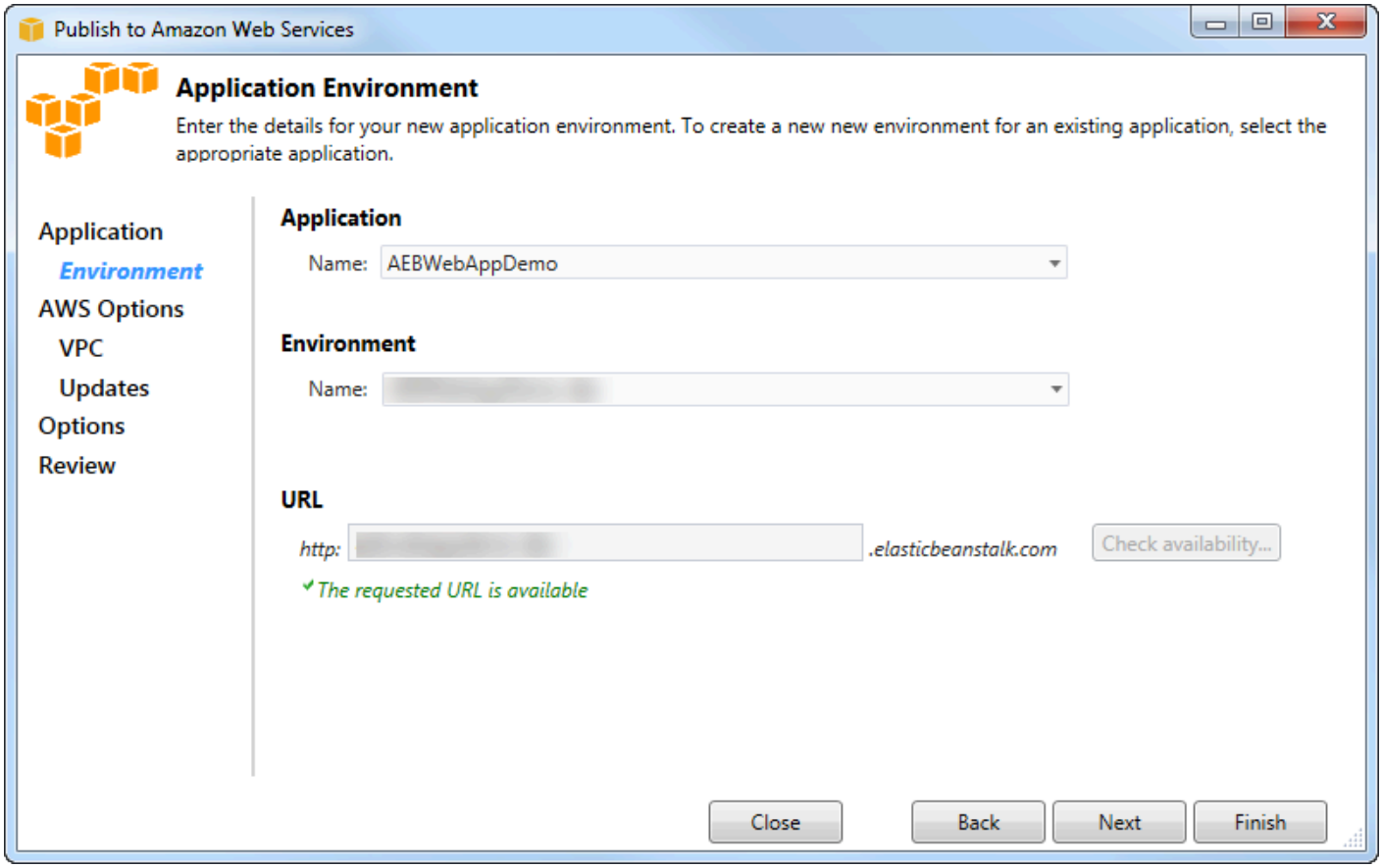
3. 리전 드롭다운 목록에서 Elastic Beanstalk를 통해 애플리케이션을 배포할 리전을 선택하세요.

4. 배포 대상에서 Create a new application environment(새 애플리케이션 환경 생성)를 선택하여 애플리케이션의 초기 배포를 수행하거나 Redeploy to an existing environment(기존 환경에 재배포)를 선택하여 이전에 배포된 애플리케이션을 다시 배포할 수 있습니다. (이전 배포는 마법사 또는 더 이상 사용되지 않는 독립 실행형 배포 도구를 사용하여 수행한 것일 수 있습니다.) Redeploy to an existing environment(기존 환경에 재배포)를 선택하면 마법사가 현재 실행 중인 이전 배포에서 정보를 검색하는 동안 지연 시간이 발생할 수 있습니다.

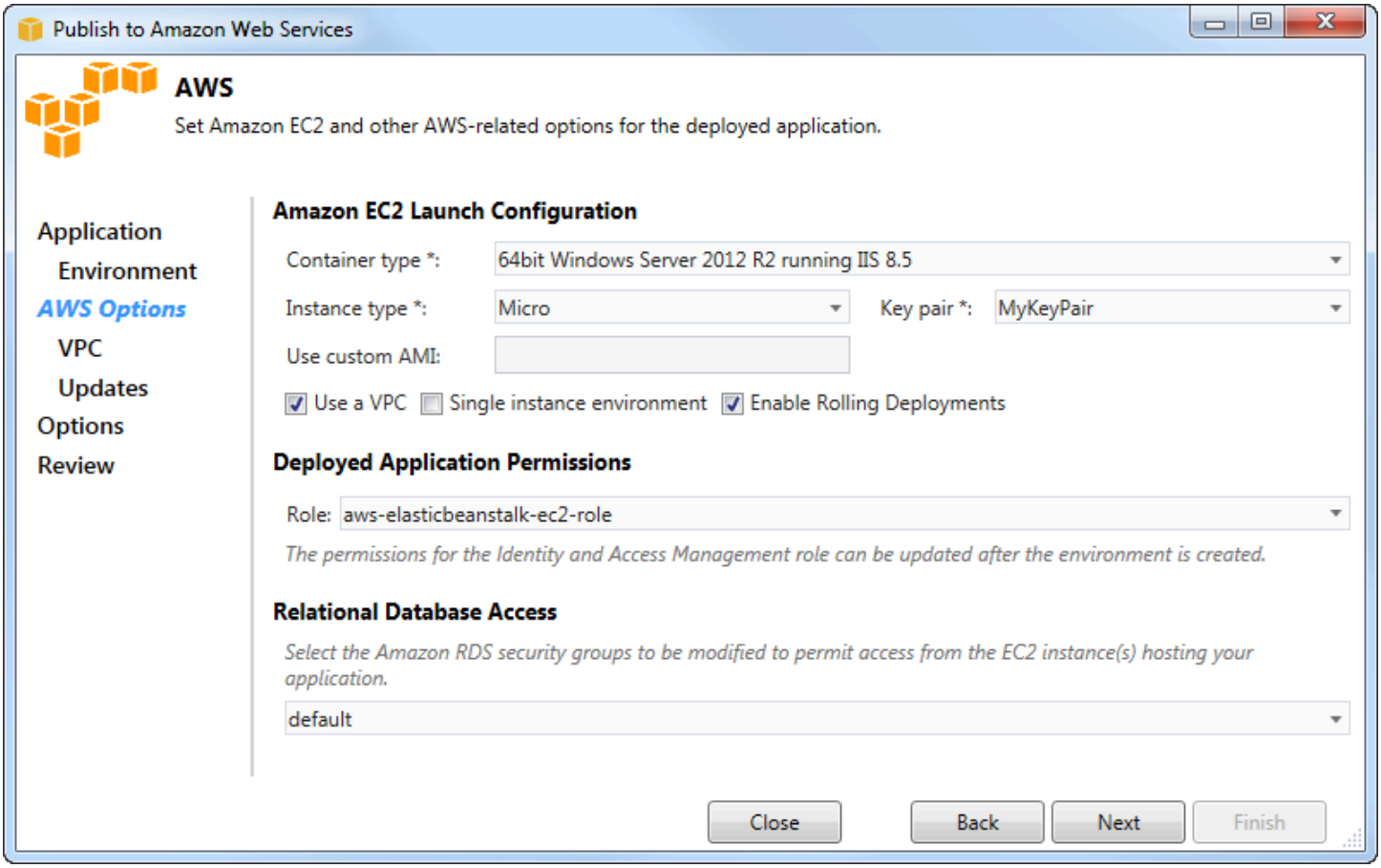
> **Note**
>
> Redeploy to an existing environment(기존 환경에 재배포)를 선택하고 목록에서 환경을 선택하고 다음을 선택하면 마법사를 통해 Application Options(애플리케이션 옵션)으로 곧바로

이동할 수 있습니다. 이 경로로 이동하려면 이 단원 후반부의 Application Options(애플리케이션 옵션) 사용 방법을 설명하는 지침으로 건너뛰십시오.

5. 다음을 선택합니다.



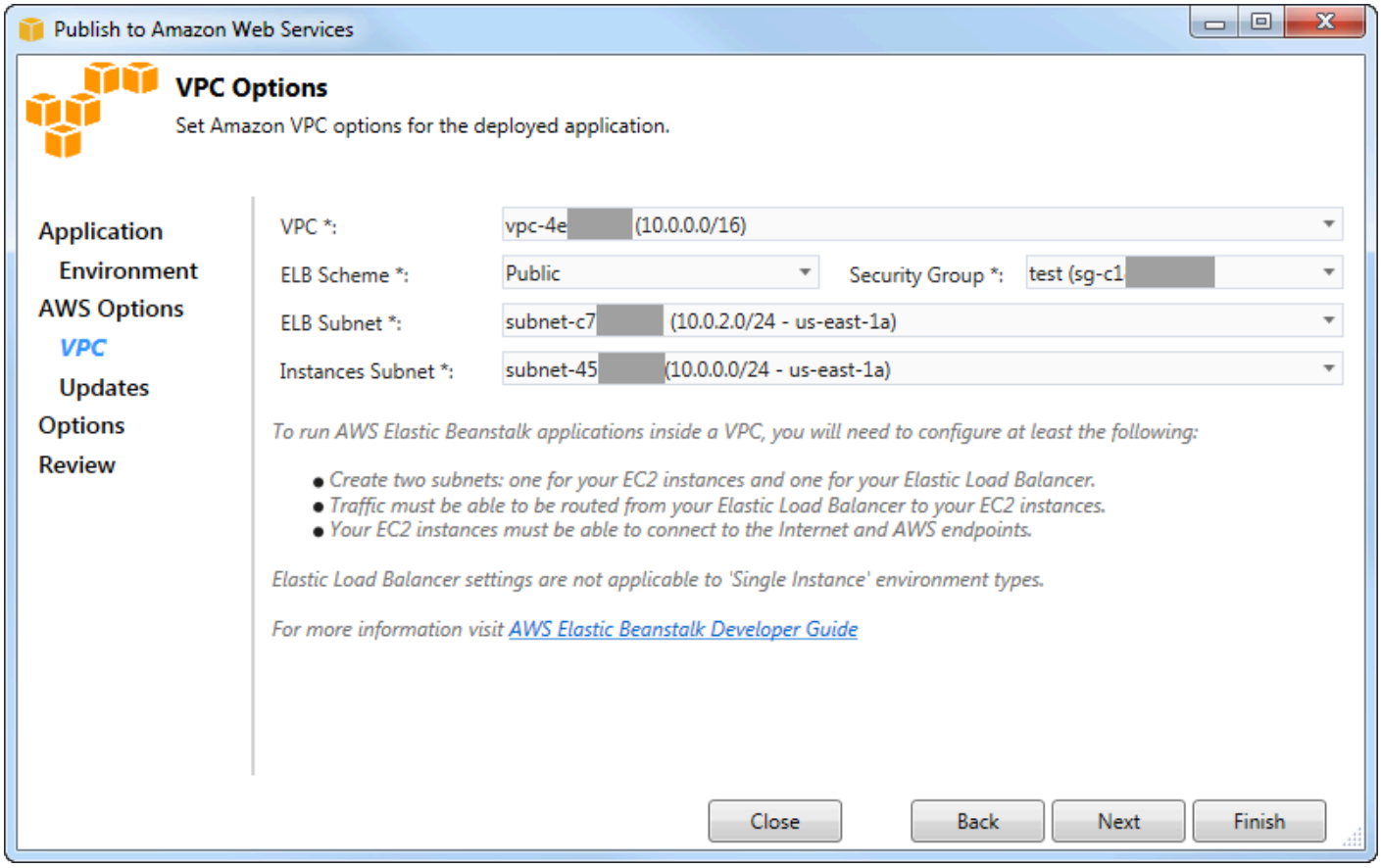
6. Application Environment(애플리케이션 환경) 페이지에서 애플리케이션 영역의 이름 드롭다운 목록에 애플리케이션의 기본 이름이 제안됩니다. 드롭다운 목록에서 다른 이름을 선택하여 기본 이름을 변경할 수 있습니다.
7. 환경 영역의 이름 드롭다운 목록에 Elastic Beanstalk 환경 이름을 입력하세요. 이 컨텍스트에서 환경은 애플리케이션의 인프라 Elastic Beanstalk 프로비저닝을 지칭합니다. 이 드롭다운 목록에 기본 이름이 이미 제안되어 있기도 합니다. 기본 이름이 제안되지 않았으면 이름을 입력하고, 드롭다운 목록에 추가 이름이 있으면 선택합니다. 환경 이름은 23자보다 길 수 없습니다.
8. URL 영역에서 웹 애플리케이션의 URL이 될 `.elasticbeanstalk.com`의 기본 하위 도메인이 상자에 제안됩니다. 새로운 하위 도메인 이름을 입력하여 기본 하위 도메인을 변경할 수 있습니다.
9. 웹 애플리케이션에 대한 URL이 이미 사용 중이 아닌지 확인하려면 가용성 확인을 선택합니다.
10. 웹 애플리케이션 URL이 사용할 수 있는 상태이면 다음을 선택합니다.



1. AWS 옵션 페이지에 있는 Amazon EC2 시작 구성의 컨테이너 유형 드롭다운 목록에서 애플리케이션에 사용할 Amazon Machine Image(AMI) 유형을 선택하세요.
2. 인스턴스 유형 드롭다운 목록에서 사용하려는 Amazon EC2 인스턴스 유형을 지정하세요. 이 예제에는 Micro(마이크로)를 사용하는 것이 좋습니다. 이렇게 하면 인스턴스 실행과 연관된 비용이 최소화됩니다. Amazon EC2 비용에 대한 자세한 정보는 EC2 요금 페이지를 참조하세요.
3. 키 페어 드롭다운 목록에서 애플리케이션에 사용할 인스턴스로 로그인하기 위한 Amazon EC2 인스턴스 키 페어를 선택하세요.
4. 선택적으로 Use custom AMI(사용자 지정 AMI 사용) 상자에서 Container type(컨테이너 유형) 드롭다운 목록에 지정된 AMI를 재정의할 사용자 지정 AMI를 지정할 수 있습니다. 사용자 지정 AMI를 생성하는 방법에 대한 자세한 정보는 AWS Elastic Beanstalk 개발자 안내서의 사용자 지정 AMI 사용 및 Amazon EC2 인스턴스에서 AMI 생성을 참조하세요.
5. 선택적으로 VPC에서 인스턴스를 시작하려면 Use a VPC(VPC 사용) 상자를 선택합니다.
6. 필요에 따라 단일 Amazon EC2 인스턴스를 시작하고 애플리케이션을 해당 인스턴스에 배포하려면 단일 인스턴스 환경 상자를 선택하세요.

이 상자를 선택하면 Elastic Beanstalk를 통해 오토 스케일링이 생성되지만 이를 구성하지는 않습니다. AWS Management Console을 사용하여 나중에 오토 스케일링을 구성할 수 있습니다.

7. 선택적으로 Enable Rolling Deployments(롤링 배포 활성화) 상자를 선택하여 애플리케이션이 인스턴스에 배포되는 조건을 제어할 수 있습니다. Single instance environment(단일 인스턴스 환경) 상자를 선택하지 않은 경우에만 이 상자를 선택할 수 있습니다.
8. 애플리케이션이 Amazon S3 및 DynamoDB와 같은 AWS 서비스를 사용하는 경우 자격 증명을 제공하는 가장 좋은 방법은 IAM 역할을 사용하는 것입니다. 배포된 애플리케이션 권한 영역에서 기존 IAM 역할을 선택하거나, 환경을 시작하기 위해 마법사에서 사용할 역할을 생성하세요. 를 사용하는 애플리케이션 AWS SDK for .NET 은 AWS 서비스에 요청할 때이 IAM 역할에서 제공하는 자격 증명을 자동으로 사용합니다.
9. 애플리케이션이 Amazon RDS 데이터베이스에 액세스하는 경우 관계형 데이터베이스 액세스 영역의 드롭다운 목록에서 마법사가 업데이트할 Amazon RDS 보안 그룹 옆의 상자를 선택하여 Amazon EC2 인스턴스가 데이터베이스에 액세스할 수 있도록 합니다.
10. 다음을 선택합니다.
    - Use a VPC(VPC 사용)를 선택하면 VPC Options(VPC 옵션) 페이지가 표시됩니다.
    - Use a VPC(VPC 사용)를 선택하지 않고 Enable Rolling Deployments(롤링 배포 활성화)를 선택하면 Rolling Deployments(롤링 배포) 페이지가 표시됩니다. 이 단원 후반부의 Rolling Deployments(롤링 배포) 사용 방법을 설명하는 지침으로 건너뛰십시오.
    - Use a VPC(VPC 사용) 또는 Enable Rolling Deployments(롤링 배포 활성화)를 선택하지 않은 경우 Application Options(애플리케이션 옵션) 페이지가 표시됩니다. 이 단원 후반부의 Application Options(애플리케이션 옵션) 사용 방법을 설명하는 지침으로 건너뛰십시오.
11. Use a VPC(VPC 사용)를 선택한 경우 VPC Options(VPC 옵션) 페이지에서 정보를 지정하여 VPC로 애플리케이션을 시작합니다.



VPC가 이미 생성되어 있어야 합니다. Toolkit for Visual Studio에서 VPC를 생성한 경우 Toolkit for Visual Studio가 이 페이지를 채웁니다. AWS 관리 콘솔에서 VPC를 생성한 경우 이 페이지에 VPC에 대한 정보를 입력하세요.

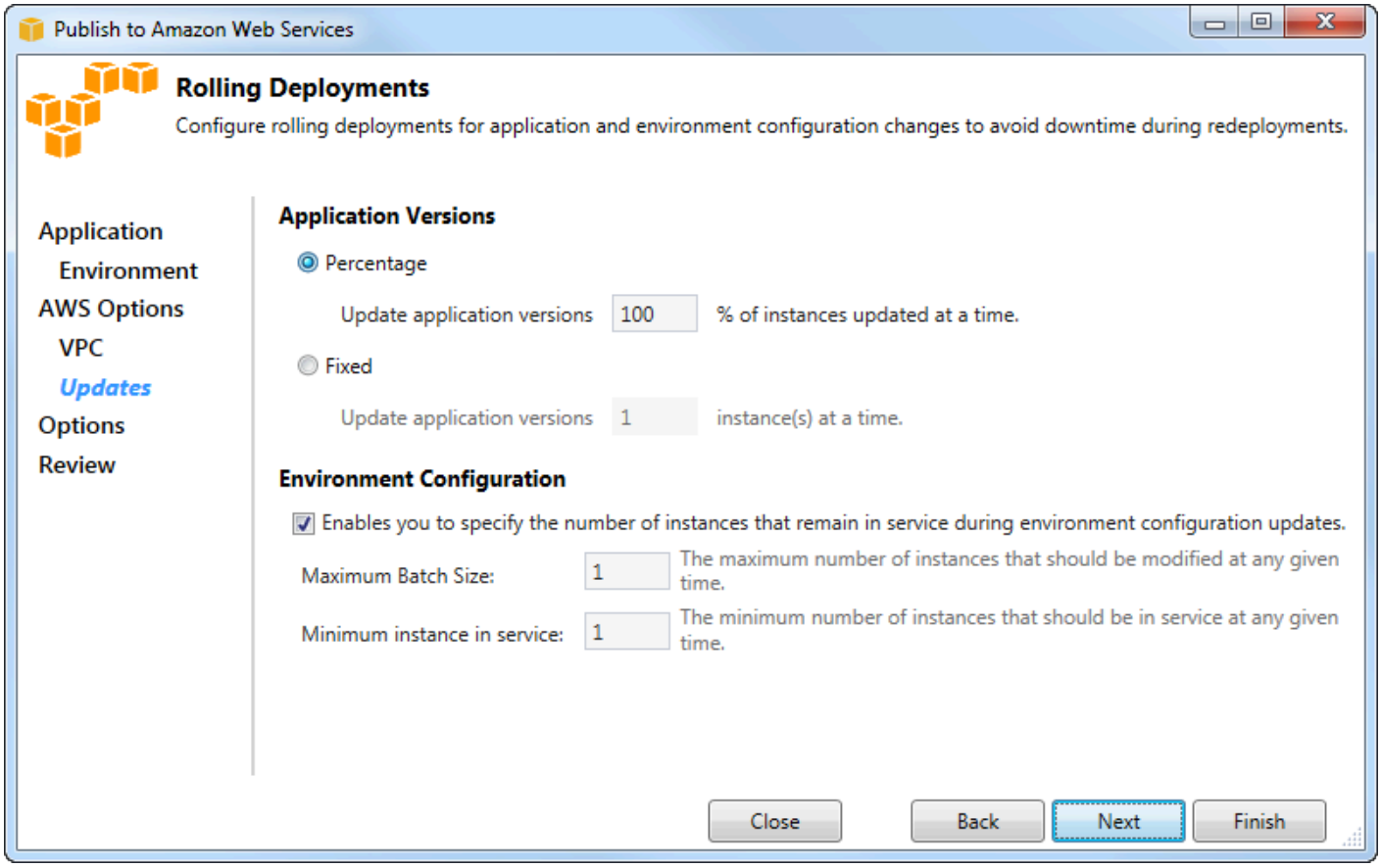
VPC에 배포하기 위한 주요 고려 사항

- VPC에는 퍼블릭 서브넷 및 프라이빗 서브넷이 최소한 한 개가 있어야 합니다.
- ELB Subnet(ELB 서브넷) 드롭다운 목록에서 퍼블릭 서브넷을 지정합니다. Toolkit for Visual Studio는 애플리케이션을 위한 Elastic Load Balancing 로드 밸런서를 퍼블릭 서브넷에 배포합니다. 퍼블릭 서브넷은 인터넷 게이트웨이를 가리키는 항목이 있는 라우팅 테이블과 연결됩니다. 인터넷 게이트웨이에는 `igw-`로 시작하는 ID(예: `igw-83cddaex`)가 있으므로 알아볼 수 있습니다. Toolkit for Visual Studio를 사용하여 생성한 퍼블릭 서브넷에는 해당 서브넷을 퍼블릭으로 식별하는 태그 값이 있습니다.
- Instances Subnet(인스턴스 서브넷) 드롭다운 목록에서 프라이빗 서브넷을 지정합니다. Toolkit for Visual Studio는 애플리케이션을 위해 프라이빗 서브넷에 Amazon EC2 인스턴스를 배포합니다.

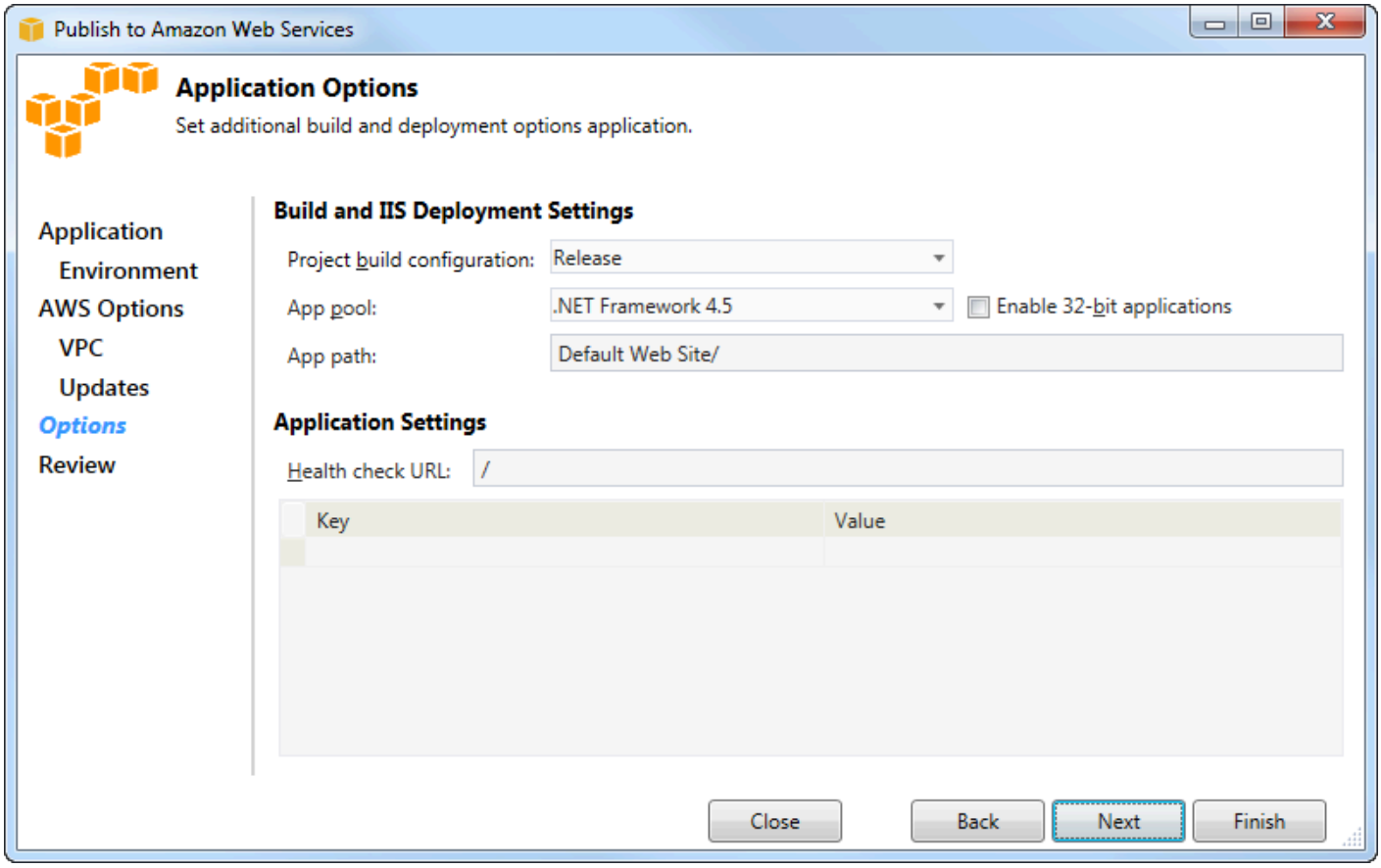
- 애플리케이션을 위한 Amazon EC2 인스턴스는 Network Address Translation(NAT)을 수행하는 퍼블릭 서브넷의 Amazon EC2 인스턴스를 통해 프라이빗 서브넷에서 인터넷으로 통신합니다. 이러한 통신을 사용하려면 프라이빗 서브넷에서 NAT 인스턴스로 트래픽의 흐름을 허용하는 VPC 보안 그룹이 필요합니다. 보안 그룹 드롭다운 목록에서 이 VPC 보안 그룹을 지정합니다.

Elastic Beanstalk 애플리케이션을 VPC에 배포하는 방법에 대한 자세한 정보는 AWS Elastic Beanstalk 개발자 안내서를 참조하세요.

1. VPC Options(VPC 옵션) 페이지에 있는 정보를 모두 채우고 다음을 선택합니다.
   - Enable Rolling Deployments(롤링 배포 활성화)를 선택하면 Rolling Deployments(롤링 배포) 페이지가 표시됩니다.
   - Enable Rolling Deployments(롤링 배포 활성화)를 선택하지 않은 경우 Application Options(애플리케이션 옵션) 페이지가 표시됩니다. 이 단원 후반부의 Application Options(애플리케이션 옵션) 사용 방법을 설명하는 지침으로 건너뛰십시오.
2. Enable Rolling Deployments(롤링 배포 활성화)를 선택한 경우 애플리케이션의 새 버전을 로드 밸런싱 환경의 인스턴스에 배포하는 방법을 구성하기 위해 Rolling Deployments(롤링 배포) 페이지에서 정보를 지정합니다. 예를 들어, 환경에 인스턴스 4개가 있고 인스턴스 유형을 변경하려는 경우 한 번에 인스턴스 2개를 변경할 수 있도록 환경을 구성할 수 있습니다. 그러면 변경하는 동안에도 애플리케이션이 실행됩니다.



3. 애플리케이션 버전 영역에서 백분율 또는 인스턴스 수로 배포를 한 번에 제어하는 옵션을 선택합니다. 원하는 백분율이나 수를 지정합니다.
4. 배포 중에 계속 서비스할 인스턴스 수를 지정하려는 경우 환경 구성 영역에서 상자를 선택합니다. 이 상자를 선택할 경우 한 번에 수정할 최대 인스턴스의 수와 한 번에 계속 서비스할 최소 인스턴스 수를 하나만 지정하거나 둘 다 지정합니다.
5. 다음을 선택합니다.
6. Application Options(애플리케이션 옵션) 페이지에서 빌드, IIS(인터넷 정보 서비스) 및 애플리케이션 설정에 대한 정보를 지정합니다.

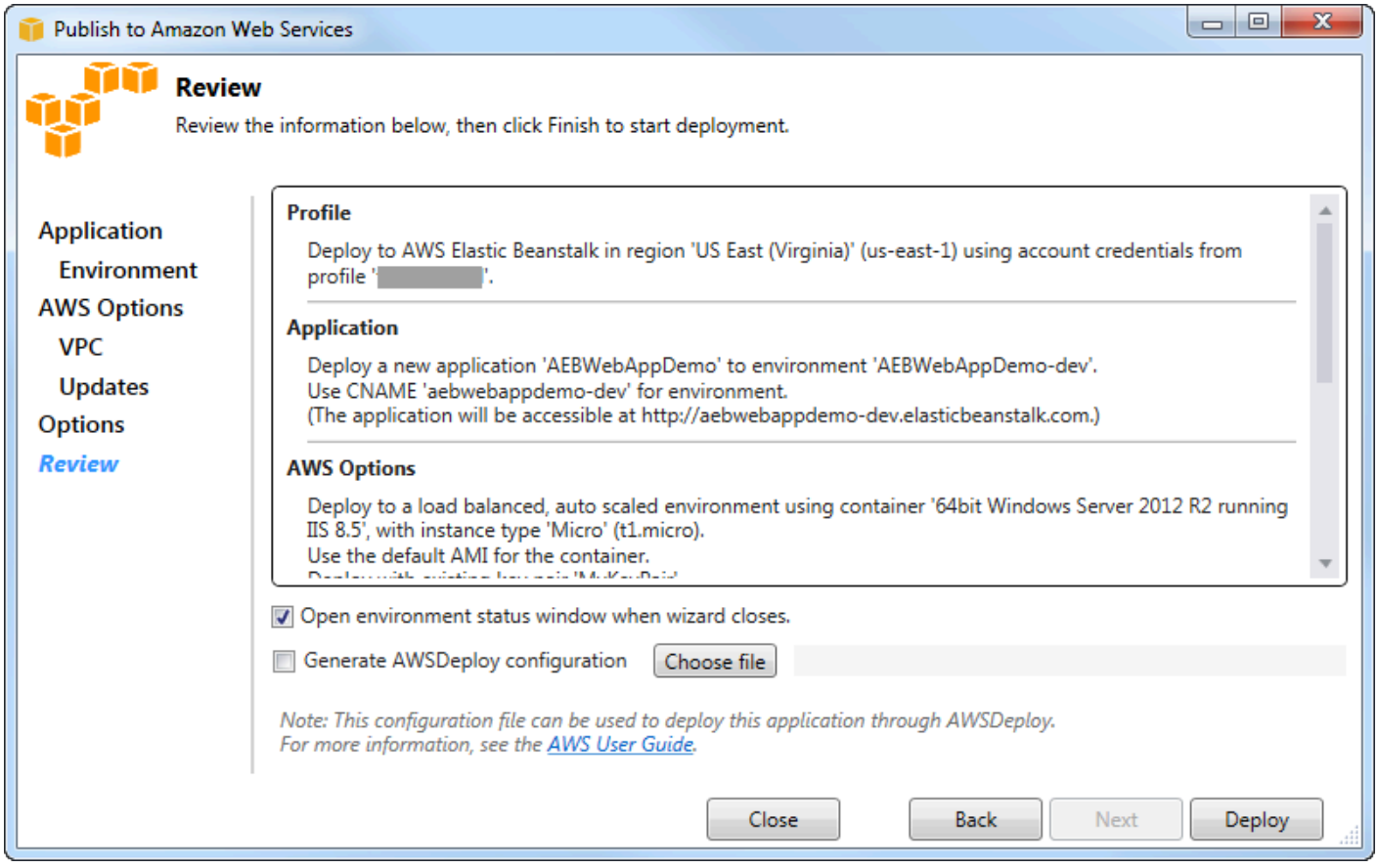


7. Build and IIS Deployment Settings(빌드 및 IIS 배포 설정) 영역의 Project build configuration(프로젝트 빌드 구성) 드롭다운 목록에서 대상 빌드 구성을 선택합니다. 마법사가 찾으면 릴리스가 표시되고 그렇지 않으면 활성화된 구성이 이 상자에 표시됩니다.
8. App pool(앱 풀) 드롭다운 목록에서 애플리케이션에 필요한 .NET Framework 버전을 선택합니다. 올바른 .NET Framework 버전이 이미 표시되어 있어야 합니다.
9. 애플리케이션이 32바이트인 경우 Enable 32-bit applications(32비트 애플리케이션 활성화) 상자를 선택합니다.
10. App path(앱 경로) 상자에서 IIS가 애플리케이션을 배포하는 데 사용할 경로를 지정합니다. 기본적으로 Default Web Site/(기본 웹 사이트/)가 지정되며 일반적으로 `c:\inetpub\wwwroot` 경로로 변환됩니다. Default Web Site/(기본 웹 사이트/) 이외의 경로를 지정하면 마법사는 지정된 경로를 가리키는 Default Web Site/(기본 웹 사이트/) 경로에 리디렉션을 배치합니다.
11. 애플리케이션 설정 영역의 상태 확인 URL 상자에 Elastic Beanstalk URL을 입력하여 웹 애플리케이션이 계속 응답하는지 확인합니다. 이 URL은 루트 서버 URL에 상대적입니다. 루트 서버 URL이 기본적으로 지정됩니다. 예를 들어, 전체 URL이 `example.com/site-is-up.html`이면 `/site-is-up.html`을 입력합니다.

12.키 및 값 영역에서 애플리케이션의 `Web.config` 파일에 추가할 키 및 값 페어를 지정할 수 있습니다.

> **Note**
>
> 권장되지는 않지만 키 및 값의 영역을 사용하여 애플리케이션이 실행되어야 하는 AWS 자격 증명을 지정할 수 있습니다. 기본 접근 방식은 AWS 옵션 페이지의 Identity and Access Management 역할 드롭다운 목록에서 IAM 역할을 지정하는 것입니다. 그러나 애플리케이션을 실행하기 위해 IAM 역할 대신 AWS 자격 증명을 사용해야 하는 경우 키 행에서 AWSAccessKey를 선택합니다. 값 열에 액세스 키를 입력합니다. AWSSecretKey에 대해 이러한 단계를 반복합니다.

13.다음을 선택합니다.



14.검토 페이지에서 구성한 옵션을 검토하고 Open environment status window when wizard closes(마법사를 닫을 때 환경 상태 창 열기) 상자를 선택합니다.

15.모든 사항이 올바르게 보이는 경우 배포를 선택합니다.

> **Note**
>
> 애플리케이션을 배포하면 애플리케이션에서 사용한 AWS 리소스에 대해 활성 계정에 요금이 발생합니다.

배포에 대한 정보는 Visual Studio 상태 표시줄 및 결과 창에 표시됩니다. 몇 분 정도 걸릴 수 있습니다. 배포가 완료되면 결과 창에 확인 메시지가 표시됩니다.

16. 배포를 삭제하려면 AWS 탐색기에서 Elastic Beanstalk 노드를 확장하고 배포의 하위 노드에 대한 컨텍스트(마우스 오른쪽 버튼 클릭) 메뉴를 연 다음 삭제를 선택합니다. 삭제 프로세스는 몇 분 정도 걸릴 수 있습니다.

Elastic Beanstalk에 ASP.NET Core 애플리케이션 배포(레거시)

> **Important**
>
> 이 설명서에서는 레거시 서비스 및 기능을 참조합니다. 업데이트된 안내서와 정보는 AWS .NET 배포 도구 안내서 및 업데이트된 AWS목차 배포를 참조하세요.

AWS Elastic Beanstalk 는 애플리케이션의 AWS 리소스 프로비저닝 프로세스를 간소화하는 서비스입니다.는 애플리케이션을 배포하는 데 필요한 모든 AWS 인프라를 AWS Elastic Beanstalk 제공합니다.

Toolkit for Visual Studio는 Elastic Beanstalk를 AWS 사용하여에 ASP.NET Core 애플리케이션 배포를 지원합니다. ASP.NET Core는 클라우드에서 실행되도록 종속성 오버헤드를 최소화하고 애플리케이션을 간소화한 모듈화된 아키텍처가 있는 ASP.NET으로 재설계되었습니다.

AWS Elastic Beanstalk 를 사용하면 다양한 언어로 애플리케이션을 쉽게 배포할 수 있습니다 AWS. Elastic Beanstalk는 기존 ASP.NET 애플리케이션과 ASP.NET Core 애플리케이션을 모두 지원합니다. 이 주제에서는 ASP.NET Core 애플리케이션의 배포에 대해 설명합니다.

배포 마법사 사용

ASP.NET Core 애플리케이션을 Elastic Beanstalk에 배포하는 가장 쉬운 방법은 Toolkit for Visual Studio를 사용하는 것입니다.

기존 ASP.NET 애플리케이션을 배포하기 위해 도구 키트를 사용한 적이 있는 경우 ASP.NET Core에 대한 환경이 매우 비슷함을 알 수 있습니다. 아래의 단계에서는 배포 환경에 대해 알아봅니다.

이전에 도구 키트를 사용한 적이 없는 경우 도구 키트를 설치한 후 가장 먼저 해야 할 일은 도구 키트에 자격 AWS 증명을 등록하는 것입니다. 자세한 내용은 Application for Visual Studio 설명서의 AWS 보안 자격 증명을 지정하는 방법을 참조하세요.

ASP.NET Core 웹 애플리케이션을 배포하려면 솔루션 탐색기에서 프로젝트를 마우스 오른쪽 버튼으로 클릭하고 게시 대상 AWS...을 선택합니다.

AWS Elastic Beanstalk 배포에 게시 마법사의 첫 번째 페이지에서 새 Elastic Beanstalk 애플리케이션을 생성하도록 선택합니다. Elastic Beanstalk 애플리케이션은 환경, 버전 및 환경 구성을 포함한 Elastic Beanstalk 구성 요소의 논리적 컬렉션입니다. 배포 마법사는 애플리케이션 버전과 환경을 포함하는 애플리케이션을 생성합니다. 환경에는 애플리케이션 버전을 실행하는 실제 AWS 리소스가 포함되어 있습니다. 애플리케이션을 배포할 때마다 새 애플리케이션 버전이 생성되며 마법사는 해당 버전에 대한 환경을 가리킵니다. Elastic Beanstalk 구성 요소에서 이러한 개념에 대해 자세히 알아볼 수 있습니다.

다음으로 애플리케이션의 이름과 첫 번째 환경을 설정합니다. 각 환경에는 배포가 완료될 때 애플리케이션에 액세스하는 데 사용할 수 있는 연관된 고유 CNAME이 있습니다.

다음 페이지인 AWS 옵션에서 사용할 AWS 리소스 유형을 구성할 수 있습니다. 이 예제의 경우 키 페어 섹션을 제외하고 기본값을 그대로 둡니다. 키 페어를 사용하면 Windows 관리자 암호를 검색할 수 있으므로 시스템에 로그인할 수 있습니다. 키 페어를 아직 생성하지 않은 경우 Create new key pair(새 키 페어 생성)를 선택할 수 있습니다.

권한

권한 페이지는 애플리케이션을 실행하는 EC2 인스턴스에 AWS 자격 증명을 할당하는 데 사용됩니다. 이는 애플리케이션이 AWS SDK for .NET 를 사용하여 다른 AWS 서비스에 액세스하는 경우 중요합니다. 애플리케이션에서 다른 서비스를 사용하지 않는 경우 이 페이지를 기본값으로 그대로 둘 수 있습니다.

애플리케이션 옵션

Application Options(애플리케이션 옵션) 페이지의 세부 사항은 기존 ASP.NET 애플리케이션을 배포할 때 지정한 사항과 다릅니다. 여기에서 애플리케이션을 패키징하는 데 사용된 빌드 구성과 프레임워크를 지정하고, 애플리케이션에 대한 IIS 리소스 경로도 지정합니다.

Application Options(애플리케이션 옵션) 페이지를 완료한 후 다음을 클릭하여 설정을 검토한 다음 배포를 클릭하여 배포 프로세스를 시작합니다.

환경 상태 확인

애플리케이션을 패키징하고에 업로드한 후 Visual Studio의 AWS Explorer에서 환경 상태 보기를 열어 Elastic Beanstalk 환경의 상태를 확인할 AWS수 있습니다.

환경이 온라인 상태가 되면 상태 표시줄에 이벤트가 표시됩니다. 모든 사항이 완료되면 환경 상태는 정상 상태로 이동합니다. URL을 클릭하여 사이트를 볼 수 있습니다. 여기에서 환경 또는 원격 데스크톱의 로그를 Elastic Beanstalk 환경의 일부인 Amazon EC2 인스턴스로 가져올 수도 있습니다.

애플리케이션의 첫 번째 배포는 새 AWS 리소스를 생성하므로 후속 재배포보다 시간이 약간 오래 걸립니다. 개발 중 애플리케이션을 반복할 때 마법사를 통해 돌아가거나 프로젝트를 오른쪽 버튼으로 클릭할 때 재게시 옵션을 선택하여 신속하게 재배포할 수 있습니다.

배포 마법사를 통해 이전 실행의 설정을 사용하여 애플리케이션에 대한 패키지를 재게시하고 애플리케이션 번들을 기존 Elastic Beanstalk 환경에 업로드합니다.

애플리케이션의 AWS 보안 자격 증명을 지정하는 방법

Elastic Beanstalk에 게시 마법사에서 지정하는 AWS 계정은 마법사가 Elastic Beanstalk에 배포하는 데 사용할 AWS 계정입니다.

권장되지는 않지만 애플리케이션이 배포된 후 AWS 서비스에 액세스하는 데 사용할 AWS 계정 자격 증명을 지정해야 할 수도 있습니다. 선호되는 방법은 IAM 역할을 지정하는 것입니다. Elastic Beanstalk에 게시 마법사에서 AWS 옵션 페이지의 Identity and Access Management 역할 드롭다운 목록을 통해 이 작업을 수행합니다. 기존 Amazon Web Services에 게시 마법사에서는 AWS 옵션 페이지의 IAM 역할 드롭다운 목록을 통해 이 작업을 수행할 수 있습니다.

IAM 역할 대신 AWS 계정 자격 증명을 사용해야 하는 경우 다음 방법 중 하나로 애플리케이션의 계정 자격 증명을 지정할 AWS 수 있습니다.

- 프로젝트 `Web.config` 파일의 `appSettings` 요소에 있는 AWS 계정 자격 증명에 해당하는 프로필을 참조합니다. (프로파일을 생성하려면 AWS 자격 증명 구성을 참조하세요.) 다음 예제에서는 프로파일 이름이 `myProfile`인 자격 증명을 지정합니다.

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

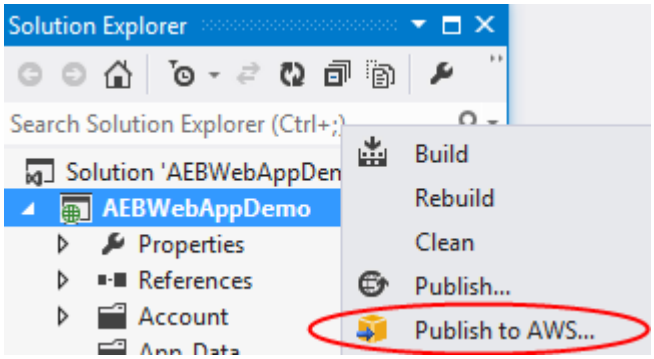
- Elastic Beanstalk에 게시 마법사를 사용하는 경우 애플리케이션 옵션 페이지의 키 및 값 영역의 키행에서 AWS AccessKey를 선택하세요. 값 열에 액세스 키를 입력합니다. AWS SecretKey에 대해 이러한 단계를 반복하세요.
- 레거시 Amazon Web Services에 게시 마법사를 사용하는 경우 애플리케이션 옵션 페이지의 애플리케이션 자격 증명 영역에서 이 자격 증명 사용을 선택한 다음 액세스 키 및 시크릿 액세스 키 상자에 액세스 키와 시크릿 액세스 키를 입력합니다.

애플리케이션을 Elastic Beanstalk 환경에 재게시하는 방법(레거시)

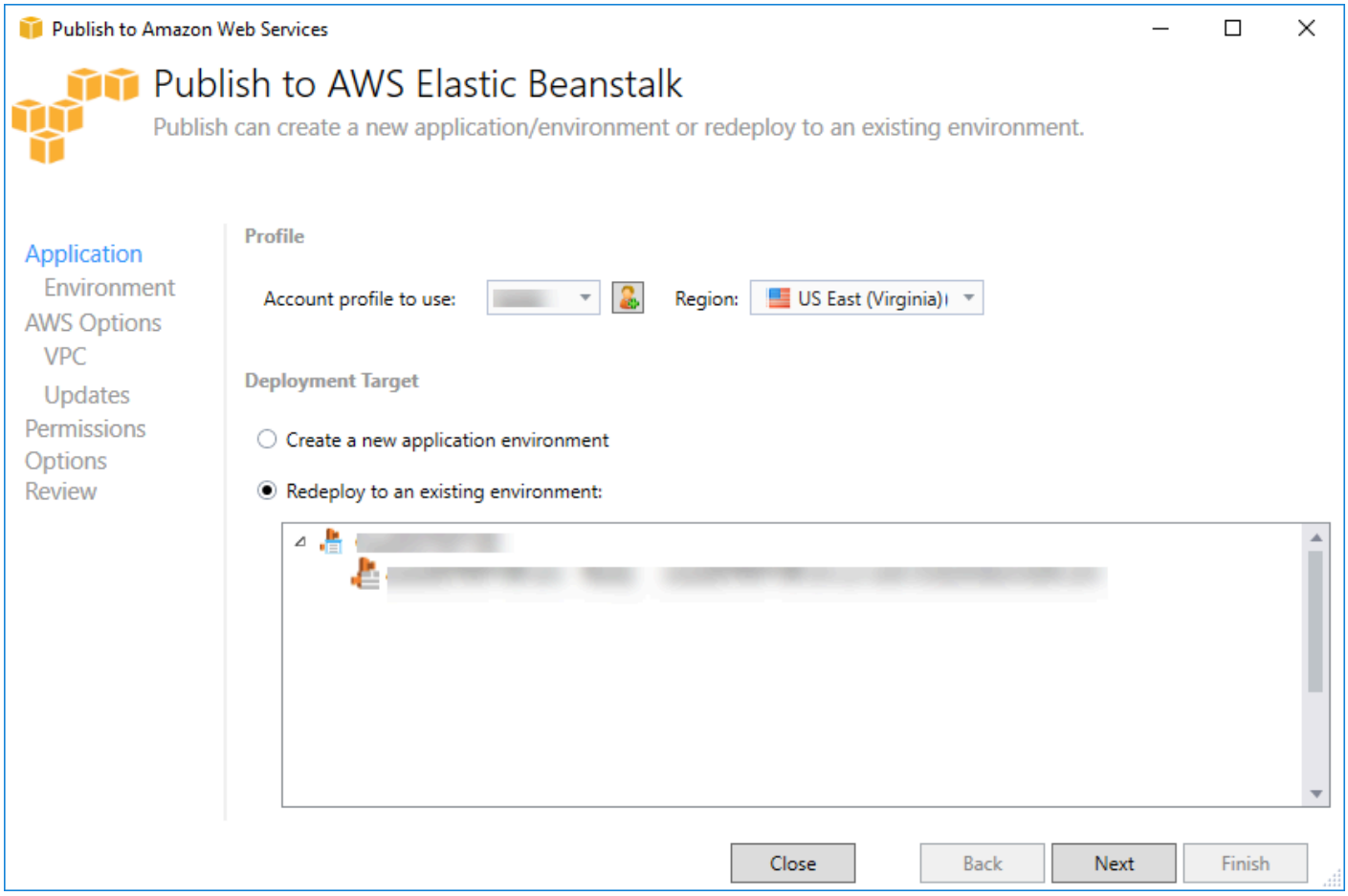
> **Important**
>
> 이 설명서에서는 레거시 서비스 및 기능을 참조합니다. 업데이트된 가이드 및 내용은 AWS .NET 배포 도구 가이드를 참조하세요.

별도의 변경을 수행한 다음 이미 시작된 Elastic Beanstalk 환경에 새 버전을 재게시하여 애플리케이션을 반복할 수 있습니다.

1. 솔루션 탐색기에서, 이전 섹션에서 게시한 프로젝트의 AEBWebAppDemo 프로젝트 폴더의 컨텍스트 메뉴(마운스 오른쪽 버튼 클릭)를 열고 AWS Elastic Beanstalk에 게시를 선택하세요다.

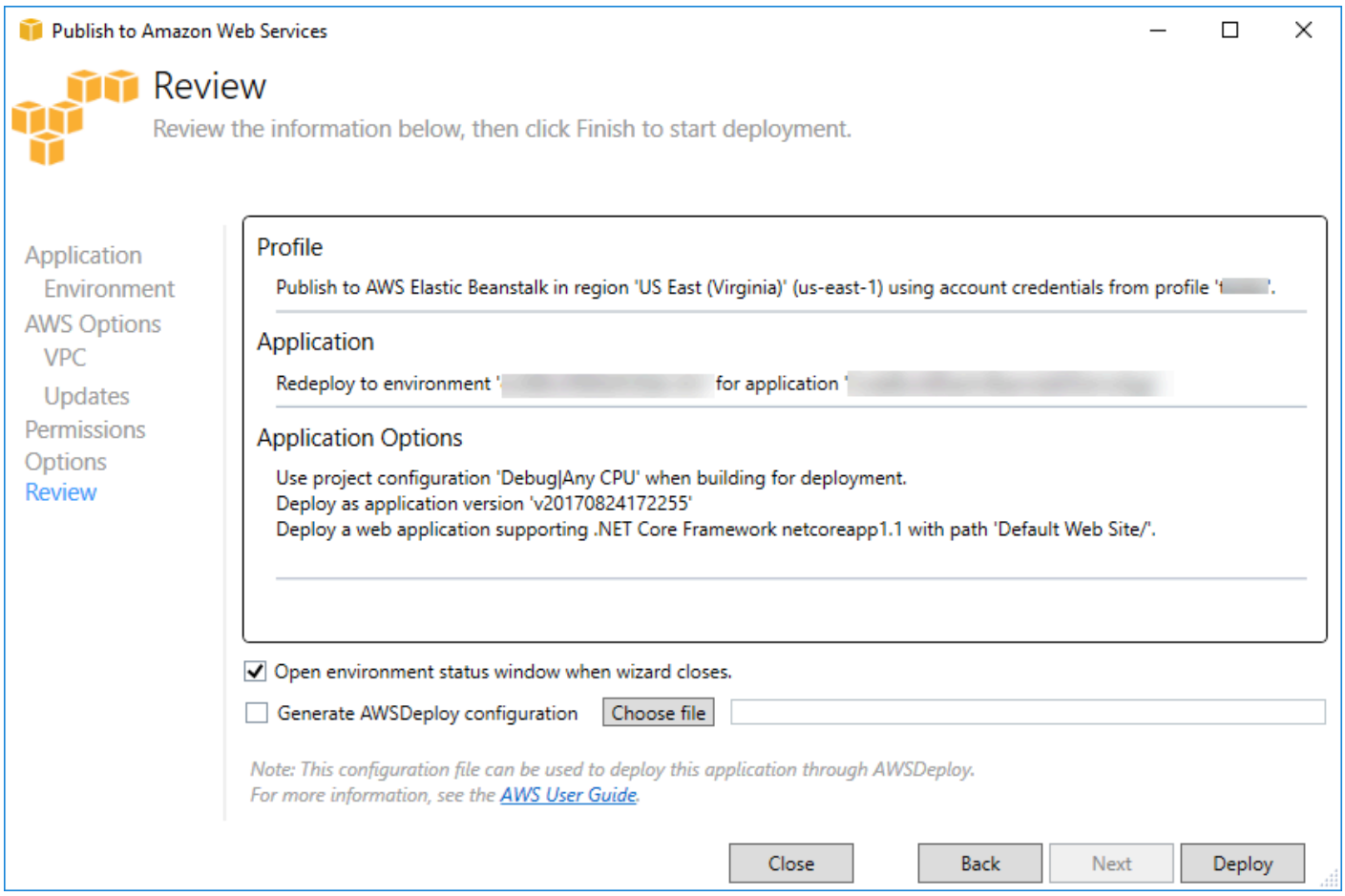


   Publish to Elastic Beanstalk(Elastic Beanstalk에 게시) 마법사가 표시됩니다.



2. Redeploy to an existing environment(기존 환경에 재배포)를 선택하고 이전에 프로젝트를 게시한 환경을 선택합니다. 다음을 클릭합니다.

   검토 마법사가 나타납니다.



3. 배포를 클릭합니다. 애플리케이션이 동일한 환경에 다시 배포됩니다.

애플리케이션이 시작 또는 종료되는 중일 경우 재게시할 수 없습니다.

사용자 지정 Elastic Beanstalk 애플리케이션 배포

이 주제에서는 Elastic Beanstalk의 Microsoft Windows 컨테이너의 배포 매니페스트가 사용자 지정 애플리케이션 배포를 지원하는 방법을 설명합니다.

사용자 지정 애플리케이션 배포는 AWS 리소스를 생성하고 관리하면서 애플리케이션이 배포되는 방식도 완벽하게 제어할 수 있는 Elastic Beanstalk를 원하는 고급 사용자를 위한 강력한 기능입니다. 사용자 지정 애플리케이션 배포의 경우 세 가지 Elastic Beanstalk 작업을 수행하는 Windows PowerShell 스크립트를 만듭니다. 설치 작업은 배포가 시작될 때 사용되고 다시 시작은 도구 키트나 웹 콘솔에서 `RestartAppServer` API가 호출될 때 사용하며 제거는 새 배포가 발생할 때마다 이전 배포에서 호출됩니다.

예를 들어, 문서 팀에서 배포에 포함하려는 정적 웹 사이트를 작성하는 동안 배포하려는 ASP.NET 애플리케이션이 있을 수 있습니다. 다음과 같이 배포 매니페스트를 작성하여 이 작업을 수행할 수 있습니다.

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

각 작업에 대해 나열된 스크립트는 배포 매니페스트 파일과 관련된 애플리케이션 번들에 있어야 합니다. 이 예에서는 애플리케이션 번들에 문서 팀에서 만든 정적 웹 사이트가 있는 documentation.zip 파일도 포함됩니다.

`install.ps1` 스크립트가 zip 파일 압축을 풀고 IIS 경로를 설정합니다.

```
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}
```

애플리케이션이 IIS에서 실행 중이므로 다시 시작 작업이 IIS 재설정을 호출합니다.

```
iisreset /timeout:1
```

제거 스크립트의 경우 설치 단계에서 사용된 모든 설정과 파일을 정리해야 합니다. 이렇게 하면 새 버전 설치 단계 중에 이전 배포와의 충돌을 피할 수 있습니다. 이 예에서는 정적 웹 사이트의 IIS 애플리케이션과 웹 사이트 파일을 제거해야 합니다.

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

이러한 스크립트 파일과 애플리케이션 번들에 포함된 documentation.zip 파일을 사용하면 배포 시 ASP.NET 애플리케이션이 생성되고 문서 사이트가 배포됩니다.

이 예제에서는 단순한 정적 웹 사이트를 배포하는 간단한 예제를 선택하지만 사용자 지정 애플리케이션 배포를 통해 모든 유형의 애플리케이션을 배포하고 Elastic Beanstalk에서 AWS 리소스를 관리할 수 있도록 합니다.

사용자 지정 ASP.NET Core Elastic Beanstalk 배포

이 주제에서는 Elastic Beanstalk 및 Toolkit for Visual Studio로 ASP.NET Core 애플리케이션을 생성할 때 배포가 작동하는 방법과 배포를 사용자 지정할 수 있는 작업을 설명합니다.

Toolkit for Visual Studio에서 배포 마법사를 완료하면 툴킷이 애플리케이션을 번들링하여 Elastic Beanstalk로 전송합니다. 애플리케이션 번들을 생성하는 첫 번째 단계는 publish 명령을 사용하여 애플리케이션의 게시를 준비하기 위해 새 dotnet CLI를 사용하는 것입니다. 프레임워크 및 구성은 마법사의 설정에서 publish 명령으로 전달됩니다. 따라서 `configuration`에 대한 릴리즈 및 `framework`에 대한 netcoreapp1.0을 선택한 경우 툴킷은 다음 명령을 실행합니다.

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

publish 명령이 끝나면 도구 키트가 게시 폴더에 새로운 배포 매니페스트를 씁니다. 배포 매니페스트는 aws-windows-deployment-manifest.json이라는 JSON 파일이며, Elastic Beanstalk Windows 컨테이너

(버전 1.2 이상)가 이 파일을 읽어 애플이케이션을 배포하는 방법을 결정합니다. 예를 들어, IIS 루트에 배포할 ASP.NET Core 애플리케이션의 경우 도구 키트가 다음과 같은 미니페스트 파일을 생성합니다.

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

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

appBundle 속성은 애플리케이션 비트가 매니페스트 파일에 관련된 위치를 나타냅니다. 이 속성은 디렉터리 또는 ZIP 아카이브를 가리킬 수 있습니다. iisPath 및 iisWebSite 속성은 IIS에서 애플리케이션을 호스팅할 위치를 나타냅니다.

매니페스트 사용자 지정

게시 폴더에 매니페스트 파일이 없는 경우에만 도구 키트가 이 파일을 씁니다. 파일이 존재하면 도구 키트가 매니페스트의 appBundle 섹션 아래에 나열된 첫 번째 애플리케이션에서 iisPath, iisWebSite 및 aspNetCoreWeb 속성을 업데이트합니다. 그러면 aws-windows-deployment-manifest.json을 프로젝트에 추가하고 매니페스트를 사용자 지정할 수 있습니다. Visual Studio에서 ASP.NET Core 웹 애플리케이션에 대해 이 작업을 수행하려면 프로젝트 루트에 새 JSON 파일을 추가하고 이름을 aws-windows-deployment-manifest.json으로 지정합니다.

매니페스트는 이름이 aws-windows-deployment-manifest.json이고 프로젝트 루트에 있어야 합니다. Elastic Beanstalk 컨테이너가 루트에서 매니페스트를 찾고 발견하면 배포 도구를 호출합니다. 파일이 없으면 Elastic Beanstalk 컨테이너는 아카이브가 msdeploy 아카이브라고 가정하는 이전 배포 도구로 폴백됩니다.

dotnet CLI publish 명령에 매니페스트를 포함하려면 project.json의 include 아래 include 섹션에 매니페스트 파일을 포함하도록 publishOptions 파일을 업데이트하십시오.

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

앱 번들에 포함되도록 선언했기 때문에 애플리케이션 배포 방법을 추가로 구성할 수 있습니다. 배포 마법사가 지원하는 것 이상으로 배포를 사용자 지정할 수 있습니다. AWSaws-windows-deployment-manifest.json 파일에 대한 JSON 스키마를 정의했으며, Toolkit for Visual Studio를 설치할 때 설치 프로그램에서 스키마의 URL을 등록했습니다.

`windows-deployment-manifest.json`을 열면 [Schema] 드롭다운 상자에 선택한 스키마 URL이 표시됩니다. URL로 이동하여 매니페스트에 설정할 수 있는 항목의 전체 설명을 볼 수 있습니다. 스키마를 선택하면 매니페스트를 편집하는 동안 Visual Studio가 IntelliSense를 제공합니다.

지원되는 사용자 지정에는 애플리케이션을 실행할 IIS 애플리케이션 풀을 구성하는 기능이 있습니다. 다음 예제에서는 60분마다 프로세스를 재생하는 IIS 애플리케이션 풀("customPool")을 정의하고 `"appPool": "customPool"`을 사용하여 애플리케이션에 할당하는 방법을 보여줍니다.

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

```
        "parameters": {
          "appPool": "customPool"
        }
      }
    ]
  }
}
```

또한 매니페스트는 설치, 다시 시작 제거 작업 전후에 실행할 Windows PowerShell 스크립트를 선언할 수 있습니다. 예를 들어, 다음 매니페스트는 Windows PowerShell 스크립트 `PostInstallSetup.ps1`을 실행하여 ASP.NET Core 애플리케이션이 IIS에 배포된 후 추가 설정 작업을 수행합니다. 이처럼 스크립트를 추가할 때 `project.json` 파일에서와 같이 스크립트가 `aws-windows-deployment-manifest.json` 파일의 publishOptions 아래에 있는 include 섹션에 추가되도록 하십시오. 그렇지 않으면 스크립트가 dotnet CLI publish 명령의 일부로 포함되지 않습니다.

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

ebextensions

Elastic Beanstalk .ebextensions 구성 파일은 다른 모든 Elastic Beanstalk 컨테이너와 마찬가지로 지원됩니다. ebextensions를 ASP.NET Core 애플리케이션에 포함하려면 `.ebextensions` 파일의 `include` 아래에 있는 `publishOptions` 섹션에 `project.json` 디렉터리를 추가하십시오. ebextensions 체크아웃에 대한 자세한 내용은 Elastic Beanstalk 개발자 안내서를 참조하십시오.

.NET 및 Elastic Beanstalk에 대한 다중 애플리케이션 지원

배포 매니페스트를 사용하면 여러 애플리케이션을 동일한 Elastic Beanstalk 환경에 배포할 수 있습니다.

배포 매니페스트는 ASP.NET Core 웹 애플리케이션과 기존 ASP.NET 애플리케이션용 msdeploy 아카이브를 지원합니다. 프런트엔드에 ASP.NET Core를 사용하고 확장 API에 웹 API 프로젝트를 사용하여 뛰어난 새 애플리케이션을 작성한 시나리오가 있다고 가정해 보겠습니다. 또한 기존 ASP.NET을 사용하여 작성한 관리자 앱도 있습니다.

도구 키트의 배포 마법사는 단일 프로젝트의 배포에 중점을 둡니다. 여러 애플리케이션 배포를 잘 활용하려면 애플리케이션 번들을 수동으로 생성해야 합니다. 시작하려면 매니페스트를 작성합니다. 이 예제의 경우 솔루션의 루트에 매니페스트를 작성합니다.

매니페스트의 배포 단원에는 두 개의 하위 요소가 있습니다. 하나는 배포할 ASP.NET Core 웹 애플리케이션의 어레이이며 다른 하나는 배포할 msdeploy 아카이브의 어레이입니다. 각 애플리케이션의 경우 IIS 경로 및 애플리케이션의 비트 위치를 매니페스트에 상대적으로 설정합니다.

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

매니페스트를 작성하면, Windows PowerShell을 사용하여 애플리케이션 번들을 생성하고 기존 Elastic Beanstalk 환경을 업데이트하여 실행합니다. 스크립트가 Visual Studio 솔루션을 포함하는 폴더에서 실행된다는 가정 하에 작성됩니다.

먼저 스크립트에서 애플리케이션 번들을 생성할 작업 영역 폴더를 설정합니다.

```
$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
}
```

폴더를 생성하면 프런트엔드를 가져올 준비가 되었습니다. 배포 마법사와 마찬가지로 .NET CLI를 사용하여 애플리케이션을 게시합니다.

```
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
```

"frontend" 하위 폴더는 출력 폴더로 사용되었으며, 매니페스트에서 설정한 폴더와 일치합니다. 이제 Web API 프로젝트에 대해서도 동일하게 수행해야 합니다.

```
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
```

관리자 사이트는 기존 ASP.NET 애플리케이션이므로, .NET CLI를 사용할 수 없습니다. 관리자 애플리케이션의 경우, msbuild를 통해 빌드 대상 패키지를 전달하여 msdeploy 아카이브를 생성합니다. 기본적으로 패키지 대상은 `obj\Release\Package` 폴더 아래에 msdeploy 아카이브를 생성하므로, 해당 아카이브를 게시 작업 영역으로 복사해야 합니다.

```
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
```

이러한 모든 애플리케이션으로 작업할 사항을 Elastic Beanstalk 환경에 알리려면 솔루션에서 게시 작업 영역으로 매니페스트를 복사한 다음 폴더를 압축하세요.

```
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)
```

이제 애플리케이션 번들이 있으므로 웹 콘솔로 이동하여 아카이브를 Elastic Beanstalk 환경에 업로드할 수 있습니다. 또는 계속 AWS PowerShell cmdlet을 사용하여 애플리케이션 번들과 함께 Elastic Beanstalk 환경을 업데이트할 수 있습니다. `Set-AWSCredentials` 및 `Set-DefaultAWSRegion` cmdlet을 사용하여 현재 프로필과 리전을 Elastic Beanstalk 환경을 포함하는 프로필과 리전으로 설정해야 합니다.

```
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
```

이제 툴킷 또는 웹 콘솔에서 Elastic Beanstalk 환경 상태 페이지를 사용하여 업데이트 상태를 확인합니다. 완료되면 배포 매니페스트에서 설정한 IIS 경로에서 배포한 각 애플리케이션을 탐색할 수 있습니다.

Amazon EC2 컨테이너 서비스에 배포

> **Important**
>
> 새로운 AWS에 게시 기능은 .NET 애플리케이션을 AWS에 게시하는 방법을 단순화하도록 설계되었습니다. AWS에 컨테이너 게시를 선택한 후 이 게시 환경으로 전환할지 묻는 메시지가 표시될 수 있습니다. 자세한 내용은 Visual Studio에서 AWS에 게시 작업 섹션을 참조하세요.

고도의 확장성과 우수한 성능을 갖춘 컨테이너 관리 서비스인 Amazon Elastic Container Service는 도커 컨테이너를 지원하며, Amazon EC2 인스턴스의 관리형 클러스터에서 손쉽게 애플리케이션을 실행할 수 있습니다.

Amazon Elastic Container Service에서 애플리케이션을 배포하려면 애플리케이션 구성 요소가 도커 컨테이너에서 실행되도록 개발되어야 합니다. 도커 컨테이너는 소프트웨어 애플리케이션이 필요한 모든 것(코드, 런타임, 시스템 도구, 시스템 라이브러리 등)을 포함하는, 소프트웨어 개발의 표준화된 단위입니다.

Toolkit for Visual Studio는 Amazon ECS를 통한 애플리케이션 게시를 간소화하는 마법사를 제공합니다. 마법사에 대해서는 다음 단원에서 설명합니다.

Amazon ECS에 대한 자세한 정보는 Elastic Container Service 설명서를 참조하세요. Docker 기본 사항 및 클러스터 생성에 대한 개요가 포함되어 있습니다.

주제

- ASP.NET Core 2 애플리케이션의 AWS 자격 증명 지정
- Amazon ECS에 ASP.NET Core 2.0 앱 배포(Fargate)(레거시)
- Amazon ECS에 ASP.NET Core 2.0 앱 배포(EC2)

ASP.NET Core 2 애플리케이션의 AWS 자격 증명 지정

Docker 컨테이너에 애플리케이션을 배포하면 배포 자격 증명과 인스턴스 자격 증명이라는 두 가지 유형의 자격 증명이 실행됩니다.

배포 자격 증명은 Publish Container to AWS 마법사에서 Amazon ECS에 환경을 생성하는 데 사용됩니다. 여기에는 작업, 서비스, IAM 역할, Docker 컨테이너 리포지토리 및 로드 밸런서(선택한 경우)가 포함되어 있습니다.

인스턴스 자격 증명은 인스턴스(애플리케이션 포함)에서 다양한 AWS 서비스에 액세스하는 데 사용됩니다. 예를 들어 ASP.NET Core 2.0 애플리케이션이 Amazon S3 객체에 대해 읽기 및 쓰기를 수행하려면 적절한 권한이 필요합니다. 환경에 따라 서로 다른 방법을 사용하여 다양한 자격 증명을 제공할 수 있습니다. 예를 들어 ASP.NET Core 2 애플리케이션이 개발 및 프로덕션 환경을 대상으로 할 수 있습니다. 개발 및 프로덕션 환경에서 정의된 역할을 위해 Docker 인스턴스 및 자격 증명을 사용할 수 있습니다.

배포 자격 증명 지정

에 컨테이너 게시 AWS 마법사에서 지정하는 AWS 계정은 마법사가 Amazon ECS에 배포하는 데 사용할 AWS 계정입니다. 계정 프로필에는 Amazon Elastic Compute Cloud, Amazon Elastic Container Service 및에 대한 권한이 있어야 합니다 AWS Identity and Access Management.

드롭다운 목록에서 옵션이 누락되어 있다면 여러분이 권한이 없기 때문일 수 있습니다. 예를 들어 애플리케이션에 대한 클러스터를 생성했지만 AWS에 컨테이너 게시 AWS 마법사의 클러스터 페이지에서 확인할 수 없는 경우가 여기에 해당됩니다. 이 문제가 발생하면 누락된 권한을 추가하고 다시 마법사를 시도해 보십시오.

개발 인스턴스 자격 증명 지정

프로덕션 환경이 아닌 경우에는 appsettings에서 자격 증명을 구성할 수 있습니다. <environment>.json 파일. 예를 들어 Visual Studio 2017의 appsettings.Development.json 파일에서 자격 증명을 구성하려면

1. 프로젝트에 AWSSDK.Extensions.NETCore.Setup NuGet 패키지를 추가합니다.
2. appsettings.Development.json에 AWS 설정을 추가합니다. 아래는 `Profile` 및 `Region`을 설정하는 구성 옵션입니다.

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

프로덕션 인스턴스 자격 증명 지정

프로덕션 인스턴스의 경우 IAM 역할을 사용하여 액세스할 수 있는 애플리케이션 및 서비스를 제어하는 것이 좋습니다. 예를 들어 AWS Management Console에서 Amazon Simple Storage Service 및 Amazon DynamoDB에 대한 권한을 가진 서비스 보안 주체로서 Amazon ECS와 함께 IAM 역할을 구성하려면 다음을 수행하세요.

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/iam/ IAM 콘솔을 엽니다.
2. IAM 콘솔의 탐색 창에서 역할을 선택하고 역할 생성을 선택합니다.
3. AWS 서비스 역할 유형을 선택한 다음, EC2 Container Service를 선택하세요.
4. EC2 Container Service Task(EC2 Container Service 작업) 사용 사례를 선택합니다. 사용 사례는 서비스에 필요한 신뢰 정책을 포함하도록 서비스에서 정의합니다. 그런 다음 다음: 권한을 선택합니다.
5. AmazonS3FullAccess 및 AmazonDynamoDBFullAccess 권한 정책을 선택합니다. 각 정책 옆 확인란을 선택한 다음, 다음: 검토를 선택합니다.
6. 역할 이름에서 이 역할의 목적을 식별하는 데 도움이 되는 역할 이름이나 역할 이름 접두사를 입력합니다. 역할 이름은 AWS 계정 내에서 고유해야 합니다. 대/소문자를 구분하지 않습니다. 예를 들어, 이름이 `PRODROLE`과 `prodrole`, 두 가지로 지정된 역할을 만들 수는 없습니다. 다양한 주체가 역할을 참조할 수 있기 때문에 역할이 생성된 후에는 역할 이름을 편집할 수 없습니다.
7. (선택 사항) 역할 설명에 새 역할에 대한 설명을 입력합니다.
8. 역할을 검토한 다음 역할 생성을 선택합니다.

AWS에 컨테이너 게시 마법사의 ECS 작업 정의 페이지에서 이 역할을 작업 역할로 사용할 수 있습니다.

자세한 내용은 서비스 기반 역할 사용을 참조하십시오.

Amazon ECS에 ASP.NET Core 2.0 앱 배포(Fargate)(레거시)

> **Important**
>
> 이 설명서에서는 레거시 서비스 및 기능을 참조합니다. 업데이트된 안내서와 정보는 AWS .NET 배포 도구 안내서 및 업데이트된 AWS목차 배포를 참조하세요.

이 섹션에서는 Toolkit for Visual Studio의 일부로 제공되는 AWS에 컨테이너 게시 마법사를 사용하여 Fargate 시작 유형을 이용해 Amazon ECS를 통해 Linux를 대상으로 하는 컨테이너화된 ASP.NET Core 2.0 애플리케이션을 배포하는 방법을 설명합니다. 웹 애플리케이션은 연속 실행이 되도록 설계되었기 때문에 서비스 방식으로 배포가 됩니다.

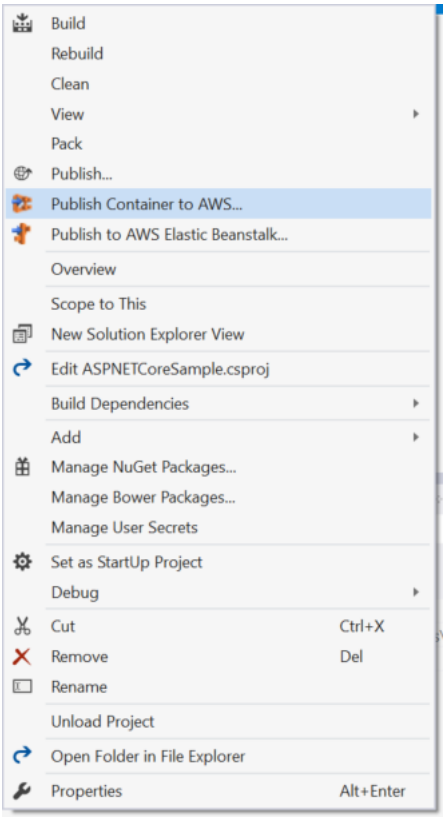
컨테이너 게시 전

AWS에 컨테이너 게시 마법사를 사용하여 ASP.NET Core 2.0 애플리케이션을 배포하기 전에 다음을 수행합니다.

- AWS 보안 인증 정보를 지정하고 Amazon ECS를 설정하세요.
- Docker를 설치합니다. Windows용 Docker를 포함하여 서로 다른 설치 옵션이 몇 가지 제공됩니다.
- Visual Studio에서, Linux를 대상으로 하는 ASP.NET Core 2.0 컨테이너화된 앱용 프로젝트를 만들거나 엽니다.

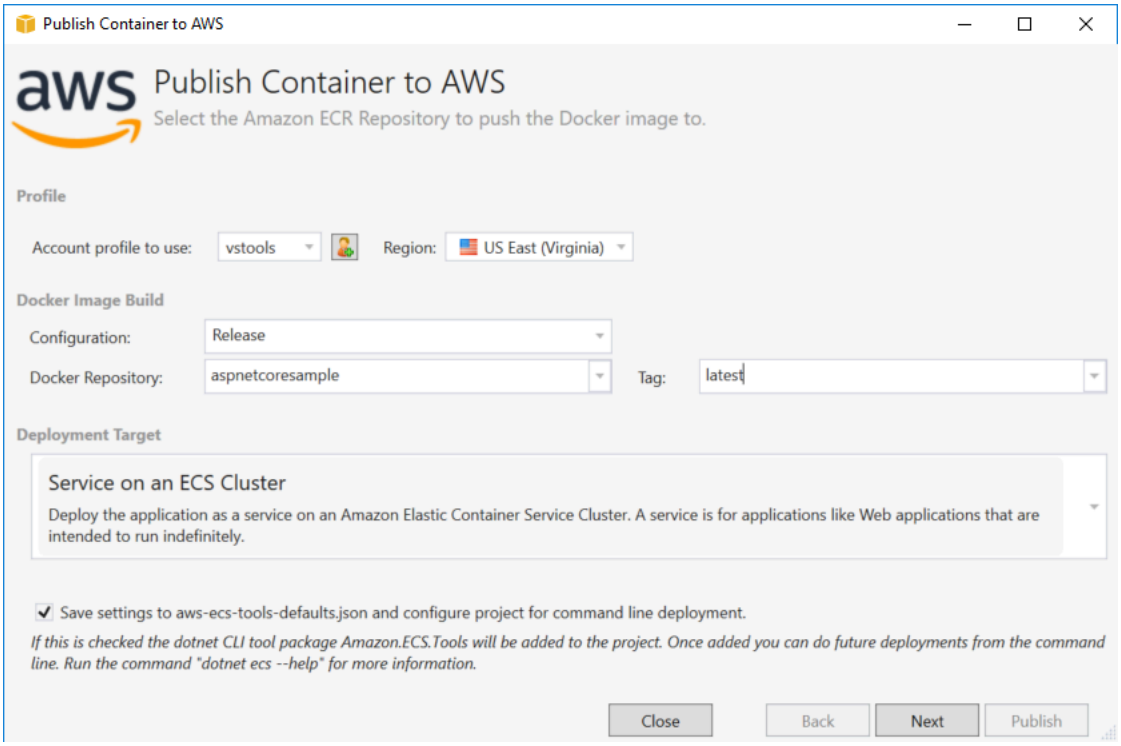
에 컨테이너 게시 AWS 마법사 액세스

Linux를 대상으로 하는 ASP.NET Core 2.0 컨테이너화된 애플리케이션을 배포하려면 솔루션 탐색기에서 해당 프로젝트를 마우스 오른쪽 버튼으로 클릭하고 AWS에 컨테이너 게시를 선택하세요.



Visual Studio 빌드 메뉴에서 AWS에 컨테이너 게시를 선택할 수도 있습니다.

AWS 마법사에 컨테이너 게시



사용할 계정 프로필 - 사용할 계정 프로필을 선택합니다.

리전 - 배포 리전을 선택합니다. 프로필과 리전은 배포 환경 리소스를 설정하고 기본 Docker 레지스트리를 선택하는 데 사용됩니다.

구성 - Docker 이미지 빌드 구성을 선택합니다.

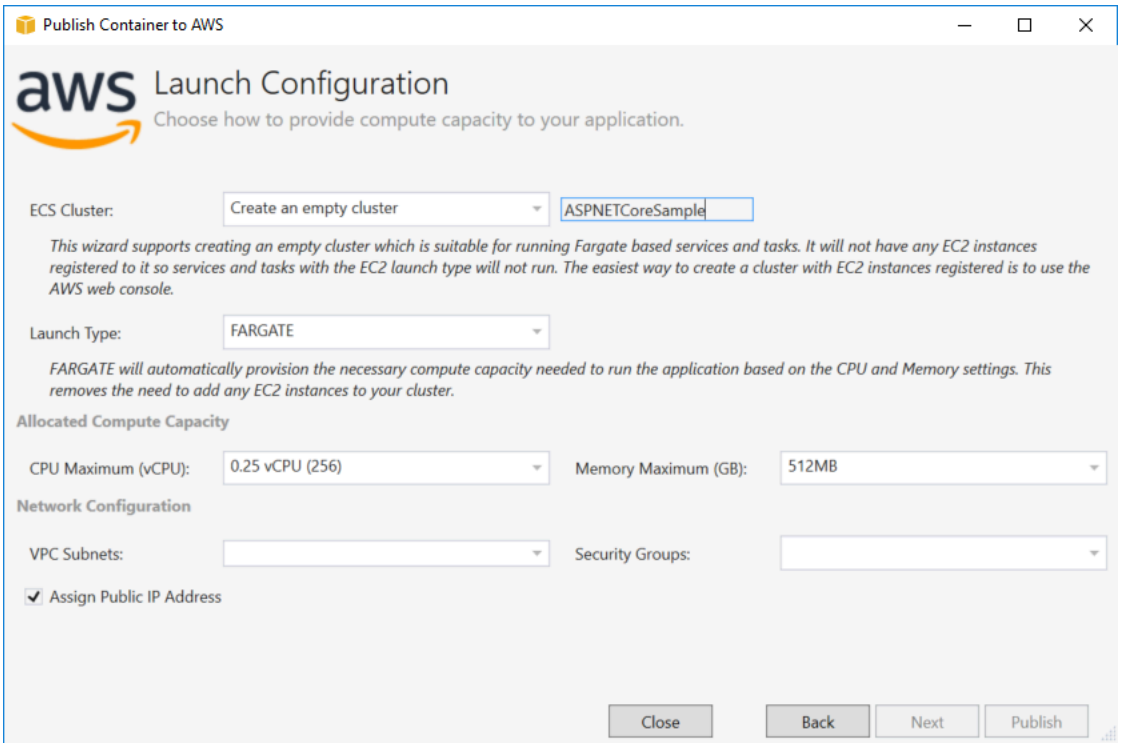
Docker 리포지토리 - 기존의 Docker 레지스트리를 선택하거나 새 레지스트리의 이름을 입력하여 레지스트리를 생성합니다. 이 레지스트리에 빌드된 컨테이너가 게시됩니다.

태그 - 기존 태그를 선택하거나 새 태그의 이름을 입력합니다. 태그는 Docker 컨테이너의 버전, 옵션 또는 기타 고유의 구성 요소 같이 중요한 세부 정보를 추적할 수 있습니다.

배포 대상 - Service on an ECS Cluster(ECS 클러스터 서비스)를 선택합니다. 이 배포 옵션은 애플리케이션이 장기 실행용으로 개발된 경우(예: ASP.NET Core 웹 애플리케이션)에 사용합니다.

설정을 **`aws-docker-tools-defaults.json`**에 저장하고 명령줄에 대한 프로젝트를 구성 - 명령줄에서 유연하게 배포하고 싶은 경우에 이 옵션을 선택합니다. 배포할 프로젝트 디렉터리의 `dotnet ecs deploy`와 `dotnet ecs publish` 컨테이너를 사용합니다.

시작 구성 페이지



ECS 클러스터 - Docker 이미지가 실행될 클러스터를 선택합니다. 빈 클러스터를 생성하겠다고 선택한 경우에는 새 클러스터의 이름을 입력합니다.

시작 유형 - FARGATE를 선택합니다.

CPU 최대 용량(vCPU) - 애플리케이션에 필요한 최대 컴퓨팅 파워를 선택합니다. CPU 및 메모리 값의 허용 범위는 작업 크기를 참조하십시오.

메모리 최대 용량(GB) - 애플리케이션에서 사용할 수 있는 최대 메모리 용량을 선택합니다.

VPC 서브넷 - 단일 VPC에 있는 서브넷을 하나 이상 선택합니다. 서브넷을 하나 이상 선택하면 여러 서브넷에 작업이 분산됩니다. 따라서 가용성을 높일 수 있습니다. 자세한 내용은 기본 VPC 및 기본 서브넷을 참조하십시오.

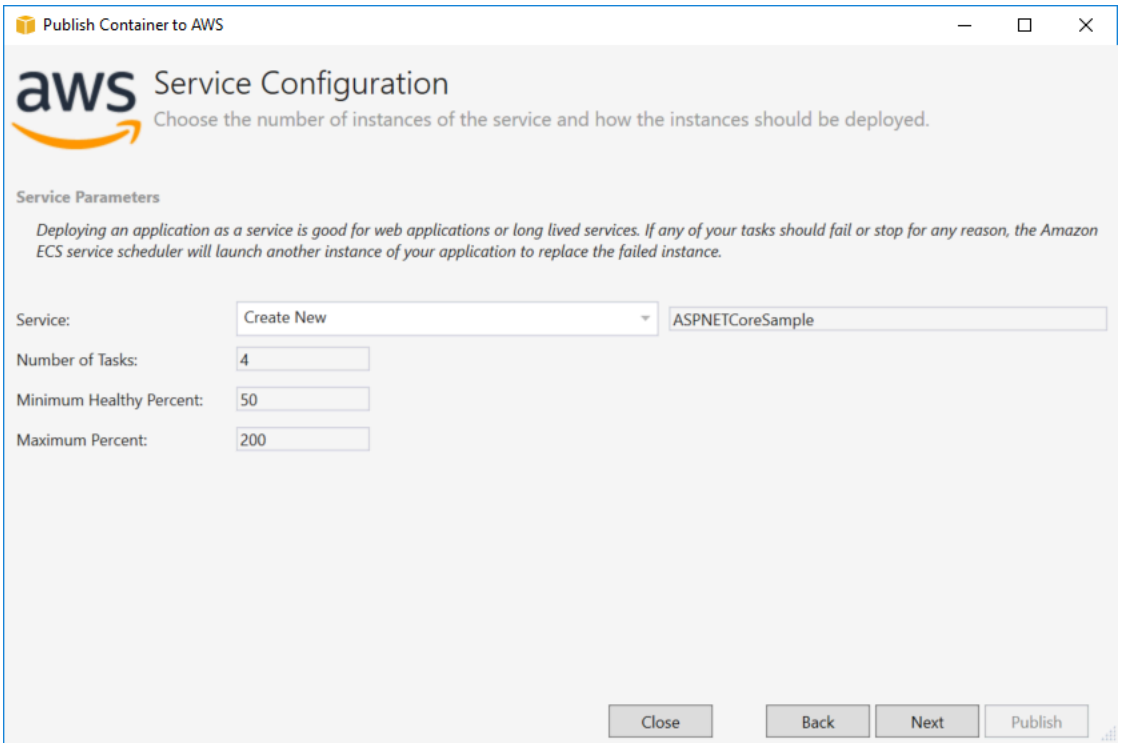
보안 그룹 - 보안 그룹을 선택합니다.

보안 그룹은 연결된 Amazon EC2 인스턴스에 대해 방화벽 역할을 하여 인스턴스 수준에서 인바운드 트래픽과 아웃바운드 트래픽을 모두 제어합니다.

기본 보안 그룹은 동일한 보안 그룹에 할당된 인스턴스에서의 인바운드 트래픽과 모든 아웃바운드 IPv4 트래픽을 허용하도록 구성되어 있습니다. 서비스가 컨테이너 리포지토리에 도달할 수 있으려면 아웃바운드 트래픽이 허용되어야 합니다.

퍼블릭 IP 주소 할당 - 인터넷에서 작업을 액세스할 수 있도록 하려면 이 확인란을 선택합니다.

서비스 구성 서비스



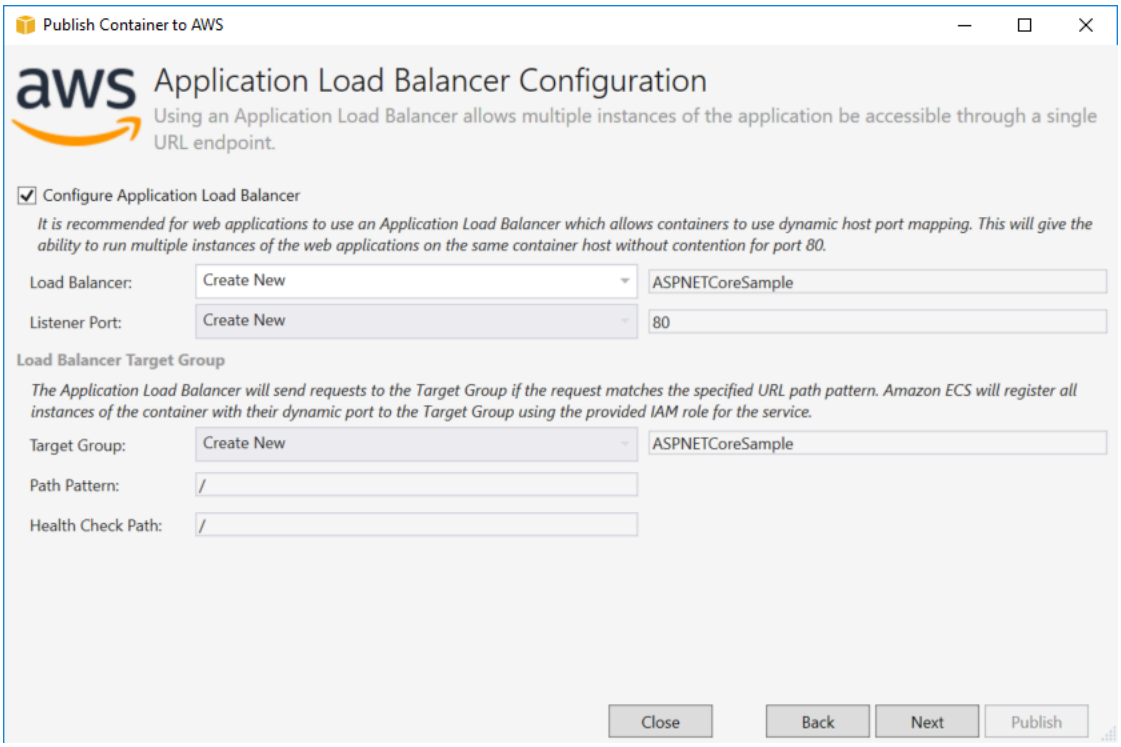
서비스 - 드롭다운에서 서비스 중 하나를 선택하여 기존 서비스에 컨테이너를 추가합니다. 또는 새로 생성을 선택하여 서비스를 새로 생성합니다. 서비스 이름은 클러스터 내에서 고유해야 하지만, 한 리전 또는 여러 리전에 걸쳐 존재하는 여러 클러스터에서 비슷한 서비스 이름을 사용할 수 있습니다.

작업 수 - 클러스터에 배포하여 계속 실행할 작업의 수입니다. 각 작업은 컨테이너 인스턴스의 하나입니다.

최소 정상 상태 백분율 - 배포 동안 RUNNING 상태를 반드시 유지해야 하는 작업의 백분율을 가장 가까운 정수로 반올림한 값입니다.

최대 정상 상태 백분율 - 배포 중에 RUNNING 또는 PENDING 상태가 허용되는 작업의 백분율을 가장 가까운 정수로 반내림한 값입니다.

Application Load Balancer 페이지



Application Load Balancer 구성 - Application Load Balancer를 구성했는지 확인합니다.

로드 밸런서 - 기존 로드 밸런서를 선택하거나 새로 생성을 선택하고 새 로드 밸런서의 이름을 입력합니다.

리스너 포트 - 기존 리스너 포트를 선택하거나 새로 생성을 선택하고 포트 번호를 입력합니다. 기본 포트인 80은 대부분의 웹 애플리케이션에 적합합니다.

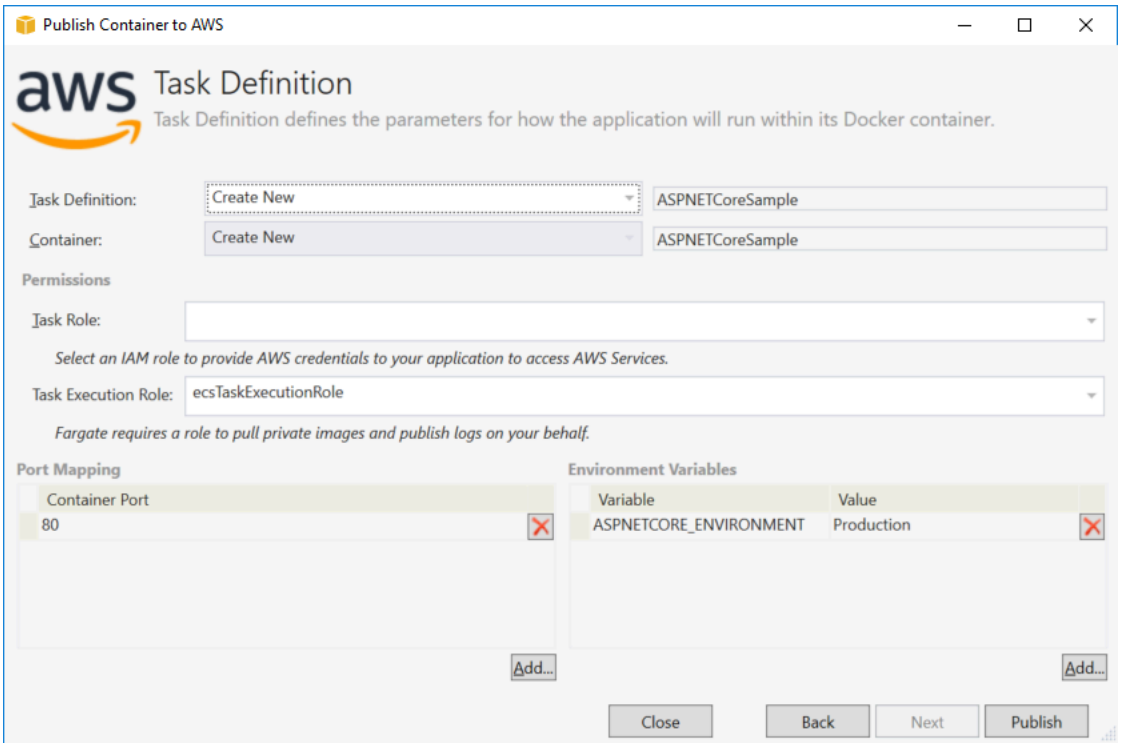
대상 그룹 - Amazon ECS가 서비스를 위해 작업을 등록할 대상 그룹을 선택합니다.

경로 패턴 - 로드 밸런서는 경로 기반 라우팅을 사용합니다. 기본 /를 수락하거나 다른 패턴을 제공합니다. 경로 이름은 대/소문자를 구별하고 최대 128자이며 선택한 문자 집합을 포함합니다.

상태 확인 경로 - 상태 확인을 위한 대상에서 목적지가 되는 ping 경로입니다. 기본 설정은 /입니다. 필요할 경우 다른 이름을 입력하십시오. 입력한 경로가 잘못된 경우에는 상태 확인이 실패하게 되고 상태가 비정상으로 간주됩니다.

여러 서비스가 배포되었고 각 서비스가 서로 다른 경로나 위치에 배포된 경우에는 사용자 지정 확인 경로가 필요할 수 있습니다.

작업 정의 페이지



작업 정의 - 기존 작업 정의를 선택하거나 새로 생성을 선택하고 새 작업 정의 이름을 입력합니다.

컨테이너 - 기존 컨테이너를 선택하거나 새로 생성을 선택하고 새 컨테이너 이름을 입력합니다.

작업 역할 - 앱이 AWS 서비스에 액세스하는 데 필요한 자격 증명이 있는 IAM 역할을 선택합니다. 이것이 바로 애플리케이션에 보안 자격 증명이 전달되는 과정입니다. 애플리케이션에 대해 AWS 보안 인증 정보를 지정하는 방법을 참조하세요.

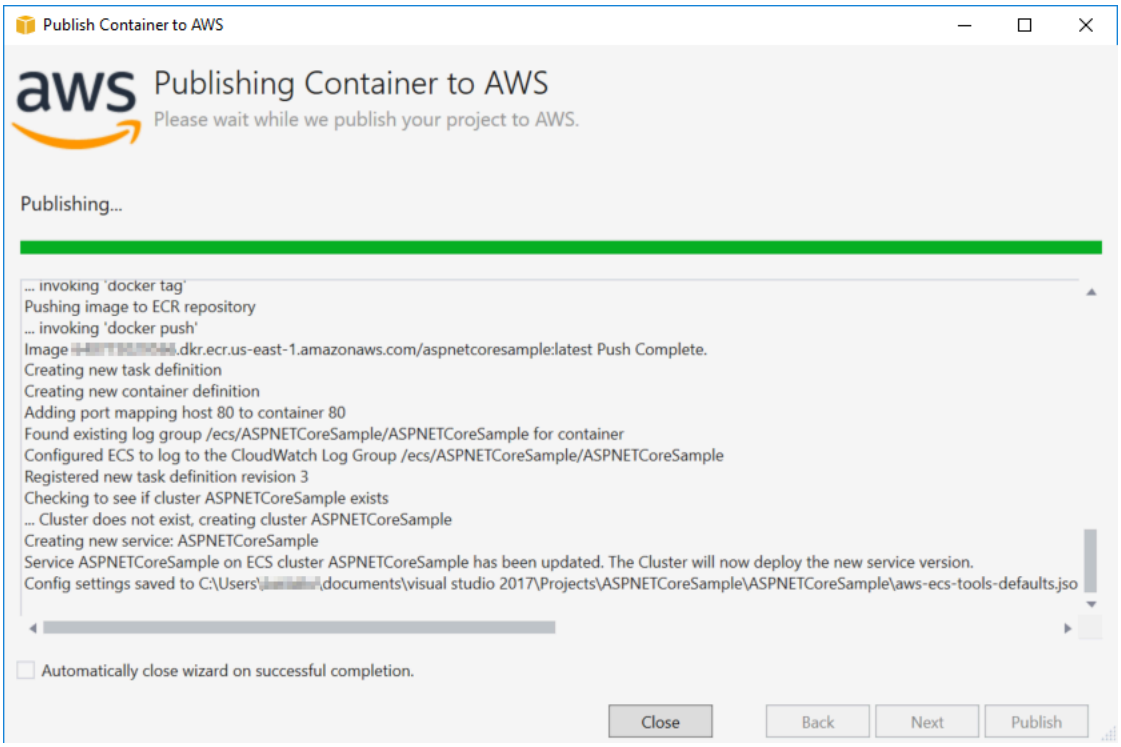
작업 실행 역할 - 프라이빗 이미지를 가져오고 로그를 게시할 권한이 있는 역할을 선택합니다. AWS Fargate는 사용자를 대신하여 해당 역할을 사용합니다.

포트 매핑 - 자동 할당된 호스트 포트에 바인딩되는 컨테이너에서 포트 번호를 선택합니다.

환경 변수 - 컨테이너의 환경 변수를 추가, 수정 또는 삭제합니다. 배포에 맞게 수정이 가능합니다.

구성에 만족하면 게시를 클릭하여 배포 프로세스를 시작합니다.

에 컨테이너 게시 AWS



배포 중에 이벤트가 표시됩니다. 이 마법사는 작업이 성공적으로 완료되면 자동으로 닫힙니다. 페이지 맨 아래에 있는 상자를 선택 해제하면 재정의를 할 수 있습니다.

AWS 탐색기에서 새 인스턴스의 URL을 찾을 수 있습니다. Amazon ECS and Clusters 노드를 확장하고 클러스터를 클릭합니다.

Amazon ECS에 ASP.NET Core 2.0 앱 배포(EC2)

이 섹션에서는 Toolkit for Visual Studio의 일부로 제공되는 AWS에 컨테이너 게시 마법사를 사용하여 Amazon ECS를 통해 Linux를 대상으로 하는 컨테이너화된 ASP.NET Core 2.0 애플리케이션을 EC2 시작 유형으로 배포하는 방법을 설명합니다. 웹 애플리케이션은 연속 실행이 되도록 설계되었기 때문에 서비스 방식으로 배포됩니다.

컨테이너 게시 전

AWS에 컨테이너 게시를 사용하여 ASP.NET Core 2.0 애플리케이션을 배포하기 전에 다음을 수행합니다.

- AWS 보안 인증 정보를 지정하고 Amazon ECS를 설정하세요.
- Docker를 설치합니다. Windows용 Docker를 포함하여 서로 다른 설치 옵션이 몇 가지 제공됩니다.

- 웹 애플리케이션의 요구에 따라 Amazon ECS 클러스터를 생성합니다. 이 작업은 몇 가지 단계만 거치면 됩니다.
- Visual Studio에서, Linux를 대상으로 하는 ASP.NET Core 2.0 컨테이너화된 애플리케이션용 프로젝트를 만들거나 엽니다.

에 컨테이너 게시 AWS 마법사 액세스

Linux를 대상으로 하는 ASP.NET Core 2.0 컨테이너화된 애플리케이션을 배포하려면 솔루션 탐색기에서 해당 프로젝트를 마우스 오른쪽 버튼으로 클릭하고 AWS에 컨테이너 게시를 선택하세요.

Visual Studio 빌드 메뉴에서 AWS에 컨테이너 게시를 선택할 수도 있습니다.

AWS 마법사에 컨테이너 게시

사용할 계정 프로필 - 사용할 계정 프로필을 선택합니다.

리전 - 배포 리전을 선택합니다. 프로필과 리전은 배포 환경 리소스를 설정하고 기본 Docker 레지스트리를 선택하는 데 사용됩니다.

구성 - Docker 이미지 빌드 구성을 선택합니다.

Docker 리포지토리 - 기존의 Docker 레지스트리를 선택하거나 새 레지스트리의 이름을 입력하여 레지스트리를 생성합니다. 이 레지스트리에 빌드된 컨테이너 이미지가 게시됩니다.

태그 - 기존 태그를 선택하거나 새 태그의 이름을 입력합니다. 태그는 Docker 컨테이너의 버전, 옵션 또는 기타 고유의 구성 요소 같이 중요한 세부 정보를 추적할 수 있습니다.

배포 - Service on an ECS Cluster(ECS 클러스터 서비스)를 선택합니다. 이 배포 옵션은 애플리케이션이 장기 실행용으로 개발된 경우(예: ASP.NET Core 2.0 웹 애플리케이션)에 사용합니다.

설정을 **`aws-docker-tools-defaults.json`**에 저장하고 명령줄에 대한 프로젝트를 구성 - 명령줄에서 유연하게 배포하고 싶은 경우에 이 옵션을 선택합니다. 배포할 프로젝트 디렉터리의 `dotnet ecs deploy`와 `dotnet ecs publish` 컨테이너를 사용합니다.

시작 구성 페이지

ECS 클러스터 - Docker 이미지가 실행될 클러스터를 선택합니다. AWS 관리 콘솔을 사용하여 ECS 클러스터를 생성할 수 있습니다.

시작 유형 - EC2를 선택합니다. Fargate 시작 유형을 사용하는 방법은 Amazon ECS에 ASP.NET Core 2.0 애플리케이션 배포(Fargate)를 참조하십시오.

서비스 구성 서비스

서비스 - 드롭다운에서 서비스 중 하나를 선택하여 기존 서비스에 컨테이너를 추가합니다. 또는 새로 생성을 선택하여 서비스를 새로 생성합니다. 서비스 이름은 클러스터 내에서 고유해야 하지만, 한 리전 또는 여러 리전에 걸쳐 존재하는 여러 클러스터에서 비슷한 서비스 이름을 사용할 수 있습니다.

작업 수 - 클러스터에 배포하여 계속 실행할 작업의 수입니다. 각 작업은 컨테이너 인스턴스의 하나입니다.

최소 정상 상태 백분율 - 배포 동안 RUNNING 상태를 반드시 유지해야 하는 작업의 백분율을 가장 가까운 정수로 반올림한 값입니다.

최대 정상 상태 백분율 - 배포 중에 RUNNING 또는 PENDING 상태가 허용되는 작업의 백분율을 가장 가까운 정수로 반내림한 값입니다.

배치 템플릿 - 작업 배치 템플릿을 선택합니다.

작업을 클러스터로 시작할 때 Amazon ECS는 CPU와 메모리 등 작업 정의에서 지정한 요구 사항에 따라 작업을 어디에 배치할지 결정해야 합니다. 마찬가지로 작업 수를 축소하는 경우, Amazon ECS는 어떤 태스크를 종료할지 결정해야 합니다.

배치 템플릿은 작업이 클러스터로 시작되는 방법을 제어합니다.

- AZ Balanced Spread - 작업을 가용 영역과 가용 영역의 컨테이너 인스턴스에 분산합니다.
- AZ Balanced BinPack - 작업을 가용 영역과 가용 메모리가 최소인 컨테이너 인스턴스에 분산합니다.
- BinPack - 사용 가능한 CPU 또는 메모리 최소량에 따라 작업을 분산합니다.
- One Task Per Host - 각 컨테이너 인스턴스에 서비스의 작업을 최대 1개 배치합니다.

자세한 내용은 Amazon ECS 작업 배치를 참조하십시오.

Application Load Balancer 페이지

Application Load Balancer 구성 - Application Load Balancer를 구성했는지 확인합니다.

서비스의 IAM 역할 선택 - 기존 역할을 선택하거나 새로 생성을 선택하여 새 역할을 생성합니다.

로드 밸런서 - 기존 로드 밸런서를 선택하거나 새로 생성을 선택하고 새 로드 밸런서의 이름을 입력합니다.

리스너 포트 - 기존 리스너 포트를 선택하거나 새로 생성을 선택하고 포트 번호를 입력합니다. 기본 포트인 80은 대부분의 웹 애플리케이션에 적합합니다.

대상 그룹 - 기본적으로 로드 밸런서는 대상 그룹에 대해 지정한 포트와 프로토콜을 사용하여 등록된 대상으로 요청을 전송합니다. 또는 대상 그룹에 각 대상을 등록할 때 이 포트를 재정의할 수 있습니다.

경로 패턴 - 로드 밸런서는 경로 기반 라우팅을 사용합니다. 기본 /를 수락하거나 다른 패턴을 제공합니다. 경로 이름은 대/소문자를 구별하고 최대 128자이며 선택한 문자 집합을 포함합니다.

상태 확인 경로 - 상태 확인을 위한 대상에서 목적지가 되는 ping 경로입니다. 기본 설정인 /는 대부분의 웹 애플리케이션에 적합합니다. 필요할 경우 다른 이름을 입력하십시오. 입력한 경로가 잘못된 경우에는 상태 확인이 실패하게 되고 상태가 비정상으로 간주됩니다.

여러 서비스가 배포되었고 각 서비스가 배포된 경로나 위치가 서로 다른 경우에는 사용자 지정 확인 경로가 필요할 수 있습니다.

ECS 작업 정의 페이지

작업 정의 - 기존 작업 정의를 선택하거나 새로 생성을 선택하고 새 작업 정의 이름을 입력합니다.

컨테이너 - 기존 컨테이너를 선택하거나 새로 생성을 선택하고 새 컨테이너 이름을 입력합니다.

메모리(MiB) - 소프트 제한 또는 하드 제한 값을 입력하거나 두 값을 모두 입력합니다.

컨테이너용으로 예약할 메모리의 소프트 제한(MiB)입니다. Docker는 소프트 한계 내에서 컨테이너 메모리를 유지하려고 합니다. 컨테이너는 메모리 파라미터에 지정된 하드 한계까지 추가 메모리를 사용하거나 컨테이너 인스턴스의 모든 가용 메모리를 사용할 수 있습니다(둘 중 먼저 발생하는 쪽).

컨테이너에 표시할 메모리의 하드 제한(MiB)입니다. 컨테이너가 여기서 지정된 메모리를 초과하려 하면 해당 컨테이너가 중지됩니다.

작업 역할 - 컨테이너 권한이 사용자를 대신하여 연결된 정책에 지정된 AWS APIs를 호출할 수 있도록 허용하는 IAM 역할에 대한 작업 역할을 선택합니다. 이것이 바로 애플리케이션에 보안 자격 증명이 전달되는 과정입니다. 애플리케이션의 AWS 보안 자격 증명을 지정하는 방법을 알아봅니다.

포트 매핑 - 컨테이너의 포트 매핑을 추가, 수정 또는 삭제합니다. 로드 밸런서가 활성화되어 있는 경우, 호스트 포트는 기본적으로 0으로 설정되고 포트 할당이 동적으로 이루어집니다.

환경 변수 - 컨테이너의 환경 변수를 추가, 수정 또는 삭제합니다.

구성에 만족하면 게시를 클릭하여 배포 프로세스를 시작합니다.

에 컨테이너 게시 AWS

배포 중에 이벤트가 표시됩니다. 이 마법사는 작업이 성공적으로 완료되면 자동으로 닫힙니다. 페이지 맨 아래에 있는 상자를 선택 해제하면 재정의를 할 수 있습니다.

AWS 탐색기에서 새 인스턴스의 URL을 찾을 수 있습니다. Amazon ECS and Clusters 노드를 확장하고 클러스터를 클릭합니다.

문제 해결 AWS Toolkit for Visual Studio

다음 섹션에는 도구 키트의 AWS Toolkit for Visual Studio 및 AWS 서비스 작업에 대한 일반적인 문제 해결 정보가 포함되어 있습니다.

> **Note**
>
> 설치 및 설정별 문제 해결 정보는 이 사용 설명서에 있는 설치 문제 해결 주제에서 확인할 수 있습니다.

주제

문제 해결 모범 사례

다음은 AWS Toolkit for Visual Studio 문제 해결 시 권장되는 모범 사례입니다.

- Visual Studio 복구 및 시스템 다시 시작
- 보고서를 보내기 전에 문제나 오류를 재생성해 봅니다.
- 재생성 프로세스 중에 각 단계, 설정 및 오류 메시지를 자세히 기록해 둡니다.
- AWS 도구 키트 로그를 수집합니다. 도구 키트 로그를 찾는 AWS 방법에 대한 자세한 설명은이 가이드 주제에 있는 AWS 로그를 찾는 방법 절차를 참조하세요.
- 미해결 요청, 알려진 솔루션을 확인하거나 AWS Toolkit for Visual Studio GitHub 리포지토리의 문제 섹션에서 해결되지 않은 AWS Toolkit for Visual Studio 문제를 보고합니다.

Visual Studio 복구 및 시스템 다시 시작

1. 실행 중인 모든 Visual Studio 인스턴스를 닫습니다.
2. Windows 시작 메뉴에서 Visual Studio 설치 프로그램을 시작합니다.

3. Visual Studio의 영향을 받는 설치에서 복구를 실행합니다. 이를 통해 Visual Studio는 설치된 확장의 인덱스를 다시 빌드할 수 있습니다.
4. Visual Studio를 다시 시작하기 전에 Windows를 다시 시작합니다.

AWS 도구 키트 로그를 찾는 방법

1. Visual Studio 기본 메뉴에서 확장을 확장하세요.
2. AWS 도구 키트를 선택하여 AWS 도구 키트 메뉴를 확장한 다음 도구 키트 로그 보기를 선택합니다.
3. 운영 체제에서 AWS 도구 키트 로그 폴더가 열리면 파일을 날짜별로 정렬하고 현재 문제와 관련된 정보가 포함된 로그 파일을 찾습니다.

Amazon Q 보안 스캔 보기 및 필터링

Visual Studio에서 Amazon Q 보안 스캔을 보려면 Visual Studio 기본 메뉴에서 보기 제목을 확장하고 오류 목록을 선택하여 Visual Studio 오류 목록을 엽니다.

기본적으로 Visual Studio 오류 목록에는 코드 기반에 대한 모든 경고 및 오류가 표시됩니다. Visual Studio 오류 목록에서 Amazon Q 보안 스캔 조사 결과를 필터링하려면 다음 절차를 완료하여 필터를 생성합니다.

> **Note**
>
> Amazon Q 보안 스캔 조사 결과는 보안 스캔을 실행하여 문제가 감지된 후에만 표시됩니다. Amazon Q 보안 스캔 조사 결과는 Visual Studio에 경고로 표시됩니다. Amazon Q 보안 스캔 조사 결과를 보려면 오류 목록에서 오류 목록 제목의 경고 옵션을 선택해야 합니다.

1. Visual Studio 기본 메뉴에서 보기 제목을 확장한 다음 오류 목록을 선택하여 오류 목록 창을 엽니다.
2. 오류 목록 창에서 헤더 행을 마우스 오른쪽 버튼으로 클릭하여 컨텍스트 메뉴를 엽니다.
3. 컨텍스트 메뉴에서 열 표시를 확장한 다음 확장된 메뉴에서 도구를 선택합니다.
4. 도구 열이 오류 목록에 추가됩니다.
5. 도구 열 헤더에서 필터 아이콘을 선택하고 Amazon Q를 선택하여 Amazon Q 보안 스캔 조사 결과를 필터링합니다.

AWS 도구 키트가 제대로 설치되지 않았습니다.

문제:

Visual Studio를 시작한 후 AWS Toolkit for Visual Studio 1분 이내에 출력 창과 정보 표시줄에 각각 다음 메시지가 표시됩니다.

```
Some Toolkit components could not be initialized. Some functionality may
not work during this IDE session.
```

```
The AWS Toolkit is not properly installed.
```

해결 방법:

확장을 업데이트하거나 설치하면 Visual Studio의 일부 내부 캐시 파일이 동기화되지 않을 수 있습니다. 다음 절차에서는 다음에 Visual Studio를 시작할 때 이러한 파일을 다시 빌드하는 방법을 설명합니다.

> **Note**
>
> 이 솔루션은 Visual Studio 사용자 지정에 영향을 미칠 수 있습니다. 이 절차를 완료한 후에는 AWS 도구 키트 확장이 설치된 것으로 나열되고 더 이상 오류 메시지를 보고하지 않아야 합니다. 다음 단계를 완료한 후에도이 문제가 계속 발생하면 AWS Toolkit for Visual Studio GitHub 리포지토리의 문제 #452를 참조하세요.

1. 최신 버전의 Visual Studio 2022를 설치합니다.

   > **Note**
   >
   > 최소 필수 버전은 17.11.5입니다.

2. 실행 중인 모든 Visual Studio 인스턴스를 닫습니다.
3. Windows에서 개발자 명령 프롬프트를 관리자 권한으로 엽니다.
4. 개발자 명령 프롬프트에서 `devenv /updateconfiguration /resetExtensions` 명령을 실행한 다음 명령이 완료될 때까지 기다립니다.
5. 명령이 완료되면 Visual Studio를 다시 시작합니다.
6. Visual Studio에서 AWS 확장은 이제 설치된 것으로 나열되며 더 이상이 문제의 맨 위에 나열된 오류 메시지를 보고하지 않습니다.

방화벽 및 프록시 설정

방화벽 및 프록시 설정 문제 해결

보안 스캔 소프트웨어는 다운로드에서 파일을 제거하거나 다운로드를 완전히 방지하여 AWS 툴킷 언어 서버에서 파일을 다운로드하는 기능을 방해할 수 있습니다.

방화벽 및 프록시 설정을 확인하려면 Visual Studio 인스턴스와 동일한 시스템에 설치된 인터넷 브라우저에서 https://aws-toolkit-language-servers.amazonaws.com/codewhisperer/0/manifest.json으로 이동합니다. 오류가 발생하거나 페이지를 로드할 수 없는 경우 방화벽 또는 프록시 필터로 인해 `aws-toolkit-language-servers.amazonaws.com`에 도달하지 못할 수 있습니다.

사용자 지정 인증서

는 Node.js 런타임에서 실행되는 언어 서버를 AWS Toolkit for Visual Studio 활용합니다. 네트워크에서 사용자 지정 인증서를 사용하는지 확인하는 방법에 대한 자세한 내용은 AWS Command Line Interface 버전 1 사용 설명서에서AWS CLI의 구성 및 자격 증명 파일 설정 주제를 참조하세요.

프록시 설정을 구성하고 인증서를 정의하려면 `HTTPS_PROXY` env 변수를 구성하고 `NODE_OPTIONS` 및 `NODE_EXTRA_CA_CERTS` 키에 대한 Windows 환경 변수를 생성해야 합니다.

`HTTPS_PROXY` env 변수를 구성하려면 다음 단계를 완료합니다.

1. Visual Studio 기본 메뉴에서 도구를 선택한 다음 옵션을 선택합니다.
2. 옵션 메뉴에서 AWS 툴킷을 확장한 다음 프록시를 선택합니다.
3. 프록시 메뉴에서 호스트 및 포트를 정의합니다.

> **Note**
>
> `HTTPS_PROXY`에서를 구성하는 방법에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 주제에 HTTP 프록시 사용을 AWS CLI AWS CLI참조하세요.

다음 키에 대한 Windows 환경 변수를 생성합니다.

- `NODE_OPTIONS = --use-openssl-ca`
- `NODE_EXTRA_CA_CERTS = Path/To/Corporate/Certs`

> **Note**
>
> 기업 루트 인증서 추출에 대한 자세한 내용은 learn.microsoft.com의 프라이빗 키로 인증서 내보내기 문서를 참조하세요. Windows 환경 변수 키에 대한 자세한 내용은 nodejs.org에서 Node.js v23.3.0 설명서를 참조하세요.

목록 및 추가 단계 허용

방화벽 설정은 AWS Toolkit 언어 서버를 방해하는 것 외에도 Amazon Q가 Amazon S3에 업로드되어 서비스 API를 호출하지 못하게 할 수 있습니다. 이러한 오류의 가능성을 최소화하려면 다음 엔드포인트에 대해 포트 443(HTTPS)에서 아웃바운드 인터넷 액세스를 허용하는 것이 좋습니다.

- `https://codewhisperer.us-east-1.amazonaws.com/`
- `https://amazonq-code-transformation-us-east-1-c6160f047e0.s3.amazonaws.com/`
- `https://aws-toolkit-language-servers.amazonaws.com/`
- `https://q.us-east-1.amazonaws.com`
- `https://client-telemetry.us-east-1.amazonaws.com`
- `https://cognito-identity.us-east-1.amazonaws.com`
- `https://oidc.us-east-1.amazonaws.com`

엔드포인트의 자세한 목록은 이 사용 설명서의 액세스를 허용하도록 방화벽 및 게이트웨이 업데이트 주제를 참조하세요. Amazon Q용 회사 프록시 구성에 대한 자세한 내용은 Amazon Q Developer 사용 설명서의 Amazon Q에서 회사 프록시 구성 주제를 참조하세요. 방화벽 및 프록시 문제가 계속 발생하면 AWS 도구 키트 로그를 수집하고 AWS Toolkit for Visual Studio GitHub 리포지토리의 AWS Toolkit for Visual Studio 문제 섹션을 통해 AWS Toolkit for Visual Studio 팀에 문의하세요. AWS 도구 키트 로그 수집에 대한 자세한 내용은이 사용 설명서 주제의 문제 해결 모범 사례 섹션에 있는 정보를 참조하세요.

에 대한 보안 AWS Toolkit for Visual Studio

Amazon Web Services(AWS)에서 가장 우선순위가 높은 것이 클라우드 보안입니다. AWS 고객으로서 여러분은 가장 높은 보안 요구 사항을 충족하기 위해 설계된 데이터 센터 및 네트워크 아키텍처의 혜택을 받게 됩니다. 보안은 AWS 와 사용자 간의 공동 책임입니다. 공동 책임 모델은 이 사항을 클라우드 내 보안 및 클라우드의 보안으로 설명합니다.

클라우드 보안 - AWS 는 클라우드에서 제공되는 모든 서비스를 실행하는 인프라를 보호하고 안전하게 사용할 수 있는 서비스를 AWS 제공할 책임이 있습니다. 당사의 보안 책임은에서 최우선 순위이며 AWS, 타사 감사자는 AWS 규정 준수 프로그램의 일환으로 보안의 효과를 정기적으로 테스트하고 검증합니다.

클라우드의 보안 - 사용자의 책임은 사용 중인 AWS 서비스와 데이터의 민감도, 조직의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요인에 따라 결정됩니다.

이 AWS 제품 또는 서비스는 지원하는 특정 Amazon Web Services(AWS) 서비스를 통해 공동 책임 모델을 따릅니다. AWS 서비스 보안 정보는 AWS 서비스 보안 설명서 페이지 및 규정 AWSAWS 준수 프로그램의 규정 준수 작업 범위에 속하는 서비스를 참조하세요.

주제

의 데이터 보호 AWS Toolkit for Visual Studio

AWS 공동 책임 모델 Amazon Q를 사용하는 AWS Toolkit for Visual Studio의 데이터 보호에 적용됩니다.이 모델에 설명된 대로 AWS 는 모든를 실행하는 글로벌 인프라를 보호할 책임이 있습니다 AWS 클라우드. 사용자는 이 인프라에 호스팅되는 콘텐츠에 대한 통제 권한을 유지할 책임이 있습니다. 사용하는 AWS 서비스 의 보안 구성과 관리 태스크에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 관한 자세한 내용은 데이터 프라이버시 FAQ를 참조하세요. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 AWS 공동 책임 모델 및 GDPR 블로그 게시물을 참조하세요.

데이터 보호를 위해 자격 증명을 보호하고 AWS 계정 AWS IAM Identity Center 또는 AWS Identity and Access Management (IAM)를 사용하여 개별 사용자를 설정하는 것이 좋습니다. 이렇게 하면 개별 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 다중 인증(MFA)을 사용합니다.
- SSL/TLS를 사용하여 AWS 리소스와 통신합니다. TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- 를 사용하여 API 및 사용자 활동 로깅을 설정합니다 AWS CloudTrail. CloudTrail 추적을 사용하여 AWS 활동을 캡처하는 방법에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 CloudTrail 추적 작업을 참조하세요.
- 내부의 모든 기본 보안 제어와 함께 AWS 암호화 솔루션을 사용합니다 AWS 서비스.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용합니다.
- 명령줄 인터페이스 또는 API를 AWS 통해에 액세스할 때 FIPS 140-3 검증 암호화 모듈이 필요한 경우 FIPS 엔드포인트를 사용합니다. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 연방 정보 처리 표준(FIPS) 140-3을 참조하세요.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 형식 텍스트 필드에 입력하지 않는 것이 좋습니다. 여기에는 콘솔 AWS CLI, API 또는 AWS SDKs를 사용하여 Amazon Q 또는 기타 AWS 서비스 에서 AWS 도구 키트를 사용하는 경우가 포함됩니다. 이름에 사용되는 태그 또는 자유 형식 텍스트 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 보안 인증 정보를 URL에 포함시켜서는 안 됩니다.

자격 증명 및 액세스 관리

AWS Identity and Access Management (IAM)는 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어하는 데 도움이 AWS 서비스 되는 입니다. IAM 관리자는 누가 AWS 리소스를 사용할 수 있는 인증(로그인) 및 권한(권한 있음)을 받을 수 있는지 제어합니다. IAM은 추가 비용 없이 사용할 수 AWS 서비스 있는 입니다.

주제

- IAM AWS 서비스 작업 방법
- AWS 자격 증명 및 액세스 문제 해결

대상

AWS Identity and Access Management (IAM)를 사용하는 방법은에서 수행하는 작업에 따라 다릅니다 AWS.

서비스 사용자 - AWS 서비스 를 사용하여 작업을 수행하는 경우 필요한 자격 증명과 권한을 관리자가 제공합니다. 더 많은 AWS 기능을 사용하여 작업을 수행하게 되면 추가 권한이 필요할 수 있습니다. 액세스 권한 관리 방법을 이해하면 관리자에게 올바른 권한을 요청하는 데 도움이 됩니다. 에서 기능에 액세스할 수 없는 경우 사용 중인의 AWS 자격 증명 및 액세스 문제 해결 또는 사용 설명서를 AWS참조 AWS 서비스 하세요.

서비스 관리자 - 회사에서 AWS 리소스를 책임지고 있는 경우에 대한 전체 액세스 권한을 가지고 있을 것입니다 AWS. 서비스 관리자는 서비스 사용자가 액세스해야 하는 AWS 기능과 리소스를 결정합니다. 그런 다음 IAM 관리자에게 요청을 제출하여 서비스 사용자의 권한을 변경해야 합니다. 이 페이지의 정보를 검토하여 IAM의 기본 개념을 이해하세요. 회사가 IAM을 사용하는 방법에 대해 자세히 알아보려면 사용 중인의 AWS 서비스 사용 설명서를 AWS참조하세요.

IAM 관리자 - IAM 관리자라면 AWS에 대한 액세스 권한 관리 정책 작성 방법을 자세히 알고 싶을 것입니다. IAM에서 사용할 수 있는 자격 AWS 증명 기반 정책 예제를 보려면 사용 중인의 사용 설명서를 참조 AWS 서비스 하세요.

ID를 통한 인증

인증은 자격 증명 자격 증명을 AWS 사용하여에 로그인하는 방법입니다. AWS 계정 루트 사용자, IAM 사용자 또는 IAM 역할을 수임하여 인증되어야 합니다.

AWS IAM Identity Center (IAM Identity Center), Single Sign-On 인증 또는 Google/Facebook 자격 증명과 같은 자격 증명 소스의 자격 증명을 사용하여 페더레이션 자격 증명으로 로그인할 수 있습니다. 로그인하는 방법에 대한 자세한 내용은 AWS 로그인 사용 설명서의 AWS 계정에 로그인하는 방법 섹션을 참조하세요.

프로그래밍 방식 액세스를 위해는 요청에 암호화 방식으로 서명할 수 있는 SDK 및 CLI를 AWS 제공합니다. 자세한 내용은 IAM 사용 설명서의 API 요청용AWS Signature Version 4 섹션을 참조하세요.

AWS 계정 루트 사용자

를 생성할 때 모든 AWS 서비스 및 리소스에 대한 완전한 액세스 권한이 있는 AWS 계정 theroot 사용자라는 하나의 로그인 자격 증명으로 AWS 계정시작합니다. 일상적인 태스크에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 자격 증명이 필요한 작업은 IAM 사용 설명서의 루트 사용자 자격 증명이 필요한 작업 섹션을 참조하세요.

페더레이션 ID

가장 좋은 방법은 인간 사용자에게 자격 증명 공급자와의 페더레이션을 사용하여 임시 자격 증명을 AWS 서비스 사용하여에 액세스하도록 요구하는 것입니다.

페더레이션 자격 증명은 엔터프라이즈 디렉터리, 웹 자격 증명 공급자 또는 자격 증명 소스의 자격 증명을 AWS 서비스 사용하여 Directory Service 에 액세스하는 사용자입니다. 페더레이션 ID는 임시 자격 증명을 제공하는 역할을 수임합니다.

중앙 집중식 액세스 관리를 위해 AWS IAM Identity Center를 추천합니다. 자세한 정보는 AWS IAM Identity Center 사용 설명서의 What is IAM Identity Center?를 참조하세요.

IAM 사용자 및 그룹

IAM 사용자는 단일 개인 또는 애플리케이션에 대한 특정 권한을 가진 ID입니다. 장기 자격 증명이 있는 IAM 사용자 대신 임시 자격 증명을 사용하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 자격 증명 공급자와의 페더레이션을 사용하여 임시 자격 증명을 AWS 사용하여에 액세스하도록 인간 사용자에게 요구하기를 참조하세요.

IAM 그룹은 IAM 사용자 모음을 지정하고 대규모 사용자 집합에 대한 관리 권한을 더 쉽게 만듭니다. 자세한 내용은 IAM 사용 설명서의 IAM 사용자 사용 사례 섹션을 참조하세요.

IAM 역할

IAM 역할은 임시 자격 증명을 제공하는 특정 권한이 있는 자격 증명입니다. 사용자에서 IAM 역할(콘솔)로 전환하거나 또는 API 작업을 호출하여 역할을 수임할 수 있습니다. AWS CLI AWS 자세한 내용은 IAM 사용 설명서의 역할 수임 방법을 참조하세요.

IAM 역할은 페더레이션 사용자 액세스, 임시 IAM 사용자 권한, 교차 계정 액세스, 교차 서비스 액세스 및 Amazon EC2에서 실행되는 애플리케이션에 유용합니다. 자세한 내용은 IAM 사용 설명서의 교차 계정 리소스 액세스를 참조하세요.

정책을 사용하여 액세스 관리

정책을 AWS 생성하고 자격 증명 또는 리소스에 연결하여 AWS 에서 액세스를 제어합니다. 정책은 자격 증명 또는 리소스와 연결될 때 권한을 정의합니다.는 보안 주체가 요청할 때 이러한 정책을 AWS 평가합니다. 대부분의 정책은에 JSON 문서 AWS 로 저장됩니다. JSON 정책 문서에 대한 자세한 내용은 IAM 사용 설명서의 JSON 정책 개요 섹션을 참조하세요.

정책을 사용하여 관리자는 어떤 보안 주체가 어떤 리소스에 대해 어떤 조건에서 작업을 수행할 수 있는지 정의하여 누가 무엇을 액세스할 수 있는지 지정합니다.

기본적으로 사용자 및 역할에는 어떠한 권한도 없습니다. IAM 관리자는 IAM 정책을 생성하고 사용자가 수임할 수 있는 역할에 추가합니다. IAM 정책은 작업을 수행하기 위해 사용하는 방법과 관계없이 작업에 대한 권한을 정의합니다.

ID 기반 정책

ID 기반 정책은 ID(사용자, 사용자 그룹 또는 역할)에 연결하는 JSON 권한 정책 문서입니다. 이러한 정책은 자격 증명이 수행할 수 있는 작업, 대상 리소스 및 이에 관한 조건을 제어합니다. ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서에서 고객 관리형 정책으로 사용자 지정 IAM 권한 정의를 참조하세요.

ID 기반 정책은 인라인 정책(단일 ID에 직접 포함) 또는 관리형 정책(여러 ID에 연결된 독립 실행형 정책)일 수 있습니다. 관리형 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용 설명서의 관리형 정책 및 인라인 정책 중에서 선택 섹션을 참조하세요.

리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 예를 들어 IAM 역할 신뢰 정책 및 Amazon S3 버킷 정책이 있습니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다.

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. 리소스 기반 정책에서는 IAM의 AWS 관리형 정책을 사용할 수 없습니다.

액세스 제어 목록(ACL)

액세스 제어 목록(ACL)은 어떤 위탁자(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACL은 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

Amazon S3 AWS WAF및 Amazon VPC는 ACLs. ACL에 관한 자세한 내용은 Amazon Simple Storage Service 개발자 가이드의 액세스 제어 목록(ACL) 개요를 참조하세요.

기타 정책 타입

AWS 는 보다 일반적인 정책 유형에서 부여한 최대 권한을 설정할 수 있는 추가 정책 유형을 지원합니다.

- 권한 경계 - ID 기반 정책에서 IAM 엔터티에 부여할 수 있는 최대 권한을 설정합니다. 자세한 정보는 IAM 사용 설명서의 IAM 엔터티의 권한 범위를 참조하세요.
- 서비스 제어 정책(SCP) - AWS Organizations내 조직 또는 조직 단위에 대한 최대 권한을 지정합니다. 자세한 내용은AWS Organizations 사용 설명서의 서비스 제어 정책을 참조하세요.
- 리소스 제어 정책(RCP) – 계정의 리소스에 사용할 수 있는 최대 권한을 설정합니다. 자세한 내용은 AWS Organizations 사용 설명서의 리소스 제어 정책(RCP)을 참조하세요.
- 세션 정책 – 역할 또는 페더레이션 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 자세한 내용은 IAM 사용 설명서의 세션 정책을 참조하세요.

여러 정책 유형

여러 정책 유형이 요청에 적용되는 경우, 결과 권한은 이해하기가 더 복잡합니다. 에서 여러 정책 유형이 관련될 때 요청을 허용할지 여부를 AWS 결정하는 방법을 알아보려면 IAM 사용 설명서의 정책 평가 로직을 참조하세요.

IAM AWS 서비스 작업 방법

가 대부분의 IAM 기능을 AWS 서비스 사용하는 방법을 개괄적으로 알아보려면 IAM 사용 설명서의 AWS IAM으로 작업하는 서비스를 참조하세요.

IAM AWS 서비스 에서 특정를 사용하는 방법을 알아보려면 관련 서비스 사용 설명서의 보안 섹션을 참조하세요.

AWS 자격 증명 및 액세스 문제 해결

다음 정보를 사용하여 및 IAM으로 작업할 때 발생할 수 있는 일반적인 문제를 진단 AWS 하고 수정할 수 있습니다.

주제

- 에서 작업을 수행할 권한이 없음 AWS

- iam:PassRole을 수행하도록 인증되지 않음
- 내 외부의 사람이 내 AWS 리소스에 액세스 AWS 계정 하도록 허용하고 싶습니다.

에서 작업을 수행할 권한이 없음 AWS

작업을 수행할 권한이 없다는 오류가 표시되면 작업을 수행할 수 있도록 정책을 업데이트해야 합니다.

다음의 예제 오류는 mateojackson IAM 사용자가 콘솔을 사용하여 가상 *my-example-widget* 리소스에 대한 세부 정보를 보려고 하지만 가상 awes:*GetWidget* 권한이 없을 때 발생합니다.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 awes:GetWidget on resource: my-example-widget
```

이 경우, awes:*GetWidget* 작업을 사용하여 *my-example-widget* 리소스에 액세스할 수 있도록 mateojackson 사용자 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

iam:PassRole을 수행하도록 인증되지 않음

iam:PassRole 작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 AWS에 역할을 전달할 수 있도록 정책을 업데이트해야 합니다.

일부 AWS 서비스 에서는 새 서비스 역할 또는 서비스 연결 역할을 생성하는 대신 기존 역할을 해당 서비스에 전달할 수 있습니다. 이렇게 하려면 역할을 서비스에 전달할 권한이 있어야 합니다.

다음 예 오류는 marymajor라는 IAM 사용자가 콘솔을 사용하여 AWS에서 작업을 수행하려고 하는 경우에 발생합니다. 하지만 작업을 수행하려면 서비스 역할이 부여한 권한이 서비스에 있어야 합니다. Mary는 서비스에 역할을 전달할 권한이 없습니다.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

이 경우, Mary가 iam:PassRole작업을 수행할 수 있도록 Mary의 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

내 외부의 사람이 내 AWS 리소스에 액세스 AWS 계정 하도록 허용하고 싶습니다.

다른 계정의 사용자 또는 조직 외부의 사람이 리소스에 액세스할 때 사용할 수 있는 역할을 생성할 수 있습니다. 역할을 수임할 신뢰할 수 있는 사람을 지정할 수 있습니다. 리소스 기반 정책 또는 액세스 제

어 목록(ACL)을 지원하는 서비스의 경우, 이러한 정책을 사용하여 다른 사람에게 리소스에 대한 액세스 권한을 부여할 수 있습니다.

자세한 내용은 다음을 참조하세요.

- 에서 이러한 기능을 AWS 지원하는지 여부를 알아보려면 섹션을 참조하세요IAM AWS 서비스 작업 방법.
- 소유 AWS 계정 한의 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 소유한 다른의 IAM 사용자에게 액세스 권한 제공을 참조 AWS 계정 하세요.
- 타사에 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 타사가 AWS 계정 소유한에 대한 액세스 권한 제공을 AWS 계정참조하세요.
- ID 페더레이션을 통해 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 외부에서 인증된 사용자에게 액세스 권한 제공(ID 페더레이션)을 참조하세요.
- 크로스 계정 액세스에 대한 역할과 리소스 기반 정책 사용의 차이점을 알아보려면 IAM 사용 설명서의 IAM의 크로스 계정 리소스 액세스를 참조하세요.

이 AWS 제품 또는 서비스에 대한 규정 준수 검증

AWS 서비스 가 특정 규정 준수 프로그램의 범위 내에 있는지 알아보려면 AWS 서비스 규정 준수 프로그램 제공 범위 내를 참조하고 관심 있는 규정 준수 프로그램을 선택합니다. 일반 정보는 AWS 규정 준수 프로그램.

를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다 AWS Artifact. 자세한 내용은 Downloading Reports inDownloading AWS Artifact을 참조하세요.

사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표, 관련 법률 및 규정에 따라 AWS 서비스 결정됩니다. 사용 시 규정 준수 책임에 대한 자세한 내용은 AWS 보안 설명서를 AWS 서비스참조하세요.

이 AWS 제품 또는 서비스는 지원하는 특정 Amazon Web Services(AWS) 서비스를 통해 공동 책임 모델을 따릅니다. AWS 서비스 보안 정보는 AWS 서비스 보안 설명서 페이지 및 규정 AWSAWS 준수 프로그램의 규정 준수 작업 범위에 속하는 서비스를 참조하세요.

이 AWS 제품 또는 서비스에 대한 복원력

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 중심으로 구축됩니다.

AWS 리전 는 지연 시간이 짧고 처리량이 많으며 중복성이 높은 네트워킹과 연결된 물리적으로 분리되고 격리된 여러 가용 영역을 제공합니다.

가용 영역을 사용하면 중단 없이 영역 간에 자동으로 장애 극복 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

AWS 리전 및 가용 영역에 대한 자세한 내용은 AWS 글로벌 인프라를 참조하세요.

이 AWS 제품 또는 서비스는 지원하는 특정 Amazon Web Services(AWS) 서비스를 통해 공동 책임 모델을 따릅니다. AWS 서비스 보안 정보는 AWS 서비스 보안 설명서 페이지 및 규정 AWSAWS 준수 프로그램의 규정 준수 작업 범위에 속하는 서비스를 참조하세요.

이 AWS 제품 또는 서비스에 대한 인프라 보안

이 AWS 제품 또는 서비스는 관리형 서비스를 사용하므로 글로벌 네트워크 보안으로 AWS 보호됩니다. AWS 보안 서비스 및가 인프라를 AWS 보호하는 방법에 대한 자세한 내용은 AWS 클라우드 보안을 참조하세요. 인프라 보안 모범 사례를 사용하여 AWS 환경을 설계하려면 보안 원칙 AWS Well-Architected Framework의 인프라 보호를 참조하세요.

AWS 에서 게시한 API 호출을 사용하여 네트워크를 통해이 AWS 제품 또는 서비스에 액세스합니다. 클라이언트는 다음을 지원해야 합니다.

- Transport Layer Security(TLS). TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- DHE(Ephemeral Diffie-Hellman) 또는 ECDHE(Elliptic Curve Ephemeral Diffie-Hellman)와 같은 완전 전송 보안(PFS)이 포함된 암호 제품군. Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

또한 요청은 액세스 키 ID 및 IAM 위탁자와 관련된 시크릿 액세스 키를 사용하여 서명해야 합니다. 또는 AWS Security Token Service(AWS STS)를 사용하여 임시 보안 자격 증명을 생성하여 요청에 서명할 수 있습니다.

이 AWS 제품 또는 서비스는 지원하는 특정 Amazon Web Services(AWS) 서비스를 통해 공동 책임 모델을 따릅니다. AWS 서비스 보안 정보는 AWS 서비스 보안 설명서 페이지 및 규정 AWSAWS 준수 프로그램의 규정 준수 작업 범위에 속하는 서비스를 참조하세요.

의 구성 및 취약성 분석 AWS Toolkit for Visual Studio

Toolkit for Visual Studio는 새로운 기능 또는 수정 사항이 개발됨에 따라 Visual Studio Marketplace에 릴리스됩니다. 이러한 업데이트에는 보안 업데이트가 포함되는 경우가 있으므로 AWS Toolkit with Amazon Q를 최신 상태로 유지하는 것이 중요합니다.

확장에 대한 자동 업데이트가 활성화되어 있는지 확인하려면

1. 도구, 확장 및 업데이트(Visual Studio 2017) 또는 확장, 확장 관리(Visual Studio 2019)를 선택하여 확장 관리자를 엽니다.
2. 확장 및 업데이트 설정 변경(Visual Studio 2017) 또는 확장 설정 변경(Visual Studio 2019)을 선택하세요.
3. 사용 환경에 맞게 설정을 조정합니다.

확장에 대한 자동 업데이트를 비활성화하도록 선택하는 경우 사용 환경에 적합한 간격으로 AWS Toolkit with Amazon Q에 대한 업데이트를 확인해야 합니다.

AWS Toolkit for Visual Studio 사용 설명서에 대한 문서 기록

문서 기록

다음 표에서는 AWS Toolkit for Visual Studio 사용자 안내서의 중요한 최근 변경 사항에 대해 설명합니다. 이 설명서에 대한 업데이트 알림을 받으려면 RSS feed를 구독하세요.

변경 사항	설명	날짜
시작하기 콘텐츠의 업데이트 사항	UI의 변경 사항을 반영하도록 AWS 콘텐츠 시작하기 및 연결에 대한 업데이트입니다.	2025년 4월 24일
액세스를 허용하도록 방화벽 및 게이트웨이 업데이트	확장을 위해 AWS Toolkit for Visual Studio with Amazon Q에서 모든 서비스 및 기능에 액세스할 수 있도록 허용해야 하는 엔드포인트 및 리소스 목록입니다.	2025년 3월 20일
방화벽 및 프록시 설정 문제 해결	AWS Toolkit for Visual Studio 및 Amazon Q에 대한 방화벽 및 프록시 설정을 다루는 새로운 문제 해결 주제가 추가되었습니다.	2024년 12월 15일
설치 업데이트 문제 해결	Microsoft의 업데이트를 고려하여 설치 문제 콘텐츠를 업데이트합니다.	2024년 11월 20일
시작하기 콘텐츠의 업데이트 사항	UI의 변경 사항을 반영하도록 AWS 콘텐츠 시작하기 및 연결에 대한 업데이트입니다.	2024년 10월 24일
AWS에 연결에 대한 업데이트	AWS 콘텐츠 연결에 대한 업데이트입니다.	2024년 9월 26일

Amazon EC2 AMI 콘텐츠 업데이트	Amazon EC2 AMI 프로세스 및 절차에 대한 변경 사항을 문서화하기 위해 콘텐츠가 업데이트되었습니다.	2024년 9월 13일
AWS 툴킷 구성 요소를 초기화할 수 없음	AWS Toolkit for Visual Studio 구성 요소가 초기화되지 않는 문제를 해결하기 위한 문제 해결 주제가 추가되었습니다.	2024년 9월 13일
Amazon Q 보안 스캔 보기 및 필터링	Amazon Q 보안 스캔을 보고 필터링하는 데 도움이 되는 문제 해결 주제가 추가되었습니다.	2024년 7월 31일
AWS Toolkit for Visual Studio 용 Amazon Q	이제 AWS Toolkit for Visual Studio에 Amazon Q를 사용할 수 있습니다.	2024년 6월 30일
콘텐츠 업데이트 및 유지 관리	UI 및 AWS 스타일 지침의 변경 사항에 대한 콘텐츠 업데이트입니다.	2024년 3월 6일
콘텐츠 업데이트 및 유지 관리	UI 및 AWS 스타일 지침의 변경 사항에 대한 콘텐츠 업데이트입니다.	2024년 3월 6일
콘텐츠 업데이트 및 유지 관리	UI 및 AWS 스타일 지침의 변경 사항에 대한 콘텐츠 업데이트입니다.	2024년 3월 6일
콘텐츠 업데이트 및 유지 관리	UI 및 AWS 스타일 지침의 변경 사항에 대한 콘텐츠 업데이트입니다.	2024년 3월 6일
콘텐츠 업데이트 및 유지 관리	UI 및 AWS 스타일 지침의 변경 사항에 대한 콘텐츠 업데이트입니다.	2024년 3월 6일

설정 및 인증을 위한 업데이트	보안 및 툴킷 온보딩 환경을 개선하기 위해 설정 및 인증 주제가 업데이트되었습니다. 변경 내용을 보려면 시작하기 및 인증 및 액세스 주제 TOC를 참조하세요.	2023년 6월 22일
인증 및 액세스	이제 AWS 보안 인증 정보를 제공하는 것은 인증 및 액세스입니다. AWS 스타일 및 보안 요구 사항을 충족하도록 TOC 및 하위 주제를 리팩터링합니다.	2023년 5월 4일
설정 섹션 및 주제에 대한 업데이트	이 사용 설명서의 AWS Toolkit for Visual Studio 설정 섹션 및 주제는 AWS Toolkit for Visual Studio에 대한 보딩 경험을 개선하기 위해 업데이트되었습니다.	2023년 1월 30일
설정 섹션 및 주제에 대한 업데이트	이 사용 설명서의 AWS Toolkit for Visual Studio 설정 섹션 및 주제는 AWS Toolkit for Visual Studio에 대한 보딩 경험을 개선하기 위해 업데이트되었습니다.	2023년 1월 30일
2022년 AWS Toolkit for Visual Studio 정보를 추가함	Visual Studio 2022에 대한 지원이 AWS Toolkit for Visual Studio에 추가되었습니다.	2022년 12월 20일
AWS에 게시 안내서 업데이트	GA 출시를 위한 서비스 변경 사항을 반영하기 위한 설명서 업데이트입니다.	2022년 7월 6일

제목 업데이트 및 재배치	내용을 더 잘 반영하기 위해 제목이 약간 변경되었습니다. 이제 이 안내서는 AWS에 게시 안내서에 있습니다.	2022년 7월 6일
AWS에 배포: 제목 및 내용 업데이트	정식으로는 AWS 툴킷을 사용한 배포라는 제목이 붙은 안내서 섹션의 목차(TOC)가 업데이트되었으며 이제 제목은 AWS에 배포입니다. Elastic Beanstalk에 배포(레거시) 및 AWS CloudFormation에 배포(레거시) 안내서는 지원 중단이 완료되어 더 이상 사용할 수 없는 안내서입니다. Elastic Beanstalk 및 Cloudformation으로의 배포와 관련된 업데이트된 내용은 이 안내서의 업데이트된 목차에서 확인할 수 있습니다.	2022년 7월 6일
이제 ASP.NET Core 2.0 앱 배포(Fargate)는 레거시 안내서입니다	이 설명서에서는 레거시 서비스 및 기능을 참조합니다. 업데이트된 안내서와 정보는 AWS .NET 배포 도구 안내서 및 업데이트된 AWS 목차 배포를 참조하세요.	2022년 7월 6일
이제 ASP.NET 앱 배포는 레거시 가이드입니다	이 설명서에서는 레거시 서비스 및 기능을 참조합니다. 업데이트된 안내서와 정보는 AWS .NET 배포 도구 안내서 및 업데이트된 AWS 목차 배포를 참조하세요.	2022년 7월 6일

이제 ASP.NET 앱 배포는 레거시 가이드입니다	이 설명서에서는 레거시 서비스 및 기능을 참조합니다. 업데이트된 안내서와 정보는 AWS .NET 배포 도구 안내서 및 업데이트된 AWS 목차 배포를 참조하세요.	2022년 7월 6일
새 안내서 주제: Visual Studio에서 CloudWatch Logs 작업	Visual Studio에서 Amazon CloudWatch Logs 통합 안내서에 대한 새로운 개요 주제를 만들었습니다.	2022년 6월 29일
새 안내서 주제: Visual Studio CloudWatch Logs 통합 설정	Visual Studio에서 Amazon CloudWatch Logs 통합 안내서에 대한 새로운 설정 섹션을 만들었습니다.	2022년 6월 29일
Visual Studio의 CloudWatch Logs 통합	Visual Studio의 CloudWatch Logs 설정 및 Visual Studio에서 CloudWatch Logs 작업 등의 안내서 주제를 포함하여 Visual Studio에서의 Amazon CloudWatch Logs 통합을 위한 새로운 안내서를 만들었습니다.	2022년 6월 29일
에 게시AWS	AWS에 게시는 더 이상 미리 보기로 제공되지 않습니다. UI 변경 사항 및 게시 제안 개선 사항을 반영하도록 업데이트되었습니다.	2022년 6월 1일
새 AWS에 게시 미리 보기 가능	애플리케이션에 적합한 AWS 서비스에 대한 지침을 제공하는 향상된 배포 환경.	2021년 10월 21일

AWS 보안 인증 정보에 대한 SSO 및 MFA 지원	AWS Single Sign-On(IAM Identity Center) 및 AWS 보안 인증 정보의 다중 인증에 대한 새로운 지원을 문서화하도록 업데이트되었습니다.	2021년 4월 21일
기본 AWS Lambda 프로젝트 생성 도커 이미지 생성	Lambda 컨테이너 이미지에 대한 런타임이 지원됩니다.	2020년 12월 1일
보안 내용	보안 콘텐츠를 추가했습니다.	2020년 2월 6일
AWS 보안 인증 정보 제공	공유 AWS 보안 인증 정보 파일에서 보안 인증 정보 프로필을 만드는 방법에 대한 정보로 업데이트되었습니다.	2019년 6월 20일
AWS Toolkit for Visual Studio의 AWS Lambda 프로젝트 사용	AWS Toolkit for Visual Studio에 Visual Studio 2019에 대한 지원이 추가되었습니다.	2019년 3월 28일
자습서: Amazon Rekognition Lambda 애플리케이션 생성	AWS Toolkit for Visual Studio에 Visual Studio 2019에 대한 지원이 추가되었습니다.	2019년 3월 28일
자습서: AWS Lambda를 사용하여 서버리스 애플리케이션 빌드 및 테스트	AWS Toolkit for Visual Studio에 Visual Studio 2019에 대한 지원이 추가되었습니다.	2019년 3월 28일
설정AWS Toolkit for Visual Studio	Visual Studio 2019에 대한 지원이 AWS Toolkit for Visual Studio에 추가되었습니다.	2019년 3월 28일
ASP.NET Core 2.0 애플리케이션 배포(Fargate)	AWS Toolkit for Visual Studio에 Visual Studio 2019에 대한 지원이 추가되었습니다.	2019년 3월 28일
ASP.NET Core 2.0 애플리케이션(EC2) 배포	AWS Toolkit for Visual Studio에 Visual Studio 2019에 대한 지원이 추가되었습니다.	2019년 3월 28일

Visual Studio에서 AWS CloudFormation 템플릿 프로젝트 생성	AWS Toolkit for Visual Studio에 Visual Studio 2019에 대한 지원이 추가되었습니다.	2019년 3월 28일
컨테이너 서비스의 상세 보기	AWS 탐색기에서 제공하는 Amazon Elastic Container Service 클러스터 및 컨테이너 리포지토리의 세부 보기에 대한 정보가 추가되었습니다.	2018년 2월 16일
Amazon EC2 컨테이너 서비스에 배포	Amazon EC2 컨테이너 서비스에 배포하는 방법에 대한 정보가 추가되었습니다.	2018년 2월 16일
Fargate를 사용하여 컨테이너 서비스 배포	Fargate 시작 유형을 사용하여 Amazon ECS를 통해 Linux를 대상으로 하는 컨테이너화된 ASP.NET Core 2.0 애플리케이션을 배포하는 방법에 대한 정보가 추가되었습니다.	2018년 2월 16일
EC2를 사용하여 컨테이너 서비스 배포	EC2 시작 유형을 사용하여 Amazon ECS를 통해 Linux를 대상으로 하는 컨테이너화된 ASP.NET Core 2.0 애플리케이션을 배포하는 방법에 대한 정보가 추가되었습니다.	2018년 2월 16일
Amazon EC2 컨테이너 서비스에 배포하기 위한 보안 인증 정보	Amazon EC2 컨테이너 서비스에 배포할 때 자격 증명을 지정하는 방법에 대한 정보가 추가되었습니다.	2018년 2월 16일

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.