aws

사용 설명서

AWS Toolkit for JetBrains

AWS Toolkit for JetBrains: 사용 설명서

Table of Contents

AWS Toolkit for JetBrains이란 무엇인가요?

AWS Toolkit for JetBrains은 JetBrains의 통합 개발 환경(IDE)에 대한 오픈 소스 플러그 인입니다. 이 도구 키트를 사용하면 JetBrains IDE에서 AWS 리소스를 사용할 수 있도록 하여 Amazon Web Services(AWS)로 서버리스 애플리케이션을 더 쉽게 개발, 디버그 및 배포할 수 있습니다.

주제

AWS Toolkit for JetBrains에 포함된 도구 키트

AWS Toolkit for JetBrains에는 다음과 같은 특정 도구 키트가 포함되어 있습니다.

- AWS Toolkit for CLion(C & C++ 개발용)
- AWS Toolkit for GoLand(Go 개발용)
- AWS Toolkit for IntelliJ(Java 개발용)
- AWS Toolkit for WebStorm(Node.js 개발용)
- AWS Toolkit for Rider(.NET 개발용)
- AWS Toolkit for PhpStorm (PHP 개발용)
- AWS Toolkit for PyCharm(Python 개발용)
- AWS Toolkit for RubyMine(Ruby 개발용)
- AWS Toolkit for DataGrip(데이터베이스 관리)

> **Note**
>
> 지원되는 JetBrains IDE용 AWS Toolkit 간의 큰 기능 차이는 이 설명서에 나와 있습니다.

AWS Toolkit for JetBrains를 사용하여 AWS Lambda 함수, AWS CloudFormation 스택 및 Amazon Elastic Container Service(Amazon ECS) 클러스터 작업을 수행할 수도 있습니다. AWS Toolkit for JetBrains에는 AWS용 애플리케이션을 단순화하는 AWS 자격 증명 관리 및 AWS 리전 관리와 같은 기능이 포함되어 있습니다.

AWS Toolkit for JetBrains 작업

AWS Toolkit for JetBrains를 사용하여 다음을 수행할 수 있습니다.

- AWS Serverless Application Model(AWS SAM) 애플리케이션 생성, 배포, 업데이트 및 삭제. AWS Toolkit for JetBrains를 통한 AWS SAM 작업에 대한 자세한 내용은 이 사용 설명서의 AWS Serverless 주제를 참조하세요.
- 원격 및 로컬에서 AWS Lambda 함수 생성, 업데이트, 실행 및 디버깅. AWS Toolkit for JetBrains를 통한 AWS Lambda 서비스 작업에 대한 자세한 내용은 이 사용 설명서의 AWS Lambda 주제를 참조하세요.
- AWS CloudFormation 스택 삭제 및 관련 이벤트 로그 보기. CloudFormation 및 AWS Toolkit for JetBrains 작업에 대한 자세한 내용은 이 사용 설명서의 AWS CloudFormation 주제를 참조하세요.
- Amazon Elastic Container Service를 사용하여 AWS 클러스터의 코드 디버그. AWS Toolkit for JetBrains를 사용한 Amazon ECS 작업에 대한 자세한 내용은 이 사용 설명서의 Amazon Elastic Container Service 주제를 참조하세요.
- Amazon EventBridge 스키마 작업. 자세한 내용은 이 사용 설명서의 Amazon EventBridge 스케줄러 주제를 참조하세요.

관련 정보

관련 비디오

- 공지 사항 | AWS Toolkit for IntelliJ IDEA 소개(16분, 2019년 4월, YouTube 웹 사이트)
- AWS Toolkit for JetBrains 시작하기(PyCharm용 AWS Toolkit 관련, 2분, 2018년 11월, YouTube 웹 사이트)
- AWS Toolkit for JetBrains를 사용한 서버리스 응용 프로그램 구축(PyCharm용 AWS Toolkit 관련, 6분, 2018년 11월, YouTube 웹 사이트)

관련 웹 페이지

- IntelliJ용 AWS Toolkit은 현재 일반 사용 가능(2019년 3월, 블로그 게시물, AWS 웹 사이트)
- AWS Toolkit for IntelliJ – 일반 공개(2019년 3월, 블로그 게시물, AWS 웹 사이트)
- 신규 – AWS Toolkits for PyCharm, IntelliJ(평가판)(2018년 11월, 블로그 게시물, AWS 웹 사이트)
- PyCharm용 AWS Toolkit 소개(2018년 11월, 블로그 게시물, AWS 웹사이트)

- IntelliJ용 AWS Toolkit (AWS 웹 사이트의 AWS Toolkit for JetBrains)
- PyCharm용 AWS Toolkit에 대한(AWS 웹 사이트의 AWS Toolkit for JetBrains)
- AWS Toolkit(JetBrains 웹 사이트)
- Develop on AWS with JetBrains Tools(JetBrains 웹 사이트)
- JetBrains의 모든 개발자 도구 및 제품(JetBrains 웹 사이트)

질문 및 도움말

AWS 개발자 커뮤니티에서 질문하거나 도움을 구하려면 다음 AWS 토론 포럼을 참조하십시오.

- C & C++ Development
- Go Development
- Java 개발
- JavaScript 개발
- .NET 개발
- PHP Development
- Python 개발
- Ruby Development

(이러한 포럼에 들어가면 AWS에서 로그인이 필요할 수 있습니다.)

또한 직접 당사에 문의할 수도 있습니다.

AWS Toolkit 버그 보고 또는 기능 요청

AWS Toolkit for JetBrains에 대한 버그를 보고하거나 기능을 요청하려면 GitHub 웹 사이트의 aws/aws-toolkit-jetbrains 리포지토리에 있는 Issues 탭으로 이동하세요. 새 문제를 선택한 다음 화면의 지침에 따라 버그 보고나 기능 요청을 완료합니다. (이 웹 사이트에 들어가면 GitHub에서 로그인을 요구할 수 있습니다.)

AWS Toolkit에 기여

여러분의 AWS Toolkit 기여에 감사 드립니다. 기여를 시작하려면 GitHub 웹 사이트의 aws/aws-toolkit-jetbrains 리포지토리에 있는 Contributing Guidelines를 읽어보세요. (이 웹 사이트에 들어가면 GitHub에서 로그인을 요구할 수 있습니다.)

AWS Toolkit for JetBrains 시작하기

AWS Toolkit for JetBrains를 통해 JetBrains 통합 개발 환경(IDE)에서 직접 AWS 서비스와 리소스를 사용할 수 있습니다.

다음 주제에서는 시작을 위한 AWS Toolkit for JetBrains 설치, 설정 및 구성 프로세스를 안내합니다.

주제

AWS Toolkit for JetBrains 설치

IDE의 JetBrains Marketplace에서 AWS Toolkit for JetBrains를 다운로드, 설치 및 설정할 수 있습니다. 또는 웹 브라우저에서 AWS Toolkit for JetBrains Marketplace 목록으로 이동하여 최신 AWS Toolkit for JetBrains 설치 파일을 다운로드할 수 있습니다.

다음 섹션에서는 JetBrains IDE에서 직접 AWS Toolkit for JetBrains를 설치하고 설정하는 방법을 설명합니다.

JetBrains IDE에서 AWS Toolkit 설치

기본 JetBrains IDE에서 직접 AWS Toolkit for JetBrains를 다운로드하여 설치하려면 다음 절차를 완료하세요.

1. JetBrains 기본 메뉴에서 기본 설정 메뉴를 엽니다(Windows 사용자의 경우 파일을 확장하고 설정을 선택).
2. 기본 설정/설정 메뉴에서 플러그인을 선택하여 플러그인 메뉴를 엽니다.
3. 플러그인 메뉴 탐색에서 마켓플레이스를 선택하여 JetBrains 플러그인 마켓플레이스를 엽니다.
4. 제공된 검색 필드에 **`AWS Toolkit`**을 입력합니다.
5. AWS Toolkit 항목에서 항목 제목 옆에 있는 녹색 설치 버튼을 선택합니다.

6. 설치 프로세스를 계속하려면 타사 플러그인 개인 정보 보호 정보에 동의하세요.
7. 설치가 완료되면 IDE를 다시 시작하라는 메시지가 나타납니다.

AWS Toolkit for JetBrains 조기 액세스 프로그램(EAP) 및 사용자 지정 빌드 설치

AWS Toolkit for JetBrains의 EAP(Early Access Program) 빌드에는 새로운 기능과 실험 기능의 미리 보기가 포함되어 있습니다.

EAP 빌드용 도구 키트를 구성하려면 다음 절차를 완료하세요.

1. JetBrains 기본 메뉴에서 기본 설정 메뉴를 엽니다(Windows 사용자의 경우 파일을 확장하고 설정을 선택).
2. 기본 설정/설정 메뉴에서 플러그인을 선택하여 플러그인 메뉴를 엽니다.
3. 플러그인 메뉴 탐색에서 설정(리포지토리 관리, 프록시 구성 또는 디스크에서 플러그인 설치) 아이콘을 확장하고 플러그인 리포지토리 관리를 선택합니다.
4. 플러그인 리포지토리 관리 메뉴에서 +(추가) 아이콘을 선택하고 AWS Toolkit의 EAP 리포지토리 필드에 **`https://plugins.jetbrains.com/plugins/eap/aws.toolkit`**를 입력합니다.
5. 확인을 선택하여 EAP 설치를 시작합니다.
6. 설치가 완료되면 IDE를 다시 시작하라는 메시지가 나타납니다.

AWS Toolkit for JetBrains EAP 및 사용자 지정 리포지토리 참조 제거

특정 버전의 AWS Toolkit for JetBrains를 사용하려면 EAP 또는 사용자 지정 리포지토리 참조를 제거해야 할 수 있습니다. 리포지토리 참조를 제거하려면 다음 절차를 완료하세요.

> **Note**
>
> 이 절차를 완료한 후에도 다른 버전을 업데이트하거나 설치하기 전에 현재 버전의 AWS Toolkit for JetBrains를 제거해야 할 수 있습니다.

EAP 리포지토리 참조 제거

1. JetBrains 기본 메뉴에서 기본 설정 메뉴를 엽니다(Windows 사용자의 경우 파일을 확장하고 설정을 선택).

2. 기본 설정/설정 메뉴에서 플러그인을 선택하여 플러그인 메뉴를 엽니다.
3. 플러그인 메뉴 탐색에서 설정(리포지토리 관리, 프록시 구성 또는 디스크에서 플러그인 설치) 아이콘을 확장하고 플러그인 리포지토리 관리를 선택합니다.
4. 플러그인 리포지토리 관리 메뉴에서 -(제거) 아이콘을 선택하고 제거를 확인합니다.

AWS Toolkit for JetBrains 탐색

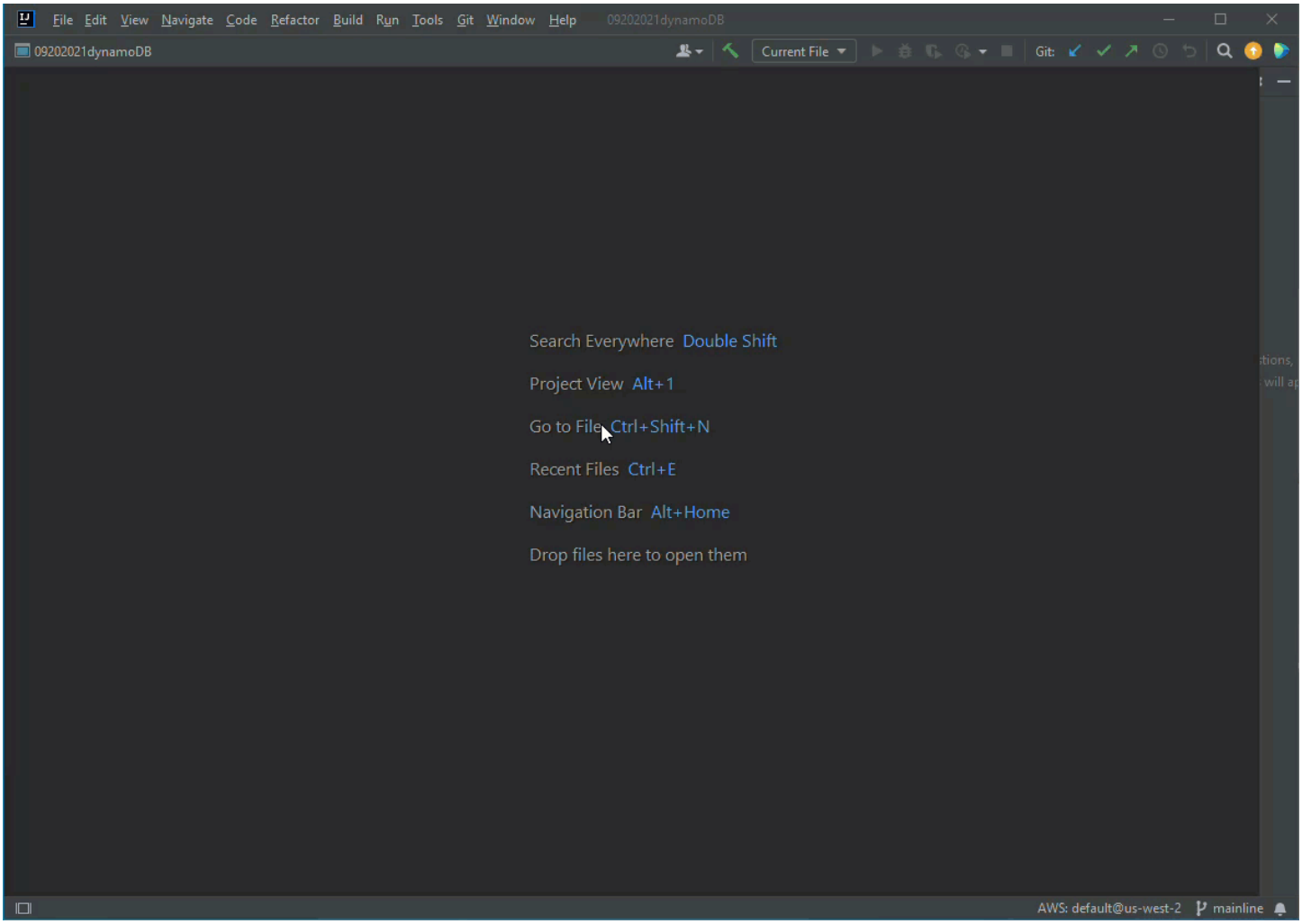
다음 주제에서는 AWS Toolkit for JetBrains의 기본 위치와 구성 요소에 대해 설명합니다.

주제

JetBrains에서 도구 키트 보기

JetBrains IDE에서 도구 키트를 보려면 다음 단계를 완료하세요.

1. JetBrains IDE에서 JetBrains IDE 왼쪽 하단에 있는 활성 도구 모음 아이콘을 사용하여 활성 도구 모음을 확장합니다.
2. 활성 도구 모음에서 AWS Toolkit를 선택합니다.
3. 이제 AWS Toolkit for JetBrains가 활성 도구 모음 창에서 열려 있습니다.



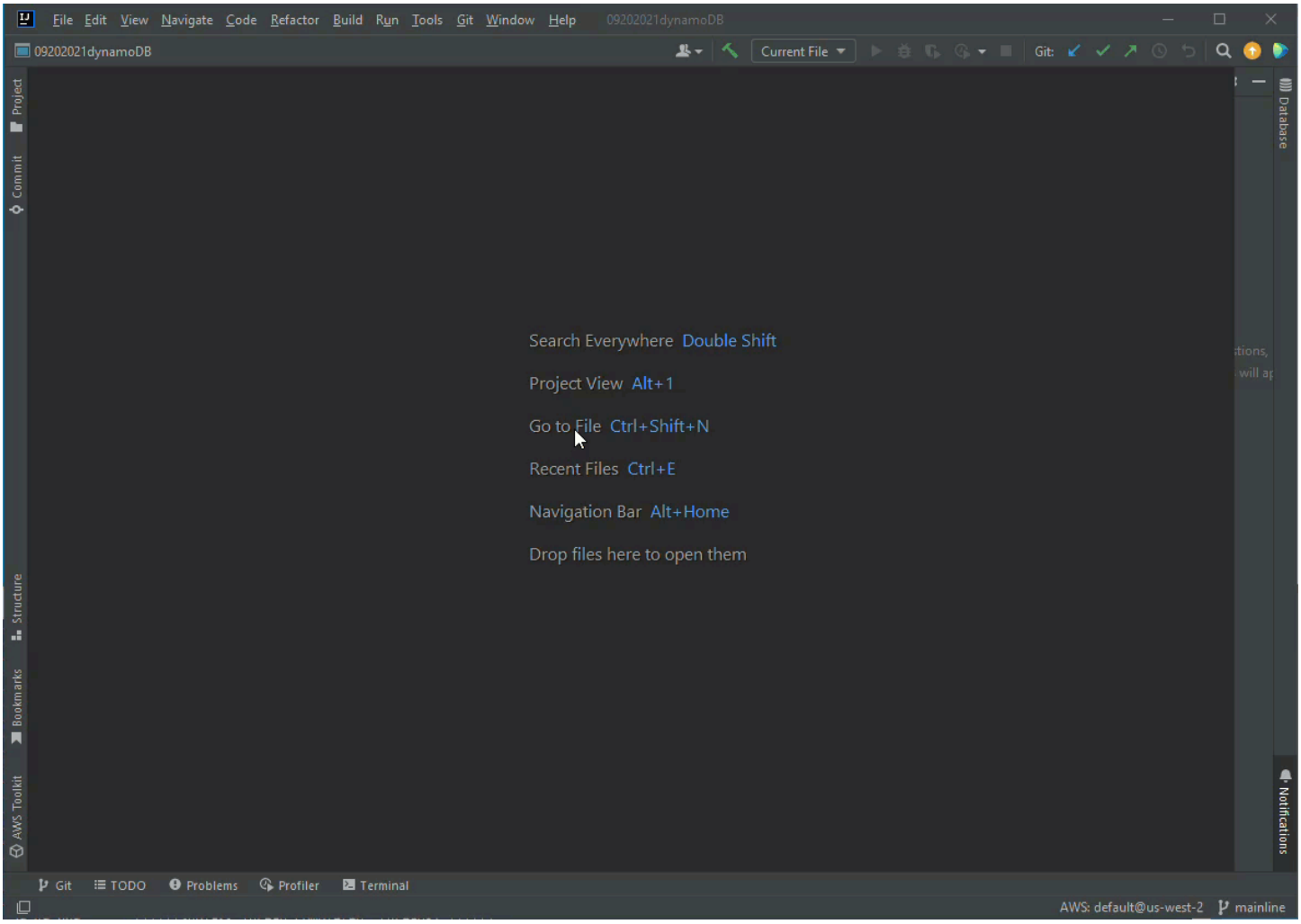
AWS 탐색기

AWS 서비스 및 리소스는 AWS Toolkit for JetBrains Explorer를 통해 사용할 수 있습니다.

> **Note**
>
> 인증을 설정하고 AWS 계정에 연결한 후에만 AWS Explorer에서 AWS 서비스 및 리소스를 볼 수 있습니다.
> 인증과 AWS Toolkit for JetBrains에 대한 추가 정보는 이 사용 설명서의 인증 및 액세스 목차를 참조하세요.
> AWS Toolkit for JetBrains에서 AWS 계정에 연결하는 방법에 대한 자세한 내용은 이 사용 설명서의 AWS에 연결 주제를 참조하세요.

AWS Toolkit for JetBrains Explorer에서 AWS 서비스 및 리소스 보기

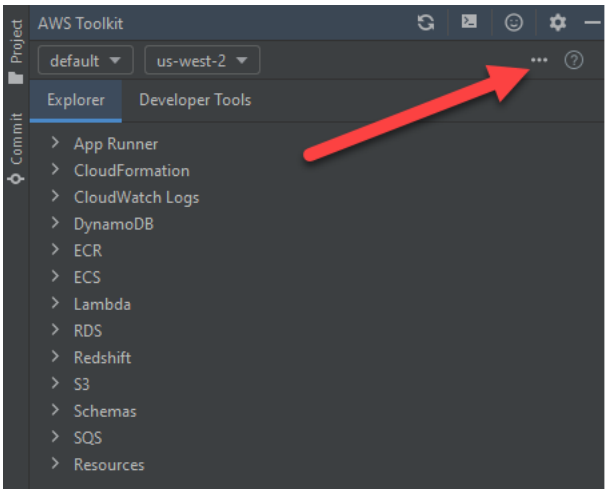
1. AWS Toolkit for JetBrains에서 Explorer 탭을 선택하여 계정 및 리전과 연결된 AWS 서비스를 봅니다.
2. 서비스를 선택하여 리소스 목록을 확장합니다.
3. 리소스의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 열어 리소스 수정을 위한 기능 목록을 봅니다.



AWS 연결

연결 설정은 AWS Toolkit for JetBrains 연결 창의 줄임표 아이콘 메뉴에 있습니다.

> **Note**
>
> AWS Toolkit for JetBrains에서 AWS 계정에 연결하는 방법에 대한 자세한 내용은 이 사용 설명서의 AWS에 연결 주제를 참조하세요.



AWS 계정에 AWS Toolkit for JetBrains 연결

대부분의 Amazon Web Services(AWS) 및 리소스는 AWS 계정을 통해 관리됩니다. AWS Toolkit for JetBrains를 사용하는 데는 AWS 계정이 필요하지 않지만, 연결하지 않으면 도구 키트의 기능이 제한됩니다.

이전에 AWS 계정을 설정하고 다른 AWS 서비스(예: AWS Command Line Interface)를 통해 인증한 경우 AWS Toolkit for JetBrains가 자동으로 보안 인증을 감지하고 연결 프로세스를 안내합니다.

AWS를 처음 사용하거나 계정을 생성하지 않은 경우 AWS Toolkit for JetBrains를 AWS 계정과 연결하는 3가지 주요 단계가 있습니다.

1. AWS 계정에 가입: AWS 가입 포털에서 AWS 계정에 가입할 수 있습니다. 새 AWS 계정 설정에 대한 자세한 내용은 AWS Setup User Guide의 Overview 주제를 참조하세요.
2. 인증 설정: AWS Toolkit for JetBrains에서 AWS 계정으로 인증하는 3가지 기본 방법이 있습니다. 이러한 각 방법에 대해 자세히 알아보려면 이 사용 설명서의 인증 및 액세스 주제를 참조하세요.
3. AWS Toolkit for JetBrains에서 AWS 계정으로 인증: AWS 계정을 생성하고 인증을 설정한 후에는 다음 섹션에 나와 있는 AWS Toolkit for JetBrains에서 AWS에 연결 절차를 완료하여 AWS Toolkit for JetBrains를 AWS 계정과 연결할 수 있습니다.

AWS Toolkit for JetBrains에서 AWS에 연결

AWS Toolkit for JetBrains에서 기존 IAM Identity Center 보안 인증을 사용하여 AWS 계정으로 인증하려면 다음 단계를 완료하세요.

> **Note**
>
> 이 프로세스는 기본 웹 브라우저에서 AWS IAM Identity Center 포털을 시작합니다. 보안 인증이 만료될 때마다 이 프로세스를 반복하여 AWS 계정과 AWS Toolkit for JetBrains 사이의 연결을 갱신해야 합니다.

1. AWS Toolkit for JetBrains에서 ...(줄임표) 아이콘을 선택하여 AWS 연결 설정 메뉴를 확장합니다.
2. AWS 연결 설정 메뉴에서 새 연결 추가...를 선택하여 AWS 도구 키트: 연결 추가 대화 상자를 엽니다.
3. AWS 도구 키트: 연결 추가 대화 상자에서 인증하려는 연결 옵션을 선택하고 연결을 선택하여 계속합니다.

   > **Note**
   >
   > AWS 계정의 인증 방법 구성에 대한 AWS Toolkit for JetBrains 관련 정보는 이 사용 설명서의 인증 및 액세스 주제를 참조하세요.

4. 인증 방법을 선택한 후 화면의 지시에 따라 AWS Toolkit for JetBrains를 AWS 계정과 연결하면 설정 프로세스 완료 시 JetBrains에서 알림을 보냅니다.

AWS Toolkit for JetBrains에 대한 AWS 리전 설정

AWS 리전에 따라 AWS 리소스가 관리되는 위치가 지정됩니다. AWS Toolkit for JetBrains에서 AWS 계정에 연결하면 기본 AWS 리전이 감지되고 AWS Explorer에 자동으로 표시됩니다.

다음 섹션에서는 AWS Toolkit for JetBrains Explorer에서 리전을 보고 변경하는 방법을 설명합니다.

현재 AWS 리전 보기

현재 선택된 AWS 리전을 확인하려면 다음 단계를 완료하세요.

1. AWS Explorer에서 설정 아이콘을 선택하여 옵션 메뉴 표시를 엽니다.

2. 옵션 표시 메뉴에서 AWS 연결 설정을 확장하여 계정에 사용할 수 있는 AWS 리전 목록을 표시합니다.
3. 현재 AWS 리전에는 리전 이름 옆에 확인 표시 아이콘이 표시됩니다.

AWS 리전 변경

현재 AWS 리전을 변경하려면 다음 단계를 완료하세요.

1. AWS Explorer에서 설정 아이콘을 선택하여 옵션 메뉴 표시를 엽니다.
2. 옵션 표시 메뉴에서 AWS 연결 설정을 확장하여 AWS 리전 목록을 표시합니다.
3. 목록에서 연결하려는 AWS 리전을 선택합니다.

> **Note**
>
> 연결하려는 리전이 표시되지 않으면 모든 리전을 선택하여 모든 AWS 리전의 전체 목록을 엽니다.

AWS Toolkit for JetBrains에 대한 HTTP 프록시 설정

AWS Toolkit for JetBrains에 대한 HTTP 프록시 설정은 JetBrains 통합 개발 환경(IDE)을 통해 처리됩니다. HTTP 프록시를 설정하는 방법에 대해 자세히 알아보려면 다음 목록에서 JetBrains IDE를 선택하세요.

- CLion – CLion 도움말 웹 사이트의 Configure HTTP proxy를 참조하세요.
- GoLand – GoLand 도움말 웹 사이트의 HTTP Proxy를 참조하세요.
- IntelliJ IDEA – IntelliJ IDEA 도움말 웹 사이트의 HTTP Proxy를 참조하세요.
- WebStorm – WebStorm 도움말 웹 사이트의 HTTP Proxy를 참조하세요.
- JetBrains Rider – JetBrains Rider 도움말 웹 사이트의 Configure HTTP Proxy를 참조하세요.
- PhpStorm – PhpStorm 도움말 웹 사이트의 HTTP Proxy를 참조하세요.
- PyCharm – PyCharm 도움말 웹 사이트의 HTTP Proxy를 참조하세요.
- RubyMine – RubyMine 도움말 웹 사이트의 HTTP Proxy를 참조하세요.

AWS Toolkit for JetBrains에 대한 인증 및 액세스

AWS Toolkit for JetBrains 작업을 시작하기 위해 AWS로 인증할 필요는 없습니다. 그러나 대부분의 AWS 리소스는 AWS 계정을 통해 관리됩니다. 모든 AWS Toolkit for JetBrains 서비스 및 기능에 액세스하려면 최소 2가지 유형의 계정 인증이 필요합니다.

1. AWS 계정에 대한 AWS Identity and Access Management(IAM) 또는 AWS IAM Identity Center 인증. 대부분의 AWS 서비스 및 리소스는 IAM 및 IAM Identity Center를 통해 관리됩니다.
2. AWS Builder ID는 다른 특정 AWS 서비스에 대한 선택 사항입니다.

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

주제

- AWS IAM Identity Center
- AWS IAM 보안 인증
- 개발자용 AWS Builder ID

AWS IAM Identity Center

AWS IAM Identity Center는 AWS 계정 인증 관리에 권장되는 모범 사례입니다.

AWS Toolkit for JetBrains와 소프트웨어 개발 키트(SDK)용 IAM Identity Center를 설정하는 방법에 대한 자세한 지침은 AWS SDKs and Tools Reference Guide의 IAM Identity Center authentication 섹션을 참조하세요.

AWS Toolkit for JetBrains에서 IAM Identity Center로 로그인

AWS Toolkit for JetBrains에서 기존 IAM Identity Center 보안 인증을 사용하여 AWS 계정으로 인증하려면 다음 단계를 완료하세요.

> **Note**
>
> 이 프로세스는 기본 웹 브라우저에서 AWS IAM Identity Center 포털을 시작합니다. 보안 인증이 만료될 때마다 이 프로세스를 반복하여 AWS 계정과 AWS Toolkit for JetBrains 사이의 연결을 갱신해야 합니다.

1. AWS Toolkit for JetBrains에서 ...(줄임표) 아이콘을 선택하여 AWS 연결 설정을 엽니다.
2. AWS 연결 설정 메뉴에서 새 연결 추가...를 선택하여 AWS 도구 키트: 연결 추가 대화 상자를 엽니다.
3. AWS Toolkit: 연결 추가 대화 상자에서 AWS IAM Identity Center를 사용하여 연결 라디오 버튼을 선택하고 시작 URL: 텍스트 필드에 IAM Identity Center 포털 URL을 입력한 다음 연결을 선택하여 계속합니다.
4. 메시지가 표시되면 기본 웹 브라우저에서 IAM Identity Center 포털 열기를 확인하고 프롬프트에 따라 인증 프로세스를 완료합니다. 인증 프로세스가 완료되면 알림을 받게 되며 브라우저 창을 닫아도 안전합니다.

AWS IAM 보안 인증

AWS IAM 사용자 보안 인증은 로컬에 저장된 액세스 키를 통해 AWS 계정으로 인증됩니다.

다음 섹션에서는 IAM 사용자 보안 인증을 통해 AWS 계정을 인증하도록 AWS Toolkit for JetBrains를 구성하는 방법을 설명합니다.

> **Important**
>
> AWS 계정으로 인증하도록 IAM 보안 인증을 설정하기 전에 다음 사항에 유의하세요.
>
> - 다른 AWS 서비스(예: AWS CLI)를 통해 이미 IAM 보안 인증을 설정한 경우 AWS Toolkit for JetBrains는 자동으로 해당 보안 인증을 감지하여 사용할 수 있게 합니다.
> - AWS에서는 IAM Identity Center 인증 사용을 권장합니다. AWS IAM 모범 사례에 대한 추가 정보는 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. 대신 AWS IAM Identity Center 사용 설명서의 What is IAM Identity Center?와 같은 ID 제공업체와의 페더레이션을 사용합니다.

필수 조건

IAM 사용자 보안 인증으로 인증하도록 AWS Toolkit for JetBrains를 구성하려면 먼저 다음 사전 조건을 충족해야 합니다. AWS Command Line Interface 등의 다른 서비스를 통해 IAM 사용자 보안 인증을 이미 설정한 경우 사전 조건 단계를 건너뛰고 다음 섹션으로 진행할 수 있습니다.

1. IAM 사용자를 생성합니다. IAM 사용자를 생성하는 방법에 대한 자세한 지침은 AWS SDKs and Tools Reference Guide의 Step 1: Create your IAM user를 참조하세요.
2. IAM 사용자 액세스 키를 받습니다. IAM 사용자 액세스 키를 받는 방법에 대한 자세한 지침은 AWS SDKs and Tools Reference Guide의 Step 2: Get your access keys를 참조하세요.
3. 선택 사항: 공유 보안 인증 파일을 업데이트합니다. 공유 보안 인증 파일을 업데이트하는 방법에 대한 자세한 지침은 AWS SDKs and Tools Reference Guide의 Step 3: Update the shared credentials file을 참조하세요.

> **Note**
>
> 선택적 사전 조건인 Step 3: Update the shared credentials file이 완료되면 AWS Toolkit for JetBrains가 다음 섹션에 설명된 AWS Toolkit for JetBrains에서 공유 보안 인증 파일 생성 중 보안 인증을 자동으로 감지합니다.

AWS Toolkit for JetBrains에서 공유 보안 인증 파일 생성

공유 구성 파일과 공유 보안 인증 파일에는 AWS 계정에 대한 구성 및 보안 인증 정보가 저장됩니다. 공유 구성 및 보안 인증에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 구성 설정이 저장되는 장소 섹션을 참조하세요.

AWS Toolkit for JetBrains에서 공유 보안 인증 파일 생성

1. AWS Toolkit for JetBrains에서 + AWS에 연결 추가를 선택하여 AWS Toolkit: 연결 추가 대화 상자를 엽니다.
2. AWS Toolkit: 연결 추가 대화 상자에서 AWS 보안 인증 파일 편집을 선택하여 보안 인증 파일 생성 확인을 엽니다.
3. 보안 인증 파일 생성 확인에서 생성을 선택하여 확인을 닫고 `credential File`을 생성합니다.
4. 생성이 완료되면 `credential File`이 IDE에서 자동으로 열립니다.

> **Note**
>
> `Credential File` 생성 프로세스 중
>
> - 오류가 발생하면 JetBrains에서 알림을 표시하고 오류 세부 정보가 포함된 생성 로그를 엽니다.

- 명명된 모든 프로파일을 단일 파일에 저장할 수 있습니다. 보안 인증과 구성 파일을 모두 사용하는 경우 기본적으로 IDE에서 보안 인증이 열립니다.
- 동일한 이름을 공유하는 프로파일에 대한 보안 인증이 두 파일 모두에 있는 경우 보안 인증 파일의 키가 우선합니다.

공유 보안 인증 구성

AWS 계정으로 AWS Toolkit for JetBrains를 인증하는 마지막 절차는 보안 인증을 구성하는 것입니다.

1. AWS Toolkit for JetBrains에서 + AWS에 연결 추가를 선택하여 AWS Toolkit: 연결 추가 대화 상자를 엽니다.
2. AWS Toolkit: 연결 추가 대화 상자에서 AWS 보안 인증 파일 편집을 선택하여 보안 인증 파일을 엽니다.
3. JetBrains에서 `credentials file`이 열리면 `[default]`라고 표시된 섹션을 찾습니다.
4. `[default]` 섹션에서 `#aws_access_key_id =` 항목을 찾아 #을 제거하고 AWS 액세스 키를 입력합니다. 항목은 다음과 유사합니다.

   `aws_access_key_id = AKIAIOSFODNN7EXAMPLE`
5. `[default]` 섹션에서 `#aws_secret_access_key =` 항목을 찾아 #을 제거하고 AWS 비밀 액세스 키를 입력합니다. 항목은 다음과 유사합니다.

   `aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`

   업데이트된 보안 인증 파일의 최종 버전은 다음과 유사합니다.

```
[default]
# The access key and secret key pair identify your account and grant access to AWS.
# Treat your secret key like a password. Never share your secret key with anyone.
 Do
# not post it in online forums, or store it in a source control system. If your
 secret
# key is ever disclosed, immediately use IAM to delete the access key and secret
 key
# and create a new key pair. Then, update this file with the replacement key
 details.
aws_access_key_id = AKIAIOSFODNN7EXAMPLE
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

6. 변경 사항을 파일에 저장하면 AWS Toolkit for JetBrains가 자동으로 업데이트된 보안 인증을 감지하고 AWS 계정에 연결합니다.

```
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

개발자용 AWS Builder ID

AWS Builder ID는 특정 AWS 서비스에 선택 사항이거나 필수인 AWS 계정입니다. 특정 서비스에 대해 AWS Builder ID를 구성하는 방법에 대한 자세한 내용은 이 설명서의 the section called “AWS Builder ID 서비스” 섹션을 참조하세요.

다음 섹션에서는 AWS Toolkit for JetBrains에서 AWS를 생성하고 인증하는 방법을 설명합니다.

AWS Builder ID 설정

AWS Toolkit for JetBrains에서 AWS Builder ID 설정

1. AWS Explorer,에서 + AWS에 연결 추가를 선택하여 AWS Toolkit: 연결 추가 대화 상자를 엽니다.
2. AWS Toolkit: 연결 추가 대화 상자에서 개인 이메일을 사용하여 가입 및 AWS Builder ID로 로그인을 선택하여 AWS Builder ID로 로그인 대화 상자를 엽니다.
3. AWS Builder ID로 로그인 대화 상자에서 열기 및 코드 복사 버튼을 선택하여 기본 웹 브라우저에서 AWS 인증 요청 사이트를 엽니다.
4. 웹 브라우저에서 제공된 필드에 확인 코드를 붙여넣고 다음을 선택하여 AWS Builder ID 생성 사이트로 진행합니다.
5. 웹 브라우저에서 각 단계를 완료하여 계속합니다. 프로세스가 완료되면 브라우저를 닫고 JetBrains로 돌아가도 안전하다는 알림이 표시됩니다.
6. JetBrains에서 + AWS에 연결 추가 드롭다운이 업데이트되어 AWS Builder ID에 연결되었음을 나타냅니다.

AWS Builder ID 서비스

특정 서비스를 AWS Builder ID에 연결하려면 추가 구성이 필요할 수 있습니다. 다음은 AWS Toolkit for JetBrains를 통해 액세스할 수 있는 AWS Builder ID 호환 서비스입니다.

- Amazon CodeCatalyst: AWS Builder ID에 대해 Amazon CodeCatalyst를 설정하는 방법에 대한 자세한 내용은 Amazon CodeCatalyst User Guide의 Setting up Amazon CodeCatalyst 섹션을 참조하세요.
- Amazon CodeWhisperer: AWS Builder ID에 대해 Amazon Code Whisper를 설정하는 방법에 대한 자세한 내용은 Amazon CodeWhisperer User Guide의 Setting up Amazon CodeWhisperer for individual developers를 참조하세요.

AWS Toolkit Explorer에서 AWS 서비스 작업

AWS 서비스 및 리소스는 AWS Toolkit for JetBrains Explorer를 통해 사용할 수 있습니다.

AWS Toolkit for JetBrains와 AWS Explorer를 탐색하는 방법에 대한 자세한 내용은 이 사용 설명서의 탐색 주제를 참조하세요.

AWS Toolkit for JetBrains의 특정 AWS 서비스 작업에 대해 자세히 알아보려면 다음 주제 목록에서 선택하세요.

주제

실험 기능 작업

실험 기능을 통해 공식 출시 전 AWS Toolkit for JetBrains 기능에 조기에 액세스할 수 있습니다.

> **Warning**
>
> 실험 기능은 계속 테스트되고 업데이트되므로 가용성 문제가 있을 수 있습니다. AWS Toolkit for JetBrains에서 예고 없이 삭제될 수도 있습니다.

JetBrains IDE 설정 창의 AWS 섹션에서 특정 AWS 서비스에 대한 실험 기능을 활성화할 수 있습니다.

1. JetBrains에서 AWS 설정을 편집하려면 파일, 설정을 선택하거나 Ctrl+Alt+S를 누릅니다.
2. 설정 창에서 도구를 확장하고 AWS, 실험 기능을 선택합니다.
3. 출시 전에 액세스하려는 실험 기능의 확인란을 선택합니다. 실험 기능을 끄려면 관련 확인란 선택을 취소합니다.
4. 실험 기능을 활성화한 후 AWS Explorer를 열고 옵션(기어 아이콘), 실험 기능을 선택하여 확인할 수 있습니다. 기능 이름 옆의 확인 표시는 해당 기능을 사용할 수 있음을 나타냅니다.

AWS App Runner와 함께 AWS Toolkit for JetBrains 사용

AWS App Runner는 소스 코드 또는 컨테이너 이미지에서 AWS 클라우드의 확장 가능하고 안전한 웹 애플리케이션으로 직접 배포하는 빠르고 간단하며 비용 효율적인 방법을 제공합니다. 이를 사용하면 새로운 기술을 배우거나 사용할 컴퓨팅 서비스를 결정하거나 AWS 리소스를 프로비저닝 및 구성하는 방법을 알 필요가 없습니다.

소스 이미지 또는 소스 코드를 기반으로 서비스를 생성하고 관리하는 데 AWS App Runner를 사용할 수 있습니다. 소스 이미지를 사용하는 경우 이미지 리포지토리에 저장된 퍼블릭 또는 프라이빗 컨테이너 이미지를 선택할 수 있습니다. App Runner는 다음 이미지 저장소 제공자를 지원합니다.

- Amazon Elastic Container Registry(Amazon ECR): AWS 계정에 프라이빗 이미지를 저장합니다.
- Amazon Elastic Container Registry 퍼블릭(Amazon ECR 퍼블릭): 공개적으로 읽을 수 있는 이미지를 저장합니다.

소스 코드 옵션을 선택하면 지원되는 리포지토리 공급자가 유지 관리하는 소스 코드 리포지토리에서 배포할 수 있습니다. 현재 App Runner는 GitHub를 소스 코드 리포지토리 제공업체로 지원합니다.

필수 조건

이 섹션에서는 사용자에게 이미 AWS 계정과 AWS App Runner를 제공하는 최신 버전의 AWS Toolkit for JetBrains가 있다고 가정합니다. 이러한 핵심 요구 사항 외에도 모든 관련 IAM 사용자에게 App

Runner 서비스와 상호 작용할 수 있는 권한이 있는지 확인하십시오. 또한 컨테이너 이미지 URI 또는 GitHub 리포지토리에 대한 연결과 같은 서비스 소스에 대한 특정 정보를 얻어야 합니다. App Runner 서비스를 생성할 때 이 정보가 필요합니다.

App Runner에 대한 IAM 권한 구성

App Runner에 필요한 권한을 부여하는 가장 쉬운 방법은 기존 AWS 관리형 정책을 관련 IAM 엔터티, 특히 사용자 또는 그룹에 연결하는 것입니다. App Runner는 IAM 사용자에게 연결할 수 있는 2개의 관리형 정책을 제공합니다.

- `AWSAppRunnerFullAccess`: 사용자가 모든 App Runner 작업을 수행할 수 있도록 허용합니다.
- `AWSAppRunnerReadOnlyAccess`: 사용자가 App Runner 리소스에 대한 세부 정보를 나열하고 볼 수 있도록 허용합니다.

또한 Amazon Elastic Container Registry(Amazon ECR)에서 프라이빗 리포지토리를 서비스 소스로 선택하는 경우 App Runner 서비스에 대해 다음 액세스 역할을 생성해야 합니다.

- `AWSAppRunnerServicePolicyForECRAccess`: App Runner가 계정에서 Amazon Elastic Container Registry(Amazon ECR) 이미지에 액세스할 수 있도록 허용합니다.

App Runner 서비스 생성 대화 상자를 사용하여 이 IAM 역할을 생성할 수 있습니다.

> **Note**
>
> AWSServiceRoleForCloudWatchCrossAccount 서비스 연결 역할을 통해 AWS App Runner가 다음 작업을 완료할 수 있습니다.
>
> - Amazon CloudWatch Logs에 로그 그룹에 로그를 푸시합니다.
> - Amazon Elastic Container Registry(Amazon ECR) 이미지 푸시를 구독하는 Amazon CloudWatch Events 규칙을 생성합니다.
>
> 서비스 연결 역할을 수동으로 생성할 필요가 없습니다. AWS Toolkit for JetBrains에서 직접적으로 호출하는 API 작업을 사용하여 또는 AWS Management Console에서 AWS App Runner를 생성할 때 AWS App Runner에서 이 서비스 연결 역할을 생성합니다.

자세한 내용은 AWS App Runner 개발자 안내서에서 App Runner의 자격 증명 및 액세스 관리를 참조하세요.

App Runner에 대한 서비스 소스 얻기

AWS App Runner를 사용하여 소스 이미지 또는 소스 코드에서 서비스를 배포할 수 있습니다.

Source image

소스 이미지에서 배포하는 경우 프라이빗 또는 퍼블릭 AWS 이미지 레지스트리에서 해당 이미지의 리포지토리에 대한 링크를 얻을 수 있습니다.

- Amazon ECR 프라이빗 레지스트리: https://console.aws.amazon.com/ecr/repositories에서 Amazon ECR 콘솔을 사용하는 프라이빗 리포지토리의 URI를 복사합니다.
- Amazon ECR 퍼블릭 레지스트리: https://gallery.ecr.aws/에서 Amazon ECR 퍼블릭 갤러리를 사용하는 퍼블릭 리포지토리의 URI를 복사합니다.

App Runner 서비스 생성 대화 상자에서 소스에 대한 세부 정보를 입력할 때 이미지 리포지토리의 URI를 지정합니다.

자세한 내용은 AWS App Runner 개발자 안내서의 소스 이미지 기반의 App Runner 서비스를 참조하세요.

Source code

소스 코드를 AWS App Runner 서비스에 배포하려면 지원되는 리포지토리 공급자가 유지 관리하는 Git 리포지토리에 해당 코드를 저장해야 합니다. App Runner는 하나의 소스 코드 리포지토리 공급자인 GitHub를 지원합니다.

GitHub 리포지토리를 설정하는 방법에 대한 자세한 내용은 GitHub의 시작하기 설명서를 참조하세요.

GitHub 리포지토리에서 App Runner 서비스에 소스 코드를 배포하기 위해 App Runner는 GitHub에 대한 연결을 설정합니다. 리포지토리가 프라이빗인 경우(즉, GitHub에서 공개적으로 액세스할 수 없는 경우) App Runner에 연결 세부 정보를 제공해야 합니다.

> **Important**
>
> GitHub 연결을 생성하려면 App Runner 콘솔(https://console.aws.amazon.com/apprunner)을 사용하여 GitHub를 AWS에 연결하는 연결을 생성해야 합니다. App Runner 서비스 생성 대화 상자를 사용하여 소스 코드 리포지토리에 대한 세부 정보를 지정할 때 GitHub 연결 페이지에서 사용 가능한 연결을 선택할 수 있습니다.
> 자세한 내용은 AWS App Runner 개발자 안내서의 App Runner 연결 관리를 참조하세요.

App Runner 서비스 인스턴스는 코드를 빌드하고 실행할 수 있는 관리형 런타임을 제공합니다. AWS App Runner에서는 현재 다음과 같은 런타임을 지원합니다.

- Python 관리형 런타임
- Node.js 관리형 런타임

AWS Toolkit for JetBrains에서 사용 가능한 App Runner 서비스 생성 대화 상자를 사용하여 App Runner 서비스가 서비스를 구축하고 시작하는 방법에 대한 정보를 제공합니다. 인터페이스에 직접 정보를 입력하거나 YAML 형식의 App Runner 구성 파일을 지정할 수 있습니다. 이 파일의 값은 App Runner에 서비스를 빌드 및 시작하고 런타임 컨텍스트를 제공하는 방법을 지시합니다. 여기에는 관련 네트워크 설정 및 환경 변수가 포함됩니다. 구성 파일의 이름이 `apprunner.yaml`로 지정되었습니다.. 애플리케이션 리포지토리의 루트 디렉토리에 자동으로 추가됩니다.

요금

애플리케이션에서 사용하는 컴퓨팅 및 메모리 리소스에 대한 요금이 청구됩니다. 또한 배포를 자동화하는 경우 해당 월의 모든 자동화된 배포를 포함하는 각 애플리케이션에 대해 설정된 월별 요금도 지불합니다. 소스 코드에서 배포하기로 선택한 경우 App Runner가 소스 코드에서 컨테이너를 빌드하는 데 걸리는 시간만큼 빌드 비용을 추가로 지불합니다.

자세한 내용은 AWS App Runner 요금을 참조하세요.

주제

- App Runner 서비스 생성
- App Runner 서비스 관리

App Runner 서비스 생성

App Runner 서비스 생성 대화 상자를 사용하여 AWS Toolkit for JetBrains에서 App Runner 서비스를 생성할 수 있습니다. App Runner 서비스의 인터페이스를 사용하여 소스 리포지토리를 선택하고 애플리케이션이 실행되는 서비스 인스턴스를 구성할 수 있습니다.

App Runner 서비스를 생성하기 전에 사전 조건을 모두 완료했는지 확인합니다. 여기에는 관련 IAM 권한을 제공하고 배포하려는 소스 리포지토리에 대한 특정 정보를 기록하는 작업이 포함됩니다.

App Runner 서비스 생성

1. AWS Explorer가 열려 있지 않은 경우 이를 엽니다.
2. App Runner 노드를 마우스 오른쪽 버튼으로 클릭하고 서비스 생성(Create Service)을 선택합니다.

   App Runner 서비스 생성 대화 상자가 표시됩니다.
3. 고유한 서비스 이름을 입력합니다.
4. 소스 유형(ECR, ECR 퍼블릭 또는 소스 코드 리포지토리)을 선택하고 관련 설정을 구성합니다.

   ECR/ECR public

   프라이빗 레지스트리를 사용하는 경우 배포 유형을 선택합니다.

   - 수동: 서비스에 대한 각 배포를 명시적으로 시작하려면 수동 배포를 사용합니다.
   - 자동: 서비스에 대한 지속적 통합 및 배포(CI/CD) 동작을 구현하려면 자동 배포를 사용합니다. 이 옵션을 선택하면 이미지 리포지토리로 새 이미지 버전을 푸시하거나 코드 리포지토리로 새 커밋을 푸시할 때마다 App Runner가 이를 자동으로 서비스에 배포합니다. 이때 사용자의 추가 작업이 필요하지 않습니다.

   컨테이너 이미지 URI에 Amazon ECR 프라이빗 레지스트리 또는 Amazon ECR 퍼블릭 갤러리에서 복사한 이미지 리포지토리의 URI를 입력합니다.

   시작 명령에 서비스 프로세스를 시작하는 명령을 입력합니다.

   포트에 서비스에서 사용하는 IP 포트를 입력합니다.

   Amazon ECR 프라이빗 레지스트리를 사용하는 경우 필요한 ECR 액세스 역할을 선택하고 생성을 선택합니다.

   - IAM 역할 생성 대화 상자에는 IAM 역할에 대한 이름, 관리형 정책 및 신뢰 관계가 표시됩니다. 생성을 선택합니다.

   Source code repository

   배포 유형을 선택합니다.

   - 수동: 서비스에 대한 각 배포를 명시적으로 시작하려면 수동 배포를 사용합니다.

- 자동: 서비스에 대한 지속적 통합 및 배포(CI/CD) 동작을 구현하려면 자동 배포를 사용합니다. 이 옵션을 선택하면 이미지 리포지토리로 새 이미지 버전을 푸시하거나 코드 리포지토리로 새 커밋을 푸시할 때마다 App Runner가 이를 자동으로 서비스에 배포합니다. 이때 사용자의 추가 작업이 필요하지 않습니다.

연결에서 GitHub 연결 페이지의 목록에서 사용 가능한 연결을 선택합니다.

리포지토리 URL에 GitHub에서 호스팅되는 원격 리포지토리에 대한 링크를 입력합니다.

분기에서 배포할 소스 코드의 Git 분기를 지정합니다.

구성에서 런타임 구성을 지정하는 방식을 지정합니다.

- 여기에서 모든 설정 구성: 애플리케이션의 런타임 환경에 대해 다음 설정을 지정하려면 이 옵션을 선택합니다.
  - 런타임(Runtime): Python 3 또는 Nodejs 12를 선택합니다.
  - 포트: 서비스에서 사용하는 IP 포트를 입력합니다.
  - 빌드 명령(Build command): 서비스 인스턴스의 런타임 환경에서 애플리케이션을 빌드하는 명령을 입력합니다.
  - 시작 명령(Start command): 서비스 인스턴스의 런타임 환경에서 애플리케이션을 시작하는 명령을 입력합니다.
- 여기에 구성 파일 설정 제공: `apprunner.yaml` 구성 파일에 정의된 설정을 사용하려면 이 옵션을 선택합니다. 이 파일은 애플리케이션 리포지토리의 루트 디렉터리에 있습니다.

5. 값을 지정하여 App Runner 서비스 인스턴스의 런타임 구성을 정의합니다.

   - CPU: App Runner 서비스의 각 인스턴스용으로 예약된 CPU 단위 수입니다(기본값: 1 vCPU).
   - 메모리: App Runner 서비스의 각 인스턴스용으로 예약된 메모리 양입니다(기본값: 2 GB).
   - 환경 변수: 서비스 인스턴스의 동작을 사용자 지정하는 데 사용하는 선택적 환경 변수입니다. 키와 값을 정의하여 환경 변수를 생성합니다.

6. 생성 선택

   서비스가 생성되면 상태가 작업 진행 중에서 실행 중으로 바뀝니다.

7. 서비스 실행을 시작한 후 서비스를 마우스 오른쪽 버튼으로 클릭하고 서비스 URL 복사(Copy Service URL)를 선택합니다.

8. 배포된 애플리케이션에 액세스하려면 복사한 URL을 웹 브라우저의 주소 표시줄에 붙여넣습니다.

App Runner 서비스 관리

App Runner 서비스를 생성한 후 AWS Explorer 창을 사용하여 다음 활동을 수행하여 이를 관리할 수 있습니다.

- App Runner 서비스 일시 중지 및 다시 시작
- App Runner 서비스 배포
- App Runner에 대한 로그 스트림 보기
- App Runner 서비스 삭제

App Runner 서비스 일시 중지 및 다시 시작

웹 애플리케이션을 일시적으로 비활성화하고 코드 실행을 중지해야 하는 경우 AWS App Runner 서비스를 일시 중지할 수 있습니다. App Runner는 서비스에 대한 컴퓨팅 용량을 0으로 줄입니다. 애플리케이션을 다시 실행할 준비가 되면 App Runner 서비스를 다시 시작합니다. App Runner는 새로운 컴퓨팅 파워를 프로비저닝하고, 애플리케이션을 배포한 후 애플리케이션을 실행합니다.

> **Important**
>
> App Runner가 실행 중일 때만 요금이 청구됩니다. 따라서 비용을 관리하는 데 필요한 경우 애플리케이션을 일시 중지했다가 다시 시작할 수 있습니다. 이는 개발 및 테스트 시나리오에서 특히 유용합니다.

App Runner 서비스 일시 중지

1. AWS Explorer가 열려 있지 않은 경우 이를 엽니다.
2. App Runner를 확장하여 서비스 목록을 봅니다.
3. 서비스를 마우스 오른쪽 버튼으로 클릭하고 Pause(일시 중지)를 선택합니다.
4. 표시되는 대화 상자에서 일시 중지를 선택합니다.

   서비스가 일시 중지되는 동안 서비스 상태는 실행 중에서 작업 진행 중으로 바뀐 다음 일시 중지됨으로 바뀝니다.

App Runner 서비스 다시 시작

1. AWS Explorer가 열려 있지 않은 경우 이를 엽니다.

2. App Runner를 확장하여 서비스 목록을 봅니다.
3. 서비스를 마우스 오른쪽 버튼으로 클릭하고 다시 시작(Resume)을 선택합니다.
4. 표시되는 대화 상자에서 재개를 선택합니다.

   서비스가 재개되는 동안 서비스 상태는 일시 중지됨에서 작업 진행 중으로 바뀐 다음 실행 중으로 바뀝니다.

App Runner 서비스 배포

서비스에 대한 수동 배포 옵션을 선택하는 경우 서비스에 대한 각 배포를 명시적으로 시작해야 합니다.

1. AWS Explorer가 열려 있지 않은 경우 이를 엽니다.
2. App Runner를 확장하여 서비스 목록을 봅니다.
3. 서비스를 마우스 오른쪽 버튼으로 클릭하고 배포를 선택합니다.
4. 애플리케이션이 배포되는 동안 서비스 상태가 작업 진행 중에서 실행 중으로 바뀝니다.
5. 애플리케이션이 성공적으로 배포되었는지 확인하려면 동일한 서비스를 마우스 오른쪽 버튼으로 클릭하고 서비스 URL 복사(Copy Service URL)를 선택합니다.
6. 배포한 웹 애플리케이션에 액세스하려면 복사한 URL을 웹 브라우저의 주소 표시줄에 붙여넣습니다.

App Runner에 대한 로그 스트림 보기

CloudWatch Logs를 사용하여 App Runner와 같은 서비스에 대한 로그 파일을 모니터링, 저장 및 액세스할 수 있습니다. CloudWatch Logs는 로그 이벤트와 로그 스트림이라는 두 가지 유형의 로그 파일을 기록합니다. 로그 이벤트는 CloudWatch Logs로 모니터링하는 애플리케이션 또는 리소스에 의해 기록된 활동 레코드입니다. 로그 스트림은 동일한 소스를 공유하는 로그 이벤트 시퀀스입니다.

App Runner에 대해 다음 두 가지 유형의 로그 스트림에 액세스할 수 있습니다.

- 서비스 로그 스트림: App Runner에서 생성된 로깅 출력을 포함합니다. 이 유형의 로그 스트림에서 로그 이벤트는 App Runner가 서비스를 관리하고 이에 대한 조치를 취하는 방법에 대한 기록입니다.
- 애플리케이션 로그 스트림: 실행 중인 애플리케이션 코드의 출력을 포함합니다.

1. App Runner를 확장하여 서비스 목록을 봅니다.
2. 서비스를 마우스 오른쪽 버튼으로 클릭하고 다음 옵션 중 하나를 선택합니다.

- 서비스 로그 스트림 보기
- 애플리케이션 로그 스트림 보기

로그 스트림 창에는 로그 스트림을 구성하는 로그 이벤트가 표시됩니다.

3. 특정 이벤트에 대한 자세한 정보를 보려면 해당 이벤트를 마우스 오른쪽 버튼으로 클릭하고 로그 스트림 내보내기, 편집기에서 열기 또는 로그 스트림 내보내기, 파일에 저장을 선택합니다.

App Runner 서비스 삭제

> **Important**
>
> App Runner 서비스를 삭제하면 영구적으로 제거되고 저장된 데이터가 삭제됩니다. 서비스를 다시 생성해야 하는 경우 App Runner는 소스를 다시 가져와 코드 리포지토리인 경우 빌드해야 합니다. 웹 애플리케이션은 새로운 App Runner 도메인을 가져옵니다.

1. AWS Explorer가 열려 있지 않은 경우 이를 엽니다.
2. App Runner를 확장하여 서비스 목록을 봅니다.
3. 서비스를 마우스 오른쪽 단추로 클릭하고 서비스 삭제(Delete Service)를 선택합니다.
4. 확인 대화 상자에 delete me를 입력한 다음 확인을 선택합니다.

삭제된 서비스는 작업 진행 중 상태로 표시된 후 목록에서 사라집니다.

Amazon CodeCatalyst for JetBrains

Amazon CodeCatalyst란?

Amazon CodeCatalyst는 소프트웨어 개발 팀을 위한 클라우드 기반 협업 공간입니다. AWS Toolkit for JetBrains Gateway를 사용하면 JetBrains Gateway에서 직접 CodeCatalyst 리소스를 보고 관리할 수 있습니다. 또한 도구 키트를 사용하여 개발 환경 가상 컴퓨팅 환경을 시작, 관리, 편집할 수 있습니다. Codecatalyst에 대한 자세한 내용은 Amazon CodeCatalyst 사용 설명서를 참조하세요.

다음 주제에서는 AWS Toolkit for JetBrains Gateway를 CodeCatalyst와 연결하는 방법과 JetBrains Gateway를 통해 CodeCatalyst 작업을 수행하는 방법을 설명합니다.

주제

CodeCatalyst 및 AWS Toolkit for JetBrains 시작하기

JetBrains Gateway에서 CodeCatalyst 작업을 시작하려면 다음을 완료하세요.

주제

JetBrains Gateway 설치

AWS Toolkit를 CodeCatalyst 계정과 통합하기 전에 최신 버전의 JetBrains Gateway를 사용하고 있는지 확인합니다. 최신 버전의 JetBrains Gateway를 다운로드하려면 다음 링크에서 원하는 JetBrains Gateway 배포판을 선택합니다.

- JetBrains Gateway for Linux
- JetBrains Gateway for Windows
- JetBrains Gateway for macOS
- JetBrains Gateway for macOS Apple Silicon

AWS Toolkit for JetBrains Gateway 설치

JetBrains를 CodeCatalyst 계정과 연결하려면 최신 버전의 도구 키트 확장을 설치해야 합니다. JetBrains Plugins Marketplace에서 직접 최신 버전을 찾아 도구 키트 설치를 완료할 수 있습니다.

JetBrains Plugins Marketplace에서 AWS Toolkit 플러그인을 설치하려면 다음 단계를 완료하세요.

1. JetBrains Gateway 기본 화면에서 애플리케이션 왼쪽 하단에 있는 Settings/Preferences 아이콘을 선택합니다.
2. Settings/Preferences를 선택하여 Settings/Preferences 뷰를 엽니다.

3. Settings/Preferences 뷰에서 Plugins을 선택하여 Plugins 뷰를 엽니다.

> **Note**
>
> Marketplace 뷰 또는 Installed 뷰로 Plugins 뷰를 열 수 있습니다.
>
> - AWS Toolkit for JetBrains Gateway를 처음 설치하는 경우 Plugins Marketplace 뷰를 선택하여 계속합니다.
> - 이전 버전의 AWS Toolkit for JetBrains Gateway가 있는 경우 Installed 뷰에서 업데이트합니다.

4. Marketplace 뷰에서 `AWS Toolkit`를 입력하고 AWS Toolkit 플러그인 항목이 나타나면 선택합니다.
5. Install을 선택하여 AWS Toolkit for JetBrains Gateway를 다운로드하고 설치합니다.

> **Note**
>
> JetBrains Gateway에는 다운로드 및 설치 진행 상태가 표시됩니다. 도구 키트가 성공적으로 설치되면 JetBrains Gateway Connections 탐색기가 Amazon CodeCatalyst 플러그인 아이콘으로 업데이트됩니다.

CodeCatalyst 계정 및 AWS Builder ID 생성

JetBrains Gateway에 연결하려면 최신 버전의 AWS Toolkit for JetBrains 설치 외에도 활성 AWS Builder ID와 CodeCatalyst 계정이 필요합니다. 활성 AWS Builder ID 또는 CodeCatalyst 계정이 없는 경우 CodeCatalyst User Guide의 Setting up with CodeCatalyst 섹션을 참조하세요.

> **Note**
>
> AWS Builder ID는 AWS 보안 인증과 다릅니다. AWS 보안 인증은 AWS에서 액세스할 수 있는 대부분의 AWS 서비스에 필요합니다. AWS Builder ID는 새 CodeCatalyst 계정을 생성하고 기존 CodeCatalyst 계정에서 작업하는 데 필요합니다. 여기에는 AWS Toolkit에서 제공되는 모든 CodeCatalyst 기능을 사용한 작업이 포함됩니다.

CodeCatalyst와 JetBrains Gateway 연결

JetBrains Gateway를 CodeCatalyst 계정과 연결하려면 다음 단계를 완료하세요.

1. JetBrains Gateway Connections 탐색기에서 Amazon CodeCatalyst 플러그인을 선택하여 Amazon CodeCatalyst 플러그인 뷰를 엽니다.
2. CodeCatalyst 플러그인 뷰에서 Sign in with AWS Builder ID를 선택하여 AWS Login Required 프롬프트를 엽니다.
3. AWS Login Required 프롬프트에서 Open Browser를 선택하여 기본 웹 브라우저에서 CodeCatalyst 콘솔 로그인 화면을 엽니다.
4. 제공된 필드에 AWS Builder ID를 입력하고 지침에 따라 계속합니다.
5. 메시지가 표시되면 Allow를 선택하여 JetBrains와 CodeCatalyst 계정 간 연결을 확인하세요. 연결 프로세스가 완료되면 브라우저를 닫아도 안전하다는 확인 메시지가 표시됩니다.
6. JetBrains Gateway에서 CodeCatalyst 플러그인 뷰가 Dev Environments 뷰로 업데이트됩니다.

JetBrains Gateway에서 Amazon CodeCatalyst 작업

JetBrains에서 개발 환경이라는 가상 컴퓨팅 환경을 시작할 수 있습니다. 개발 환경은 스페이스의 여러 팀원 간에 복사하고 공유할 수 있는 사용자 지정 가능한 클라우드 개발 환경입니다. 개발 환경에 대한 자세한 내용과 CodeCatalyst에서 이에 액세스하는 방법은 Amazon CodeCatalyst User Guide의 Dev Environments 섹션을 참조하세요.

다음 섹션에서는 JetBrains Gateway에서 개발 환경을 생성하고 열고 작업하는 방법을 설명합니다.

주제

- 개발 환경 열기
- Dev Environment 생성
- 타사 리포지토리에서 개발 환경 생성
- 개발 환경 설정 구성
- 개발 환경 일시 중지
- Dev Environment 재개
- Dev Environment 삭제
- 개발 환경 기본값 구성

개발 환경 열기

JetBrains Gateway에서 기존 개발 환경을 열려면 다음 단계를 완료하세요.

1. Connections 탐색기에서 Amazon CodeCatalyst 플러그인을 선택합니다.
2. Remote Development 마법사 본문에서 열려는 개발 환경의 상위 스페이스와 프로젝트로 이동합니다.
3. 열려는 개발 환경을 선택합니다.
4. 개발 환경에 대한 열기 프로세스를 확인하여 계속합니다.

> **Note**
>
> 새 상태 창에 진행 상황이 표시됩니다. 열기 프로세스가 완료되면 개발 환경이 새 창에서 열립니다.

Dev Environment 생성

새 개발 환경 생성

1. Connections 탐색기에서 CodeCatalyst 플러그인을 선택합니다.
2. Remote Development 마법사 헤더 섹션에서 Create a Dev Environment 링크를 선택하여 New CodeCatalyst Dev Environment 뷰를 엽니다.
3. New CodeCatalyst Dev Environment 뷰에서 다음 필드를 사용하여 개발 환경 기본 설정을 구성합니다.
   - IDE: 개발 환경에서 실행할 기본 JetBrains IDE를 선택합니다.
   - CodeCatalyst Project: 개발 환경의 CodeCatalyst Space와 프로젝트를 선택합니다.
   - Dev Environment Alias: 개발 환경의 대체 이름을 입력합니다.
   - Compute: 개발 환경의 가상 하드웨어 구성을 선택합니다.
   - Persistent storage: 개발 환경의 영구 스토리지 양을 선택합니다.
   - Inactivity timeout: 개발 환경이 대기로 전환되기 전에 경과되는 시스템 유휴 시간을 선택합니다.
4. 개발 환경을 생성하려면 개발 환경 생성을 선택합니다.

> **Note**
>
> Create Dev Environment를 선택하면 New Dev Environment 뷰가 닫히고 개발 환경을 생성하는 프로세스가 시작됩니다. 이 프로세스는 몇 분 정도 걸릴 수 있으며 개발 환경이 생성될 때까지 다른 JetBrains Gateway 기능을 사용할 수 없습니다.
> 새 상태 창에 진행 상황이 표시됩니다. 프로세스가 완료되면 개발 환경이 새 창에서 열립니다.

타사 리포지토리에서 개발 환경 생성

리포지토리에 소스로 연결하여 타사 리포지토리에서 개발 환경을 생성할 수 있습니다.

타사 리포지토리에 소스로 연결은 CodeCatalyst의 프로젝트 수준에서 처리됩니다. 개발 환경에 타사 리포지토리를 연결하는 방법에 대한 지침과 추가 세부 정보는 Amazon CodeCatalyst User Guide의 Linking a source repository 주제를 참조하세요.

개발 환경 설정 구성

JetBrains Gateway에서 기존 개발 환경의 설정을 변경하려면 다음 단계를 완료하세요.

> **Note**
>
> 개발 환경이 생성되면 개발 환경에 할당한 스토리지 양을 수정할 수 없습니다.

1. Connections 탐색기에서 Amazon CodeCatalyst 플러그인을 선택합니다.
2. Remote Development 마법사 본문에서 구성하려는 개발 환경의 상위 스페이스와 프로젝트로 이동합니다.
3. 구성하려는 개발 환경 옆의 Settings 아이콘을 선택하여 Configure Dev Environment: 설정을 엽니다.
4. Configure Dev Environment: 설정 메뉴에서 다음 옵션을 변경하여 개발 환경을 구성합니다.
   - 개발 환경 별칭: 개발 환경의 대체 이름을 지정하는 선택적 필드입니다.
   - IDE: 개발 환경 내에서 시작하려는 JetBrains IDE를 선택합니다.
   - Compute: 개발 환경의 가상 하드웨어 구성을 선택합니다.

- Inactivity timeout: 개발 환경이 대기로 전환되기 전에 경과되는 시스템 유휴 시간을 선택합니다.

개발 환경 일시 중지

개발 환경 내 활동은 영구적으로 저장됩니다. 따라서 작업 손실 없이 개발 환경을 일시 중지하고 재개할 수 있습니다.

개발 환경을 일시 중지하려면 다음 단계를 완료하세요.

1. Connections 탐색기에서 Amazon CodeCatalyst 플러그인을 선택합니다.
2. Remote Development 마법사 본문에서 일시 중지하려는 개발 환경의 상위 스페이스와 프로젝트로 이동합니다.
3. 활성 개발 환경 옆의 Pause 아이콘을 선택하여 Confirm Pause 대화 상자를 엽니다.
4. Yes를 선택하여 Confirm Pause 대화 상자를 닫고 일시 중지 프로세스를 시작합니다.

> **Note**
>
> 새 상태 창에 일시 중지 프로세스의 진행 상황이 표시됩니다. 개발 환경이 중지되면 Pause 아이콘이 사용자 인터페이스에서 제거됩니다.

Dev Environment 재개

개발 환경 내 활동은 영구적으로 저장됩니다. 따라서 이전 작업의 손실 없이 일시 중지된 개발 환경을 재개할 수 있습니다.

일시 중지된 개발 환경을 재개하려면 다음 단계를 완료하세요.

1. Connections 탐색기에서 Amazon CodeCatalyst 플러그인을 선택합니다.
2. Remote Development 마법사 본문에서 재개하려는 개발 환경의 상위 스페이스와 프로젝트로 이동합니다.
3. 재개하려는 개발 환경을 선택합니다.

> **Note**
>
> 새 상태 창에 재개 프로세스의 진행 상황이 표시됩니다. 개발 환경이 재개되면 개발 환경의 Settings 아이콘 옆에 Pause 아이콘이 추가됩니다.

Dev Environment 삭제

개발 환경을 삭제하려면 다음 단계를 완료하세요.

1. Connections 탐색기에서 Amazon CodeCatalyst 플러그인을 선택합니다.
2. Remote Development 마법사 본문에서 삭제하려는 개발 환경의 상위 스페이스와 프로젝트로 이동합니다.
3. 개발 환경 옆의 X 아이콘 버튼을 선택하여 Confirm Deletion 대화 상자를 엽니다.
4. Yes를 선택하여 대화 상자를 닫고 개발 환경을 삭제합니다.

> **Important**
>
> Yes를 선택하면 개발 환경이 삭제되고 검색할 수 없습니다. 개발 환경을 삭제하기 전에 코드 변경 사항을 커밋하고 원본 소스 리포지토리에 푸시해야 합니다. 그렇지 않으면 저장하지 않은 변경 사항이 영구적으로 손실됩니다.
> 개발 환경이 삭제되면 Remote Development 마법사가 업데이트되고 개발 환경이 더 이상 리소스에 나열되지 않습니다.

개발 환경 기본값 구성

개발 환경의 `devfile`에서 개발 환경 기본 설정을 구성할 수 있습니다. `devfile` 사양은 YAML 문서에서 업데이트할 수 있는 개방형 표준입니다.

`devfile`을 정의하고 구성하는 방법에 대한 자세한 내용은 devfile.io를 참조하세요.

JetBrains Gateway Dev Environment 인스턴스에서 `devfile`을 열고 편집하려면 다음 단계를 완료하세요.

1. 활성 JetBrains Dev Environment의 Navigation bar에서 Amazon CodeCatalyst Dev Environment 노드를 확장하여 Backend Status Details 메뉴를 엽니다.
2. Configure Dev Environment 탭을 선택하고 Open Devfile을 선택하여 JetBrains Editor에서 `devfile`을 엽니다.
3. 편집기에서 `devfile`을 변경하고 작업을 저장합니다.
4. 변경 사항 저장 시 Amazon CodeCatalyst Dev Environment 노드가 개발 환경을 재구축해야 한다는 알림을 표시합니다.
5. Amazon CodeCatalyst Dev Environment 노드를 확장하고 Configure Dev Environment 탭에서 Rebuild Dev Environment 노드를 선택합니다.

AWS Toolkit for JetBrains를 사용하여 AWS CloudFormation 작업

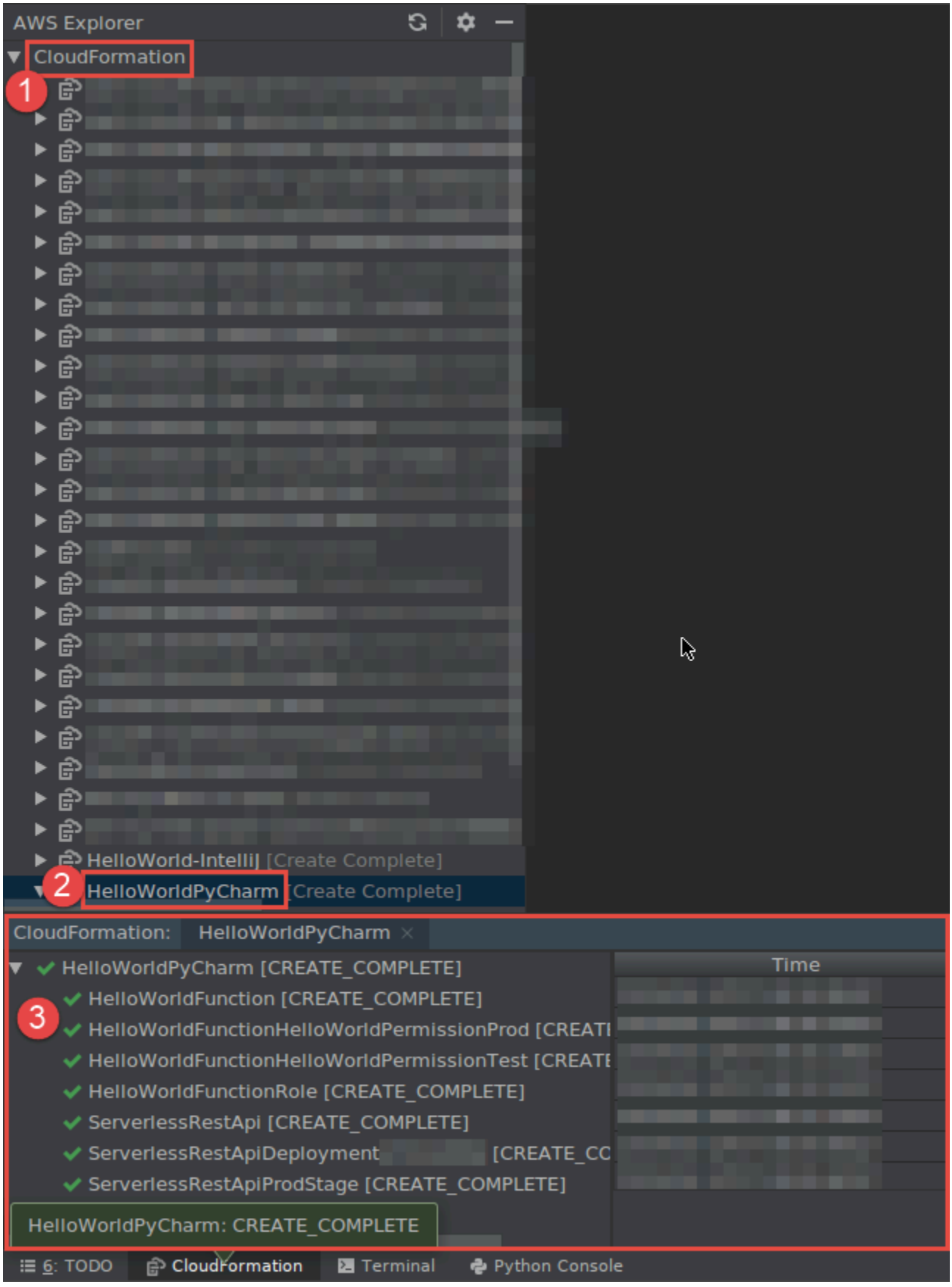
다음 주제에서는 AWS Toolkit for JetBrains을 사용하여 AWS 계정에서 AWS CloudFormation 스택 작업을 수행하는 방법을 설명합니다.

주제

AWS Toolkit for JetBrains를 사용하여 AWS CloudFormation 스택에 대한 이벤트 로그 보기

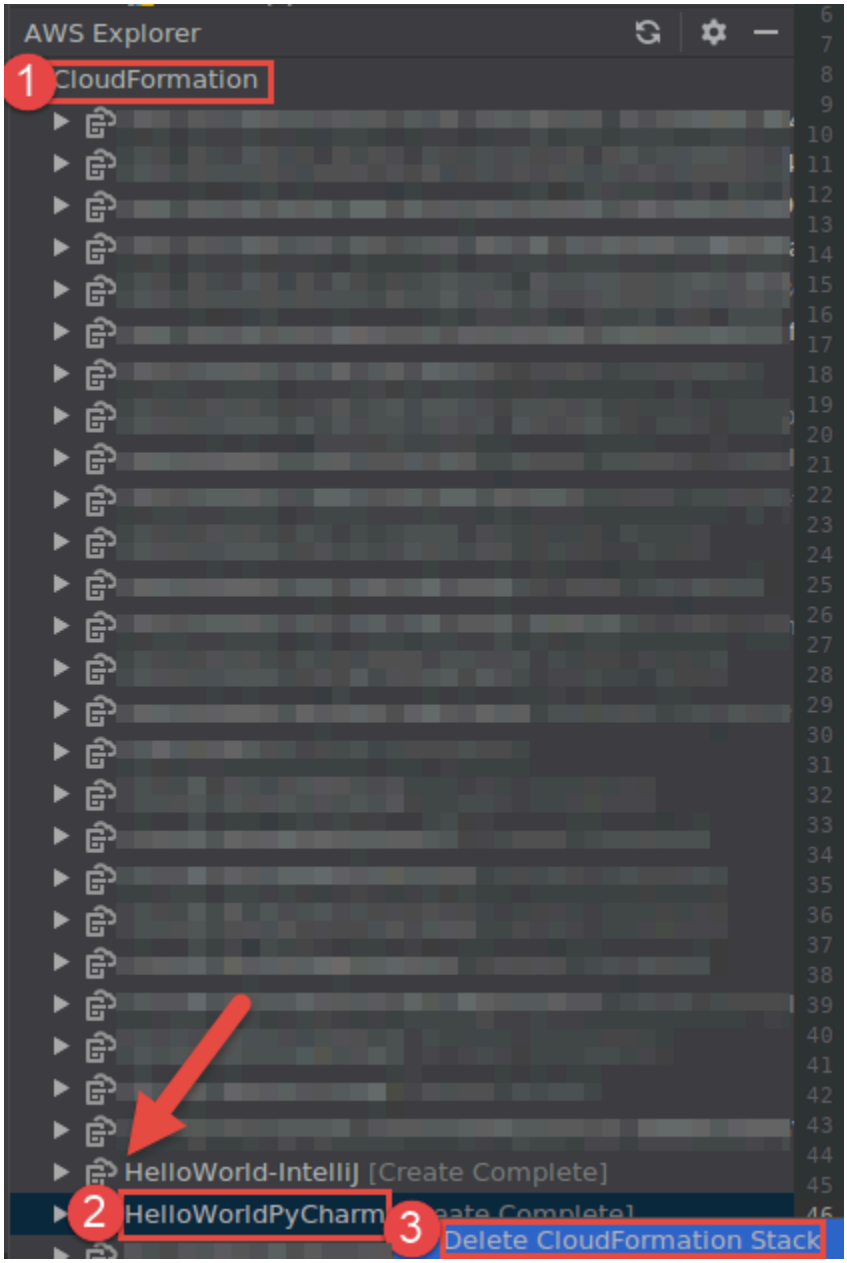
1. AWS Explorer가 열려 있지 않은 경우 이를 엽니다. 스택이 현재 리전과 다른 AWS 리전에 있는 경우 스택이 있는 다른 AWS 리전으로 전환합니다.
2. CloudFormation을 확장합니다.
3. 스택에 대한 이벤트 로그를 보려면 스택의 이름을 마우스 오른쪽 버튼으로 클릭합니다. AWS Toolkit for JetBrains은 CloudFormation 도구 창에 이벤트 로그를 표시합니다.

   CloudFormation 도구 창을 숨기거나 표시하려면 기본 메뉴에서 보기, 도구 창, CloudFormation을 선택합니다.


AWS Explorer
CloudFormation
1
HelloWorld-IntelliJ [Create Complete]
2
HelloWorldPyCharm [Create Complete]
CloudFormation: HelloWorldPyCharm
Time
HelloWorldPyCharm [CREATE_COMPLETE]
3
HelloWorldFunction [CREATE_COMPLETE]
HelloWorldFunctionHelloWorldPermissionProd [CREATI
HelloWorldFunctionHelloWorldPermissionTest [CREATE
HelloWorldFunctionRole [CREATE_COMPLETE]
ServerlessRestApi [CREATE_COMPLETE]
ServerlessRestApiDeployment [CREATE_CC
ServerlessRestApiProdStage [CREATE_COMPLETE]
HelloWorldPyCharm: CREATE_COMPLETE
6: TODO
CloudFormation
Terminal
Python Console

AWS Toolkit for JetBrains를 사용하여 AWS CloudFormation 스택 삭제

1. AWS Explorer가 열려 있지 않은 경우 이를 엽니다. 스택이 포함된 다른 AWS 리전으로 전환해야 하는 경우 지금 전환합니다.
2. CloudFormation을 확장합니다.
3. 삭제할 스택의 이름을 마우스 오른쪽 단추로 클릭한 다음 CloudFormation 스택 삭제를 선택합니다.



4. 스택의 이름을 입력하여 삭제되었는지 확인한 다음 확인을 선택합니다. 스택 삭제에 성공하면 AWS Toolkit for JetBrains가 AWS Explorer의 CloudFormation 목록에서 해당 스택 이름을 제거합니다. 스택 삭제에 실패하면 스택에 대한 이벤트 로그를 확인하여 문제를 해결할 수 있습니다.

AWS Toolkit for JetBrains를 사용하여 CloudWatch Logs 작업

Amazon CloudWatch Logs는 확장성이 뛰어난 단일 서비스에서 사용하는 모든 시스템, 애플리케이션 및 AWS 서비스에서 로그를 중앙 집중화할 수 있습니다. 그런 다음 로그를 쉽게 보고, 특정 오류 코드 또는 패턴이 있는지 검색하고, 특정 필드를 기반으로 필터링하거나, 향후 분석을 위해 안전하게 보관할 수 있습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서에서 Amazon CloudWatch Logs란 무엇입니까?를 참조하세요.

다음 주제에서는 AWS 계정에서 AWS Toolkit for JetBrains를 사용하여 CloudWatch Logs 작업을 수행하는 방법을 설명합니다.

주제

- AWS Toolkit for JetBrains를 사용하여 CloudWatch 로그 그룹 및 로그 스트림 보기
- AWS Toolkit for JetBrains를 사용하여 로그 스트림의 CloudWatch 로그 이벤트 작업
- AWS Toolkit for JetBrains를 사용하여 CloudWatch Logs Insights 작업

AWS Toolkit for JetBrains를 사용하여 CloudWatch 로그 그룹 및 로그 스트림 보기

로그 스트림은 동일한 소스를 공유하는 로그 이벤트 시퀀스입니다. CloudWatch Logs에서 각 별도의 로그 소스가 별도의 로그 스트림을 구성합니다.

로그 그룹은 동일한 보존 기간, 모니터링 및 액세스 제어 설정을 공유하는 로그 스트림의 그룹입니다. 로그 그룹을 정의하고 각 그룹에 배치할 스트림을 지정할 수 있습니다. 하나의 로그 그룹에서 포함할 수 있는 로그 스트림의 수에는 제한이 없습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서에서 로그 그룹 및 로그 스트림 작업을 참조하세요.

주제

- CloudWatch Logs 노드를 사용하여 로그 그룹 및 로그 스트림 보기
- Lambda 노드로 로그 스트림 보기
- Amazon ECS 노드로 로그 스트림 보기

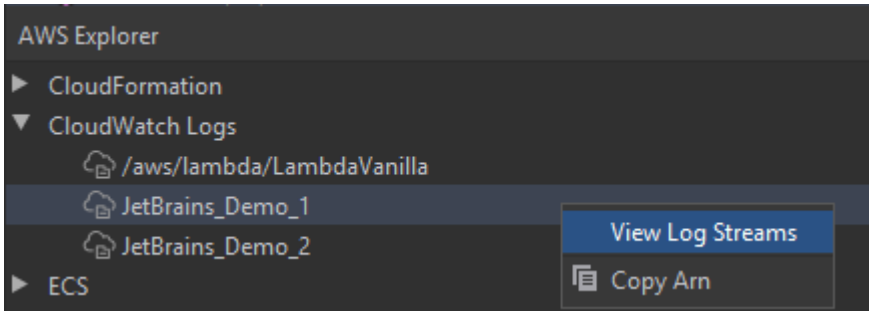
CloudWatch Logs 노드를 사용하여 로그 그룹 및 로그 스트림 보기

1. AWS Explorer가 열려 있지 않은 경우 이를 엽니다.
2. [CloudWatch Logs] 노드를 클릭하여 로그 그룹 목록을 확장합니다.

   현재 AWS 리전에 대한 로그 그룹이 CloudWatch Logs 노드 아래에 표시됩니다.
3. 로그 그룹의 로그 스트림을 보려면 다음 중 하나를 수행하세요.

   - 로그 그룹의 이름을 두 번 클릭합니다.
   - 로그 그룹의 이름을 마우스 오른쪽 버튼으로 클릭하고 로그 스트림 보기를 선택합니다.

   로그 그룹의 내용은 로그 스트림 창에 표시됩니다. 각 스트림의 로그 이벤트와 상호 작용하는 방법에 대한 자세한 내용은 CloudWatch 로그 이벤트 작업 섹션을 참조하세요.



Lambda 노드로 로그 스트림 보기

AWS Explorer에서 Lambda 노드를 사용하여 AWS Lambda 함수에 대한 CloudWatch Logs를 볼 수 있습니다.

> **참고**
>
> AWS Explorer에서 CloudWatch Logs 노드를 사용하여 Lambda 함수를 포함한 모든 AWS 서비스에 대한 로그 스트림을 볼 수도 있습니다. 그러나 Lambda 함수에 관련된 로그 데이터의 개요를 보려면 Lambda 노드를 사용하는 것이 좋습니다.

1. AWS Explorer가 열려 있지 않은 경우 이를 엽니다.
2. Lambda 노드를 클릭하여 Lambda 함수 목록을 확장합니다.

   현재 AWS 리전에 대한 Lambda 함수는 Lambda 노드 아래에 표시됩니다.

3. Lambda 함수를 마우스 오른쪽 버튼으로 클릭하고 로그 스트림 보기를 선택합니다.

   함수에 대한 로그 스트림이 로그 스트림 창에 표시됩니다. 각 스트림의 로그 이벤트와 상호 작용하는 방법에 대한 자세한 내용은 CloudWatch 로그 이벤트 작업 섹션을 참조하세요.

Amazon ECS 노드로 로그 스트림 보기

AWS Explorer에서 Amazon ECS 노드를 사용하여 Amazon Elastic Container Service에서 실행되고 유지 관리되는 클러스터와 컨테이너에 대한 CloudWatch Logs를 볼 수 있습니다.

> **참고**
>
> AWS Explorer에서 CloudWatch Logs 노드를 사용하여 Amazon ECS를 포함한 모든 AWS 서비스에 대한 로그 그룹을 볼 수도 있습니다. Amazon ECS 클러스터 및 컨테이너와 관련된 로그 데이터의 개요를 보려면 Amazon ECS 노드를 사용하는 것이 좋습니다.

1. AWS Explorer가 열려 있지 않은 경우 이를 엽니다.
2. Amazon ECS 노드를 클릭하여 Amazon ECS 클러스터의 목록을 확장합니다.

   현재 AWS 리전의 Amazon ECS 클러스터가 Amazon ECS 노드 아래에 표시됩니다.
3. 클러스터를 마우스 오른쪽 버튼으로 클릭하고 로그 스트림 보기를 선택합니다.

   클러스터에 대한 로그 스트림이 로그 스트림 창에 표시됩니다.
4. 특정 컨테이너에 대한 로그 스트림을 보려면 클러스터를 클릭하여 등록된 컨테이너 목록을 확장합니다.

   클러스터에 등록된 컨테이너가 아래에 표시됩니다.
5. 컨테이너를 마우스 오른쪽 버튼으로 클릭하고 컨테이너 로그 스트림 보기를 선택합니다.

   컨테이너에 대한 로그 스트림이 로그 스트림 창에 표시됩니다. 클러스터 및 컨테이너의 로그 이벤트와 상호 작용하는 방법에 대한 자세한 내용은 CloudWatch 로그 이벤트 작업 섹션을 참조하세요.

AWS Toolkit for JetBrains를 사용하여 로그 스트림의 CloudWatch 로그 이벤트 작업

로그 스트림 창을 연 후 각 스트림의 로그 이벤트에 액세스할 수 있습니다. 로그 이벤트는 모니터링 중인 애플리케이션 또는 리소스에 의해 기록된 활동의 기록입니다.

주제

스트림의 로그 이벤트 보기 및 필터링

로그 스트림을 열면 로그 스트림 창에 해당 스트림의 로그 이벤트 시퀀스가 표시됩니다.

1. 보려는 로그 스트림을 찾으려면 로그 스트림 창을 엽니다(CloudWatch 로그 그룹 및 로그 스트림 보기 참조).

   > **참고**
   >
   > 패턴 매칭을 사용하여 목록에서 스트림을 찾을 수 있습니다. 로그 스트림 창을 클릭하고 텍스트 입력을 시작합니다. 사용자가 입력한 텍스트와 일치하는 텍스트가 포함된 첫 번째 로그 스트림 이름이 강조 표시됩니다. 마지막 이벤트 시간 열의 상단을 클릭하여 목록을 재정렬할 수도 있습니다.

2. 로그 스트림을 두 번 클릭하여 로그 이벤트의 시퀀스를 봅니다.

   로그 이벤트 창에는 로그 스트림을 구성하는 로그 이벤트가 표시됩니다.

3. 콘텐츠에 따라 로그 이벤트를 필터링하려면 로그 스트림 필터링 필드에 텍스트를 입력하고 Return 키를 누릅니다.

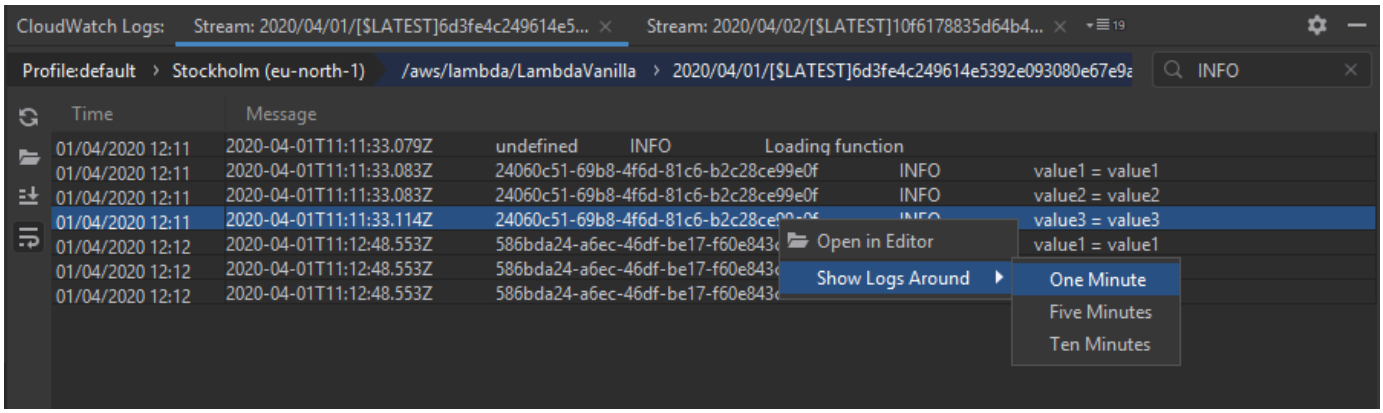
   결과는 필터 텍스트와 대/소문자를 구분하여 일치하는 텍스트가 포함된 로그 이벤트입니다. 필터는 화면에 표시되지 않는 이벤트를 포함하여 전체 로그 스트림을 검색합니다.

   > **Note**
   >
   > 패턴 일치를 사용하여 창에서 로그 이벤트를 찾을 수도 있습니다. 로그 이벤트 창을 클릭하고 텍스트 입력을 시작합니다. 사용자가 입력한 텍스트와 일치하는 텍스트가 포함된 첫

번째 로그 이벤트가 강조 표시됩니다. 로그 스트림 필터링 검색과 달리 화면의 이벤트만 확인됩니다.

4. 시간에 따라 로그 이벤트를 필터링하려면 로그 이벤트를 마우스 오른쪽 버튼으로 클릭하고 다음 시간 전후의 로그 표시를 선택합니다.

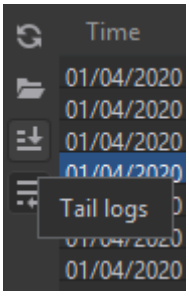
   1분, 5분 또는 10분을 선택할 수 있습니다. 예를 들어, 5분을 선택하면 선택한 항목 5분 전후에 발생한 로그 이벤트만 필터링된 목록에 표시됩니다.



로그 이벤트 창 왼쪽의 로그 작업은 로그 이벤트와 상호 작용하는 더 많은 방법을 제공합니다.

로그 작업

로그 이벤트 창 왼쪽의 4가지 로그 작업을 통해 CloudWatch 로그 이벤트를 새로 고치고, 편집하고, 테일링하고, 래핑할 수 있습니다.



1. 상호 작용할 로그 이벤트를 찾으려면 로그 스트림 창을 엽니다.
2. 다음 로그 작업 중 하나를 선택합니다.

   - 새로 고침 – 로그 이벤트 창이 열린 후 발생한 로그 이벤트로 목록을 업데이트합니다.
   - 편집기에서 열기 – IDE의 기본 편집기에서 화면 로그 이벤트를 엽니다.

> **Note**
>
> 이 작업은 화면의 로그 이벤트만 IDE 편집기로 내보냅니다. 편집기에서 모든 스트림의 이벤트를 보려면 로그 스트림 내보내기 옵션을 선택합니다.

- 테일 로그 - 로그 이벤트 창으로 새 로그 이벤트를 스트리밍합니다. 이 기능은 Amazon EC2 인스턴스 및 AWS CodeBuild 빌드와 같은 장기 실행 서비스를 지속적으로 업데이트하는 데 유용합니다.
- 로그 래핑 – 창 크기로 인해 긴 항목이 숨겨지는 경우 로그 이벤트 텍스트를 여러 줄로 표시합니다.

파일 또는 편집기로 CloudWatch 로그 이벤트 내보내기

CloudWatch 로그 스트림을 내보내면 로그 이벤트를 IDE의 기본 편집기에서 열거나 로컬 폴더에 다운로드할 수 있습니다.

1. 액세스하려는 로그 스트림을 찾으려면 로그 스트림 창을 엽니다.
2. 로그 스트림을 마우스 오른쪽 버튼으로 클릭한 다음 로그 스트림 내보내기, 편집기에서 열기 또는 로그 스트림 내보내기, 파일에 저장을 선택합니다.

   - 편집기에서 열기 – IDE의 기본 편집기에서 선택한 스트림을 구성하는 로그 이벤트를 엽니다.

   > **Note**
   >
   > 이 옵션은 로그 스트림의 모든 이벤트를 IDE 편집기로 내보냅니다.

   - 파일에 저장 - 로그 스트림 다운로드 대화 상자를 엽니다. 이 대화 상자에서 다운로드 폴더를 선택하고 로그 이벤트가 포함된 파일의 이름을 바꿀 수 있습니다.

AWS Toolkit for JetBrains를 사용하여 CloudWatch Logs Insights 작업

AWS Toolkit for JetBrains를 사용하여 CloudWatch Logs Insights 작업을 수행할 수 있습니다. CloudWatch Logs Insights를 사용하면 Amazon CloudWatch Logs 내 로그 데이터를 대화식으로 검색해 분석할 수 있습니다. 자세한 내용은 Amazon CloudWatch Logs 사용 설명서의 CloudWatch Logs Insights를 사용한 로그 분석을 참조하세요.

CloudWatch Logs Insights에 대한 IAM 권한

CloudWatch Logs Insights 쿼리 결과를 실행하고 보려면 다음 권한이 필요합니다.

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "logs:StartQuery",
        "logs:GetQueryResults",
        "logs:GetLogRecord",
        "logs:describeLogGroups",
        "logs:describeLogStreams"
      ],
      "Resource" : "*"
    }
  ]
}
```

다음 권한은 필수는 아니지만 연결된 결과 창이나 IDE를 닫을 때 AWS Toolkit for JetBrains가 현재 실행 중인 쿼리를 자동으로 중지할 수 있게 합니다.

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "logs:StopQuery"
      ],
      "Resource" : "*"
    }
  ]
}
```

CloudWatch Logs Insights 작업

CloudWatch Logs Insights 쿼리 편집기 열기

1. AWS Explorer를 엽니다.
2. CloudWatch Logs 노드를 두 번 클릭하여 로그 그룹 목록을 확장합니다.
3. 열려는 로그 그룹을 마우스 오른쪽 버튼으로 클릭한 다음 쿼리 편집기 열기를 선택합니다.

CloudWatch Logs Insights 쿼리 시작

1. 쿼리 로그 그룹 창에서 쿼리 파라미터를 원하는 대로 변경합니다.

   날짜 또는 상대 시간별로 시간 범위를 선택할 수 있습니다.

   쿼리 로그 그룹 필드는 CloudWatch Logs Insights 쿼리 구문을 허용합니다. 자세한 내용은 Amazon CloudWatch Logs 사용 설명서의 CloudWatch Logs Insights 쿼리 구문을 참조하세요.
2. 실행을 선택하여 쿼리를 시작합니다.

CloudWatch Logs Insights 쿼리 저장

1. 쿼리 이름을 입력합니다.
2. 쿼리 저장을 선택합니다.

   선택한 로그 그룹과 쿼리가 AWS 계정에 저장됩니다. 시간 범위는 저장되지 않습니다.

   CloudWatch Logs Insights AWS Management Console 페이지에서 저장된 쿼리를 검색하고 재사용할 수 있습니다.

저장된 CloudWatch Logs Insights 쿼리 검색

1. 쿼리 로그 그룹 창에서 저장된 쿼리 검색을 선택합니다.
2. 원하는 쿼리를 선택하고 확인을 선택합니다.

   선택한 로그 그룹과 쿼리가 기존 대화 상자의 모든 항목을 대체합니다.

쿼리 결과 탐색

- CloudWatch Logs Insights 쿼리 결과 창의 오른쪽 상단에서 쿼리 편집기 열기를 선택합니다.

개별 로그 레코드 보기

- 쿼리 결과 창에서 행을 두 번 클릭하여 해당 로그 레코드에 대한 세부 정보가 포함된 새 탭을 엽니다.

  오른쪽 상단에서 로그 스트림 보기를 선택하여 로그 레코드의 연결된 로그 스트림으로 이동할 수도 있습니다.

Amazon CodeWhisperer 작업

CodeWhisperer란 무엇입니까?

Amazon CodeWhisperer는 실시간으로 코드 권장 사항을 제공하는 범용 기계 학습 기반 코드 생성기입니다. 코드를 작성할 때 CodeWhisperer는 기존 코드 및 주석을 기반으로 제안을 자동으로 생성합니다. 맞춤형 권장 사항은 한 줄 주석부터 완전한 형태의 함수에 이르기까지 크기와 범위가 다양합니다.

CodeWhisperer는 코드를 스캔하여 보안 문제를 강조 표시하고 정의할 수도 있습니다.

CodeWhisperer는 다음 JetBrains 제품에 사용할 수 있습니다.

- IntelliJ IDEA
- PyCharm
- WebStorm
- Rider

자세한 정보는 Amazon CodeWhisperer 사용 설명서를 참조하세요.

AWS Toolkit for JetBrains의 Amazon DynamoDB

Amazon DynamoDB는 완전관리형 NoSQL 데이터베이스 서비스로서 원활한 확장성과 함께 예측 가능한 성능을 제공합니다. DynamoDB 서비스에 대한 자세한 내용은 Amazon DynamoDB 사용 설명서를 참조하세요.

다음 주제에서는 AWS Toolkit for JetBrains에서 DynamoDB 서비스 작업을 수행하는 방법을 설명합니다.

주제

- AWS Toolkit for JetBrains에서 Amazon DynamoDB 작업

AWS Toolkit for JetBrains에서 Amazon DynamoDB 작업

AWS Toolkit for JetBrains를 사용하면 JetBrains IDE에서 직접 Amazon 리소스 이름(ARN)을 보고, 복사하고, Amazon DynamoDB 리소스를 삭제할 수 있습니다.

다음 섹션에서는 AWS Toolkit for JetBrains에서 이러한 서비스 기능 작업을 수행하는 방법을 설명합니다.

DynamoDB 리소스 보기

현재 도구 키트에서 DynamoDB 리소스를 직접 생성할 수 없지만 리소스는 표시됩니다. DynamoDB 리소스를 보려면 다음 단계를 완료하세요.

1. AWS Toolkit for JetBrains의 Explorer 탭으로 이동합니다.
2. DynamoDB 노드를 확장합니다.
3. DynamoDB 노드 아래에 DynamoDB 리소스가 표시됩니다.

DynamoDB 리소스 ARN 복사

Amazon 리소스 이름(ARN)은 DynamoDB 테이블을 포함한 모든 AWS 리소스에 할당되는 고유 ID입니다. DynamoDB 리소스의 ARN ID를 복사하려면 다음 단계를 완료하세요.

1. AWS Toolkit for JetBrains의 Explorer 탭으로 이동합니다.
2. DynamoDB 노드를 확장합니다.
3. ARN ID를 복사할 DynamoDB 리소스에 대한 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 엽니다.
4. Copy ARN를 선택하여 리소스 ARN ID를 OS 클립보드에 복사합니다.

DynamoDB 리소스 삭제

DynamoDB 리소스를 삭제하려면 다음 단계를 완료하세요.

1. AWS Toolkit for JetBrains의 Explorer 탭으로 이동합니다.
2. DynamoDB 노드를 확장합니다.
3. 삭제할 DynamoDB 리소스에 대한 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 엽니다.

4. 테이블 삭제...를 선택하여 테이블 삭제... 확인 대화 상자를 엽니다.
5. 확인 지침을 완료하여 DynamoDB 테이블을 삭제합니다.

AWS Toolkit for JetBrains에서 Amazon DynamoDB 테이블 작업

Amazon DynamoDB의 기본 리소스는 데이터베이스 테이블입니다. 다음 섹션에서는 AWS Toolkit for JetBrains에서 DynamoDB 테이블 작업을 수행하는 방법을 설명합니다.

DynamoDB 테이블 보기

DynamoDB 테이블을 보려면 다음 단계를 완료하세요.

1. AWS Toolkit for JetBrains의 Explorer 탭으로 이동합니다.
2. DynamoDB 노드를 확장합니다.
3. DynamoDB 리소스 목록에서 테이블을 두 번 클릭하여 편집기 창에서 봅니다.

> **Note**
>
> 테이블 데이터를 처음 볼 때 최대 결과 제한이 50개 항목인 최초 스캔이 검색됩니다.

최대 결과 제한 설정

검색된 테이블 항목의 기본 제한을 변경하려면 다음 단계를 완료하세요.

1. AWS Explorer에서 테이블을 두 번 클릭하여 JetBrains 편집기 창에서 테이블을 봅니다.
2. 테이블 뷰에서 편집기 창의 오른쪽 상단에 있는 설정 아이콘을 선택합니다.
3. 최대 결과를 옵션을 마우스로 가리켜서 사용 가능한 최대 결과 값 목록을 봅니다.

DynamoDB 테이블 스캔

DynamoDB 테이블을 스캔하려면 다음 단계를 완료하세요.

> **Note**
>
> 이 스캔은 PartiQL 쿼리를 생성하며 올바른 AWS Identity and Access Management(AWS IAM) 정책이 마련되어 있어야 합니다. PartiQL 보안 정책 요구 사항에 대해 자세히 알아보려면

Amazon DynamoDB 개발자 안내서의 DynamoDB용 PartiQL을 사용하는 IAM 보안 정책 주제를 참조하세요.

1. AWS Explorer에서 테이블을 두 번 클릭하여 JetBrains 편집기 창에서 테이블을 봅니다.
2. 테이블 뷰에서 스캔 헤더를 확장합니다.
3. 드롭다운 메뉴에서 스캔하려는 테이블/인덱스를 선택합니다.
4. 실행을 선택하여 스캔을 진행합니다. 편집기 창에 테이블 데이터가 반환되면 스캔이 완료됩니다.

AWS Toolkit for JetBrains를 사용하여 Amazon Elastic Container Service 작업

다음 주제에서는 AWS Toolkit for JetBrains를 사용하여 AWS 계정에서 Amazon ECS 리소스 작업을 수행하는 방법을 설명합니다.

주제

- AWS Toolkit에서 Amazon Elastic Container Service(Amazon ECS) 사용

AWS Toolkit에서 Amazon Elastic Container Service(Amazon ECS) 사용

Amazon ECS Exec 기능을 사용하면 AWS Toolkit에서 직접 단일 명령을 실행하거나 Amazon Elastic Container Service(Amazon ECS) 컨테이너에서 쉘을 실행할 수 있습니다.

Important

Amazon ECS Exec을 활성화하고 비활성화하면 AWS 계정의 리소스 상태가 바뀝니다. 여기에는 서비스 중지 및 재시작이 포함됩니다. Amazon ECS Exec이 활성화된 상태에서 리소스 상태를 변경하면 예상치 못한 결과가 발생할 수 있습니다. Amazon ECS Exec에 대한 자세한 내용은 개발자 안내서의 Using Amazon ECS Exec for Debugging을 참조하세요.

Amazon ECS Exec 사전 조건

Amazon ECS Exec 기능을 사용하기 위한 사전 조건이 있습니다.

> **Important**
>
> 특정 서비스에 대해 Amazon ECS Exec을 활성화하려면 해당 서비스에 대해 Amazon ECS 클라우드 디버깅을 비활성화해야 합니다.

Amazon ECS 사전 조건

작업이 Amazon EC2에서 호스팅되는지 AWS Fargate에서 호스팅되는지에 따라 Amazon ECS Exec의 버전 요구 사항이 다릅니다.

- Amazon EC2를 사용하는 경우 2021년 1월 20일 이후에 출시된 Amazon ECS 최적화 AMI를 에이전트 버전 1.50.2 이상으로 사용해야 합니다. 추가 정보는 개발자 안내서의 Amazon ECS optimized AMIs에서 확인할 수 있습니다.
- AWS Fargate를 사용하는 경우 플랫폼 버전 1.4.0 이상을 사용해야 합니다. Fargate 요구 사항에 대한 추가 정보는 개발자 안내서의 AWS Fargate platform versions에서 확인할 수 있습니다.

AWS 계정 구성 및 IAM 권한

Amazon ECS Exec 기능을 사용하려면 AWS 계정과 연결된 기존 Amazon ECS 클러스터가 있어야 합니다. Amazon ECS Exec은 Systems Manager를 사용하여 클러스터의 컨테이너와의 연결을 설정하며 SSM 서비스와의 통신을 위한 특정 태스크 IAM 역할 권한을 필요로 합니다.

Amazon ECS Exec과 관련된 IAM 역할 및 정책 정보는 개발자 안내서의 IAM permissions required for ECS Exec에서 확인할 수 있습니다.

Amazon ECS Exec 작업

AWS Toolkit for JetBrains의 AWS Explorer에서 바로 Amazon ECS Exec을 활성화하거나 비활성화할 수 있습니다. Amazon ECS Exec이 활성화되면 Amazon ECS 메뉴에서 컨테이너를 선택하고 해당 컨테이너에 대해 명령을 실행할 수 있습니다.

Amazon ECS Exec 활성화

1. AWS Explorer에서 Amazon ECS 메뉴를 확장합니다.
2. 클러스터 섹션을 확장하고 수정하려는 클러스터를 선택합니다.
3. 수정하려는 서비스의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 열고 명령 실행 활성화를 선택합니다.

> **Note**
>
> 이 서비스에 대해 Amazon ECS 클라우드 디버깅이 활성화된 경우 명령 실행 활성화 옵션을 사용할 수 없습니다. 클라우드 디버깅을 비활성화하면 옵션은 복원되지만 서비스가 중지되었다가 다시 시작됩니다.

> **Important**
>
> 서비스의 새 배포가 시작되며 몇 분 정도 걸릴 수 있습니다. 자세한 내용은 이 섹션의 시작 부분에 나오는 참고를 참조하세요.

Amazon ECS Exec 비활성화

1. AWS Explorer에서 Amazon ECS 메뉴를 확장합니다.
2. 클러스터 섹션을 확장하고 수정하려는 클러스터를 선택합니다.
3. 수정하려는 서비스의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 열고 명령 실행 비활성화를 선택합니다.

> **Important**
>
> 서비스의 새 배포가 시작되며 몇 분 정도 걸릴 수 있습니다. 자세한 내용은 이 섹션의 시작 부분에 나오는 참고를 참조하세요.

컨테이너에 대한 명령 실행

AWS Explorer를 사용하여 컨테이너에 대해 명령을 실행하려면 Amazon ECS Exec을 활성화해야 합니다. 활성화되지 않은 경우 이 섹션의 Amazon ECS Exec 활성화 절차를 참조하세요.

1. AWS Explorer에서 Amazon ECS 메뉴를 확장합니다.
2. 클러스터 섹션을 확장하고 수정하려는 클러스터를 선택합니다.
3. 서비스를 확장하여 해당 컨테이너를 나열합니다.
4. 수정하려는 컨테이너에 대한 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 열고 컨테이너에서 명령 실행을 선택합니다.

5. 컨테이너에서 명령 실행 대화 상자에서 원하는 태스크 ARN을 선택합니다.
6. 실행하려는 명령을 입력하거나 동일한 세션 동안 실행된 명령 목록에서 명령을 선택할 수 있습니다.
7. 실행(Execute)을 선택합니다.

쉘 내에서 명령 실행

AWS Explorer를 사용하여 쉘 내에서 컨테이너에 대해 명령을 실행하려면 Amazon ECS Exec을 활성화해야 합니다. 활성화되지 않은 경우 이 섹션의 Amazon ECS Exec 활성화 절차를 참조하세요.

1. AWS Explorer에서 Amazon ECS 메뉴를 확장합니다.
2. 클러스터 섹션을 확장하고 수정하려는 클러스터를 선택합니다.
3. 서비스를 확장하여 해당 컨테이너를 나열합니다.
4. 수정하려는 컨테이너에 대한 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 열고 대화형 쉘 열기를 선택합니다.
5. 대화형 쉘 대화 상자에서 원하는 태스크 ARN을 선택합니다.
6. 해당 드롭다운에서 쉘을 선택하거나 상호 작용하려는 쉘의 이름을 입력합니다.
7. 설정에 만족하면 실행을 선택합니다.
8. 터미널에서 쉘이 열리면 명령을 입력하여 컨테이너와 상호 작용할 수 있습니다.

AWS Toolkit for JetBrains를 사용하여 Amazon EventBridge 작업

다음 주제에서는 AWS Toolkit for JetBrains를 사용하여 AWS 계정에서 Amazon EventBridge 스키마 작업을 수행하는 방법을 설명합니다.

주제

- Amazon EventBridge 스키마 작업

Amazon EventBridge 스키마 작업

다음과 같이 AWS Toolkit for JetBrains를 사용하여 Amazon EventBridge 스키마 작업을 수행할 수 있습니다.

> **Note**
>
> EventBridge 스키마 작업은 현재 AWS Toolkit for IntelliJ와 AWS Toolkit for PyCharm에서만 지원됩니다.

다음 정보는 이미 AWS Toolkit for JetBrains를 설정했다고 가정합니다.

목차

사용 가능한 스키마 보기

1. AWS 탐색기 도구 창을 표시하고 스키마를 확장합니다.
2. 보려는 스키마가 포함된 레지스트리의 이름을 확장합니다. 예를 들어, AWS에서 제공하는 대부분의 스키마는 aws.events 레지스트리에 있습니다.
3. 편집기에서 스키마를 보려면 스키마 제목을 마우스 오른쪽 버튼으로 클릭하고 컨텍스트 메뉴에서 스키마 보기를 선택합니다.

사용 가능한 스키마 찾기

AWS 탐색기 도구 창이 표시된 상태에서 다음 중 하나를 실행합니다.

- 찾으려는 스키마의 제목을 입력하기 시작합니다. AWS 탐색기는 일치 항목을 포함하는 스키마 제목을 강조 표시합니다.
- 스키마를 마우스 오른쪽 버튼으로 클릭하고 컨텍스트 메뉴에서 스키마 검색을 선택합니다. EventBridge 스키마 검색 대화 상자에서 찾으려는 스키마의 제목 입력을 시작합니다. 대화 상자에 일치 항목이 포함된 스키마 제목이 표시됩니다.
- 스키마를 확장합니다. 찾으려는 스키마가 포함된 레지스트리의 이름을 마우스 오른쪽 단추로 클릭한 다음 레지스트리에서 스키마 검색을 선택합니다. EventBridge 스키마 검색 대화 상자에서 찾으려는 스키마의 제목 입력을 시작합니다. 대화 상자에 일치 항목이 포함된 스키마 제목이 표시됩니다.

일치 항목 목록에서 스키마를 보려면 다음 중 하나를 수행합니다.

- 편집기에 스키마를 표시하려면 AWS 탐색기에서 스키마 제목을 마우스 오른쪽 단추로 클릭한 다음 스키마 보기를 선택합니다.
- EventBridge 스키마 검색 대화 상자에서 스키마를 표시할 스키마의 제목을 선택합니다.

사용 가능한 스키마에 대한 코드 생성

1. AWS 탐색기 도구 창을 표시하고 스키마를 확장합니다.
2. 코드를 생성할 스키마가 포함된 레지스트리의 이름을 확장합니다.
3. 스키마 제목을 마우스 오른쪽 버튼으로 클릭한 다음 Download code bindings(코드 바인딩 다운로드)를 선택합니다.
4. 코드 바인딩 다운로드 대화 상자에서 다음을 선택합니다.
   - 코드를 생성할 스키마의 버전입니다.
   - 코드를 생성할 지원 프로그래밍 언어와 언어 버전입니다.
   - 로컬 개발 시스템에 생성된 코드를 저장할 파일 위치입니다.
5. 다운로드를 선택합니다.

사용 가능한 스키마를 사용하는 AWS Serverless Application Model 애플리케이션 생성

1. File 메뉴에서 New와 Project를 차례대로 선택합니다.
2. 새 프로젝트 대화 상자에서 AWS을 선택합니다.
3. AWS 서버리스 애플리케이션를 선택한 다음를 선택합니다.
4. 다음을 지정합니다.
   - 프로젝트의 프로젝트 이름입니다.
   - 프로젝트에 대한 로컬 개발 시스템의 프로젝트 위치입니다.
   - 프로젝트에 지원되는 AWS Lambda 런타임 입니다.
   - 프로젝트에 대한 AWS Serverless Application Model(AWS SAM) SAM 템플릿입니다. 현재 선택 사항에는 다음이 포함됩니다.
     - AWS SAM EventBridge Hello World(EC2 인스턴스 상태 변경) – 배포되면 AWS 계정에 AWS Lambda 함수와 연결된 Amazon API Gateway 엔드포인트가 생성됩니다. 기본적으로 이 함수와 엔드포인트는 Amazon EC2 인스턴스 상태 변경에만 응답합니다.

- (스키마 레지스트리의 모든 이벤트 트리거에 대해) 새로 생성한 AWS SAM EventBridge 앱 – 배포되면 AWS 계정에 AWS Lambda 함수와 연결된 Amazon API Gateway 엔드포인트가 생성됩니다. 이 함수와 엔드포인트는 지정한 스키마에서 이용 가능한 이벤트에만 응답합니다.

  이 템플릿을 선택하는 경우 다음을 지정해야 합니다.

  - 사용할 명명된 프로파일, 자격 증명입니다.
  - 사용할 AWS 리전입니다.
  - 사용할 EventBridge 이벤트 스키마
- 프로젝트(프로젝트 SDK)에 사용할 SDK버전입니다.

AWS 서버리스 애플리케이션 프로젝트를 만든 후 다음을 수행할 수 있습니다.

- 애플리케이션 배포
- 애플리케이션의 설정 변경(업데이트)
- 배포된 애플리케이션 삭제

애플리케이션의 일부인 Lambda 함수를 사용하여 다음을 수행할 수도 있습니다.

- 함수의 로컬 버전을 실행(호출) 또는 디버깅
- 함수의 원격 버전 실행(호출)
- 함수의 설정 변경
- 함수 삭제

AWS Toolkit for JetBrains에서 AWS Lambda 작업

다음 주제에서는 AWS Toolkit for JetBrains에서 AWS Lambda 함수 작업을 수행하는 방법을 설명합니다.

주제

- AWS Toolkit for JetBrains의 AWS Lambda 런타임 및 지원
- AWS Toolkit for JetBrains를 사용하여 AWS Lambda 함수 생성
- AWS Toolkit for JetBrains를 사용하여 AWS Lambda 함수의 로컬 버전 실행(간접적으로 호출) 또는 디버깅
- AWS Toolkit for JetBrains를 사용하여 AWS Lambda 함수의 원격 버전 실행(간접적으로 호출)

- AWS Toolkit for JetBrains를 사용하여 AWS Lambda 함수 설정 변경(업데이트)
- AWS Toolkit for JetBrains를 사용하여 AWS Lambda 함수 삭제

AWS Toolkit for JetBrains의 AWS Lambda 런타임 및 지원

AWS Lambda는 런타임을 사용하여 여러 언어를 지원합니다. 런타임은 Lambda와 함수 간에 호출 이벤트, 컨텍스트 정보 및 응답을 릴레이하는 언어별 환경을 제공합니다. Lambda 서비스 및 지원되는 런타임에 대한 자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 런타임을 참조하세요.

다음은 현재 AWS Toolkit for JetBrains에 사용할 수 있는 런타임 환경에 대한 설명입니다.

이름	식별자	운영 체제	아키텍처
Node.js 18	nodejs18.x	Amazon Linux 2	x86_64, arm64
Node.js 16	nodejs16.x	Amazon Linux 2	x86_64, arm64
Node.js 14	nodejs14.x	Amazon Linux 2	x86_64, arm64
Python 3.11	python3.11	Amazon Linux 2	x86_64, arm64
Python 3.10	python3.10	Amazon Linux 2	x86_64, arm64
Python 3.9	python3.9	Amazon Linux 2	x86_64, arm64
Python 3.8	python3.8	Amazon Linux 2	x86_64, arm64
Python 3.7	python3.7	Amazon Linux 2	x86_64
Java 17	java17	Amazon Linux 2	x86_64, arm64
Java 11	java11	Amazon Linux 2	x86_64, arm64
Java 8	java8.al2	Amazon Linux 2	x86_64, arm64
Java 8	java8	Amazon Linux 2	x86_64
.NET 6	dotnet6	Amazon Linux 2	x86_64, arm64
Go 1.x	go1.x	Amazon Linux 2	x86_64

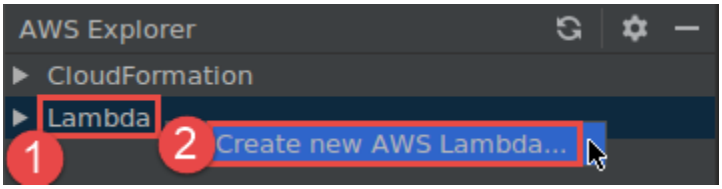
AWS Toolkit for JetBrains를 사용하여 AWS Lambda 함수 생성

AWS Toolkit for JetBrains를 사용하여 AWS 서버리스 애플리케이션의 일부인 AWS Lambda 함수를 만들 수 있습니다. 또는 독립 실행형 Lambda 함수를 생성할 수 있습니다.

AWS 서버리스 애플리케이션의 일부인 Lambda 함수를 생성하려면 이 주제의 나머지 부분을 건너뛰고 대신 애플리케이션 생성 섹션을 참조하세요.

독립 실행형 Lambda 함수를 생성하려면 먼저 AWS Toolkit for JetBrains를 설치하고 아직 설치하지 않은 경우 AWS 계정에 처음 연결해야 합니다. 그런 다음 IntelliJ IDEA, PyCharm, WebStorm 또는 JetBrains Rider가 이미 실행되고 있는 상태에서 다음 중 하나를 수행합니다.

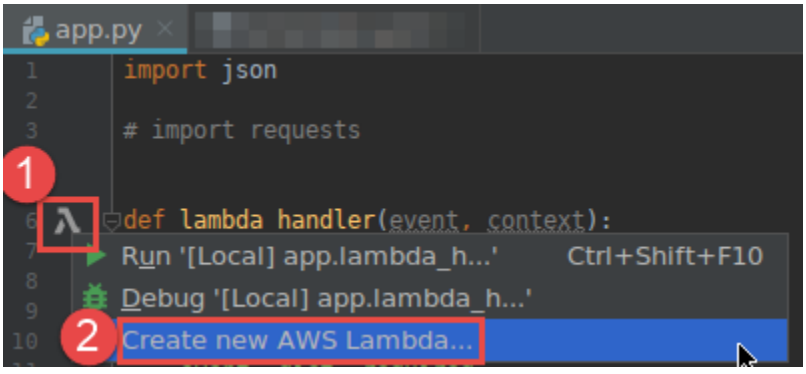
- AWS Explorer가 열려 있지 않은 경우 이를 엽니다. 함수를 생성하기 위해 다른 AWS 리전으로 변경해야 하는 경우 지금 수행하세요. 그런 다음 Lambda를 마우스 오른쪽 버튼으로 클릭하고 새 AWS Lambda 생성을 선택합니다.

  함수 만들기 대화 상자를 완료한 다음 함수 만들기를 선택합니다. AWS Toolkit for JetBrains는 배포를 위한 해당 AWS CloudFormation 스택을 생성하고 AWS Explorer의 Lambda 목록에 함수 이름을 추가합니다. 배포가 실패하면 스택에 대한 이벤트 로그를 확인하여 이유를 파악할 수 있습니다.

- Java, Python, Node.js, C#에 대한 함수 핸들러를 구현하는 코드 파일을 만듭니다.

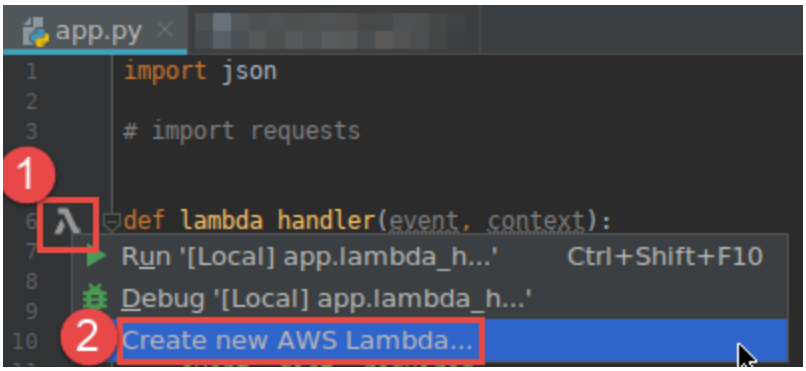
  실행(간접적으로 호출)할 원격 함수를 생성하기 위해 다른 AWS 리전으로 전환해야 하는 경우 지금 수행하세요. 그런 다음 코드 파일에서 함수 핸들러 옆의 거터에서 Lambda 아이콘을 선택한 다음 새 AWS Lambda 생성을 선택합니다. 함수 만들기 대화 상자를 완료한 다음 함수 만들기를 선택합니다.

> **Note**
>
> 함수 핸들러 옆의 거터에 Lambda 아이콘이 표시되지 않으면 설정/기본 설정: 도구, AWS, 프로젝트 설정, 모든 예상 AWS Lambda 핸들러에서 거터 아이콘 표시 상자를 선택하여 현재 프로젝트에 아이콘을 표시해 봅니다. 또한 함수 핸들러가 해당 AWS SAM 템플릿에 이미 정의되어 있으면 AWS Lambda 새로 만들기 명령이 나타나지 않습니다.

함수 생성을 선택하면 AWS Toolkit for JetBrains에서 연결된 AWS 계정에 대한 해당 함수를 Lambda 서비스에 생성합니다. 작업이 성공하면 AWS Explorer를 새로 고친 후 Lambda 목록에 새 함수의 이름이 표시됩니다.

- AWS Lambda 함수가 포함된 프로젝트가 이미 있고, 먼저 다른 AWS 리전으로 전환하여 함수를 작성해야 하는 경우 지금 수행하세요. 그런 다음 Java, Python, Node.js 또는 C#에 대한 함수 핸들러가 포함된 코드 파일의 함수 핸들러 옆의 거터에서 Lambda 아이콘을 선택하세요. 새로 AWS Lambda 만들기를 선택하고 함수 만들기 대화 상자를 완료한 다음 함수 만들기를 선택합니다.



> **Note**
>
> 함수 핸들러 옆의 거터에 Lambda 아이콘이 표시되지 않으면 설정/기본 설정: 도구, AWS, 프로젝트 설정, 모든 예상 AWS Lambda 핸들러에서 거터 아이콘 표시 상자를 선택하여 현재 프로젝트에 아이콘을 표시해 봅니다. 또한 함수 핸들러가 해당 AWS SAM 템플릿에 이미 정의되어 있으면 새 AWS Lambda 생성 명령이 나타나지 않습니다.

함수 생성을 선택하면 AWS Toolkit for JetBrains에서 연결된 AWS 계정에 대한 해당 함수를 Lambda 서비스에 생성합니다. 작업이 성공하면 AWS Explorer를 새로 고친 후 새 함수의 이름이 Lambda 목록에 나타납니다.

함수를 만든 후 함수의 로컬 버전을 실행(호출) 또는 디버깅하거나 원격 버전을 실행(호출)할 수 있습니다.

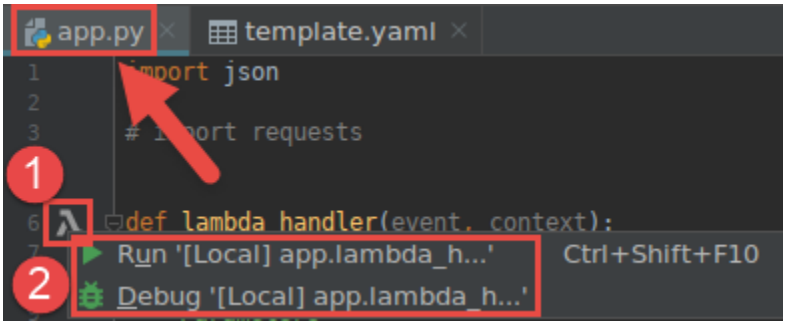
AWS Toolkit for JetBrains를 사용하여 AWS Lambda 함수의 로컬 버전 실행(간접적으로 호출) 또는 디버깅

이 절차를 완료하려면 실행(간접적으로 호출)하거나 디버깅하려는 AWS Lambda 함수를 아직 생성하지 않은 경우 해당 함수를 생성해야 합니다.

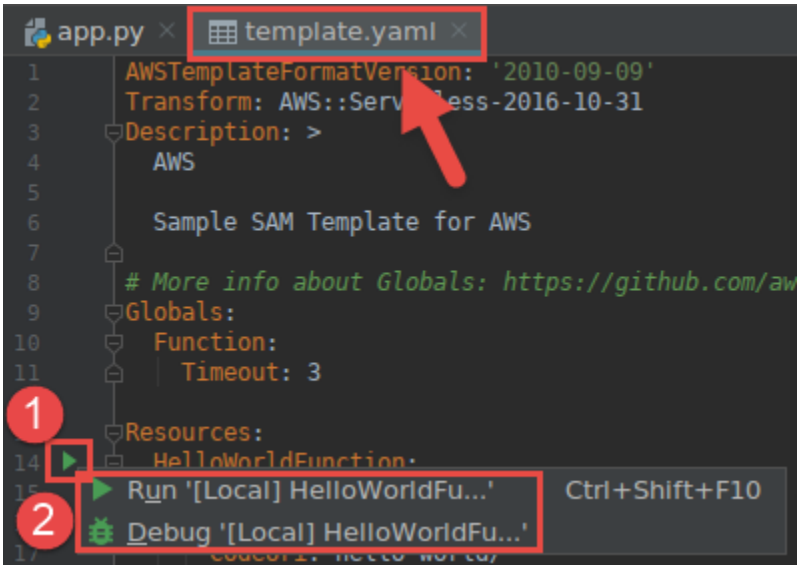
> **Note**
>
> Lambda 함수의 로컬 버전을 실행(간접적으로 호출)하거나 디버깅하고 기본값이 아닌 속성 또는 선택적 속성을 사용하여 해당 함수를 로컬에서 실행(간접적으로 호출)하거나 디버깅하려면 먼저 함수의 해당 AWS SAM 템플릿 파일(예: 프로젝트 내에서 파일 이름이 `template.yaml`인 파일)에 해당 속성을 설정해야 합니다. 사용 가능한 속성 목록은 GitHub의 awslabs/serverless-application-model 리포지토리에 있는 AWS::Serverless::Function을 참조하세요.

1. 다음 중 하나를 수행하세요.

   - Java, Python, Node.js 또는 C#에 대한 함수 핸들러가 포함된 코드 파일의 함수 핸들러 옆의 거터에서 Lambda 아이콘을 선택하세요. 실행 '[로컬]' 또는 디버깅'[로컬] '을 선택합니다.

   - 프로젝트 도구 창이 이미 열려 있고 함수가 포함된 프로젝트가 표시된 상태에서 프로젝트의 `template.yaml` 파일을 엽니다. 함수의 리소스 정의 옆에 있는 거터에서 실행 아이콘을 선택한 다음 실행'[로컬]' 또는 디버깅'[로컬]'을 선택합니다.



2. 구성 편집(로컬 함수 설정) 대화 상자가 표시되면 완료한 다음 실행 또는 디버깅을 선택합니다. 실행 또는 디버깅 도구 창에 결과가 표시됩니다.

   - 구성 편집 대화 상자가 나타나지 않고 기존 구성을 변경하려는 경우 먼저 구성을 변경한 다음 처음부터 이 절차를 반복합니다.
   - 구성 세부 정보가 누락된 경우 템플릿, AWS Lambda를 확장한 다음 로컬을 선택합니다. 확인을 선택한 다음 처음부터 이 절차를 반복합니다.

AWS Toolkit for JetBrains를 사용하여 AWS Lambda 함수의 원격 버전 실행(간접적으로 호출)

AWS Lambda 함수의 원격 버전은 소스 코드가 AWS 계정의 Lambda 서비스 내부에 이미 존재하는 함수입니다.

이 절차를 완료하려면 먼저 AWS Toolkit for JetBrains를 설치하고 아직 연결하지 않은 경우 AWS 계정에 처음 연결해야 합니다. 그런 다음 IntelliJ IDEA, PyCharm, WebStorm 또는 JetBrains Rider가 실행되고 있는 상태에서 다음을 수행합니다.

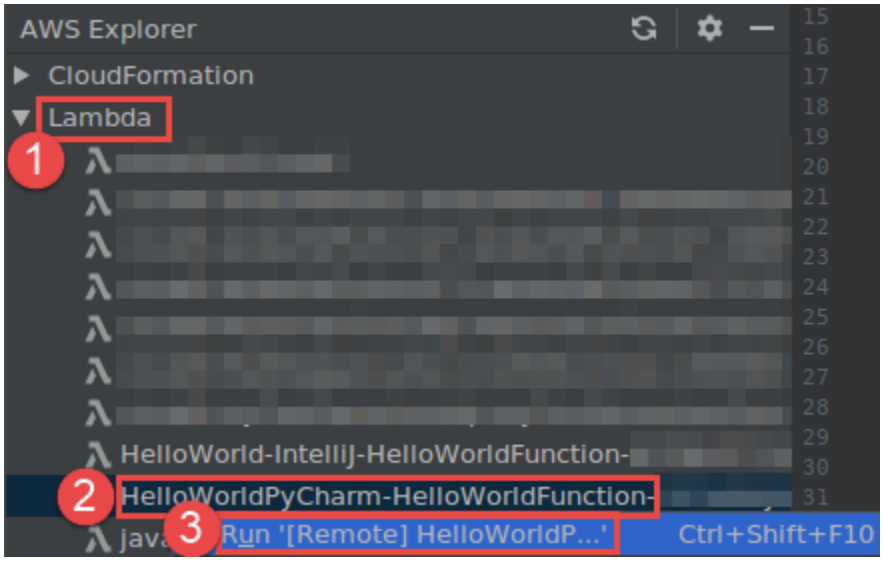
1. AWS Explorer가 열려 있지 않은 경우 이를 엽니다. 함수가 포함된 다른 AWS 리전으로 전환해야 하는 경우 지금 수행하세요.
2. Lambda를 확장하고 함수 이름이 표시되는지 확인합니다. 표시되는 경우 이 절차의 3단계로 건너뜁니다.

   함수 이름이 표시되지 않으면 실행(간접적으로 호출)하려는 Lambda 함수를 생성합니다.

함수를 AWS 서버리스 애플리케이션의 일부로 생성한 경우 해당 애플리케이션을 배포해야 합니다.

Java, Python, Node.js 또는 C#용 함수 핸들러를 구현하는 코드 파일을 생성하여 함수를 생성한 경우 코드 파일에서 함수 핸들러 옆에 있는 Lambda 아이콘을 선택합니다. 그런 다음 새 AWS Lambda 만들기를 선택합니다. 함수 만들기 대화 상자를 완료한 다음 함수 만들기를 선택합니다.

3. AWS Explorer에서 Lambda를 연 상태에서 함수 이름을 마우스 오른쪽 버튼으로 클릭한 다음 '[원격]' 실행을 선택합니다.



4. 구성 편집(원격 함수 설정) 대화 상자가 표시되면 완료한 다음 실행 또는 디버깅을 선택합니다. 실행 또는 디버깅 도구 창에 결과가 표시됩니다.

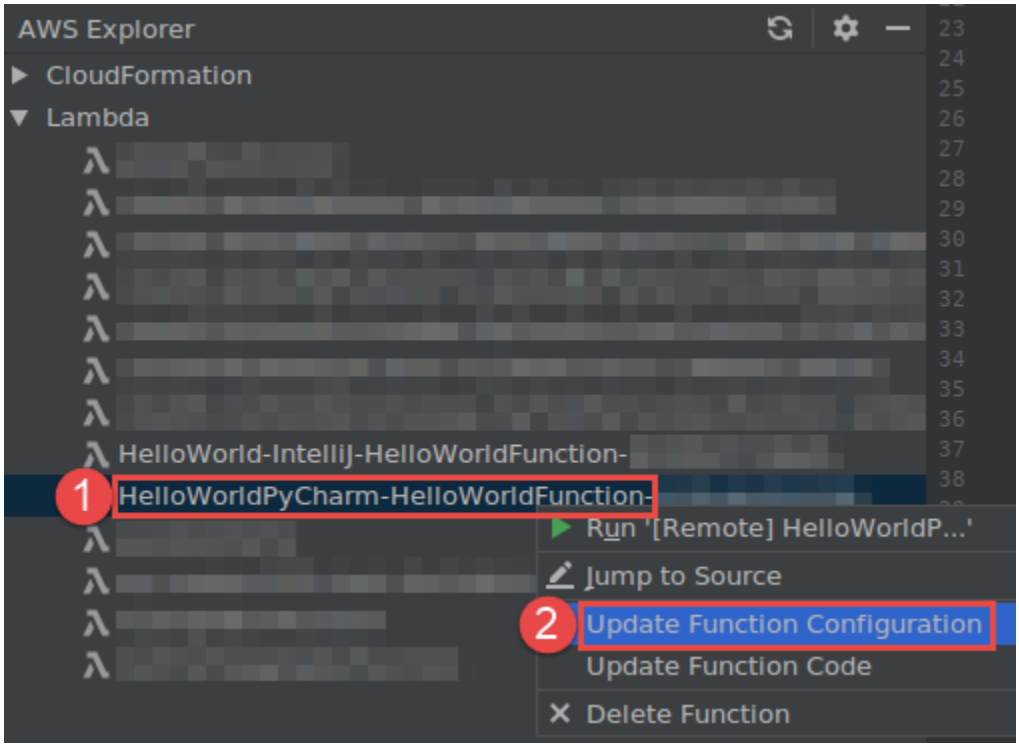
   - 구성 편집 대화 상자가 나타나지 않고 기존 구성을 변경하려는 경우 먼저 구성을 변경한 다음 처음부터 이 절차를 반복합니다.
   - 구성 세부 정보가 누락된 경우 템플릿, AWS Lambda를 확장한 다음 로컬을 선택합니다. 확인을 선택한 다음 처음부터 이 절차를 반복합니다.

AWS Toolkit for JetBrains를 사용하여 AWS Lambda 함수 설정 변경(업데이트)

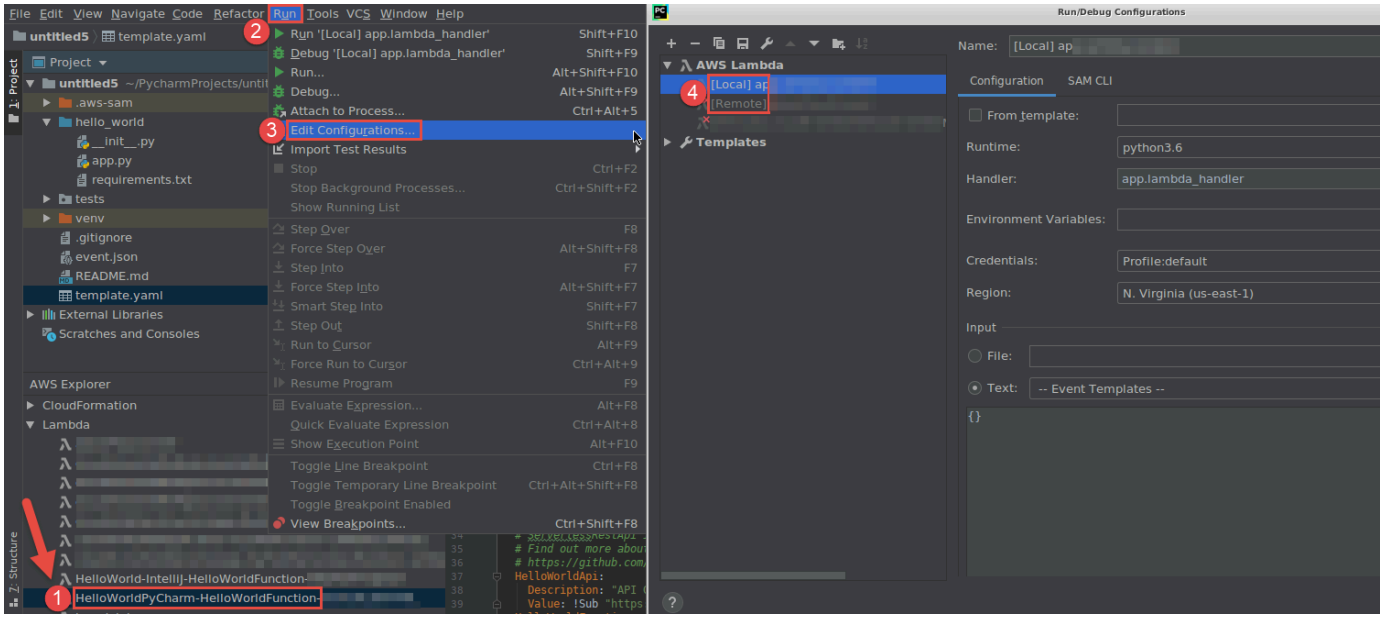
AWS Toolkit for JetBrains를 사용해 AWS Lambda 함수 설정을 변경(업데이트)하려면 다음 중 하나를 수행합니다.

- Java, Python, Node.js 또는 C#용 함수 핸들러가 포함된 코드 파일을 연 상태에서 기본 메뉴의 실행, 구성 편집을 선택합니다. 실행/디버깅 구성 대화 상자를 완료한 다음 확인을 선택합니다.

- AWS Explorer가 열려 있지 않은 경우 이를 엽니다. 함수가 포함된 다른 AWS 리전으로 전환해야 하는 경우 지금 수행하세요. Lambda를 확장하고 구성을 변경할 함수 이름을 선택한 후 다음 중 하나를 수행합니다.
  - 시간 초과, 메모리, 환경 변수 및 실행 역할과 같은 설정 변경 - 함수 이름을 마우스 오른쪽 버튼으로 클릭한 다음 함수 구성 업데이트를 선택합니다.

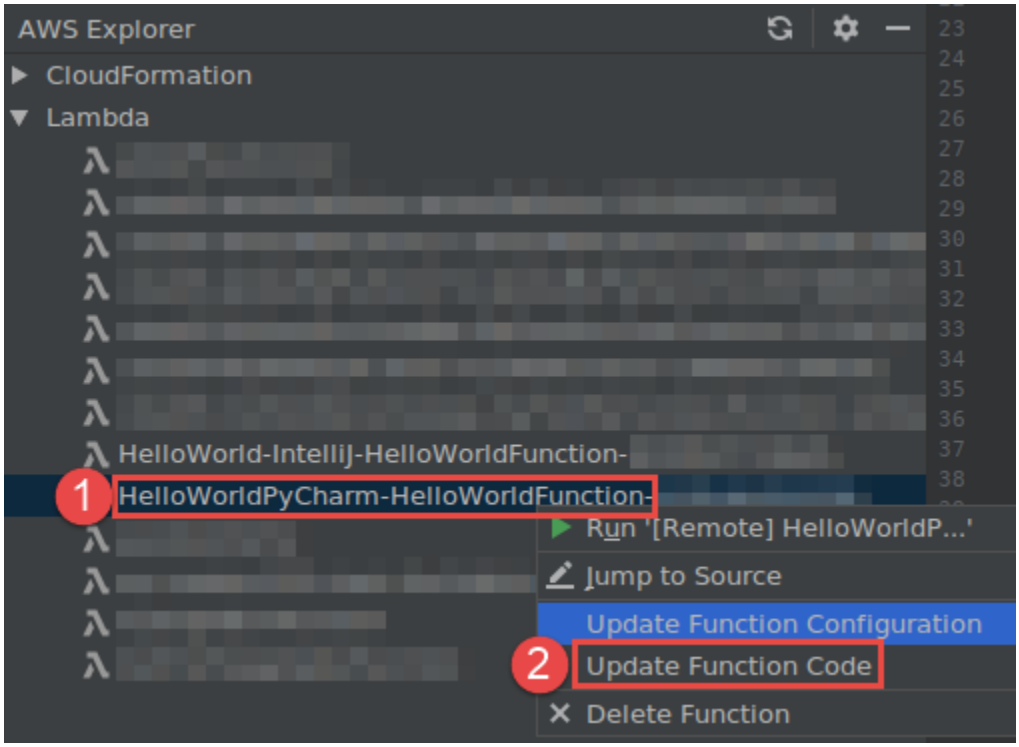


    구성 업데이트 대화 상자를 완료한 다음 업데이트를 선택합니다.
  - 입력 페이로드와 같은 설정 변경 – 기본 메뉴에서 실행, 구성 편집을 선택합니다. 실행/디버깅 구성 대화 상자를 완료한 다음 확인을 선택합니다.



구성 세부 정보가 누락된 경우 먼저 템플릿, AWS Lambda을 확장한 다음 로컬(함수의 로컬 버전) 또는 원격(동일한 함수의 원격 버전)을 선택합니다. 확인을 선택한 다음 처음부터 이 절차를 반복합니다.)

- 함수 핸들러 이름 또는 Amazon Simple Storage Service(S3) 소스 버킷과 같은 설정 변경 – 함수 이름을 마우스 오른쪽 버튼으로 클릭한 다음 함수 코드 업데이트를 선택합니다.



코드 업데이트 대화 상자를 완료한 다음 업데이트를 선택합니다.

- 이전 설명에 언급되지 않은 사용 가능한 다른 속성 설정 – 함수의 해당 AWS SAM 템플릿 파일(예: 프로젝트 내에서 이름이 `template.yaml`로 지정된 파일)에서 해당 설정을 변경합니다.

  사용 가능한 속성 설정 목록은 GitHub의 awslabs/serverless-application-model 리포지토리에 있는 AWS::Serverless::Function을 참조하세요.

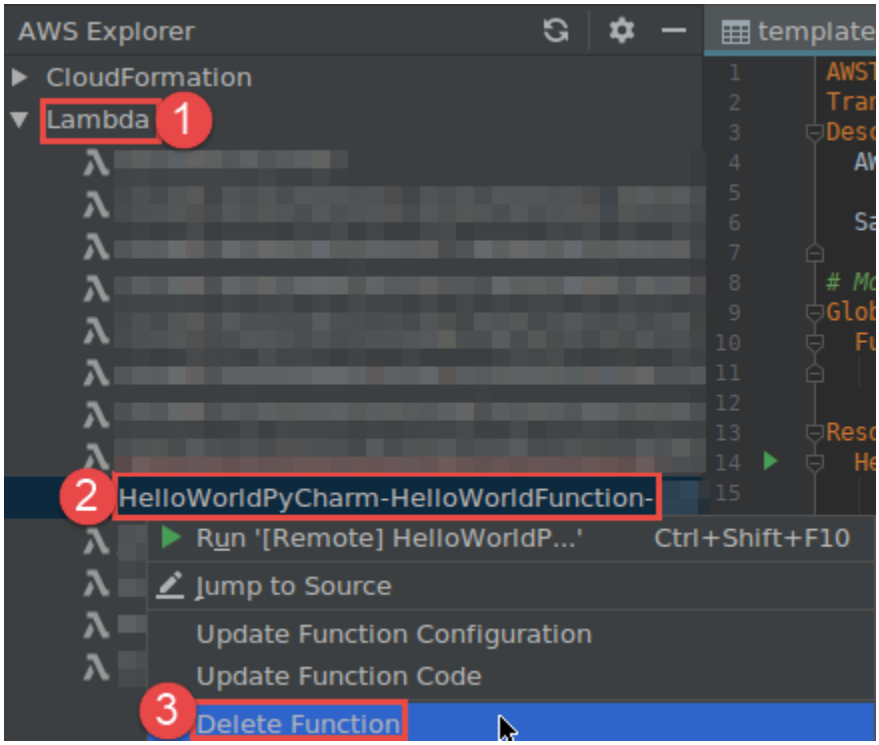
AWS Toolkit for JetBrains를 사용하여 AWS Lambda 함수 삭제

AWS Toolkit를 사용하여 AWS 서버리스 애플리케이션의 일부인 AWS Lambda 함수를 삭제하거나, 독립 실행형 Lambda 함수를 삭제할 수 있습니다.

AWS 서버리스 애플리케이션의 일부인 Lambda 함수를 삭제하려면 이 주제의 나머지 부분을 건너뛰고 대신 애플리케이션 삭제 섹션을 참조하세요.

독립 실행형 Lambda 함수를 삭제하려면 다음을 수행합니다.

1. AWS Explorer가 열려 있지 않은 경우 이를 엽니다. 함수가 포함된 다른 AWS 리전으로 전환해야 하는 경우 지금 수행하세요.
2. Lambda를 확장합니다.
3. 삭제할 함수 이름을 마우스 오른쪽 단추로 클릭한 다음 함수 삭제를 선택합니다.



4. 함수 이름을 입력하여 삭제를 확인한 다음 확인을 선택합니다. 함수 삭제가 성공하면 AWS Toolkit for JetBrains는 Lambda 목록에서 함수 이름을 제거합니다.

AWS Toolkit for JetBrains를 사용하여 Amazon RDS에 액세스

Amazon Relational Database Service(Amazon RDS)를 사용하면 클라우드에서 SQL 관계형 데이터베이스 시스템을 프로비저닝하고 관리할 수 있습니다. AWS Toolkit for JetBrains를 사용하면 다음 Amazon RDS 데이터베이스 엔진에 연결하고 상호 작용할 수 있습니다.

- Aurora – 클라우드용으로 구축된 MySQL 및 PostgreSQL 호환 관계형 데이터베이스입니다. 자세한 내용은 Amazon Aurora 사용 설명서를 참조하세요.
- MySQL – Amazon RDS는 오픈 소스 관계형 데이터베이스의 여러 주요 버전을 지원합니다. 자세한 내용은 Amazon RDS 사용 설명서의 Amazon RDS의 MySQL을 참조하세요.
- PostgreSQL – Amazon RDS는 오픈 소스 객체 관계형 데이터베이스의 여러 주요 버전을 지원합니다. 자세한 내용은 Amazon RDS 사용 설명서의 Amazon RDS for PostgreSQL을 참조하세요.

다음 주제에서는 RDS 데이터베이스에 액세스하기 위한 사전 조건과 AWS Toolkit for JetBrains를 사용하여 데이터베이스 인스턴스에 연결하는 방법을 설명합니다.

주제

- Amazon RDS 데이터베이스 액세스를 위한 사전 조건
- Amazon RDS 데이터베이스에 연결

Amazon RDS 데이터베이스 액세스를 위한 사전 조건

AWS Toolkit for JetBrains를 사용하여 Amazon RDS 데이터베이스에 연결하려면 먼저 다음 태스크를 완료해야 합니다.

- DB 인스턴스 생성 및 인증 방법 설정
- DataGrip 다운로드 및 설치

Amazon RDS DB 인스턴스 생성 및 인증 방법 구성

AWS Toolkit for JetBrains를 사용하면 AWS에서 이미 생성되고 구성된 Amazon RDS DB 인스턴스에 연결할 수 있습니다. DB 인스턴스는 사용자가 생성한 여러 데이터베이스를 포함할 수 있는 클라우드에

서 실행되는 격리된 데이터베이스 환경입니다. 지원되는 데이터베이스 엔진용 DB 인스턴스 생성에 대한 자세한 내용은 Amazon RDS 사용 설명서의 Amazon RDS 시작하기를 참조하세요.

AWS Toolkit for JetBrains를 사용하여 데이터베이스에 연결할 때 사용자는 IAM 보안 인증 또는 Secrets Manager를 사용하여 인증하도록 선택할 수 있습니다. 다음 표에는 두 옵션의 주요 기능 및 정보 리소스가 설명되어 있습니다.

인증 방법	작동 방식	추가 정보
IAM 보안 인증을 사용하여 연결	IAM 데이터베이스 인증을 사용하면 인증이 AWS Identity and Access Management(IAM) 보안 인증을 사용하여 외부에서 관리되므로 데이터베이스에 사용자 보안 인증을 저장할 필요가 없습니다. DB 인스턴스에서는 기본적으로 IAM 데이터베이스 인증이 비활성화되어 있습니다. IAM 데이터베이스 인증은 AWS Management Console, AWS CLI 또는 API를 사용하여 활성화하거나 다시 비활성화할 수 있습니다.	• Amazon RDS User Guide의 Amazon RDS의 보안 인증 및 액세스 관리 • AWS 지식 센터 문서: 사용자가 IAM 보안 인증을 사용하여 Amazon RDS for MySQL DB 인스턴스에 인증하도록 허용하려면 어떻게 해야 하나요?
AWS Secrets Manager를 사용하여 연결	데이터베이스 관리자는 데이터베이스에 대한 보안 인증을 Secrets Manager에 보안 정보로 저장할 수 있습니다. Secrets Manager는 보안 인증을 암호화하여 보안 암호 내에 보호되는 보안 암호 텍스트로 저장합니다. 권한이 있는 애플리케이션이 데이터베이스에 액세스하면 Secrets Manager는 보호된 비	• AWS Secrets Manager 사용 설명서의 AWS Secrets Manager란 무엇인가요? • AWS Secrets Manager 사용 설명서의 자습서: AWS Secrets Manager 보안 암호 교체 • AWS Security 블로그: Rotate Amazon RDS database credentials

인증 방법	작동 방식	추가 정보
	밀 텍스트를 해독하고 보안 채널을 통해 반환합니다. 클라이언트는 보안 인증, 연결 문자열 및 기타 필요한 정보를 구문 분석한 다음 해당 정보를 사용하여 데이터베이스에 액세스합니다.	automatically with Secrets Manager

DataGrip을 사용하여 Amazon RDS 데이터베이스 작업

Amazon RDS 데이터 소스에 연결한 후 데이터 소스와 상호 작용을 시작할 수 있습니다. JetBrains의 DataGrip을 사용하면 SQL 작성, 쿼리 실행, 데이터 가져오기/내보내기와 같은 데이터베이스 태스크를 수행할 수 있습니다. DataGrip에서 제공하는 기능은 다양한 JetBrains IDE의 데이터베이스 플러그인에서도 사용할 수 있습니다. DataGrip에 대한 자세한 내용은 https://www.jetbrains.com/datagrip/을 참조하세요.

Amazon RDS 데이터베이스에 연결

AWS Explorer를 사용하면 Amazon RDS 데이터베이스를 선택하고 인증 방법을 선택한 다음 연결 설정을 구성할 수 있습니다. 연결 테스트에 성공하면 JetBrains DataGrip을 사용하여 데이터 소스와의 상호 작용을 시작할 수 있습니다.

> **Important**
>
> 사용자가 Amazon RDS 데이터베이스에 액세스하고 상호 작용할 수 있도록 사전 조건을 완료했는지 확인하세요.

기본 인증 방법을 사용하여 데이터베이스 인스턴스에 연결하는 방법에 대한 지침을 보려면 탭을 선택합니다.

Connect with IAM credentials

1. AWS Explorer가 열려 있지 않은 경우 이를 엽니다.
2. Amazon RDS 노드를 클릭하여 지원되는 데이터베이스 엔진 목록을 확장합니다.

3. 지원되는 데이터베이스 엔진(Aurora, MySQL 또는 PostgreSQL) 노드를 클릭하여 사용 가능한 데이터베이스 인스턴스 목록을 확장합니다.

> **Note**
>
> Aurora를 선택하면 MySQL 클러스터 확장과 PostgreSQL 클러스터 확장 중에서 선택할 수 있습니다.

4. 데이터베이스를 마우스 오른쪽 버튼으로 클릭하고 IAM 보안 인증으로 연결을 선택합니다.

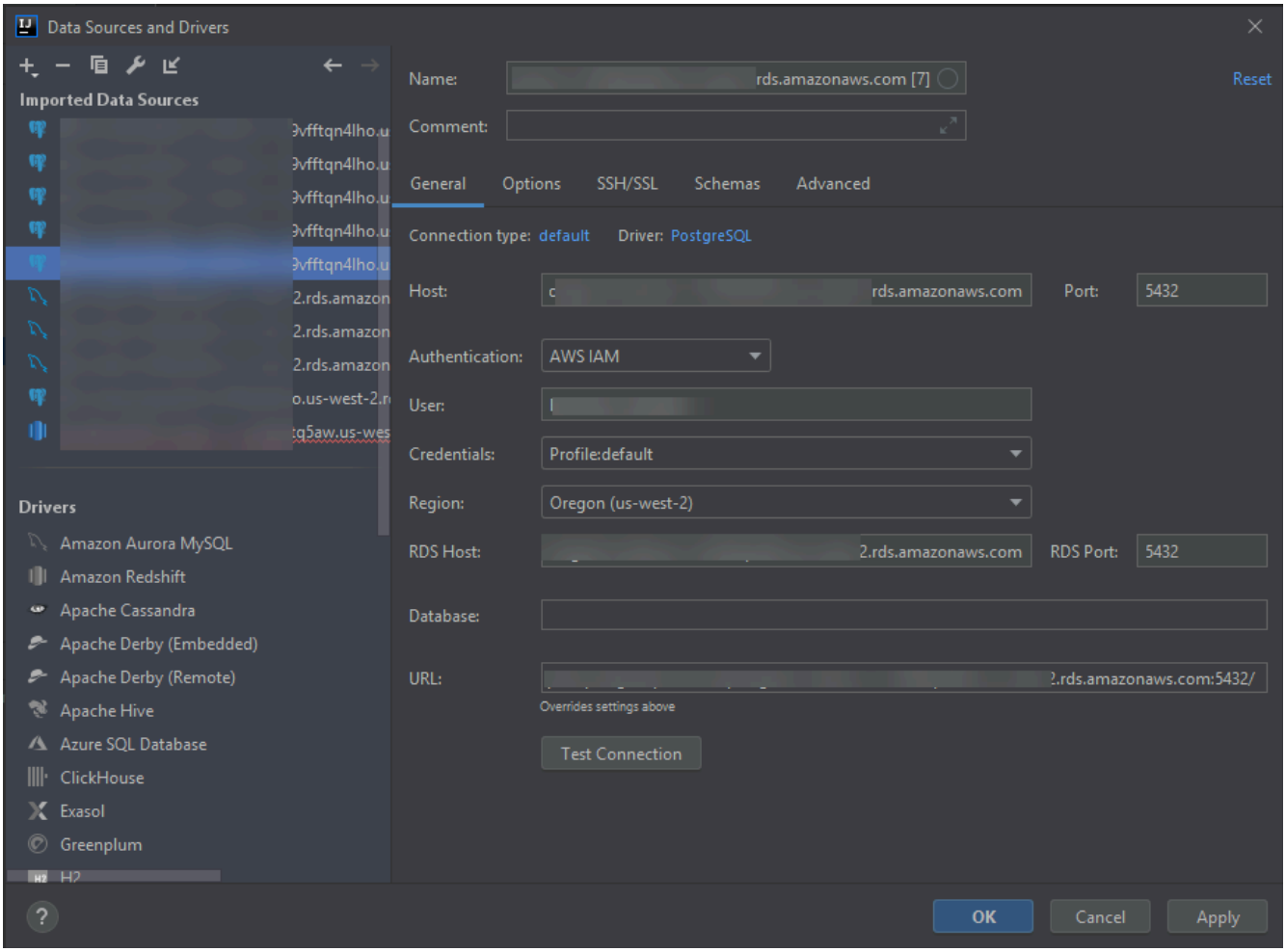
> **Note**
>
> ARN 복사를 선택하여 클립보드에 데이터베이스의 Amazon 리소스 이름(ARN)을 추가할 수도 있습니다.

5. 데이터 소스 및 드라이버 대화 상자에서 다음을 수행하여 데이터베이스 연결을 열 수 있도록 합니다.

   - 가져온 데이터 소스 창에서 올바른 데이터 소스가 선택되었는지 확인합니다.
   - 누락된 드라이버 파일을 다운로드해야 한다는 메시지가 표시되면 드라이버로 이동(렌치 아이콘)을 선택하여 필요한 파일을 다운로드합니다.

6. 설정 창의 일반 탭에서 다음 필드에 올바른 값이 표시되는지 확인합니다.

   - 호스트/포트 - 데이터베이스 연결에 사용되는 엔드포인트와 포트입니다. AWS Cloud에서 호스팅되는 Amazon RDS 데이터베이스의 경우 엔드포인트는 항상 `rds.amazon.com`으로 끝납니다. 프록시를 통해 DB 인스턴스에 연결하는 경우 이 필드를 사용하여 프록시의 연결 세부 정보를 지정합니다.
   - 인증 - AWS IAM(IAM 보안 인증을 사용한 인증)입니다.
   - 사용자 – 데이터베이스 사용자 계정의 이름입니다.
   - 보안 인증 - AWS 계정에 액세스하는 데 사용되는 보안 인증입니다.
   - 리전 - 데이터베이스가 호스팅되는 AWS 리전입니다.
   - RDS 호스트/포트 - AWS Management Console에 나열된 데이터베이스의 엔드포인트 및 포트입니다. 다른 엔드포인트를 사용하여 DB 인스턴스에 연결하는 경우 호스트/포트 필드에 프록시의 연결 세부 정보를 지정합니다(앞서 설명).
   - 데이터베이스 – 데이터베이스의 이름입니다.
   - URL – JetBrains IDE가 데이터베이스에 연결하는 데 사용할 URL입니다.



> **Note**
>
> 데이터 소스 및 드라이버 대화 상자를 사용하여 구성할 수 있는 연결 설정에 대한 자세한 설명은 사용 중인 JetBrains IDE의 설명서를 참조하세요.

7. 연결 설정이 올바른지 확인하려면 연결 테스트를 선택합니다.

   녹색 확인 표시는 테스트 성공을 나타냅니다.

8. 적용을 선택하여 설정을 적용한 다음 확인을 선택하여 데이터 소스 작업을 시작합니다.

   데이터베이스 도구 창이 열립니다. 그러면 사용 가능한 데이터 소스가 스키마, 테이블, 키 등의 데이터베이스 요소를 나타내는 노드가 있는 트리로 표시됩니다.

> **Important**
>
> 데이터베이스 도구 창을 사용하려면 먼저 JetBrains에서 DataGrip을 다운로드하여 설치해야 합니다. 자세한 내용은 https://www.jetbrains.com/datagrip/을 참조하세요.

Connect with Secrets Manager

1. AWS Explorer가 열려 있지 않은 경우 이를 엽니다.
2. Amazon RDS 노드를 클릭하여 지원되는 데이터베이스 엔진 목록을 확장합니다.
3. 지원되는 데이터베이스 엔진(Aurora, MySQL 또는 PostgreSQL) 노드를 클릭하여 사용 가능한 데이터베이스 인스턴스 목록을 확장합니다.

   > **Note**
   >
   > Aurora를 선택하면 MySQL 클러스터 확장과 PostgreSQL 클러스터 확장 중에서 선택할 수 있습니다.

4. 데이터베이스를 마우스 오른쪽 버튼으로 클릭하고 Secrets Manager와 연결을 선택합니다.

   > **Note**
   >
   > ARN 복사를 선택하여 클립보드에 데이터베이스의 Amazon 리소스 이름(ARN)을 추가할 수도 있습니다.

5. 데이터베이스 보안 암호 선택 대화 상자에서 드롭다운 필드를 사용하여 데이터베이스에 대한 보안 인증을 선택한 다음 생성을 선택합니다.
6. 데이터 소스 및 드라이버 대화 상자에서 다음을 수행하여 데이터베이스 연결을 열 수 있도록 합니다.

   - 가져온 데이터 소스 창에서 올바른 데이터 소스가 선택되었는지 확인합니다.
   - 누락된 드라이버 파일을 다운로드해야 한다는 메시지가 표시되면 드라이버로 이동(렌치 아이콘)을 선택하여 필요한 파일을 다운로드합니다.

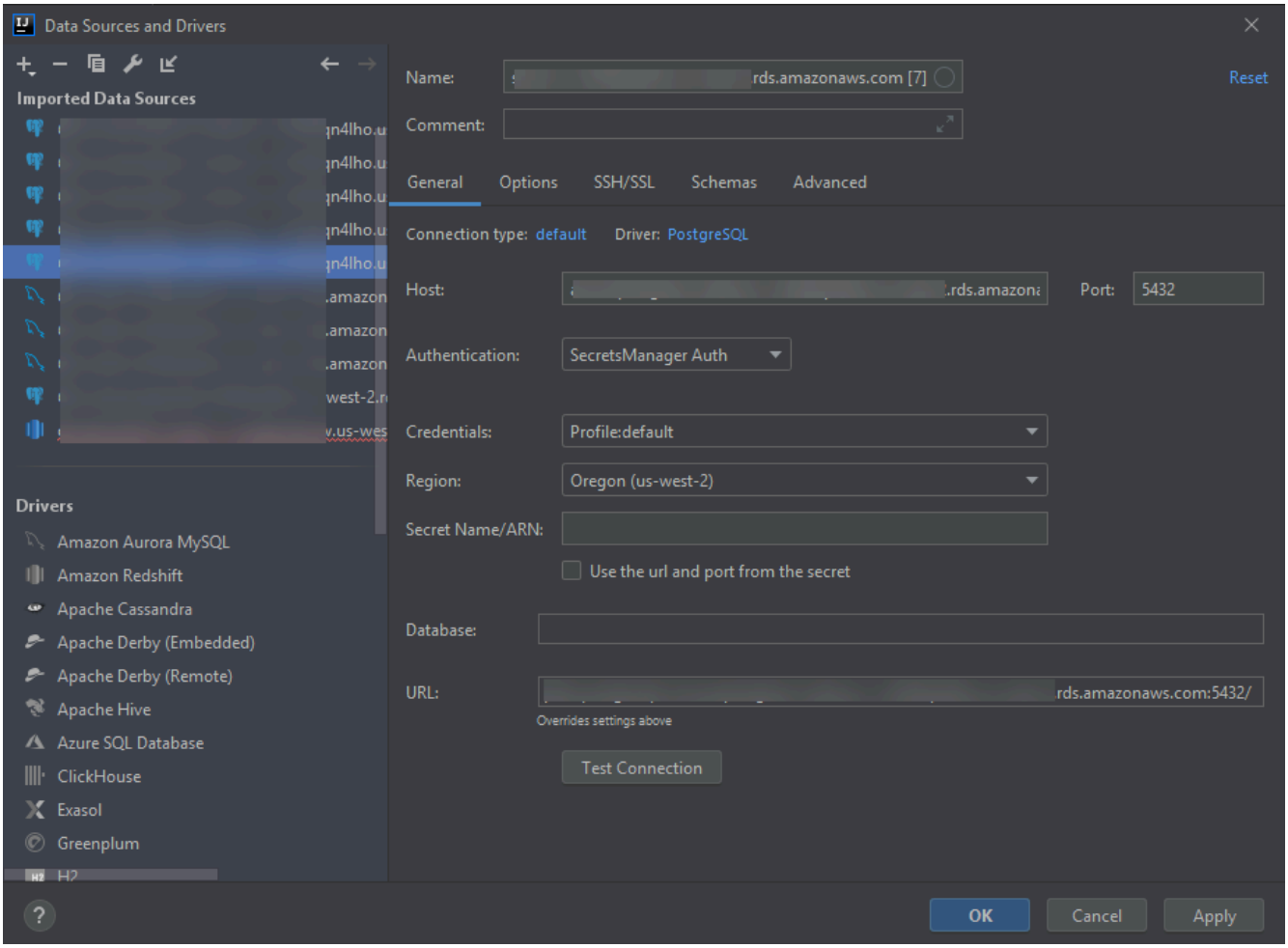
7. 설정 창의 일반 탭에서 다음 필드에 올바른 값이 표시되는지 확인합니다.

   - 호스트/포트 - 데이터베이스 연결에 사용되는 엔드포인트와 포트입니다. AWS Cloud에서 호스팅되는 Amazon RDS 데이터베이스의 경우 엔드포인트는 항상 `rds.amazon.com`으로 끝

납니다. 프록시 데이터베이스를 통해 데이터베이스에 연결하는 경우 이 필드를 사용하여 프록시의 연결 세부 정보를 지정합니다.

- 인증 – SecretsManager Auth(AWS Secrets Manager를 사용하여 인증)입니다.
- 보안 인증 - AWS 계정에 액세스하는 데 사용되는 보안 인증입니다.
- 리전 - 데이터베이스가 호스팅되는 AWS 리전입니다.
- 보안 암호/ARN - 인증 보안 인증 정보가 포함된 보안 암호의 이름과 ARN입니다. 호스트/포트 필드의 연결 설정을 재정의하려면 보안 암호의 URL 및 포트 사용 확인란을 선택합니다.
- 데이터베이스 - AWS Explorer에서 선택한 데이터베이스 인스턴스의 이름입니다.
- URL – JetBrains IDE가 데이터베이스에 연결하는 데 사용할 URL입니다.

> **Note**
>
> 인증에 Secrets Manager를 사용하는 경우 데이터베이스의 사용자 이름 및 암호 필드가 없습니다. 이 정보는 보안 암호의 암호화된 보안 암호 데이터 부분에 포함되어 있습니다.



> **Note**
>
> 데이터 소스 및 드라이버 대화 상자를 사용하여 구성할 수 있는 연결 설정에 대한 자세한 설명은 사용 중인 JetBrains IDE의 설명서를 참조하세요.

8. 연결 설정이 올바른지 확인하려면 연결 테스트를 선택합니다.

   녹색 확인 표시는 테스트 성공을 나타냅니다.

9. 적용을 선택하여 설정을 적용한 다음 확인을 선택하여 데이터 소스 작업을 시작합니다.

   데이터베이스 도구 창이 열립니다. 그러면 사용 가능한 데이터 소스가 스키마, 테이블, 키 등의 데이터베이스 요소를 나타내는 노드가 있는 트리로 표시됩니다.

> **Important**
>
> 데이터베이스 도구 창을 사용하려면 먼저 JetBrains에서 DataGrip을 다운로드하여 설치해야 합니다. 자세한 내용은 https://www.jetbrains.com/datagrip/을 참조하세요.

AWS Toolkit for JetBrains를 사용하여 Amazon Redshift에 액세스

Amazon Redshift 데이터웨어 하우스는 엔터프라이즈 급 관계형 데이터베이스 쿼리 및 관리 시스템입니다. AWS Toolkit for JetBrains를 사용하면 Amazon Redshift 클러스터에 연결하고 상호 작용할 수 있습니다. Amazon Redshift 클러스터는 클라이언트가 해당 클러스터에서 호스팅되는 데이터베이스를 쿼리할 수 있게 하는 노드 모음으로 구성됩니다.

다음 주제에서는 Amazon Redshift 클러스터에 액세스하기 위한 사전 조건과 AWS Toolkit for JetBrains를 사용하여 클러스터의 인스턴스에 연결하는 방법을 설명합니다.

주제

Amazon Redshift 클러스터에 액세스하기 위한 사전 조건

AWS Toolkit for JetBrains를 사용하여 Amazon Redshift 클러스터와 상호 작용을 시작하기 전에 다음 태스크를 완료해야 합니다.

Amazon Redshift 클러스터 생성 및 인증 방법 구성

AWS Toolkit for JetBrains를 사용하면 AWS에서 이미 생성되고 구성된 Amazon Redshift 클러스터에 연결할 수 있습니다. 각 클러스터에는 하나 이상의 데이터베이스가 포함되어 있습니다. Amazon Redshift 클러스터를 생성하고 구성하는 방법에 대한 자세한 내용은 Amazon Redshift 시작 안내서의 Amazon Redshift Serverless 시작하기를 참조하세요.

AWS Toolkit for JetBrains를 사용하여 클러스터에 연결할 때 사용자는 IAM 보안 인증 또는 AWS Secrets Manager를 사용하여 인증하도록 선택할 수 있습니다. 다음 표에는 두 옵션의 주요 기능 및 정보 리소스가 설명되어 있습니다.

인증 방법	작동 방식	추가 정보
IAM 보안 인증을 사용하여 연결	IAM 데이터베이스 인증을 사용하면 인증이 AWS Identity and Access Management(IAM) 보안 인증을 사용하여 외부에서 관리되므로 데이터베이스에 사용자 보안 인증을 저장할 필요가 없습니다. 데이터베이스 클러스터에서는 기본적으로 IAM 데이터베이스 인증이 비활성화되어 있습니다. IAM 데이터베이스 인증은 AWS Management Console, AWS CLI 또는 API를 사용하여 활성화하거나 다시 비활성화할 수 있습니다.	• Amazon Redshift 관리 안내서의 Amazon Redshift의 Identity and Access Management
AWS Secrets Manager를 사용하여 연결	데이터베이스 관리자는 데이터베이스에 대한 보안 인증을 Secrets Manager에 보안 정보로 저장할 수 있습니다. Secrets Manager는 보안 인증을 암호화하여 보안 암호 내에 보호되는 보안 암호 텍스트로 저장합니다. 권한이 있는 애플리케이션이 데이터베이스에 액세스하면 Secrets Manager는 보호된 비밀 텍스트를 해독하고 보안 채널을 통해 반환합니다. 클라이언트는 보안 인증, 연결 문자열	• AWS Secrets Manager 사용 설명서의 AWS Secrets Manager란 무엇인가요? • AWS Secrets Manager 사용 설명서의 AWS Secrets Manager 보안 암호 교체 • AWS Security 블로그: How to rotate Amazon DocumentDB and Amazon Redshift credentials in Secrets Manager

인증 방법	작동 방식	추가 정보
	및 기타 필요한 정보를 구문 분석한 다음 해당 정보를 사용하여 데이터베이스에 액세스합니다.	

DataGrip을 사용하여 Amazon RDS 데이터베이스 작업

Amazon Redshift 클러스터의 데이터베이스에 연결한 후 해당 데이터베이스와 상호 작용을 시작할 수 있습니다. JetBrains의 DataGrip을 사용하여 SQL 작성, 쿼리 실행, 데이터 가져오기/내보내기와 같은 데이터베이스 태스크를 수행할 수 있습니다. DataGrip에서 제공하는 기능은 다양한 JetBrains IDE의 데이터베이스 플러그인에서도 사용할 수 있습니다. DataGrip에 대한 자세한 내용은 https://www.jetbrains.com/datagrip/을 참조하세요.

Amazon Redshift 클러스터에 연결

AWS Explorer를 사용하면 Amazon Redshift 클러스터를 선택하고 인증 방법을 선택한 다음 연결 설정을 구성할 수 있습니다. 연결 테스트에 성공하면 JetBrains DataGrip을 사용하여 데이터 소스와의 상호 작용을 시작할 수 있습니다.

> **Important**
>
> 사용자가 Amazon Redshift 클러스터와 데이터베이스에 액세스하고 상호 작용할 수 있도록 사전 조건을 완료했는지 확인하세요.

기본 인증 방법을 사용하여 클러스터에 연결하는 방법에 대한 지침을 보려면 탭을 선택합니다.

Connect with IAM credentials

1. AWS Explorer가 열려 있지 않은 경우 이를 엽니다.
2. Amazon Redshift 노드를 클릭하여 사용 가능한 클러스터 목록을 확장합니다.
3. 클러스터를 마우스 오른쪽 버튼으로 클릭하고 IAM 보안 인증으로 연결을 선택합니다.

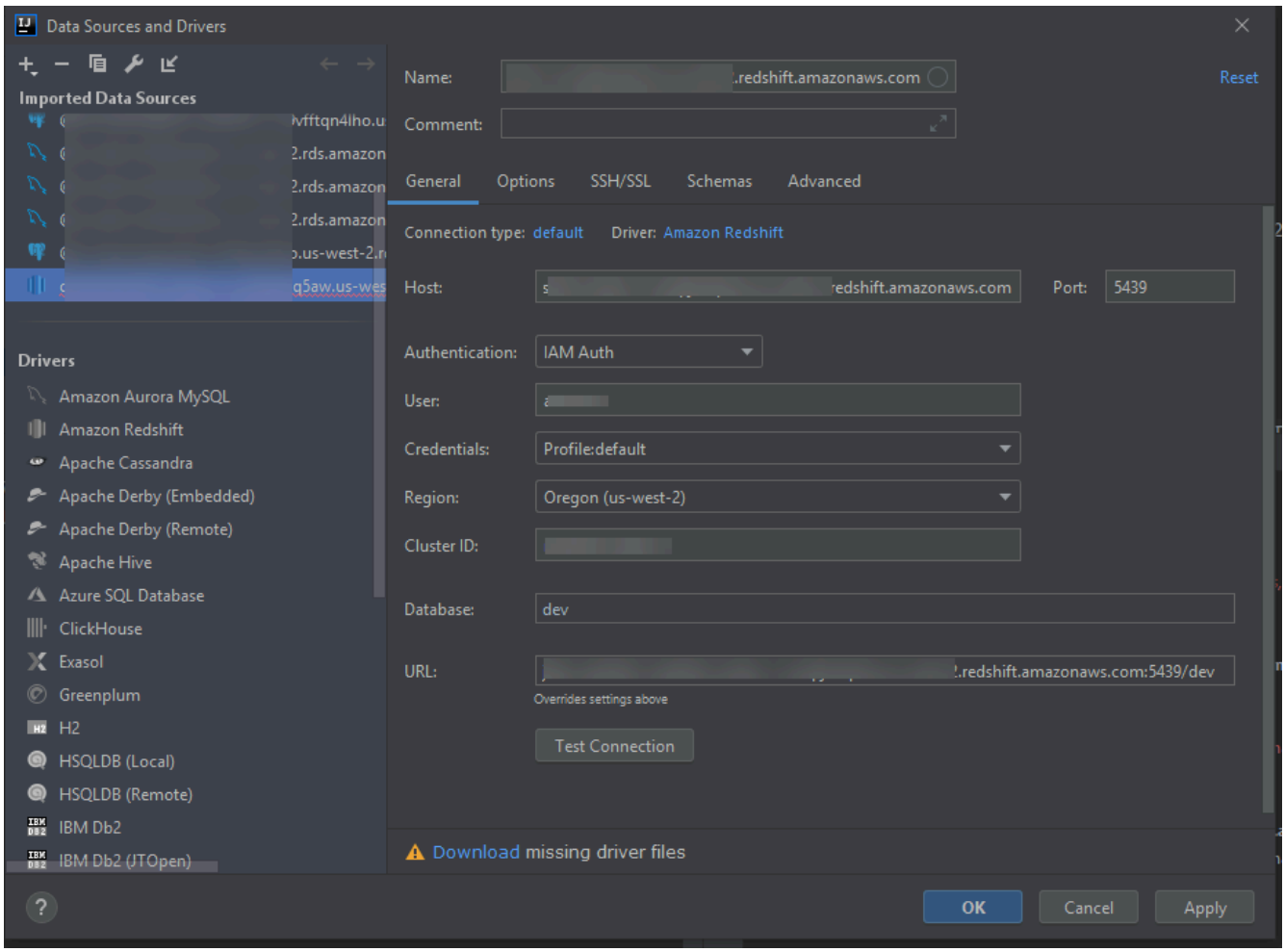
> **Note**
>
> ARN 복사를 선택하여 클립보드에 클러스터의 Amazon 리소스 이름(ARN)을 추가할 수도 있습니다.

4. 데이터 소스 및 드라이버 대화 상자에서 다음을 수행하여 데이터베이스 연결을 열 수 있도록 합니다.

   - 가져온 데이터 소스 창에서 올바른 데이터 소스가 선택되었는지 확인합니다.
   - 누락된 드라이버 파일을 다운로드해야 한다는 메시지가 표시되면 드라이버로 이동(렌치 아이콘)을 선택하여 필요한 파일을 다운로드합니다.

5. 설정 창의 일반 탭에서 다음 필드에 올바른 값이 표시되는지 확인합니다.

   - 호스트/포트 - 클러스터 연결에 사용되는 엔드포인트와 포트입니다. AWS Cloud에서 호스팅되는 Amazon Redshift 클러스터의 경우 엔드포인트는 항상 `redshift.amazon.com`으로 끝납니다.
   - 인증 - AWS IAM(IAM 보안 인증을 사용한 인증)입니다.
   - 사용자 – 데이터베이스 사용자 계정의 이름입니다.
   - 보안 인증 - AWS 계정에 액세스하는 데 사용되는 보안 인증입니다.
   - 리전 - 데이터베이스가 호스팅되는 AWS 리전입니다.
   - 클러스터 ID – AWS Explorer에서 선택한 클러스터의 ID입니다.
   - 데이터베이스 - 연결할 클러스터의 데이터베이스 이름입니다.
   - URL – JetBrains IDE가 클러스터의 데이터베이스에 연결하는 데 사용할 URL입니다.



> **Note**
>
> 데이터 소스 및 드라이버 대화 상자를 사용하여 구성할 수 있는 연결 설정에 대한 자세한 설명은 사용 중인 JetBrains IDE의 설명서를 참조하세요.

6. 연결 설정이 올바른지 확인하려면 연결 테스트를 선택합니다.

   녹색 확인 표시는 테스트 성공을 나타냅니다.

7. 적용을 선택하여 설정을 적용한 다음 확인을 선택하여 데이터 소스 작업을 시작합니다.

   데이터베이스 도구 창이 열립니다. 그러면 사용 가능한 데이터 소스가 스키마, 테이블, 키 등의 데이터베이스 요소를 나타내는 노드가 있는 트리로 표시됩니다.

> **Important**
>
> 데이터베이스 도구 창을 사용하려면 먼저 JetBrains에서 DataGrip을 다운로드하여 설치해야 합니다. 자세한 내용은 https://www.jetbrains.com/datagrip/을 참조하세요.

Connect with Secrets Manager

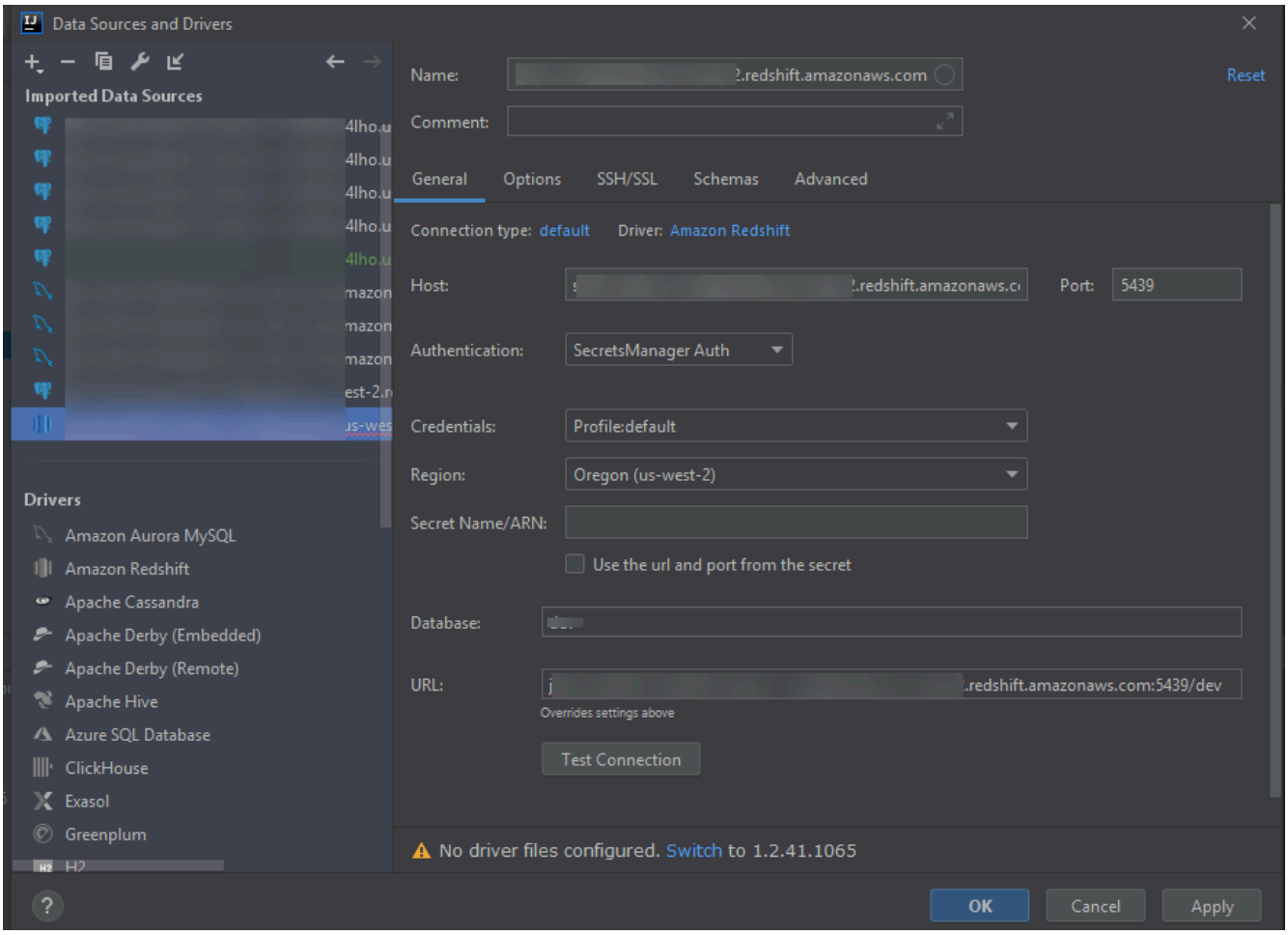
1. AWS Explorer가 열려 있지 않은 경우 이를 엽니다.
2. Amazon Redshift 노드를 클릭하여 사용 가능한 클러스터 목록을 확장합니다.
3. 클러스터를 마우스 오른쪽 버튼으로 클릭하고 Secrets Manager와 연결을 선택합니다.

> **Note**
>
> ARN 복사를 선택하여 클립보드에 클러스터의 Amazon 리소스 이름(ARN)을 추가할 수도 있습니다.

4. 데이터베이스 보안 암호 선택 대화 상자에서 드롭다운 필드를 사용하여 데이터베이스에 대한 보안 인증을 선택한 다음 생성을 선택합니다.
5. 데이터 소스 및 드라이버 대화 상자에서 다음을 수행하여 데이터베이스 연결을 열 수 있도록 합니다.
   - 가져온 데이터 소스에서 올바른 데이터 소스가 선택되었는지 확인합니다.
   - 누락된 드라이버 파일을 다운로드해야 한다는 메시지가 나타나면 드라이버로 이동(렌치 아이콘)을 선택하여 필요한 파일을 다운로드합니다.
6. 설정 창의 일반 탭에서 다음 필드에 올바른 값이 표시되는지 확인합니다.
   - 호스트/포트 - 클러스터 연결에 사용되는 엔드포인트와 포트입니다. AWS Cloud에서 호스팅되는 Amazon Redshift 클러스터의 경우 엔드포인트는 항상 `redshift.amazon.com`으로 끝납니다.
   - 인증 – SecretsManager Auth(AWS Secrets Manager를 사용하여 인증)입니다.
   - 보안 인증 - AWS 계정에 연결하는 데 사용되는 보안 인증입니다.
   - 리전 - 클러스터가 호스팅되는 AWS 리전입니다.

- 보안 암호/ARN - 인증 보안 인증 정보가 포함된 보안 암호의 이름과 ARN입니다. 호스트/포트 필드의 연결 설정을 재정의하려는 경우 보안 암호의 URL 및 포트 사용 확인란을 선택합니다.
- 데이터베이스 - 연결할 클러스터의 데이터베이스 이름입니다.
- URL – JetBrains IDE가 데이터베이스에 연결하는 데 사용할 URL입니다.

> **Note**
>
> 인증에 AWS Secrets Manager를 사용하는 경우 클러스터의 사용자 이름과 암호를 지정하는 필드가 없습니다. 이 정보는 보안 암호의 암호화된 보안 암호 데이터 부분에 포함되어 있습니다.



> **Note**
>
> 데이터 소스 및 드라이버 대화 상자를 사용하여 구성할 수 있는 연결 설정에 대한 자세한 설명은 사용 중인 JetBrains IDE의 설명서를 참조하세요.

7. 연결 설정이 올바른지 확인하려면 연결 테스트를 선택합니다.

   녹색 확인 표시는 테스트 성공을 나타냅니다.

8. 적용을 선택하여 설정을 적용한 다음 확인을 선택하여 데이터 소스 작업을 시작합니다.

   데이터베이스 도구 창이 열립니다. 그러면 사용 가능한 데이터 소스가 스키마, 테이블, 키 등의 데이터베이스 요소를 나타내는 노드가 있는 트리로 표시됩니다.

> **Important**
>
> 데이터베이스 도구 창을 사용하려면 먼저 JetBrains에서 DataGrip을 다운로드하여 설치해야 합니다. 자세한 내용은 https://www.jetbrains.com/datagrip/을 참조하세요.

AWS Toolkit for JetBrains를 사용하여 Amazon S3 작업

다음 주제에서는 AWS Toolkit for JetBrains를 사용하여 AWS 계정에서 Amazon S3 버킷 및 객체 작업을 수행하는 방법을 설명합니다.

주제

- AWS Toolkit for JetBrains를 사용하여 Amazon S3 버킷 작업
- AWS Toolkit for JetBrains를 사용하여 Amazon S3 객체 작업

AWS Toolkit for JetBrains를 사용하여 Amazon S3 버킷 작업

Amazon S3에 저장한 모든 객체는 버킷에 존재합니다. 디렉토리로 파일 시스템 내 파일을 그룹화하듯 버킷으로 관련 객체를 그룹화할 수 있습니다.

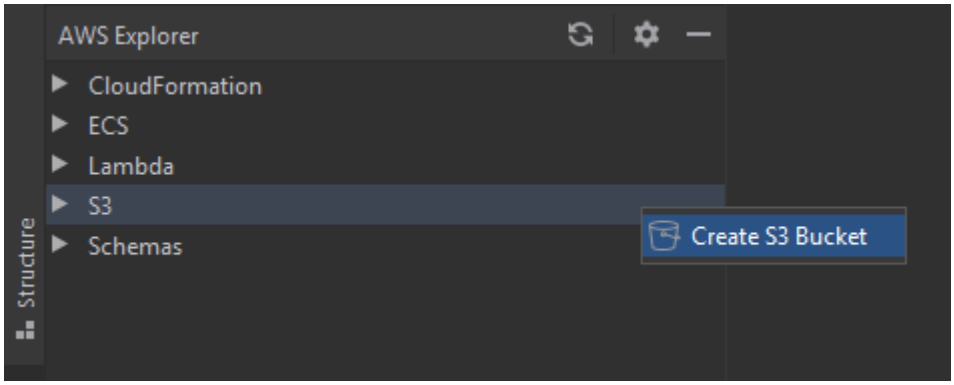
주제

- Amazon S3 버킷 생성
- Amazon S3 버킷 보기

- Amazon S3 버킷 삭제

Amazon S3 버킷 생성

1. AWS Explorer가 열려 있지 않은 경우 이를 엽니다.
2. Amazon S3 노드를 마우스 오른쪽 버튼으로 클릭하고 S3 버킷 생성을 선택합니다.



3. S3 버킷 만들기 대화 상자에서 버킷 이름을 입력합니다.

> **참고**
>
> Amazon S3는 버킷이 공개적으로 액세스할 수 URL로 사용되는 것을 허용하기 때문에 선택하는 버킷 이름이 전역적으로 고유해야 합니다. 다른 계정이 이미 선택한 이름을 가진 버킷을 생성한 경우 다른 이름을 사용해야 합니다. 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 버킷 규제 및 제한을 참조하세요.

4. 생성을 선택합니다.

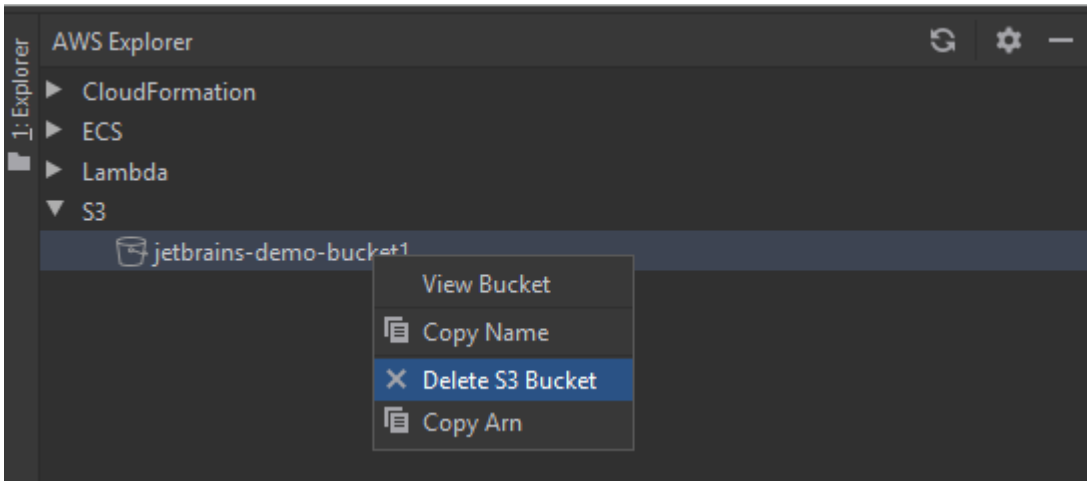
Amazon S3 버킷 보기

1. AWS Explorer가 열려 있지 않은 경우 이를 엽니다.
2. Amazon S3 노드를 클릭하여 버킷 목록을 확장합니다.
   - 현재 AWS 리전의 S3 버킷이 Amazon S3 노드 아래에 표시됩니다.

Amazon S3 버킷 삭제

1. AWS Explorer를 엽니다(열려 있지 않은 경우).
2. Amazon S3 노드를 클릭하여 버킷 목록을 확장합니다.

3. 삭제할 버킷을 마우스 오른쪽 단추로 클릭한 다음 S3 버킷 삭제를 선택합니다.



4. 버킷의 이름을 입력하여 삭제를 확인한 다음 확인을 선택합니다.

   - 버킷에 객체가 포함되어 있으면 삭제하기 전에 버킷이 비워집니다. 삭제가 완료되면 알림이 표시됩니다.

AWS Toolkit for JetBrains를 사용하여 Amazon S3 객체 작업

객체는 Amazon S3에 저장되는 기본 개체입니다. 객체는 객체 데이터와 메타데이터로 구성됩니다.

주제

- Amazon S3 버킷의 객체 보기
- IDE에서 객체 열기
- 객체 업로드
- 객체 다운로드
- 객체 삭제

Amazon S3 버킷의 객체 보기

이 절차에서 S3 버킷 뷰어를 엽니다. 뷰어를 사용해 폴더별로 그룹화된 Amazon S3 버킷의 객체를 보고, 업로드하고, 다운로드하고, 삭제할 수 있습니다.

1. AWS Explorer가 열려 있지 않은 경우 이를 엽니다.
2. 버킷의 객체를 보려면 다음 중 하나를 수행합니다.

   - 버킷 이름을 두 번 클릭합니다.

- 버킷의 이름을 마우스 오른쪽 단추로 클릭한 다음 버킷 보기를 선택합니다.

S3 버킷 뷰어는 버킷 이름, Amazon 리소스 이름(ARN) 및 생성 날짜에 대한 정보를 표시합니다. 버킷의 객체와 폴더는 아래 창에서 사용할 수 있습니다.

IDE에서 객체 열기

Amazon S3 버킷의 객체가 IDE에서 인식하는 파일 형식인 경우 읽기 전용 복사본을 다운로드하여 IDE에서 열 수 있습니다.

1. 다운로드할 객체를 찾으려면 S3 버킷 뷰어를 엽니다(Amazon S3 버킷의 객체 보기 참조).
2. 객체 이름을 두 번 클릭합니다.

해당 파일 형식에 대한 기본 IDE 창에서 파일이 열립니다.

객체 업로드

1. 객체를 업로드할 폴더를 찾으려면 S3 버킷 뷰어를 엽니다(Amazon S3 버킷의 객체 보기 참조).
2. 폴더를 마우스 오른쪽 단추로 클릭한 다음 업로드를 선택합니다.
3. 대화 상자에서 업로드할 파일을 선택합니다.

> **참고**
>
> 한 번에 여러 파일을 업로드할 수 있습니다. 디렉터리를 업로드할 수 없습니다.

4. 확인(OK)을 선택합니다.

객체 다운로드

1. 객체를 다운로드할 폴더를 찾으려면 S3 버킷 뷰어를 엽니다(Amazon S3 버킷의 객체 보기 참조).
2. 객체를 표시할 폴더를 선택합니다.
3. 객체를 마우스 오른쪽 단추로 클릭한 다음 다운로드를 선택합니다.
4. 대화 상자에서 다운로드 위치를 선택합니다.

> **참고**
>
> 여러 파일을 다운로드하는 경우 폴더 대신 경로 이름을 선택해야 합니다. 디렉터리를 다운로드할 수 없습니다.

5. 확인(OK)을 선택합니다.

> **참고**
>
> 파일이 다운로드 위치에 이미 있는 경우 덮어쓰거나 다운로드를 건너 뛰고 그대로 둘 수 있습니다.

객체 삭제

1. 삭제할 객체를 찾으려면 S3 버킷 뷰어를 엽니다(Amazon S3 버킷의 객체 보기 참조).
2. 객체를 선택한 후 다음 중 하나를 수행하여 객체를 삭제합니다.
   - Delete(삭제)를 누릅니다.
   - 마우스 오른쪽 버튼을 클릭한 다음 Delete(삭제)를 선택합니다.

> **참고**
>
> 한 번에 여러 객체를 선택하고 삭제할 수 있습니다.

3. 삭제를 확인하려면 삭제를 선택합니다.

AWS Toolkit for JetBrains를 사용하여 AWS 서버리스 애플리케이션 작업

다음 항목에서는 AWS Toolkit for JetBrains을 사용하여 AWS 계정에서 AWS 서버리스 애플리케이션 작업을 하는 방법을 설명합니다.

주제

- AWS Toolkit for JetBrains를 사용하여 AWS 서버리스 애플리케이션 생성

- AWS Toolkit for JetBrains에서 AWS SAM 애플리케이션 동기화
- AWS Toolkit for JetBrains를 사용하여 AWS 서버리스 애플리케이션 설정 변경(업데이트)
- AWS Toolkit for JetBrains를 사용하여 AWS 서버리스 애플리케이션 삭제

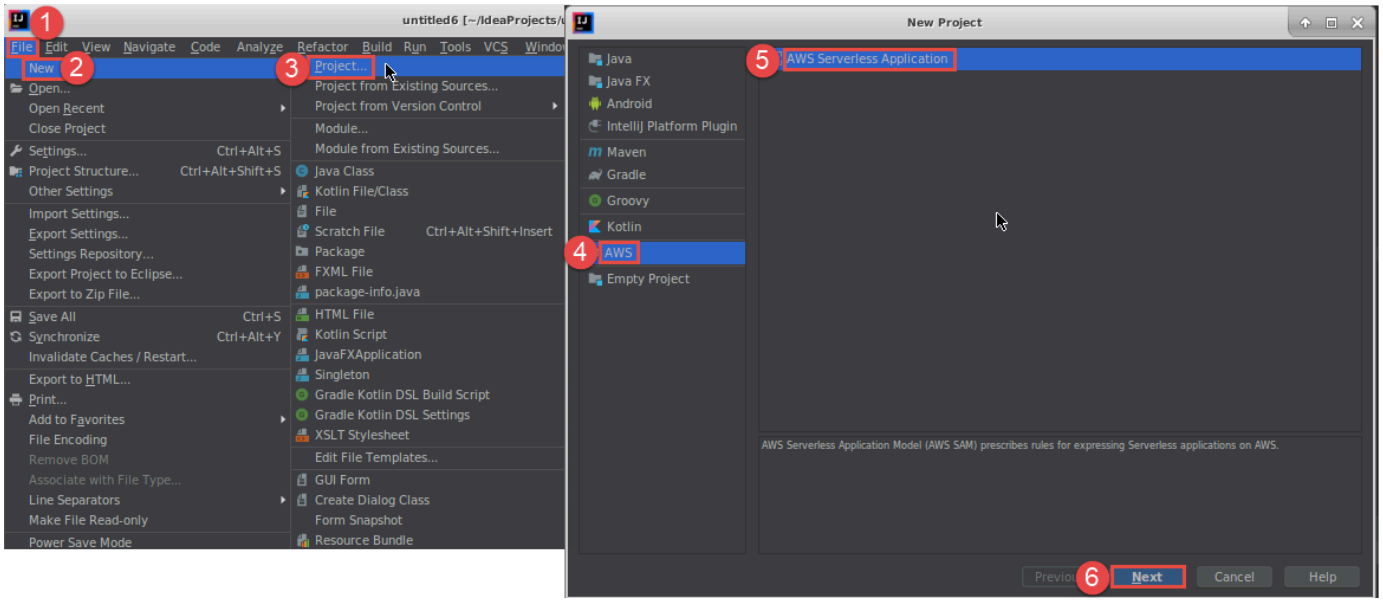
AWS Toolkit for JetBrains를 사용하여 AWS 서버리스 애플리케이션 생성

이 절차를 완료하려면 먼저 AWS Toolkit를 설치하고 아직 연결하지 않은 경우 AWS 계정에 처음 연결해야 합니다. 그런 다음 IntelliJ IDEA, PyCharm, WebStorm 또는 JetBrains Rider가 이미 실행되고 있는 상태에서 다음을 수행합니다.

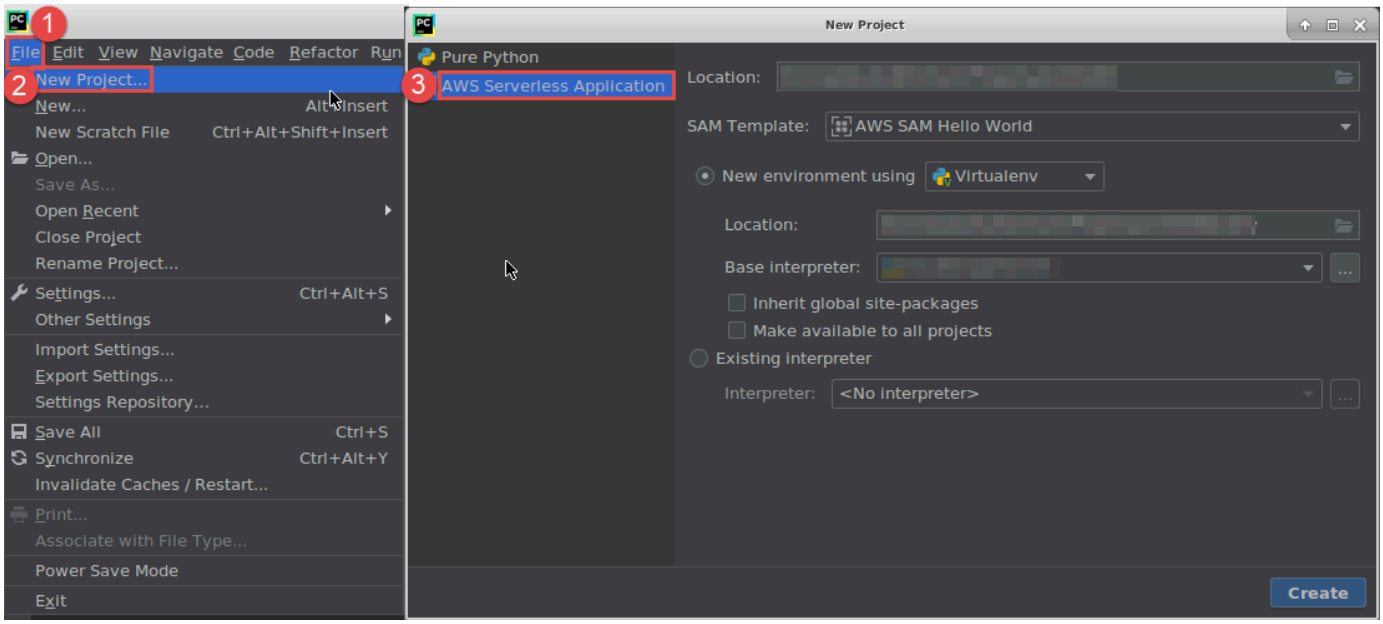
1. IntelliJ IDEA, PyCharm, WebStorm 또는 JetBrains Rider가 이미 실행되고 있는 상태에서 다음 중 하나를 수행합니다.

   - IntelliJ IDEA 또는 WebStorm의 경우 파일, 신규, 프로젝트를 선택합니다.
   - PyCharm의 경우 파일, 새 프로젝트를 선택합니다.
   - JetBrains Rider의 경우 새 솔루션에 대해 파일, 신규를 선택합니다. 또는 탐색기 도구 창에서 기존 솔루션을 마우스 오른쪽으로 클릭하고 추가, 새 프로젝트를 선택합니다.

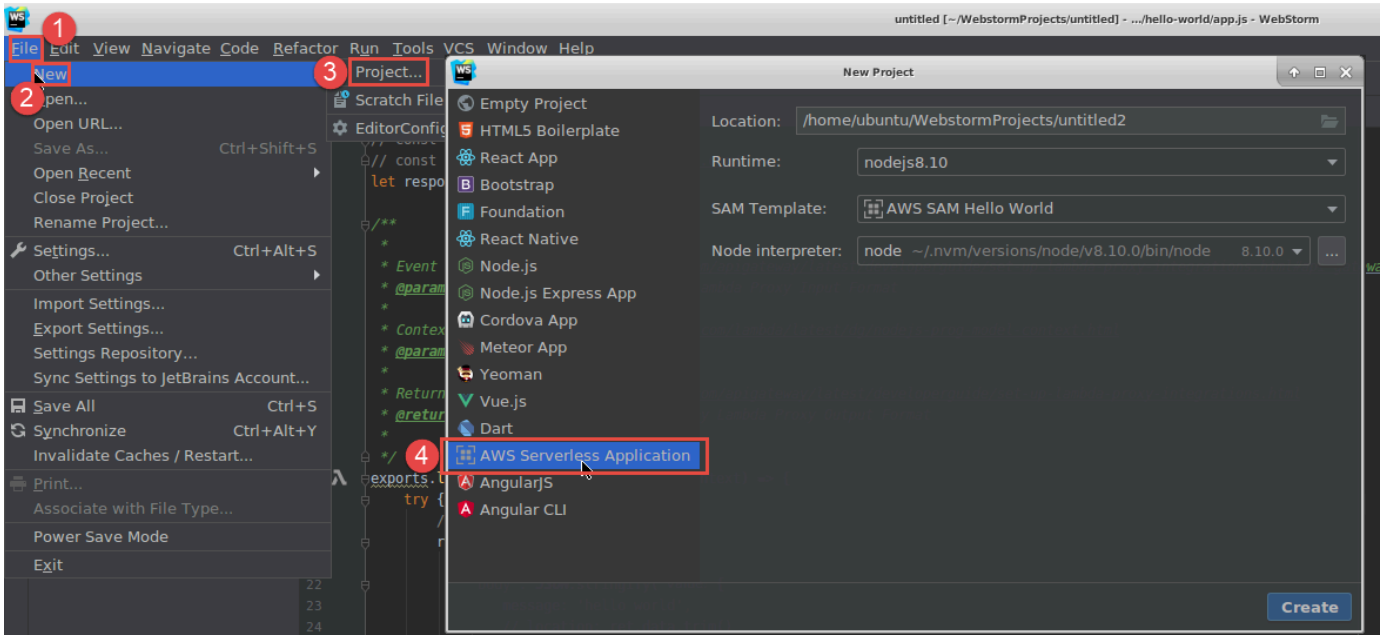
2. IntelliJ IDEA의 경우 AWS, AWS 서버리스 애플리케이션을 선택하고 다음을 선택합니다.



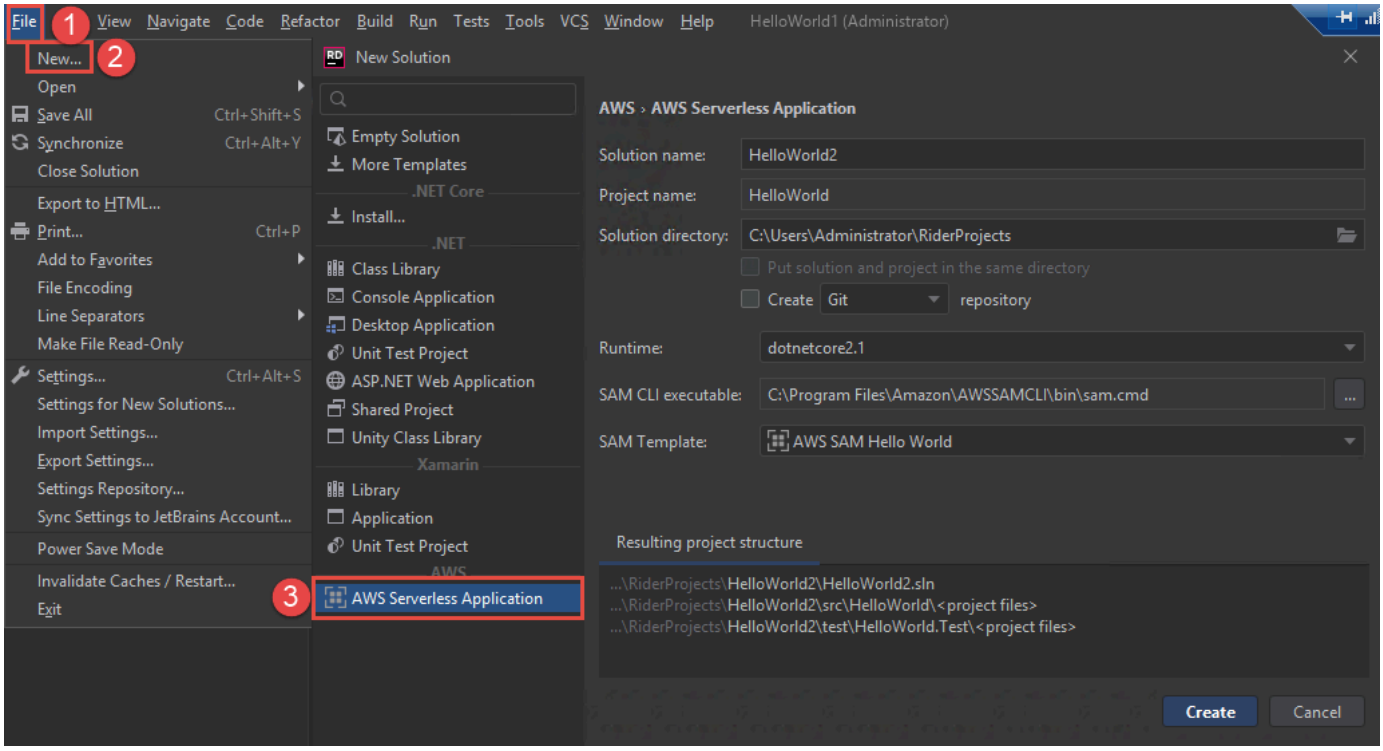
PyCharm의 경우 AWS Serverless Application을 선택합니다.



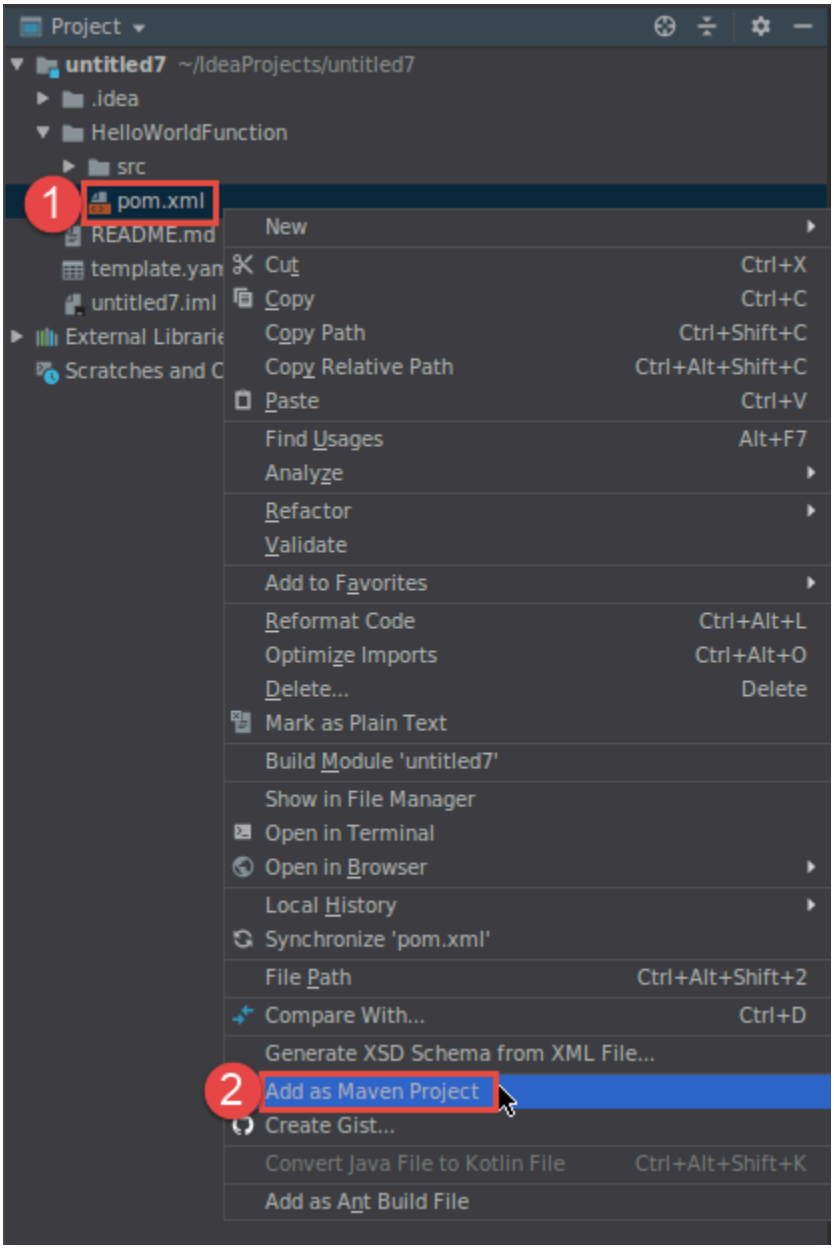
WebStorm의 경우 AWS Serverless Application을 선택합니다.



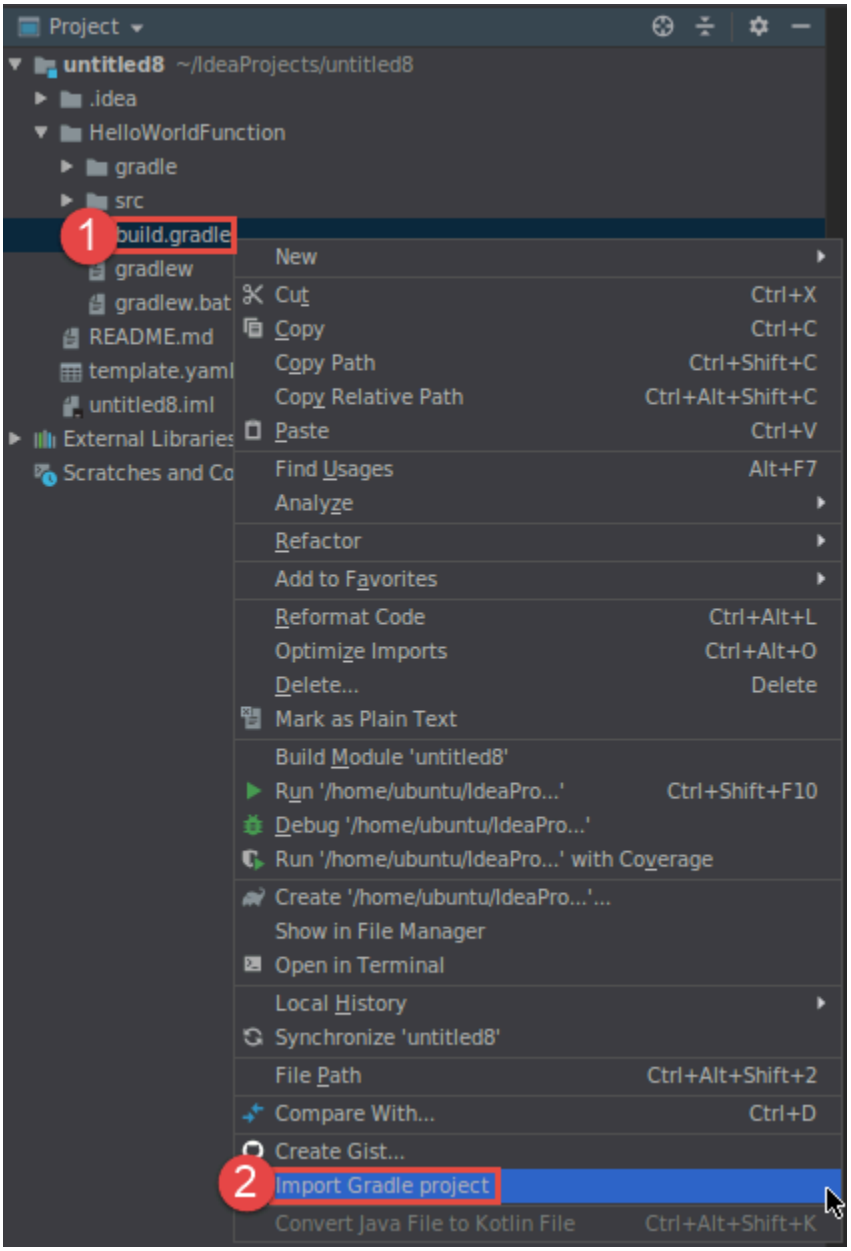
JetBrains Rider의 경우 AWS Serverless Application을 선택합니다.



3. 새 프로젝트 대화 상자 또는 새 솔루션 대화 상자(JetBrains Rider의 경우)를 완료하고 완료(IntelliJ IDEA의 경우) 또는 생성(PyCharm, WebStorm 또는 JetBrains Rider의 경우)을 선택합니다. AWS Toolkit for JetBrains은 프로젝트를 만들고 새 프로젝트에 서버리스 애플리케이션의 코드 파일을 추가합니다.
4. IntelliJ IDEA를 사용하는 경우 프로젝트 도구 창이 이미 열려 있고 서버리스 애플리케이션의 파일을 포함한 프로젝트가 표시되는 경우 다음 중 하나를 실행합니다.
   - Maven 기반 프로젝트라면 프로젝트의 `pom.xml` 파일을 마우스 오른쪽으로 클릭하고 Maven 프로젝트 추가를 선택합니다.



- Gradle 기반 프로젝트라면 프로젝트의 `build.gradle` 파일을 마우스 오른쪽으로 클릭하고 Gradle 프로젝트 가져오기를 선택합니다.



Gradle에서 모듈 가져오기 대화 상자를 완료하고 확인을 선택합니다.

서버리스 애플리케이션을 생성한 후 해당 애플리케이션에 포함된 AWS Lambda 함수의 로컬 버전을 실행(간접적으로 호출)하거나 디버깅할 수 있습니다.

서버리스 애플리케이션을 배포할 수도 있습니다. 배포한 후에는 배포된 애플리케이션의 일부인 Lambda 함수의 원격 버전을 간접적으로 호출할 수 있습니다.

AWS Toolkit for JetBrains에서 AWS SAM 애플리케이션 동기화

AWS Serverless Application Model(AWS SAM) `sam sync`는 서버리스 애플리케이션의 변경 사항을 자동으로 식별한 다음 해당 변경 사항을 구축하고 AWS 클라우드에 배포하는 가장 좋은 방법을 선택하는 AWS SAM-CLI 명령 배포 프로세스입니다. 인프라를 변경하지 않고 애플리케이션 코드만 변경한 경우 AWS SAM Sync는 CloudFormation 스택을 재배포하지 않고 애플리케이션을 업데이트합니다.

`sam sync` 및 AWS SAM CLI 명령에 대한 자세한 내용은 AWS Serverless Application Model 개발자 안내서의 AWS SAM CLI command reference 주제를 참조하세요.

다음 섹션에서는 AWS SAM Sync 작업을 시작하는 방법을 설명합니다.

필수 조건

AWS SAM Sync 작업 전에 다음 사전 조건을 충족해야 합니다.

- 작동 중인AWS SAM 애플리케이션이 있습니다. AWS SAM 애플리케이션 생성에 대한 자세한 내용은 이 사용 설명서의 AWS SAM 작업 주제를 참조하세요.
- AWS SAM CLI. 버전 1.78.0 이상이 설치되어 있어야 합니다. AWS SAM CLI 설치에 대한 자세한 내용은 AWS Serverless Application Model 개발자 안내서의 Installing the AWS SAM CLI 주제를 참조하세요.
- 애플리케이션이 개발 환경에서 실행 중이어야 합니다.

> **Note**
>
> 본 속성이 아닌 속성이 있는 AWS Lambda 함수가 포함된 서버리스 애플리케이션을 동기화하고 배포하려면 배포 전에 AWS Lambda 함수와 연결된 AWS SAM 템플릿 파일에 선택적 속성을 설정해야 합니다.
> AWS Lambda 속성에 대해 자세히 알아보려면 GitHub에서 AWS Serverless Application Model User Guide의 AWS::Serverless::Function 섹션을 참조하세요.

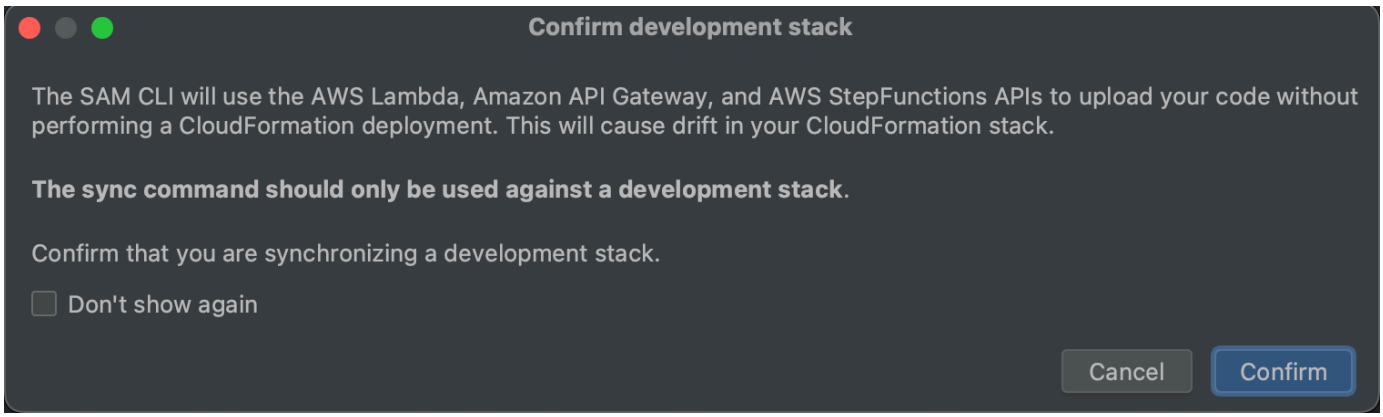
시작하기

AWS SAM Sync 작업을 시작하려면 다음 절차를 완료하세요.

> **Note**
>
> AWS 리전이 서버리스 애플리케이션과 연결된 위치로 설정되어 있는지 확인하세요.

AWS Toolkit for JetBrains에서 AWS 리전을 변경하는 방법에 대해 자세히 알아보려면 이 사용 설명서의 AWS 리전 간 전환 주제를 참조하세요.

1. 프로젝트 도구 창의 서버리스 애플리케이션 프로젝트에서 `template.yaml` 파일에 대한 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 엽니다.
2. `template.yaml` 컨텍스트 메뉴에서 서버리스 애플리케이션 동기화(이전에 배포)를 선택하여 개발 스택 확인 대화 상자를 엽니다.
3. 개발 스택에서 작업 중인지 확인하여 서버리스 애플리케이션 동기화 대화 상자를 엽니다.



4. 서버리스 애플리케이션 동기화 대화 상자의 단계를 완료한 다음 동기화를 선택하여 AWS SAM Sync 프로세스를 시작합니다. 서버리스 애플리케이션 동기화 대화 상자에 대해 자세히 알아보려면 아래에 있는 the section called “서버리스 애플리케이션 동기화 대화 상자” 섹션을 참조하세요.
5. 동기화 프로세스 중에 AWS Toolkit for JetBrains 실행 창이 배포 상태로 업데이트됩니다.
6. 동기화에 성공하면 CloudFormation 스택의 이름이 AWS Explorer에 추가됩니다.

   동기화에 실패하면 JetBrains 실행 창 또는 CloudFormation 이벤트 로그에서 문제 해결 세부 정보를 확인할 수 있습니다. CloudFormation 이벤트 로그 보기에 대해 자세히 알아보려면 이 사용 설명서의 스택에 대한 이벤트 로그 보기 주제를 참조하세요.

서버리스 애플리케이션 동기화 대화 상자

서버리스 애플리케이션 동기화 대화 상자는 AWS SAM 동기화 프로세스를 지원합니다. 다음 섹션은 여러 대화 상자 구성 요소 각각에 대한 설명과 세부 정보입니다.

스택 생성 또는 스택 업데이트

필수: 새 배포 스택을 생성하려면 제공된 필드에 이름을 입력하여 서버리스 애플리케이션 배포를 위한 CloudFormation 스택을 생성하고 설정합니다.

또는 기존 CloudFormation 스택에 배포하려면 AWS 계정과 연결된 자동으로 채워지는 스택 목록에서 스택 이름을 선택합니다.

템플릿 파라미터

선택 사항: 프로젝트 `template.yaml` 파일에서 감지된 파라미터 목록을 채웁니다. 파라미터 값을 지정하려면 값 열에 있는 제공된 텍스트 필드에 새 파라미터 값을 입력합니다.

S3 Bucket

필수: CloudFormation 템플릿을 저장할 기존 Amazon Simple Storage Service(S3) 버킷을 선택하려면 목록에서 해당 버킷을 선택합니다.

스토리지용 새 Amazon S3 버킷을 생성하고 사용하려면 생성을 선택하고 프롬프트를 따릅니다.

ECR 리포지토리

필수, 이미지 패키지 유형 작업 시에만 표시됨: 서버리스 애플리케이션 배포를 위해 기존 Amazon Elastic Container Registry(Amazon ECR) 리포지토리 URI를 선택합니다.

AWS Lambda 패키지 유형에 대한 자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 배포 패키지를 참조하세요.

CloudFormation 기능

필수: 스택을 생성할 때 CloudFormation에서 사용할 수 있는 기능을 선택합니다.

태그

선택 사항: 파라미터에 태그를 지정하려면 제공된 텍스트 필드에 기본 태그를 입력합니다.

컨테이너 내부에 함수 빌드

선택 사항, Docker 필요: 이 옵션을 선택하면 배포 전 로컬 Docker 컨테이너 안에 서버리스 애플리케이션 함수가 빌드됩니다. 이 옵션은 함수가 기본적으로 컴파일된 종속 항목 또는 프로그램이 있는 패키지에 종속되는 경우에 유용합니다.

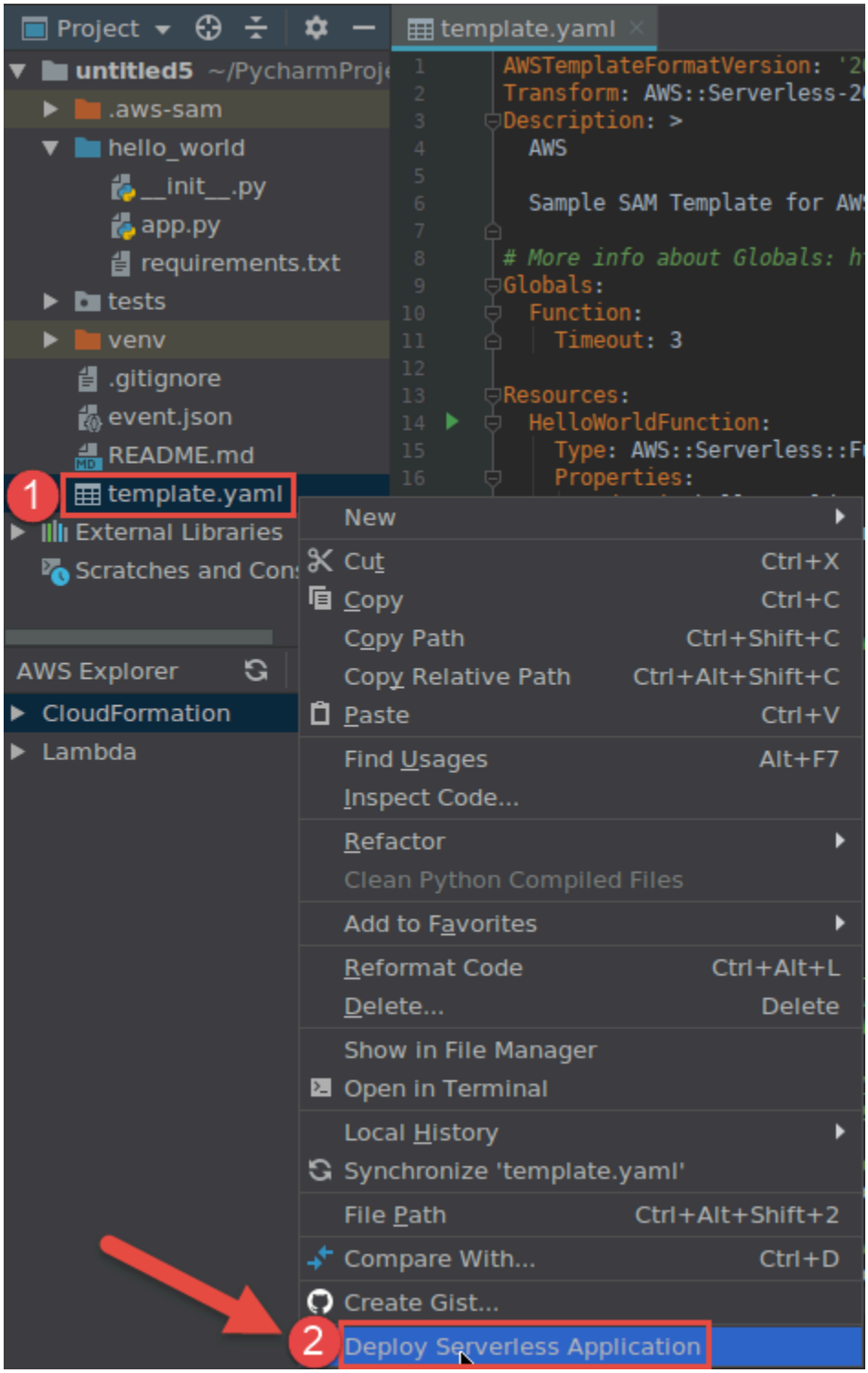
자세한 내용은 AWS Serverless Application Model 개발자 안내서의 Building applications 주제를 참조하세요.

AWS Toolkit for JetBrains를 사용하여 AWS 서버리스 애플리케이션 설정 변경(업데이트)

변경하려는 AWS 서버리스 애플리케이션을 아직 배포하지 않은 경우 먼저 해당 애플리케이션을 배포해야 합니다.

> **Note**
>
> AWS Lambda 함수가 포함된 서버리스 애플리케이션을 배포하고 해당 함수를 기본값이 아닌 속성 또는 선택적 속성으로 배포하려면 먼저 함수의 해당 AWS SAM 템플릿 파일 (예: 프로젝트 내에서 파일 이름이 `template.yaml`인 파일)에 해당 속성을 설정해야 합니다. 사용 가능한 속성 목록은 GitHub의 awslabs/serverless-application-model 리포지토리에 있는 AWS::Serverless::Function을 참조하세요.

1. 프로젝트 도구 창이 이미 열려 있고 서버리스 애플리케이션의 파일이 포함된 프로젝트가 표시된 상태에서 프로젝트의 `template.yaml` 파일을 엽니다. 파일의 내용을 변경하여 새 설정을 반영한 다음 파일을 저장하고 닫습니다.
2. 서버리스 애플리케이션을 배포하기 위해 다른 AWS 리전으로 전환해야 하는 경우 지금 수행하세요.
3. 프로젝트의 `template.yaml` 파일을 마우스 오른쪽 단추로 클릭한 다음 서버리스 애플리케이션 배포를 선택합니다.



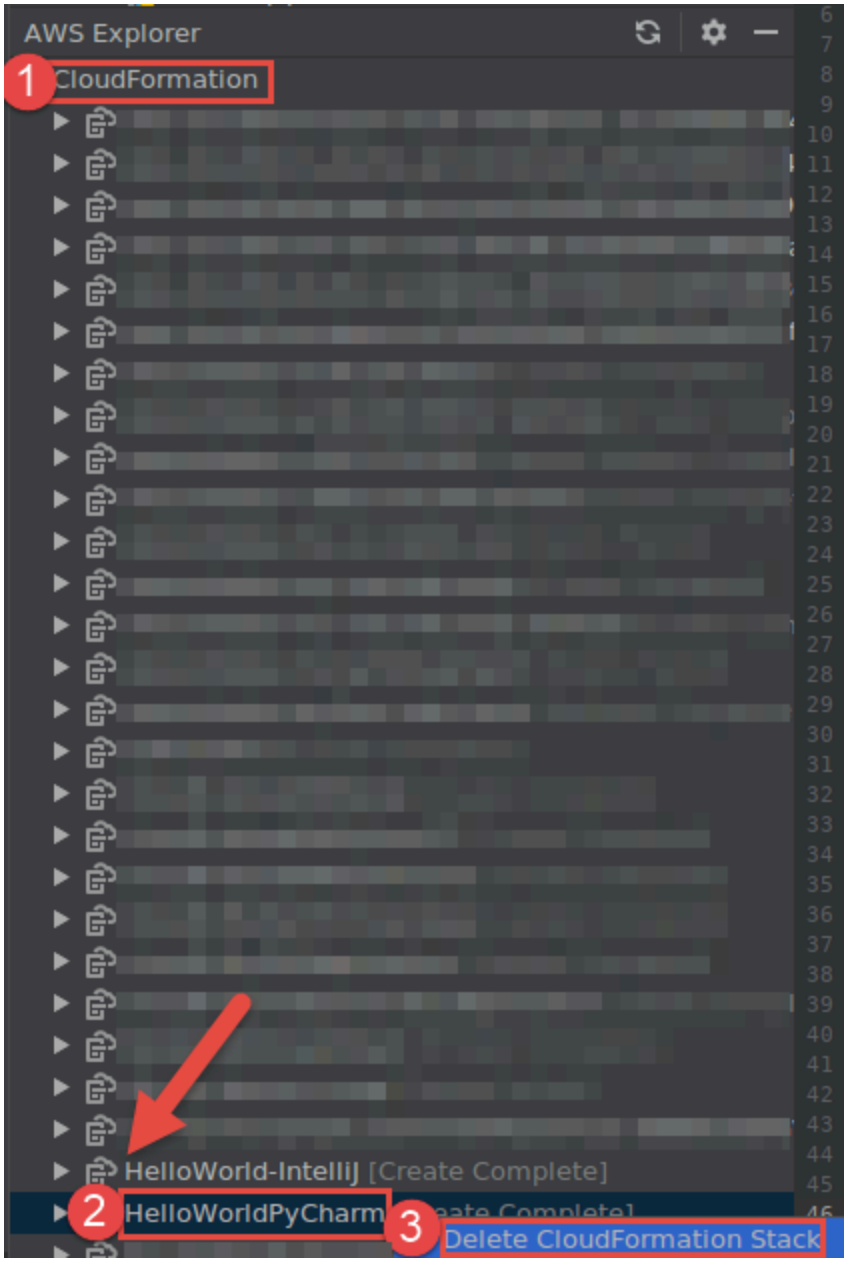
4. 서버리스 애플리케이션 배포 대화 상자를 완료한 다음 배포를 선택합니다. AWS Toolkit for JetBrains에서는 배포에 대한 해당 AWS CloudFormation 스택을 업데이트합니다.

   배포가 실패하면 스택에 대한 이벤트 로그를 확인하여 이유를 파악할 수 있습니다.

AWS Toolkit for JetBrains를 사용하여 AWS 서버리스 애플리케이션 삭제

AWS 서버리스 애플리케이션을 삭제하기 전에 먼저 배포해야 합니다.

1. AWS Explorer가 열려 있지 않은 경우 이를 엽니다. 서버리스 애플리케이션이 포함된 다른 AWS 리전으로 전환해야 하는 경우 지금 수행하세요.
2. CloudFormation을 확장합니다.
3. 삭제할 서버리스 애플리케이션을 포함한 AWS CloudFormation 스택의 이름을 마우스 오른쪽 단추로 클릭한 다음 CloudFormation 스택 삭제를 선택합니다.



4. 스택의 이름을 입력하여 삭제를 확인한 다음 확인을 선택합니다. 스택 삭제에 성공하면 AWS Toolkit for JetBrains가 AWS Explorer의 CloudFormation 목록에서 해당 스택 이름을 제거합니다. 스택 삭제에 실패하면 스택에 대한 이벤트 로그를 확인하여 이유를 파악할 수 있습니다.

AWS Toolkit for JetBrains에서 Amazon Simple Queue Service 대기열 작업

다음 주제에서는 AWS Toolkit for JetBrains에서 Amazon Simple Queue Service 서비스 작업을 수행하는 방법을 설명합니다.

주제

Amazon Simple Queue Service 대기열 작업

다음 주제에서는 AWS Toolkit for JetBrains를 사용하여 Amazon Simple Queue Service 대기열 및 메시지 작업을 수행하는 방법을 설명합니다.

표준과 선입후출(FIFO)은 AWS Toolkit for JetBrains에서 Amazon SQS를 사용하여 전송할 수 있는 두 가지 종류의 메시지입니다.

Amazon SQS 대기열 생성

1. AWS Toolkit for JetBrains에서 AWS Explorer를 확장하여 AWS 서비스를 봅니다.
2. AWS Explorer에서 Amazon SQS 서비스의 컨텍스트 메뉴(마우스 오른쪽 버튼 클릭)를 연 다음 대기열 생성...을 선택합니다.
3. 대기열 이름을 제공하고 대기열 유형을 선택합니다.

> **Note**
>
> 대기열 유형에 대한 자세한 내용은 Amazon Simple Queue Service 개발자 안내서의 Amazon SQS standard queues와 Amazon SQS FIFO (First-In-First-Out) queues 주제를 참조하세요.

4. 생성을 선택합니다.

Amazon SQS 메시지 보기

1. AWS Toolkit for JetBrains에서 AWS Explorer를 확장하여 AWS 서비스를 봅니다.
2. AWS Explorer에서 Amazon SQS 서비스를 확장하여 기존 대기열 목록을 봅니다.
3. 보려는 대기열을 마우스 오른쪽 버튼으로 클릭하고 메시지 보기를 선택합니다.
4. 메시지 보기를 선택하여 대기열의 메시지를 봅니다.

Amazon SQS 대기열 속성 편집

1. AWS Toolkit for JetBrains에서 AWS Explorer를 확장하여 AWS 서비스를 봅니다.
2. AWS Explorer에서 Amazon SQS 서비스를 확장하여 기존 대기열 목록을 봅니다.
3. 편집하려는 대기열을 마우스 오른쪽 버튼으로 클릭하고 대기열 속성 편집...을 선택합니다.
4. 열리는 대기열 속성 편집 대화 상자에서 대기열 속성을 검토하고 수정합니다. 자세한 내용은 Amazon Simple Queue Service 개발자 안내서의 Configuring queue parameters (console)를 참조하세요.

표준 메시지 전송

1. AWS Toolkit for JetBrains에서 AWS Explorer를 확장하여 AWS 서비스를 봅니다.
2. AWS Explorer에서 Amazon SQS 서비스를 확장하여 기존 대기열 목록을 봅니다.
3. 메시지 전송 대기열을 마우스 오른쪽 버튼으로 클릭하고 메시지 전송을 선택합니다.
4. 메시지를 입력하고 전송을 선택합니다. 메시지를 전송한 후 메시지 ID가 포함된 확인 메시지가 표시됩니다.

FIFO 메시지 전송

1. AWS Toolkit for JetBrains에서 AWS Explorer를 확장하여 AWS 서비스를 봅니다.
2. AWS Explorer에서 Amazon SQS 서비스를 확장하여 기존 대기열 목록을 봅니다.
3. 메시지 전송 대기열을 마우스 오른쪽 버튼으로 클릭하고 메시지 전송을 선택합니다.
4. 메시지, 그룹 ID 및 선택적 중복 제거 ID를 입력합니다.

> **Note**
>
> 중복 제거 ID가 제공되지 않은 경우 하나 생성됩니다.

5. Send(전송)를 선택합니다. 메시지를 전송한 후 메시지 ID가 포함된 확인 메시지가 표시됩니다.

Amazon SQS 대기열 삭제

1. 대기열을 삭제하기 전에 대기열이 비어 있는지 확인합니다. 자세한 내용은 Amazon Simple Queue Service 개발자 안내서의 Confirming that a queue is empty를 참조하세요.
2. AWS Toolkit for JetBrains에서 AWS Explorer를 확장하여 AWS 서비스를 봅니다.
3. AWS Explorer에서 Amazon SQS 서비스를 확장하여 기존 대기열 목록을 봅니다.
4. Amazon SQS를 마우스 오른쪽 버튼으로 클릭하고 대기열 삭제...를 선택합니다.
5. 대기열을 삭제할지 확인하고 삭제 대화 상자에서 확인을 선택합니다.

AWS Toolkit for JetBrains에서 AWS Lambda와 함께 Amazon SQS 사용

다음 절차에서는 AWS Toolkit for JetBrains에서 Amazon SQS 대기열을 Lambda 트리거로 구성하는 방법을 자세히 설명합니다.

Lambda 트리거로 Amazon SQS 대기열 구성

1. AWS Toolkit for JetBrains에서 AWS Explorer를 확장하여 AWS 서비스를 봅니다.
2. AWS Explorer에서 Amazon SQS 서비스를 확장하여 기존 대기열 목록을 봅니다.
3. 작업을 수행하려는 대기열을 마우스 오른쪽 버튼으로 클릭하고 Lambda 트리거 구성을 선택합니다.
4. 대화 상자의 드롭다운 메뉴에서 트리거할 Lambda 함수를 선택합니다.
5. 구성(Configure)을 선택합니다.
6. Lambda 함수에 Amazon SQS가 함수를 실행하는 데 필요한 IAM 권한이 없는 경우 도구 키트가 Lambda 함수의 IAM 역할에 추가할 수 있는 최소 정책을 생성합니다.

   정책 추가를 선택합니다.

대기열을 구성하고 나면 해당하는 오류 메시지를 포함하여 적용된 변경 사항에 대한 상태 메시지가 나타납니다.

AWS Toolkit for JetBrains에서 Amazon SNS와 함께 Amazon SQS 사용

다음 절차에서는 AWS Toolkit for JetBrains를 사용하여 표준 Amazon SQS 대기열을 Amazon SNS 주제에 구독하는 방법을 자세히 설명합니다.

> **Note**
>
> FIFO Amazon SQS 대기열에서는 Amazon SNS 주제를 구독할 수 없습니다.

표준 Amazon SQS 대기열에서 Amazon SNS 주제 구독

1. AWS Toolkit for JetBrains에서 AWS Explorer를 확장하여 AWS 서비스를 봅니다.
2. AWS Explorer에서 Amazon SQS 서비스를 확장하여 기존 대기열 목록을 봅니다.
3. 작업을 수행하려는 대기열을 마우스 오른쪽 버튼으로 클릭하고 SNS 주제 구독...을 선택합니다.
4. 대화 상자의 드롭다운 메뉴에서 Amazon SNS 주제를 선택한 다음 구독을 선택합니다.

리소스 작업

AWS Explorer에 기본적으로 나열되는 AWS 서비스에 액세스하는 것 외에도 리소스(Resources)로 이동하여 수백 개의 리소스 중에서 선택하여 인터페이스에 추가할 수도 있습니다. AWS에서 리소스란 작업할 수 있는 엔터티입니다. 추가할 수 있는 리소스 중 일부는 Amazon AppFlow, Amazon Kinesis Data Streams,AWS IAM 역할, Amazon VPC 및 Amazon CloudFront 배포입니다.

선택한 후 리소스로 이동하고 리소스 유형을 확장하여 해당 유형에 사용 가능한 리소스를 나열할 수 있습니다. 예를 들어 `AWS::Lambda::Function` 리소스 유형을 선택하면 다양한 기능, 해당 속성 및 특성을 정의하는 리소스에 액세스할 수 있습니다.

리소스 유형을 리소스(Resources)에 추가한 후 다음과 같은 방법으로 해당 리소스와 상호 작용할 수 있습니다.

- 이 리소스 유형에 대해 현재 AWS 리전에서 사용할 수 있는 기존 리소스 목록을 봅니다.
- 리소스를 설명하는 JSON 파일의 읽기 전용 버전을 봅니다.
- 리소스의 리소스 식별자를 복사합니다.
- 리소스 모델링을 위한 리소스 유형 및 스키마(JSON 및 YAML 형식)의 목적을 설명하는 AWS 설명서를 봅니다.

- 스키마에 맞는 JSON 형식의 템플릿을 편집하고 저장하여 새 리소스를 생성합니다.*
- 기존 리소스를 업데이트하거나 삭제합니다.*

> **Important**
>
> *AWS Toolkit for JetBrains의 현재 릴리스에서는 리소스를 생성, 편집 및 삭제하는 옵션이 실험 기능입니다. 실험 기능은 계속 테스트되고 업데이트되므로 가용성 문제가 있을 수 있습니다. AWS Toolkit for JetBrains에서 예고 없이 삭제될 수도 있습니다.
> 리소스에 실험 기능 사용을 허용하려면 JetBrains IDE에서 설정 창을 열고 도구를 확장한 후 AWS, 실험 기능을 선택합니다. 리소스를 생성, 업데이트, 삭제할 수 있도록 하려면 JSON 리소스 수정을 선택합니다.
> 자세한 내용은 실험 기능 작업 섹션을 참조하세요.

리소스 액세스를 위한 IAM 권한

AWS 서비스와 연결된 리소스에 액세스하려면 특정 AWS Identity and Access Management 권한이 필요합니다. 예를 들어 사용자 또는 역할과 같은 IAM 엔터티는 `AWS::Lambda::Function` 리소스에 액세스하기 위해 Lambda 권한이 필요합니다.

서비스 리소스에 대한 권한 외에도 IAM 엔터티에는 AWS Toolkit for JetBrains가 대신 AWS Cloud Control API 작업을 직접적으로 호출하도록 허용하는 권한이 필요합니다. Cloud Control API 작업을 통해 IAM 사용자 또는 역할이 원격 리소스에 액세스하고 업데이트할 수 있습니다.

권한을 부여하는 가장 쉬운 방법은 AWS 관리형 정책, PowerUserAccess를 도구 키트 인터페이스를 사용하여 이러한 API 작업을 호출하는 IAM 엔터티에 연결하는 것입니다. 이 관리형 정책은 API 작업 호출을 포함하여 애플리케이션 개발 작업을 수행할 수 있는 다양한 권한을 부여합니다.

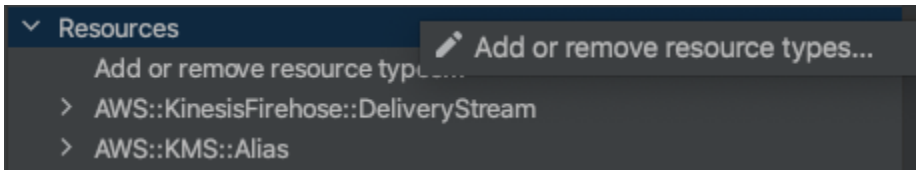
원격 리소스에서 허용 가능한 API 작업을 정의하는 특정 권한은 AWS Cloud Control API 사용 설명서를 참조하세요.

기존 리소스 추가 및 상호 작용

1. AWS Explorer에서 리소스를 마우스 오른쪽 버튼으로 클릭하고 리소스 추가 또는 제거를 선택합니다.

   설정 창의 추가 탐색기 리소스에는 선택할 수 있는 리소스 유형 목록이 표시됩니다.

> **Note**
>
> 리소스 아래에 있는 리소스 추가 또는 제거 노드를 두 번 클릭하여 리소스 유형 목록을 표시할 수도 있습니다.



2. 추가 탐색기 리소스에서 AWS Explorer에 추가할 리소스 유형을 선택하고 Return 키를 누르거나 확인을 선택하여 확인합니다.

   선택한 리소스 유형이 리소스 아래에 나열됩니다.

   > **Note**
   >
   > 이미 AWS Explorer에 리소스 유형을 추가한 다음 해당 유형에 대한 확인란 선택을 취소한 경우 확인을 선택한 후에는 해당 유형이 더 이상 리소스 아래에 나열되지 않습니다. 현재 선택된 리소스 유형만 AWS Explorer에 표시됩니다.

3. 리소스 유형에 대해 이미 존재하는 리소스를 보려면 해당 유형의 항목을 확장합니다.

   사용 가능한 리소스 목록이 해당 리소스 유형 아래에 표시됩니다.

4. 특정 리소스와 상호 작용하려면 해당 이름을 마우스 오른쪽 버튼으로 클릭하고 다음 옵션 중 하나를 선택합니다.

   - 리소스 보기: 리소스를 설명하는 JSON 형식 템플릿의 읽기 전용 버전을 봅니다.

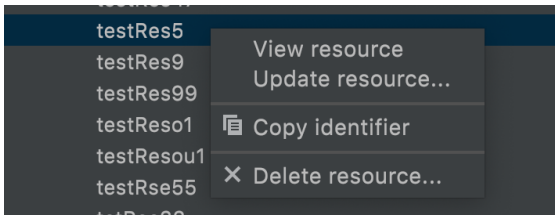
     템플릿이 표시된 후 편집을 선택하여 템플릿을 변경할 수 있습니다(필수 ??? 활성화 시).

     > **Note**
     >
     > 리소스를 두 번 클릭하여 볼 수도 있습니다.

   - 식별자 복사: 특정 리소스의 식별자를 클립보드에 복사합니다. (예를 들어 `AWS::DynamoDB::Table` 속성을 사용하여 `TableName` 리소스를 식별할 수 있습니다.)

- 리소스 업데이트: JetBrains 편집기에서 리소스의 JSON 형식 템플릿을 편집합니다. 자세한 내용은 리소스 생성 및 업데이트 섹션을 참조하세요.
- 리소스 삭제: 표시된 대화 상자에서 삭제를 확인하여 리소스를 삭제합니다. 리소스 삭제는 현재 이 버전의 AWS Toolkit for JetBrains에서 ???입니다.

> **Warning**
>
> 리소스를 삭제하면 해당 리소스를 사용하는 AWS CloudFormation 스택이 업데이트되지 않습니다. 이 업데이트 문제를 해결하려면 리소스를 다시 생성하거나 스택의 CloudFormation 템플릿에서 리소스에 대한 참조를 제거해야 합니다. 자세한 내용은 이 지식 센터 문서를 참조하세요.



리소스 생성 및 업데이트

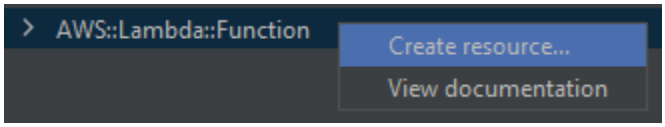
> **Important**
>
> 리소스 생성 및 업데이트는 현재 이 버전의 AWS Toolkit for JetBrains에서 ???입니다.

새 리소스를 생성하려면 리소스 목록에 리소스 유형을 추가한 다음 리소스와 해당 속성을 정의하는 JSON 형식의 템플릿을 편집해야 합니다.

예를 들어, `AWS::SageMaker::UserProfile` 리소스 유형에 속하는 리소스는 Amazon SageMaker Studio용 사용자 프로파일을 생성하는 템플릿으로 정의됩니다. 이 사용자 프로파일 리소스를 정의하는 템플릿은 `AWS::SageMaker::UserProfile`의 리소스 유형 스키마에 맞아야 합니다. 예를 들어, 속성이 누락되거나 올바르지 않아 템플릿이 스키마를 준수하지 않는 경우 리소스를 생성하거나 업데이트할 수 없습니다.

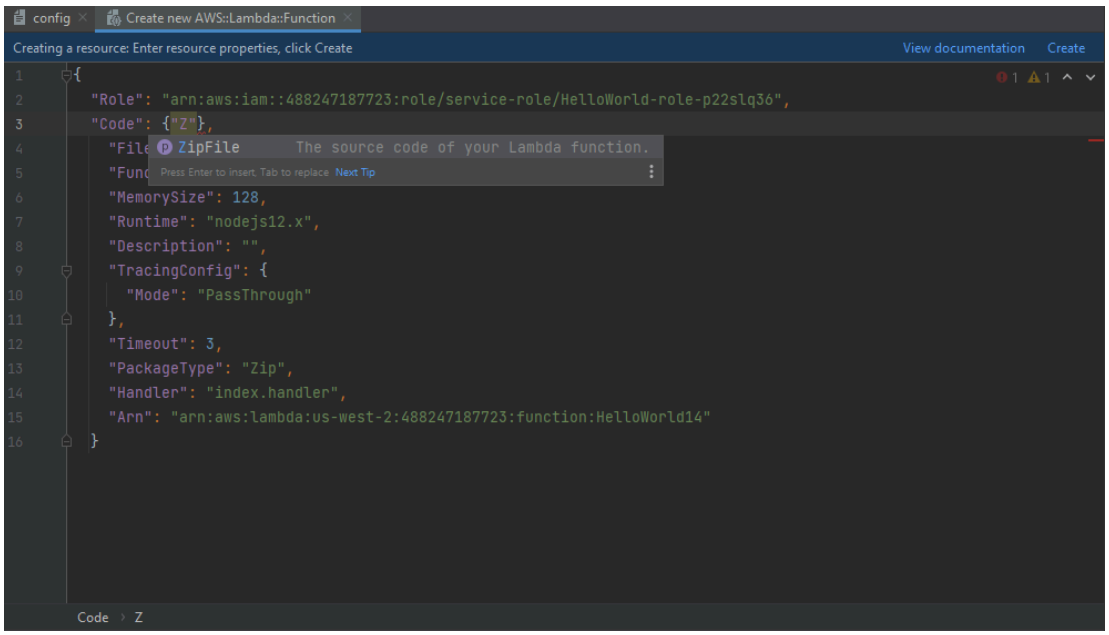
1. 리소스를 마우스 오른쪽 버튼으로 클릭하고 리소스 추가 또는 제거를 선택하여 생성하려는 리소스에 대한 리소스 유형을 추가합니다.

2. 리소스 아래에 리소스 유형을 추가한 후 해당 이름을 마우스 오른쪽 버튼으로 클릭하고 리소스 생성을 선택합니다. 설명서 보기를 선택하여 리소스를 모델링하는 방법에 대한 정보에 액세스할 수도 있습니다.



3. 편집기에서 리소스 템플릿을 구성하는 속성을 정의하기 시작합니다. 자동 완성 기능은 템플릿의 스키마에 맞는 속성 이름을 제안합니다. 템플릿이 JSON 구문에 완전히 맞으면 오류 개수가 녹색 확인 표시로 바뀝니다. 스키마에 대한 자세한 내용을 보려면 설명서 보기를 선택합니다.

> **Note**
>
> 템플릿은 기본 JSON 구문뿐만 아니라 리소스 유형을 모델링하는 스키마에 맞아야 합니다. 원격 리소스를 생성하거나 업데이트하려고 하면 스키마 모델에 대해 템플릿이 검증됩니다.



4. 리소스 선언을 마친 후 생성을 선택하여 템플릿을 검증하고 리소스를 원격 AWS 클라우드에 저장합니다. 기존 리소스를 수정하는 경우 업데이트를 선택합니다.

   템플릿이 해당 스키마에 따라 리소스를 정의하는 경우 리소스가 생성되었음을 확인하는 메시지가 표시됩니다. 리소스가 이미 있는 경우 리소스가 업데이트되었음을 확인하는 메시지가 표시됩니다.

생성된 리소스는 리소스 유형 제목 아래의 목록에 추가됩니다.

5. 파일에 오류가 있는 경우 리소스를 생성하거나 업데이트할 수 없음을 설명하는 메시지가 표시됩니다. 이벤트 로그를 열어 수정해야 할 템플릿 요소를 식별합니다.

AWS Toolkit for JetBrains에 대한 사용자 인터페이스 참조

AWS Toolkit for JetBrains 사용자 인터페이스 작업에 대한 도움말은 다음 주제를 참조하세요.

주제

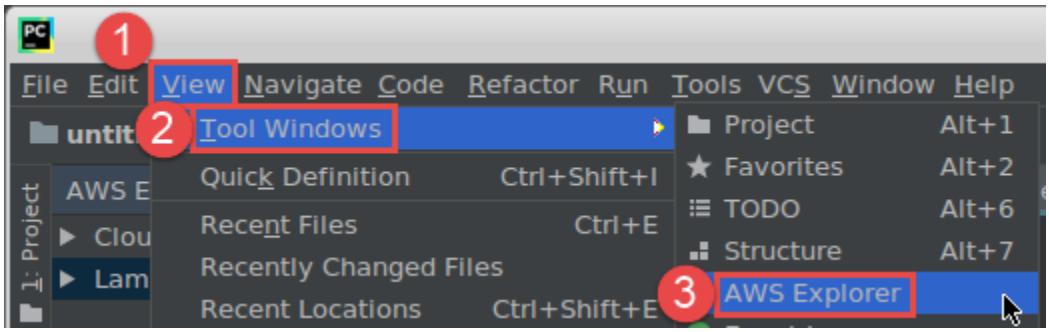
AWS 탐색기

AWS Explorer를 사용하면 AWS Toolkit for JetBrains의 여러 기능에 편리하게 액세스할 수 있습니다. 여기에는 도구 키트에서 AWS 계정으로의 연결 관리, AWS 리전 전환, 계정의 AWS Lambda 함수 및 AWS CloudFormation 스택 작업 등이 포함됩니다.

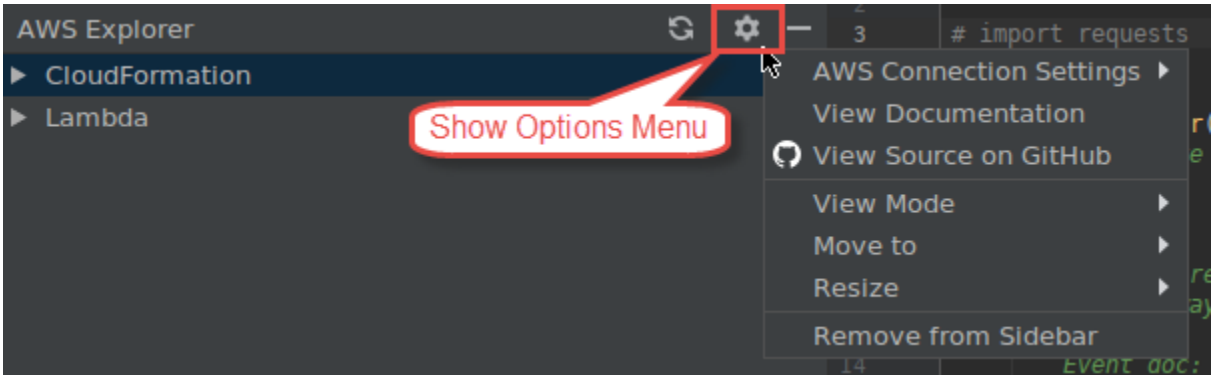
AWS Explorer를 열려면 AWS Toolkit for JetBrains가 설치되고 IntelliJ IDEA, PyCharm, WebStorm 또는 JetBrains Rider가 실행되고 있는 상태에서 다음 중 하나를 수행합니다.

- 도구 창 모음에서 AWS 탐색기를 선택합니다.

- 기본 메뉴에서 보기, 도구 창, AWS 탐색기를 선택합니다.



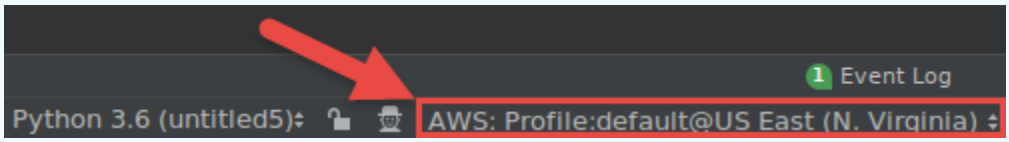
AWS Explorer에서 다음 옵션에 대한 설정 아이콘(옵션 표시 메뉴)을 선택합니다.



AWS 연결 설정

다음 옵션이 포함되어 있습니다.

- AWS 리전 목록 - AWS Toolkit for JetBrains가 선택한 리전을 사용합니다. 도구 키트에서 다른 리전을 사용하게 하려면 나열된 다른 리전을 선택합니다.
- 최근 보안 인증 목록 - AWS Toolkit for JetBrains에서 AWS 계정으로의 최근 연결을 나열합니다. 도구 키트는 선택한 연결을 사용합니다. 도구 키트에서 다른 최근 연결을 사용하게 하려면 해당 연결의 이름을 선택합니다.
- 모든 보안 인증 - AWS Toolkit for JetBrains에서 AWS 계정으로의 가능한 모든 연결을 나열합니다. 도구 키트는 선택한 연결을 사용합니다. 도구 키트에서 다른 연결을 사용하게 하려면 해당 연결의 이름을 선택합니다. 다른 연결 태스크를 수행하려면 AWS 보안 인증 편집 파일을 선택합니다.

> **Note**
>
> 상태 표시줄의 AWS 연결 설정 영역에는 AWS 계정 연결 및 AWS Toolkit for JetBrains에서 현재 사용 중인 리전이 표시됩니다.
>
> 
>
>
> 옵션 표시 메뉴와 동일한 AWS 연결 설정 옵션을 보려면 이 영역을 선택합니다.

설명서 보기

AWS Toolkit for JetBrains 사용 설명서(이 설명서)로 이동합니다.

GitHub에서 소스 보기

GitHub 웹 사이트의 aws/aws-toolkit-jetbrains 리포지토리로 이동합니다.

보기 모드

편집기 또는 다른 도구 창에서 작업할 때 빠르게 액세스하고 공간을 절약할 수 있도록 AWS 탐색기 도구 창을 조정합니다.

IntelliJ IDEA 뷰 모드는 IntelliJ IDEA 도움말 웹 사이트의 Tool window view modes를 참조하세요.

PyCharm 뷰 모드는 PyCharm 도움말 웹 사이트의 Tool window view modes를 참조하세요.

WebStorm 뷰 모드는 WebStorm 도움말 웹 사이트의 Tool window view modes를 참조하세요.

JetBrains Rider 뷰 모드는 JetBrains Rider 도움말 웹 사이트의 Tool window view modes를 참조하세요.

다음으로 이동

AWS Explorer 도구 창을 IntelliJ IDEA, PyCharm, WebStorm 또는 JetBrains Rider의 다른 위치로 이동합니다.

크기 조정

AWS 탐색기 도구 창의 크기를 변경합니다.

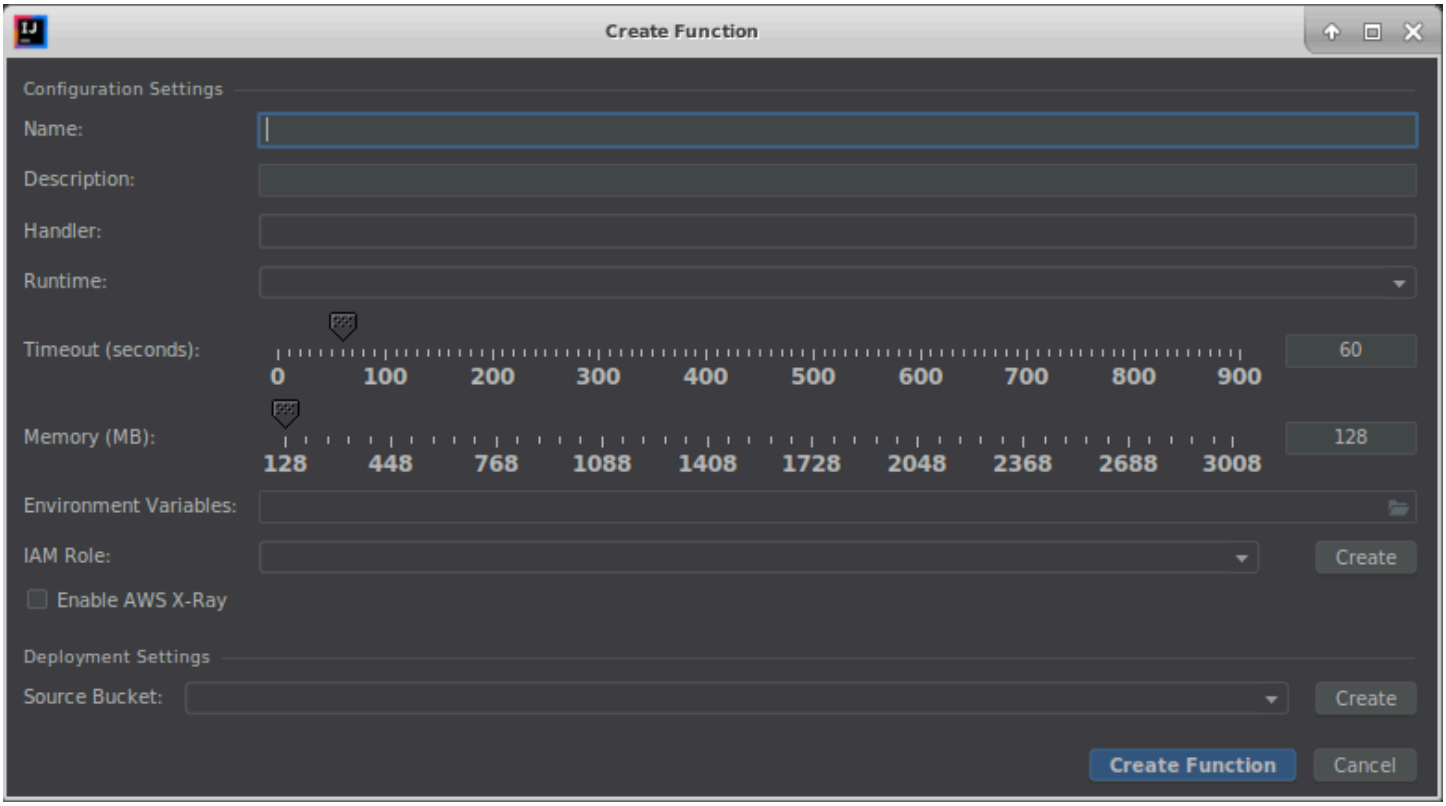
사이드바에서 제거

도구 창 모음에서 AWS Explorer 도구 창을 제거합니다. 다시 표시하려면 기본 메뉴 모음에서 보기, 도구 창, AWS 탐색기를 선택합니다.

또한 AWS Explorer를 사용하여 Lambda 함수 작업을 수행하고 AWS 계정의 AWS CloudFormation 스택 작업을 수행할 수도 있습니다.

함수 생성 대화 상자

AWS Toolkit for JetBrains의 함수 생성 대화 상자는 독립 실행형 AWS Lambda Lambda 함수를 생성할 때 표시됩니다.



함수 생성 대화 상자에는 다음 항목이 포함되어 있습니다.

이름

(필수) 함수의 이름입니다. 대문자(A~Z), 소문자(a~z), 숫자(0~9), 하이픈(-) 및 밑줄(_)만 포함할 수 있습니다. 이름은 64자 미만이어야 합니다.

설명

(선택 사항) 함수에 대한 의미 있는 설명입니다.

핸들러

(필수) Java, Python, Node.js, C#에 대한 해당 함수 핸들러의 ID입니다.

런타임

(필수) 사용할 Lambda 런타임의 ID입니다.

시간 초과(초)

(필수) Lambda가 함수를 중지하기 전에 실행을 허용하는 시간입니다. 최대 900초(15분)까지 지정합니다.

메모리(MB)

(필수) 함수가 실행될 때 사용할 수 있는 메모리 양입니다. 128MB에서 3,008MB 사이의 크기를 64MB 단위로 지정합니다.

환경 변수

(선택 사항) Lambda 함수에서 사용할 환경 변수로, 키-값 페어로 지정됩니다. 환경 변수를 추가, 변경 또는 삭제하려면 폴더 아이콘을 선택한 다음 화면의 지침을 따릅니다.

IAM 역할

(필수) Lambda에서 함수에 사용할 연결된 AWS 계정의 사용 가능한 Lambda 실행 역할을 선택합니다. 계정에서 실행 역할을 생성하고 Lambda가 해당 역할을 대신 사용하도록 하려면 생성을 선택한 다음 화면의 지침을 따릅니다.

AWS X-Ray 활성화

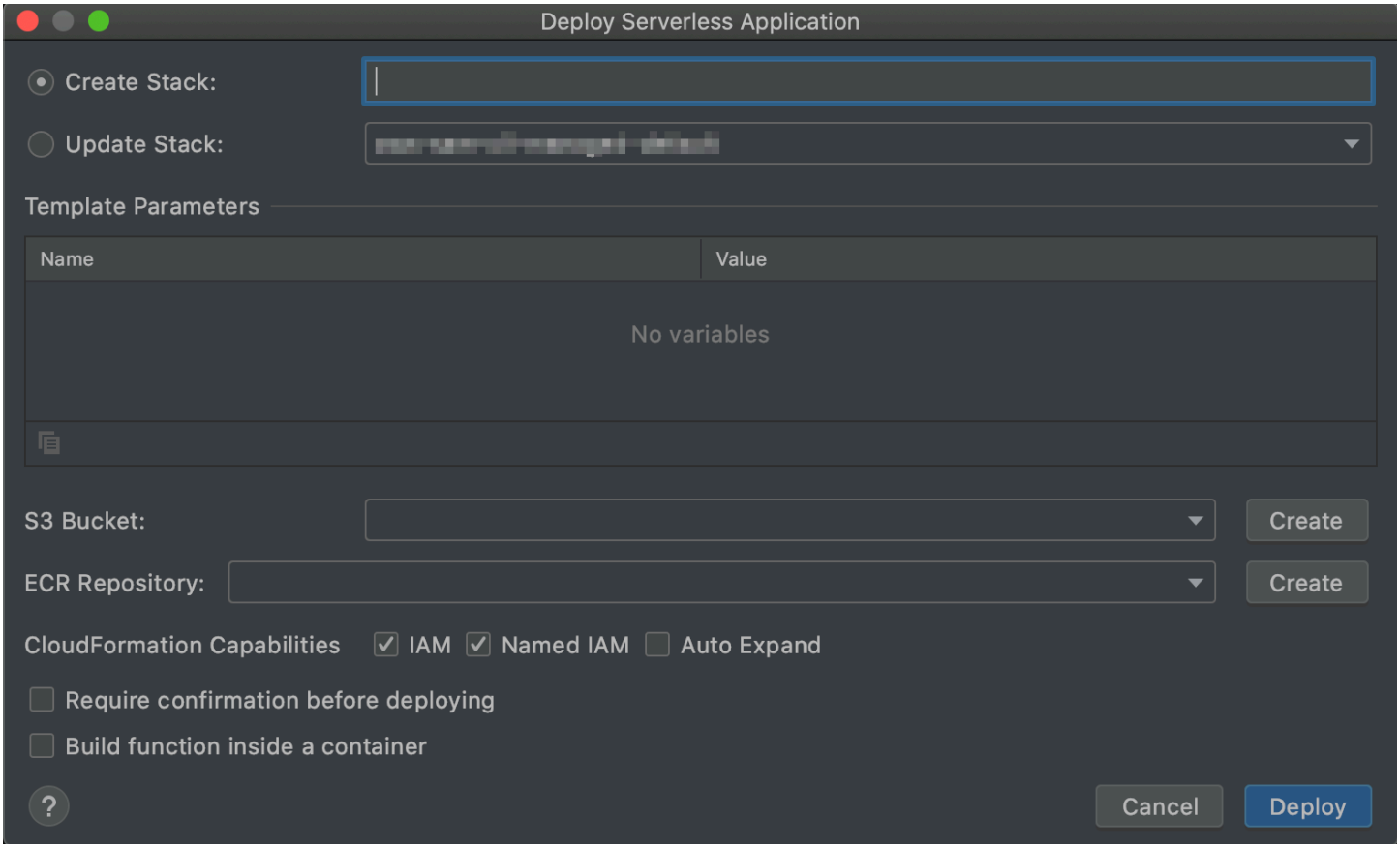
(선택 사항) 이 항목을 선택하면 AWS X-Ray가 함수의 성능 문제를 감지, 분석 및 최적화할 수 있습니다. X-Ray는 Lambda와 함수를 구성하는 모든 업스트림 또는 다운스트림 서비스에서 메타데이터를 수집합니다. X-Ray는 이러한 메타데이터를 이용하여 성능 병목 현상, 지연 시간 스파이크 및 함수 성능에 영향을 미치는 기타 문제를 보여주는 상세한 서비스 그래픽을 생성합니다.

원본 버킷

(필수) AWS Serverless Application Model(AWS SAM) 명령줄 인터페이스(CLI)에서 Lambda에 함수를 배포하는 데 사용할 연결된 AWS 계정의 사용 가능한 Amazon Simple Storage Service(S3) 버킷을 선택합니다. 계정에 Amazon S3 버킷을 생성하고 AWS SAM CLI에서 해당 버킷을 대신 사용하도록 하려면 생성을 선택한 다음 화면의 지침을 따릅니다.

서버리스 애플리케이션 배포 대화 상자

AWS Toolkit for JetBrains의 Deploy Serverless Application 대화 상자는 AWS 서버리스 애플리케이션을 배포할 때 표시됩니다.



서버리스 애플리케이션 배포 대화 상자에는 다음 항목이 포함되어 있습니다.

스택 만들기

(필수) AWS CloudFormation에서 연결된 AWS 계정에 생성할 AWS Serverless Application Model(AWS SAM) 명령줄 인터페이스(CLI)에 대한 스택의 이름을 제공합니다. 그런 다음 AWS SAM CLI는 이 스택을 사용하여 AWS 서버리스 애플리케이션을 배포합니다.

스택 업데이트

(필수) AWS SAM CLI에서 AWS 서버리스 애플리케이션을 배포하는 데 사용할 연결된 AWS 계정의 기존 CloudFormation 스택 이름을 선택합니다.

> **Note**
>
> 스택 생성 또는 스택 업데이트가 필요합니다(둘 다는 아님).

템플릿 파라미터

(선택 사항) AWS Toolkit for JetBrains이 해당 프로젝트의 `template.yaml` 파일에서 감지하는 모든 파라미터입니다. 매개변수 값을 지정하려면 매개변수 옆의 값 열에서 상자를 선택하고 값을 입력한 다음 Enter 키를 누릅니다. 자세한 내용은 AWS CloudFormation 사용 설명서의 파라미터를 참조하세요.

S3 Bucket

(필수) AWS SAM CLI에서 AWS 서버리스 애플리케이션을 배포하는 데 사용할 연결된 AWS 계정의 기존 Amazon Simple Storage Service(S3) 버킷을 선택합니다. 계정에 Amazon S3 버킷을 생성하고 AWS SAM CLI에서 해당 버킷을 대신 사용하도록 하려면 생성을 선택한 다음 화면의 지침을 따릅니다.

ECR 리포지토리

(`Image` 패키지 유형의 경우에만 필수) AWS SAM CLI에서 AWS 서버리스 애플리케이션을 배포하는 데 사용할 연결된 AWS 계정의 기존 Amazon Elastic Container Registry(Amazon ECR) 리포지토리 URI를 선택합니다. AWS Lambda 패키지 유형에 대한 자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 배포 패키지를 참조하세요.

배포 전에 확인 필요

(선택 사항) 이 옵션을 선택하면 CloudFormation은 CloudFormation에서 스택의 현재 변경 세트를 실행하여 해당 스택의 생성 또는 업데이트가 완료될 때까지 기다리도록 지시합니다. 이 변경 세트를 실행하지 않으면 AWS 서버리스 애플리케이션이 배포 단계로 이동하지 않습니다.

컨테이너 내부에 함수를 빌드합니다.

(선택 사항) 이 옵션을 선택하면 AWS SAM CLI는 배포 전에 Lambda와 같은 Docker 컨테이너 내부에 로컬로 서버리스 애플리케이션의 기능을 빌드합니다. 이 기능은 함수가 기본적으로 컴파일된 종속성 또는 프로그램을 가진 패키지에 의존하는 경우에 유용합니다. 자세한 내용은 AWS Serverless Application Model 개발자 안내서의 애플리케이션 빌드를 참조하세요.

새 프로젝트 대화 상자

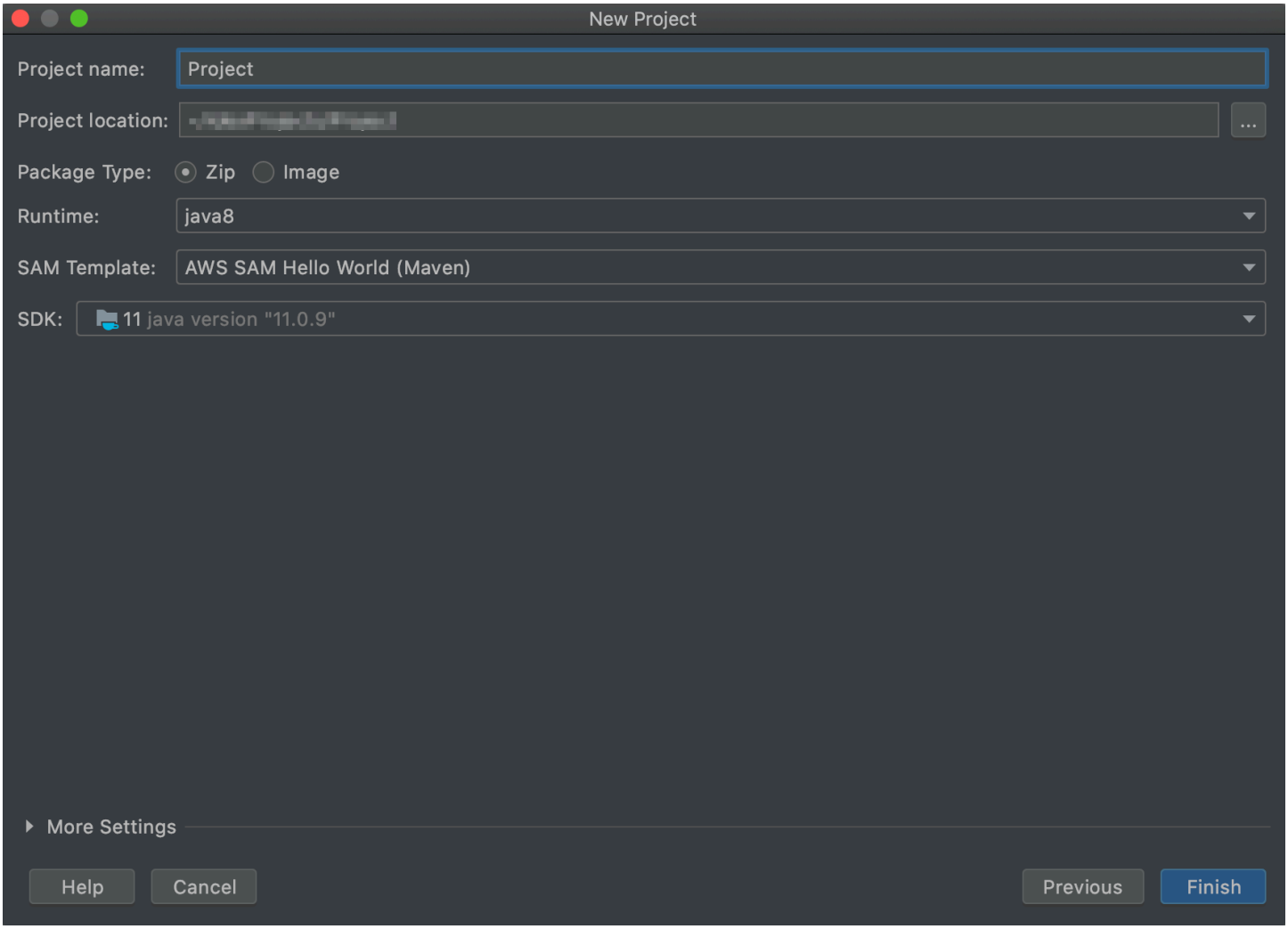
AWS 서버리스 애플리케이션을 생성하면 AWS Toolkit for JetBrains의 새 프로젝트 대화 상자가 표시됩니다.

주제

- 새 프로젝트 대화 상자(IntelliJ IDEA, PyCharm 및 WebStorm)
- 새 프로젝트 대화 상자(JetBrains Rider)

새 프로젝트 대화 상자(IntelliJ IDEA, PyCharm 및 WebStorm)

> **Note**
>
> 다음 스크린샷은 IntelliJ IDEA의 새 프로젝트 대화 상자를 보여주지만 필드 설명은 PyCharm과 WebStorm에도 적용됩니다.



새 프로젝트 대화 상자에는 다음 항목이 포함되어 있습니다.

프로젝트 이름

(필수) 프로젝트의 이름입니다.

프로젝트 위치

(필수) IntelliJ IDEA가 프로젝트를 생성하는 위치입니다.

패키지 유형

(필수) AWS Lambda 함수의 배포 패키지 유형(`Zip` 또는 `Image`)입니다. `Zip` 패키지 유형과 `Image` 패키지 유형의 차이점에 대한 자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 배포 패키지를 참조하세요.

런타임

(필수) 사용할 Lambda 런타임의 ID입니다.

SAM 템플릿

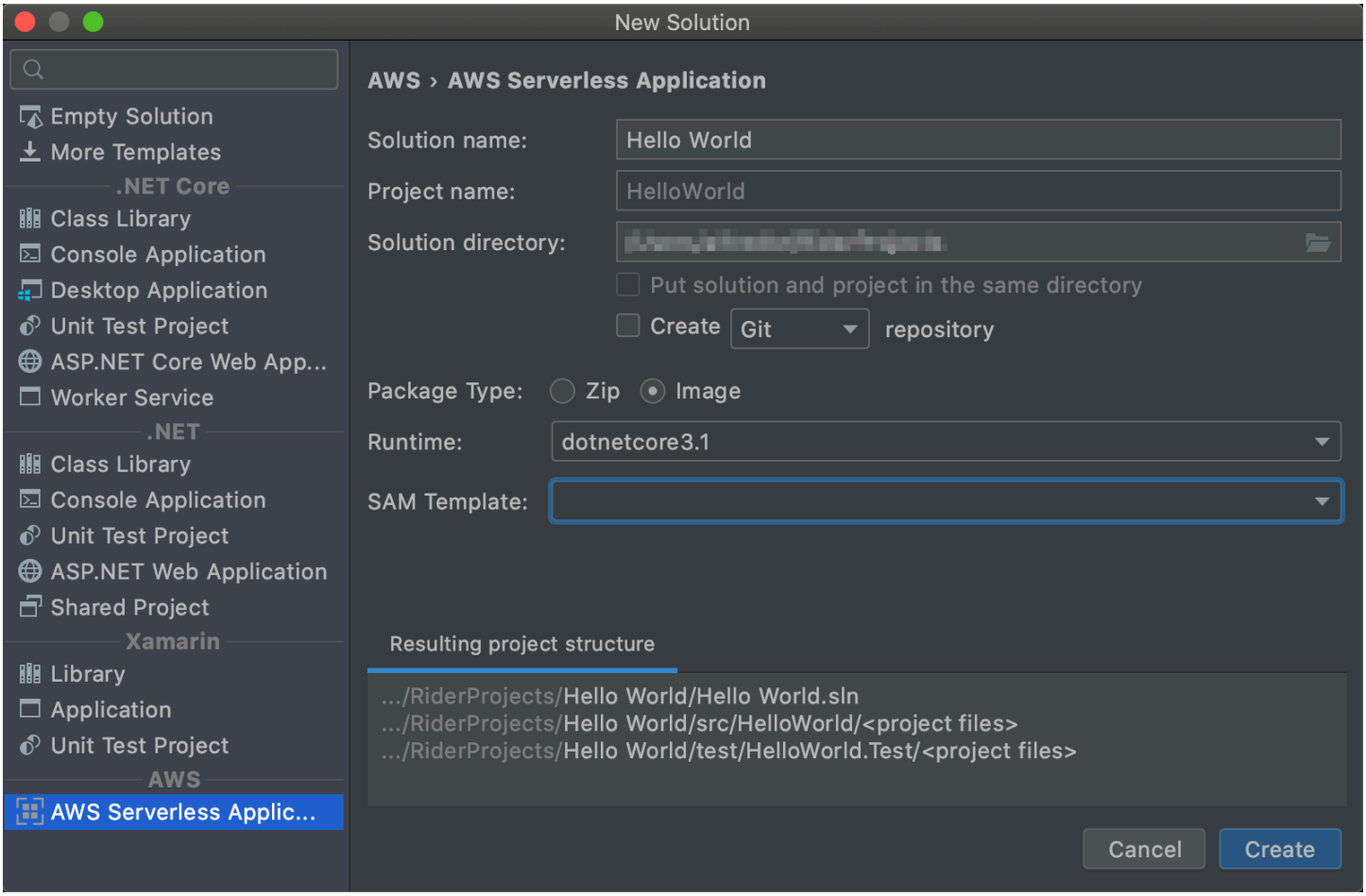
(필수) 사용할 AWS Serverless Application Model(AWS SAM) 템플릿의 이름입니다.

프로젝트 SDK

(필수) 사용할 Java 개발 키트(JDK)입니다. 자세한 내용은 IntelliJ IDEA 도움말 웹 사이트의 Java Development Kit (JDK)를 참조하세요.

새 프로젝트 대화 상자(JetBrains Rider)

> **Note**
>
> 새 솔루션을 생성하면 이 대화 상자에 새 프로젝트 대신 새 솔루션이라는 제목이 포함됩니다. 그러나 대화 상자의 내용은 동일합니다.



새 프로젝트 대화 상자에는 다음 항목이 포함되어 있습니다.

솔루션 이름

(필수) 솔루션의 이름입니다.

프로젝트 이름

(필수) 프로젝트의 이름입니다.

솔루션 디렉터리

(필수) 솔루션 디렉터리의 경로입니다.

솔루션과 프로젝트를 동일한 디렉터리에 넣으십시오.

(선택 사항) 이 항목을 선택하면 솔루션 파일이 프로젝트 파일과 같은 위치에 배치됩니다.

리포지토리 생성

(선택 사항) 이 항목을 선택하면 지정된 제공업체를 사용하여 프로젝트에 대한 원격 리포지토리가 생성됩니다.

패키지 유형

(필수) Lambda 함수의 패키지 유형(`Zip` 또는 `Image`)입니다. `Zip` 패키지 유형과 `Image` 패키지 유형의 차이점에 대한 자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 배포 패키지를 참조하세요.

런타임

(필수) 사용할 Lambda 런타임의 ID입니다.

SAM 템플릿

(필수) 사용할 AWS SAM 템플릿의 이름입니다.

결과 프로젝트 구조

(편집 불가) 생성된 프로젝트의 디렉터리 및 파일 경로입니다.

실행/디버깅 구성 대화 상자

AWS Toolkit for JetBrains의 실행/디버깅 구성 대화 상자는 로컬, 원격 또는 Amazon Elastic Container Service(Amazon ECS) 클러스터에서 실행/디버깅 구성을 변경하려고 할 때마다 표시됩니다.

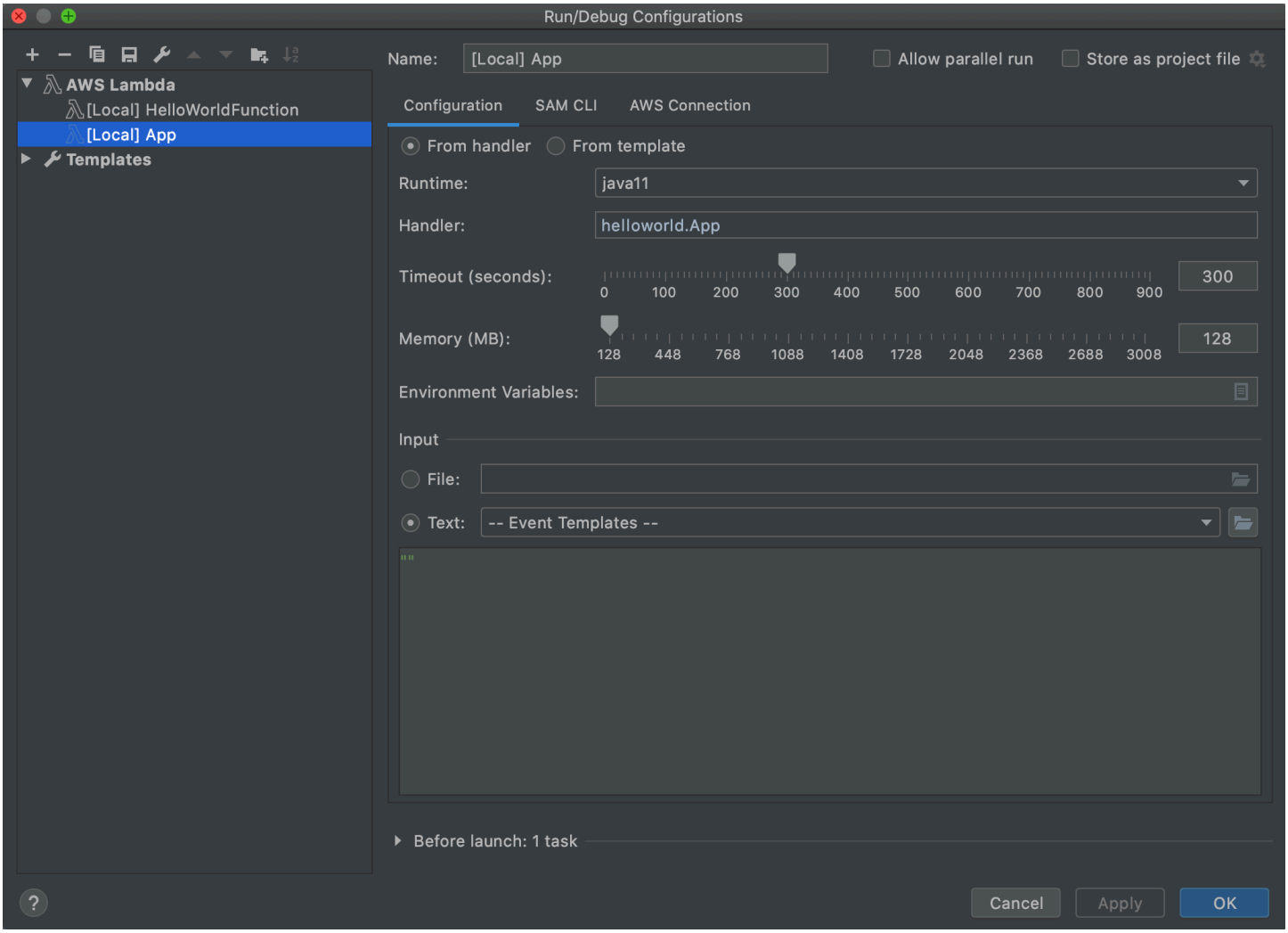
주제

실행/디버깅 구성 대화 상자(로컬 함수 설정)

이 대화 상자는 AWS Lambda 함수의 로컬 버전에 대한 설정을 업데이트할 때마다 표시됩니다.

> **Note**
>
> 동일한 함수의 원격 버전에 대한 설정을 업데이트하려면(함수의 소스 코드는 AWS 계정의 Lambda에 있음) 실행/디버깅 구성 대화 상자(원격 함수 설정) 섹션을 대신 참조하세요.

이 대화 상자에는 구성, SAM CLI 및 AWS 연결이라는 세 가지 탭이 있습니다.



로컬 함수 설정에 대한 실행/디버깅 구성 대화 상자의 구성 탭에는 다음 항목이 포함되어 있습니다.

이름

(필수) 이 구성의 이름입니다.

병렬 실행 허용/병렬로 실행 허용

(선택 사항) 이 항목을 선택하면 IntelliJ IDEA, PyCharm, WebStorm 또는 JetBrains Rider가 병렬로 실행할 구성 인스턴스를 필요한 만큼 실행할 수 있습니다.[1]

핸들러에서/템플릿에서

(필수) 선택하는 옵션에 따라 추가 설정을 구성해야 합니다.

런타임

(필수) 사용할 Lambda 런타임의 ID입니다.

핸들러

(핸들러에서 옵션의 경우 필수) Java, Python, Node.js 또는 C#에 대한 해당 함수 핸들러의 식별자입니다.

시간 초과(초)

(핸들러에서 옵션의 경우 필수) Lambda가 함수를 중지하기 전에 실행을 허용하는 시간입니다. 최대 900초(15분)까지 지정합니다.

메모리(MB)

(핸들러에서 옵션의 경우 필수) 함수가 실행될 때 사용할 수 있는 메모리 양입니다. 128MB에서 3,008MB 사이의 크기를 64MB 단위로 지정합니다.

환경 변수

(핸들러에서 옵션의 경우 선택 사항) Lambda 함수에서 사용할 환경 변수로, 키-값 페어로 지정됩니다. 환경 변수를 추가, 변경 또는 삭제하려면 폴더 아이콘을 선택한 다음 화면의 지침을 따릅니다.

Template

(템플릿에서 옵션의 경우 필수)이 구성에 사용할 AWS Serverless Application Model(AWS SAM) 템플릿의 위치 및 파일 이름(예: `template.yaml`) 및 이 구성과 연결할 해당 템플릿의 리소스입니다.

파일

(필수) 함수에 전달할 이벤트 데이터의 위치와 파일 이름(JSON 형식)입니다. 이벤트 데이터 예제는 AWS Lambda 개발자 안내서의 간접적으로 Lambda 함수 호출과 AWS Serverless Application Model 개발자 안내서의 Generating sample event payloads를 참조하세요.

Text

(필수) 함수에 전달할 이벤트 데이터(JSON 형식)입니다. 이벤트 데이터 예제는 AWS Lambda 개발자 안내서의 간접적으로 Lambda 함수 호출과 AWS Serverless Application Model 개발자 안내서의 Generating sample event payloads를 참조하세요.

> **Note**
>
> 파일 또는 텍스트가 필요합니다(둘 다는 아님).

시작 전: 창

(선택 사항) 이 구성을 시작하기 전에 수행해야 하는 태스크를 나열합니다.[2]

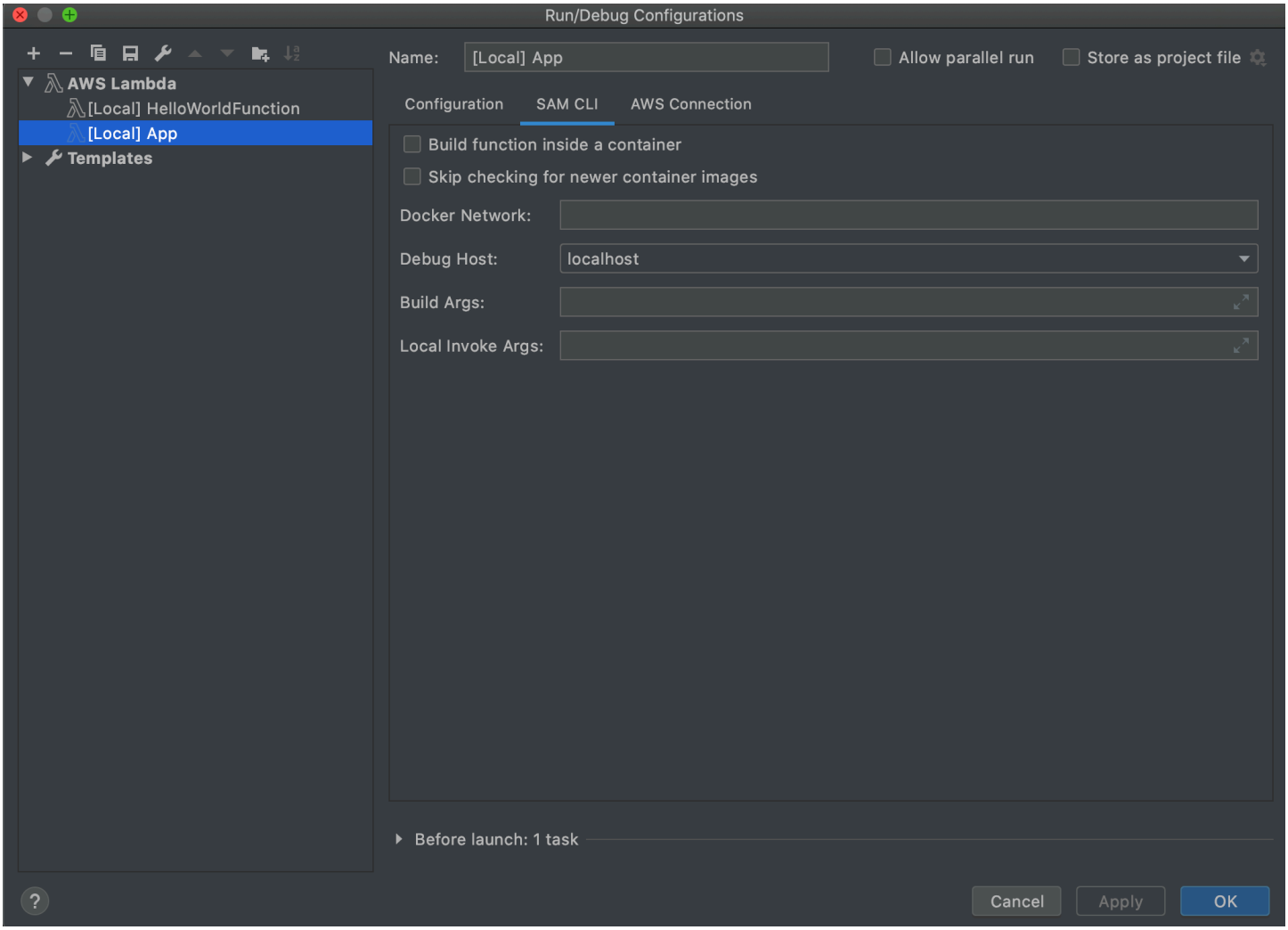
참고

[1] 자세한 내용은 다음을 참조하십시오.

- IntelliJ IDEA의 경우 IntelliJ IDEA 도움말 웹 사이트의 Common settings를 참조하세요.
- PyCharm의 경우 PyCharm 도움말 웹 사이트의Common settings를 참조하세요.
- WebStorm의 경우 WebStorm 도움말 웹 사이트의 Common settings를 참조하세요.
- JetBrains Rider의 경우 JetBrains Rider 도움말 웹 사이트의 Common options를 참조하세요.

[2] 자세한 내용은 다음을 참조하십시오.

- IntelliJ IDEA의 경우 IntelliJ IDEA 도움말 웹 사이트의 Before launch를 참조하세요.
- PyCharm의 경우 PyCharm 도움말 웹 사이트의Before launch를 참조하세요.
- WebStorm의 경우 WebStorm 도움말 웹 사이트의 Before launch를 참조하세요.
- JetBrains Rider의 경우 JetBrains Rider 도움말 웹 사이트의 Before launch를 참조하세요.



로컬 함수 설정에 대한 실행/디버깅 구성 대화 상자의 SAM CLI 탭에는 다음 항목이 포함되어 있습니다.

이름

(필수) 이 구성의 이름입니다.

병렬 실행 허용/병렬로 실행 허용

(선택 사항) 이 항목을 선택하면 IntelliJ IDEA, PyCharm, WebStorm 또는 JetBrains Rider가 병렬로 실행할 구성 인스턴스를 필요한 만큼 실행할 수 있습니다.[1]

컨테이너 내부에 함수를 빌드합니다.

(선택 사항) 이 옵션을 선택하면 AWS SAM CLI는 배포 전에 Lambda와 같은 Docker 컨테이너 내부에 로컬로 서버리스 애플리케이션의 기능을 빌드합니다. 이 기능은 함수가 기본적으로 컴파일된 종

속성 또는 프로그램을 가진 패키지에 의존하는 경우에 유용합니다. 자세한 내용은 AWS Serverless Application Model 개발자 안내서의 애플리케이션 빌드를 참조하세요.

최신 컨테이너 이미지 확인 건너 뛰기

(선택 사항) 이 항목을 선택하면 AWS SAM CLI가 구성 탭에 지정된 런타임에 대한 최신 Docker 이미지 풀다운을 건너뜁니다.

Docker 네트워크

(선택 사항) Docker 컨테이너가 연결할 기존 Docker 네트워크의 이름 또는 ID와 기본 브리지 네트워크입니다. 지정하지 않으면 Lambda 컨테이너는 기본 브리지 Docker 네트워크에만 연결됩니다.

시작 전: 창

(선택 사항) 이 구성을 시작하기 전에 수행해야 하는 태스크를 나열합니다.[2]

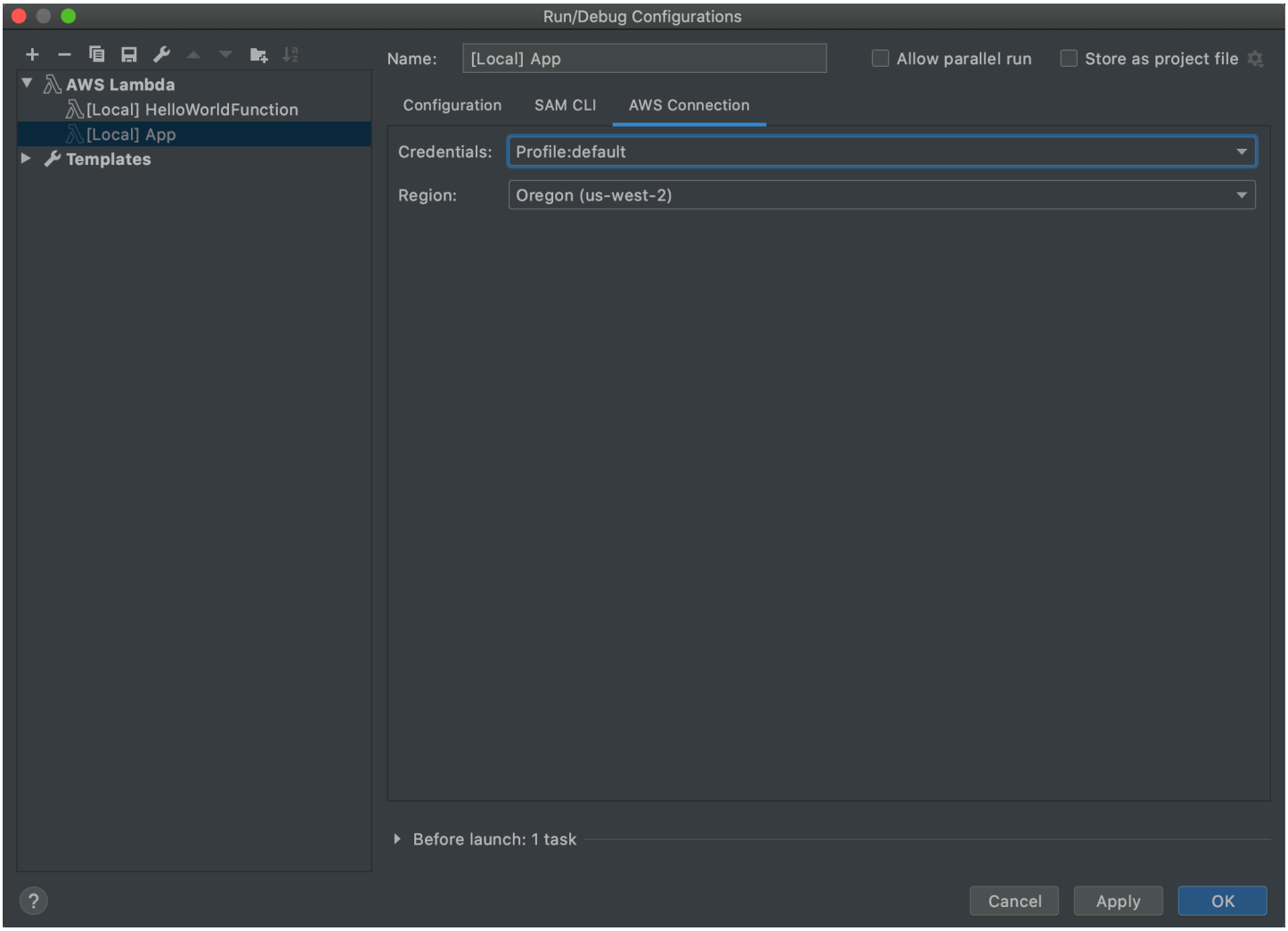
참고

[1] 자세한 내용은 다음을 참조하십시오.

- IntelliJ IDEA의 경우 IntelliJ IDEA 도움말 웹 사이트의 Common settings를 참조하세요.
- PyCharm의 경우 PyCharm 도움말 웹 사이트의Common settings를 참조하세요.
- WebStorm의 경우 WebStorm 도움말 웹 사이트의 Common settings를 참조하세요.
- JetBrains Rider의 경우 JetBrains Rider 도움말 웹 사이트의 Common options를 참조하세요.

[2] 자세한 내용은 다음을 참조하십시오.

- IntelliJ IDEA의 경우 IntelliJ IDEA 도움말 웹 사이트의 Before launch를 참조하세요.
- PyCharm의 경우 PyCharm 도움말 웹 사이트의Before launch를 참조하세요.
- WebStorm의 경우 WebStorm 도움말 웹 사이트의 Before launch를 참조하세요.
- JetBrains Rider의 경우 JetBrains Rider 도움말 웹 사이트의 Before launch를 참조하세요.



로컬 함수 설정에 대한 실행/디버깅 구성 대화 상자의 AWS 연결 탭에는 다음 항목이 포함되어 있습니다.

보안 인증

(필수) 사용할 기존 AWS 계정 연결의 이름입니다.

리전(Region)

(필수) 연결된 계정에 사용할 AWS 리전의 이름입니다.

참고

[1] 자세한 내용은 다음을 참조하십시오.

- IntelliJ IDEA의 경우 IntelliJ IDEA 도움말 웹 사이트의 Common settings를 참조하세요.
- PyCharm의 경우 PyCharm 도움말 웹 사이트의Common settings를 참조하세요.

- WebStorm의 경우 WebStorm 도움말 웹 사이트의 Common settings를 참조하세요.
- JetBrains Rider의 경우 JetBrains Rider 도움말 웹 사이트의 Common options를 참조하세요.

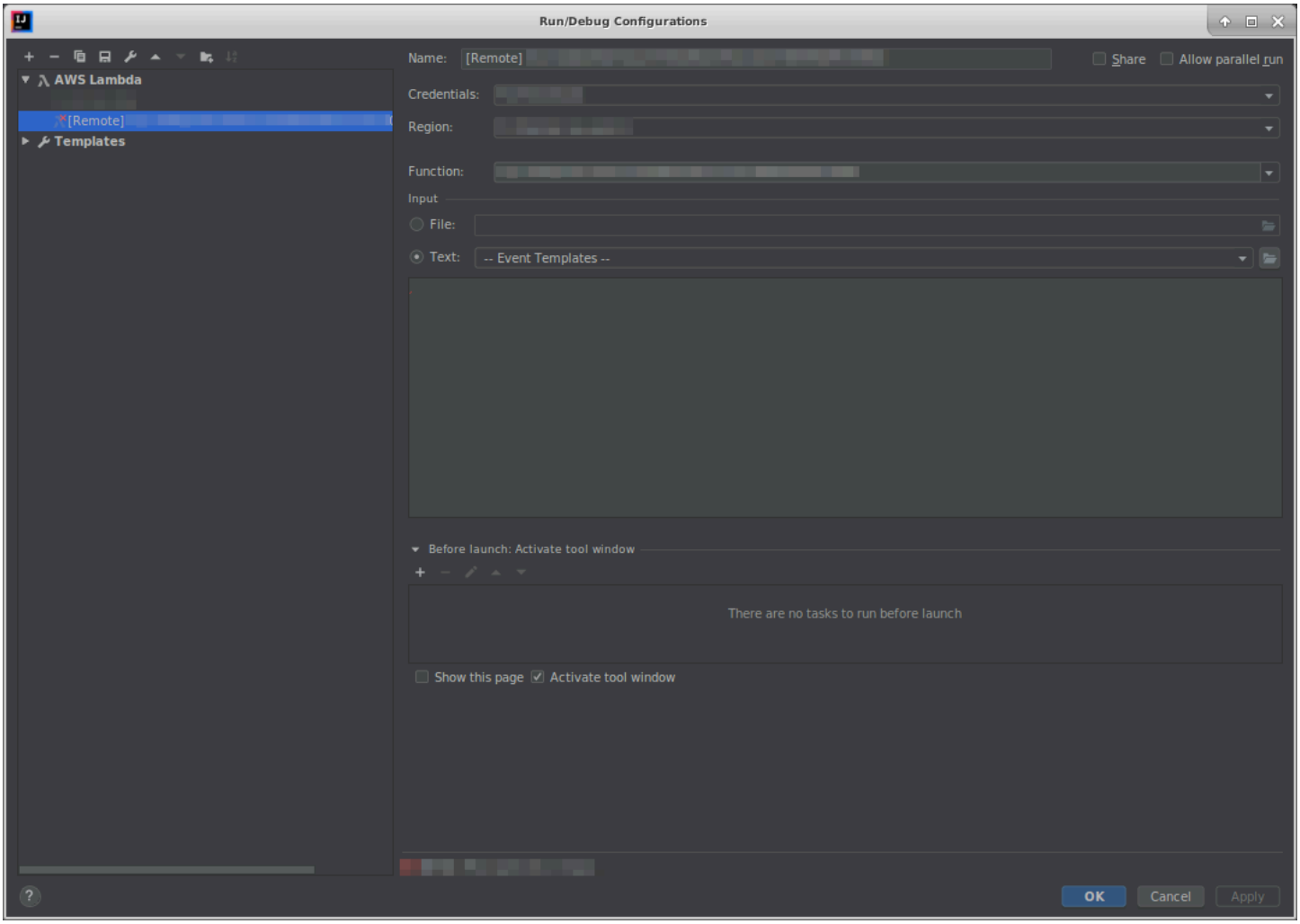
[2] 자세한 내용은 다음을 참조하십시오.

- IntelliJ IDEA의 경우 IntelliJ IDEA 도움말 웹 사이트의 Before launch를 참조하세요.
- PyCharm의 경우 PyCharm 도움말 웹 사이트의Before launch를 참조하세요.
- WebStorm의 경우 WebStorm 도움말 웹 사이트의 Before launch를 참조하세요.
- JetBrains Rider의 경우 JetBrains Rider 도움말 웹 사이트의 Before launch를 참조하세요.

실행/디버깅 구성 대화 상자(원격 함수 설정)

이 대화 상자는 AWS Lambda 함수의 원격 버전에 대한 설정을 업데이트할 때마다 표시됩니다(함수의 소스 코드는 AWS 계정의 Lambda에 있음).

> **Note**
>
> 동일한 함수의 로컬 버전에 대한 설정을 업데이트하려면 대신 실행/디버깅 구성 대화 상자(로컬 함수 설정) 섹션을 참조하세요.
> 대화 상자의 이름은 실행/디버깅 구성이지만 AWS Toolkit for JetBrains를 사용하여 Lambda 함수의 원격 버전을 디버깅할 수 없습니다. 함수 실행만 가능합니다.



원격 함수 설정에 대한 실행/디버깅 구성 대화 상자에는 다음 항목이 포함되어 있습니다.

이름

(필수) 이 구성의 이름입니다.

공유/VCS를 통해 공유

(선택 사항) 이 옵션을 선택하면 다른 팀원이 이 구성을 사용할 수 있습니다.[1]

병렬 실행 허용/병렬로 실행 허용

(선택 사항) 이 항목을 선택하면 IntelliJ IDEA, PyCharm, WebStorm 또는 JetBrains Rider가 병렬로 실행할 구성 인스턴스를 필요한 만큼 실행할 수 있습니다.[1]

보안 인증

(필수) 사용할 기존 AWS 계정 연결의 이름입니다.

리전(Region)

(필수) 연결된 계정에 사용할 AWS 리전의 이름입니다.

함수

(필수) 사용할 Lambda 함수의 이름입니다.

파일

(필수) 함수에 전달할 이벤트 데이터의 위치와 파일 이름(JSON 형식)입니다. 이벤트 데이터 예제는 AWS Lambda 개발자 안내서의 간접적으로 Lambda 함수 호출과 AWS Serverless Application Model 개발자 안내서의 Generating sample event payloads를 참조하세요.

Text

(필수) 함수에 전달할 이벤트 데이터(JSON 형식)입니다. 이벤트 데이터 예제는 AWS Lambda 개발자 안내서의 간접적으로 Lambda 함수 호출과 AWS Serverless Application Model 개발자 안내서의 Generating sample event payloads를 참조하세요.

> **Note**
>
> 파일 또는 텍스트가 필요합니다(둘 다는 아님).

시작 전: 도구 창 활성화

(선택 사항) 이 구성을 시작하기 전에 수행해야 하는 태스크를 나열합니다.[2]

이 페이지 표시

(선택 사항) 이 옵션을 선택하면 이 구성을 시작하기 전에 이러한 구성 설정이 표시됩니다.[2]

도구 창 활성화

(선택 사항) 이 옵션을 선택하면 이 구성을 시작할 때 실행 또는 디버깅 도구 창이 열립니다.[2]

참고

[1] 자세한 내용은 다음을 참조하십시오.

- IntelliJ IDEA의 경우 IntelliJ IDEA 도움말 웹 사이트의 Common settings를 참조하세요.
- PyCharm의 경우 PyCharm 도움말 웹 사이트의Common settings를 참조하세요.

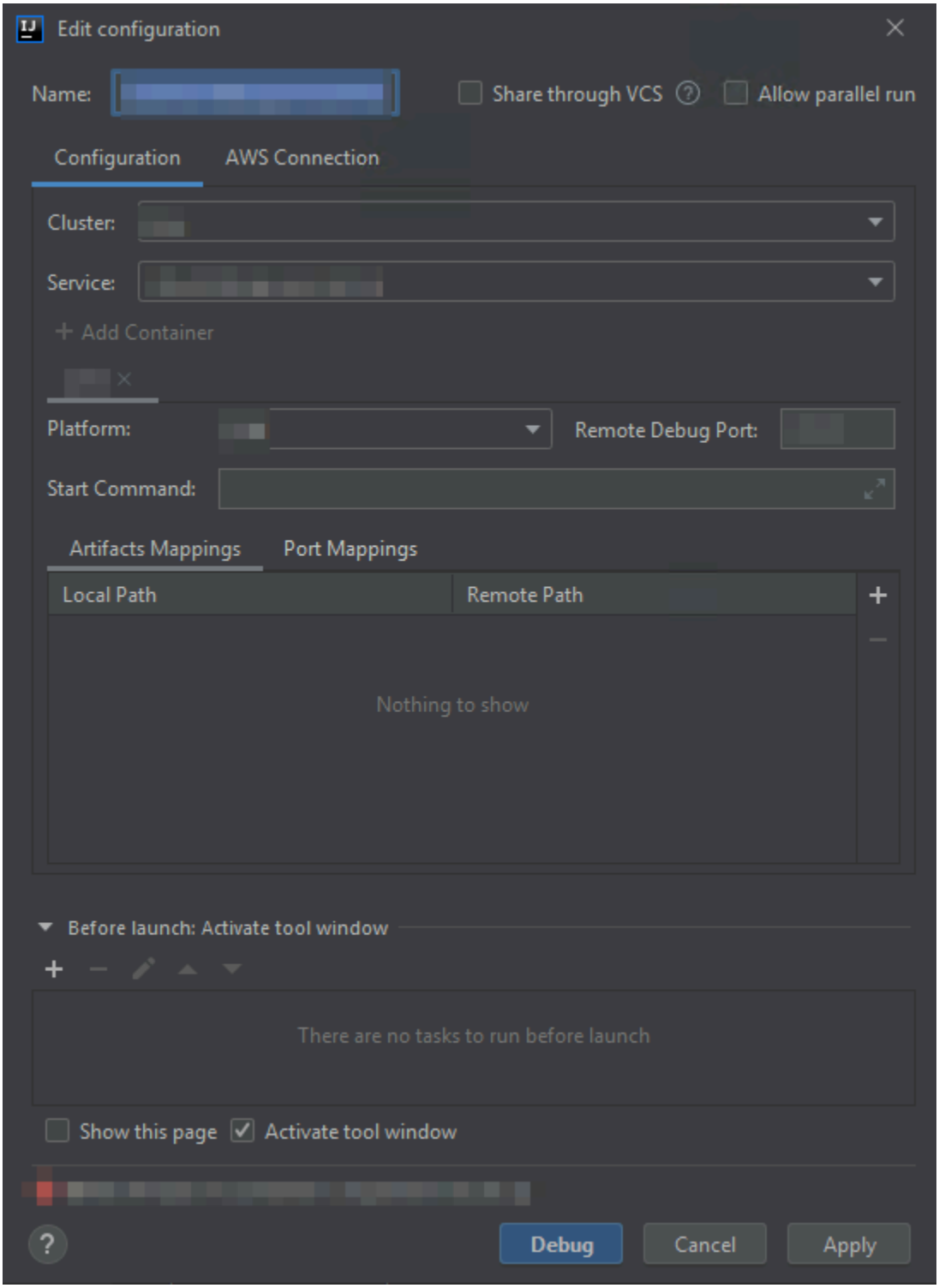
- WebStorm의 경우 WebStorm 도움말 웹 사이트의 Common settings를 참조하세요.
- JetBrains Rider의 경우 JetBrains Rider 도움말 웹 사이트의 Common options를 참조하세요.

[2] 자세한 내용은 다음을 참조하십시오.

- IntelliJ IDEA의 경우 IntelliJ IDEA 도움말 웹 사이트의 Before launch를 참조하세요.
- PyCharm의 경우 PyCharm 도움말 웹 사이트의Before launch를 참조하세요.
- WebStorm의 경우 WebStorm 도움말 웹 사이트의 Before launch를 참조하세요.
- JetBrains Rider의 경우 JetBrains Rider 도움말 웹 사이트의 Before launch를 참조하세요.

구성 편집 대화 상자(Amazon ECS 클러스터)

구성 편집 대화 상자에는 구성 및 AWS연결 탭이 있습니다.



구성 편집 대화 상자의 구성 탭에는 다음 항목이 포함되어 있습니다.

이름

(필수) 이 구성의 이름입니다.

공유/VCS를 통해 공유

(선택 사항) 이 옵션을 선택하면 다른 팀원이 이 구성을 사용할 수 있습니다.[1]

병렬 실행 허용/병렬로 실행 허용

(선택 사항) 이 항목을 선택하면 IntelliJ IDEA, PyCharm, WebStorm 또는 JetBrains Rider가 병렬로 실행할 구성 인스턴스를 필요한 만큼 실행할 수 있습니다.[1]

클러스터

(필수) 디버깅할 클러스터의 Amazon Elastic Container Service(Amazon ECS) 이름입니다.

서비스

(필수) 디버깅할 클러스터의 Amazon ECS 서비스 이름입니다.

컨테이너 추가

이 구성에 컨테이너를 추가합니다. 하나 이상의 탭이 이미 표시되어 있는 경우 선택 사항입니다. 각 탭은 별도의 컨테이너를 나타냅니다.

선택한 컨테이너에는 플랫폼, 원격 디버그 포트, 시작 명령, 아티팩트 매핑 및 포트 매핑 항목이 적용됩니다.

플랫폼

(필수) 사용할 디버그 플랫폼입니다.

원격 디버그 포트

(선택 사항) 디버거에 연결할 포트입니다. 일반적으로 서비스에서 포트 20,020~20,030을 사용하지 않는 한 이 값을 지정하면 안 됩니다. 만약 서비스에서 포트 20,020~20,030을 사용하는 경우 여기에 포트를 지정하여 컨테이너가 다른 곳에서 사용 중인 포트를 바인딩하지 않도록 하세요.

시작 명령

(필수) 디버거가 연결할 수 있도록 프로그램을 시작하는 명령입니다. Java의 경우 `java`로 시작하고 디버거 정보(예: `-Xdebug`)를 포함하지 않아야 합니다. Python의 경우 `python`, `python2`, `python3` 중 하나로 시작하고 실행할 파일의 경로와 이름을 이어서 사용합니다.

아티팩트 매핑

(필수) 컨테이너 내의 원격 경로에 매핑되는 로컬 개발 머신의 로컬 경로입니다. 실행할 계획인 모든 코드와 아티팩트를 매핑해야 합니다. 로컬 및 원격 경로 매핑을 지정하려면 추가(+ 아이콘)를 선택합니다.

포트 매핑

(선택 사항) 컨테이너 내의 원격 포트에 매핑되는 로컬 개발 머신의 로컬 포트입니다. 이렇게 하면 로컬 포트가 원격 리소스의 포트와 직접 통신할 수 있습니다. 예를 들어 `curl localhost:3422` 명령의 경우 3422 포트는 일부 서비스에 매핑됩니다. 로컬 및 원격 포트 매핑을 지정하려면 추가(+ 아이콘)를 선택합니다.

시작 전: 도구 창 활성화

(선택 사항) 이 구성을 시작하기 전에 수행해야 하는 태스크를 나열합니다.[2]

이 페이지 표시

(선택 사항) 이 옵션을 선택하면 이 구성을 시작하기 전에 이러한 구성 설정이 표시됩니다.[2]

도구 창 활성화

(선택 사항) 이 옵션을 선택하면 이 구성을 시작할 때 실행 또는 디버깅 도구 창이 열립니다.[2]

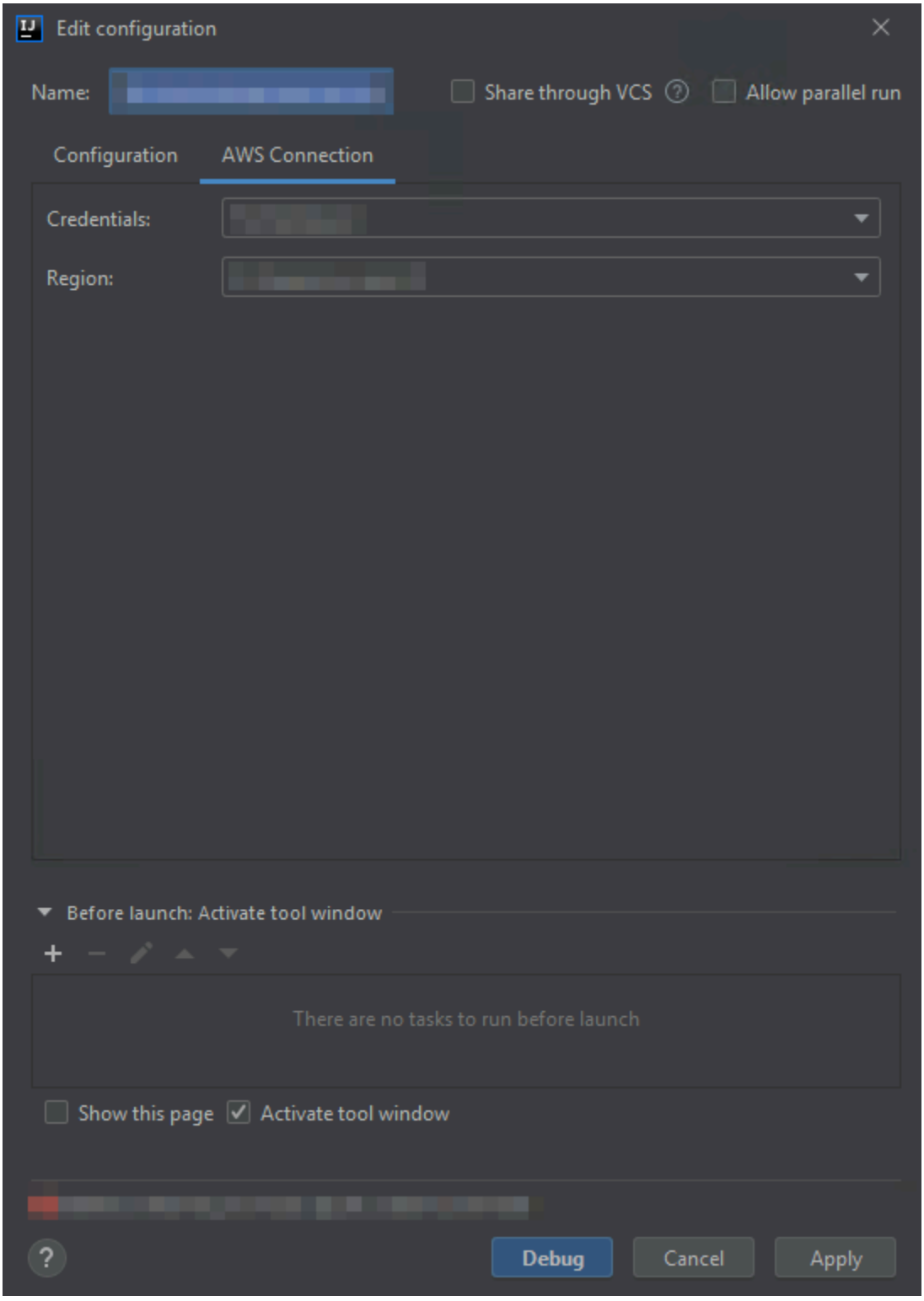
참고

[1] 자세한 내용은 다음을 참조하십시오.

- IntelliJ IDEA의 경우 IntelliJ IDEA 도움말 웹 사이트의 Common settings를 참조하세요.
- PyCharm의 경우 PyCharm 도움말 웹 사이트의Common settings를 참조하세요.
- WebStorm의 경우 WebStorm 도움말 웹 사이트의 Common settings를 참조하세요.
- JetBrains Rider의 경우 JetBrains Rider 도움말 웹 사이트의 Common options를 참조하세요.

[2] 자세한 내용은 다음을 참조하십시오.

- IntelliJ IDEA의 경우 IntelliJ IDEA 도움말 웹 사이트의 Before launch를 참조하세요.
- PyCharm의 경우 PyCharm 도움말 웹 사이트의Before launch를 참조하세요.
- WebStorm의 경우 WebStorm 도움말 웹 사이트의 Before launch를 참조하세요.
- JetBrains Rider의 경우 JetBrains Rider 도움말 웹 사이트의 Before launch를 참조하세요.



구성 편집 대화 상자의 AWS 연결 탭에는 다음 항목이 포함되어 있습니다.

이름

(필수) 이 구성의 이름입니다.

보안 인증

(필수) 사용할 기존 AWS 계정 연결의 이름입니다.

리전(Region)

(필수) 연결된 계정에 사용할 AWS 리전의 이름입니다.

공유/VCS를 통해 공유

(선택 사항) 이 옵션을 선택하면 다른 팀원이 이 구성을 사용할 수 있습니다.[1]

병렬 실행 허용/병렬로 실행 허용

(선택 사항) 이 항목을 선택하면 IntelliJ IDEA, PyCharm, WebStorm 또는 JetBrains Rider가 병렬로 실행할 구성 인스턴스를 필요한 만큼 실행할 수 있습니다.[1]

시작 전: 도구 창 활성화

(선택 사항) 이 구성을 시작하기 전에 수행해야 하는 태스크를 나열합니다.[2]

이 페이지 표시

(선택 사항) 이 옵션을 선택하면 이 구성을 시작하기 전에 이러한 구성 설정이 표시됩니다.[2]

도구 창 활성화

(선택 사항) 이 옵션을 선택하면 이 구성을 시작할 때 실행 또는 디버깅 도구 창이 열립니다.[2]

참고

[1] 자세한 내용은 다음을 참조하십시오.

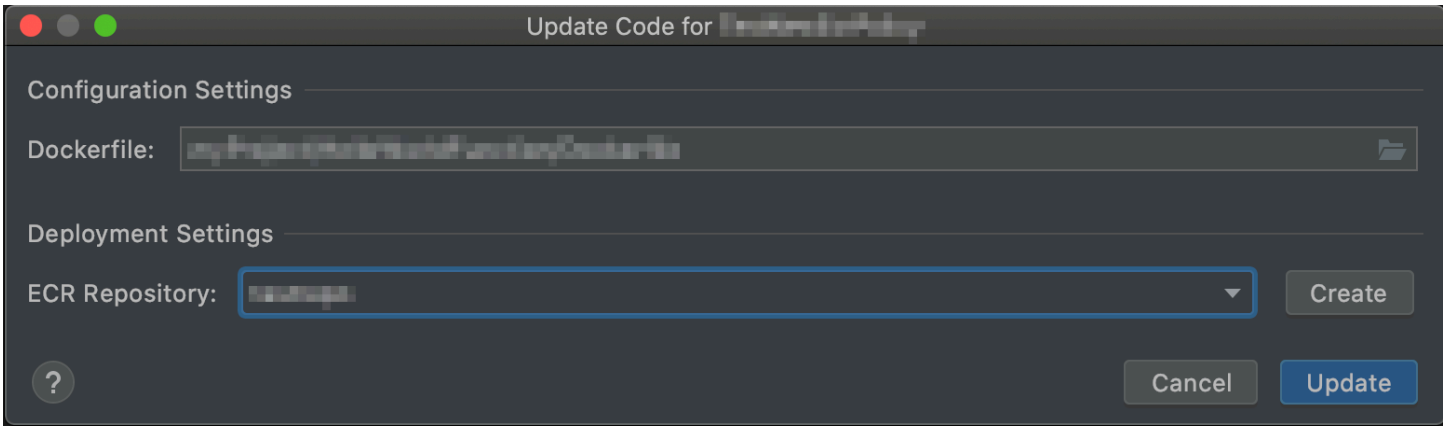
- IntelliJ IDEA의 경우 IntelliJ IDEA 도움말 웹 사이트의 Common settings를 참조하세요.
- PyCharm의 경우 PyCharm 도움말 웹 사이트의Common settings를 참조하세요.
- WebStorm의 경우 WebStorm 도움말 웹 사이트의 Common settings를 참조하세요.
- JetBrains Rider의 경우 JetBrains Rider 도움말 웹 사이트의 Common options를 참조하세요.

[2] 자세한 내용은 다음을 참조하십시오.

- IntelliJ IDEA의 경우 IntelliJ IDEA 도움말 웹 사이트의 Before launch를 참조하세요.
- PyCharm의 경우 PyCharm 도움말 웹 사이트의Before launch를 참조하세요.
- WebStorm의 경우 WebStorm 도움말 웹 사이트의 Before launch를 참조하세요.
- JetBrains Rider의 경우 JetBrains Rider 도움말 웹 사이트의 Before launch를 참조하세요.

코드 업데이트 대화 상자

AWS Lambda 함수를 업데이트할 때마다 AWS Toolkit for JetBrains의 코드 업데이트 대화 상자가 표시됩니다.



코드 업데이트 대화 상자에는 다음 항목이 포함되어 있습니다.

핸들러

(필수) Java, Python, Node.js, C#에 대한 해당 Lambda 함수 핸들러의 ID입니다.

원본 버킷

(`Zip` 패키지 유형의 경우에만 필수) AWS Serverless Application Model(AWS SAM) 명령줄 인터페이스(CLI)에서 Lambda에 함수를 배포하는 데 사용할 연결된 AWS 계정의 기존 Amazon Simple Storage Service(S3) 버킷을 선택합니다. 계정에 Amazon S3 버킷을 생성하고 AWS SAM CLI에서 해당 버킷을 대신 사용하도록 하려면 생성을 선택한 다음 화면의 지침을 따릅니다. Lambda 패키지 유형에 대한 자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 배포 패키지를 참조하세요.

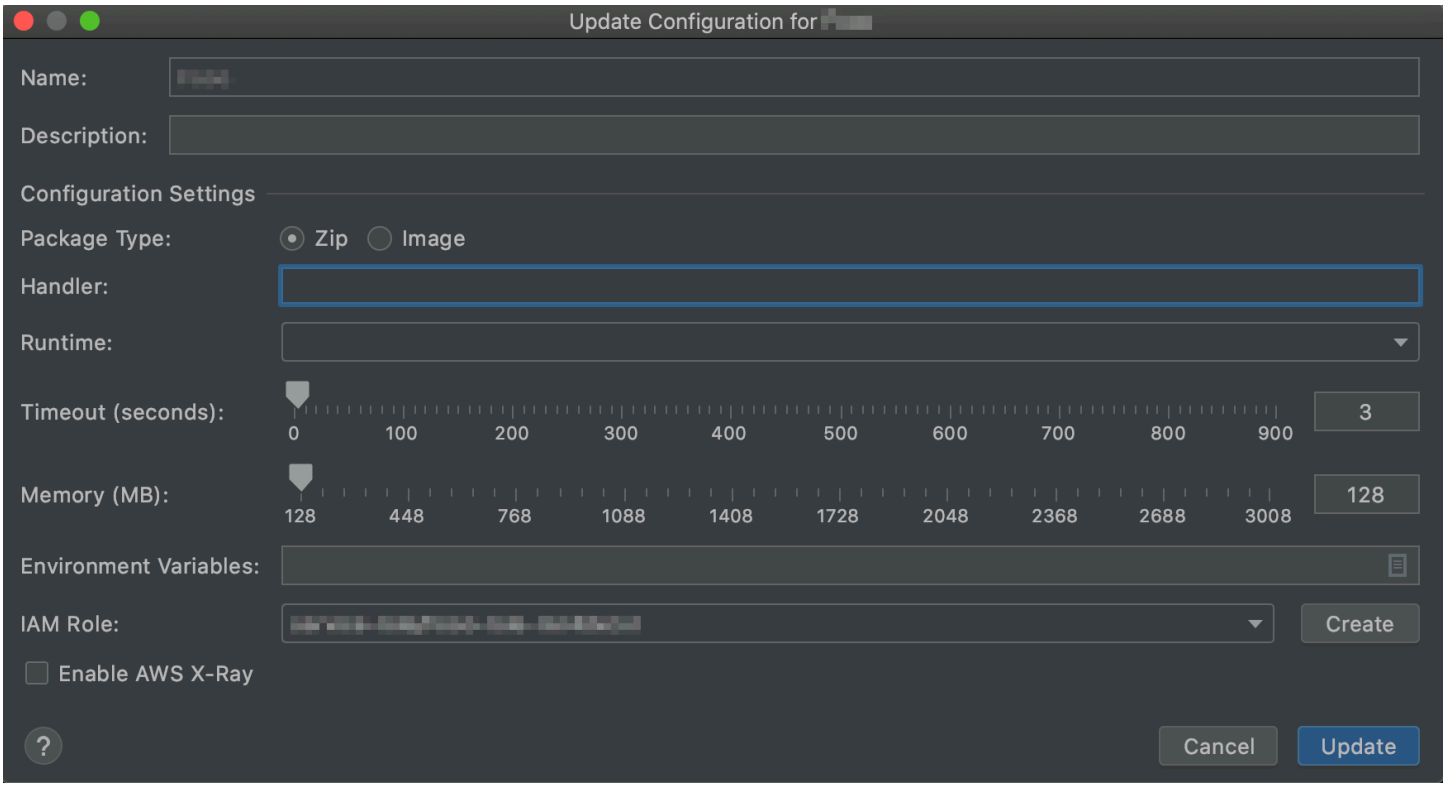
ECR 리포지토리

(`Image` 패키지 유형의 경우에만 필수) AWS SAM CLI에서 Lambda에 함수를 배포하는 데 사용할 연결된 AWS 계정의 기존 Amazon Elastic Container Registry(Amazon ECR) 리포지토리를 선택합니다.

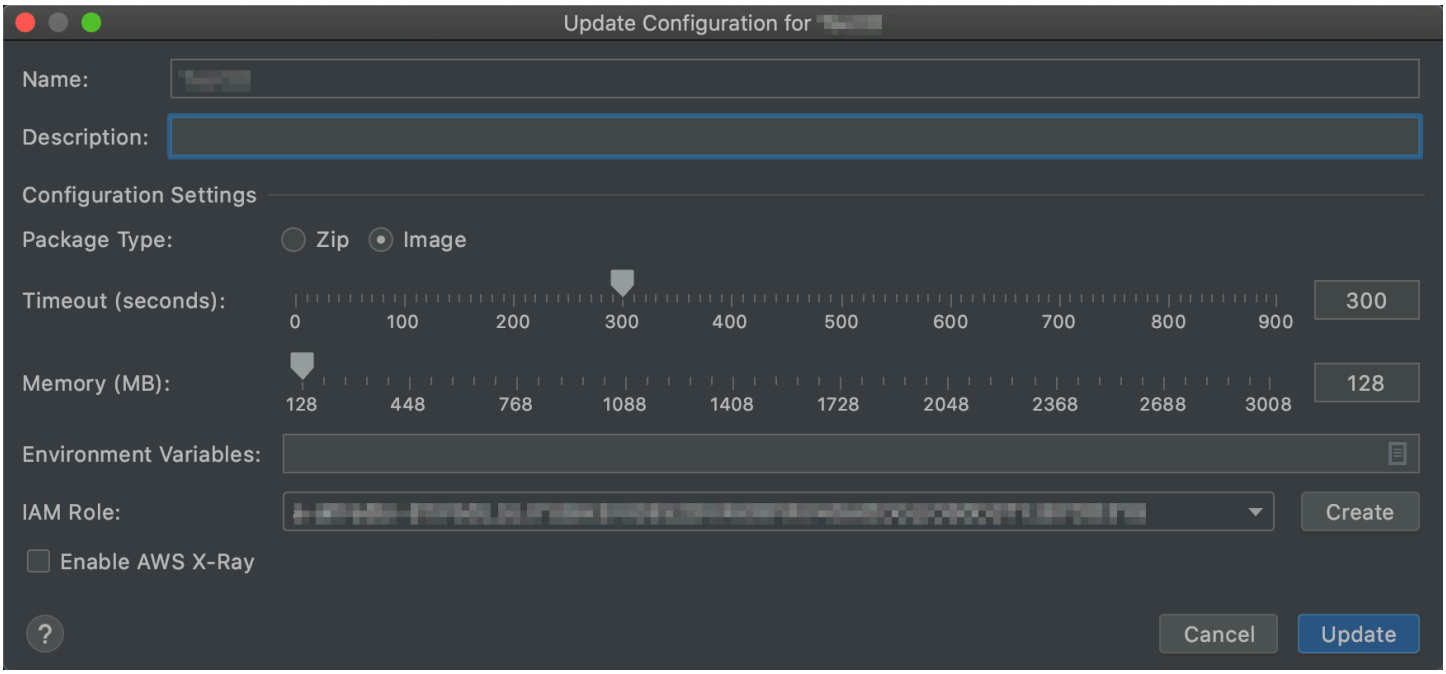
구성 업데이트 대화 상자

AWS Lambda 함수의 구성을 업데이트할 때마다 AWS Toolkit for JetBrains의 구성 업데이트 대화 상자가 표시됩니다. 제공하는 정보는 프로젝트의 Lambda 함수가 패키지 유형 `Zip`인지 `Image`인지에 따라 약간 다릅니다.

`Zip` 패키지 유형에 대한 구성 업데이트 대화 상자:



`Image` 패키지 유형에 대한 구성 업데이트 대화 상자:



구성 업데이트 대화 상자에는 다음 항목이 포함되어 있습니다.

이름

(필수) 함수의 이름입니다. 대문자(A~Z), 소문자(a~z), 숫자(0~9), 하이픈(-) 및 밑줄(_)만 포함할 수 있습니다. 이름은 64자 미만이어야 합니다.

설명

(선택 사항) 함수에 대한 의미 있는 설명입니다.

패키지 유형

(필수) Lambda 함수의 패키지 유형(`Zip` 또는 `Image`)입니다.

핸들러

(`Zip` 패키지의 경우에만 필수) Java, Python, Node.js 또는 C#에 대한 해당 Lambda 함수 핸들러의 ID입니다.

런타임

(`Zip` 패키지의 경우에만 필수) 사용할 Lambda 런타임의 ID입니다.

시간 초과(초)

(필수) Lambda가 함수를 중지하기 전에 실행을 허용하는 시간입니다. 최대 900초(15분)까지 지정합니다.

메모리(MB)

(필수) 함수가 실행될 때 사용할 수 있는 메모리 양입니다. 128MB에서 3,008MB 사이의 크기를 64MB 단위로 지정합니다.

환경 변수

(선택 사항) Lambda 함수에서 사용할 환경 변수로, 키-값 페어로 지정됩니다. 환경 변수를 추가, 변경 또는 삭제하려면 폴더 아이콘을 선택한 다음 화면의 지침을 따릅니다.

IAM 역할

(필수) Lambda에서 함수에 사용할 연결된 AWS 계정의 사용 가능한 Lambda 실행 역할을 선택합니다. 계정에서 실행 역할을 생성하고 Lambda가 해당 역할을 대신 사용하도록 하려면 생성을 선택한 다음 화면의 지침을 따릅니다.

AWS X-Ray 활성화

(선택 사항) 이 항목을 선택하면 AWS X-Ray가 함수의 성능 문제를 감지, 분석 및 최적화할 수 있습니다. X-Ray는 Lambda와 함수를 구성하는 모든 업스트림 또는 다운스트림 서비스에서 메타데이터

를 수집합니다. X-Ray는 이러한 메타데이터를 이용하여 성능 병목 현상, 지연 시간 스파이크 및 함수 성능에 영향을 미치는 기타 문제를 보여주는 상세한 서비스 그래픽을 생성합니다.

이 AWS 제품 또는 서비스에 대한 보안

Amazon Web Services(AWS)에서 가장 우선순위가 높은 것이 클라우드 보안입니다. AWS 고객은 보안에 매우 민감한 조직의 요구 사항에 부합하도록 구축된 데이터 센터 및 네트워크 아키텍처의 혜택을 누릴 수 있습니다. 보안은 AWS와 귀하의 공동 책임입니다. 공동 책임 모델은 이 사항을 클라우드 내 보안 및 클라우드의 보안으로 설명합니다.

클라우드의 보안 – AWS는 AWS 클라우드에서 모든 서비스를 실행하는 인프라를 보호하며 안전하게 사용할 수 있는 서비스를 제공합니다. 당사의 보안 책임은 AWS에서 우선 순위가 가장 높으며, 서드 파티 감사자는 AWS 규정 준수 프로그램의 일환으로 정기적으로 보안 효율성을 테스트하고 검증합니다.

클라우드 내 보안 – 사용자의 책임은 사용하는 AWS 서비스, 그리고 데이터의 민감도, 조직의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 의해 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 준수 프로그램의 AWS 규정 준수 업무 범위에 속하는 AWS 서비스를 참조하세요.

주제

- AWS Toolkit for JetBrains의 데이터 보호
- Identity and Access Management
- 이 AWS 제품 또는 서비스에 대한 규정 준수 확인
- 이 AWS 제품 또는 서비스에 대한 복원력
- 이 AWS 제품 또는 서비스에 대한 인프라 보안

AWS Toolkit for JetBrains의 데이터 보호

AWS Shared Responsibility Model은 AWS Toolkit for JetBrains의 데이터 보호에 적용됩니다. 이 모델에서 설명하는 것처럼 AWS는 모든 AWS 클라우드를 실행하는 글로벌 인프라를 보호할 책임이 있습니다. 이 인프라에서 호스팅되는 콘텐츠에 대한 제어를 유지하는 것은 사용자의 책임입니다. 이 콘텐츠에는 사용하는 AWS 서비스 서비스의 보안 구성과 관리 작업이 포함돼 있습니다. 데이터 프라이버시에 대한 자세한 내용은 데이터 프라이버시 FAQ를 참조하세요. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 AWS 공동 책임 모델 및 GDPR 블로그 게시물을 참조하세요.

데이터를 보호하려면 AWS 계정 보안 인증 정보를 보호하고 AWS IAM Identity Center 또는 AWS Identity and Access Management(IAM)를 통해 개별 사용자 계정을 설정하는 것이 좋습니다. 이러한 방

식에는 각 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정마다 멀티 팩터 인증(MFA)을 사용합니다.
- SSL/TLS를 사용하여 AWS 리소스와 통신합니다. TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- AWS CloudTrail로 API 및 사용자 활동 로깅을 설정합니다.
- AWS 암호화 솔루션을 AWS 서비스 내의 모든 기본 보안 컨트롤과 함께 사용합니다.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용합니다.
- 명령줄 인터페이스 또는 API를 통해 AWS에 액세스할 때 FIPS 140-2 검증된 암호화 모듈이 필요한 경우 FIPS 엔드포인트를 사용합니다. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 Federal Information Processing Standard(FIPS) 140-2를 참조하세요.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 양식 필드에 입력하지 않는 것이 좋습니다. 여기에는 콘솔, API, AWS CLI 또는 AWS SDK를 사용하여 AWS Toolkit for JetBrains 또는 기타 AWS 서비스 작업을 수행하는 경우가 포함됩니다. 이름에 사용되는 태그 또는 자유 형식 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 자격 증명 정보를 URL에 포함시켜서는 안됩니다.

Identity and Access Management

AWS Identity and Access Management(IAM)는 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어할 수 있도록 지원하는 AWS 서비스입니다. IAM 관리자는 어떤 사용자가 AWS 리소스를 사용할 수 있는 인증(로그인) 및 권한(권한 있음)을 받을 수 있는지 제어합니다. IAM은 추가 비용 없이 사용할 수 있는 AWS 서비스입니다.

주제

- 대상
- 자격 증명을 통한 인증
- 정책을 사용하여 액세스 관리
- AWS 서비스에서 IAM을 사용하는 방식
- AWS 자격 증명 및 액세스 문제 해결

대상

AWS Identity and Access Management(IAM)를 사용하는 방법은 AWS에서 수행하는 작업에 따라 달라집니다.

서비스 사용자 – AWS 서비스를 사용하여 작업을 수행하는 경우 필요한 보안 인증 정보와 권한을 관리자가 제공합니다. 더 많은 AWS 기능을 사용하여 작업을 수행하게 되면 추가 권한이 필요할 수 있습니다. 액세스 권한 관리 방식을 이해하면 적절한 권한을 관리자에게 요청할 수 있습니다. AWS의 기능에 액세스할 수 없는 경우 AWS 자격 증명 및 액세스 문제 해결 또는 사용 중인 AWS 서비스의 사용 설명서를 참조하세요.

서비스 관리자 - 회사에서 AWS 리소스를 책임지고 있는 경우 AWS에 대한 전체 액세스 권한을 가지고 있을 것입니다. 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. 회사에서 AWS와 함께 IAM을 사용하는 방법에 대한 자세한 내용은 사용 중인 AWS 서비스의 사용 설명서를 참조하세요.

IAM 관리자 - IAM 관리자라면 AWS에 대한 액세스 권한 관리 정책 작성 방법을 자세히 알고 싶을 것입니다. IAM에서 사용할 수 있는 AWS 자격 증명 기반 정책 예제를 보려면 사용 중인 AWS 서비스의 사용 설명서를 참조하세요.

자격 증명을 통한 인증

인증은 ID 자격 증명을 사용하여 AWS에 로그인하는 방식입니다. AWS 계정 루트 사용자나 IAM 사용자로 또는 IAM 역할을 수임하여 인증(AWS에 로그인)되어야 합니다.

자격 증명 소스를 통해 제공된 보안 인증 정보를 사용하여 페더레이션형 ID로 AWS에 로그인할 수 있습니다. AWS IAM Identity Center (IAM Identity Center) 사용자, 회사의 Single Sign-On 인증, Google 또는 Facebook 자격 증명이 페더레이션형 ID의 예입니다. 페더레이션형 ID로 로그인할 때 관리자가 이전에 IAM 역할을 사용하여 아이덴티티 페더레이션을 설정했습니다. 페더레이션을 사용하여 AWS에 액세스하면 간접적으로 역할을 수임합니다.

사용자 유형에 따라AWS Management Console 또는AWS 액세스 포털에 로그인할 수 있습니다. AWS에 로그인하는 방법에 대한 자세한 내용은 AWS 로그인 사용 설명서의 AWS 계정에 로그인하는 방법을 참조하세요.

AWS에 프로그래밍 방식으로 액세스하는 경우, AWS에서는 보안 인증 정보를 사용하여 요청에 암호화 방식으로 서명할 수 있는 소프트웨어 개발 키트(SDK) 및 명령줄 인터페이스(CLI)를 제공합니다. AWS 도구를 사용하지 않는다면 요청에 직접 서명해야 합니다. 권장 방법을 사용하여 요청에 직접 서명하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 AWS API 요청에 서명을 참조하세요.

사용하는 인증 방법에 상관 없이 추가 보안 정보를 제공해야 할 수도 있습니다. 예를 들어, AWS는 멀티 팩터 인증(MFA)을 사용하여 계정의 보안을 강화하는 것을 권장합니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서의 다중 인증 및 IAM 사용 설명서의 AWS에서 다중 인증(MFA) 사용을 참조하세요.

AWS 계정 루트 사용자

AWS 계정을 생성할 때는 해당 계정의 모든 AWS 서비스 및 리소스에 대한 완전한 액세스 권한이 있는 단일 로그인 ID로 시작합니다. 이 자격 증명은 AWS 계정 루트 사용자라고 하며, 계정을 생성할 때 사용한 이메일 주소와 암호로 로그인하여 액세스합니다. 일상적인 태스크에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 보안 인증 정보를 보호하고 루트 사용자만 수행할 수 있는 태스크를 수행하는 데 사용합니다. 루트 사용자로 로그인해야 하는 태스크의 전체 목록은 AWS Account Management 참조 안내서의 루트 사용자 보안 인증이 필요한 태스크를 참조하세요.

연동 자격 증명

가장 좋은 방법은 관리자 액세스가 필요한 사용자를 포함한 사용자가 자격 증명 공급자와의 페더레이션을 사용하여 임시 보안 인증 정보를 사용하여 AWS 서비스에 액세스하도록 요구합니다.

연동 자격 증명은 엔터프라이즈 사용자 디렉터리, 웹 자격 증명 공급자, Directory Service, Identity Center 디렉터리의 사용자 또는 자격 증명 소스를 통해 제공된 자격 증명을 사용하여 AWS 서비스에 액세스하는 모든 사용자입니다. 연동 자격 증명은 AWS 계정에 액세스할 때 역할을 수임하고 역할은 임시 자격 증명을 제공합니다.

중앙 집중식 액세스 관리를 위해 AWS IAM Identity Center를 사용하는 것이 좋습니다. IAM Identity Center에서 사용자 및 그룹을 생성하거나 모든 AWS 계정 및 애플리케이션에서 사용하기 위해 고유한 자격 증명 소스의 사용자 및 그룹 집합에 연결하고 동기화할 수 있습니다. IAM Identity Center에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서에서 IAM Identity Center란 무엇입니까?를 참조하세요.

IAM 사용자 및 그룹

IAM 사용자는 단일 개인 또는 애플리케이션에 대한 특정 권한을 가지고 있는 AWS 계정 내 자격 증명입니다. 가능하면 암호 및 액세스 키와 같은 장기 보안 인증 정보가 있는 IAM 사용자를 생성하는 대신 임시 보안 인증을 사용하는 것이 좋습니다. 하지만 IAM 사용자의 장기 보안 인증이 필요한 특정 사용 사례가 있는 경우 액세스 키를 교체하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 장기 보안 인증이 필요한 사용 사례의 경우 정기적으로 액세스 키 교체를 참조하세요.

IAM 그룹은 IAM 사용자 컬렉션을 지정하는 자격 증명입니다. 귀하는 그룹으로 로그인할 수 없습니다. 그룹을 사용하여 여러 사용자의 권한을 한 번에 지정할 수 있습니다. 그룹을 사용하면 대규모 사용자

집합의 권한을 더 쉽게 관리할 수 있습니다. 예를 들어, IAMAdmins라는 그룹이 있고 이 그룹에 IAM 리소스를 관리할 권한을 부여할 수 있습니다.

사용자는 역할과 다릅니다. 사용자는 한 사람 또는 애플리케이션과 고유하게 연결되지만, 역할은 해당 역할이 필요한 사람이라면 누구나 수임할 수 있습니다. 사용자는 영구적인 장기 자격 증명을 가지고 있지만, 역할은 임시 자격 증명만 제공합니다. 자세한 정보는 IAM 사용 설명서의 IAM 사용자를 만들어야 하는 경우(역할이 아님)를 참조하세요.

IAM 역할

IAM 역할은 특정 권한을 가지고 있는 AWS 계정 계정 내 ID입니다. IAM 사용자와 유사하지만, 특정 개인과 연결되지 않습니다. 역할을 전환하여 AWS Management Console에서 IAM 역할을 임시로 수임할 수 있습니다. AWS CLI 또는 AWS API 태스크를 호출하거나 사용자 지정 URL을 사용하여 역할을 수임할 수 있습니다. 역할 사용 방법에 대한 자세한 정보는 IAM 사용 설명서의 IAM 역할 사용을 참조하세요.

임시 자격 증명이 있는 IAM 역할은 다음과 같은 상황에서 유용합니다.

- 페더레이션 사용자 액세스 - 페더레이션 아이덴티티에 권한을 부여하려면 역할을 생성하고 해당 역할의 권한을 정의합니다. 페더레이션 아이덴티티가 인증되면 역할이 연결되고 역할에 정의된 권한이 부여됩니다. 페더레이션 역할에 대한 자세한 내용은 IAM 사용 설명서의 Creating a role for a third-party Identity Provider(서드 파티 자격 증명 공급자의 역할 만들기) 부분을 참조하세요. IAM Identity Center를 사용하는 경우 권한 집합을 구성합니다. 인증 후 아이덴티티가 액세스할 수 있는 항목을 제어하기 위해 IAM Identity Center는 권한 세트를 IAM의 역할과 연관짓습니다. 권한 세트에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 권한 세트를 참조하세요.
- 임시 IAM 사용자 권한 - IAM 사용자 또는 역할은 IAM 역할을 수임하여 특정 태스크에 대한 다양한 권한을 임시로 받을 수 있습니다.
- 크로스 계정 액세스 - IAM 역할을 사용하여 다른 계정의 사용자(신뢰할 수 있는 보안 주체)가 내 계정의 리소스에 액세스하도록 허용할 수 있습니다. 역할은 크로스 계정 액세스를 부여하는 기본적인 방법입니다. 그러나 일부 AWS 서비스를 사용하면 역할을 (프록시로 사용하는 대신) 리소스에 정책을 직접 연결할 수 있습니다. 교차 계정 액세스를 위한 역할과 리소스 기반 정책의 차이점을 알아보려면 IAM 사용 설명서의 IAM 역할과 리소스 기반 정책의 차이를 참조하세요.
- 교차 서비스 액세스 – 일부 AWS 서비스는 다른 AWS 서비스의 기능을 사용합니다. 예를 들어 서비스에서 호출을 수행하면 일반적으로 해당 서비스는 Amazon EC2에서 애플리케이션을 실행하거나 Amazon S3에 객체를 저장합니다. 서비스는 호출하는 보안 주체의 권한을 사용하거나, 서비스 역할을 사용하거나, 또는 서비스 연결 역할을 사용하여 이 태스크를 수행할 수 있습니다.
  - 보안 주체 권한 - IAM 사용자 또는 역할을 사용하여 AWS에서 작업을 수행하는 사람은 보안 주체로 간주됩니다. 정책은 보안 주체에게 권한을 부여합니다. 일부 서비스를 사용할 때는 다른 서비스

에서 다른 태스크를 트리거하는 태스크를 수행할 수 있습니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. 작업에 정책의 추가 종속 작업이 필요한지 여부를 확인하려면 서비스 권한 부여 참조의 AWS 서비스에 대한 작업, 리소스 및 조건 키를 참조하세요.

- 서비스 역할 - 서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하기 위해 수임하는 IAM 역할입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 정보는 IAM 사용 설명서의 AWS 서비스에 대한 권한을 위임할 역할 생성을 참조하세요.
- 서비스 연결 역할 – 서비스 연결 역할은 AWS 서비스에 연결된 서비스 역할의 한 유형입니다. 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은 AWS 계정에 나타나고, 서비스가 소유합니다. IAM 관리자는 서비스 연결 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.

- Amazon EC2에서 실행 중인 애플리케이션 – IAM 역할을 사용하여 EC2 인스턴스에서 실행되고 AWS CLI 또는 AWS API 요청을 수행하는 애플리케이션의 임시 자격 증명을 관리할 수 있습니다. 이는 EC2 인스턴스 내에 액세스 키를 저장할 때 권장되는 방법입니다. EC2 인스턴스에 AWS 역할을 할당하고 해당 역할을 모든 애플리케이션에서 사용할 수 있도록 하려면 인스턴스에 연결된 인스턴스 프로파일을 생성합니다. 인스턴스 프로파일에는 역할이 포함되어 있으며 EC2 인스턴스에서 실행되는 프로그램이 임시 자격 증명을 얻을 수 있습니다. 자세한 정보는 IAM 사용 설명서의 IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여를 참조하세요.

IAM 역할을 사용할지 또는 IAM 사용자를 사용할지를 알아보려면 IAM 사용 설명서의 IAM 역할(사용자 대신)을 생성하는 경우를 참조하세요.

정책을 사용하여 액세스 관리

정책을 생성하고 AWS ID 또는 리소스에 연결하여 AWS 내 액세스를 제어합니다. 정책은 ID 또는 리소스와 연결될 때 해당 권한을 정의하는 AWS의 객체입니다. AWS는 보안 주체(사용자, 루트 사용자 또는 역할 세션)가 요청을 보낼 때 이러한 정책을 평가합니다. 정책에서 권한은 요청이 허용되거나 거부되는지를 결정합니다. 대부분의 정책은 AWS에 JSON 문서로서 저장됩니다. JSON 정책 문서의 구조와 콘텐츠에 대한 자세한 정보는 IAM 사용 설명서의 JSON 정책 개요를 참조하세요.

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지를 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

기본적으로, 사용자와 역할에는 어떠한 권한도 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다. 그런 다음 관리자가 IAM 정책을 역할에 추가하고, 사용자가 역할을 맡을 수 있습니다.

IAM 정책은 태스크를 수행하기 위해 사용하는 방법과 상관없이 작업에 대한 권한을 정의합니다. 예를 들어, `iam:GetRole` 태스크를 허용하는 정책이 있다고 가정합니다. 해당 정책이 있는 사용자는 AWS Management Console, AWS CLI 또는 AWS API에서 역할 정보를 가져올 수 있습니다.

자격 증명 기반 정책

자격 증명 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 자격 증명에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자와 역할이 어떤 리소스와 어떤 조건에서 어떤 태스크를 수행할 수 있는지를 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성을 참조하세요.

자격 증명 기반 정책은 인라인 정책 또는 관리형 정책으로 한층 더 분류할 수 있습니다. 인라인 정책은 단일 사용자, 그룹 또는 역할에 직접 포함됩니다. 관리형 정책은 AWS 계정에 속한 다수의 사용자, 그룹 및 역할에 독립적으로 추가할 수 있는 정책입니다. 관리형 정책에는 AWS 관리형 정책과 고객 관리형 정책이 포함되어 있습니다. 관리형 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용 설명서의 관리형 정책과 인라인 정책의 선택을 참조하세요.

리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 문서입니다. 리소스 기반 정책의 예는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 제어할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 태스크를 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 페더레이션 사용자 또는 AWS 서비스가 포함될 수 있습니다.

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. 리소스 기반 정책에서는 IAM의 AWS 관리형 정책을 사용할 수 없습니다.

ACL(액세스 제어 목록)

ACL(액세스 제어 목록)은 어떤 보안 주체(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 WAF 및 Amazon VPC는 ACL을 지원하는 대표적인 서비스입니다. ACL에 대해 자세히 알아보려면 Amazon Simple Storage Service 개발자 안내서의 ACL(액세스 제어 목록) 개요를 참조하세요.

기타 정책 유형

AWS는 비교적 일반적이지 않은 추가 정책 유형을 지원합니다. 이러한 정책 유형은 더 일반적인 정책 유형에 따라 사용자에게 부여되는 최대 권한을 설정할 수 있습니다.

- 권한 경계 – 권한 경계는 자격 증명 기반 정책에 따라 IAM 엔터티(IAM 사용자 또는 역할)에 부여할 수 있는 최대 권한을 설정하는 고급 기능입니다. 엔터티에 대한 권한 경계를 설정할 수 있습니다. 그 결과로 얻는 권한은 개체의 자격 증명 기반 정책과 그 권한 경계의 교집합입니다. `Principal` 필드에서 사용자나 역할을 보안 주체로 지정하는 리소스 기반 정책은 권한 경계를 통해 제한되지 않습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 권한 경계에 대한 자세한 정보는 IAM 사용 설명서의 IAM 엔터티에 대한 권한 경계를 참조하세요.
- 서비스 제어 정책(SCP) – SCP는 AWS Organizations에서 조직 또는 조직 단위(OU)에 최대 권한을 지정하는 JSON 정책입니다. AWS Organizations는 기업이 소유하는 여러 개의 AWS 계정을 그룹화하고 중앙에서 관리하기 위한 서비스입니다. 조직에서 모든 기능을 활성화할 경우 서비스 제어 정책(SCP)을 임의의 또는 모든 계정에 적용할 수 있습니다. SCP는 각 AWS 계정 루트 사용자를 비롯하여 멤버 계정의 엔터티에 대한 권한을 제한합니다. 조직 및 SCP에 대한 자세한 정보는 AWS Organizations 사용 설명서의 SCP 작동 방식을 참조하세요.
- 세션 정책 – 세션 정책은 역할 또는 연합된 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 결과적으로 얻는 세션의 권한은 사용자 또는 역할의 자격 증명 기반 정책의 교차와 세션 정책입니다. 또한 권한을 리소스 기반 정책에서 가져올 수도 있습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 자세한 정보는 IAM 사용 설명서의 세션 정책을 참조하세요.

여러 정책 유형

여러 정책 유형이 요청에 적용되는 경우 결과 권한은 이해하기가 더 복잡합니다. 여러 정책 유형이 관련될 때 AWS가 요청을 허용할지를 결정하는 방법을 알아보려면 IAM 사용 설명서의 정책 평가 로직을 참조하세요.

AWS 서비스에서 IAM을 사용하는 방식

AWS 서비스에서 대부분의 IAM 기능을 사용하는 방법을 전체적으로 알아보려면 IAM 사용 설명서의 IAM으로 작업하는 AWS 서비스를 참조하세요.

특정 AWS 서비스에 IAM을 사용하는 방법을 알아보려면 관련 서비스 사용 설명서의 보안 섹션을 참조하세요.

AWS 자격 증명 및 액세스 문제 해결

다음 정보를 사용하여 AWS 및 IAM에서 발생할 수 있는 공통적인 문제를 진단하고 수정할 수 있습니다.

주제

AWS에서 작업을 수행할 권한이 없음

작업을 수행할 권한이 없다는 오류가 수신되면, 작업을 수행할 수 있도록 정책을 업데이트해야 합니다.

다음 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 보안 인증 정보를 제공한 사람입니다.

iam:PassRole를 수행하도록 인증되지 않음

iam:PassRole 작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 AWS에 역할을 전달할 수 있도록 정책을 업데이트해야 합니다.

일부 AWS 서비스에서는 새 서비스 역할 또는 서비스 연결 역할을 생성하는 대신, 해당 서비스에 기존 역할을 전달할 수 있습니다. 이렇게 하려면 사용자가 서비스에 역할을 전달할 수 있는 권한을 가지고 있어야 합니다.

다음 예시 오류는 marymajor라는 IAM 사용자가 콘솔을 사용하여 AWS에서 작업을 수행하려고 하는 경우에 발생합니다. 하지만 작업을 수행하려면 서비스에 서비스 역할이 부여한 권한이 있어야 합니다. Mary는 서비스에 역할을 전달할 수 있는 권한을 가지고 있지 않습니다.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

이 경우 Mary가 `iam:PassRole` 작업을 수행할 수 있도록 Mary의 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하세요. 관리자는 로그인 보안 인증 정보를 제공한 사람입니다.

내 AWS 계정 외부의 사람이 내 AWS 리소스에 액세스할 수 있게 허용하기를 원합니다.

다른 계정의 사용자 또는 조직 외부의 사람이 리소스에 액세스하는 데 사용할 수 있는 역할을 생성할 수 있습니다. 역할을 수임할 신뢰할 수 있는 사람을 지정할 수 있습니다. 리소스 기반 정책 또는 ACL(액세스 제어 목록)을 지원하는 서비스의 경우 이러한 정책을 사용하여 다른 사람에게 리소스에 대한 액세스 권한을 부여할 수 있습니다.

자세히 알아보려면 다음을 참조하세요.

- AWS에서 이러한 기능을 지원하는지 여부를 알아보려면 AWS 서비스에서 IAM을 사용하는 방식 단원을 참조하세요.
- 소유하고 있는 AWS 계정의 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 자신이 소유한 다른 AWS 계정의 IAM 사용자에 대한 액세스 권한 제공을 참조하세요.
- 리소스에 대한 액세스 권한을 서드 파티 AWS 계정에게 제공하는 방법을 알아보려면 IAM 사용 설명서의 서드 파티가 소유한 AWS 계정에 대한 액세스 제공을 참조하세요.
- 자격 증명 연동을 통해 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 외부에서 인증된 사용자에게 액세스 권한 제공(자격 증명 연동)을 참조하세요.
- 교차 계정 액세스를 위한 역할과 리소스 기반 정책 사용의 차이점을 알아보려면 IAM 사용 설명서의 IAM 역할과 리소스 기반 정책의 차이를 참조하세요.

이 AWS 제품 또는 서비스에 대한 규정 준수 확인

AWS 서비스가 특정 규정 준수 프로그램의 범위에 포함되는지 알아보려면 규정 준수 프로그램 제공 범위 내 AWS 서비스를 참조하고 관심 있는 규정 준수 프로그램을 선택하세요. 일반적인 정보는 AWS 규정 준수 프로그램을 참조하세요.

AWS Artifact를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다. 자세한 내용은 AWS Artifact에서 보고서 다운로드를 참조하세요.

AWS 서비스 사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률과 규정에 따라 결정됩니다. AWS에서는 규정 준수를 지원할 다음과 같은 리소스를 제공합니다.

- 보안 및 규정 준수 빠른 시작 안내서 – 이 배포 안내서에서는 아키텍처 고려 사항에 대해 설명하고 보안 및 규정 준수에 중점을 둔 기본 AWS 환경을 배포하기 위한 단계를 제공합니다.
- Amazon Web Services에서 HIPAA 보안 및 규정 준수 기술 백서 설계 - 이 백서는 기업에서 AWS를 사용하여 HIPAA를 준수하는 애플리케이션을 만드는 방법을 설명합니다.

> **Note**
>
> 모든 AWS 서비스에 HIPAA 자격이 있는 것은 아닙니다. 자세한 내용은 HIPAA 적격 서비스 참조를 참조하세요.

- AWS 규정 준수 리소스 - 고객 조직이 속한 산업 및 위치에 적용될 수 있는 워크북 및 가이드 콜렉션입니다.
- AWS Config 개발자 안내서의 규칙을 사용하여 리소스 평가 – AWS Config 서비스는 내부 사례, 산업 지침 및 규제에 대한 리소스 구성의 준수 상태를 평가합니다.
- AWS Security Hub CSPM – 이 AWS 서비스는 AWS 내의 보안 상태에 대한 포괄적인 보기를 제공합니다. Security Hub는 보안 제어를 사용하여 AWS 리소스를 평가하고 보안 업계 표준 및 모범 사례에 대한 규정 준수를 확인합니다. 지원되는 서비스 및 제어 목록은 Security Hub 제어 참조를 참조하세요.
- AWS Audit Manager - 이 AWS 서비스는 AWS 사용을 지속해서 감사하여 위험을 관리하고 규정 및 업계 표준을 준수하는 방법을 간소화할 수 있도록 지원합니다.

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 준수 프로그램의 AWS 규정 준수 업무 범위에 속하는 AWS 서비스를 참조하세요.

이 AWS 제품 또는 서비스에 대한 복원력

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 중심으로 구축됩니다.

AWS 리전에서는 물리적으로 분리되고 격리된 다수의 가용 영역을 제공하며 이러한 가용 영역은 짧은 지연 시간, 높은 처리량 및 높은 중복성을 갖춘 네트워크에 연결되어 있습니다.

가용 영역을 사용하면 중단 없이 영역 간에 자동으로 장애 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 준수 프로그램의 AWS 규정 준수 업무 범위에 속하는 AWS 서비스를 참조하세요.

이 AWS 제품 또는 서비스에 대한 인프라 보안

이 AWS 제품 또는 서비스는 관리형 서비스를 사용하므로 AWS 글로벌 네트워크 보안으로 보호됩니다. AWS 보안 서비스와 AWS의 인프라 보호 방법에 대한 자세한 내용은 AWS 클라우드 보안을 참조하세요. 인프라 보안에 대한 모범 사례를 사용하여 AWS 환경을 설계하려면 보안 원칙 AWS Well-Architected Framework의 인프라 보호를 참조하세요.

AWS에서 게시한 API 호출을 사용하여 네트워크를 통해 이 AWS 제품 또는 서비스에 액세스합니다. 고객은 다음을 지원해야 합니다.

- 전송 계층 보안(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 주체와 관련된 보안 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 준수 프로그램의 AWS 규정 준수 업무 범위에 속하는 AWS 서비스를 참조하세요.

AWS Toolkit for JetBrains 사용 설명서에 대한 문서 이력

다음 표에는 AWS Toolkit for JetBrains 사용 설명서의 주요 문서 업데이트가 나열되어 있습니다.

AWS Toolkit for JetBrains의 자세한 변경 사항 목록은 GitHub 웹 사이트의 aws/aws-toolkit-jetbrains 리포지토리에 있는 .changes 디렉터리를 참조하세요.

변경 사항	설명	날짜
Amazon Elastic Container Service Exec 사용 설명서 작성	다음은 Amazon ECS Exec의 개요입니다.	2021년 12월 16일
실험 기능 지원	AWS 서비스에 대한 실험 기능을 활성화하기 위한 지원이 추가되었습니다.	2021년 10월 14일
AWS 리소스에 대한 지원	리소스 생성, 편집, 삭제를 위한 인터페이스 옵션과 함께 리소스 유형 액세스에 대한 지원이 추가되었습니다.	2021년 10월 14일
이제 AWS App Runner 작업 가능	AWS Toolkit for JetBrains를 사용한 App Runner 작업을 통해 소스 코드 또는 컨테이너 이미지에서 AWS Cloud의 확장 가능하고 안전한 웹 애플리케이션으로 직접 배포할 수 있습니다.	2021년 5월 26일
이제 서버리스 애플리케이션에서 Lambda 컨테이너 이미지 작업 가능	이제 AWS Toolkit를 사용하여 서버리스 애플리케이션에서 AWS Lambda 컨테이너 이미지 작업을 수행할 수 있습니다.	2020년 12월 1일
이제 CloudWatch Logs Insights 작업 가능	이제 AWS Toolkit for JetBrains를 사용하여 CloudWatch Logs Insights 작업을 수행할 수 있습니다.	2020년 11월 24일

이제 Amazon SQS 작업 가능	이제 AWS Toolkit for JetBrains를 사용하여 Amazon Simple Queue Service(Amazon SQS) 작업을 수행할 수 있습니다.	2020년 11월 24일
이제 Amazon RDS 및 Amazon Redshift 작업 가능	이제 AWS Toolkit를 사용하여 Amazon Relational Database Service(RDS) 및 Amazon Redshift 작업을 수행할 수 있습니다.	2020년 9월 23일
이제 IAM Identity Center 지원 사용 가능	이제 AWS Toolkit에서 AWS IAM Identity Center 지원을 사용할 수 있습니다.	2020년 9월 23일
이제 4개의 추가 JetBrains IDE에 AWS Toolkit 사용 가능	이제 다음 4개의 추가 JetBrains IDE용 플러그인으로 AWS Toolkit를 사용할 수 있습니다. • AWS Toolkit for CLion(C & C++ 개발용) • AWS Toolkit for GoLand(Go 개발용) • AWS Toolkit for PhpStorm(PHP 개발용) • AWS Toolkit for RubyMine(Ruby 개발용)	2020년 5월 28일
이제 CloudWatch Logs 작업 가능	이제 AWS Toolkit를 사용하여 Amazon CloudWatch Logs 작업을 수행할 수 있습니다.	2020년 4월 15일
이제 Amazon S3 버킷 및 객체 작업 가능	이제 AWS Toolkit를 사용하여 Amazon Simple Storage Service(S3) 버킷 및 객체 작업을 수행할 수 있습니다.	2020년 3월 27일

이제 EventBridge 스키마 작업 가능	이제 AWS Toolkit를 사용하여 Amazon EventBridge 스키마 작업을 수행할 수 있습니다.	2019년 12월 2일
이제 Amazon ECS 클러스터의 코드 디버깅 베타 이용 가능	이제 베타에서 AWS Toolkit를 사용하여 Amazon Elastic Container Service(Amazon ECS) 클러스터의 코드를 디버깅할 수 있습니다.	2019년 11월 25일
이제 AWS Toolkit for Rider 사용 가능	이제 AWS Toolkit for Rider를 사용할 수 있습니다.	2019년 11월 25일
이제 AWS Toolkit for WebStorm 사용 가능	이제 AWS Toolkit for WebStorm을 사용할 수 있습니다.	2019년 10월 23일
AWS Toolkit for IntelliJ 일반 공개	AWS Toolkit for IntelliJ가 일반 공개되었습니다. 그에 따라 해당 문서를 갱신했습니다.	2019년 3월 27일
최초 릴리스	이는 AWS Toolkit for JetBrains 사용 설명서의 최초 릴리스입니다. AWS Toolkit for PyCharm가 일반 공개되었습니다. AWS Toolkit for IntelliJ는 여전히 개발자 미리 보기 상태입니다.	2018년 11월 27일