사용자 가이드

AWS CodeStar

AWS CodeStar: 사용자 가이드

Table of Contents

2024년 7월 31일부터 Amazon Web Services (AWS) 는 프로젝트 생성 및 보기에 AWS CodeStar 대한 지원을 중단합니다. 2024년 7월 31일 이후에는 더 이상 AWS CodeStar 콘솔에 액세스하거나 새 프로젝트를 생성할 수 없습니다. 하지만 소스 리포지토리 AWS CodeStar, 파이프라인, 빌드를 포함하여 에서 생성한 AWS 리소스는 이번 변경의 영향을 받지 않고 계속 작동합니다. AWS CodeStar 연결 및 AWS CodeStar 알림은 이번 중단으로 인해 영향을 받지 않습니다.

작업을 추적하고, 코드를 개발하고, 애플리케이션을 구축, 테스트 및 배포하려는 경우 CodeCatalyst Amazon은 간소화된 시작 프로세스와 소프트웨어 프로젝트를 관리할 수 있는 추가 기능을 제공합니다. Amazon의 기능 및 가격에 대해 자세히 알아보십시오 CodeCatalyst.

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.

AWS CodeStar이란 무엇입니까?

AWS CodeStar는 AWS에서 소프트웨어 개발 프로젝트를 생성, 관리, 작업하기 위한 클라우드 기반 서비스입니다. AWS CodeStar 프로젝트를 통해 AWS에서 애플리케이션을 빠르게 개발, 빌드, 배포할 수 있습니다. AWS CodeStar 프로젝트는 프로젝트 개발 도구 체인에 대한 AWS 서비스를 생성 및 통합합니다. 선택한 AWS CodeStar 프로젝트 템플릿에 따라, 도구 체인에 소스 제어, 빌드, 배포, 가상 서버 또는 서버리스 리소스 등이 포함될 수 있습니다. 또한 AWS CodeStar는 (팀원이라고 하는) 프로젝트 사용자에 필요한 권한을 관리합니다. 프로젝트 소유자는 사용자를 AWS CodeStar 프로젝트에 팀원으로 추가하여 각 팀원의 역할에 따라 프로젝트 및 리소스에 대한 액세스 권한을 쉽고 빠르게 부여할 수 있습니다.

주제

AWS CodeStar로 할 수 있는 작업은 무엇입니까?

AWS CodeStar를 사용하여 클라우드상에서 애플리케이션 개발을 설정하고 하나의 중앙 집중식 대시보드에서 개발을 관리할 수 있습니다. 구체적으로 다음 작업이 가능합니다.

- 웹 애플리케이션, 웹 서비스 등에 대한 템플릿을 사용하여 AWS에서 몇 분 이내에 새 소프트웨어 프로젝트를 시작합니다. AWS CodeStar에는 다양한 프로젝트 유형과 프로그래밍 언어에 대한 프로젝트 템플릿이 포함되어 있습니다. AWS CodeStar에서 설정을 관리하므로 모든 프로젝트 리소스는 함께 작동하도록 구성됩니다.
- 팀에 대한 프로젝트 액세스 관리: AWS CodeStar는 프로젝트 팀원에게 도구 및 리소스에 액세스하는 데 필요한 역할을 할당할 수 있는 중앙 콘솔을 제공합니다. 이러한 권한은 프로젝트에 사용되는 모든 AWS 서비스에서 자동으로 적용되므로, 복잡한 IAM 정책을 생성하거나 관리할 필요가 없습니다.
- 한 곳에서 프로젝트에 대한 시각화, 운영 협업: AWS CodeStar에는 프로젝트, 도구 체인 및 중요 이벤트를 전체적으로 보여주는 프로젝트 대시보드가 포함되어 있습니다. 최신 프로젝트 활동(예: 최근 코드 커밋) 모니터링, 코드 변경, 빌드 결과 및 배포 상태 추적 등과 같은 작업을 모두 동일한 웹 페이지에서 수행할 수 있습니다. 단일 대시보드에서 프로젝트의 진행 상황을 모니터링하고 문제를 세부적으로 분석할 수 있습니다.
- 필요한 모든 도구로 신속하게 반복: AWS CodeStar에는 프로젝트에 대한 통합 개발 도구 체인이 포함되어 있습니다. 팀원이 코드를 푸시하면 변경 사항이 자동으로 배포됩니다. 문제 추적 기능을 통합

하여 팀원이 다음에 수행 작업을 추적할 수 있습니다. 사용자와 팀이 모든 코드 전달 단계에서 빠르고 효율적으로 협력할 수 있습니다.

AWS CodeStar을 시작하는 방법

AWS CodeStar을 시작하려면

1. AWS CodeStar설정의 단계에 따라 AWS CodeStar를 사용할 준비를 하십시오.
2. AWS CodeStar 시작하기 자습서의 단계에 따라 AWS CodeStar를 실험하십시오.
3. 프로젝트를 다른 개발자와 공유하려면 AWS CodeStar 프로젝트에 팀원 추가 의 단계를 따라 하십시오.
4. IDE를 AWS CodeStar와 함께 사용의 단계에 따라 즐겨 사용하는 IDE를 통합하십시오.

AWS CodeStar설정

사용을 AWS CodeStar시작하기 전에 다음 단계를 완료해야 합니다.

주제

1단계: 계정 생성

가입하세요. AWS 계정

계정이 없는 경우 다음 단계를 완료하여 계정을 만드세요. AWS 계정

가입하려면 AWS 계정

1. https://portal.aws.amazon.com/billing/signup을 여세요.
2. 온라인 지시 사항을 따르세요.

   등록 절차 중에는 전화를 받고 키패드로 인증 코드를 입력하는 과정이 있습니다.

   에 AWS 계정가입하면 AWS 계정 루트 사용자a가 생성됩니다. 루트 사용자에게는 계정의 모든 AWS 서비스 및 리소스 액세스 권한이 있습니다. 보안 모범 사례는 사용자에게 관리 액세스 권한을 할당하고, 루트 사용자만 사용하여 루트 사용자 액세스 권한이 필요한 작업을 수행하는 것입니다.

AWS 가입 절차가 완료된 후 확인 이메일을 보냅니다. 언제든지 https://aws.amazon.com/으로 가서 내 계정(My Account)을 선택하여 현재 계정 활동을 보고 계정을 관리할 수 있습니다.

관리자 액세스 권한이 있는 사용자 생성

등록한 AWS 계정후에는 일상적인 작업에 루트 사용자를 사용하지 않도록 관리 사용자를 보호하고 AWS IAM Identity Center활성화하고 생성하십시오 AWS 계정 루트 사용자.

보안을 유지하세요. AWS 계정 루트 사용자

1. 루트 사용자를 선택하고 AWS 계정 이메일 주소를 입력하여 계정 AWS Management Console소유자로 로그인합니다. 다음 페이지에서 비밀번호를 입력합니다.

   루트 사용자를 사용하여 로그인하는 데 도움이 필요하면AWS 로그인 사용 설명서의 루트 사용자로 로그인을 참조하세요.
2. 루트 사용자의 다중 인증(MFA)을 활성화합니다.

   지침은 IAM 사용 설명서의 AWS 계정 루트 사용자 (콘솔) 에 대한 가상 MFA 디바이스 활성화를 참조하십시오.

관리자 액세스 권한이 있는 사용자 생성

1. IAM Identity Center를 활성화합니다.

   지침은 AWS IAM Identity Center 사용 설명서의 AWS IAM Identity Center설정을 참조하세요.
2. IAM Identity Center에서 사용자에게 관리 액세스 권한을 부여합니다.

   를 ID 소스로 사용하는 방법에 대한 자습서는 사용 설명서의 기본값으로 IAM Identity Center 디렉터리사용자 액세스 구성을 참조하십시오. IAM Identity Center 디렉터리 AWS IAM Identity Center

관리 액세스 권한이 있는 사용자로 로그인

- IAM IDentity Center 사용자로 로그인하려면 IAM IDentity Center 사용자를 생성할 때 이메일 주소로 전송된 로그인 URL을 사용합니다.

  IAM Identity Center 사용자를 사용하여 로그인하는 데 도움이 필요하면 사용 설명서의 AWS 액세스 포털에 로그인을 참조하십시오.AWS 로그인

추가 사용자에게 액세스 권한 할당

1. IAM Identity Center에서 최소 권한 적용 모범 사례를 따르는 권한 세트를 생성합니다.

지침은AWS IAM Identity Center 사용 설명서의 Create a permission set를 참조하세요.

2. 사용자를 그룹에 할당하고, 그룹에 Single Sign-On 액세스 권한을 할당합니다.

   지침은AWS IAM Identity Center 사용 설명서의 Add groups를 참조하세요.

2단계: AWS CodeStar 서비스 역할 생성

사용자를 대신하여 AWS 리소스 및 IAM AWS CodeStar 권한을 관리할 수 있는 권한을 부여하는 데 사용되는 서비스 역할을 생성합니다. 서비스 역할은 한 번만 만들어야 합니다.

> **Important**
>
> 서비스 역할을 만들려면 관리 사용자(또는 루트 계정)로 로그인해야 합니다. 자세한 내용은 첫 번째 IAM 관리자 및 그룹 생성을 참조하세요.

1. https://console.aws.amazon.com/codestar/ 에서 AWS CodeStar 콘솔을 엽니다.
2. [Start project]를 선택합니다.

   프로젝트 시작이 보이지 않고 대신 프로젝트 목록 페이지로 이동한다면, 서비스 역할이 생성된 것입니다.
3. 서비스 역할 생성에서 예, 역할 생성을 선택합니다.
4. 마법사를 종료합니다. 나중에 다시 돌아옵니다.

3단계: 사용자의 IAM 권한 구성

관리 사용자 외에도 IAM 사용자, 연동 사용자, 루트 사용자 또는 수임된 AWS CodeStar 역할로 사용할 수 있습니다. IAM 사용자와 연동 사용자를 위해 수행할 AWS CodeStar 수 있는 작업에 대한 자세한 내용은 을 참조하십시오. AWS CodeStar IAM역할

IAM 사용자를 설정하지 않았다면 IAM 사용자 단원을 참조하세요.

액세스 권한을 제공하려면 사용자, 그룹 또는 역할에 권한을 추가하세요:

- 내 사용자 및 그룹: AWS IAM Identity Center

권한 세트를 생성합니다. AWS IAM Identity Center 사용 설명서의 권한 세트 생성의 지침을 따르세요.

- ID 제공자를 통해 IAM에서 관리되는 사용자:

  ID 페더레이션을 위한 역할을 생성합니다. IAM 사용 설명서의 서드 파티 자격 증명 공급자의 역할 만들기(연합)의 지침을 따르세요.
- IAM 사용자:
  - 사용자가 맡을 수 있는 역할을 생성합니다. IAM 사용 설명서에서 IAM 사용자의 역할 생성의 지침을 따르세요.
  - (권장되지 않음)정책을 사용자에게 직접 연결하거나 사용자를 사용자 그룹에 추가합니다. IAM 사용 설명서에서 사용자(콘솔)에 권한 추가의 지침을 따르세요.

4단계: AWS CodeStar 프로젝트에 대한 Amazon EC2 키 페어 만들기

많은 AWS CodeStar 프로젝트에서 AWS CodeDeploy 또는 AWS Elastic Beanstalk 를 사용하여 Amazon EC2 인스턴스에 코드를 배포합니다. 프로젝트와 연결된 Amazon EC2 인스턴스에 액세스하려면 IAM 사용자의 Amazon EC2 키 페어를 만듭니다. IAM 사용자에게 Amazon EC2 키를 만들고 관리할 권한이 있어야 합니다(예: `ec2:CreateKeyPair` 및 `ec2:ImportKeyPair` 작업을 수행할 수 있는 권한). 자세한 내용은 Amazon EC2 키 페어를 참조하세요.

5단계: 콘솔 열기 AWS CodeStar

에 로그인한 다음 https://console.aws.amazon.com/codestar/ 에서 AWS CodeStar 콘솔을 엽니다. AWS Management Console

다음 단계

축하합니다. 설정을 완료했습니다. 작업을 시작하려면 AWS CodeStar을 참조하십시오AWS CodeStar 시작하기.

AWS CodeStar 시작하기

이 자습서에서는 AWS CodeStar를 사용해 웹 애플리케이션을 생성합니다. 이 프로젝트의 소스 리포지토리, 지속적인 배포 도구 체인 및 프로젝트 대시보드에 샘플 코드가 포함되어 있으며, 이러한 위치에서 프로젝트를 보고 모니터링할 수 있습니다.

단계를 수행하면 다음 작업을 실행하게 됩니다.

- AWS CodeStar에서 프로젝트를 만듭니다.
- 프로젝트 탐색
- 코드 변경 커밋
- 자동으로 배포된 코드 변경 보기
- 프로젝트의 작업에 다른 사용자 추가
- 더 이상 필요하지 않은 프로젝트 리소스 정리

> **Note**
>
> 아직 하지 않았다면, 2단계: AWS CodeStar 서비스 역할 생성을 포함한 AWS CodeStar설정의 단계를 먼저 완료해 주십시오. IAM의 관리 사용자인 계정으로 로그인해야 합니다. 프로젝트를 생성하려면 **`AWSCodeStarFullAccess`**정책이 있는 IAM 사용자를 사용하여 AWS Management Console에 로그인해야 합니다.

주제

- 자습서: AWS CLI를 이용하여 AWS CodeStar에서 프로젝트 생성
- 자습서: AWS CodeStar에 Alexa Skill 프로젝트 생성
- 자습서: GitHub 소스 리포지토리를 사용하여 프로젝트 만들기

1단계: AWS CodeStar 프로젝트 만들기

이 단계에서는 웹 애플리케이션에 대한 JavaScript(Node.js) 소프트웨어 개발 프로젝트를 만듭니다. AWS CodeStar 프로젝트 템플릿을 사용하여 프로젝트를 만듭니다.

> **Note**
>
> 이 자습서에서 사용한 AWS CodeStar 프로젝트 템플릿은 다음과 같은 옵션을 사용합니다.
>
> - Application category: 웹 애플리케이션
> - Programming language: Node.js
> - AWS 서비스: Amazon EC2
>
> 다른 옵션을 선택할 경우 사용자 환경이 이 자습서에 설명된 것과 일치하지 않을 수 있습니다.

AWS CodeStar에서 프로젝트를 만들려면

1. 그런 다음 AWS Management Console에 로그인하고 https://console.aws.amazon.com/codestar/에서 AWS CodeStar 콘솔을 열 수 있습니다.

   프로젝트 및 그 리소스를 만들려는 AWS 리전에 로그인해야 합니다. 예를 들어 미국 동부(오하이오)에서 프로젝트를 만들려면 해당 AWS 리전을 선택해야 합니다. AWS CodeStar이 지원되는 AWS 리전에 대한 자세한 내용은 AWS 일반 참조의 리전 및 엔드포인트를 참조하세요.
2. AWS CodeStar 페이지에서 프로젝트 생성을 선택합니다.
3. 프로젝트 템플릿 선택 페이지의 AWS CodeStar 프로젝트 템플릿 목록에서 프로젝트 유형을 선택합니다. 필터 막대를 사용하여 선택 범위를 좁힐 수 있습니다. 예를 들어 Amazon EC2 인스턴스에 배포할 Node.js로 작성된 웹 애플리케이션 프로젝트의 경우 웹 애플리케이션, Node.js, Amazon EC2 확인란을 선택합니다. 그런 다음 해당 옵션 세트에 사용 가능한 템플릿 중에서 선택합니다.

   자세한 내용은 AWS CodeStar 프로젝트 템플릿 섹션을 참조하세요.
4. 다음을 선택합니다.

5. 프로젝트 이름 텍스트 입력 필드에 프로젝트 이름(예: *My First Project*)을 입력합니다. 프로젝트 ID에서 프로젝트 ID는 이 프로젝트 이름에서 파생되지만, 15자로 제한됩니다.

   예를 들어 *My First Project*라는 프로젝트의 기본 ID는 *my-first-projec*입니다. 이 프로젝트 ID는 해당 프로젝트와 관련된 모든 리소스 이름의 토대입니다. AWS CodeStar는 이 프로젝트 ID를 코드 리포지토리에 대한 URL의 일부로 사용하고, IAM의 관련 보안 액세스 역할 및 정책 이름으로도 사용합니다. 프로젝트를 만든 후에는 프로젝트 ID를 변경할 수 없습니다. 프로젝트를 만들기 전에 프로젝트 ID를 편집하려면 프로젝트 ID에 사용하려는 ID를 입력합니다.

   프로젝트 이름 및 프로젝트 ID 제한에 대한 자세한 내용은 AWS CodeStar의 제한 값 단원을 참조하십시오.

   > **Note**
   >
   > 프로젝트 ID는 AWS 리전 내 AWS 계정에 대해 고유해야 합니다.

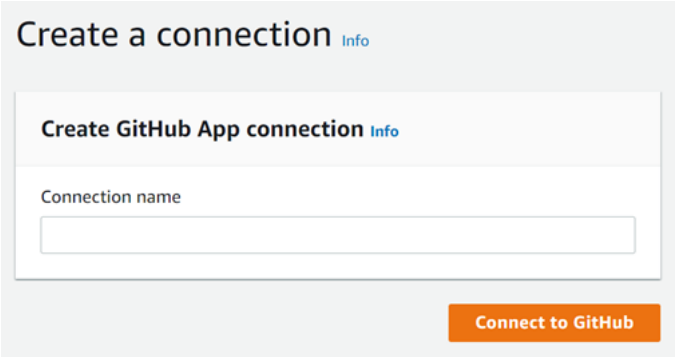
6. 리포지토리 공급자 AWS CodeCommit 또는 GitHub를 선택합니다.

7. AWS CodeCommit를 선택한 경우 리포지토리 이름에는 기본 AWS CodeCommit 리포지토리 이름을 그대로 이용하거나 다른 이름을 입력합니다. 그런 다음 9단계로 건너뜁니다.

8. GitHub를 선택한 경우 연결 리소스를 선택하거나 생성해야 합니다. 기존 연결이 있는 경우 검색 필드에서 해당 연결을 선택합니다. 기존 연결이 없는 경우 지금 새 연결을 생성합니다. GitHub에 연결을 선택합니다.

   연결 생성 페이지가 표시됩니다.

   > **Note**
   >
   > 연결을 생성하려면 GitHub 계정이 필요합니다. 조직 연결을 생성하는 경우 조직 소유자여야 합니다.



a. GitHub 앱 연결 생성 아래의 연결 이름 입력 텍스트 필드에 연결 이름을 입력합니다. GitHub에 연결을 선택합니다.

   GitHub에 연결 페이지가 나타나고 GitHub 앱 필드가 표시됩니다.

b. GitHub 앱에서 앱 설치를 선택하거나 새 앱 설치를 선택하여 앱을 새로 만듭니다.

> **Note**
>
> 특정 공급자에 대한 모든 연결에 대해 하나의 앱을 설치합니다. 이미 설치한 경우 AWS Connector for GitHub 앱을 선택한 다음 이 단계를 건너뜁니다.

c. AWS Connect for GitHub 설치 페이지에서 앱을 설치할 계정을 선택합니다.

> **Note**
>
> 이전에 앱을 설치한 경우 구성을 선택하여 앱 설치의 수정 페이지로 이동하거나 뒤로 버튼을 사용하여 콘솔로 돌아갈 수 있습니다.

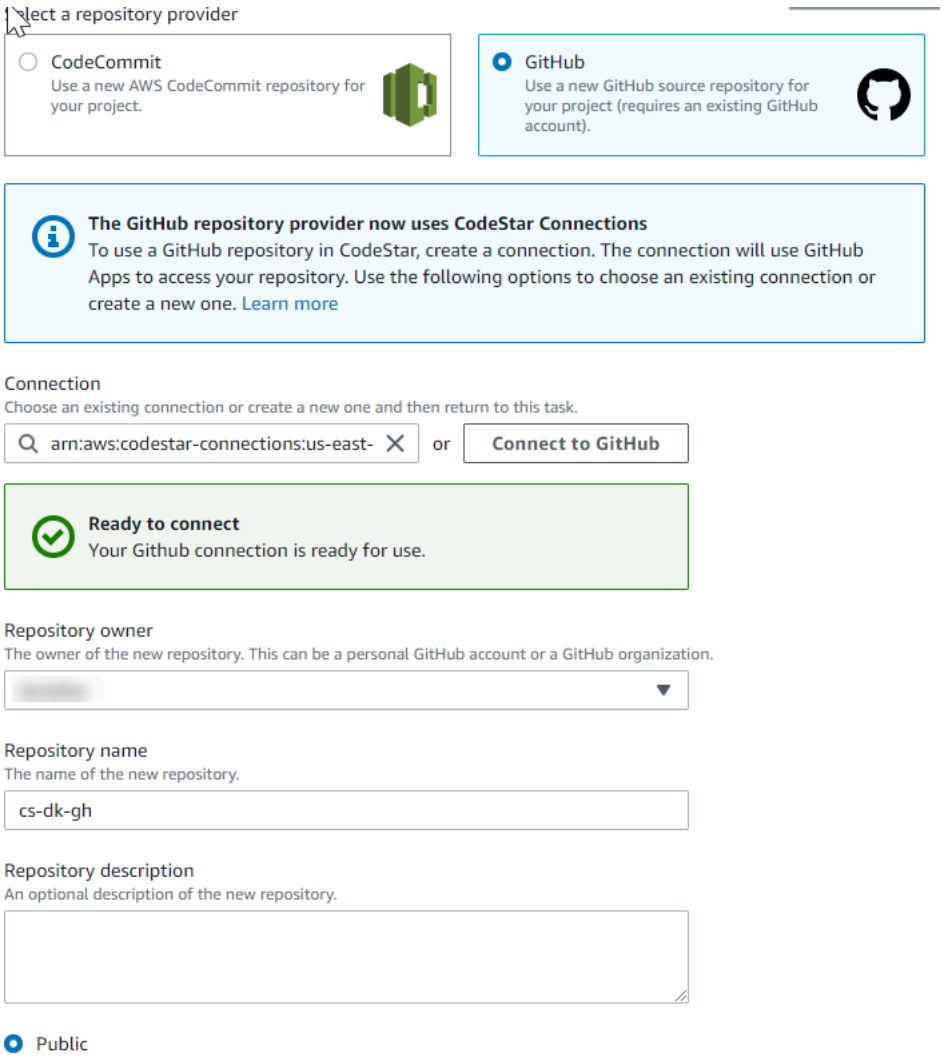
d. 계속하려면 암호 확인 페이지가 표시되면 GitHub 암호를 입력한 다음 로그인을 선택합니다.

e. GitHub용 AWS 커넥터 설치 페이지에서 기본값을 그대로 두고 설치를 선택합니다.

f. GitHub에 연결 페이지의 GitHub 앱 텍스트 입력 필드에 새 설치의 설치 ID가 표시됩니다.

   연결이 생성되면 CodeStar 프로젝트 생성 페이지에 연결 준비 완료 메시지가 표시됩니다.

> **Note**
>
> 개발자 도구 콘솔의 설정에서 연결을 볼 수 있습니다. 자세한 정보는 연결 시작하기를 참조하십시오.



g. 리포지토리 소유자의 경우 GitHub 조직이나 사용자의 개인 GitHub 계정을 선택합니다.

h. 리포지토리 이름에는 기본 GitHub 리포지토리 이름을 그대로 이용하거나 다른 이름을 입력합니다.

i. 퍼블릭 또는 프라이빗을 선택합니다.

> **Note**
>
> 또한 개발 환경으로 AWS Cloud9를 사용하려면 퍼블릭을 선택해야 합니다.

j. (선택 사항) 리포지토리 설명에 GitHub 리포지토리에 대한 설명을 입력합니다.

> **Note**
>
> Alexa Skill 프로젝트 템플릿을 선택하는 경우 Amazon 개발자 계정을 연결해야 합니다. Alexa Skill 프로젝트 작업에 대한 자세한 내용은 자습서: AWS CodeStar에 Alexa Skill 프로젝트 생성 단원을 참조하십시오.

9. 프로젝트가 Amazon EC2 인스턴스에 배포되어 있고 변경하려는 경우 Amazon EC2 구성에서 Amazon EC2 인스턴스를 구성하십시오. 예를 들어, 프로젝트에 사용 가능한 인스턴스 유형 중에서 선택할 수 있습니다.

> **Note**
>
> Amazon EC2 인스턴스 유형마다 서로 다른 수준의 컴퓨팅 성능을 제공하므로 관련 비용도 다를 수 있습니다. 자세한 내용은 Amazon EC2 인스턴스 유형 및 Amazon EC2 요금을 참조하세요.
>
> Amazon Virtual Private Cloud에서 만든 서브넷이 여러 개이거나 Virtual Private Cloud(VPC)가 두 개 이상인 경우, 사용할 서브넷 및 VPC를 선택할 수도 있습니다. 그러나 전용 인스턴스에서 지원되지 않는 Amazon EC2 인스턴스 유형을 선택하는 경우, 인스턴스 테넌시가 전용으로 설정된 VPC를 선택할 수 없습니다.
>
> 자세한 내용은 Amazon VPC란 무엇입니까?와 전용 인스턴스 기본 사항 단원을 참조하십시오.

키 페어에서 4단계: AWS CodeStar 프로젝트에 대한 Amazon EC2 키 페어 만들기에서 생성한 Amazon EC2 키 페어를 선택합니다. 프라이빗 키 파일에 대한 액세스 권한이 있음을 인정함을 선택합니다.

10. 다음을 선택합니다.
11. 리소스와 구성 세부 정보를 검토합니다.

12. [Next] 또는 [Create project]를 선택합니다. (표시되는 선택 사항은 프로젝트 템플릿에 따라 다릅니다.)

    프로젝트(리포지토리 포함)를 만드는 데 몇 분 정도 걸릴 수 있습니다.

13. 프로젝트에 리포지토리가 있으면 리포지토리 페이지를 사용하여 리포지토리에 대한 액세스를 구성할 수 있습니다. 다음 단계의 링크를 사용하여 IDE를 구성하거나, 이슈 트래킹을 설정하거나, 프로젝트에 팀 구성원을 추가할 수 있습니다.

2단계: AWS CodeStar 사용자 프로필에 대한 표시 정보 추가

프로젝트를 만들면 프로젝트 팀에 소유자로 추가됩니다. AWS CodeStar를 처음 사용한다면 다음을 입력해야 합니다.

- 다른 사용자에게 표시할 표시 이름
- 다른 사용자에게 표시할 이메일 주소

이 정보는 AWS CodeStar 사용자 프로필에 사용됩니다. 사용자 프로필은 프로젝트에 특정하지 않고 AWS 리전으로 한정됩니다. 자신이 프로젝트에 속해 있는 모든 AWS 리전별로 사용자 프로필을 하나씩 생성해야 합니다. 원하는 경우 각 프로필에 다른 정보를 포함할 수 있습니다.

사용자 이름과 이메일 주소를 입력한 다음 다음을 선택합니다.

> **Note**
>
> 이 사용자 이름 및 이메일 주소는 AWS CodeStar 사용자 프로필에 사용됩니다. 프로젝트가 GitHub 리포지토리 또는 Atlassian JIRA의 문제 같은 AWS 외부의 리소스를 사용하는 경우, 리소스 공급자는 서로 다른 사용자 이름 및 이메일 주소를 가진 자체 사용자 프로필을 사용할 수 있습니다. 자세한 내용은 리소스 공급자 설명서를 참조하십시오.

3단계: 프로젝트 보기

AWS CodeStar 프로젝트 대시보드에서 사용자와 사용자의 팀은 프로젝트에 대한 최신 커밋, 지속적 제공 파이프라인의 상태, 인스턴스 성능을 비롯한 프로젝트 리소스의 상태를 확인합니다. 이러한 리소스에 대한 자세한 내용을 보려면 탐색 모음에서 해당 페이지를 선택하세요.

새 프로젝트의 탐색 막대에는 다음과 같은 페이지가 있습니다.

- 개요 페이지에는 프로젝트 활동, 프로젝트 리소스, 프로젝트의 README 콘텐츠에 대한 정보가 들어 있습니다.
- IDE 페이지에서 프로젝트를 통합 개발 환경(IDE)에 연결하여 소스 코드 변경을 수정, 테스트 및 푸시합니다. 여기에는 GitHub 및 AWS CodeCommit 리포지토리 모두에 대한 IDE를 구성하는 지침과 AWS Cloud9 환경에 대한 정보가 포함되어 있습니다.
- 리포지토리 페이지에는 이름, 제공자, 최종 수정 날짜, 복제 URL 등 리포지토리 세부 정보가 표시됩니다. 또한 가장 최근의 커밋에 대한 정보를 확인하고 Pull 요청을 보고 생성할 수 있습니다.
- 파이프라인 페이지에는 파이프라인에 대한 CI/CD 정보가 표시됩니다. 이름, 가장 최근 작업, 상태와 같은 파이프라인 세부 정보를 볼 수 있습니다. 파이프라인 기록을 확인하고 변경 사항을 릴리스할 수 있습니다. 파이프라인의 개별 단계 상태를 볼 수도 있습니다.
- 모니터링 페이지에는 프로젝트 구성에 따라 Amazon EC2 또는 AWS Lambda 지표가 표시됩니다. 예를 들어 파이프라인의 AWS Elastic Beanstalk 또는 CodeDeploy 리소스에 배포된 모든 Amazon EC2 인스턴스의 CPU 사용률이 표시됩니다. AWS Lambda를 사용하는 프로젝트에서는 Lambda 함수에 대한 간접 호출 및 오류 지표가 표시됩니다. 이 정보는 시간별로 표시됩니다. 이 자습서에 대해 제안된 AWS CodeStar 프로젝트 템플릿을 사용한 경우 애플리케이션이 인스턴스에 처음 배포될 때 활동이 급격하게 증가해야 합니다. 모니터링을 새로 고쳐서 인스턴스 상태 변경을 볼 수 있습니다. 그러면 더 많은 리소스의 문제점이나 요구 사항을 파악하는 데 도움이 됩니다.
- 이슈 페이지는 AWS CodeStar 프로젝트를 Atlassian JIRA 프로젝트와 통합하기 위한 것입니다. 사용자와 프로젝트 팀은 이 타일을 구성하여 프로젝트 대시보드에서 JIRA 문제를 추적할 수 있습니다.

콘솔 왼쪽의 탐색 창에서는 프로젝트, 팀 및 설정 페이지 사이를 탐색할 수 있습니다.

4단계: 변경 커밋

먼저 프로젝트에 포함된 샘플 애플리케이션을 살펴보고 애플리케이션의 모양에 대해 알아보십시오. 프로젝트 탐색의 어느 곳에서든 애플리케이션 보기를 선택하여 애플리케이션이 어떻게 보이는지 확인할 수 있습니다. 샘플 웹 애플리케이션이 새 창 또는 브라우저 탭에 표시됩니다. 이는 AWS CodeStar가 빌드하고 배포한 프로젝트 샘플입니다.

코드를 보려면 탐색 모음에서 리포지토리를 선택합니다. 리포지토리 이름 아래의 링크를 선택하면 프로젝트 리포지토리가 새 탭 또는 창에 열립니다. 리포지토리 추가 정보 파일(README.md)의 내용을 읽고 해당 파일의 콘텐츠를 찾아봅니다.

이 단계에서는 코드를 변경한 다음 해당 변경 사항을 리포지토리에 푸시합니다. 이 작업을 여러 방법으로 수행할 수 있습니다.

- 프로젝트의 코드가 CodeCommit나 GitHub 리포지토리에 저장되어 있다면, 도구 설치 없이 AWS Cloud9을 웹 브라우저에서 바로 코드와 함께 사용할 수 있습니다. 자세한 내용은 프로젝트에 대한 AWS Cloud9 환경 생성 섹션을 참조하세요.
- 프로젝트의 코드가 CodeCommit 리포지토리에 저장되어 있고 Visual Studio 또는 Eclipse가 설치되어 있다면, AWS Toolkit for Visual Studio 또는 AWS Toolkit for Eclipse를 사용하여 코드에 더욱 쉽게 연결할 수 있습니다. 자세한 내용은 IDE를 AWS CodeStar와 함께 사용 섹션을 참조하세요. Visual Studio 또는 Eclipse가 없다면 경우 Git 클라이언트를 설치하고 이 단계의 뒷부분에 나오는 지침을 따릅니다.
- 프로젝트의 코드가 GitHub 리포지토리에 저장되어 있는 경우 IDE 도구를 사용하여 GitHub에 연결할 수 있습니다.
  - Visual Studio의 경우 Visual Studio용 GitHub Extension과 같은 도구를 사용할 수 있습니다. 자세한 내용은 Visual Studio용 GitHub 웹 사이트의 개요 페이지 및 GitHub 웹 사이트의 Getting Started with GitHub for Visual Studio를 참조하십시오.
  - Eclipse의 경우 Eclipse용 EGit과 같은 도구를 사용할 수 있습니다. 자세한 내용은 EGit 웹 사이트에 있는 EGit 설명서를 참조하십시오.
  - 다른 IDE에 대해서는 IDE 문서를 참조하십시오.
- 다른 유형의 코드 리포지토리의 경우 리포지토리 공급자 설명서를 참조하십시오.

다음 지침은 샘플에 사소한 변경사항을 적용하는 방법을 보여 줍니다.

변경을 커밋하도록 컴퓨터를 설정하려면(IAM 사용자)

> **Note**
>
> 이 절차에서는 프로젝트의 코드가 CodeCommit 리포지토리에 저장되어 있다고 가정합니다. 다른 유형의 코드 리포지토리의 경우 리포지토리 공급자 설명서를 참조한 후 다음 절차인 프로젝트 리포지토리를 복제하고 변경하려면으로 건너뜁니다.
> 코드가 CodeCommit에 저장되었고 이미 CodeCommit를 사용 중이거나 AWS CodeStar 콘솔을 이용해 프로젝트에 대한 AWS Cloud9 개발 환경을 만들었다면, 추가 구성은 필요 없습니다. 다음 절차인 프로젝트 리포지토리를 복제하고 변경하려면으로 건너뜁니다.

1. 로컬 컴퓨터에 Git를 설치합니다.
2. AWS Management Console에 로그인하여 https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.

CodeCommit에서 AWS CodeStar 프로젝트 리포지토리 접속을 위해 Git 자격 증명을 사용할 IAM 사용자로 로그인합니다.

3. IAM 콘솔의 탐색 창에서 사용자를 선택하고 사용자 목록에서 해당 IAM 사용자를 선택합니다.
4. 사용자 세부 정보 페이지에서 보안 인증 정보 탭을 선택하고 CodeCommit의 HTTPS Git 보안 인증 정보에서 생성을 선택합니다.

> **Note**
>
> Git 자격 증명에 대한 고유한 로그인 자격 증명을 선택할 수는 없습니다. 이에 관한 자세한 내용은 CodeCommit으로 Git 보안 인증 정보 및 HTTPS 사용을 참조하십시오.

5. IAM이 생성한 로그인 자격 증명을 복사하십시오. [Show]를 선택한 다음 이 정보를 복사하여 로컬 컴퓨터에 있는 안전한 파일에 붙여넣거나, [Download credentials]를 선택하여 이 정보를 .CSV 파일로 다운로드할 수 있습니다. CodeCommit에 접속하려면 이 정보가 필요합니다.

   자격 증명을 저장한 후 닫기를 선택합니다.

> **Important**
>
> 이때가 사용자가 로그인 자격 증명을 저장할 수 있는 유일한 기회입니다. 이 정보를 저장하지 않는 경우, 사용자 이름은 IAM 콘솔에서 복사할 수 있지만 암호는 찾을 수 없습니다. 그러므로 암호를 재설정한 후 저장해야 합니다.

변경을 커밋하도록 컴퓨터를 설정하려면(연합된 사용자)

콘솔을 이용해 리포지토리에 파일을 업로드하거나, Git를 이용해 로컬 컴퓨터에서 연결할 수 있습니다. 연동된 액세스를 이용한다면, 이러한 단계를 밟아 Git를 이용해 로컬 컴퓨터에서 리포지토리를 연결하고 복제하십시오.

> **Note**
>
> 이 절차에서는 프로젝트의 코드가 CodeCommit 리포지토리에 저장되어 있다고 가정합니다. 다른 유형의 코드 리포지토리의 경우 리포지토리 공급자 설명서를 참조한 후 다음 절차인 프로젝트 리포지토리를 복제하고 변경하려면으로 건너뜁니다.

1. 로컬 컴퓨터에 Git를 설치합니다.

2. AWS CLI를 설치합니다.
3. 연합된 사용자에 대한 임시 보안 자격 증명을 구성합니다. 자세한 내용은 CodeCommit 리포지토리에 대한 임시 액세스 단원을 참조하십시오. 임시 자격 증명은 다음 요소로 구성됩니다.

   - AWS 액세스 키
   - AWS 비밀 키
   - 세션 토큰

   임시 자격 증명에 대한 자세한 내용은 GetFederationToken에 대한 권한 단원을 참조하십시오.
4. AWS CLI 보안 인증 도우미를 이용해 리포지토리를 연결합니다. 자세한 내용은 AWS CLI 보안 인증 도우미를 사용하여 Linux, macOS 또는 Unix에서 CodeCommit 리포지토리에 HTTPS 연결을 설정하는 단계나 AWS CLI 보안 인증 도우미를 사용하여 Windows에서 CodeCommit 리포지토리에 HTTPS 연결을 설정하는 단계를 참조하십시오.
5. 다음 예제는 CodeCommit 리포지토리에 연결하고 커밋을 리포지토리로 푸시하는 방법을 보여줍니다.

예제: 프로젝트 리포지토리를 복제하고 변경하려면

> **Note**
>
> 이 절차는 프로젝트의 코드 리포지토리를 컴퓨터에 복제하고 프로젝트의 `index.html` 파일을 변경한 다음 변경 내용을 원격 리포지토리에 푸시하는 방법을 보여 줍니다. 이 절차에서는 프로젝트의 코드가 CodeCommit 리포지토리에 저장되어 있고, 명령줄에서 Git 클라이언트를 사용하고 있다고 가정합니다. 다른 유형의 코드 리포지토리 또는 도구의 경우, 리포지토리를 복제하고 파일을 변경한 다음 코드를 푸시하는 방법에 대한 공급자 설명서를 참조하십시오.

1. AWS CodeStar 콘솔을 이용해 프로젝트에 대한 AWS Cloud9 개발 환경을 만들었다면, 개발 환경을 열고 이 절차의 3단계로 건너뛰십시오. 개발 환경을 여는 방법은 프로젝트에 대한 AWS Cloud9 환경 열기 단원을 참조하십시오.

   AWS CodeStar 콘솔에서 프로젝트를 열어 둔 상태로 탐색 모음에서 리포지토리를 선택합니다. 복제 URL에서 CodeCommit에 대해 설정한 연결 유형에 대한 프로토콜을 선택한 다음 링크를 복사합니다. 예를 들어 이전 절차의 단계에 따라 CodeCommit에 대한 Git 자격 증명을 설정한 경우 HTTPS를 선택합니다.

2. 로컬 컴퓨터에서 터미널 또는 명령줄 창을 열고 디렉터리를 임시 디렉터리로 변경합니다. git clone 명령을 실행하여 리포지토리를 컴퓨터에 복제합니다. 복사한 링크를 붙여넣습니다. 예를 들어 CodeCommit에 대해 HTTPS를 사용하는 경우:

```
git clone https://git-codecommit.us-east-2.amazonaws.com/v1/repos/my-first-projec
```

처음 접속하는 경우에는 리포지토리에 대한 로그인 자격 증명을 묻는 메시지가 표시됩니다. CodeCommit의 경우 이전 절차에서 다운로드한 Git 자격 증명 로그인 자격 증명을 입력합니다.

3. 컴퓨터의 복제 디렉터리로 이동하여 콘텐츠를 찾아봅니다.
4. 퍼블릭 폴더에서 `index.html` 파일을 열고 파일을 변경합니다. 예를 들어 <H2> 태그 뒤에 다음과 같은 단락을 추가합니다.

```
<P>Hello, world!</P>
```

파일을 저장합니다.

5. 터미널 또는 명령 프롬프트에서 변경된 파일을 추가한 다음 변경 사항을 커밋하고 푸시합니다.

```
git add index.html
git commit -m "Making my first change to the web app"
git push
```

6. 리포지토리 페이지에서 진행 중인 변경 사항을 확인하십시오. 리포지토리의 커밋 이력이 사용자의 커밋으로 업데이트되고 커밋 메시지가 표시됩니다. 또한 파이프라인 페이지에서 리포지토리에 대한 변경 사항을 선택하고 이를 빌드 및 배포하기 시작합니다. 앱 애플리케이션을 배포한 후 애플리케이션 보기를 선택하여 변경 사항을 확인할 수 있습니다.

> **Note**
>
> 파이프라인의 어느 단계에서든 실패가 표시된다면, 문제 해결을 위해 다음 단계를 확인합니다.
>
> - 소스 단계에 대해서는 AWS CodeCommit사용 설명서의 AWS CodeCommit 문제 해결을 참조하십시오.
> - 빌드 단계에 대해서는 AWS CodeBuild사용 설명서의 AWS CodeBuild 문제 해결을 참조하십시오.

- 배포 단계에 대해서는 AWS CloudFormation사용 설명서의 AWS CloudFormation 문제 해결을 참조하십시오.
- 그 밖의 문제는 문제 해결 AWS CodeStar 단원을 참조하십시오.

6단계: 팀원 추가

모든 AWS CodeStar 프로젝트는 세 가지 AWS CodeStar 역할로 미리 구성되어 있습니다. 각 역할은 프로젝트와 해당 리소스에 대한 자체 액세스 수준을 제공합니다.

- 소유자: 팀원을 추가 및 제거하고, 프로젝트 대시보드를 변경하고, 프로젝트를 삭제할 수 있습니다.
- 기고자: 코드가 CodeCommit에 저장되어 있는 경우 프로젝트 대시보드를 변경하고 코드를 제공할 수는 있으나, 팀원을 추가 또는 제거하거나 프로젝트를 삭제할 수는 없습니다. AWS CodeStar 프로젝트에서 대부분의 팀원에 대해 이 역할을 선택해야 합니다.
- 최종 사용자: 코드가 CodeCommit에 저장되어 있는 경우 프로젝트 대시보드, 프로젝트 코드 및 프로젝트 상태는 볼 수 있으나, 프로젝트 대시보드에서 타일을 이동, 추가 또는 제거할 수는 없습니다.

> **Important**
>
> 프로젝트가 GitHub 리포지토리 또는 Atlassian JIRA의 문제 같은 AWS 외부의 리소스를 사용하는 경우, 해당 리소스에 대한 액세스는 AWS CodeStar가 아닌 리소스 제공자가 제어합니다. 자세한 내용은 리소스 공급자 설명서를 참조하십시오.
> AWS CodeStar 프로젝트에 액세스할 수 있는 사용자는 AWS CodeStar 콘솔을 사용하여 AWS 외부에 있지만 해당 프로젝트와 관련된 리소스에 액세스할 수 있습니다.
> AWS CodeStar는 팀원이 프로젝트에 관한 어떠한 관련 AWS Cloud9 개발 환경에도 참가하도록 허용하지 않습니다. 팀원이 공유 환경에 참여하도록 허용하는 방법은 프로젝트 팀원과 AWS Cloud9 환경 공유 단원을 참조하십시오.

팀 및 프로젝트 역할에 대한 자세한 내용은 AWS CodeStar 팀 작업을 참조하십시오.

AWS CodeStar 프로젝트에 팀원을 추가하려면(콘솔)

1. https://console.aws.amazon.com/codestar/에서 AWS CodeStar 콘솔을 엽니다.

2. 탐색 창에서 프로젝트를 선택한 후 프로젝트를 선택합니다.
3. 프로젝트의 측면 탐색 모음에서 팀을 선택합니다.
4. [Team members] 페이지에서 [Add team member]를 선택합니다.
5. 사용자 선택에서 다음 중 하나를 수행합니다.

   - 추가할 사람의 IAM 사용자가 이미 있다면, 목록에서 해당 IAM 사용자 이름을 선택합니다.

     > **Note**
     >
     > 다른 AWS CodeStar 프로젝트에 이미 추가한 사용자는 기존 AWS CodeStar 사용자 목록에 표시됩니다.

     프로젝트 역할에서 이 사용자에게 부여할 AWS CodeStar 역할(소유자, 기고자, 최종 사용자)을 선택합니다. 이는 프로젝트의 소유자만 변경할 수 있는 AWS CodeStar 프로젝트 수준 역할입니다. IAM 사용자에게 적용하면 이 역할은 AWS CodeStar 프로젝트 리소스에 액세스하는 데 필요한 모든 권한을 제공합니다. 이는 IAM에서 CodeCommit에 저장된 코드에 대한 Git 자격 증명을 만들고 관리하거나 IAM 사용자의 Amazon EC2 SSH 키를 업로드하는 데 필요한 정책을 적용합니다.

     > **Important**
     >
     > IAM 사용자의 표시 이름 또는 이메일 정보를 입력하거나 변경할 수 없습니다. 해당 사용자로 콘솔에 로그인한 경우에만 가능합니다. 자세한 내용은 AWS CodeStar 사용자 프로필에 대한 표시 정보 관리 섹션을 참조하세요.

     팀원 추가를 선택합니다.

   - 프로젝트에 추가할 사람의 IAM 사용자가 없다면, 새 IAM 사용자 생성을 선택합니다. 새 IAM 사용자를 생성할 수 있는 IAM 콘솔로 리디렉션됩니다. 자세한 내용은 IAM 사용 설명서의 IAM 사용자 생성을 참조하십시오. IAM 사용자를 생성한 후 AWS CodeStar 콘솔로 돌아가 사용자 목록을 새로 고치고 드롭다운 목록에서 생성한 IAM 사용자를 선택합니다. 이 새 사용자에게 적용할 AWS CodeStar 표시 이름, 이메일 주소, 프로젝트 역할을 입력한 다음 팀원 추가를 선택합니다.

> **Note**
>
> 관리하기 쉽도록 하나 이상의 사용자에 프로젝트의 소유자 역할이 할당되어 있어야 합니다.

6. 새 팀원에게 다음 정보를 보냅니다.

   - AWS CodeStar 프로젝트의 연결 정보
   - 소스 코드가 CodeCommit에 저장되어 있는 경우 로컬 컴퓨터에서 Git 자격 증명을 사용하여 CodeCommit 리포지토리에 액세스하도록 설정하는 방법에 대한 지침입니다.
   - AWS CodeStar 사용자 프로필 작업에 설명된 대로 사용자가 표시 이름, 이메일 주소, 퍼블릭 Amazon EC2 SSH 키를 관리하는 방법에 대한 정보입니다.
   - 일회용 암호와 연결 정보(사용자가 AWS를 처음 사용하며 해당 사용자의 IAM 사용자를 만든 경우) 암호는 사용자가 처음 로그인하면 만료됩니다. 사용자는 새 암호를 선택해야 합니다.

6단계: 정리

축하합니다! 본 자습서를 완수하셨습니다. 이 프로젝트 및 리소스 사용을 중단하고 싶다면 프로젝트를 삭제하여 AWS 계정에 요금이 계속 청구되지 않게 해야 합니다.

AWS CodeStar에서 프로젝트를 삭제하려면

1. https://console.aws.amazon.com/codestar/에서 AWS CodeStar 콘솔을 엽니다.
2. 탐색 창에서 프로젝트를 선택합니다.
3. 삭제할 프로젝트를 선택하고 삭제를 선택합니다.

   프로젝트를 열고 콘솔 왼쪽의 탐색 창에서 설정을 선택합니다. 프로젝트 세부 정보 페이지에서 [Delete project]를 선택합니다.
4. 삭제 확인 페이지에서 delete를 입력합니다. 프로젝트 리소스를 삭제하려면 리소스 삭제를 선택한 상태로 유지하십시오. 삭제를 선택합니다.

   프로젝트를 삭제하는 데 몇 분이 걸릴 수 있습니다. 삭제한 후에는 AWS CodeStar 콘솔의 프로젝트 목록에 해당 프로젝트가 더 이상 표시되지 않습니다.

> **Important**
>
> 프로젝트가 GitHub 리포지토리 또는 Atlassian JIRA의 문제 같은 AWS 외부의 리소스를 사용하는 경우, 해당 리소스는 확인란을 선택하더라도 삭제되지 않습니다.
> AWS CodeStar 관리형 정책이 IAM 사용자가 아닌 역할에 수동으로 연결돼 있다면 프로젝트를 삭제할 수 없습니다. 프로젝트의 관리형 정책을 연합된 사용자의 역할에 연결했다면, 정책을 분리해야 프로젝트를 삭제할 수 있습니다. 자세한 내용은 ??? 섹션을 참조하세요.

7단계: 생산 환경을 위한 프로젝트 준비

프로젝트 생성이 끝나면 코드를 생성, 테스트 및 배포할 수 있습니다. 생산 환경에서 프로젝트를 유지하는 데 필요한 다음 고려 사항을 검토하십시오.

- 애플리케이션이 사용하는 종속성에 대해 패치를 정기적으로 적용하고 보안 모범 사례를 검토하십시오. 자세한 내용은 AWS CodeStar 리소스에 대한 보안 모범 사례 섹션을 참조하세요.
- 프로젝트의 프로그래밍 언어가 제안하는 환경 설정을 정기적으로 모니터링하십시오.

다음 단계

AWS CodeStar에 대해 자세히 알아보는 데 도움이 되는 추가 리소스입니다.

- 자습서: AWS CodeStar에서 서버리스 프로젝트 생성 및 관리에서는 AWS Lambda의 로직을 사용하여 웹 서비스를 생성 및 배포하는 프로젝트 및 Amazon API Gateway의 API에서 호출할 수 있는 프로젝트를 사용합니다.
- AWS CodeStar 프로젝트 템플릿에서는 생성할 수 있는 다른 유형의 프로젝트를 설명합니다.
- AWS CodeStar 팀 작업 단원은 다른 사람들이 프로젝트 작업에 도움을 줄 수 있는 방법에 대한 정보를 제공합니다.

자습서: AWS CodeStar에서 서버리스 프로젝트 생성 및 관리

이 자습서는 AWS CodeStar를 사용하여 AWS Serverless Application Model(AWS SAM)을 사용하는 AWS Lambda에서 호스팅되는 웹 서비스의 AWS 리소스를 생성하고 관리하는 방법을 보여 줍니다.

AWS CodeStar는 AWS CloudFormation을 활용하는 AWS SAM을 사용하여 Amazon API Gateway API, AWS Lambda 함수 및 Amazon DynamoDB 테이블을 포함한, 지원되는 AWS 리소스를 생성하고 관리하는 단순한 방법을 제공합니다. (이 프로젝트는 Amazon DynamoDB 테이블을 사용하지 않습니다.)

SAM에 대한 자세한 내용은 GitHub의 AWS Serverless Application Model(AWS SAM)을 참조하십시오.

사전 조건: AWS CodeStar설정의 단계 완료.

> **Note**
>
> AWS 계정에는 AWS CodeStar에서 사용하는 AWS 서비스 비용을 포함한, 이 자습서와 관련된 비용이 청구될 수 있습니다. 자세한 내용은 AWS CodeStar 요금을 참조하세요.

주제

개요

이 자습서에서는 다음 작업을 하게 됩니다.

1. AWS CodeStar를 사용하여 AWS SAM을 통해 Python 기반 웹 서비스를 빌드 및 배포하는 프로젝트를 생성할 수 있습니다. 이 웹 서비스는 AWS Lambda에서 호스팅되며 Amazon API Gateway를 통해 액세스할 수 있습니다.

2. 프로젝트의 주요 리소스는 다음과 같습니다.
   - 프로젝트의 소스 코드가 저장된 AWS CodeCommit 리포지토리입니다. 이 소스 코드는 웹 서비스의 로직을 포함하고 관련 AWS 리소스를 정의합니다.
   - 소스 코드 빌드를 자동화하는 AWS CodePipeline 파이프라인입니다. 이 파이프라인은 AWS SAM을 사용하여 AWS Lambda에 함수를 생성 및 배포하고, Amazon API Gateway에 관련 API를 생성한 후 해당 함수에 API를 연결합니다.
   - AWS Lambda에 배포된 함수입니다.
   - Amazon API Gateway에서 생성되는 API입니다.
3. 웹 서비스를 테스트하여 AWS CodeStar가 예상대로 웹 서비스를 빌드하고 배포했는지 확인합니다.
4. 프로젝트의 소스 코드로 작업할 수 있도록 로컬 워크스테이션을 설정합니다.
5. 로컬 워크스테이션을 사용하여 프로젝트의 소스 코드를 변경합니다. 프로젝트에 함수를 추가한 다음 변경 내용을 소스 코드로 푸시하면 AWS CodeStar가 웹 서비스를 다시 빌드하고 배포합니다.
6. 웹 서비스를 다시 테스트하여 AWS CodeStar가 예상대로 웹 서비스를 다시 빌드하고 배포했는지 확인합니다.
7. 수동 테스트 중 일부를 자동 테스트로 바꾸려면 로컬 워크스테이션을 사용하여 단위 테스트를 작성합니다. 단위 테스트를 푸시하면 AWS CodeStar는 웹 서비스를 다시 빌드 및 배포하고 단위 테스트를 실행합니다.
8. 단위 테스트의 결과를 봅니다.
9. 프로젝트를 정리합니다. 이 단계를 이용하면 이 자습서와 관련된 비용을 AWS 계정에 청구되지 않게 할 수 있습니다.

1단계: 프로젝트 생성

이 단계에서는 AWS CodeStar 콘솔을 사용하여 프로젝트를 생성합니다.

1. AWS Management Console에 로그인하고 https://console.aws.amazon.com/codestar/에서 AWS CodeStar 콘솔을 열 수 있습니다.

   > **Note**
   >
   > AWS CodeStar설정에서 생성하거나 식별한 IAM 사용자와 연결된 자격 증명을 사용하여 AWS Management Console에 로그인해야 합니다. 이 사용자는 연결된 **AWSCodeStarFullAccess** 관리형 정책을 보유해야 합니다.

2. 프로젝트와 리소스를 생성하려는 AWS 리전을 선택합니다.

AWS CodeStar가 지원되는 AWS 리전에 대한 자세한 내용은 AWS 일반 참조의 리전 및 엔드포인트를 참조하세요.

3. 프로젝트 만들기를 선택합니다.
4. 프로젝트 템플릿 선택 페이지에서:
   - 애플리케이션 유형에서 웹 서비스를 선택합니다.
   - 프로그래밍 언어에서 Python을 선택합니다.
   - AWS 서비스에서 AWS Lambda를 선택합니다.
5. 선택 항목을 포함하는 상자를 선택합니다. 다음을 선택합니다.
6. 프로젝트 이름에 프로젝트의 이름(예: **`My SAM Project`**)을 입력합니다. 예제에서와 다른 이름을 사용한다면, 이 자습서 전체에서 해당 이름을 사용해야 합니다.

   프로젝트 ID에서 AWS CodeStar는 이 프로젝트에 대한 관련 식별자를 선택합니다(예: my-sam-project). 다른 프로젝트 ID를 사용하는 경우 이 자습서 전체에서 이를 사용해야 합니다.

   AWS CodeCommit를 선택한 상태로 두고 리포지토리 이름 값은 변경하지 않습니다.
7. 다음을 선택합니다.
8. 설정을 검토한 다음 프로젝트 생성을 선택합니다.

   이 AWS 리전에서 AWS CodeStar를 처음 사용한다면, 표시 이름 및 이메일에 AWS CodeStar가 IAM 사용자에 대해 사용하려는 표시 이름과 이메일 주소를 입력합니다. 다음을 선택합니다.
9. AWS CodeStar가 프로젝트를 생성하는 동안 기다립니다. 몇 분 정도 걸릴 수 있습니다. 새로 고칠 때 프로젝트 프로비저닝 배너가 표시될 때까지 계속하지 마세요.

2단계: 프로젝트 리소스 탐색

이 단계에서는 프로젝트의 AWS 리소스 네 가지를 탐색하여 프로젝트의 작동 방식을 이해합니다.

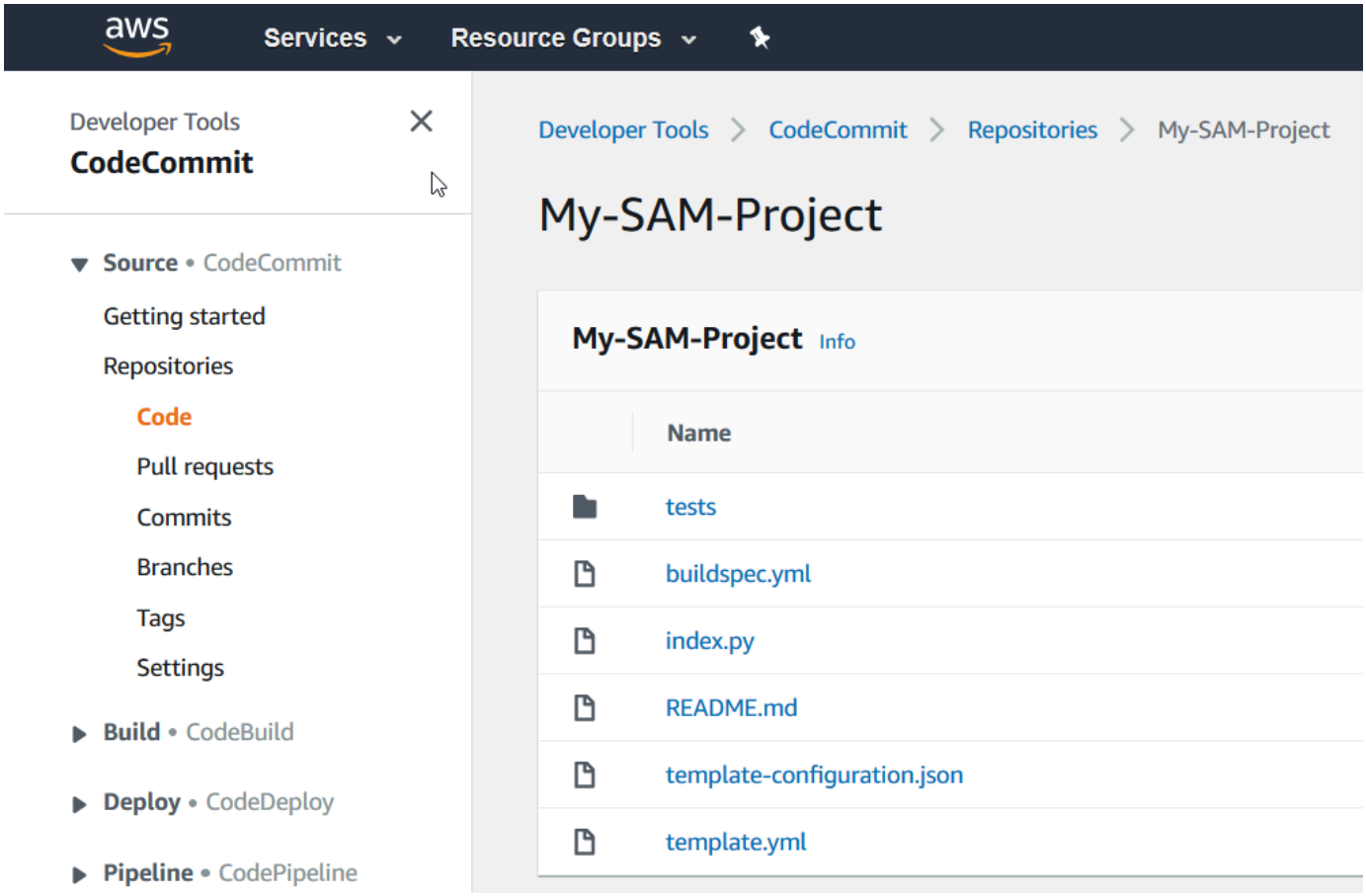
- 프로젝트의 소스 코드가 저장된 AWS CodeCommit 리포지토리입니다. AWS CodeStar는 리포지토리에 my-sam-project라는 이름을 부여합니다. 여기서 my-sam-project는 프로젝트의 이름입니다.
- CodeBuild와 AWS SAM을 이용해 웹 서비스의 Lambda 함수와 API Gateway의 API를 자동으로 빌드 및 배포하는 AWS CodePipeline 파이프라인입니다. AWS CodeStar는 파이프라인에 my-sam-project--Pipeline이라는 이름을 부여하며, 여기서 my-sam-project는 프로젝트의 ID입니다.
- 웹 서비스의 로직을 포함하는 Lambda 함수입니다. AWS CodeStar은 함수에 awscodestar-my-sam-project-lambda-HelloWorld-***`RANDOM_ID`***라는 이름을 부여하며, 여기서

  - my-sam-project는 프로젝트의 ID입니다.
  - HelloWorld는 AWS CodeCommit 리포지토리의 `template.yaml` 파일에 지정된 함수 ID입니다. 이 파일은 나중에 살펴봅니다.
  - *`RANDOM_ID`*는 고유성을 보장하기 위해 AWS SAM이 함수에 지정한 임의의 ID입니다.
- Lambda 함수를 더 쉽게 직접적으로 호출할 수 있게 해 주는 API Gateway의 API입니다. AWS CodeStar는 API에 awscodestar-my-sam-project--lambda라는 이름을 부여하며, 여기서 my-sam-project는 프로젝트의 ID입니다.

CodeCommit에서 소스 코드 리포지토리를 탐색하려면

1. AWS CodeStar 콘솔에서 프로젝트를 열어 둔 상태로 탐색 모음에서 리포지토리를 선택합니다.
2. 리포지토리 세부 정보에서 CodeCommit 리포지토리 링크(**`My-SAM-Project`**)를 선택합니다.
3. CodeCommit 콘솔의 코드 페이지에 프로젝트의 소스 코드 파일이 표시됩니다.

   - `buildspec.yml` - CodePipeline은 CodeBuild가 빌드 단계에서 AWS SAM을 사용하여 웹 서비스를 패키징하도록 지시합니다.
   - `index.py` - Lambda 함수에 대한 로직을 포함합니다. 이 함수는 ISO 형식의 문자열 `Hello World`와 타임스탬프를 출력합니다.
   - `README.md` - 리포지토리에 대한 일반 정보를 포함합니다.
   - `template-configuration.json` - 프로젝트 ID로 리소스에 태그를 지정하는 데 사용되는 자리 표시자와 함께 프로젝트 ARN을 포함합니다.
   - `template.yml` - AWS SAM이 웹 서비스를 패키징하고 API Gateway에서 API를 만드는 데 사용합니다.



파일의 콘텐츠를 보려면 목록에서 선택하십시오.

CodeCommit 콘솔 사용에 대한 자세한 내용은 AWS CodeCommit 사용 설명서를 참조하세요.

CodePipeline에서 파이프라인을 탐색하려면

1. 파이프라인에 대한 정보를 보려면 AWS CodeStar 콘솔에서 프로젝트를 열어 둔 상태로 측면 탐색 모음에서 파이프라인을 선택하면 파이프라인에 다음이 포함되어 있는 것을 볼 수 있습니다.
   - CodeCommit에서 소스 코드를 가져오기 위한 소스 단계.
   - CodeBuild로 소스 코드를 빌드하기 위한 빌드 단계.
   - AWS SAM을 사용하여 빌드된 소스 코드 및 AWS 리소스를 배포하기 위한 배포 단계.
2. 파이프라인에 대한 자세한 내용을 보려면 파이프라인 세부정보에서 파이프라인을 선택하여 CodePipeline 콘솔에서 파이프라인을 여십시오.

CodeCommit 콘솔 사용에 대한 자세한 내용은 AWS CodePipeline 사용 설명서를 참조하세요.

개요 페이지에서 프로젝트 활동 및 AWS 서비스 리소스를 탐색하려면

1. AWS CodeStar 콘솔에서 프로젝트를 열고 탐색 모음에서 개요를 선택합니다.
2. 프로젝트 활동 및 프로젝트 리소스 목록을 검토합니다.

Lambda에서 함수를 살펴보려면

1. AWS CodeStar 콘솔에서 프로젝트를 열어 둔 상태로 측면 탐색 모음에서 개요를 선택합니다.
2. 프로젝트 리소스의 ARN 열에서 Lambda 함수 링크를 선택합니다.

   Lambda 콘솔에 해당 함수의 코드가 표시됩니다.

Lambda 콘솔 사용에 대한 자세한 내용은 AWS Lambda 개발자 안내서를 참조하세요.

API Gateway에서 API를 탐색하려면

1. AWS CodeStar 콘솔에서 프로젝트를 열어 둔 상태로 측면 탐색 모음에서 개요를 선택합니다.
2. 프로젝트 리소스의 ARN 열에서 Amazon API Gateway API 링크를 선택합니다.

   API에 대한 리소스가 API Gateway 콘솔에 표시됩니다.

Amazon Gateway 콘솔에 대한 자세한 내용은 API Gateway 개발자 가이드를 참조하세요.

3단계: 웹 서비스 테스트

이 단계에서는 AWS CodeStar에서 방금 빌드하고 배포한 웹 서비스를 테스트합니다.

1. 이전 단계의 프로젝트를 열어 둔 상태로 측면 탐색 모음에서 파이프라인을 선택합니다.
2. 계속하기 전에 소스, 빌드, 배포 단계에 성공이 표시되는지 확인하세요. 몇 분 정도 걸릴 수 있습니다.

> **Note**
>
> 어느 단계에든 Failed가 표시되는 경우 문제 해결을 위해 다음 단계를 확인합니다.
>
> - 소스 단계에 대해서는 AWS CodeCommit사용 설명서의 AWS CodeCommit 문제 해결을 참조하십시오.

- 빌드 단계에 대해서는 AWS CodeBuild사용 설명서의 AWS CodeBuild 문제 해결을 참조하십시오.
- 배포 단계에 대해서는 AWS CloudFormation사용 설명서의 AWS CloudFormation 문제 해결을 참조하십시오.
- 그 밖의 문제는 문제 해결 AWS CodeStar 단원을 참조하십시오.

3. 애플리케이션 보기를 선택합니다.

웹 브라우저에서 열리는 새 탭에서 웹 서비스는 다음 응답 결과를 표시합니다.

```
{"output": "Hello World", "timestamp": "2017-08-30T15:53:42.682839"}
```

4단계: 프로젝트 코드를 편집하기 위한 로컬 워크스테이션 설정

이 단계에서는 로컬 워크스테이션을 설정하여 AWS CodeStar 프로젝트의 소스 코드를 편집합니다. 로컬 워크스테이션은 macOS, Windows 또는 Linux를 실행하는 실제 또는 가상 컴퓨터일 수 있습니다.

1. 이전 단계의 프로젝트를 계속 열어 둔 상태에서:
   - 탐색 모음에서 IDE를 선택한 다음 프로젝트 코드 액세스를 확장합니다.
   - 명령줄 인터페이스 아래에서 지침 보기를 선택합니다.

     Visual Studio 또는 Eclipse가 설치되어 있다면, 대신 Visual Studio 또는 Eclipse에서 지침 보기를 선택하고 다음 지침을 따른 다음 5단계: 웹 서비스에 로직 추가 단계로 건너뜁니다.
2. 지침을 따라 다음 작업을 완료합니다.
   a. 로컬 워크스테이션에 Git을 설정합니다.
   b. IAM 콘솔을 사용하여 IAM 사용자에 대한 Git 자격 증명을 생성합니다.
   c. 프로젝트의 CodeCommit 리포지토리를 로컬 워크스테이션에 복제합니다.
3. 왼쪽 탐색 창에서 프로젝트를 선택하여 프로젝트 개요로 돌아갑니다.

5단계: 웹 서비스에 로직 추가

이 단계에서는 로컬 워크스테이션을 사용하여 웹 서비스에 로직을 추가합니다. 특히 Lambda 함수를 추가한 다음 API Gateway의 API에 연결합니다.

1. 로컬 워크스테이션에서 복제된 소스 코드 리포지토리가 있는 디렉터리로 이동합니다.
2. 해당 디렉터리에서 `hello.py`라는 파일을 만듭니다. 다음 코드를 추가한 후 파일을 저장합니다.

```
import json

def handler(event, context):
  data = {
    'output': 'Hello ' + event["pathParameters"]["name"]
  }
  return {
    'statusCode': 200,
    'body': json.dumps(data),
    'headers': {'Content-Type': 'application/json'}
  }
```

   앞의 코드는 `Hello` 문자열과 호출자가 함수에 보내는 문자열을 출력합니다.

3. 동일한 디렉터리에서 `template.yml` 파일을 엽니다. 파일 끝에 다음 코드를 추가하고 파일을 저장합니다.

```
  Hello:
    Type: AWS::Serverless::Function
    Properties:
      FunctionName: !Sub 'awscodestar-${ProjectId}-lambda-Hello'
      Handler: hello.handler
      Runtime: python3.7
      Role:
        Fn::GetAtt:
        - LambdaExecutionRole
        - Arn
      Events:
        GetEvent:
          Type: Api
          Properties:
            Path: /hello/{name}
            Method: get
```

AWS SAM은 이 코드를 사용하여 Lambda에 함수를 만들고, API Gateway에서 API에 대한 새 메서드와 경로를 추가한 다음 해당 메서드와 경로를 새 함수에 연결합니다.

> **Note**
>
> 위에 나온 코드의 들여쓰기가 중요합니다. 코드를 표시된 대로 추가하지 않으면 프로젝트가 제대로 빌드되지 않을 수 있습니다.

4. git add .를 실행해 복제된 리포지토리의 스테이징 영역에 파일 변경 사항을 추가합니다. 변경된 모든 파일을 추가하는 점(.)을 잊지 마십시오.

> **Note**
>
> 명령줄 대신 Visual Studio 또는 Eclipse를 사용하는 경우 Git 사용 지침이 다를 수 있습니다. Visual Studio 또는 Eclipse 문서를 참조하십시오.

5. git commit -m "Added hello.py and updated template.yaml."를 실행하여 복제된 리포지토리의 스테이징된 파일을 커밋하십시오.
6. git push를 실행해 커밋을 원격 리포지토리에 푸시합니다.

> **Note**
>
> 이전에 생성한 로그인 자격 증명을 입력하라는 메시지가 표시될 수 있습니다. 원격 리포지토리와 상호 작용할 때마다 메시지가 표시되지 않도록 하려면 Git 자격 증명 관리자를 설치하고 구성하는 것이 좋습니다. 예를 들어 macOS 또는 Linux에서 터미널에 git config credential.helper 'cache --timeout 900'을 실행하여 매 15분이 지나자마자 메시지가 표시되게 할 수 있습니다. 또는 git config credential.helper 'store --file ~/.git-credentials'를 실행하여 다시 표시되지 않게 할 수도 있습니다. Git은 홈 디렉터리의 일반 파일에 일반 텍스트로 자격 증명을 저장합니다. 자세한 내용은 Git 웹사이트의 Git Tools - Credential Storage를 참조하십시오.

AWS CodeStar는 푸시를 감지한 후 CodePipeline이 CodeBuild 및 AWS SAM을 사용하여 웹 서비스를 다시 빌드 및 배포하도록 지시합니다. 파이프라인 페이지에서 배포 진행 상황을 확인할 수 있습니다.

AWS SAM은 새로운 함수에 awscodestar-my-sam-project-lambda-Hello-***RANDOM_ID***라는 이름을 부여합니다. 여기에서

- my-sam-project는 프로젝트의 ID입니다.
- Hello는 `template.yaml` 파일에 지정된 함수 ID입니다.
- *RANDOM_ID*는 고유성을 위해 AWS SAM이 함수에 지정한 임의의 ID입니다.

6단계: 고급 웹 서비스 테스트

이 단계에서는 이전 단계에서 추가한 로직에 따라 AWS CodeStar에서 빌드하고 배포한 고급 웹 서비스를 테스트합니다.

1. AWS CodeStar 콘솔에서 프로젝트를 열어 둔 상태로 탐색 모음에서 프로젝트를 선택합니다.
2. 파이프라인이 다시 실행되었는지 확인하고 계속하기 전에 소스, 빌드, 배포 단계에 성공이 표시되는지 확인하세요. 몇 분 정도 걸릴 수 있습니다.

   > **Note**
   >
   > 어느 단계에든 Failed가 표시되는 경우 문제 해결을 위해 다음 단계를 확인합니다.
   >
   > - 소스 단계에 대해서는 AWS CodeCommit사용 설명서의 AWS CodeCommit 문제 해결을 참조하십시오.
   > - 빌드 단계에 대해서는 AWS CodeBuild사용 설명서의 AWS CodeBuild 문제 해결을 참조하십시오.
   > - 배포 단계에 대해서는 AWS CloudFormation사용 설명서의 AWS CloudFormation 문제 해결을 참조하십시오.
   > - 그 밖의 문제는 문제 해결 AWS CodeStar 단원을 참조하십시오.

3. 애플리케이션 보기를 선택합니다.

   웹 브라우저에서 열리는 새 탭에서 웹 서비스는 다음 응답 결과를 표시합니다.

   ```
   {"output": "Hello World", "timestamp": "2017-08-30T15:53:42.682839"}
   ```

4. 탭의 주소 표시줄에서 경로 **/hello/** 및 이름을 URL(예: https://***API_ID***.execute-api.***REGION_ID***.amazonaws.com/Prod/hello/***YOUR_FIRST_NAME***)의 끝에 추가한 다음 Enter를 누릅니다.

이름이 Mary라면 웹 서비스는 다음 응답 결과를 표시합니다.

```
{"output": "Hello Mary"}
```

7단계: 웹 서비스에 단위 테스트 추가

이 단계에서는 로컬 워크스테이션을 사용하여 AWS CodeStar가 웹 서비스에서 실행되는 테스트를 추가합니다. 이 테스트는 앞서 수행한 수동 테스트를 대체합니다.

1. 로컬 워크스테이션에서 복제된 소스 코드 리포지토리가 있는 디렉터리로 이동합니다.
2. 해당 디렉터리에서 `hello_test.py`라는 파일을 만듭니다. 다음 코드를 추가한 후 파일을 저장합니다.

   ```
   from hello import handler

   def test_hello_handler():

     event = {
       'pathParameters': {
         'name': 'testname'
       }
     }

     context = {}

     expected = {
       'body': '{"output": "Hello testname"}',
       'headers': {
         'Content-Type': 'application/json'
       },
       'statusCode': 200
     }

     assert handler(event, context) == expected
   ```

   이 테스트는 Lambda 함수의 출력이 예상된 형식인지 여부를 확인합니다. 예상된 형식인 경우, 테스트가 성공한 것입니다. 그렇지 않으면 테스트가 실패한 것입니다.
3. 동일한 디렉터리에서 `buildspec.yml` 파일을 엽니다. 파일 콘텐츠를 다음 코드로 바꾼 다음 파일을 저장합니다.

```
version: 0.2

phases:
   install:
      runtime-versions:
         python: 3.7

      commands:
         - pip install pytest
         # Upgrade AWS CLI to the latest version
         - pip install --upgrade awscli


   pre_build:
      commands:
         - pytest

   build:
      commands:
         # Use AWS SAM to package the application by using AWS CloudFormation
         - aws cloudformation package --template template.yml --s3-bucket
 $S3_BUCKET --output-template template-export.yml

         # Do not remove this statement. This command is required for AWS CodeStar
 projects.
         # Update the AWS Partition, AWS Region, account ID and project ID in the
 project ARN on template-configuration.json file so AWS CloudFormation can tag
 project resources.
         - sed -i.bak 's/\$PARTITION\$/'${PARTITION}'/g;s/\$AWS_REGION
\$/'${AWS_REGION}'/g;s/\$ACCOUNT_ID\$/'${ACCOUNT_ID}'/g;s/\$PROJECT_ID\
$/'${PROJECT_ID}'/g' template-configuration.json

artifacts:
   type: zip
   files:
      - template-export.yml
      - template-configuration.json
```

이 빌드 사양은 CodeBuild가 Python 테스트 프레임워크인 pytest를 빌드 환경에 설치하도록 지시합니다. CodeBuild는 pytest를 사용하여 단위 테스트를 실행합니다. 나머지 빌드 사양은 이전과 동일합니다.

4. Git을 사용하여 이러한 변경 사항을 원격 리포지토리에 푸시합니다.

```
git add .

git commit -m "Added hello_test.py and updated buildspec.yml."

git push
```

8단계: 단위 테스트 결과 보기

이 단계에서는 단위 테스트의 성공 및 실패 여부를 확인할 수 있습니다.

1. AWS CodeStar 콘솔에서 프로젝트를 열어 둔 상태로 탐색 모음에서 파이프라인을 선택합니다.
2. 계속하기 전에 파이프라인이 다시 실행되었는지 확인하세요. 몇 분 정도 걸릴 수 있습니다.

   유닛 테스트가 성공하면 빌드 단계에 성공이 표시됩니다.
3. 단위 테스트 결과 세부 정보를 보려면 빌드 단계에서 CodeBuild 링크를 선택합니다.
4. CodeBuild 콘솔에서 Build Project: my-sam-project 페이지의 빌드 기록에서 테이블의 빌드 실행 열에 있는 링크를 선택합니다.
5. my-sam-project:***BUILD_ID*** 페이지의 빌드 로그에서 전체 로그 보기 링크를 선택합니다.
6. Amazon CloudWatch Logs 콘솔의 로그 출력에서 다음과 유사한 테스트 결과를 찾습니다. 다음 테스트 결과에서 테스트가 통과되었습니다.

```
...
============================= test session starts ==============================
platform linux2 -- Python 2.7.12, pytest-3.2.1, py-1.4.34, pluggy-0.4.0
rootdir: /codebuild/output/src123456789/src, inifile:
collected 1 item

hello_test.py .

========================== 1 passed in 0.01 seconds ===========================
...
```

   테스트가 실패한 경우, 로그 출력에 오류 해결에 도움이 되는 로그 출력에 세부 정보가 있어야 합니다.

9단계: 정리

이 단계에서는 이 프로젝트에 대한 요금이 청구되지 않도록 프로젝트를 정리합니다.

이 프로젝트를 계속 사용하려면 이 단계를 건너뛰어도 되지만, AWS 계정에 요금이 계속 청구될 수 있습니다.

1. AWS CodeStar 콘솔에서 프로젝트를 열어 둔 상태로 탐색 모음에서 설정을 선택합니다.
2. 프로젝트 세부 정보 페이지에서 프로젝트 삭제를 선택합니다.
3. **`delete`**를 입력하고 리소스 삭제 확인란을 선택한 상태로 유지한 다음 삭제를 선택합니다.

> **Important**
>
> 이 확인란의 선택을 취소하면 프로젝트 기록이 AWS CodeStar에서 삭제되지만 프로젝트의 AWS 리소스 대부분은 그대로 보존됩니다. AWS 계정에 요금이 계속 청구될 수 있습니다.

AWS CodeStar가 이 프로젝트에 대해 생성한 Amazon S3 버킷이 있다면 다음 단계를 밟아 삭제하십시오.

1. https://console.aws.amazon.com/s3/에서 Amazon S3 콘솔을 엽니다.
2. 버킷 목록에서 aws-codestar-***REGION_ID*-*ACCOUNT_ID***-my-sam-project--pipe 옆의 아이콘을 선택합니다. 여기에서
   - *REGION_ID*는 방금 삭제한 프로젝트에 대한 AWS 리전의 ID입니다.
   - *ACCOUNT_ID*는 AWS 계정 ID입니다.
   - my-sam-project는 방금 삭제한 프로젝트의 ID입니다.
3. 버킷 비우기를 선택합니다. 버킷의 이름을 입력한 다음 확인을 선택합니다.
4. 버킷 삭제를 선택합니다. 버킷의 이름을 입력한 다음 확인을 선택합니다.

다음 단계

이제 이 자습서를 완료했으므로 다음 리소스를 검토하는 것이 좋습니다.

- AWS CodeStar 시작하기 자습서에서는 Amazon EC2 인스턴스에서 실행되는 Node.js 기반 웹 애플리케이션을 생성하고 배포하는 프로젝트를 사용합니다.

- AWS CodeStar 프로젝트 템플릿에서는 생성할 수 있는 다른 유형의 프로젝트를 설명합니다.
- AWS CodeStar 팀 작업에서는 다른 사람들이 프로젝트에서 어떤 도움이 될 수 있는지 보여 줍니다.

자습서: AWS CLI를 이용하여 AWS CodeStar에서 프로젝트 생성

이 자습서는 AWS CLI를 이용해 샘플 소스 코드와 샘플 도구 체인으로 AWS CodeStar 프로젝트를 만드는 방법을 설명합니다. AWS CodeStar는 AWS CloudFormation 도구 체인 템플릿에서 지정하는 AWS 인프라와 IAM 리소스를 제공합니다. 프로젝트는 사용자의 도구 체인 리소스를 관리해 소스 코드를 빌드하고 배포합니다.

AWS CodeStar는 AWS CloudFormation을 이용해 샘플 코드를 빌드하고 배포합니다. 이 샘플 코드는 AWS Lambda에서 호스트하는 웹 서비스를 생성하며 Amazon API Gateway를 통해 액세스할 수 있습니다.

사전 조건:

- AWS CodeStar설정의 단계를 수행합니다.
- 생성된 Amazon S3 스토리지 버킷이 있어야 합니다. 이번 자습서에서는 샘플 소스 코드와 도구 체인 템플릿을 이 위치에 업로드합니다.

> **Note**
>
> AWS 계정에는 AWS CodeStar에서 사용하는 AWS 서비스 비용을 포함한, 이 자습서와 관련된 비용이 청구될 수 있습니다. 자세한 내용은 AWS CodeStar 요금을 참조하세요.

주제

1 단계: 샘플 소스 코드 다운로드 및 검토

이번 자습서에서는 zip 파일을 다운로드할 수 있습니다. 여기에는 Lambda 컴퓨팅 플랫폼의 Node.js 샘플 애플리케이션을 위한 샘플 소스 코드가 포함되어 있습니다. 소스 코드를 리포지토리에 배치하면, 폴더와 파일이 다음과 같이 표시됩니다.

```
tests/
app.js
buildspec.yml
index.js
package.json
README.md
template.yml
```

다음 프로젝트 요소가 샘플 소스 코드에 표시됩니다:

- `tests/`: 이 프로젝트의 CodeBuild 프로젝트에 대한 단위 테스트 설정입니다. 이 폴더는 샘플 코드에 포함되지만, 반드시 프로젝트를 생성할 필요는 없습니다.
- `app.js`: 프로젝트에 대한 애플리케이션 소스 코드입니다.
- `buildspec.yml`: CodeBuild 리소스의 빌드 단계에 대한 빌드 지침입니다. 이 파일은 CodeBuild 리소스가 있는 도구 체인 템플릿에 필요합니다.
- `package.json`: 애플리케이션 소스 코드의 종속성 정보입니다.
- `README.md`: 모든 AWS CodeStar 프로젝트에 포함되는 프로젝트 readme 파일입니다. 이 파일은 샘플 코드에 포함되지만, 반드시 프로젝트를 생성할 필요는 없습니다.
- `template.yml`: 모든 AWS CodeStar 프로젝트에 포함되는 인프라 템플릿 파일 또는 SAM 템플릿 파일입니다. 이번 자습서 후반에 업로드하는 도구 체인 템플릿인 template.yml과는 다른 파일입니다. 이 파일은 샘플 코드에 포함되지만, 반드시 프로젝트를 생성할 필요는 없습니다.

단계 2: 샘플 도구 체인 템플릿 다운로드

이 자습서에 제공된 샘플 도구 체인 템플릿은 리포지토리(CodeCommit), 파이프라인(CodePipeline), 빌드 컨테이너(CodeBuild)를 생성하며 AWS CloudFormation을 이용해 소스 코드를 Lambda 플랫폼에 배포합니다. 이러한 리소스 외에도, 실행 시간 환경 권한의 범위를 결정하는 데 사용하는 IAM 역할, CodePipeline이 배포 아티팩트를 저장하는 데 사용하는 Amazon S3 버킷, 코드를 리포지토리로 푸시할 때 파이프라인 배포를 트리거하는 데 사용하는 CloudWatch Events 규칙 등이 존재합니다. AWS IAM 모범 사례와 부합하도록, 이번 예제에서 정의된 도구 체인 역할의 정책 규모를 축소하십시오.

샘플 AWS CloudFormation 템플릿을 YAML 형식으로 다운로드하고 압축을 풉니다.

이번 자습서 후반부에서 create-project명령어를 실행하면, 이 템플릿은 AWS CloudFormation에 다음과 같은 사용자 지정 도구 체인 리소스를 생성합니다. 이 자습서에서 생성한 리소스에 대한 자세한 내용은 AWS CloudFormation 사용 설명서에 있는 다음 주제를 참조하십시오.

- AWS::CodeCommit::Repository AWS CloudFormation 리소스는 CodeCommit 리포지토리를 생성합니다.
- AWS::CodeBuild::Project AWS CloudFormation 리소스는 CodeBuild 빌드 프로젝트를 생성합니다.
- AWS::CodeDeploy::Application AWS CloudFormation 리소스는 CodeDeploy 애플리케이션을 생성합니다.
- AWS::CodePipeline::Pipeline AWS CloudFormation 리소스는 CodePipeline 파이프라인을 생성합니다.
- AWS::S3::Bucket AWS CloudFormation 리소스는 파이프라인의 아티팩트 버킷을 생성합니다.
- AWS::S3::BucketPolicy AWS CloudFormation 리소스는 파이프라인의 아티팩트 버킷을 위한 아티팩트 버킷 정책을 생성합니다.
- AWS::IAM::Role AWS CloudFormation 리소스는 CodeBuild 빌드 프로젝트 관리를 위한 AWS CodeStar 권한을 제공하는 CodeBuild IAM 작업자 역할을 생성합니다.
- AWS::IAM::Role AWS CloudFormation 리소스는 파이프라인 생성을 위한 AWS CodeStar 권한을 제공하는 CodePipeline IAM 작업자 역할을 생성합니다.
- AWS::IAM::Role AWS CloudFormation 리소스는 리소스 스택 생성을 위한 AWS CodeStar 권한을 제공하는 AWS CloudFormation IAM 작업자 역할을 생성합니다.
- AWS::IAM::Role AWS CloudFormation 리소스는 리소스 스택 생성을 위한 AWS CodeStar 권한을 제공하는 AWS CloudFormation IAM 작업자 역할을 생성합니다.
- AWS::IAM::Role AWS CloudFormation 리소스는 리소스 스택 생성을 위한 AWS CodeStar 권한을 제공하는 AWS CloudFormation IAM 작업자 역할을 생성합니다.
- AWS::Events::Rule AWS CloudFormation 리소스는 리포지토리의 푸시 이벤트를 모니터링하는 CloudWatch 규칙을 생성합니다.
- AWS::IAM::Role AWS CloudFormation 리소스는 CloudWatch Events IAM Role 리소스를 생성합니다.

단계 3: AWS CloudFormation의 도구 체인 템플릿 테스트

도구 체인 템플릿을 업로드하기 전에, AWS CloudFormation의 도구 체인 템플릿을 테스트하고 오류를 수정할 수 있습니다.

1. 업데이트된 템플릿을 로컬 컴퓨터에 저장하고 AWS CloudFormation 콘솔을 엽니다. 스택 생성을 선택합니다. 목록에 새로운 리소스가 표시됩니다.
2. 스택을 보면서 스택 성성 오류가 있는지 확인합니다.
3. 테스트가 끝나면, 스택을 삭제하십시오.

> **Note**
>
> 스택과 AWS CloudFormation에서 생성한 리소스를 모두 삭제했는지 확인하십시오. 모두 삭제하지 않으면, 프로젝트 생성 시 이미 사용 중인 리소스 이름 때문에 오류가 발생할 수 있습니다.

단계 4: 소스 코드 및 도구 체인 템플릿 업로드

AWS CodeStar 프로젝트를 생성하려면, 소스 코드를 .zip 파일로 압축한 다음 Amazon S3로 옮겨야 합니다. AWS CodeStar는 이러한 콘텐츠로 리포지토리를 초기화합니다. 이 위치는 AWS CLI에서 프로젝트를 생성하는 명령을 실행할 때 입력 파일에서 지정합니다.

`toolchain.yml` 파일도 업로드하고 Amazon S3로 옮겨야 합니다. 이 위치는 AWS CLI에서 프로젝트를 생성하는 명령을 실행할 때 입력 파일에서 지정합니다.

소스 코드 및 도구 체인 템플릿을 업로드하려면

1. 다음 샘플 파일 구조는 압축 및 업로드할 준비가 끝난 소스 파일과 도구 체인 템플릿을 보여줍니다. 샘플 코드에는 `template.yml` 파일이 포함됩니다. 명심하십시오. 이 파일은 `toolchain.yml` 파일과는 다른 파일입니다.

```
ls
src toolchain.yml

ls src/
README.md    app.js        buildspec.yml    index.js    package.json
 template.yml    tests
```

2. 소스 코드 파일의 .zip 파일을 생성합니다.

```
cd src; zip -r "../src.zip" *; cd ../
```

3. cp 명령을 사용하고 파일을 파라미터로 포함합니다.

다음 명령어는 .zip 파일과 `toolchain.yml`을 Amazon S3로 업로드합니다.

```
aws s3 cp src.zip s3://MyBucket/src.zip
aws s3 cp toolchain.yml s3://MyBucket/toolchain.yml
```

소스 코드 공유를 위해 Amazon S3 버킷을 업로드하려면

- 소스 코드와 도구 체인을 Amazon S3에 저장하므로, Amazon S3 버킷 정책과 객체 ACL을 이용해 다른 IAM 사용자나 AWS 계정이 샘플에서 프로젝트를 생성하게 할 수 있습니다. AWS CodeStar를 이용하면 사용자 지정 프로젝트를 생성한 모든 사용자는 사용하고 싶은 도구 체인과 소스에 액세스할 수 있습니다.

  다른 사람이 샘플을 사용하게 하려면, 다음 명령을 실행하십시오.

  ```
  aws s3api put-object-acl --bucket MyBucket --key toolchain.yml --acl public-read
  aws s3api put-object-acl --bucket MyBucket --key src.zip --acl public-read
  ```

5단계: AWS CodeStar에서 프로젝트 만들기

이상의 단계를 따라 프로젝트를 생성하십시오.

> **Important**
>
> AWS CLI에서 원하는 AWS 리전을 구성했는지 확인하십시오. 프로젝트는 AWS CLI에서 구성한 AWS 리전에 생성됩니다.

1. create-project 명령을 실행하고 `--generate-cli-skeleton` 파라미터를 포함하십시오.

   ```
   aws codestar create-project --generate-cli-skeleton
   ```

   JSON 형식 데이터가 출력에 표시됩니다. AWS CLI가 설치된 로컬 컴퓨터 또는 인스턴스의 위치에 있는 파일(예: *`input.json`*)에 데이터를 복사합니다. 복사된 데이터를 다음과 같이 수정하고 결과를 저장합니다. 이 입력 파일은 버킷 이름이 `myBucket`인 `MyProject`라는 프로젝트에 대해 구성되었습니다.

- `roleArn` 파라미터를 입력했는지 확인하십시오. 이 자습서의 샘플 템플릿과 같은 사용자 지정 템플릿의 경우 역할을 입력해야 합니다. 이 역할은 단계 2: 샘플 도구 체인 템플릿 다운로드에 지정된 모든 리소스를 생성할 수 있는 권한이 있어야 합니다.
- `stackParameters`에 `ProjectId`를 입력했는지 확인하십시오. 이 자습서에서 제공하는 샘플 템플릿은 이 파라미터를 요구합니다.

```
{
    "name": "MyProject",
    "id": "myproject",
    "description": "Sample project created with the CLI",
    "sourceCode": [
        {
            "source": {
                "s3": {
                    "bucketName": "MyBucket",
                    "bucketKey": "src.zip"
                }
            },
            "destination": {
                "codeCommit": {
                    "name": "myproject"

                }
            }
        }
    ],
    "toolchain": {
        "source": {
            "s3": {
                "bucketName": "MyBucket",
                "bucketKey": "toolchain.yml"
            }
        },
        "roleArn": "role_ARN",
        "stackParameters": {
            "ProjectId": "myproject"
        }
    }
}
```

2. 방금 저장한 파일이 들어 있는 디렉터리로 전환한 다음, create-project 명령을 다시 실행합니다. `--cli-input-json` 파라미터를 포함합니다.

```
aws codestar create-project --cli-input-json file://input.json
```

3. 이 명령이 제대로 실행되면 다음과 비슷한 데이터가 출력에 표시됩니다.

```
{
    "id": "project-ID",
    "arn": "arn"
}
```

- 출력에는 새 프로젝트에 대한 정보가 들어 있습니다.
  - `id` 값은 프로젝트 ID를 나타냅니다.
  - `arn` 값은 프로젝트의 ARN을 나타냅니다.

4. describe-project 명령을 이용해 프로젝트 생성 상태를 확인하십시오. `--id` 파라미터를 포함합니다.

```
aws codestar describe-project --id <project_ID>
```

다음과 비슷한 데이터가 결과에 나타납니다.

```
{
    "name": "MyProject",
    "id": "myproject",
    "arn": "arn:aws:codestar:us-east-1:account_ID:project/myproject",
    "description": "",
    "createdTimeStamp": 1539700079.472,
    "stackId": "arn:aws:cloudformation:us-east-1:account_ID:stack/awscodestar-
myproject/stack-ID",
    "status": {
        "state": "CreateInProgress"
    }
}
```

- 출력에는 새 프로젝트에 대한 정보가 들어 있습니다.
  - `id` 값은 고유 프로젝트 ID를 나타냅니다.

- `state` 값은 프로젝트 생성 상태(`CreateInProgress` 또는 `CreateComplete` 등)를 나타냅니다.

프로젝트를 만드는 중에 팀원을 추가하거나 명령줄 또는 선호하는 IDE에서 프로젝트 리포지토리에 대한 액세스를 구성할 수 있습니다.

자습서: AWS CodeStar에 Alexa Skill 프로젝트 생성

AWS CodeStar는 AWS에서 애플리케이션을 신속하게 개발, 구축 및 배포하는 데 필요한 도구를 제공하는 AWS의 클라우드 기반 개발 서비스입니다. AWS CodeStar에서는 몇 분 만에 전체 지속적 전달 도구 체인을 구성할 수 있으므로 코드 릴리스를 더욱 빠르게 시작할 수 있습니다. AWS CodeStar의 Alexa 스킬 프로젝트 템플릿을 사용하면 단 몇 번의 클릭만으로 AWS 계정에서 간단한 Hello World Alexa 스킬을 생성할 수 있습니다. 템플릿은 또한 스킬 개발을 위해 지속적 통합(CI) 워크플로를 시작하는 기본 배포 파이프라인을 생성합니다.

AWS CodeStar에서 Alexa 스킬을 생성할 때의 주요 이점은 AWS에서 스킬 개발을 시작하고 Amazon 개발자 계정을 프로젝트에 연결하여 AWS에서 직접 개발 단계에 스킬을 배포할 수 있다는 것입니다. 또한 프로젝트의 모든 소스 코드가 있는 리포지토리로 배포(CI) 파이프라인을 사용할 준비가 된 것입니다. 선호하는 IDE로 이 리포지토리를 구성하여 익숙한 도구로 스킬을 생성할 수 있습니다.

필수 조건

- https://developer.amazon.com로 이동하여 Amazon 개발자 계정을 생성하십시오. 가입은 무료입니다. 이 계정은 Alexa skill을 소유합니다.
- AWS 계정이 없는 경우 다음 절차에 따라 하나를 생성하십시오.

  AWS에 가입하려면

  1. https://aws.amazon.com/을 연 다음 AWS 계정 생성을 선택합니다.

     > **Note**
     >
     > 전에 AWS 계정 루트 사용자 자격 증명을 사용하여 AWS Management Console 에 로그인한 적이 있는 경우 Sign in to a different account(다른 계정으로 로그인)를 선택합니다. 이전에 IAM 자격 증명을 사용하여 콘솔에 로그인한 적이 있는 경우 Sign-in using AWS 계정 루트 사용자 credentials(루트 계정 자격 증명으로 로그인)를 선택합니다. 그런 다음 Create a new AWS account(새 AWS 계정 생성)을 선택합니다.

2. 온라인 지시 사항을 따릅니다.

> **Important**
>
> Alexa 스킬 프로젝트를 생성한 후에는 프로젝트 리포지토리에서만 모든 수정 작업을 수행합니다. ASK CLI 또는 ASK 개발자 콘솔과 같은 다른 Alexa Skill 키트 도구를 사용하여 이 스킬을 직접 편집하지 않는 것이 좋습니다. 이러한 도구는 프로젝트 리포지토리와 통합되지 않습니다. 이를 사용하면 스킬과 리포지토리 코드가 동기화되지 않습니다.

1단계: 프로젝트 생성 및 Amazon 개발자 계정 연결

이 자습서에서는 AWS Lambda에서 실행되는 Node.js를 사용하여 스킬을 생성합니다. 스킬 이름이 다를지라도 대부분의 단계는 다른 언어에서도 동일합니다. 선택한 특정 프로젝트 템플릿에 대한 세부 정보는 프로젝트 리포지토리의 README.md 파일을 참조하십시오.

1. 그런 다음 AWS Management Console에 로그인하고 https://console.aws.amazon.com/glue/에서 AWS CodeStar 콘솔을 열 수 있습니다.
2. 프로젝트와 리소스를 생성하려는 AWS 리전을 선택합니다. Alexa skill 실행 시간은 다음 AWS 리전에서만 사용 가능합니다.
   - 아시아 태평양(도쿄)
   - EU(아일랜드)
   - 미국 동부(버지니아 북부)
   - 미국 서부(오레건)
3. 프로젝트 만들기를 선택합니다.
4. 프로젝트 템플릿 선택 페이지에서:
   a. 애플리케이션 유형에서 Alexa Skill을 선택합니다.
   b. 프로그래밍 언어에서 Node.js를 선택합니다.
5. 선택 항목을 포함하는 상자를 선택합니다.
6. 프로젝트 이름에 프로젝트의 이름(예: **`My Alexa Skill`**)을 입력합니다. 다른 이름을 사용하는 경우 이 자습서에 이 이름을 사용해야 합니다. AWS CodeStar는 프로젝트 ID(예: my-alexa-skill)의 이 프로젝트에 관련 식별자를 선택합니다. 다른 프로젝트 ID를 사용하는 경우 이 자습서 전체에서 이를 사용해야 합니다.

7. 이 자습서의 리포지토리에 대해 AWS CodeCommit을 선택하고 리포지토리 이름 값을 변경하지 않습니다.
8. Connect Amazon developer account(Amazon 개발자 계정 연결)을 선택하여 스킬 호스팅을 위해 Amazon 개발자 계정에 연결합니다. Amazon 개발자 계정이 없는 경우 먼저 계정을 만들고 Amazon Developers에서 등록을 완료하십시오.
9. Amazon 개발자 자격 증명으로 로그인합니다. 허용을 선택한 다음 확인을 선택하여 연결을 완료합니다.
10. Amazon 개발자 계정과 연결된 공급업체 ID가 여러 개인 경우 이 프로젝트에 사용할 공급업체 ID를 선택합니다. 관리자 또는 개발자 역할이 지정된 계정을 사용하는지 확인합니다.
11. 다음을 선택합니다.
12. (선택 사항) 이 AWS 리전에서 AWS CodeStar를 처음 사용하는 경우 AWS CodeStar에서 사용하려는 표시 이름과 이메일 주소를 IAM 사용자용으로 입력합니다. 다음을 선택합니다.
13. AWS CodeStar가 프로젝트를 생성하는 동안 기다립니다. 몇 분 정도 걸릴 수 있습니다. 프로젝트 프로비저닝 배너가 표시될 때까지 계속하지 마세요.

2단계: Alexa Simulator에서 스킬 테스트

첫 번째 단계에서는 AWS CodeStar이 사용자를 위한 스킬을 생성하여 Alexa skill 개발 단계에 배포했습니다. 다음으로 Alexa Simulator에서 스킬을 테스트합니다.

1. AWS CodeStar 콘솔의 프로젝트에서 애플리케이션 보기를 선택합니다. Alexa Simulator에서 새 탭이 열립니다.
2. 1단계에서 프로젝트에 연결한 계정의 Amazon 개발자 자격 증명으로 로그인합니다.
3. 테스트 아래에서 개발을 선택하여 테스트를 활성화합니다.
4. `ask hello node hello`을 입력합니다. 스킬의 기본 간접 호출 이름은 `hello node`입니다.
5. 스킬이 `Hello World!`에 응답해야 합니다.

Alexa Simulator에서 스킬이 활성화되면 Amazon 개발자 계정에 등록된 Alexa 활성화 디바이스에서 스킬을 간접 호출할 수도 있습니다. 디바이스에 대한 스킬을 테스트하려면 Alexa, hello node에 인사하라고 요청하세요라고 말합니다.

Alexa Simulator에 대한 자세한 내용은 개발자 콘솔에서 스킬 테스트를 참조하십시오.

3단계: 프로젝트 리소스 탐색

프로젝트 생성의 일환으로 AWS CodeStar는 사용자를 대신하여 AWS 리소스를 생성했습니다. 이러한 리소스에는 CodeCommit을 사용하는 프로젝트 리포지토리, CodePipeline을 사용하는 배포 파이프라인 및 AWS Lambda 함수가 포함됩니다. 탐색 모음에서 이러한 리소스에 액세스할 수 있습니다. 예를 들어 리포지토리를 선택하면 CodeCommit 리포지토리에 대한 세부 정보가 표시됩니다. 파이프라인 페이지에서 파이프라인 배포 상태를 볼 수 있습니다. 탐색 모음에서 개요를 선택하여 프로젝트의 일부로 생성된 AWS 리소스의 전체 목록을 볼 수 있습니다. 이 목록에는 각 리소스의 링크가 포함되어 있습니다.

4단계: 스킬 응답 변경

이 단계에서는 반복 주기를 이해하기 위해 스킬의 응답을 약간 변경합니다.

1. 탐색 모음에서 리포지토리를 선택합니다. 리포지토리 이름 아래의 링크를 선택하면 프로젝트 리포지토리가 새 탭 또는 창에 열립니다. 이 리포지토리에는 빌드 사양 (buildspec.yml), AWS CloudFormation 애플리케이션 스택 (template.yml), readme 파일 및 스킬 패키지 형식 (프로젝트 구조)의 스킬 소스 코드가 포함되어 있습니다.
2. lambda > custom > index.js 파일(Node.js.의 경우)로 이동합니다. 이 파일에는 ASK SDK를 사용하는 요청 처리 코드가 들어 있습니다.
3. 편집을 선택합니다.
4. 24행의 문자열 `Hello World!`을 문자열 `Hello. How are you?`로 바꿉니다.
5. 파일의 끝까지 스크롤을 내립니다. 작성자 이름과 이메일 주소 및 커밋 메시지(선택 사항)를 입력합니다.
6. 변경 사항 커밋을 선택하여 변경 내용을 리포지토리에 커밋합니다.
7. AWS CodeStar의 프로젝트로 돌아가서 파이프라인 페이지를 확인하세요. 이제 파이프라인 배포가 표시되어야 합니다.
8. 파이프라인 배포가 완료되면 Alexa Simulator에서 다시 스킬을 테스트합니다. 이제 스킬이`Hello. How are you?`로 응답해야 합니다

5단계: 프로젝트 리포지토리에 연결하기 위해 로컬 워크스테이션 설정

이전에는 CodeCommit 콘솔에서 직접 소스 코드를 약간 변경했습니다. 이 단계에서는 로컬 워크스테이션으로 프로젝트 리포지토리를 구성하여 명령줄이나 선호하는 IDE에서 코드를 편집하고 관리할 수 있습니다. 다음 단계에서는 명령줄 도구를 설정하는 방법을 설명합니다.

1. 필요한 경우 AWS CodeStar의 프로젝트 대시보드로 이동합니다.
2. 탐색 창에서 IDE를 선택합니다.
3. 프로젝트 코드 액세스에서 명령줄 인터페이스 아래의 지침을 확인하세요.
4. 지침을 따라 다음 작업을 완료합니다.
   a. Git 다운로드와 같은 웹 사이트에서 로컬 워크스테이션에 Git을 설치합니다.
   b. AWS CLI를 설치합니다. 자세한 내용은 AWS 명령줄 인터페이스 설치 단원을 참조하세요.
   c. IAM 사용자 액세스 키 및 비밀 키로 AWS CLI를 구성합니다. 자세한 내용은 AWS CLI 구성을 참조하세요.
   d. 프로젝트의 CodeCommit 리포지토리를 로컬 워크스테이션에 복제합니다. 자세한 내용은 CodeCommit 리포지토리에 연결을 참조하세요.

다음 단계

이 자습서에서는 기본 스킬을 시작하는 방법을 보여 주었습니다. 스킬 개발 과정을 계속 진행하려면 다음 리소스를 참조하십시오.

- Alexa 개발자 YouTube 채널에서 Alexa Skills 작업 및 기타 동영상을 시청하여 스킬의 기본 사항을 이해하십시오.
- 스킬 패키지 형식, 스킬 매니페스트 스키마 및 상호 작용 모델 스키마에 대한 설명서를 검토하여 스킬의 다양한 구성 요소를 이해하십시오.
- Alexa Skill 키트 및 ASK SDK에 대한 설명서를 검토하여 아이디어를 스킬로 전환하십시오.

자습서: GitHub 소스 리포지토리를 사용하여 프로젝트 만들기

AWS CodeStar를 사용하면 리포지토리를 설정하여 프로젝트 팀과 함께 pull 요청을 만들고, 검토하고, 병합할 수 있습니다.

이 튜토리얼에서는 GitHub 리포지토리의 샘플 웹 애플리케이션 소스 코드가 포함된 프로젝트, 변경 사항을 배포하는 파이프라인, 애플리케이션이 클라우드에 호스팅되는 EC2 인스턴스를 생성합니다. 프로젝트가 생성된 후 이 자습서에서는 웹 애플리케이션의 홈 페이지를 변경하는 GitHub pull 요청을 만들고 병합하는 방법을 보여줍니다.

주제

- 1단계: 프로젝트 생성 및 GitHub 리포지토리 생성

- 2단계: 소스 코드 확인
- 3단계: GitHub Pull 요청 생성

1단계: 프로젝트 생성 및 GitHub 리포지토리 생성

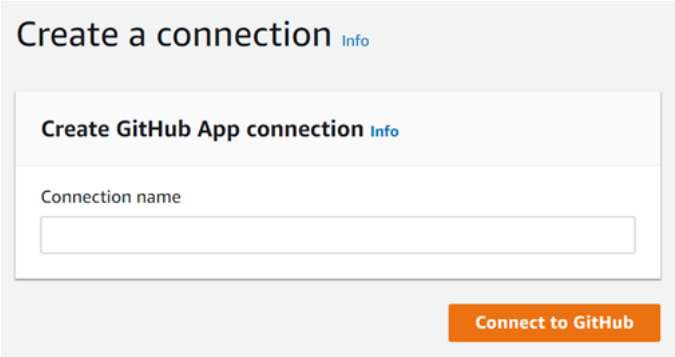
이 단계에서는 콘솔을 사용하여 프로젝트를 만들고 새 GitHub 리포지토리에 연결합니다. GitHub 리포지토리에 액세스하려면 GitHub를 통한 권한 부여를 관리하는 데 AWS CodeStar에서 사용하는 연결 리소스를 생성해야 합니다. 프로젝트가 생성되면 추가 리소스가 자동으로 프로비저닝됩니다.

1. 그런 다음 AWS Management Console에 로그인하고 https://console.aws.amazon.com/codestar/에서 AWS CodeStar 콘솔을 열 수 있습니다.
2. 프로젝트와 리소스를 생성하려는 AWS 리전을 선택합니다.
3. AWS CodeStar 페이지에서 프로젝트 생성을 선택합니다.
4. 프로젝트 템플릿 선택 페이지에서 웹 애플리케이션, Node.js 및 Amazon EC2 확인란을 선택합니다. 그런 다음 해당 옵션 세트에 사용 가능한 템플릿 중에서 선택합니다.

   자세한 내용은 AWS CodeStar 프로젝트 템플릿 섹션을 참조하세요.
5. 다음을 선택합니다.
6. 프로젝트 이름에 프로젝트의 이름(예: **`MyTeamProject`**)을 입력합니다. 다른 이름을 사용하는 경우 이 자습서 전체에서 해당 이름을 사용해야 합니다.
7. 프로젝트 리포지토리에서 GitHub를 선택합니다.
8. GitHub를 선택한 경우 연결 리소스를 선택하거나 생성해야 합니다. 기존 연결이 있는 경우 검색 필드에서 해당 연결을 선택합니다. 기존 연결이 없는 경우 여기에서 새 연결을 생성합니다. GitHub에 연결을 선택합니다.

   연결 생성 페이지가 표시됩니다.

> **Note**
>
> 연결을 생성하려면 GitHub 계정이 필요합니다. 조직 연결을 생성하는 경우 조직 소유자여야 합니다.



a. GitHub 앱 연결 생성 아래의 연결 이름에 연결 이름을 입력합니다. GitHub에 연결을 선택합니다.

   GitHub에 연결 페이지가 나타나고 GitHub 앱 필드가 표시됩니다.

b. GitHub 앱에서 앱 설치를 선택하거나 새 앱 설치를 선택하여 앱을 새로 만듭니다.

> **Note**
>
> 특정 공급자에 대한 모든 연결에 대해 하나의 앱을 설치합니다. 이미 설치한 경우 AWS Connector for GitHub 앱을 선택한 다음 이 단계를 건너뜁니다.

c. AWS Connect for GitHub 설치 페이지에서 앱을 설치할 계정을 선택합니다.

> **Note**
>
> 이전에 앱을 설치한 경우 구성을 선택하여 앱 설치의 수정 페이지로 이동하거나 뒤로 버튼을 사용하여 콘솔로 돌아갈 수 있습니다.

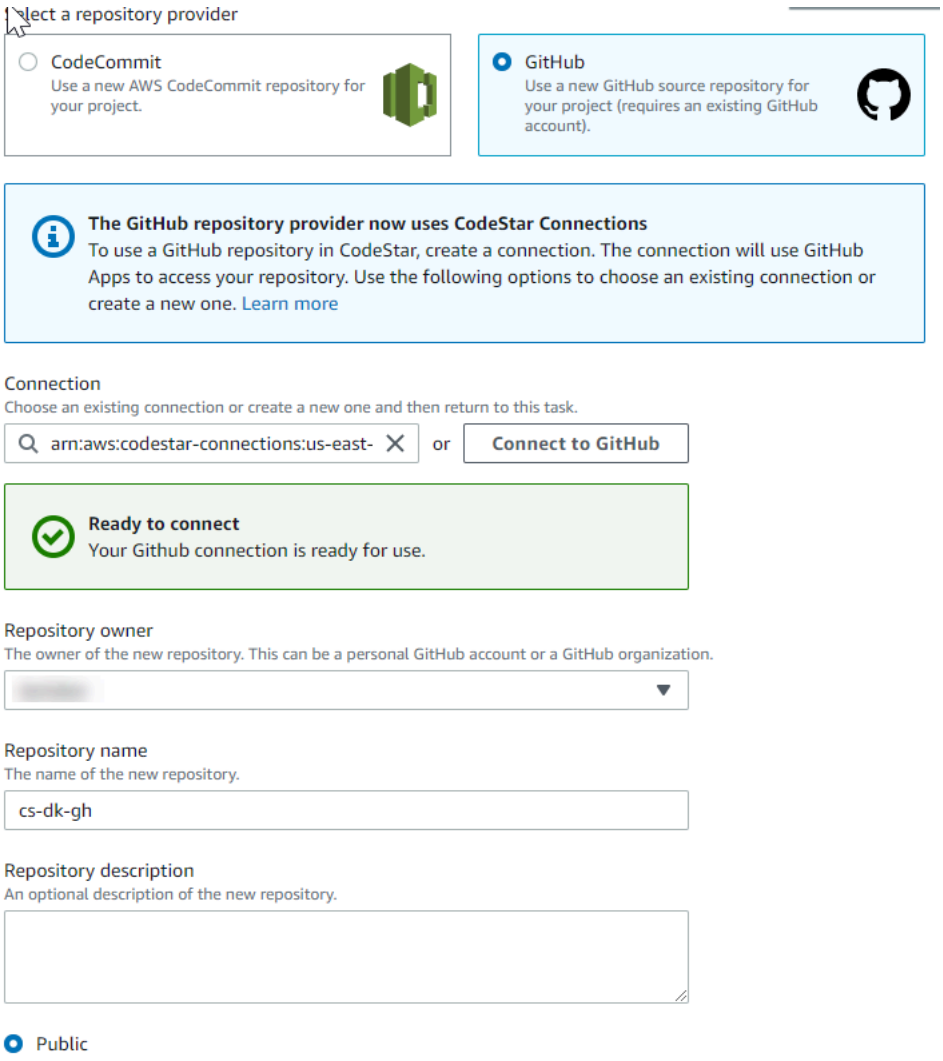
d. 계속하려면 암호 확인 페이지가 표시되면 GitHub 암호를 입력한 다음 로그인을 선택합니다.

e. AWS GitHub용 커넥터 설치 페이지에서 기본값을 그대로 두고 설치를 선택합니다.

f. GitHub에 연결 페이지의 GitHub 앱에 새 설치의 설치 ID가 표시됩니다.

   연결이 성공적으로 생성되면 CodeStar 프로젝트 생성 페이지에 연결 준비 완료 메시지가 표시됩니다.

> **Note**
>
> 개발자 도구 콘솔의 설정에서 연결을 볼 수 있습니다. 자세한 정보는 연결 시작하기를 참조하십시오.



g. 리포지토리 소유자의 경우 GitHub 조직이나 사용자의 개인 GitHub 계정을 선택합니다.

h. 리포지토리 이름에는 기본 GitHub 리포지토리 이름을 그대로 이용하거나 다른 이름을 입력합니다.

i. 퍼블릭 또는 프라이빗을 선택합니다.

> **Note**
>
> 또한 개발 환경으로 AWS Cloud9를 사용하려면 퍼블릭 리포지토리를 선택해야 합니다.

j. (선택 사항) 리포지토리 설명에 GitHub 리포지토리에 대한 설명을 입력합니다.

9. 프로젝트가 Amazon EC2 인스턴스에 배포되어 있고 변경하려는 경우 Amazon EC2 구성에서 Amazon EC2 인스턴스를 구성하십시오. 예를 들어, 프로젝트에 사용 가능한 인스턴스 유형 중에서 선택할 수 있습니다.

   키 페어에서 4단계: AWS CodeStar 프로젝트에 대한 Amazon EC2 키 페어 만들기에서 생성한 Amazon EC2 키 페어를 선택합니다. 프라이빗 키 파일에 대한 액세스 권한이 있음을 인정함을 선택합니다.

10. 다음을 선택합니다.
11. 리소스와 구성 세부 정보를 검토합니다.
12. [Next] 또는 [Create project]를 선택합니다. (표시되는 선택 사항은 프로젝트 템플릿에 따라 다릅니다.)

    프로젝트가 생성되는 동안 몇 분 정도 기다려 주세요.

13. 프로젝트를 만든 후 애플리케이션 보기를 선택하여 웹 애플리케이션을 확인합니다.

2단계: 소스 코드 확인

이 단계에서는 소스 코드와 소스 리포지토리에 사용할 수 있는 도구를 확인합니다.

1. 프로젝트의 탐색 모음에서 리포지토리를 선택합니다.

   GitHub에서 커밋 목록을 보려면 커밋 보기를 선택합니다. 그러면 GitHub에서 커밋 기록이 열립니다.

   이슈를 보려면 프로젝트의 이슈 탭을 선택하세요. GitHub에서 새 이슈를 생성하려면 GitHub 이슈 생성을 선택합니다. 그러면 GitHub에서 리포지토리 이슈 양식이 열립니다.

2. 리포지토리 탭에서 리포지토리 이름 아래의 링크를 선택하면 프로젝트 리포지토리가 새 탭 또는 창에 열립니다. 이 리포지토리에는 프로젝트의 소스 코드가 들어 있습니다.

3단계: GitHub Pull 요청 생성

이 단계에서는 소스 코드를 약간 변경하고 Pull 요청을 생성합니다.

1. GitHub에서 리포지토리에 새 기능 브랜치를 만드세요. 기본 브랜치 드롭다운 필드를 선택하고 이름이 지정된 `feature-branch` 필드에 새 브랜치를 입력합니다. 브랜치 생성을 선택합니다. 브랜치가 자동으로 생성되고 체크아웃됩니다.
2. GitHub에서 `feature-branch` 브랜치를 변경합니다. 공용 폴더를 열고 `index.html` 파일을 엽니다.
3. GitHub에서 Pull 요청을 생성하려면 AWS CodeStar 콘솔의 Pull 요청에서 Pull 요청 생성을 선택합니다. 그러면 GitHub에서 리포지토리 Pull 요청 양식이 열립니다. GitHub에서 연필 아이콘을 선택하여 파일을 편집합니다.

   `Congratulations!` 후에 `Well done, <name>!` 문자열을 추가하고 사용자 이름으로 `<name>`을 바꿉니다. 변경 사항 커밋을 선택합니다. 변경 사항은 기능 브랜치에 커밋됩니다.
4. AWS CodeStar 콘솔에서 프로젝트를 선택합니다. 리포지토리 탭을 선택합니다. Pull 요청에서 Pull 요청 생성을 선택합니다.

   양식이 GitHub에서 열립니다. 메인 브랜치는 기본 브랜치에 그대로 두십시오. 비교 대상에서 기능 브랜치를 선택하십시오. 차이 보기
5. GitHub에서 Pull 요청 생성을 선택합니다. Update index.html이라는 이름의 Pull 요청이 생성됩니다.
6. AWS CodeStar 콘솔에서 새 Pull 요청을 확인할 수 있습니다. 변경 사항 병합을 선택하여 변경 내용을 리포지토리에 커밋하고 Pull 요청을 리포지토리의 기본 브랜치에 병합합니다.
7. AWS CodeStar의 프로젝트로 돌아가서 파이프라인 페이지를 확인하세요. 이제 파이프라인 배포가 표시되어야 합니다.
8. 프로젝트를 만든 후 애플리케이션 보기를 선택하여 웹 애플리케이션을 확인합니다.

AWS CodeStar 프로젝트 템플릿

AWS CodeStar 프로젝트 템플릿을 이용하면 배포 프로젝트 지원을 위해 개발된 AWS 리소스를 이용해 샘플 애플리케이션을 시작하고 배포할 수 있습니다. AWS CodeStar 프로젝트 템플릿을 선택하면 애플리케이션 유형, 프로그래밍 언어, 컴퓨팅 플랫폼이 제공됩니다. 웹 애플리케이션, 웹 서비스, Alexa Skills, 정적 웹 페이지를 이용해 프로젝트를 만들고 나면, 샘플 애플리케이션을 자신의 애플리케이션으로 교체할 수 있습니다.

AWS CodeStar가 프로젝트를 생성하면, 애플리케이션 전달을 지원하는 AWS 리소스를 수정할 수 있습니다. AWS CodeStar는 AWS CloudFormation와 함께 작동해 사용자가 코드를 이용해 서비스와 클라우드의 서버/서버리스 플랫폼을 지원할 수 있게 합니다. AWS CloudFormation을 이용하면 전체 인프라를 텍스트 파일에 모델링할 수 있습니다.

주제

AWS CodeStar 프로젝트 파일 및 리소스

AWS CodeStar 프로젝트는 소스 코드와 코드 배포를 위해 생성한 리소스의 결합입니다. 코드 빌드, 릴리스, 배포에 도움이 되는 리소스 모음을 도구 체인 리소스라고 합니다. 프로젝트 생성 시 AWS CloudFormation 템플릿은 지속적인 통합/지속적인 배포(CI/CD) 파이프라인에 도구 체인 리소스를 공급합니다.

AWS 리소스 생성 경험 수준에 따라 두 가지 방식으로 AWS CodeStar를 이용해 프로젝트를 만들 수 있습니다.

- 콘솔을 이용해 프로젝트를 만들면, AWS CodeStar는 리포지토리를 포함한 도구 체인 리소스를 만들고 샘플 애플리케이션 코드와 프로젝트 파일로 리포지토리를 채웁니다. 콘솔을 이용해 사전 구성된 프로젝트 옵션을 바탕으로 샘플 프로젝트를 빠르게 설정하십시오.
- CLI를 사용하여 프로젝트를 생성하면, 도구 체인 리소스를 생성하는 AWS CloudFormation 템플릿 및 애플리케이션 소스 코드를 제공하게 됩니다. CLI를 이용해 AWS CodeStar가 템플릿을 바탕으로 프로젝트를 만든 다음 리포지토리를 샘플 코드로 채우게 하십시오.

AWS CodeStar 프로젝트는 관리를 위한 단일 창구를 제공합니다. 콘솔에서 프로젝트 만들기 마법사를 이용해 샘플 프로젝트를 설정할 수 있습니다. 그런 다음 해당 프로젝트를 팀이 권한과 리소스를 관리하는 협업 플랫폼으로 사용하십시오. 자세한 내용은 AWS CodeStar이란 무엇입니까? 섹션을 참조하세요. 콘솔을 이용해 프로젝트를 만들면, 소스 코드는 샘플 코드로 제공되며, CI/CD 도구 체인 리소스가 사용자를 위해 생성됩니다.

콘솔에서 프로젝트를 만들면, AWS CodeStar는 다음 리소스를 제공합니다.

- GitHub 또는 CodeCommit의 코드 리포지토리
- 파일과 디렉터리의 세부 정보를 제공하는, 프로젝트 리포지토리의 `README.md` 파일
- 애플리케이션의 실행 시간 스택에 대한 정의를 보관하는, 프로젝트 리포지토리의 `template.yml` 파일 이 파일을 이용해 도구 체인 리소스가 아닌 프로젝트 리소스(예: 알림, 데이터베이스 지원, 모니터링, 추적에 활용되는 AWS 리소스)를 추가 또는 수정할 수 있습니다.
- Amazon S3 아티팩트 버킷, Amazon CloudWatch Events 및 관련 서비스 역할 같은 파이프라인과 관련해 생성된 AWS 서비스 및 리소스
- 전체 소스 코드와 퍼블릭 HTTP 엔드포인트가 있는 작동하는 샘플 애플리케이션
- AWS CodeStar 프로젝트 템플릿 유형을 바탕으로 한 AWS 컴퓨팅 리소스:
  - Lambda 함수
  - Amazon EC2 인스턴스
  - AWS Elastic Beanstalk 환경
- 2018년 12월 6일(PDT) 부터:
  - 권한 경계 - 프로젝트 리소스에 대한 액세스를 제어하기 위한 특수 IAM 정책 권한 경계는 샘플 프로젝트의 역할에 기본적으로 연결되어 있습니다. 자세한 내용은 작업자 역할의 IAM 권한 경계를 참조하십시오.
  - AWS CloudFormation을 사용하여 프로젝트 리소스를 생성하기 위한 AWS CloudFormation IAM 역할(IAM 역할을 포함하여 지원되는 모든 AWS CloudFormation 리소스에 대한 권한 포함)
  - 도구 체인 IAM 역할
  - 애플리케이션 스택에 정의된 Lambda의 실행 역할(수정 가능)
- 2018년 12월 6일(PDT) 이전:
  - 제한된 AWS CloudFormation 리소스 세트를 지원하는 프로젝트 리소스를 생성하기 위한 AWS CloudFormation IAM 역할
  - CodePipeline 리소스를 생성하기 위한 IAM 역할
  - CodeBuild 리소스를 생성하기 위한 IAM 역할

- CodeDeploy 리소스를 생성하기 위한 IAM 역할(프로젝트 유형에 해당하는 경우)
- Amazon EC2 웹 앱을 생성하기 위한 IAM 역할(프로젝트 유형에 해당하는 경우)
- CloudWatch Events 리소스를 생성하기 위한 IAM 역할.
- 일부 리소스 세트를 포함하도록 동적으로 수정된 Lambda의 실행 역할

또한 프로젝트는 상태, 팀 관리 링크, IDE 또는 리포지토리 설정 지침 링크, 리포지토리의 소스 코드 변경 커밋 이력을 표시하는 세부정보 페이지도 제공합니다. Jira 같은 외부 문제 추적 도구를 연결하는 도구도 선택할 수 있습니다.

시작하기: 프로젝트 템플릿 선택

콘솔에서 AWS CodeStar 프로젝트를 선택하면, 빠른 시작을 위해 샘플 코드 및 리소스로 사전 구성된 다양한 옵션을 선택할 수 있습니다. 이러한 옵션을 프로젝트 템플릿이라고 합니다. 각각의 AWS CodeStar 프로젝트 템플릿은 프로그래밍 언어, 애플리케이션 유형, 컴퓨팅 플랫폼으로 구성됩니다. 선택한 조합에 따라 프로젝트 템플릿이 결정됩니다.

템플릿 컴퓨팅 플랫폼을 선택합니다.

각 템플릿은 다음 컴퓨팅 플랫폼 유형 중 하나를 구성합니다.

- AWS Elastic Beanstalk 프로젝트를 선택하면 AWS Elastic Beanstalk 환경을 클라우드의 Amazon Elastic Compute Cloud 인스턴스에 배포하게 됩니다.
- Amazon EC2 프로젝트를 선택하면 AWS CodeStar는 클라우드의 애플리케이션을 호스팅할 Linux EC2 인스턴스를 생성합니다. 프로젝트 팀원은 인스턴스에 액세스할 수 있고, 팀은 사용자가 SSH에 입력한 키 페어를 이용해 Amazon EC2 인스턴스에 들어갑니다. AWS CodeStar는 팀원 권한을 이용해 키 페어 연결을 관리하는 관리형 SSH도 이용합니다.
- AWS Lambda를 선택하면, AWS CodeStar는 인스턴스나 서버를 유지하지 않아도 되는, Amazon API Gateway를 통해 액세스하는 서버리스 환경을 생성합니다.

Template Application Type(템플릿 애플리케이션 유형)을 선택합니다.

각 템플릿은 다음 애플리케이션 유형 중 하나를 구성합니다.

- 웹 서비스

웹 서비스는 API 직접 호출처럼 배경에서 실행하는 작업에 사용합니다. AWS CodeStar가 샘플 웹 서비스 프로젝트를 생성하면, 엔드포인트 URL을 선택해 Hello World 출력을 볼 수 있지만, 이 애플리케이션 유형의 주된 용도는 사용자 인터페이스(UI)가 아닙니다. 이 범주의 AWS CodeStar 프로젝트 템플릿은 Ruby, Java, ASP.NET, PHP, Node.js 등을 사용한 개발을 지원합니다.

- 웹 애플리케이션

  웹 애플리케이션은 UI를 제공합니다. AWS CodeStar가 샘플 웹 애플리케이션 프로젝트를 만들면, 사용자는 엔드포인트 URL을 선택해 대화형 웹 애플리케이션을 볼 수 있습니다. 이 범주의 AWS CodeStar 프로젝트 템플릿은 Ruby, Java, ASP.NET, PHP, Node.js 등을 사용한 개발을 지원합니다.

- 정적 웹 페이지

  HTML 웹 사이트용 프로젝트를 원한다면 이 템플릿을 선택하십시오. 이 범주의 AWS CodeStar 프로젝트 템플릿은 HTML5를 사용한 개발을 지원합니다.

- Alexa 스킬

  AWS Lambda 함수를 포함한 Alexa 스킬 프로젝트를 원하는 경우 이 템플릿을 선택합니다. 스킬 프로젝트를 생성할 때 AWS CodeStar는 서비스 엔드포인트로 사용할 수 있는 Amazon 리소스 이름(ARN)을 반환합니다. 자세한 내용은 Lambda 함수로 사용자 지정 스킬 호스팅 항목을 참조하십시오.

  > **Note**
  >
  > Alexa 스킬에 대한 Lambda 함수는 미국 동부(버지니아 북부), 미국 서부(오레곤), EU(아일랜드) 및 아시아 태평양(도쿄) 리전에서만 지원됩니다.

- Config 규칙

  계정의 AWS 리소스에 대한 규칙을 자동화할 수 있는 AWS Config 규칙용 프로젝트를 원한다면 이 템플릿을 선택하십시오. 이 기능은 규칙용 서비스 엔드포인트로 사용할 수 있는 ARN을 반환합니다.

템플릿 프로그래밍 언어 선택

프로젝트 템플릿을 선택할 때는 Ruby, Java, ASP.NET, PHP, Node.js 등의 프로그래밍 언어를 선택해야 합니다.

AWS CodeStar 프로젝트에 변경 사항을 적용하는 방법

다음을 수정하면 프로젝트를 업데이트할 수 있습니다.

- 애플리케이션을 위한 샘플 코드 및 프로그래밍 언어 리소스
- 애플리케이션이 저장되고 배포된 인프라(운영 체제, 지원 애플리케이션 및 서비스, 배포 파라미터, 클라우드 컴퓨팅 플랫폼)를 구성하는 리소스 `template.yml` 파일에 있는 애플리케이션 리소스를 수정할 수 있습니다. 이것은 애플리케이션의 실행 시간 환경을 모델링하는 AWS CloudFormation 파일입니다.

> **Note**
>
> Alexa 스킬 AWS CodeStar 프로젝트로 작업하는 경우 AWS CodeStar 소스 리포지토리(CodeCommit 또는 GitHub) 외부에서 스킬을 변경할 수 없습니다. Alexa 개발자 포털에서 스킬을 편집하면 변경 사항이 소스 리포지토리에 표시되지 않을 수도 있고 두 버전이 동기화되지 않을 수 있습니다.

애플리케이션 소스 코드 및 푸시 변경 사항 변경

샘플 소스 코드, 스크립트 및 기타 애플리케이션 소스 파일을 수정하려면, 다음 방법으로 소스 리포지토리에 있는 파일을 편집하십시오.

- CodeCommit 또는 GitHub의 편집 모드 사용
- AWS Cloud9 같은 IDE에서 프로젝트 열기
- 리포지토리를 로컬로 복제한 다음 변경사항을 커밋하고 푸시 자세한 내용은 4단계: 변경 커밋을 참조하세요.

Template.yml 파일로 애플리케이션 리소스 변경

인프라 리소스를 직접 수정하는 대신, AWS CloudFormation을 사용해 애플리케이션의 실행 시간 리소스를 모델링하고 배포합니다.

프로젝트 리포지토리에 있는 `template.yml` 파일을 편집하면 실행 시간 스택에 있는 Lambda 함수 같은 애플리케이션 리소스를 수정 또는 추가할 수 있습니다. AWS CloudFormation 리소스로 사용할 수 있는 리소스는 모두 추가할 수 있습니다.

AWS Lambda 함수의 코드나 설정을 변경하는 방법은 프로젝트에 리소스 추가 단원을 참조하십시오.

프로젝트의 리포지토리에 있는 `template.yml` 파일을 수정해 애플리케이션 리소스인 AWS CloudFormation 리소스 유형을 추가하십시오. 애플리케이션 리소스를 `template.yml` 파일의 `Resources` 섹션에 추가하면, AWS CloudFormation 및 AWS CodeStar가 사용자를 위한 리소스를 생성합니다. AWS CloudFormation 리소스 및 필수 속성 목록은 AWS 리소스 유형 참조에서 확인할 수 있습니다. 자세한 내용은 1단계: IAM의 CloudFormation 작업자 역할 편집에 있는 이 예제를 참조하십시오.

AWS CodeStar를 이용하면 사용자는 애플리케이션의 실행 시간 환경을 구성하고 모델링해 모범 사례를 구현할 수 있습니다.

애플리케이션 리소스 변경을 위해 권한을 관리하는 방법

AWS CloudFormation을 이용해 Lambda 함수 같은 실행 시간 애플리케이션 리소스를 추가하면, AWS CloudFormation 작업자 역할은 자신이 보유 중인 권한을 사용할 수 있습니다. 일부 실행 시간 애플리케이션 리소스의 경우, `template.yml` 파일을 편집하려면 AWS CloudFormation 작업자 역할의 권한을 직접 조정해야 합니다.

AWS CloudFormation 작업자 역할의 권한을 변경하는 예제는 5단계: 인라인 정책이 있는 리소스 권한 추가에서 확인할 수 있습니다.

AWS CodeStar 모범 사례

AWS CodeStar은 여러 제품 및 서비스와 통합되어 있습니다. 다음 단원에서는 AWS CodeStar의 모범 사례 및 이와 관련된 제품과 서비스를 설명합니다.

주제

AWS CodeStar 리소스에 대한 보안 모범 사례

애플리케이션이 사용하는 종속성에 대해 패치를 정기적으로 적용하고 보안 모범 사례를 검토해야 합니다. 이러한 보안 모범 사례를 이용해 샘플 코드를 업데이트하고 생산 환경의 프로젝트를 유지하십시오.

- 현재 진행 중인 보안 알림 및 프레임워크에 대한 업데이트를 추적합니다.
- 프로젝트를 배포하기 전에, 프레임워크를 위해 개발한 모범 사례를 따르십시오.
- 프레임워크에 대한 종속성을 정기적으로 검토하고 필요하다면 업데이트하십시오.
- 각각의 AWS CodeStar 템플릿에는 프로그래밍 언어에 대한 구성 지침이 포함되어 있습니다. 프로젝트의 소스 리포지토리에 있는 `README.md` 파일을 확인하십시오.
- 프로젝트 리소스를 격리하는 가장 좋은 방법은 AWS CodeStar의 보안에서 소개한 다중 계정 전략을 사용하여 AWS 리소스에 대한 최소 권한 액세스를 관리하는 것입니다.

종속성 버전 설정의 모범 사례

AWS CodeStar 프로젝트의 샘플 소스 코드는 소스 리포지토리의 `package.json` 파일에 나와 있는 종속성을 사용합니다. 모범 사례는 언제나 종속성이 특정 버전을 가리키도록 설정하는 것입니다. 이것을 버전 고정이라고 합니다. 버전을 `latest`로 설정하는 것은 권장하지 않습니다. 예고 없이 애플리케이션이 중단될 수 있는 변경사항이 적용될 수 있기 때문입니다.

AWS CodeStar 리소스에 대한 모니터링 및 로깅 모범 사례

AWS의 로깅 기능을 사용하여 사용자가 계정에서 수행한 작업과 사용한 리소스를 확인할 수 있습니다. 로그 파일에는 다음 정보가 들어 있습니다.

- 작업의 시간과 날짜
- 작업의 소스 IP 주소
- 부족한 권한으로 인해 실패한 작업

AWS CloudTrail을 사용하여 AWS 계정에서 또는 이를 대신하여 수행된 AWS API 직접 호출 및 관련 이벤트를 기록할 수 있습니다. 자세한 내용은 AWS CloudTrail을 사용하여 AWS CodeStar API 직접 호출 로깅 섹션을 참조하세요.

AWS CodeStar의 프로젝트 작업

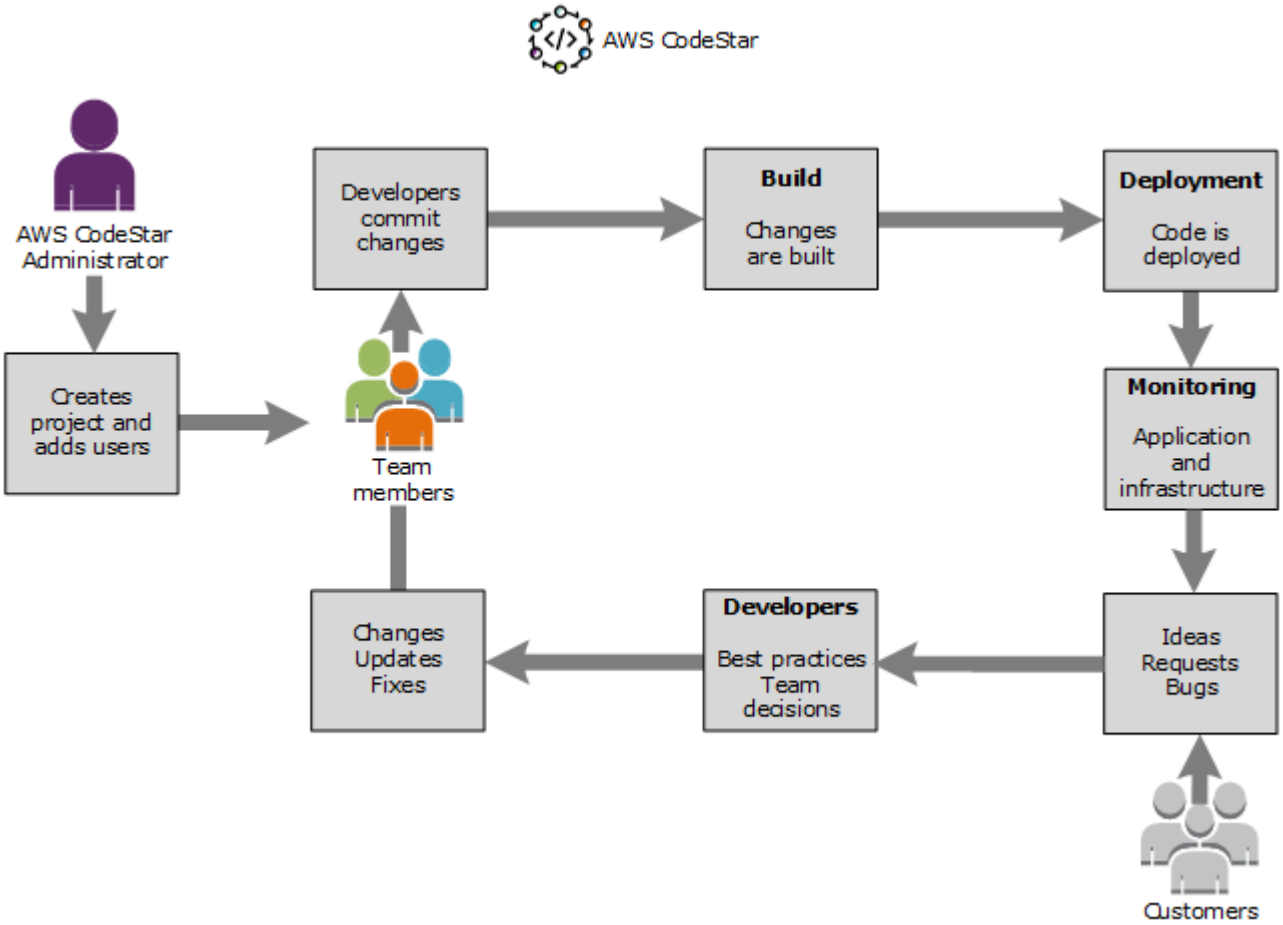
AWS CodeStar 프로젝트 템플릿을 사용할 경우 다음을 비롯하여 필요한 리소스와 함께 이미 구성되어 있는 프로젝트를 빠르게 생성할 수 있습니다.

- 소스 리포지토리
- 빌드 환경
- 배포 및 호스팅 리소스
- 프로그래밍 언어

프로젝트 작업을 즉시 시작할 수 있도록 템플릿에는 샘플 소스 코드가 포함되어 있습니다.

프로젝트를 생성하면 리소스를 추가 또는 제거하고, 프로젝트 대시보드를 사용자 지정하고 진행률을 모니터링할 수 있습니다.

다음 다이어그램은 AWS CodeStar 프로젝트의 기본 워크플로우를 보여 줍니다.



다이어그램의 기본 워크플로우는 AWSCodeStarFullAccess 정책을 적용한 개발자가 프로젝트를 생성하고 팀원을 프로젝트에 추가하는 것을 보여줍니다. 또한 코드를 작성, 빌드, 테스트 및 배포합니다. 프로젝트 대시보드는 애플리케이션 활동을 보고, 빌드 및 배포 파이프라인을 통한 코드 흐름 등을 모니터링하기 위해 실시간으로 사용할 수 있는 도구를 제공합니다. 팀에서는 팀 wiki 타일을 사용하여 정보, 모범 사례 및 링크를 공유합니다. 또한 문제 추적 소프트웨어를 통합하여 진행률 및 작업을 추적할 수 있습니다. 고객이 요청과 피드백을 제공하면, 팀은 이 정보를 프로젝트에 추가한 후 프로젝트 계획 및 개발에 통합합니다. 프로젝트가 증가하면 팀은 팀원을 추가하여 코드 베이스를 지원합니다.

AWS CodeStar에서 프로젝트 만들기

AWS CodeStar 콘솔을 사용하여 프로젝트를 만듭니다. 프로젝트 템플릿을 사용하면 사용자에게 필요한 리소스가 설정됩니다. 템플릿에는 코딩을 시작하는 데 도움이 되는 샘플 코드도 포함돼 있습니다.

프로젝트를 만들려면 AWSCodeStarFullAccess 정책이 있거나 이와 동등한 권한이 있는 사용자로 AWS Management Console에 로그인합니다. 자세한 내용은 AWS CodeStar설정 섹션을 참조하세요.

> **Note**
>
> 이 주제의 절차를 완료하려면 먼저 AWS CodeStar설정 단원의 단계를 완료해야 합니다.

주제

- AWS CodeStar에서 프로젝트 생성(콘솔)
- AWS CodeStar에서 프로젝트 생성(AWS CLI)

AWS CodeStar에서 프로젝트 생성(콘솔)

AWS CodeStar 콘솔을 사용하여 프로젝트를 만듭니다.

AWS CodeStar에서 프로젝트를 만들려면

1. 그런 다음 AWS Management Console에 로그인하고 https://console.aws.amazon.com/glue/에서 AWS CodeStar 콘솔을 열 수 있습니다.

   프로젝트 및 그 리소스를 만들려는 AWS 리전에 로그인해야 합니다. 예를 들어 미국 동부(오하이오)에서 프로젝트를 만들려면 해당 AWS 리전을 선택해야 합니다. AWS CodeStar이 지원되는 AWS 리전에 대한 자세한 내용은 AWS 일반 참조의 리전 및 엔드포인트를 참조하세요.

2. AWS CodeStar 페이지에서 프로젝트 생성을 선택합니다.
3. 프로젝트 템플릿 선택 페이지의 AWS CodeStar 프로젝트 템플릿 목록에서 프로젝트 유형을 선택합니다. 필터 막대를 사용하여 선택 범위를 좁힐 수 있습니다. 예를 들어 Amazon EC2 인스턴스에 배포할 Node.js로 작성된 웹 애플리케이션 프로젝트의 경우 웹 애플리케이션, Node.js, Amazon EC2 확인란을 선택합니다. 그런 다음 해당 옵션 세트에 사용 가능한 템플릿 중에서 선택합니다.

   자세한 내용은 AWS CodeStar 프로젝트 템플릿 섹션을 참조하세요.
4. 다음을 선택합니다.
5. 프로젝트 이름 텍스트 입력 필드에 프로젝트 이름(예: *My First Project*)을 입력합니다. 프로젝트 ID에서 프로젝트 ID는 이 프로젝트 이름에서 파생되지만, 15자로 제한됩니다.

   예를 들어 *My First Project*라는 프로젝트의 기본 ID는 *my-first-projec*입니다. 이 프로젝트 ID는 해당 프로젝트와 관련된 모든 리소스 이름의 토대입니다. AWS CodeStar는 이 프로젝트 ID를 코드 리포지토리에 대한 URL의 일부로 사용하고, IAM의 관련 보안 액세스 역할 및 정책 이름으로도 사용합니다. 프로젝트를 만든 후에는 프로젝트 ID를 변경할 수 없습니다. 프로젝트를 만들기 전에 프로젝트 ID를 편집하려면 프로젝트 ID에 사용하려는 ID를 입력합니다.

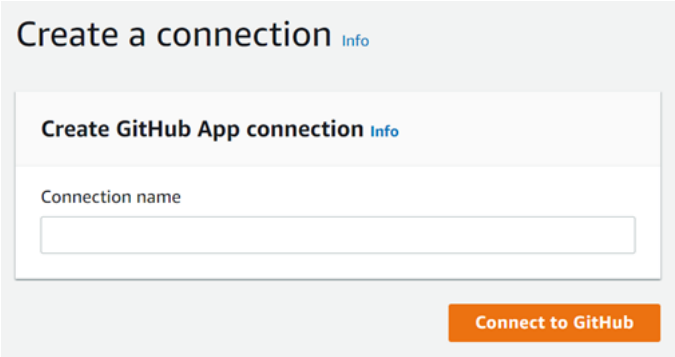
   프로젝트 이름 및 프로젝트 ID 제한에 대한 자세한 내용은 AWS CodeStar의 제한 값 단원을 참조하십시오.

   > **Note**
   >
   > 프로젝트 ID는 AWS 리전 내 AWS 계정에 대해 고유해야 합니다.

6. 리포지토리 공급자 AWS CodeCommit 또는 GitHub를 선택합니다.
7. AWS CodeCommit를 선택한 경우 리포지토리 이름에는 기본 AWS CodeCommit 리포지토리 이름을 그대로 이용하거나 다른 이름을 입력합니다. 그런 다음 9단계로 건너뜁니다.
8. GitHub를 선택한 경우 연결 리소스를 선택하거나 생성해야 합니다. 기존 연결이 있는 경우 검색 필드에서 해당 연결을 선택합니다. 기존 연결이 없는 경우 지금 새 연결을 생성합니다. GitHub에 연결을 선택합니다.

   연결 생성 페이지가 표시됩니다.

   > **Note**
   >
   > 연결을 생성하려면 GitHub 계정이 필요합니다. 조직 연결을 생성하는 경우 조직 소유자여야 합니다.



a. GitHub 앱 연결 생성 아래의 연결 이름 입력 텍스트 필드에 연결 이름을 입력합니다. GitHub에 연결을 선택합니다.

   GitHub에 연결 페이지가 나타나고 GitHub 앱 필드가 표시됩니다.

b. GitHub 앱에서 앱 설치를 선택하거나 새 앱 설치를 선택하여 앱을 새로 만듭니다.

> **Note**
>
> 특정 공급자에 대한 모든 연결에 대해 하나의 앱을 설치합니다. 이미 설치한 경우 AWS Connector for GitHub 앱을 선택한 다음 이 단계를 건너뜁니다.

c. AWS Connect for GitHub 설치 페이지에서 앱을 설치할 계정을 선택합니다.

> **Note**
>
> 이전에 앱을 설치한 경우 구성을 선택하여 앱 설치의 수정 페이지로 이동하거나 뒤로 버튼을 사용하여 콘솔로 돌아갈 수 있습니다.

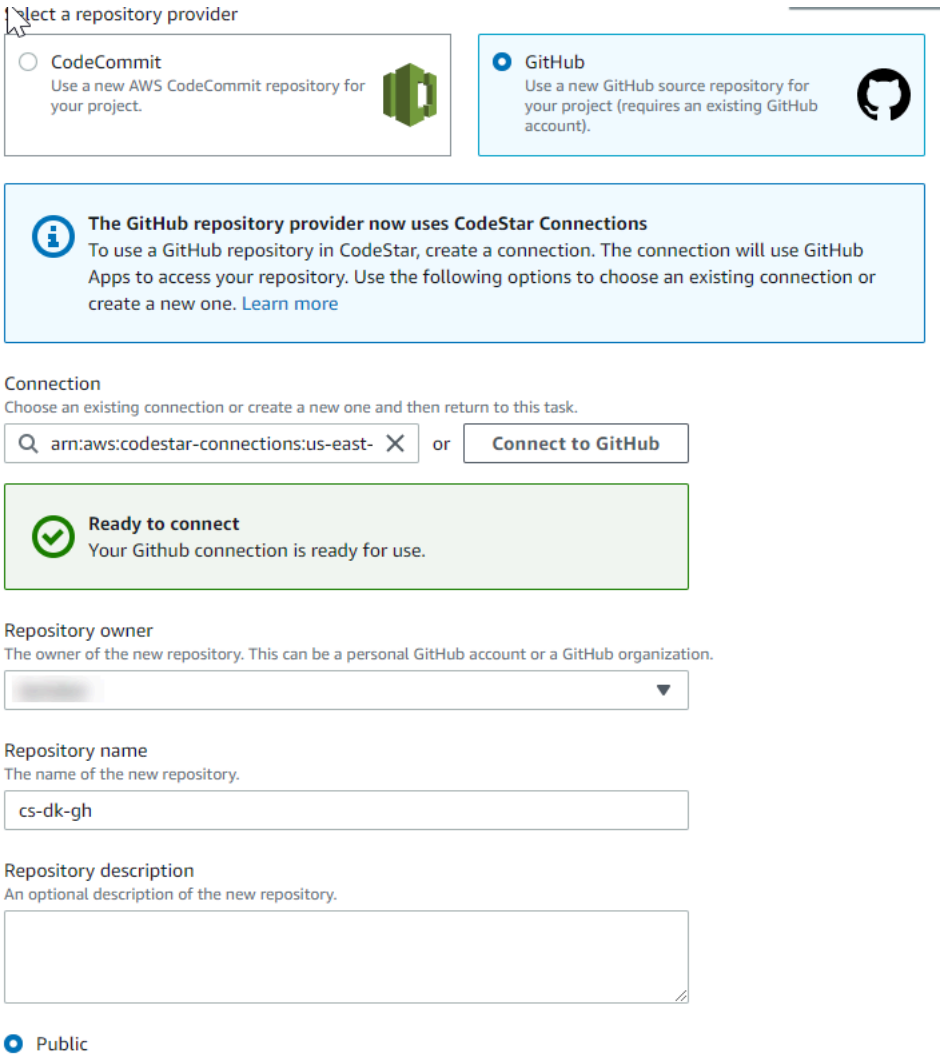
d. 계속하려면 암호 확인 페이지가 표시되면 GitHub 암호를 입력한 다음 로그인을 선택합니다.

e. GitHub용 AWS 커넥터 설치 페이지에서 기본값을 그대로 두고 설치를 선택합니다.

f. GitHub에 연결 페이지의 GitHub 앱 텍스트 입력 필드에 새 설치의 설치 ID가 표시됩니다.

   연결이 생성되면 CodeStar 프로젝트 생성 페이지에 연결 준비 완료 메시지가 표시됩니다.

> **Note**
>
> 개발자 도구 콘솔의 설정에서 연결을 볼 수 있습니다. 자세한 정보는 연결 시작하기를 참조하십시오.



g. 리포지토리 소유자의 경우 GitHub 조직이나 사용자의 개인 GitHub 계정을 선택합니다.

h. 리포지토리 이름에는 기본 GitHub 리포지토리 이름을 그대로 이용하거나 다른 이름을 입력합니다.

i. 퍼블릭 또는 프라이빗을 선택합니다.

> **Note**
>
> 또한 개발 환경으로 AWS Cloud9를 사용하려면 퍼블릭을 선택해야 합니다.

j. (선택 사항) 리포지토리 설명에 GitHub 리포지토리에 대한 설명을 입력합니다.

> **Note**
>
> Alexa Skill 프로젝트 템플릿을 선택하는 경우 Amazon 개발자 계정을 연결해야 합니다. Alexa Skill 프로젝트 작업에 대한 자세한 내용은 자습서: AWS CodeStar에 Alexa Skill 프로젝트 생성 단원을 참조하십시오.

9. 프로젝트가 Amazon EC2 인스턴스에 배포되어 있고 변경하려는 경우 Amazon EC2 구성에서 Amazon EC2 인스턴스를 구성하십시오. 예를 들어, 프로젝트에 사용 가능한 인스턴스 유형 중에서 선택할 수 있습니다.

> **Note**
>
> Amazon EC2 인스턴스 유형마다 서로 다른 수준의 컴퓨팅 성능을 제공하므로 관련 비용도 다를 수 있습니다. 자세한 내용은 Amazon EC2 인스턴스 유형 및 Amazon EC2 요금을 참조하세요.
>
> Amazon Virtual Private Cloud에서 만든 서브넷이 여러 개이거나 Virtual Private Cloud(VPC)가 두 개 이상인 경우, 사용할 서브넷 및 VPC를 선택할 수도 있습니다. 그러나 전용 인스턴스에서 지원되지 않는 Amazon EC2 인스턴스 유형을 선택하는 경우, 인스턴스 테넌시가 전용으로 설정된 VPC를 선택할 수 없습니다.
>
> 자세한 내용은 Amazon VPC란 무엇입니까?와 전용 인스턴스 기본 사항 단원을 참조하십시오.

키 페어에서 4단계: AWS CodeStar 프로젝트에 대한 Amazon EC2 키 페어 만들기에서 생성한 Amazon EC2 키 페어를 선택합니다. 프라이빗 키 파일에 대한 액세스 권한이 있음을 인정함을 선택합니다.

10. 다음을 선택합니다.
11. 리소스와 구성 세부 정보를 검토합니다.

12. [Next] 또는 [Create project]를 선택합니다. (표시되는 선택 사항은 프로젝트 템플릿에 따라 다릅니다.)

    프로젝트(리포지토리 포함)를 만드는 데 몇 분 정도 걸릴 수 있습니다.

13. 프로젝트에 리포지토리가 있으면 리포지토리 페이지를 사용하여 리포지토리에 대한 액세스를 구성할 수 있습니다. 다음 단계의 링크를 사용하여 IDE를 구성하거나, 이슈 트래킹을 설정하거나, 프로젝트에 팀 구성원을 추가할 수 있습니다.

프로젝트를 만드는 중에 팀원을 추가하거나 명령줄 또는 선호하는 IDE에서 프로젝트 리포지토리에 대한 액세스를 구성할 수 있습니다.

AWS CodeStar에서 프로젝트 생성(AWS CLI)

AWS CodeStar 프로젝트는 소스 코드와 코드 배포를 위해 생성한 리소스의 결합입니다. 코드 빌드, 릴리스, 배포에 도움이 되는 리소스 모음을 도구 체인 리소스라고 합니다. 프로젝트 생성 시 AWS CloudFormation 템플릿은 지속적인 통합/지속적인 배포(CI/CD) 파이프라인에 도구 체인 리소스를 공급합니다.

콘솔을 사용하여 프로젝트를 만들면, 도구 체인 템플릿이 생성됩니다. AWS CLI를 사용하여 프로젝트를 만들면, 도구 체인 리소스를 생성하는 도구 체인 템플릿을 생성하게 됩니다.

온전한 도구 체인은 다음과 같은 권장 리소스를 요구합니다.

1. 소스 코드가 포함된 CodeCommit 또는 GitHub 리포지토리입니다.
2. 리포지토리의 변경 사항을 주시하도록 구성된 CodePipeline 파이프라인입니다.
   a. CodeBuild를 이용해 단위 또는 통합 테스트를 실행할 때는, 파이프라인에 빌드 단계를 추가해 빌드 아티팩트를 생성하는 방법을 권장합니다.
   b. 되도록 CodeDeploy 또는 AWS CloudFormation을 사용하는 배포 단계를 파이프라인에 추가해 빌드 아티팩트와 소스 코드를 실행 시간 인프라에 배포하십시오.

> **Note**
>
> CodePipeline은 파이프라인에서 최소한 두 단계를 요구하며, 첫 번째 단계는 소스 단계여야 하므로 빌드나 배포 단계는 두 번째 단계로 추가하십시오.

AWS CodeStar 도구 체인은 CloudFormation 템플릿으로 정의됩니다.

이 작업을 진행하고 샘플 리소스를 설정하는 방법은 자습서: AWS CLI를 이용하여 AWS CodeStar에서 프로젝트 생성에서 확인할 수 있습니다.

사전 조건:

프로젝트를 만들 때는, 입력 파일에 다음과 같은 파라미터를 입력해야 합니다. 다음 파라미터를 입력하지 않으면, AWS CodeStar는 빈 프로젝트를 생성합니다.

- 소스 코드입니다. 이 파라미터가 요청에 포함되면, 도구 체인 템플릿도 포함해야 합니다.
  - 소스 코드에는 프로젝트를 실행하는 데 필요한 애플리케이션 코드가 포함되어야 합니다.
  - 소스 코드는 CodeBuild 프로젝트를 위한 buildspec.yml이나 CodeDeploy 배포를 위한 appspec.yml 같은 필수 구성 파일을 포함해야 합니다.
  - 소스 코드에 README나 도구 체인이 아닌 AWS 리소스 같은 선택 사항 항목을 포함할 수도 있습니다.
- 도구 체인 템플릿입니다. 도구 체인 템플릿은 프로젝트를 위해 관리해야 하는 AWS 리소스와 IAM 역할을 제공합니다.
- 소스 위치입니다. 프로젝트를 위한 소스 코드와 도구 체인 템플릿을 지정할 때는, 위치를 지정해야 합니다. 소스 파일과 도구 체인 템플릿을 버킷에 업로드하십시오. AWS CodeStar는 파일을 검색하고 이를 이용해 프로젝트를 생성합니다.

> **Important**
>
> AWS CLI에서 원하는 AWS 리전을 구성했는지 확인하십시오. 프로젝트는 AWS CLI에서 구성한 AWS 리전에 생성됩니다.

1. create-project 명령을 실행하고 `--generate-cli-skeleton` 파라미터를 포함하십시오.

   ```
   aws codestar create-project --generate-cli-skeleton
   ```

   JSON 형식 데이터가 출력에 표시됩니다. AWS CLI가 설치된 로컬 컴퓨터 또는 인스턴스의 위치에 있는 파일(예: *input.json*)에 데이터를 복사합니다. 복사된 데이터를 다음과 같이 수정하고 결과를 저장합니다.

   ```
   {
       "name": "project-name",
       "id": "project-id",
   ```

```
    "description": "description",
    "sourceCode": [
        {
            "source": {
                "s3": {
                    "bucketName": "s3-bucket-name",
                    "bucketKey": "s3-bucket-object-key"
                }
            },
            "destination": {
                "codeCommit": {
                    "name": "codecommit-repository-name"
                },
                "gitHub": {
                    "name": "github-repository-name",
                    "description": "github-repository-description",
                    "type": "github-repository-type",
                    "owner": "github-repository-owner",
                    "privateRepository": true,
                    "issuesEnabled": true,
                    "token": "github-personal-access-token"
                }
            }
        }
    ],
    "toolchain": {
        "source": {
            "s3": {
                "bucketName": "s3-bucket-name",
                "bucketKey": "s3-bucket-object-key"
            }
        },
        "roleArn": "service-role-arn",
        "stackParameters": {
            "KeyName": "key-name"
        }
    },
    "tags": {
        "KeyName": "key-name"
    }
}
```

다음을 바꿉니다.

- *project-name*: 필수 항목입니다. 이 AWS CodeStar 프로젝트의 표시 이름입니다.
- *project-id*: 필수 항목입니다. 이 AWS CodeStar 프로젝트의 프로젝트 ID입니다.

> **Note**
>
> 프로젝트를 생성할 때는 고유 프로젝트 ID가 있어야 합니다. 프로젝트 ID가 이미 존재하는 입력 파일을 전송하면 오류가 발생합니다.

- *description*: 선택 사항입니다. 이 AWS CodeStar 프로젝트의 설명입니다.
- *sourceCode*: 선택 사항입니다. 프로젝트에 제공된 소스 코드의 구성 정보입니다. 현재는 단일 sourceCode 객체만 지원됩니다. 각각의 sourceCode 객체에는 AWS CodeStar가 소스 코드를 검색한 위치와 소스 코드가 입력되는 대상 주소에 대한 정보가 포함됩니다.
  - *source*: 필수 항목입니다. 소스 코드를 업로드한 위치를 정의합니다. 유일하게 지원되는 소스는 Amazon S3입니다. AWS CodeStar는 소스 코드를 검색하며 프로젝트가 생성되면 이를 리포지토리에 포함합니다.
    - *S3*: 선택 사항입니다. 소스 코드의 Amazon S3 위치입니다.
      - *bucket-name*: 소스 코드를 포함하는 버킷입니다.
      - *bucket-key*: 소스 코드를 포함하는 .zip 파일(예: src.zip)을 지칭하는 버킷 접두사와 객체 키입니다.
  - *destination*: 선택 사항입니다. 프로젝트 생성 시 소스 코드가 입력되는 대상 위치입니다. 소스 코드에 지원되는 대상은 CodeCommit와 GitHub입니다.

    다음 두 가지 옵션 중 하나만 제공할 수 있습니다.
    - *codeCommit*: 유일한 필수 속성은 소스 코드를 포함해야 하는 CodeCommit 리포지토리의 이름입니다. 이 리포지토리는 사용자의 도구 체인 템플릿에 있어야 합니다.

      > **Note**
      >
      > CodeCommit의 경우에는 도구 체인 스택에서 정의한 리포지토리의 이름을 입력해야 합니다. AWS CodeStar는 Amazon S3에서 입력한 소스 코드를 이용해 이 리포지토리를 초기화합니다.

    - *gitHub*: 이 객체는 GitHub 리포지토리를 생성하는 데 필요한 정보를 나타내며, 소스 코드를 이용해 이를 시드합니다. GitHub 리포지토리를 선택하면, 다음 값을 입력해야 합니다.

> **Note**
>
> GitHub의 경우에는 기존 GitHub 리포지토리를 지정해선 안 됩니다. AWS CodeStar는 사용자를 대신해 리포지토리를 생성하고 사용자가 Amazon S3에 업로드한 소스 코드를 입력합니다. AWS CodeStar는 다음 정보를 이용해 GitHub에 리포지토리를 생성합니다.

- *name*: 필수 항목입니다. GitHub 리포지토리의 이름입니다.
- *description*: 필수 항목입니다. GitHub 리포지토리의 설명입니다.
- *type*: 필수 항목입니다. GitHub 리포지토리의 유형입니다. 유효한 값은 User 또는 Organization입니다.
- *owner*: 필수 항목입니다. The GitHub user 리포지토리 소유자의 이름입니다. 리포지토리를 GitHub 조직이 소유해야 한다면, 조직 이름을 입력하십시오.
- *privateRepository*: 필수 항목입니다. 이 리포지토리의 프라이빗/퍼블릭 여부를 결정합니다. 유효한 값은 true 또는 false입니다.
- *issuesEnabled*: 필수 항목입니다. 이 리포지토리를 이용해 GitHub에서 문제를 활성화할지를 결정합니다. 유효한 값은 true 또는 false입니다.
- *##*: 선택사항. AWS CodeStar가 사용자의 GitHub 계정에 액세스할 때 사용하는 개인용 액세스 토큰입니다. 이 토큰에는 다음 범위가 포함되어야 합니다. repo, user 및 admin:repo_hook. GitHub에서 개인용 액세스 토큰을 검색하려면 GitHub 웹 사이트의 명령줄에 대한 개인용 액세스 토큰 생성을 참조하십시오.

  > **Note**
  >
  > CLI를 사용하여 GitHub 소스 리포지토리가 있는 프로젝트를 생성하는 경우, AWS CodeStar는 토큰을 사용하여 OAuth 앱을 통해 리포지토리에 액세스합니다. 콘솔을 사용하여 GitHub 소스 리포지토리로 프로젝트를 생성하는 경우 AWS CodeStar는 GitHub 앱으로 리포지토리에 액세스하는 연결 리소스를 사용합니다.

- *## ##*: 프로젝트 생성 시 설정해야 하는 CI/CD 도구 체인 관련 정보입니다. 여기에는 도구 체인 템플릿을 업로드한 위치가 포함됩니다. 템플릿은 도구 체인 리소스가 포함된 AWS CloudFormation 스택을 생성합니다. 또한 참조를 위해 AWS CloudFormation이 재정의한

파라미터와 스택 생성을 위해 사용한 역할도 포함됩니다. AWS CodeStar는 템플릿을 검색하고 AWS CloudFormation을 이용해 템플릿을 실행합니다.

- *source*: 필수 항목입니다. 툴체인 템플릿의 위치입니다. Amazon S3가 유일하게 지원되는 소스 위치입니다.
  - *S3*: 선택 사항입니다. 도구 체인 템플릿을 업로드한 Amazon S3 위치입니다.
    - *bucket-name*: Amazon S3 버킷 이름입니다.
    - *bucket-key*: 도구 체인 템플릿을 포함하는 .yml 또는 .json 파일(예: `files/toolchain.yml`)을 지칭하는 버킷 접두사와 객체 키입니다.
- *stackParameters*: 선택 사항입니다. AWS CloudFormation으로 전달해야 하는 키-값 페어가 포함되어 있습니다. 이것은 (해당하는 경우) 도구 체인 템플릿이 참조를 위해 설정하는 파라미터입니다.
- *role*: 선택 사항입니다. 계정에서 도구 체인 리소스를 생성하는 데 사용한 역할입니다. 필요한 역할은 다음과 같습니다.
  - 역할을 입력하지 않으면, 도구 체인이 AWS CodeStar 빠른 시작 템플릿인 경우 AWS CodeStar는 사용자 계정에 대해 생성된 기본 서비스 역할을 사용합니다. 계정에 서비스 역할이 없다면 새로 만들면 됩니다. 자세한 내용은 2단계: AWS CodeStar 서비스 역할 생성을 참조하세요.
  - 자체의 사용자 지정 도구 체인 템플릿을 업로드하고 사용하는 경우에는 역할을 입력해야 합니다. AWS CodeStar 서비스 역할 및 정책 설명에 따라 역할을 생성할 수 있습니다. 이 정책 설명의 예제는 AWSCodeStarServiceRole 정책 단원을 참조하십시오.
- *tags*: 선택 사항입니다. 태그는 AWS CodeStar 프로젝트에 연결됩니다.

> **Note**
>
> 이 태그는 프로젝트에 포함된 리소스에는 연결되지 않습니다.

2. 방금 저장한 파일이 들어 있는 디렉터리로 전환한 다음, create-project 명령을 다시 실행합니다. `--cli-input-json` 파라미터를 포함합니다.

```
aws codestar create-project --cli-input-json file://input.json
```

3. 이 명령이 제대로 실행되면 다음과 비슷한 데이터가 출력에 표시됩니다.

```
{
    "id": "project-ID",
    "arn": "arn"
```

```
}
```

- 출력에는 새 프로젝트에 대한 정보가 들어 있습니다.
  - `id` 값은 프로젝트 ID를 나타냅니다.
  - `arn` 값은 프로젝트의 ARN을 나타냅니다.

4. describe-project 명령을 이용해 프로젝트 생성 상태를 확인하십시오. `--id` 파라미터를 포함합니다.

```
aws codestar describe-project --id <project_ID>
```

다음과 비슷한 데이터가 결과에 나타납니다.

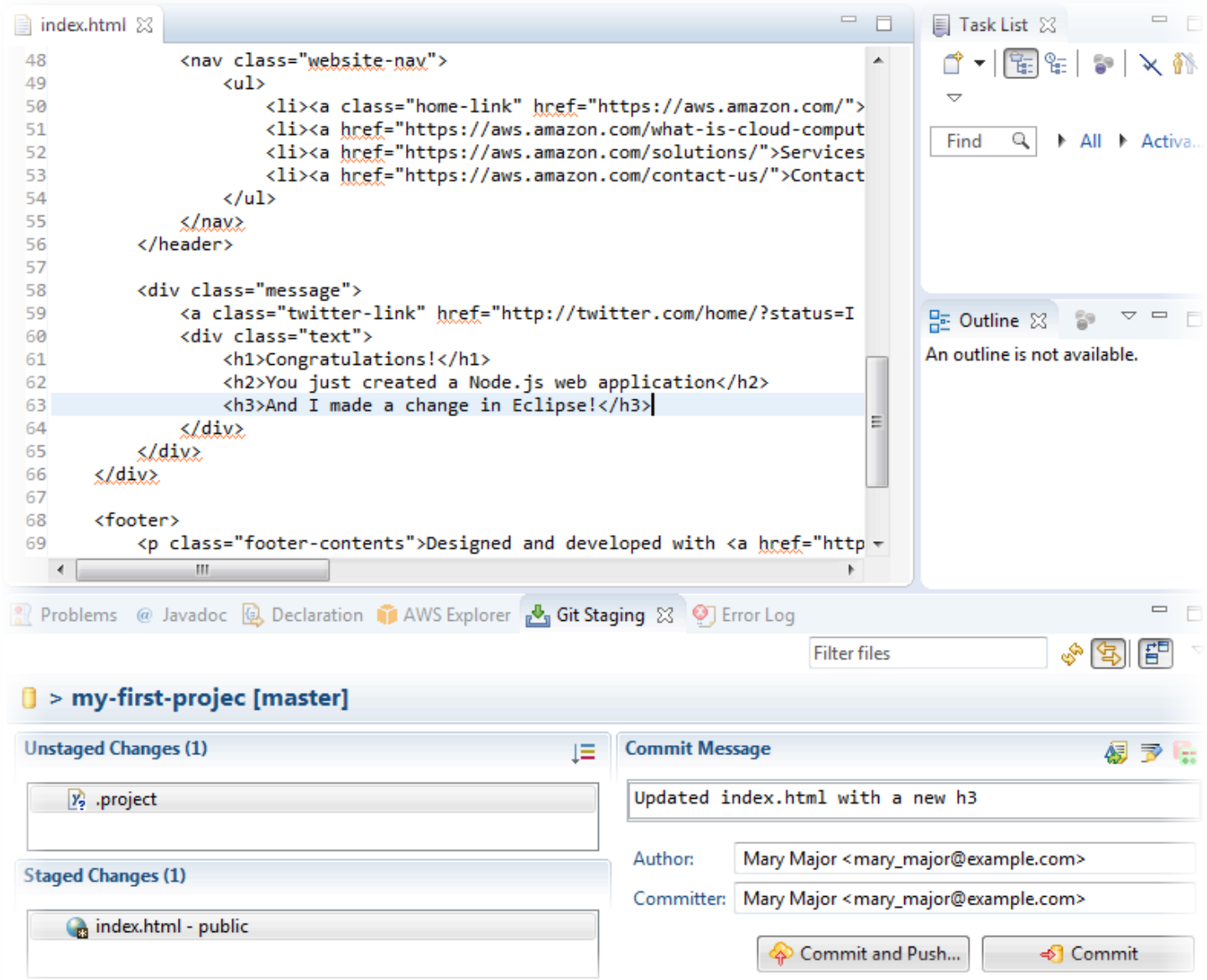
```
{
    "name": "MyProject",
    "id": "myproject",
    "arn": "arn:aws:codestar:us-east-1:account_ID:project/myproject",
    "description": "",
    "createdTimeStamp": 1539700079.472,
    "stackId": "arn:aws:cloudformation:us-east-1:account_ID:stack/awscodestar-
myproject/stack-ID",
    "status": {
        "state": "CreateInProgress"
    }
}
```

- 출력에는 새 프로젝트에 대한 정보가 들어 있습니다.
  - `state` 값은 프로젝트 생성 상태(`CreateInProgress` 또는 `CreateComplete` 등)를 나타냅니다.

프로젝트를 만드는 중에 팀원을 추가하거나 명령줄 또는 선호하는 IDE에서 프로젝트 리포지토리에 대한 액세스를 구성할 수 있습니다.

IDE를 AWS CodeStar와 함께 사용

IDE를 AWS CodeStar와 통합하면 원하는 환경에서 계속 코드를 작성하고 개발할 수 있습니다. 변경 사항은 코드를 커밋하고 푸시할 때마다 AWS CodeStar 프로젝트에 포함됩니다.



주제

- AWS CodeStar와 함께 AWS Cloud9 사용
- Eclipse를 AWS CodeStar와 함께 사용
- AWS CodeStar를 Visual Studio와 함께 사용

AWS CodeStar와 함께 AWS Cloud9 사용

AWS Cloud9을 사용하여 AWS CodeStar 프로젝트에서 코드를 변경하고 소프트웨어를 개발할 수 있습니다. AWS Cloud9은 웹 브라우저로 액세스할 수 있는 온라인 IDE입니다. IDE는 여러 프로그래밍 언어와 런타임 디버거 및 터미널을 갖춘 강력한 코드 편집 환경을 제공합니다. 배경에서는 Amazon EC2 인

스턴스가 AWS Cloud9 개발 환경을 호스팅합니다. 이 환경은 AWS Cloud9와 AWS CodeStar 프로젝트의 코드 파일에 대한 액세스를 제공합니다. 자세한 정보는 AWS Cloud9 사용 설명서를 참조하세요.

AWS CodeStar 콘솔이나 AWS Cloud9 콘솔을 이용해 코드를 CodeCommit에 저장하는 프로젝트를 위한 AWS Cloud9 개발 환경을 생성할 수 있습니다. 코드를 GitHub에 저장하는 AWS CodeStar 프로젝트의 경우, AWS Cloud9 콘솔만 사용할 수 있습니다. 이번 주제는 두 콘솔을 사용하는 방법을 설명합니다.

AWS Cloud9을 사용하려면 다음이 필요합니다.

- AWS CodeStar 프로젝트에 팀원으로 추가된 IAM 사용자
- AWS CodeStar 프로젝트가 소스 코드를 CodeCommit에 저장한다면, IAM 사용자에 대한 AWS 자격 증명

주제

프로젝트에 대한 AWS Cloud9 환경 생성

이러한 단계에 따라 AWS CodeStar 프로젝트를 위한 AWS Cloud9 개발 환경을 생성합니다.

1. 새 프로젝트를 생성하고자 할 경우 프로젝트 만들기에 나와 있는 단계를 따르세요.
2. AWS CodeStar 콘솔에서 프로젝트를 엽니다. 탐색 모음에서 IDE를 선택합니다. 환경 생성을 선택하고 다음 단계를 따릅니다.

> **Important**
>
> 프로젝트가 AWS Cloud9이 지원되지 않는 AWS 리전에 있다면, 탐색 모음에 IDE 탭의 AWS Cloud9 옵션이 표시되지 않을 것입니다. 하지만 AWS Cloud9 콘솔을 이용해 개발 환경을 만들고, 새 환경을 연 다음 이를 프로젝트의 AWS CodeCommit 리포지토리에 연결하는 방법이 있습니다. 다음 단계를 건너뛰고 AWS Cloud9사용자 안내서의 환경 만들기,

환경 열기 및 AWS CodeCommit샘플을 참조하세요. 지원되는 AWS 리전 목록은 Amazon Web Services 일반 참조의 AWS Cloud9 섹션을 참조하세요.

AWS Cloud9 환경 생성에서 프로젝트 기본값을 사용자 지정합니다.

1. 환경을 호스팅할 Amazon EC2 인스턴스의 기본 유형을 변경하려면 인스턴스 유형에서 인스턴스 유형을 선택합니다.
2. AWS Cloud9은 AWS 계정에서 Amazon VPC(Amazon Virtual Private Cloud)를 사용하여 인스턴스와 통신합니다. AWS 계정에서 Amazon VPC가 어떻게 설정되었는가에 따라 다음 중 하나를 수행하세요.

계정에 VPC가 있으며 해당 VPC에 서브넷이 하나 이상 있습니까?	해당 VPC가 AWS Cloud9가 계정에서 기본 VPC로 사용하게 할 VPC입니까?	VPC에 단일 서브넷이 있습니까?	조치
아니요	—	—	VPC가 없다면, 하나 만드십시오. 네트워크 설정을 확장합니다. 네트워크(VPC)에서 VPC 만들기를 선택하고 페이지의 지침을 따릅니다. 자세한 내용은 AWS Cloud9 사용 설명서의 AWS Cloud9에 대한 Amazon VPC 생성을 참조하세요. VPC가 있지만 서브넷이 없다면, 하나 만드십시오. 네트워크 설정을 확장합니다. 네트워크(VPC)에서 서브넷 생성을 선택하고 지침을 따릅니다. 자세한 내용은 AWS Cloud9 사용 설명서의 AWS Cloud9에 대한 서브넷 생성을 참조하세요.
예	예	예	이 절차의 4단계로 건너뛰십시오. (AWS Cloud9은 기본 VPC와 단일 서브넷을 이용합니다.)

계정에 VPC가 있으며 해당 VPC에 서브넷이 하나 이상 있습니까?	해당 VPC가 AWS Cloud9가 계정에서 기본 VPC로 사용하게 할 VPC입니까?	VPC에 단일 서브넷이 있습니까?	조치
예	예	아니요	서브넷에서 AWS Cloud9가 사전 선택된 기본 VPC에서 사용할 서브넷을 선택합니다.
예	아니요	[Yes] 또는 [No]	Network(VPC)에서 AWS Cloud9이 사용할 VPC를 선택합니다. 서브넷에는 해당 VPC에서 AWS Cloud9이 사용할 서브넷을 선택합니다.

자세한 내용은 AWS Cloud9 사용 설명서의 AWS Cloud9 개발 환경을 위한 Amazon VPC 설정을 참조하세요.

3. 환경 이름을 입력하고 선택적으로 환경 설명을 추가합니다.

> **Note**
>
> 환경 이름은 사용자별로 고유해야 합니다.

4. AWS Cloud9을 사용하지 않을 때 환경을 닫는 기본 시간을 변경하고 싶다면, 비용 절감 설정을 확장하고 설정을 변경합니다.
5. 환경 생성을 선택합니다.

환경을 여는 방법은 프로젝트에 대한 AWS Cloud9 환경 열기 단원을 참조하십시오.

이러한 단계를 이용하면 한 프로젝트에 하나 이상의 환경을 생성할 수 있습니다. 예를 들어 환경 하나는 코드의 일정 부분에 작동하며, 다른 환경은 코드의 같은 부분에 다른 설정을 적용하게 할 수 있습니다.

프로젝트에 대한 AWS Cloud9 환경 열기

이러한 단계에 따라 AWS CodeStar 프로젝트에 대해 생성한 AWS Cloud9 개발 환경을 엽니다.

1. AWS CodeStar 콘솔에서 프로젝트를 열어 둔 상태로, 탐색 모음에서 IDE를 선택합니다.

   > **Important**
   >
   > 프로젝트의 소스 코드가 GitHub에 저장돼 있다면, 탐색 모음에 IDE가 표시되지 않을 것입니다. AWS Cloud9 콘솔을 이용해 기존 환경을 열 수도 있습니다. 이 절차의 나머지 부분은 건너뛰고 AWS Cloud9 사용 설명서 및 GitHub를 AWS Cloud9과 함께 사용의 환경 열기를 참조하세요.

2. 당신의 AWS Cloud9 환경 또는 공유 AWS Cloud9 환경에서 열고 싶은 환경의 IDE 열기를 선택합니다.

AWS Cloud9 IDE가 프로젝트의 AWS CodeCommit 리포지토리에 있는 코드로 바로 시작하게 할 수도 있습니다. 자세한 내용은 AWS Cloud9 사용 안내서의 환경 창, 편집기, 탭 및 창, 터미널 및 AWS CodeCommit 사용 설명서의 기본 Git 명령을 참조하세요.

프로젝트 팀원과 AWS Cloud9 환경 공유

AWS CodeStar 프로젝트에 대한 AWS Cloud9 개발 환경을 생성하면, 프로젝트 팀원을 포함한 AWS 계정에 있는 다른 사용자를 초대해 같은 환경에 액세스하게 할 수 있습니다. 이 기능은 프로그래머 두 명이 화면을 공유하거나 같은 워크스테이션에 앉아 번갈아 가며 코딩하면서 조언을 나누는 페어 프로그래밍을 할 때 특히 유용합니다. 환경 회원은 공유 AWS Cloud9 IDE를 이용해 코드 편집기에서 강조 표시된 각 회원의 코드 변경사항을 확인하고 코딩 중에 다른 회원과 문자 채팅을 할 수 있습니다.

팀원을 프로젝트에 추가해도 프로젝트의 모든 관련 AWS Cloud9 개발 환경에 대한 구성원의 참여를 자동으로 허용하지는 않습니다. 프로젝트 팀원을 프로젝트의 환경에 액세스하도록 초대하려면, 올바른 환경 회원 액세스 역할을 결정하고, AWS 관리형 정책을 사용자에게 적용하고, 사용자를 자신의 환경에 초대해야 합니다. 자세한 내용은 AWS Cloud9 사용 설명서의 환경 구성원 액세스 역할 정보 및 IAM 사용자를 환경에 초대를 참조하세요.

프로젝트 팀원을 프로젝트의 환경에 액세스하도록 초대하면, AWS CodeStar 콘솔은 해당 팀원에게 환경을 표시합니다. 환경은 프로젝트 AWS CodeStar 콘솔의 IDE 탭에 있는 공유 환경 목록에 표시됩니다. 이 목록을 표시하려면, 팀원이 콘솔에서 프로젝트를 열고 탐색 모음에서 IDE를 선택하게 해야 합니다.

> **Important**
>
> 프로젝트의 소스 코드가 GitHub에 저장돼 있다면, 탐색 모음에 IDE가 표시되지 않을 것입니다. 하지만 AWS Cloud9 콘솔을 이용해 프로젝트 팀원을 포함한 다른 사용자를 AWS 계정에 초대

해 환경에 액세스하게 할 수도 있습니다. 이를 수행하려면 이 안내서의 GitHub를 AWS Cloud9과 함께 사용 내용을 참조하고, AWS Cloud9 사용 설명서의 환경 구성원 액세스 역할 정보 및 IAM 사용자를 사용자 환경에 초대를 참조하세요.

프로젝트 팀원이 아닌 사용자도 환경에 액세스하도록 초대할 수 있습니다. 예를 들어 사용자가 프로젝트의 코드는 이용할 수 있지만 프로젝트의 다른 부분은 액세스하지 못하게 할 수도 있습니다. 자세한 내용은 AWS Cloud9 사용 설명서의 환경 구성원 액세스 역할 정보 및 IAM 사용자를 환경에 초대를 참조하세요. 프로젝트 팀원이 아닌 사용자를 프로젝트의 환경에 액세스하도록 초대하면, 해당 사용자는 AWS Cloud9 콘솔을 이용해 환경에 액세스합니다. 자세한 내용을 알아보려면 AWS Cloud9 사용 설명서의 환경 열기를 참조하세요.

프로젝트에서 AWS Cloud9 환경 삭제

AWS CodeStar에서 프로젝트와 프로젝트의 모든 AWS 리소스를 삭제하면, AWS CodeStar 콘솔로 생성한 모든 관련 AWS Cloud9 개발 환경도 삭제되며 복구할 수 없게 됩니다. 프로젝트를 삭제하지 않고 개발 환경만 프로젝트에서 삭제할 수도 있습니다.

1. AWS CodeStar 콘솔에서 프로젝트를 열어 둔 상태로, 탐색 모음에서 IDE를 선택합니다.

> **Important**
>
> 프로젝트의 소스 코드가 GitHub에 저장돼 있다면, 탐색 모음에 IDE가 표시되지 않을 것입니다. 하지만 AWS Cloud9 콘솔을 이용해 개발 환경을 삭제할 수도 있습니다. 이 절차의 나머지 부분은 건너뛰고 AWS Cloud9 사용 설명서의 환경 삭제를 참조하세요.

2. Cloud9 환경 환경에서 삭제할 환경을 선택하고 삭제를 선택합니다.
3. **delete**를 입력하여 개발 환경의 삭제를 확인한 다음, 삭제를 선택합니다.

> **Warning**
>
> 삭제한 개발 환경은 복구할 수 없습니다. 커밋하지 않은 환경 내 모든 코드 변경 사항은 취소됩니다.

GitHub를 AWS Cloud9과 함께 사용

소스 코드를 GitHub에 저장한 AWS CodeStar 프로젝트의 경우, AWS CodeStar 콘솔은 AWS Cloud9 개발 환경과의 직접 작업을 지원하지 않습니다. 하지만 AWS Cloud9 콘솔을 사용하여 GitHub 리포지토리의 소스 코드를 작업할 수는 있습니다.

1. AWS Cloud9 콘솔을 사용하여 AWS Cloud9 개발 환경을 생성합니다. 자세한 내용을 알아보려면 AWS Cloud9 사용 설명서의 환경 생성을 참조하세요.
2. AWS Cloud9 콘솔을 사용하여 개발 환경을 엽니다. 자세한 내용을 알아보려면 AWS Cloud9 사용 설명서의 환경 열기를 참조하세요.
3. IDE에서 터미널 세션을 이용해 GitHub 리포지토리에 연결합니다(복제라고 하는 과정입니다). 터미널 세션이 작동하지 않는다면, IDE의 메뉴 모음에서 창, 새 터미널을 선택합니다. GitHub 리포지토리 복제에 사용하는 명령은 GitHub Help 웹사이트의 리포지토리 복제 단원에서 확인할 수 있습니다.

   GitHub 리포지토리의 기본 페이지를 탐색하려면, AWS CodeStar 콘솔에서 프로젝트를 열어 둔 상태로 측면 탐색 모음에서 코드를 선택합니다.
4. IDE의 환경 창과 편집기 탭을 이용해 코드를 보고, 변경하고, 저장합니다. 자세한 내용은 AWS Cloud9 사용 설명서의 환경 창과 편집기, 탭 및 창을 참조하세요.
5. IDE 터미널 세션의 Git를 사용하여 코드 변경사항을 GitHub 리포지토리로 푸시하고 정기적으로 리포지토리에서 다른 사용자의 코드 변경사항을 가져옵니다. 자세한 내용은 GitHub Help 웹 사이트에서 원격 리포지토리로 푸시하기 및 원격 리포지토리 가져오기를 참조하십시오. Git 명령은 GitHub Help 웹사이트의 Git 치트 시트에서 확인할 수 있습니다.

   > **Note**
   >
   > 리포지토리로 코드를 푸시하거나 리포지토리에서 코드를 가져올 때마다 Git에서 GitHub 로그인 자격 증명을 요청하도록 하려면 보안 인증 도우미를 사용하면 됩니다. 자세한 내용은 GitHub 웹 사이트의 GitHub 암호를 Git에 저장 단원을 참조하십시오.

추가 리소스

AWS Cloud9 사용에 대한 자세한 내용은 AWS Cloud9 사용 설명서의 다음 항목을 참조하십시오.

- 자습서
- 환경과의 작업

- IDE 작업
- 샘플

Eclipse를 AWS CodeStar와 함께 사용

Eclipse를 사용하여 AWS CodeStar 프로젝트에서 코드를 변경하고 소프트웨어를 개발할 수 있습니다. Eclipse를 사용하여 AWS CodeStar 프로젝트 코드를 편집한 다음, 변경 사항을 커밋하고 AWS CodeStar 프로젝트의 소스 리포지토리에 푸시할 수 있습니다.

> **Note**
>
> 이 주제의 정보는 소스 코드를 CodeCommit에 저장하는 AWS CodeStar 프로젝트에만 적용됩니다. AWS CodeStar 프로젝트가 소스 코드를 GitHub에 저장하는 경우 Eclipse용 EGit와 같은 도구를 사용할 수 있습니다. 자세한 내용은 EGit 웹 사이트에 있는 EGit 설명서를 참조하십시오.

AWS CodeStar 프로젝트가 CodeCommit에 소스 코드를 저장한다면, AWS CodeStar를 지원하는 AWS Toolkit for Eclipse 버전을 설치해야 합니다. 또한 소유자 또는 기고자 역할이 있는 AWS CodeStar 프로젝트 팀원이어야 합니다.

Eclipse를 사용하려면 다음 항목도 필요합니다.

- AWS CodeStar 프로젝트에 팀원으로 추가된 IAM 사용자입니다.
- AWS CodeStar 프로젝트가 소스 코드를 CodeCommit에 저장한다면, IAM 사용자에 대한 Git 자격 증명(로그인 자격 증명)입니다.
- 로컬 컴퓨터에 Eclipse와 AWS Toolkit for Eclipse를 설치할 권한

주제

- 1단계: AWS Toolkit for Eclipse 설치
- 2단계: AWS CodeStar 프로젝트를 Eclipse로 가져오기
- 3단계: Eclipse에서 AWS CodeStar 프로젝트 코드 편집

1단계: AWS Toolkit for Eclipse 설치

Toolkit for Eclipse는 Eclipse에 추가할 수 있는 소프트웨어 패키지입니다. Eclipse의 다른 소프트웨어 패키지와 동일한 방법으로 설치 및 관리합니다. AWS CodeStar 도구 키트는 Toolkit for Eclipse의 일부로 포함됩니다.

Toolkit for Eclipse를 AWS CodeStar 모듈과 함께 설치하는 방법

1. 로컬 컴퓨터에 Eclipse를 설치합니다. 지원되는 Eclipse 버전은 Luna, Mars, Neon 등입니다.
2. Toolkit for Eclipse를 다운로드하고 설치합니다. 자세한 내용은 AWS Toolkit for Eclipse 시작 안내서를 참조하세요.
3. Eclipse에서 [Help]를 선택한 다음 [Install New Software]를 선택합니다.
4. [Available Software]에서 [Add]를 선택합니다.
5. [Add Repository]에서 [Archive]를 선택하고, .zip 파일을 저장한 위치로 이동한 다음, 해당 파일을 엽니다. [Name]을 비워 두고 [OK]를 선택합니다.
6. 사용 가능한 소프트웨어에서 모두 선택을 선택하여 AWS 핵심 관리 도구 및 개발자 도구 선택한 후 다음을 선택합니다.
7. [Install Details]에서 [Next]를 선택합니다.
8. [Review Licenses]에서 라이선스 계약을 검토합니다. 라이선스 계약 약관에 동의합니다를 선택하고 완료를 선택합니다. Eclipse를 다시 시작합니다.

2단계: AWS CodeStar 프로젝트를 Eclipse로 가져오기

Toolkit for Eclipse를 설치한 후 AWS CodeStar 프로젝트를 가져오고 IDE에서 코드를 편집, 커밋 및 푸시할 수 있습니다.

> **Note**
>
> Eclipse에서 하나의 작업 영역에 AWS CodeStar 프로젝트를 여러 개 추가할 수 있지만, 한 프로젝트에서 다른 프로젝트로 변경할 때 프로젝트 자격 증명을 업데이트해야 합니다.

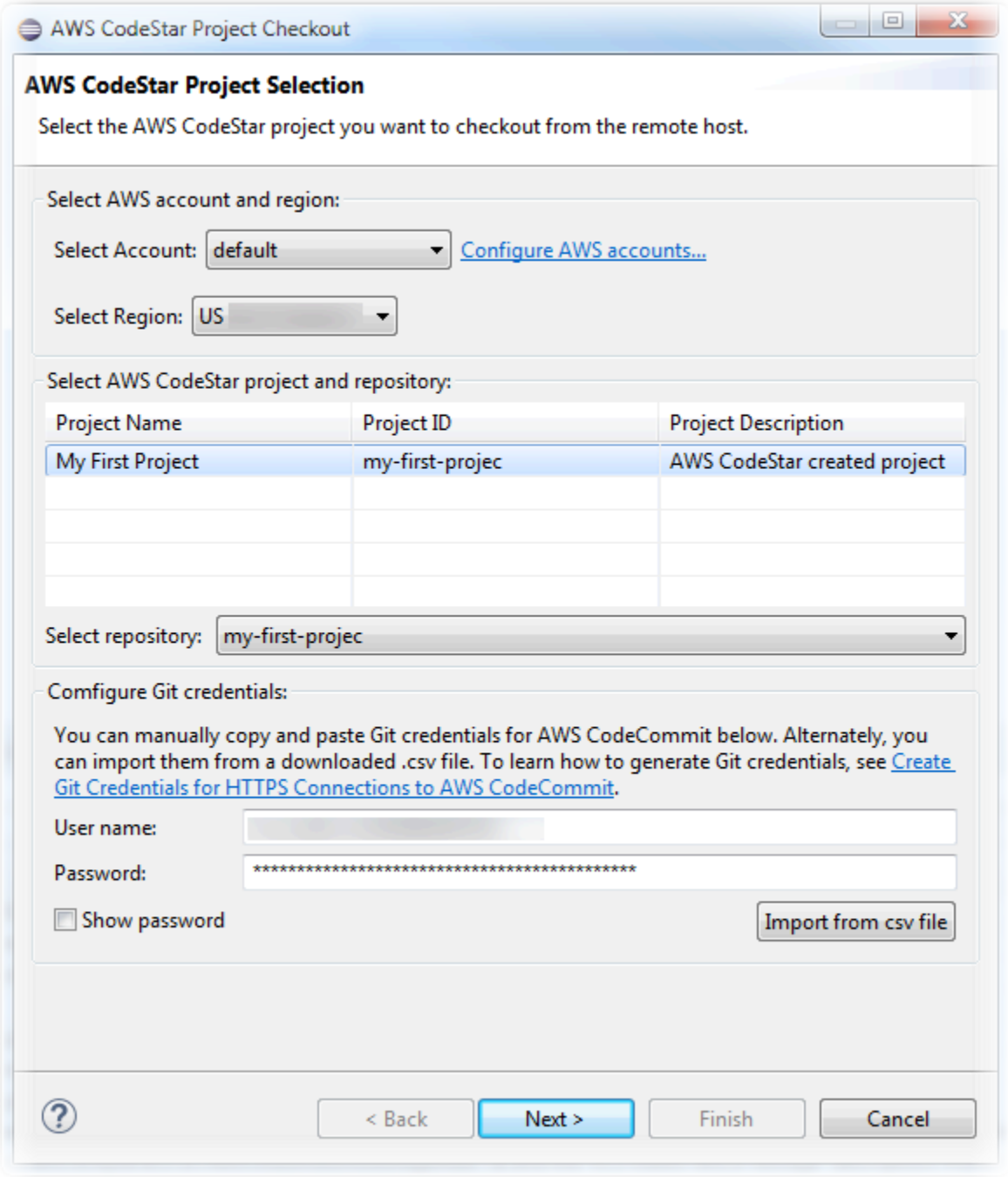
AWS CodeStar 프로젝트를 가져오려면

1. AWS 메뉴에서 AWS CodeStar 프로젝트 가져오기를 선택합니다. 또는 [File]을 선택한 다음 [Import]를 선택합니다. 선택에서 AWS를 확장하고 AWS CodeStar 프로젝트를 선택합니다.

다음을 선택합니다.

2. AWS CodeStar 프로젝트 선택에서 AWS 프로필과 AWS CodeStar 프로젝트를 호스트하는 AWS 리전을 선택합니다. 컴퓨터에서 액세스 키 및 보안 키로 AWS 프로필을 구성하지 않았다면, AWS 계정 구성을 선택하고 지침을 따릅니다.

   AWS CodeStar 프로젝트 및 리포지토리 선택에서 AWS CodeStar 프로젝트를 선택합니다. Git 자격 증명 구성에 프로젝트의 리포지토리에 액세스하기 위해 생성한 로그인 자격 증명을 입력합니다. (Git 자격 증명이 없는 경우 시작하기를 참조하십시오.) 다음을 선택합니다.



3. 프로젝트 리포지토리의 모든 브랜치는 기본적으로 선택되어 있습니다. 하나 이상의 브랜치를 가져오지 않으려면 해당 상자의 선택을 취소한 다음 [Next]를 선택합니다.
4. 로컬 대상에서 가져오기 마법사가 컴퓨터에서 로컬 리포지토리를 만들 대상을 선택한 다음, 완료를 선택합니다.
5. Project Explorer에서 프로젝트 트리를 확장하여 AWS CodeStar 프로젝트의 파일을 찾아봅니다.

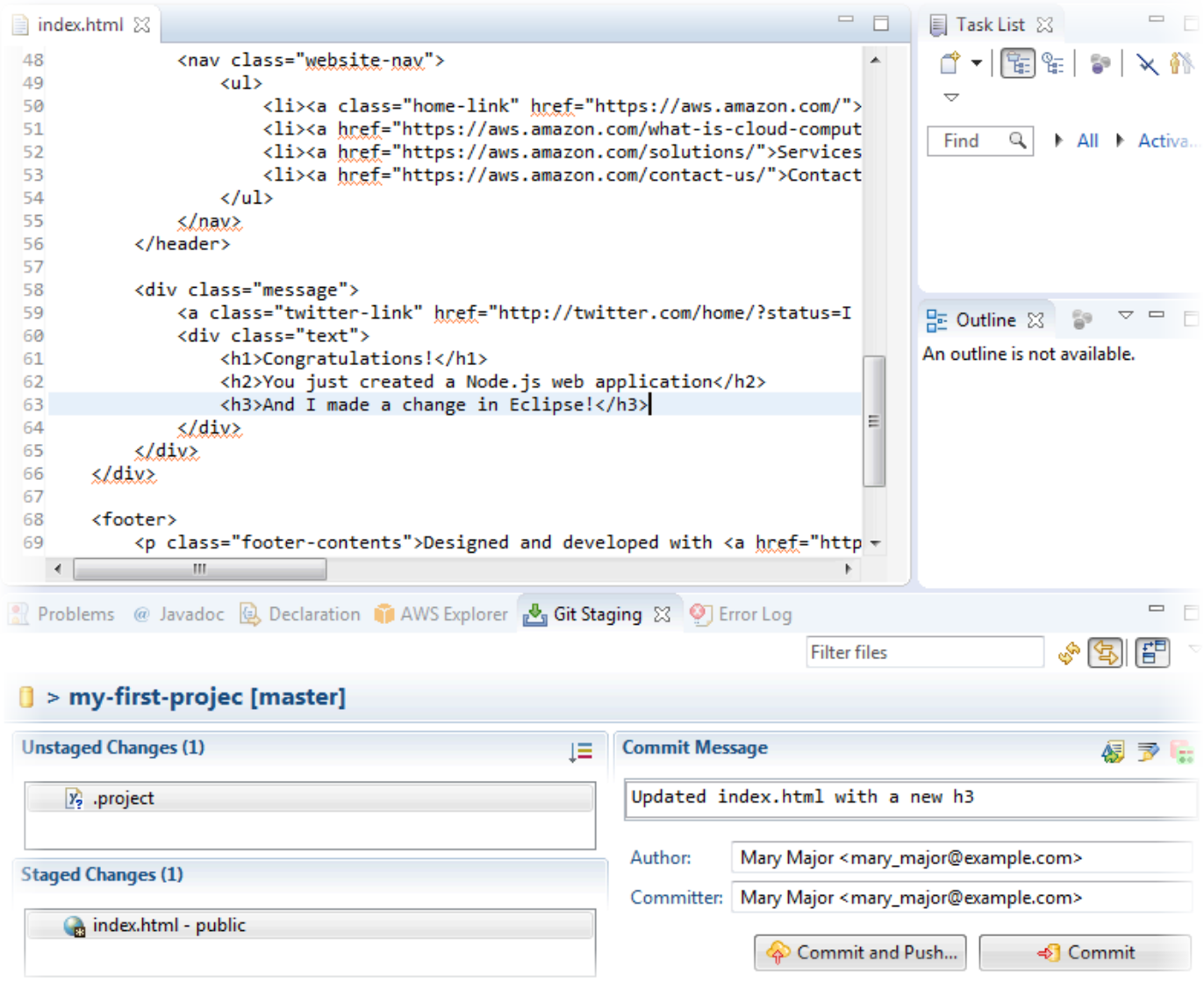
3단계: Eclipse에서 AWS CodeStar 프로젝트 코드 편집

AWS CodeStar 프로젝트를 Eclipse 작업 영역으로 가져온 후 프로젝트의 코드를 편집하고, 변경 사항을 저장한 후, 코드를 커밋하고 프로젝트의 소스 리포지토리에 푸시할 수 있습니다. Eclipse용 EGit 플러그인을 사용하는 모든 Git 리포지토리의 경우에도 동일한 프로세스를 따릅니다. 자세한 내용은 Eclipse 웹 사이트에 있는 EGit 사용 설명서를 참조하십시오.

프로젝트 코드를 편집하고 AWS CodeStar 프로젝트의 소스 리포지토리에 첫 커밋을 하려면

1. Project Explorer에서 프로젝트 트리를 확장하여 AWS CodeStar 프로젝트의 파일을 찾아봅니다.
2. 하나 이상의 코드 파일을 편집하고 변경 사항을 저장합니다.
3. 변경 사항을 커밋할 준비가 되면 해당 파일의 컨텍스트 메뉴를 열고, [Team]을 선택한 다음, [Commit]을 선택합니다.

   Git 스테이징 창을 프로젝트 보기에서 이미 열었다면 이 단계를 건너뛰어도 됩니다.
4. Git 스테이징에서 변경된 파일을 스테이징된 변경 사항으로 이동하여 변경 사항을 스테이징합니다. 커밋 메시지에 커밋 메시지를 입력한 다음, 커밋 및 푸시를 선택합니다.



코드 변경 사항의 배포를 보려면 프로젝트의 대시보드로 돌아갑니다. 자세한 내용은 3단계: 프로젝트 보기 섹션을 참조하세요.

AWS CodeStar를 Visual Studio와 함께 사용

Visual Studio를 사용하여 AWS CodeStar 프로젝트에서 코드를 변경하고 소프트웨어를 개발할 수 있습니다.

> **Note**
>
> Visual Studio for Mac은 AWS 툴킷을 지원하지 않으므로 AWS CodeStar와 함께 사용할 수 없습니다.

이 주제의 정보는 소스 코드를 CodeCommit에 저장하는 AWS CodeStar 프로젝트에만 적용됩니다. AWS CodeStar 프로젝트가 소스 코드를 GitHub에 저장하는 경우 Visual Studio용 GitHub Extension과 같은 도구를 사용할 수 있습니다. 자세한 내용은 Visual Studio용 GitHub 웹 사이트의 개요 페이지 및 GitHub 웹 사이트의 Getting Started with GitHub for Visual Studio를 참조하십시오.

Visual Studio를 사용하여 AWS CodeStar 프로젝트에 대한 소스 리포지토리의 코드를 편집하려면 AWS CodeStar를 지원하는 AWS Toolkit for Visual Studio 버전을 설치해야 합니다. 소유자 또는 기고자 역할이 있는 AWS CodeStar 프로젝트 팀원이어야 합니다.

Visual Studio를 사용하려면 다음 항목도 필요합니다.

- AWS CodeStar 프로젝트에 팀원으로 추가된 IAM 사용자입니다.
- IAM 사용자의 AWS 자격 증명(예: 액세스 키와 보안 키)
- 로컬 컴퓨터에 Visual Studio와 AWS Toolkit for Visual Studio를 설치할 권한

Toolkit for Visual Studio는 Visual Studio에 추가할 수 있는 소프트웨어 패키지입니다. Visual Studio의 다른 소프트웨어 패키지와 동일한 방법으로 설치 및 관리합니다.

AWS CodeStar 모듈로 Toolkit for Visual Studio를 설치하고 프로젝트 리포지토리에 대한 액세스를 구성하려면

1. 로컬 컴퓨터에 Visual Studio를 설치합니다.
2. Toolkit for Visual Studio를 다운로드 및 설치하고 .zip 파일을 로컬 폴더 또는 디렉터리에 저장합니다. AWS Toolkit for Visual Studio 시작하기 페이지에 AWS 자격 증명을 입력하거나 가져온 후, 저장하고 닫기를 선택합니다.
3. Visual Studio에서 Team Explorer를 엽니다. 호스팅된 서비스 공급자에서 CodeCommit를 찾은 다음 연결을 선택합니다.
4. 연결 관리에서 복제를 선택합니다. 프로젝트의 리포지토리와 리포지토리를 복제할 로컬 컴퓨터의 폴더를 선택한 다음, 확인을 선택합니다.
5. Git 자격 증명을 만들라는 메시지가 나타나면 예를 선택합니다. 사용자를 대신하여 도구 키트에서 자격 증명을 만들려고 시도합니다. 자격 증명 파일을 안전한 위치에 저장합니다. 이 자격 증명을 저장하려면 지금 해야 합니다. 나중에는 불가능합니다. 도구 키트가 사용자를 대신하여 자격 증명을 만들 수 없거나 [No]를 선택한 경우, 자체 Git 자격 증명을 만들고 입력해야 합니다. 자세한 내용

은 변경을 커밋하도록 컴퓨터를 설정하려면(IAM 사용자)을 참조하거나 온라인 지침을 따르십시오.

프로젝트 복제를 종료하면, Visual Studio에서 코드를 편집하고 CodeCommit에서 변경 사항을 프로젝트의 리포지토리에 커밋, 푸시할 준비가 됩니다.

AWS CodeStar 프로젝트에서 AWS 리소스 변경

AWS CodeStar에서 프로젝트를 생성한 후 AWS CodeStar에서 프로젝트에 추가하는 기본 AWS 리소스를 변경할 수 있습니다.

지원되는 리소스 변경

다음 표에는 AWS CodeStar 프로젝트의 기본 AWS 리소스에 대한 지원되는 변경 사항이 나열되어 있습니다.

변경 사항	주의
단계를 AWS CodePipeline에 추가합니다.	단계를 AWS CodePipeline에 추가합니다. 섹션을 참조하세요.
Elastic Beanstalk 환경 설정을 변경합니다.	AWS Elastic Beanstalk 환경 설정을 변경합니다. 섹션을 참조하세요.
API Gateway에서 AWS Lambda 함수의 코드 또는 설정, IAM 역할 또는 API를 변경합니다.	소스 코드의 AWS Lambda 함수를 변경합니다. 섹션을 참조하세요.
AWS Lambda 프로젝트에 리소스를 추가하고 권한을 확장해 새 리소스를 생성 및 액세스합니다.	프로젝트에 리소스 추가 섹션을 참조하세요.
AWS Lambda 함수에 CodeDeploy를 이용하는 트래픽 이동을 추가합니다.	AWS Lambda 프로젝트의 트래픽 이동 섹션을 참조하세요.
AWS X-Ray 지원 추가	프로젝트에 트레이스 활성화 섹션을 참조하세요.

변경 사항	주의
프로젝트의 buildspec.yml 파일을 편집해 AWS CodeBuild가 실행할 단위 테스트 빌드 단계를 추가합니다.	서버리스 프로젝트 자습서의 7단계: 웹 서비스에 단위 테스트 추가를 참조하십시오.
프로젝트에 고유한 IAM 역할을 추가합니다.	프로젝트에 IAM 역할 추가 섹션을 참조하세요.
IAM 역할 정의를 변경합니다.	애플리케이션 스택에 정의된 역할에 해당합니다. 도구 체인 또는 AWS CloudFormation 스택에 정의된 역할은 변경할 수 없습니다.
엔드포인트를 추가하도록 Lambda 프로젝트를 수정합니다.	
엔드포인트를 추가하도록 EC2 프로젝트를 수정합니다.	
엔드포인트를 추가하도록 Elastic Beanstalk 프로젝트를 수정합니다.	
Prod 단계 및 엔드포인트를 추가하도록 프로젝트를 편집합니다.	프로젝트에 Prod 단계 및 엔드포인트 추가 섹션을 참조하세요.
AWS CodeStar 프로젝트에서 SSM 파라미터를 안전하게 사용합니다.	the section called “프로젝트에서 SSM 파라미터를 안전하게 사용 AWS CodeStar” 섹션을 참조하세요.

다음 변경은 지원되지 않습니다.

- 다른 배포 대상으로 전환(예: AWS CodeDeploy 대신 AWS Elastic Beanstalk에 배포)
- 기억하기 쉬운 웹 엔드포인트 이름을 추가합니다.
- CodeCommit 리포지토리 이름을 변경합니다(CodeCommit에 연결된 AWS CodeStar 프로젝트의 경우).
- GitHub 리포지토리에 연결된 AWS CodeStar 프로젝트의 경우, GitHub 리포지토리를 연결 해제한 다음 리포지토리를 해당 프로젝트에 다시 연결하거나 다른 리포지토리를 해당 프로젝트에 연결합니다. AWS CodeStar 콘솔이 아닌 CodePipeline 콘솔을 사용하여 파이프라인의 소스 단계에서 GitHub를 연결 해제했다가 다시 연결할 수 있습니다. 하지만 소스 단계를 다른 GitHub 리포지토리에 다시

연결한 다음 프로젝트의 AWS CodeStar 대시보드에 다시 연결하면 리포지토리 및 문제 타일의 정보는 잘못되거나 오래된 것일 수 있습니다. GitHub 리포지토리를 연결 해제해도 해당 리포지토리의 정보는 커밋 기록에서 제거되지 않으며 GitHub는 AWS CodeStar 프로젝트 대시보드에서 타일을 발급합니다. 이 정보를 제거하려면 GitHub 웹 사이트를 사용하여 AWS CodeStar 프로젝트에서 GitHub에 대한 액세스를 비활성화합니다. 액세스 권한을 취소하려면 GitHub 웹 사이트에서 GitHub 계정 프로필에 대한 설정 페이지의 Authorized OAuth Apps(권한 있는 OAuth 앱) 섹션을 사용합니다.

- CodeCommit 리포지토리(CodePipeline에 연결된 AWS CodeStar 프로젝트의 경우)를 연결 해제한 다음 리포지토리를 해당 프로젝트에 다시 연결하거나 다른 리포지토리를 해당 프로젝트에 연결합니다.

단계를 AWS CodePipeline에 추가합니다.

AWS CodeStar에서 프로젝트에 생성하는 파이프라인에 새 단계를 추가할 수 있습니다. 자세한 내용을 알아보려면 AWS CodePipeline 사용 설명서의 Edit a Pipeline in AWS CodePipeline을 참조하세요.

> **Note**
>
> 새 단계가 AWS CodeStar에서 생성되지 않은 AWS 리소스에 종속되는 경우 파이프라인이 중단될 수 있습니다. AWS CodeStar에서 AWS CodePipeline에 대해 생성된 IAM 역할은 기본적으로 이러한 리소스에 대한 액세스 권한이 없기 때문입니다.
>
> AWS CodeStar에서 생성되지 않은 AWS 리소스에 대한 AWS CodePipeline 액세스 권한을 부여하려면 AWS CodeStar에서 생성된 IAM 역할을 변경할 수 있습니다. 이 기능은 지원되지 않는데, 프로젝트에 대한 정기 업데이트 검사를 수행할 때 AWS CodeStar에서 IAM 역할 변경을 제거할 수 있기 때문입니다.

AWS Elastic Beanstalk 환경 설정을 변경합니다.

AWS CodeStar가 프로젝트에서 생성하는 Elastic Beanstalk 환경 설정을 변경할 수 있습니다. 예를 들어 AWS CodeStar 프로젝트의 기본 Elastic Beanstalk 환경을 단일 인스턴스에서 로드 밸런서로 변경할 수 있습니다. 이렇게 하려면 프로젝트 리포지토리에서 `template.yml` 파일을 편집합니다. 프로젝트의 작업자 역할에 대한 권한을 변경해야 할 수도 있습니다. 템플릿 변경을 푸시한 후 AWS CodeStar 및 AWS CloudFormation이 리소스를 프로비저닝합니다.

`template.yml` 파일을 편집하는 방법에 대한 자세한 내용은 Template.yml 파일로 애플리케이션 리소스 변경 단원을 참조하십시오. Elastic Beanstalk 환경에 대한 자세한 내용은 AWS Elastic Beanstalk 개발자 안내서의 AWS Elastic Beanstalk 환경 관리 콘솔을 참조하십시오.

소스 코드의 AWS Lambda 함수를 변경합니다.

AWS CodeStar가 프로젝트에서 생성하는 Lambda 함수의 코드 또는 설정, IAM 역할 또는 API Gateway API를 변경할 수 있습니다. 이렇게 하려면 프로젝트의 CodeCommit 리포지토리에서 AWS 서버리스 애플리케이션 모델(AWS SAM)을 `template.yaml` 파일과 함께 사용하는 것이 좋습니다. 이 `template.yaml` 파일은 API Gateway의 함수 이름, 핸들러, 실행 시간, IAM 역할, API를 정의합니다. 자세한 내용은 GitHub 웹 사이트의 AWS SAM을 이용해 서버리스 애플리케이션을 만드는 방법 단원을 참조하십시오.

프로젝트에 트레이스 활성화

AWS X-Ray는 추적 기능을 제공하는데, 이를 이용하면 분산된 애플리케이션의 성능 동작(예: 응답 시간의 지연)을 분석할 수 있습니다. AWS CodeStar 프로젝트에 트레이스를 추가하면, AWS X-Ray 콘솔로 애플리케이션 보기와 응답 시간을 확인할 수 있습니다.

> **Note**
>
> 다음 프로젝트 지원 변경으로 생성된 다음 프로젝트에 이러한 단계를 사용할 수 있습니다.
>
> - 모든 Lambda 프로젝트
> - 2018년 8월 3일 이후에 만든 Amazon EC2 또는 Elastic Beanstalk 프로젝트의 경우, AWS CodeStar는 프로젝트 리포지토리에 `/template.yml` 파일을 프로비저닝했습니다.

모든 AWS CodeStar 템플릿에는 데이터베이스 테이블이나 Lambda 함수 같은 애플리케이션의 AWS 실행 시간 종속성을 보여주는 AWS CloudFormation 파일이 있습니다. 이 파일은 `/template.yml` 파일의 소스 리포지토리에 저장됩니다.

이 파일을 수정해 `Resources` 섹션에 AWS X-Ray 리소스를 추가하면 트레이스를 추가할 수 있습니다. 그런 다음 프로젝트에 대한 IAM 권한을 수정해 AWS CloudFormation이 리소스를 생성하게 할 수도 있습니다. 템플릿 요소와 형식에 대한 정보는 AWS 리소스 유형 참조를 참조하세요.

이것은 템플릿을 사용자 지정하려면 거쳐야 하는 상위 단계입니다.

1. 1단계: 추적을 위해 IAM의 작업자 역할 편집
2. 2단계: 추적을 위해 template.yml 파일 수정
3. 3단계: 추적을 위해 템플릿 변경사항을 커밋 및 푸시
4. 4단계: 추적을 위해 AWS CloudFormation 스택 업데이트 모니터링

1단계: 추적을 위해 IAM의 작업자 역할 편집

1단계와 4단계를 수행하려면 관리자로 로그인해야 합니다. 이 단계에서는 Lambda 프로젝트에 대한 권한을 편집하는 예를 보여줍니다.

> **Note**
>
> 프로젝트가 권한 경계 정책으로 프로비저닝된 경우 이 단계를 건너뛸 수 있습니다. 2018년 12월 6일(PDT) 이후에 만든 프로젝트의 경우, AWS CodeStar는 프로젝트를 권한 경계 정책으로 프로비저닝했습니다.

1. AWS Management Console에 로그인하고 https://console.aws.amazon.com/codestar/에서 AWS CodeStar 콘솔을 엽니다.
2. 프로젝트를 생성하거나 `template.yml file`이 있는 기존 프로젝트를 선택한 다음, 프로젝트 리소스 페이지를 엽니다.
3. 프로젝트 리소스의 리소스 목록에서 CodeStarWorker/Lambda 역할에 대해 생성된 IAM 역할을 찾습니다. 역할 이름은 다음 형식을 취합니다. `role/CodeStarWorker-`*`Project_name`*`-lambda-`*`Function_name`*. 역할에 대한 ARN을 선택합니다.
4. 역할이 IAM 콘솔에서 열립니다. 정책 연결을 선택합니다. `AWSXrayWriteOnlyAccess` 정책을 찾고, 옆에 있는 확인란을 선택한 다음 정책 연결을 선택합니다.

2단계: 추적을 위해 template.yml 파일 수정

1. https://console.aws.amazon.com/codestar/에서 AWS CodeStar 콘솔을 엽니다.
2. 서버리스 프로젝트를 선택하고 코드 페이지를 엽니다. 리포지토리 최상위에서 `template.yml` 파일을 찾아 편집합니다. `Resources`에서 리소스를 `Properties` 섹션으로 붙여넣습니다.

```
      Tracing: Active
```

이 예제는 수정된 템플릿을 보여줍니다.

```
Resources:
  GetHelloWorld:
    Type: AWS::Serverless::Function
    Properties:
      Handler: index.get
      Runtime: nodejs4.3
      Tracing: Active # Enable X-Ray tracing for the function
      Role:
        Fn::ImportValue:
          !Join ['-', [!Ref 'ProjectId', !Ref 'AWS::Region', 'LambdaTrustRole']]
      Events:
        GetEvent:
          Type: Api
          Properties:
            Path: /
            Method: get
```

3단계: 추적을 위해 템플릿 변경사항을 커밋 및 푸시

- `template.yml` 파일의 변경사항을 커밋하고 푸시합니다.

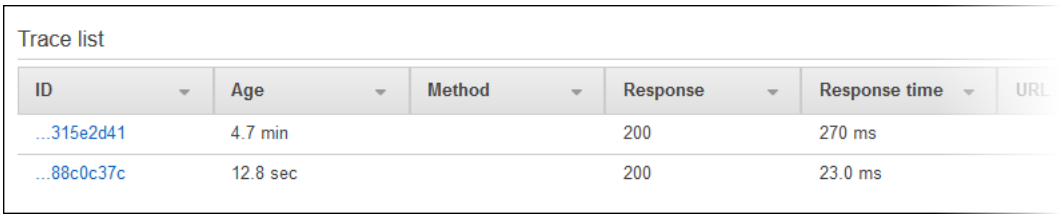
> **Note**
>
> 이 작업은 파이프라인을 가동합니다. IAM 권한을 업데이트하기 전에 변경사항을 커밋하면, 파이프라인이 가동되며 AWS CloudFormation 스택 업데이트에 오류가 발생하고, 스택 업데이트가 취소됩니다. 이 문제가 발생하면 권한을 수정하고 파이프라인을 다시 시작하십시오.

4단계: 추적을 위해 AWS CloudFormation 스택 업데이트 모니터링

1. AWS CloudFormation 스택 업데이트는 프로젝트의 파이프라인이 배포 단계를 개시하면 시작됩니다. 스택 업데이트 상태를 보려면, AWS CodeStar 대시보드에서 파이프라인의 AWS CloudFormation 단계를 선택하십시오.

   AWS CloudFormation의 스택 업데이트에 오류가 발생한다면, AWS CloudFormation: 누락된 권한 때문에 스택 생성이 취소됨의 문제 해결 지침을 참조하십시오. 작업자 역할에서 권한이 누락되었다면, 프로젝트의 Lambda 작업자 역할에 연결된 정책을 편집하십시오. 1단계: 추적을 위해 IAM의 작업자 역할 편집 섹션을 참조하세요.
2. 대시보드를 이용해 파이프라인이 성공적으로 완수되었는지 확인하십시오. 이제 애플리케이션에서 추적이 활성화되었습니다.
3. Lambda 콘솔에서 함수의 세부 정보를 확인해 추적 활성화 여부를 확인하십시오.

4. 프로젝트의 애플리케이션 엔드포인트를 선택합니다. 애플리케이션과의 이러한 상호작용이 추적됩니다. AWS X-Ray 콘솔에서 추적 정보를 확인할 수 있습니다.



프로젝트에 리소스 추가

모든 프로젝트의 각 AWS CodeStar 템플릿은 데이터베이스 테이블 및 Lambda 함수 같은 애플리케이션의 AWS 실행 시간 종속성을 보여주는 AWS CloudFormation 파일과 함께 제공됩니다. 이는 `/template.yml` 파일의 소스 리포지토리에 저장됩니다.

> **Note**
>
> 다음 프로젝트 지원 변경으로 생성된 다음 프로젝트에 이러한 단계를 사용할 수 있습니다.
>
> - 모든 Lambda 프로젝트
> - 2018년 8월 3일 이후에 만든 Amazon EC2 또는 Elastic Beanstalk 프로젝트의 경우, AWS CodeStar는 프로젝트 리포지토리에 `/template.yml` 파일을 프로비저닝했습니다.

AWS CloudFormation 리소스를 `Resources` 섹션에 추가하면 이 파일을 수정할 수 있습니다. `template.yml` 파일을 수정하면 AWS CodeStar와 AWS CloudFormation이 새 리소스를 프로젝트에 추가할 수 있습니다. 리소스에 따라 프로젝트의 CloudFormation 작업자 역할을 위해 다른 권한을 정책에 추가해야 할 수도 있습니다. 템플릿 요소와 형식에 대한 정보는 AWS 리소스 유형 참조를 참조하세요.

프로젝트에 추가할 리소스를 결정하면, 이러한 상위 단계를 밟아 템플릿을 사용자 지정하십시오. AWS CloudFormation 리소스 및 필수 속성 목록은 AWS 리소스 유형 참조에서 확인할 수 있습니다.

1. 1단계: IAM의 CloudFormation 작업자 역할 편집(필요 시)
2. 2단계: template.yml 파일 수정
3. 3단계: 템플릿 변경사항을 커밋 및 푸시
4. 4단계: AWS CloudFormation 스택 업데이트 모니터링
5. 5단계: 인라인 정책이 있는 리소스 권한 추가

이 섹션의 단계를 이용해 AWS CodeStar 프로젝트 템플릿을 수정하여 리소스를 추가한 다음 프로젝트의 IAM 내 CloudFormation 작업자 역할 권한을 확장하십시오. 이 예제에서는 AWS::SQS::Queue 리소스를 `template.yml` 파일에 추가했습니다. 변경사항은 Amazon Simple Queue Service 대기열을 프로젝트에 추가하는 자동 응답을 AWS CloudFormation에서 시작합니다.

1단계: IAM의 CloudFormation 작업자 역할 편집

1단계와 5단계를 수행하려면 관리자로 로그인해야 합니다.

> **Note**
> 프로젝트가 권한 경계 정책으로 프로비저닝된 경우 이 단계를 건너뛸 수 있습니다. 2018년 12월 6일(PDT) 이후에 만든 프로젝트의 경우, AWS CodeStar는 프로젝트를 권한 경계 정책으로 프로비저닝했습니다.

1. AWS Management Console에 로그인하고 https://console.aws.amazon.com/codestar/에서 AWS CodeStar 콘솔을 열 수 있습니다.
2. 프로젝트를 생성하거나 `template.yml file`이 있는 기존 프로젝트를 선택한 다음, 프로젝트 리소스 페이지를 엽니다.
3. 프로젝트 리소스의 리소스 목록에서 CodeStarWorker/AWS CloudFormation 역할에 대해 생성된 IAM 역할을 찾습니다. 역할 이름은 다음 형식을 취합니다. `role/CodeStarWorker-`*`Project_name`*`-CloudFormation`.
4. 역할이 IAM 콘솔에서 열립니다. 권한 탭의 Inline Policies(인라인 정책)에서 서비스 역할 정책의 열을 확장하고 정책 편집을 선택합니다.
5. JSON 탭을 선택해 정책을 편집합니다.

   > **Note**
   > 작업자 역할에 연결된 정책은 `CodeStarWorkerCloudFormationRolePolicy`입니다.

6. JSON 필드의 `Statement` 요소에 다음 정책 설명을 추가합니다.

```
{
```

```
  "Action": [
    "sqs:CreateQueue",
    "sqs:DeleteQueue",
    "sqs:GetQueueAttributes",
    "sqs:SetQueueAttributes",
    "sqs:ListQueues",
    "sqs:GetQueueUrl"
  ],
  "Resource": [
    "*"
  ],
  "Effect": "Allow"
}
```

7. 정책 검토를 선택해 정책에 오류가 없는지 확인하고, 변경 사항 저장을 선택합니다.

2단계: template.yml 파일 수정

1. https://console.aws.amazon.com/codestar/에서 AWS CodeStar 콘솔을 엽니다.
2. 서버리스 프로젝트를 선택하고 코드 페이지를 엽니다. 리포지토리 최상위에서 `template.yml` 위치를 기록해 둡니다.
3. IDE, 콘솔이나 로컬 리포지토리의 명령줄을 이용해 리포지토리의 `template.yml` 파일을 편집합니다. 리소스를 `Resources` 섹션으로 붙여넣습니다. 이 예제에서는 다음 텍스트를 복사할 때 `Resources` 섹션이 추가됩니다.

```
Resources:
  TestQueue:
    Type: AWS::SQS::Queue
```

이 예제는 수정된 템플릿을 보여줍니다.

```
Resources:
  HelloWorld:
    Type: AWS::Serverless::Function
    Properties:
      Handler: index.handler
      Runtime: python3.6
      Role:
        Fn::ImportValue:
          !Join ['-', [!Ref 'ProjectId', !Ref 'AWS::Region', 'LambdaTrustRole']]
      Events:
        GetEvent:
          Type: Api
          Properties:
            Path: /
            Method: get
        PostEvent:
          Type: Api
          Properties:
            Path: /
            Method: post

  TestQueue:
    Type: AWS::SQS::Queue
```

3단계: 템플릿 변경사항을 커밋 및 푸시

- 2단계에서 저장한 `template.yml` 파일의 변경사항을 커밋하고 푸시합니다.

> **Note**
>
> 이 작업은 파이프라인을 가동합니다. IAM 권한을 업데이트하기 전에 변경사항을 커밋하면, 파이프라인이 가동되며 AWS CloudFormation 스택 업데이트에 오류가 발생하고, 따라서 스택 업데이트가 취소됩니다. 이 문제가 발생하면 권한을 수정하고 파이프라인을 다시 시작하십시오.

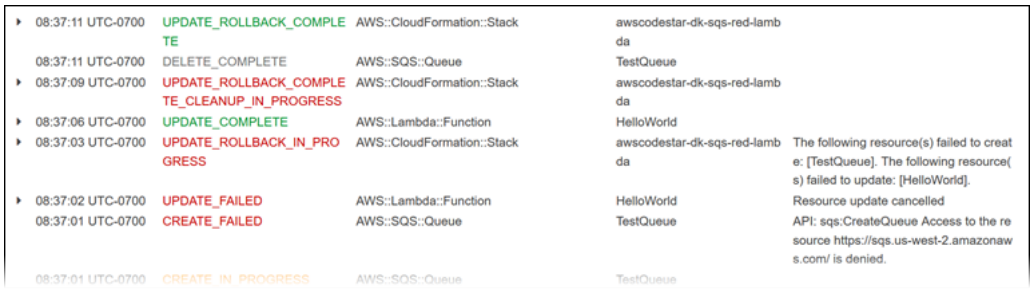
4단계: AWS CloudFormation 스택 업데이트 모니터링

1. 프로젝트의 파이프라인이 배포 단계를 개시하면 AWS CloudFormation 스택 업데이트가 시작됩니다. AWS CodeStar 대시보드에서 파이프라인의 AWS CloudFormation 단계를 선택하면 스택 업데이트를 확인할 수 있습니다.

   문제 해결:

   필수 리소스 권한이 없으면 스택 업데이트가 실패하게 됩니다. AWS CodeStar 대시보드 보기에서 프로젝트 파이프라인의 실패 상태를 확인해 보십시오.

파이프라인의 배포 단계에서 CloudFormation 링크를 선택해 AWS CloudFormation 콘솔의 결함을 해결하십시오. 콘솔의 이벤트 목록에서 프로젝트를 선택해 스택 생성 세부 정보를 확인하십시오. 실패 세부 정보를 표시하는 메시지가 있습니다. 이 예제에서는 `sqs:CreateQueue` 권한이 누락되었습니다.



프로젝트의 AWS CloudFormation 작업자 역할에 연결된 정책을 편집해 누락된 권한을 추가하십시오. 1단계: IAM의 CloudFormation 작업자 역할 편집 섹션을 참조하세요.

2. 파이프라인을 성공적으로 실행하면, 리소스는 AWS CloudFormation 스택에서 생성됩니다. AWS CloudFormation의 리소스 목록에서 프로젝트용으로 생성된 리소스를 확인합니다. 이 예제에서 TestQueue 대기열은 리소스 섹션에 나열됩니다.

   대기열 URL은 AWS CloudFormation에서 사용할 수 있습니다. 대기열 URL은 다음 형식을 따릅니다.

   ```
   https://{REGION_ENDPOINT}/queue.|api-domain|/{YOUR_ACCOUNT_NUMBER}/
   {YOUR_QUEUE_NAME}
   ```

   자세한 내용은 Amazon SQS 메시지 전송, Amazon SQS 대기열에서 메시지 수신, Amazon SQS 대기열에서 메시지 삭제를 참조하세요.

5단계: 인라인 정책이 있는 리소스 권한 추가

적절한 인라인 정책을 사용자의 역할에 추가해 팀원이 새 리소스에 액세스할 수 있게 합니다. 권한을 추가하지 않아도 되는 리소스도 있습니다. 다음 단계를 시행하려면, 루트 사용자나 계정의 관리자, 또는 `AdministratorAccess` 관리형 정책 또는 동급의 정책이 있는 IAM 사용자 또는 연합된 사용자로 콘솔에 로그인해야 합니다.

JSON 정책 편집기를 사용하여 정책을 생성하려면

1. AWS Management Console에 로그인하여 https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.

2. 왼쪽의 탐색 창에서 정책을 선택합니다.

   정책을 처음으로 선택하는 경우 관리형 정책 소개 페이지가 나타납니다. 시작을 선택합니다.
3. 페이지 상단에서 정책 생성을 선택합니다.
4. 정책 편집기 섹션에서 JSON 옵션을 선택합니다.
5. 다음 JSON 정책 문서를 입력합니다.

```
{
 "Action": [
   "sqs:CreateQueue",
   "sqs:DeleteQueue",
   "sqs:GetQueueAttributes",
   "sqs:SetQueueAttributes",
   "sqs:ListQueues",
   "sqs:GetQueueUrl"
 ],
 "Resource": [
   "*"
 ],
 "Effect": "Allow"
}
```

6. 다음을 선택합니다.

> **Note**
>
> 언제든지 시각적 편집기 옵션과 JSON 편집기 옵션을 서로 전환할 수 있습니다. 그러나 변경을 적용하거나 시각적 편집기에서 다음을 선택한 경우 IAM은 시각적 편집기에 최적화되도록 정책을 재구성할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 정책 재구성을 참조하세요.

7. 검토 및 생성 페이지에서 생성하는 정책에 대한 정책 이름과 설명(선택 사항)을 입력합니다. 이 정책에 정의된 권한을 검토하여 정책이 부여한 권한을 확인합니다.
8. 정책 생성을 선택하고 새로운 정책을 저장합니다.

프로젝트에 IAM 역할 추가

2018년 12월 6일부터 애플리케이션 스택(template.yml)에서 고유한 역할과 정책을 정의할 수 있습니다. 권한 상승 및 안전하지 않은 작업의 위험을 완화하려면 생성하는 모든 IAM 엔터티에 대해 프로젝트

별 권한 경계를 설정해야 합니다. 함수가 여러 개 있는 Lambda 프로젝트가 있는 경우, 각 함수에 대한 IAM 역할을 만드는 것이 좋습니다.

프로젝트에 IAM 역할을 추가하려면

1. 프로젝트에 대한 `template.yml` 파일을 편집합니다.
2. `Resources:` 섹션에서 다음 예제의 형식을 사용하여 IAM 리소스를 추가합니다.

```
  SampleRole:
  Description: Sample Lambda role
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Statement:
      - Effect: Allow
        Principal:
          Service: [lambda.amazonaws.com]
        Action: sts:AssumeRole
    ManagedPolicyArns:
      -  arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole
    PermissionsBoundary: !Sub 'arn:${AWS::Partition}:iam::${AWS::AccountId}:policy/
CodeStar_${ProjectId}_PermissionsBoundary'
```

3. 파이프라인을 통해 변경 사항을 릴리스하고 성공을 확인합니다.

프로젝트에 Prod 단계 및 엔드포인트 추가

이 단원의 절차에 따라 파이프라인에 새 프로덕션(Prod) 단계를 추가하고 파이프라인의 배포와 Prod 단계 사이에 수동 승인 단계를 추가합니다. 그러면 프로젝트 파이프라인을 실행할 때 추가 리소스 스택이 생성됩니다.

> **Note**
>
> 다음의 경우에 이러한 절차를 사용할 수 있습니다.
>
> - 2018년 8월 3일 이후에 생성된 프로젝트의 경우, AWS CodeStar는 프로젝트 리포지토리 파일과 함께 Amazon EC2, Elastic Beanstalk 또는 Lambda 프로젝트를 프로비저닝했습니다.

- 2018년 12월 6일(PDT) 이후에 만든 프로젝트의 경우, AWS CodeStar는 프로젝트를 권한 경계 정책으로 프로비저닝했습니다.

모든 AWS CodeStar 프로젝트는 Linux 인스턴스 및 Lambda 함수 같은 애플리케이션의 AWS 실행 시간 종속성을 보여주는 AWS CloudFormation 템플릿 파일을 사용합니다. `/template.yml` 파일은 소스 리포지토리에 저장됩니다.

`/template.yml` 파일에서 `Stage` 파라미터를 사용하여 프로젝트 파이프라인의 새 단계에 대한 리소스 스택을 추가합니다.

```
  Stage:
    Type: String
    Description: The name for a project pipeline stage, such as Staging or Prod, for
which resources are provisioned and deployed.
    Default: ''
```

`Stage` 파라미터는 리소스에서 참조된 프로젝트 ID가 있는 명명된 모든 리소스에 적용됩니다. 예를 들어 다음 역할 이름은 템플릿의 명명된 리소스입니다.

```
RoleName: !Sub 'CodeStar-${ProjectId}-WebApp${Stage}'
```

필수 조건

AWS CodeStar 콘솔에서 템플릿 옵션을 사용하여 프로젝트를 만듭니다.

IAM 사용자에게 다음 권한이 있는지 확인하십시오.

- 프로젝트 AWS CloudFormation 역할에 대한 `iam:PassRole`.
- 프로젝트 도구 체인 역할에 대한 `iam:PassRole`.
- `cloudformation:DescribeStacks`
- `cloudformation:ListChangeSets`

Elastic Beanstalk 또는 Amazon EC2 프로젝트에만 해당:

- `codedeploy:CreateApplication`
- `codedeploy:CreateDeploymentGroup`

- `codedeploy:GetApplication`
- `codedeploy:GetDeploymentConfig`
- `codedeploy:GetDeploymentGroup`
- `elasticloadbalancing:DescribeTargetGroups`

주제

1단계: CodeDeploy에서 새 배포 그룹 생성 (Amazon EC2 프로젝트만 해당)

CodeDeploy 애플리케이션을 선택한 후 새 인스턴스와 연결된 새 배포 그룹을 추가합니다.

> **Note**
>
> 프로젝트가 Lambda 또는 Elastic Beanstalk 프로젝트인 경우, 이 단계를 건너뛸 수 있습니다.

1. https://console.aws.amazon.com/codedeploy에서 CodeDeploy 콘솔을 엽니다.
2. AWS CodeStar에서 프로젝트를 만들 때 생성된 CodeDeploy 애플리케이션을 선택합니다.
3. [Deployment groups]에서 [Create deployment group]을 선택합니다.
4. Deployment group name(배포 그룹 이름)에 ***<project-id>-prod-Env***을 입력합니다.
5. 서비스 역할에서 AWS CodeStar 프로젝트의 도구 체인 작업자 역할을 선택합니다.
6. 배포 유형 아래에서 인 플레이스를 선택합니다.
7. 환경 구성 아래에서 Amazon EC2 인스턴스 탭을 선택합니다.
8. 태그 그룹 아래의 키 아래에서 `aws:cloudformation:stack-name`을 선택합니다. 값 아래에서 `awscodestar-<projectid>-infrastructure-prod`(GenerateChangeSet 작업에 대해 생성될 스택)를 선택합니다.
9. 배포 설정에서 `CodeDeployDefault.AllAtOnce`를 선택합니다.
10. 로드 밸런서 선택을 지웁니다.

11. [Create deployment group]을 선택합니다.

    이제 두 번째 배포 그룹이 생성되었습니다.

2단계: Prod 단계에 대한 새 파이프라인 단계 추가

프로젝트의 배포 단계와 동일한 배포 작업 세트가 있는 단계를 추가합니다. 예를 들어 Amazon EC2 프로젝트의 새 Prod 단계에는 프로젝트에 대해 생성된 배포 단계와 동일한 작업이 있어야 합니다.

배포 단계에서 파라미터와 필드를 복사하려면

1. AWS CodeStar 프로젝트 대시보드에서 파이프라인 세부 정보를 선택하여 CodePipeline 콘솔에서 파이프라인을 엽니다.
2. 편집을 선택합니다.
3. 배포 단계에서 단계 편집을 선택합니다.
4. GenerateChangeSet 작업의 편집 아이콘을 선택합니다. 다음 필드에 값을 적어 둡니다. 새 작업을 만들 때 이러한 값을 사용합니다.
   - 스택 이름
   - 변경 세트 이름
   - 템플릿
   - 템플릿 구성
   - 입력 아티팩트
5. 고급을 확장한 후 파라미터에 프로젝트의 파라미터를 복사합니다. 이러한 파라미터를 새 작업에 붙여넣습니다. 예를 들어 여기에서 JSON 형식으로 표시된 파라미터를 복사합니다.
   - Lambda 프로젝트:

     ```
     {
       "ProjectId":"MyProject"
     }
     ```

   - Amazon EC2 프로젝트:

     ```
     {

       "ProjectId":"MyProject",
       "InstanceType":"t2.micro",
     ```

```
  "WebAppInstanceProfile":"awscodestar-MyProject-WebAppInstanceProfile-
EXAMPLEY5VSFS",
  "ImageId":"ami-EXAMPLE1",
  "KeyPairName":"my-keypair",
  "SubnetId":"subnet-EXAMPLE",
  "VpcId":"vpc-EXAMPLE1"
}
```

- Elastic Beanstalk 프로젝트:

```
{
  "ProjectId":"MyProject",
  "InstanceType":"t2.micro",
  "KeyPairName":"my-keypair",
  "SubnetId":"subnet-EXAMPLE",
  "VpcId":"vpc-EXAMPLE",
  "SolutionStackName":"64bit Amazon Linux 2018.03 v3.0.5 running Tomcat 8 Java
 8",
  "EBTrustRole":"CodeStarWorker-myproject-EBService",
  "EBInstanceProfile":"awscodestar-myproject-EBInstanceProfile-11111EXAMPLE"
}
```

6. 단계 편집 창에서 취소를 선택합니다.

새 Prod 단계에서 GenerateChangeSet 작업을 만들려면

> **Note**
>
> 새 작업을 추가한 후에도 편집 모드에서 편집용으로 새 작업을 다시 열면 일부 필드가 표시되지 않을 수 있습니다. 스택 stack-name이(가) 존재하지 않음이 표시될 수도 있습니다.
> 이 오류로 인해 파이프라인을 저장할 수 없는 것은 아닙니다. 그러나 누락된 필드를 복원하려면 새 작업을 삭제하고 다시 추가해야 합니다. 파이프라인을 저장하고 실행하면 스택이 인식되고 오류가 다시 나타나지 않습니다.

1. 파이프라인이 아직 표시되지 않은 경우 AWS CodeStar 프로젝트 대시보드에서 파이프라인 세부 정보를 선택하여 콘솔에서 파이프라인을 엽니다.
2. 편집을 선택합니다.
3. 다이어그램의 하단에서 + 단계 추가를 선택합니다.
4. 단계 이름(예: **Prod**)을 입력한 후 + 작업 그룹 추가를 선택합니다.

5. 작업 이름에 이름을 입력합니다(예: **GenerateChangeSet**).
6. 작업 공급자에서 AWS CloudFormation을 선택합니다.
7. 작업 모드에서 변경 사항 세트 생성 또는 교체를 선택합니다.
8. 스택 이름에 이 작업으로 생성될 AWS CloudFormation 스택의 새 이름을 입력합니다. 배포 스택 이름과 동일한 이름으로 시작한 후 **-prod**를 추가합니다.

   - Lambda 프로젝트: `awscodestar-<project_name>-lambda-prod`
   - Amazon EC2 및 Elastic Beanstalk 프로젝트: `awscodestar-<project_name>-infrastructure-prod`

   > **Note**
   >
   > 스택 이름은 정확히 **awscodestar-<project_name>-**로 시작해야 합니다. 그렇지 않으면 스택 생성이 실패합니다.

9. 변경 세트 이름에 기존 배포 단계에 입력한 것과 동일한 변경 세트 이름을 입력합니다(예: **pipeline-changeset**).
10. 입력 아티팩트에서 빌드 아티팩트를 선택합니다.
11. 템플릿에 기존 배포 단계에 입력한 것과 동일한 변경 템플릿 이름을 입력합니다(예: **<project-ID>-BuildArtifact::template.yml**).
12. 템플릿 구성에 배포 단계에 입력한 것과 동일한 변경 템플릿 구성 파일 이름을 입력합니다(예: **<project-ID>-BuildArtifact::template-configuration.json**).
13. 기능에서 CAPABILITY_NAMED_IAM을 선택합니다.
14. 역할 이름에서 프로젝트의 AWS CloudFormation 작업자 역할의 이름을 선택합니다.
15. 고급을 확장한 후 파라미터에 프로젝트의 파라미터를 붙여넣습니다. 여기에서 JSON 형식으로 표시된 Amazon EC2 프로젝트의 `Stage` 파라미터를 포함시킵니다.

```
{

  "ProjectId":"MyProject",
  "InstanceType":"t2.micro",
  "WebAppInstanceProfile":"awscodestar-MyProject-WebAppInstanceProfile-
EXAMPLEY5VSFS",
  "ImageId":"ami-EXAMPLE1",
  "KeyPairName":"my-keypair",
```

```
  "SubnetId":"subnet-EXAMPLE",
  "VpcId":"vpc-EXAMPLE1",
  "Stage":"Prod"
}
```

> **Note**
>
> 변경하려는 새 파라미터만이 아니라 프로젝트의 모든 파라미터를 복사해야 합니다.

16. Save를 선택합니다.
17. AWS CodePipeline 창에서 파이프라인 변경 사항 저장을 선택한 다음 변경 사항 저장을 선택합니다.

> **Note**
>
> 변경 감지 리소스가 삭제 및 추가되었음을 알리는 메시지가 표시될 수 있습니다. 메시지를 확인하고 이 자습서의 다음 단계로 계속 진행합니다.

업데이트된 파이프라인을 확인합니다.

새 Prod 단계에서 ExecuteChangeSet 작업을 생성하려면

1. 파이프라인이 아직 보이지 않는 경우 AWS CodeStar 프로젝트 대시보드에서 파이프라인 세부 정보를 선택하여 콘솔에서 파이프라인을 엽니다.
2. 편집을 선택합니다.
3. 새 Prod 단계에서 새 GenerateChangeSet 작업 후에 + 작업 그룹 추가를 선택합니다.
4. 작업 이름에 이름을 입력합니다(예: **`ExecuteChangeSet`**).
5. 작업 공급자에서 AWS CloudFormation을 선택합니다.
6. 작업 모드에서 변경 세트 실행을 선택합니다.
7. 스택 이름에 GenerateChangeSet 작업에 입력한 AWS CloudFormation 스택의 새 이름을 입력합니다(예: **`awscodestar-<project-ID>-infrastructure-prod`**).
8. 변경 세트 이름에 배포 단계에서 사용한 것과 동일한 변경 세트 이름을 입력합니다(예: **`pipeline-changeset`**).
9. 완료를 선택합니다.

10. AWS CodePipeline 창에서 파이프라인 변경 사항 저장을 선택한 다음 변경 사항 저장을 선택합니다.

> **Note**
>
> 변경 감지 리소스가 삭제 및 추가되었음을 알리는 메시지가 표시될 수 있습니다. 메시지를 확인하고 이 자습서의 다음 단계로 계속 진행합니다.

업데이트된 파이프라인을 확인합니다.

새 Prod 단계에서 CodeDeploy 배포 작업을 생성하려면(Amazon EC2 프로젝트만 해당)

1. Prod 단계의 새 작업 후에 + 작업을 선택합니다.
2. 작업 이름에 이름을 입력합니다(예: **`Deploy`**).
3. 작업 공급자에서 AWS CodeDeploy를 선택합니다.
4. 애플리케이션 이름에서 프로젝트의 CodeDeploy 애플리케이션 이름을 선택합니다.
5. 배포 그룹에서 2단계에서 만든 새 CodeDeploy 배포 그룹의 이름을 선택합니다.
6. 입력 아티팩트에서 기존 단계에서 사용한 것과 동일한 빌드 아티팩트를 선택합니다.
7. 완료를 선택합니다.
8. AWS CodePipeline 창에서 파이프라인 변경 사항 저장을 선택한 다음 변경 사항 저장을 선택합니다. 업데이트된 파이프라인을 확인합니다.

3단계: 수동 승인 단계 추가

새 프로덕션 단계 앞에 수동 승인 단계를 추가하는 것이 좋습니다.

1. 왼쪽 위에서 편집을 선택합니다.
2. 파이프라인 다이어그램의 배포 단계와 Prod 배포 단계 사이에서 + 단계 추가를 선택합니다.
3. 단계 편집에서 단계 이름(예: **`Approval`**)을 입력한 후 + 작업 그룹 추가를 선택합니다.
4. 작업 이름에 이름을 입력합니다(예: **`Approval`**).
5. [Approval type]에서 [Manual approval]을 선택합니다.
6. (선택 사항) 구성 아래의 SNS 주제 ARN에서 생성 및 구독한 SNS 주제를 선택합니다.
7. 작업 추가를 선택합니다.

8. AWS CodePipeline 창에서 파이프라인 변경 사항 저장을 선택한 다음 변경 사항 저장을 선택합니다. 업데이트된 파이프라인을 확인합니다.
9. 변경 사항을 제출하고 파이프라인 빌드를 시작하려면 변경 사항 배포를 선택한 다음 릴리스를 선택합니다.

4단계: 변경 푸시 및 AWS CloudFormation 스택 업데이트 모니터링

1. 파이프라인이 실행되는 동안 여기의 단계를 사용하여 새 단계를 위한 스택 및 엔드포인트 생성을 수행할 수 있습니다.
2. 파이프라인이 배포 단계를 시작하면, AWS CloudFormation 스택 업데이트가 시작됩니다. AWS CodeStar 대시보드에서 파이프라인의 AWS CloudFormation 단계를 선택하면 스택 업데이트 알림을 확인할 수 있습니다. 스택 생성 세부 정보를 보려면 콘솔의 이벤트 목록에서 프로젝트를 선택합니다.
3. 파이프라인을 성공적으로 완수하면, 리소스는 AWS CloudFormation 스택에서 생성됩니다. AWS CloudFormation 콘솔에서 프로젝트의 인프라 스택을 선택합니다. 스택 이름은 다음 형식을 따릅니다.

   - Lambda 프로젝트: `awscodestar-<project_name>-lambda-prod`
   - Amazon EC2 및 Elastic Beanstalk 프로젝트: `awscodestar-<project_name>-infrastructure-prod`

   AWS CloudFormation 콘솔의 리소스 목록에서 프로젝트용으로 생성된 리소스를 확인합니다. 이 예에서는 새 Amazon EC2 인스턴스가 리소스 섹션에 표시됩니다.
4. 프로덕션 단계의 엔드포인트에 액세스합니다.

   - Elastic Beanstalk 프로젝트의 경우 AWS CloudFormation 콘솔에서 새로운 스택을 열고 리소스를 확장합니다. Elastic Beanstalk 애플리케이션을 선택합니다. Elastic Beanstalk 콘솔에서 링크가 열립니다. 환경을 선택합니다. URL에서 해당 URL을 선택하여 브라우저에서 엔드포인트를 엽니다.
   - Lambda 프로젝트의 경우 AWS CloudFormation 콘솔에서 새로운 스택을 열고 리소스를 확장합니다. API Gateway 리소스를 선택합니다. API Gateway 콘솔에서 링크가 열립니다. 단계를 선택합니다. URL 간접 호출에서 해당 URL을 선택하여 브라우저에서 엔드포인트를 엽니다.
   - Amazon EC2 프로젝트의 경우 AWS CodeStar 콘솔의 프로젝트 리소스 목록에서 새 Amazon EC2 인스턴스를 선택합니다. Amazon EC2 콘솔의 인스턴스 페이지에서 링크가 열립니다. 설명 탭을 선택하고 퍼블릭 DNS(IPv4)의 URL을 복사한 후 브라우저에서 URL을 엽니다.

5. 변경이 배포되었는지 확인합니다.

프로젝트에서 SSM 파라미터를 안전하게 사용 AWS CodeStar

많은 고객은 시스템 관리자 파라미터 스토어 파라미터에 자격 증명 등의 암호를 저장합니다. 이제 AWS CodeStar 프로젝트에서 이러한 파라미터를 안전하게 사용할 수 있습니다. 예를 들어, 빌드 사양에서 CodeBuild 또는 툴체인 스택 (template.yml) 에서 애플리케이션 리소스를 정의할 때 SSM 파라미터를 사용하는 것이 좋습니다.

AWS 프로젝트에서 SSM 파라미터를 사용하려면 AWS CodeStar 프로젝트 ARN으로 파라미터에 수동으로 태그를 지정해야 합니다. CodeStar 또한 태그를 지정한 파라미터에 액세스할 수 있는 적절한 권한을 AWS CodeStar 툴체인 작업자 역할에 제공해야 합니다.

시작하기 전

- 새 파라미터를 생성하거나 액세스하려는 정보가 포함된 기존 시스템 관리자 파라미터를 식별합니다.
- 사용하려는 AWS CodeStar 프로젝트를 식별하거나 새 프로젝트를 생성하십시오.
- CodeStar 프로젝트 ARN을 기록해 둡니다. `arn:aws:codestar:`*`region-id`*`:`*`account-id`*`:project/`*`project-id`*와 같은 형태입니다.

AWS CodeStar 프로젝트 ARN으로 파라미터에 태그 지정

단계별 지침은 Systems Manager 파라미터 태그 지정 섹션을 참조하십시오.

1. 키에 `awscodestar:projectArn`을 입력합니다.
2. 값에 프로젝트 ARN 출처 CodeStar: 를 입력합니다. `arn:aws:codestar:`*`region-id`*`:`*`account-id`*`:project/`*`project-id`*
3. 저장을 선택합니다.

이제 template.yml 파일의 SSM 파라미터를 참조할 수 있습니다. 도구 체인 작업자 역할과 함께 사용하려면 추가 권한을 부여해야 합니다.

AWS CodeStar 프로젝트 도구 모음에서 태그가 지정된 파라미터를 사용할 수 있는 권한 부여

> **Note**
>
> 이러한 단계는 2018년 12월 6일(PDT) 이후에 생성된 프로젝트에만 적용됩니다.

1. 사용하려는 CodeStar 프로젝트의 AWS 프로젝트 대시보드를 엽니다.
2. 프로젝트를 클릭하여 생성된 리소스의 목록을 보고 도구 체인 작업자 역할을 찾습니다. `role/CodeStarWorker-`*`project-id`*`-ToolChain` 형식의 이름을 가진 IAM 리소스입니다.
3. ARN을 클릭하여 IAM 콘솔에서 엽니다.
4. 필요한 경우 위치를 ToolChainWorkerPolicy 찾아 확장하십시오.
5. 정책 편집을 클릭합니다.
6. `Action:` 아래에서 다음 줄을 추가합니다.

   ```
   ssm:GetParameter*
   ```
7. 정책 검토를 클릭한 다음 변경 내용 저장을 클릭합니다.

2018년 12월 6일(PDT) 이전에 생성된 프로젝트의 경우 각 서비스에 대한 작업자 역할에 다음 권한을 추가해야 합니다.

```
        {
            "Action": [
                "ssm:GetParameter*"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Condition": {
                "StringEquals": {
                    "ssm:ResourceTag/awscodestar:projectArn": "arn:aws:codestar:region-
id:account-id:project/project-id"
                }
            }
        }
```

AWS Lambda 프로젝트의 트래픽 이동

AWS CodeDeploy는 AWS CodeStar 서버리스 제품에서의 AWS Lambda 함수를 위한 함수 버전 배포를 지원합니다. AWS Lambda 배포가 수신 트래픽을 기존 Lambda 함수에서 업데이트된 Lambda 함수 버전으로 전환합니다. 필요하다면 별도의 버전을 배포한 다음 최초 버전으로 되돌려 업데이트된 Lambda 함수를 테스트할 수도 있습니다.

이 섹션의 단계를 이용해 AWS CodeStar 프로젝트 템플릿을 수정하고 CodeStarWorker 역할 IAM 권한을 업데이트하십시오. 이 작업은 별칭이 있는 AWS Lambda 함수를 생성하고 AWS CodeDeploy에게 트래픽을 업데이트된 환경으로 옮기도록 지시하는 AWS CloudFormation에서의 자동 응답을 시작합니다.

> **Note**
>
> 2018년 12월 12일 전에 AWS CodeStar 프로젝트를 생성한 경우에만 다음 단계를 완료하십시오.

AWS CodeDeploy에는 사용자가 트래픽을 애플리케이션의 AWS Lambda 함수 버전으로 옮기게 하는 3가지 배포 옵션이 있습니다.

- 카나리(Canary): 트래픽이 두 증분으로 나뉘어 이동합니다. 나머지 트래픽이 두 번째 증분으로 이동하기 전에 첫 증분에서 업데이트된 Lambda 함수 버전으로 이동할 트래픽 비율(%), 간격(분)을 지정하는 사전 정의된 Canary 옵션 중에서 선택할 수 있습니다.
- 리니어(Linear): 트래픽이 동일한 증분 이동하며 각 증분 간에 시간 간격(분)이 동일합니다. 각 증분에서 이동할 트래픽 비율(%)과 각 증분 간의 시간 간격(분)을 지정하는 사전 정의된 선형 옵션에서 선택할 수 있습니다. 트래픽이 동일한 증분으로 이동하며 각 증분 간에 시간(분)이 동일합니다. 각 증분에서 이동할 트래픽 비율(%)과 각 증분 간의 시간 간격(분)을 지정하는 사전 정의된 선형 옵션에서 선택할 수 있습니다.
- All-at-once: 모든 트래픽이 기존 Lambda 함수에서 업데이트된 Lambda 함수 버전으로 한번에 이동합니다.

배포 기본 설정 유형
Canary10Percent30Minutes
Canary10Percent5Minutes

배포 기본 설정 유형
Canary10Percent10Minutes
Canary10Percent15Minutes
Linear10PercentEvery10Minutes
Linear10PercentEvery1Minute
Linear10PercentEvery2Minutes
Linear10PercentEvery3Minutes
AllAtOnce

AWS Lambda 컴퓨팅 플랫폼에서의 AWS CodeDeploy 배포에 대한 자세한 내용은 AWS Lambda 컴퓨팅 플랫폼에서의 배포를 참조하세요.

AWS SAM에 대한 자세한 내용은 GitHub의 AWS Serverless Application Model(AWS SAM)을 참조하세요.

사전 조건:

서버리스 제품을 만들 때는 Lambda 컴퓨팅 플랫폼이 있는 템플릿을 선택하십시오. 4~6단계를 수행하려면 관리자로 로그인해야 합니다.

1단계: SAM 템플릿을 수정해 AWS Lambda 버전 배포 파라미터를 추가

1. https://console.aws.amazon.com/codestar/에서 AWS CodeStar 콘솔을 엽니다.
2. 프로젝트를 만들거나 `template.yml` 파일이 있는 기존 프로젝트를 선택한 다음, 코드 페이지를 엽니다. 리포지토리 최상위에서 `template.yml`이라는 이름의 수정할 SAM 템플릿 위치를 기록해 둡니다.
3. IDE 또는 로컬 리포지토리에서 `template.yml` 파일을 엽니다. 다음 텍스트를 복사해 `Globals` 섹션을 파일에 추가합니다. 이 자습서의 샘플 텍스트는 `Canary10Percent5Minutes` 옵션을 선택합니다.

```
Globals:
  Function:
```

```
    AutoPublishAlias: live
    DeploymentPreference:
      Enabled: true
      Type: Canary10Percent5Minutes
```

이 예제는 `Globals` 섹션 추가 후 수정된 템플릿을 보여줍니다.

```
AWSTemplateFormatVersion: 2010-09-09
Transform:
- AWS::Serverless-2016-10-31
- AWS::CodeStar

Parameters:
  ProjectId:
    Type: String
    Description: CodeStar projectId used to associate new resources to team members

Globals:
  Function:
    AutoPublishAlias: live
    DeploymentPreference:
      Enabled: true
      Type: Canary10Percent5Minutes

Resources:
  HelloWorld:
    Type: AWS::Serverless::Function
    Properties:
      Handler: index.handler
      Runtime: python3.6
      Role:
        Fn::ImportValue:
          !Join ['-', [!Ref 'ProjectId', !Ref 'AWS::Region', 'LambdaTrustRole']]
      Events:
```

자세한 내용은 SAM 템플릿의 Globals Section 참조 설명서를 참조하십시오.

2단계: AWS CloudFormation 역할을 편집해 권한을 추가

1. AWS Management Console에 로그인하고 https://console.aws.amazon.com/codestar/에서 AWS CodeStar 콘솔을 엽니다.

   > **Note**
   >
   > AWS CodeStar설정에서 생성하거나 식별한 IAM 사용자와 연결된 자격 증명을 사용하여 AWS Management Console에 로그인해야 합니다. 이 사용자는 연결된 AWS **`AWSCodeStarFullAccess`** 관리형 정책을 보유해야 합니다.

2. 기존 서버리스 프로젝트를 선택하고 프로젝트 리소스 페이지를 엽니다.
3. 리소스에서 CodeStarWorker/AWS CloudFormation 역할에 대해 생성된 IAM 역할을 선택합니다. 역할이 IAM 콘솔에서 열립니다.

4. 권한 탭으로 가서 인라인 정책의 서비스 역할 정책 열에서 정책 편집을 선택합니다. JSON 탭을 선택해 JSON 형식의 정책을 편집합니다.

> **Note**
>
> 서비스 역할의 이름은 `CodeStarWorkerCloudFormationRolePolicy`입니다.

5. JSON 필드의 `Statement` 요소에 다음 정책 설명을 추가합니다. *`region`* 및 *`id`* 자리 표시자를 리전 및 계정 ID로 바꿉니다.

```
{
  "Action": [
    "s3:GetObject",
    "s3:GetObjectVersion",
    "s3:GetBucketVersioning"
  ],
  "Resource": "*",
  "Effect": "Allow"
},
{
  "Action": [
    "s3:PutObject"
  ],
  "Resource": [
    "arn:aws:s3:::codepipeline*"
  ],
  "Effect": "Allow"
},
{
  "Action": [
    "lambda:*"
  ],
  "Resource": [
    "arn:aws:lambda:region:id:function:*"
  ],
  "Effect": "Allow"
},
{
  "Action": [
    "apigateway:*"
  ],
  "Resource": [
```

```
      "arn:aws:apigateway:region::*"
    ],
    "Effect": "Allow"
  },
  {
    "Action": [
      "iam:GetRole",
      "iam:CreateRole",
      "iam:DeleteRole",
      "iam:PutRolePolicy"
    ],
    "Resource": [
      "arn:aws:iam::id:role/*"
    ],
    "Effect": "Allow"
  },
  {
    "Action": [
      "iam:AttachRolePolicy",
      "iam:DeleteRolePolicy",
      "iam:DetachRolePolicy"
    ],
    "Resource": [
      "arn:aws:iam::id:role/*"
    ],
    "Effect": "Allow"
  },
  {
    "Action": [
      "iam:PassRole"
    ],
    "Resource": [
      "*"
    ],
    "Effect": "Allow"
  },
  {
    "Action": [
      "codedeploy:CreateApplication",
      "codedeploy:DeleteApplication",
      "codedeploy:RegisterApplicationRevision"
    ],
    "Resource": [
      "arn:aws:codedeploy:region:id:application:*"
```

```
    ],
    "Effect": "Allow"
},
{
    "Action": [
        "codedeploy:CreateDeploymentGroup",
        "codedeploy:CreateDeployment",
        "codedeploy:DeleteDeploymentGroup",
        "codedeploy:GetDeployment"
    ],
    "Resource": [
        "arn:aws:codedeploy:region:id:deploymentgroup:*"
    ],
    "Effect": "Allow"
},
{
    "Action": [
        "codedeploy:GetDeploymentConfig"
    ],
    "Resource": [
        "arn:aws:codedeploy:region:id:deploymentconfig:*"
    ],
    "Effect": "Allow"
}
```

6. 정책 검토를 선택해 정책에 오류가 없는지 확인합니다. 정책에 오류가 없으면 변경 사항 저장을 선택합니다.

3단계: 템플릿 변경사항을 커밋 및 푸시해 AWS Lambda 버전 이동을 시작

1. 1단계에서 저장한 `template.yml` 파일의 변경사항을 커밋하고 푸시합니다.

> **Note**
>
> 이 작업은 파이프라인을 가동합니다. IAM 권한을 업데이트하기 전에 변경사항을 커밋하면, 파이프라인이 가동되며 AWS CloudFormation 스택 업데이트에 오류가 발생하고, 스택 업데이트가 취소됩니다. 이 문제가 발생하면 권한 수정 후 파이프라인을 다시 시작하십시오.

2. AWS CloudFormation 스택 업데이트는 프로젝트의 파이프라인이 배포 단계를 개시하면 시작됩니다. 배포 시작 시 스택 업데이트 알림을 보려면, AWS CodeStar 대시보드에서 파이프라인의 AWS CloudFormation 단계를 선택하십시오.

   스택 업데이트 중에 AWS CloudFormation이 다음과 같이 프로젝트 리소스를 자동으로 업데이트합니다.

   - AWS CloudFormation은 별칭이 있는 Lambda 함수, 이벤트 후크, 리소스를 생성해 `template.yml` 파일을 처리합니다.
   - AWS CloudFormation은 Lambda를 직접적으로 호출해 함수의 새 버전을 생성합니다.
   - AWS CloudFormation은 AppSpec 파일을 생성하고 AWS CodeDeploy를 직접적으로 호출해 트래픽을 이동합니다.

   SAM에서 별칭이 있는 Lambda 함수를 게시하는 방법에 대한 자세한 내용은 AWS Serverless Application Model(SAM) 템플릿 참조를 참조하십시오. AWS CodeDeploy AppSpec 파일에서의 이벤트 후크와 리소스에 대한 자세한 내용은 AppSpec '리소스' 섹션(AWS Lambda 배포 전용) 및 AWS Lambda 배포용 AppSpec '후크' 섹션 단원을 참조하십시오.
3. 파이프라인을 성공적으로 완수하면, 리소스는 AWS CloudFormation 스택에서 생성됩니다. 프로젝트 페이지의 프로젝트 리소스 목록에서 AWS CodeDeploy 애플리케이션과 AWS CodeDeploy 배포 그룹 및 프로젝트에 대해 생성된 AWS CodeDeploy 서비스 역할 리소스를 확인하십시오.
4. 새 버전을 생성하려면 리포지토리의 Lambda 함수를 변경해야 합니다. 새 배포가 시작되어 SAM 템플릿에 나온 배포 유형에 따라 트래픽을 옮깁니다. 새 버전으로 이동 중인 트래픽 상태를 확인하려면 프로젝트 페이지의 프로젝트 리소스 목록에서 AWS CodeDeploy 배포로 가는 링크를 선택하십시오.
5. 각 개정의 세부 정보를 확인하려면 개정에서 AWS CodeDeploy 배포 그룹으로 가는 링크를 선택하십시오.
6. 로컬 작업 디렉터리에서 AWS Lambda 함수에 변경사항을 적용하고 프로젝트 리포지토리에 대한 변경사항을 커밋할 수 있습니다. AWS CloudFormation은 AWS CodeDeploy를 이용해 같은 방법으로 다른 개정을 관리하는 기능을 지원합니다. Lambda 배포 재배포, 중단 또는 롤백에 대한 자세한 내용은 AWS Lambda 컴퓨팅 플랫폼에서의 배포 단원을 참조하세요.

AWS CodeStar 프로젝트를 프로덕션으로 전환

AWS CodeStar 프로젝트를 사용하여 애플리케이션을 생성하고 AWS CodeStar가 제공하는 사항을 확인한 후에는 프로젝트를 프로덕션 용도로 전환할 수 있습니다. 이를 수행하는 방법은 애플리케이션의

AWS 리소스를 AWS CodeStar 외부에서 복제하는 것입니다. 리포지토리, 빌드 프로젝트, 파이프라인 및 배포가 필요하지만, AWS CodeStar가 자동으로 생성하도록 하는 대신 AWS CloudFormation을 사용하여 다시 생성합니다.

> **Note**
>
> 먼저 AWS CodeStar 빠른 시작 중 하나를 사용하여 유사한 프로젝트를 생성하거나 확인한 다음, 이를 자체 프로젝트의 템플릿으로 사용하여 필요한 리소스와 정책을 포함시키는 것이 도움이 될 수 있습니다.

AWS CodeStar 프로젝트는 소스 코드와 코드 배포를 위해 생성한 리소스의 결합입니다. 코드 빌드, 릴리스, 배포에 도움이 되는 리소스 모음을 도구 체인 리소스라고 합니다. 프로젝트 생성 시 AWS CloudFormation 템플릿은 지속적인 통합/지속적인 배포(CI/CD) 파이프라인에 도구 체인 리소스를 공급합니다.

콘솔을 사용하여 프로젝트를 만들면, 도구 체인 템플릿이 생성됩니다. AWS CLI를 사용하여 프로젝트를 만들면, 도구 체인 리소스를 생성하는 도구 체인 템플릿을 생성하게 됩니다.

온전한 도구 체인은 다음과 같은 권장 리소스를 요구합니다.

1. 소스 코드가 포함된 CodeCommit 또는 GitHub 리포지토리입니다.
2. 리포지토리의 변경 사항을 주시하도록 구성된 CodePipeline 파이프라인입니다.
   a. AWS CodeBuild를 이용해 단위 또는 통합 테스트를 실행할 때는, 파이프라인에 빌드 단계를 추가해 빌드 아티팩트를 생성하는 방법을 권장합니다.
   b. 되도록 CodeDeploy 또는 AWS CloudFormation을 사용하는 배포 단계를 파이프라인에 추가해 빌드 아티팩트와 소스 코드를 실행 시간 인프라에 배포하십시오.

   > **Note**
   >
   > CodePipeline은 파이프라인에서 최소한 두 단계를 요구하며, 첫 번째 단계는 소스 단계여야 하므로 빌드나 배포 단계는 두 번째 단계로 추가하십시오.

주제

- GitHub 리포지토리 만들기

GitHub 리포지토리 만들기

GitHub 리포지토리를 도구 체인 템플릿에 정의하여 만듭니다. 소스 코드가 포함된 ZIP 파일의 위치를 이미 생성했으므로 코드를 리포지토리에 업로드할 수 있습니다. 또한 GitHub에 개인 액세스 토큰을 이미 생성했으므로 사용자를 대신하여 AWS가 GitHub에 연결할 수 있습니다. GitHub의 개인 액세스 토큰 외에도 전달하는 `Code` 객체에 대한 `s3.GetObject` 권한이 있어야 합니다.

퍼블릭 GitHub 리포지토리를 지정하려면 AWS CloudFormation의 도구 체인 템플릿에 다음과 같은 코드를 추가합니다.

```
GitHubRepo:
Condition: CreateGitHubRepo
Description: GitHub repository for application source code
Properties:
  Code:
    S3:
      Bucket: MyCodeS3Bucket
      Key: MyCodeS3BucketKey
  EnableIssues: true
  IsPrivate: false
  RepositoryAccessToken: MyGitHubPersonalAccessToken
  RepositoryDescription: MyAppCodeRepository
  RepositoryName: MyAppSource
  RepositoryOwner: MyGitHubUserName
Type: AWS::CodeStar::GitHubRepository
```

이 코드는 다음 정보를 지정합니다.

- 포함하려는 코드의 위치(Amazon S3 버킷이어야 함).
- GitHub 리포지토리에서 문제를 활성화할지 여부.
- GitHub 리포지토리가 프라이빗 리포지토리인지 여부.
- 생성한 GitHub 개인 액세스 토큰.
- 생성 중인 리포지토리에 대한 설명, 이름 및 소유자.

지정할 정보에 대한 자세한 내용은 AWS CloudFormation 사용 설명서의 AWS: :CodeStar: :GitHubRepository를 참조하십시오.

AWS CodeStar의 프로젝트 태그 작업

AWS CodeStar의 프로젝트에 태그를 연결할 수 있습니다. 태그는 프로젝트를 관리하는 데 도움이 될 수 있습니다. 예를 들어 베타 버전을 위해 조직에서 작업 중인 모든 프로젝트에 `Release` 키 및 `Beta` 값이 있는 태그를 추가할 수 있습니다.

프로젝트에 태그 추가

1. AWS CodeStar 콘솔에서 프로젝트를 열어 둔 상태로, 측면 탐색 모음에서 설정을 선택합니다.
2. 태그에서 편집을 선택합니다.
3. 키에 태그 이름을 입력합니다. 값에는 태그의 값을 입력합니다.
4. 선택 사항: 태그 추가를 선택하여 태그를 추가합니다.
5. 태그 추가를 완료했으면 저장을 선택합니다.

프로젝트에서 태그 제거

1. AWS CodeStar 콘솔에서 프로젝트를 열어 둔 상태로, 측면 탐색 모음에서 설정을 선택합니다.
2. 태그에서 편집을 선택합니다.
3. 태그에서 제거할 태그를 찾은 다음 태그 제거를 선택합니다.
4. Save를 선택합니다.

프로젝트 태그 목록 가져오기

AWS CLI를 사용하여 AWS CodeStar list-tags-for-project 명령을 실행하고 프로젝트 이름을 지정합니다.

```
aws codestar list-tags-for-project --id my-first-projec
```

성공하면 태그 목록이 다음과 비슷한 출력에 표시됩니다.

```
{
  "tags": {
    "Release": "Beta"
  }
}
```

AWS CodeStar 프로젝트 삭제

프로젝트가 더 이상 필요하지 않은 경우 AWS에서 추가 요금이 발생하지 않도록 해당 프로젝트와 그 리소스를 삭제할 수 있습니다. 프로젝트를 삭제하면 해당 프로젝트에서 모든 팀원이 제거됩니다. 이들의 IAM 사용자에서 해당 프로젝트 역할이 제거되지만, AWS CodeStar에서 이들의 사용자 프로필은 변경되지 않습니다. AWS CodeStar 콘솔 또는 AWS CLI를 사용하여 프로젝트를 삭제할 수 있습니다. 프로젝트를 삭제하려면 AWS CodeStar 서비스 역할인 `aws-codestar-service-role`이 필요합니다. 이 역할은 수정되지 않고 AWS CodeStar에서 수임되어야 합니다.

Important

AWS CodeStar에서의 프로젝트 삭제는 취소할 수 없습니다. 기본적으로 다음을 포함한 해당 프로젝트의 모든 AWS 리소스는 AWS 계정에서 삭제됩니다.

- 프로젝트의 CodeCommit 리포지토리와 해당 리포지토리에 저장된 모든 항목입니다.
- AWS CodeStar 프로젝트 역할과 해당 프로젝트 및 해당 리소스에 대해 구성된 관련 IAM 정책입니다.
- 프로젝트에 대해 생성된 모든 Amazon EC2 인스턴스입니다.
- 다음과 같은 배포 애플리케이션 및 관련 리소스
  - CodeDeploy 애플리케이션 및 관련 배포 그룹입니다.
  - AWS Lambda 함수 및 관련 API Gateway API입니다.
  - AWS Elastic Beanstalk 애플리케이션 및 관련 환경입니다.
- CodePipeline의 프로젝트에 대한 지속적 배포 파이프라인입니다.
- 프로젝트와 연결된 AWS CloudFormation 스택입니다.
- AWS CodeStar 콘솔로 생성한 모든 AWS Cloud9 개발 환경입니다. 커밋하지 않은 환경 내 모든 코드 변경 사항은 취소됩니다.

프로젝트와 함께 모든 프로젝트 리소스를 삭제하려면 리소스 삭제 확인란을 선택합니다. 이 옵션의 선택을 취소하면, AWS CodeStar에서 프로젝트가 삭제되고 IAM에서 해당 리소스에 액세스할 수 있는 프로젝트 역할이 삭제되지만, 다른 모든 리소스는 유지됩니다. AWS에서 이러한 리소스에 대한 요금이 계속 발생할 수 있습니다. 이러한 리소스 중 하나 이상을 더 이상 사용하지 않기로 결정한 경우, 이를 수동으로 삭제해야 합니다. 자세한 내용은 프로젝트 삭제: AWS CodeStar 프로젝트가 삭제되었지만 리소스가 아직 남아 있습니다. 섹션을 참조하세요.

프로젝트를 삭제할 때 리소스를 유지하기로 결정한 경우, 프로젝트 세부 정보 페이지에서 리소스 목록을 복사하는 것이 좋습니다. 이러한 방법으로 프로젝트가 더 이상 존재하지 않더라도 보관한 모든 리소스에 대한 레코드를 보유할 수 있습니다.

주제

AWS CodeStar에서 프로젝트 삭제(콘솔)

AWS CodeStar 콘솔을 사용하여 프로젝트를 삭제할 수 있습니다.

AWS CodeStar에서 프로젝트를 삭제하려면

1. https://console.aws.amazon.com/codestar/에서 AWS CodeStar 콘솔을 엽니다.
2. 탐색 창에서 프로젝트를 선택합니다.
3. 삭제할 프로젝트를 선택하고 삭제를 선택합니다.

   프로젝트를 열고 콘솔 왼쪽의 탐색 창에서 설정을 선택합니다. 프로젝트 세부 정보 페이지에서 [Delete project]를 선택합니다.
4. 삭제 확인 페이지에서 delete를 입력합니다. 프로젝트 리소스를 삭제하려면 리소스 삭제를 선택한 상태로 유지하십시오. 삭제를 선택합니다.

   프로젝트를 삭제하는 데 몇 분이 걸릴 수 있습니다. 삭제한 후에는 AWS CodeStar 콘솔의 프로젝트 목록에 해당 프로젝트가 더 이상 표시되지 않습니다.

> **Important**
>
> 프로젝트가 GitHub 리포지토리 또는 Atlassian JIRA의 문제 같은 AWS 외부의 리소스를 사용하는 경우, 해당 리소스는 확인란을 선택하더라도 삭제되지 않습니다.
> AWS CodeStar 관리형 정책이 IAM 사용자가 아닌 역할에 수동으로 연결돼 있다면 프로젝트를 삭제할 수 없습니다. 프로젝트의 관리형 정책을 연합된 사용자의 역할에 연결했다면, 정책을 분리해야 프로젝트를 삭제할 수 있습니다. 자세한 내용은 ??? 섹션을 참조하세요.

AWS CodeStar에서 프로젝트 삭제(AWS CLI)

AWS CLI를 사용하여 프로젝트를 삭제할 수 있습니다.

AWS CodeStar에서 프로젝트를 삭제하려면

1. 터미널(Linux, macOS 또는 Unix) 또는 명령 프롬프트(Windows)에서 프로젝트 이름을 포함하는 delete-project 명령을 실행합니다. 예를 들어, ID가 *my-2nd-project*인 프로젝트를 삭제하려면:

   ```
   aws codestar delete-project --id my-2nd-project
   ```

   다음과 비슷한 출력이 반환됩니다.

   ```
   {
       "projectArn":"arn:aws:codestar:us-east-2:111111111111:project/my-2nd-project"
   }
   ```

   프로젝트는 즉시 삭제되지 않습니다.

2. 프로젝트 이름을 포함하여 describe-project 명령을 실행합니다. 예를 들어, ID가 *my-2nd-project*인 프로젝트 상태를 확인하려면:

   ```
   
       aws codestar describe-project --id my-2nd-project
   
   ```

   프로젝트가 아직 삭제되지 않은 경우 이 명령은 다음과 유사한 출력을 반환합니다.

   ```
   {
       "name": "my project",
       "id": "my-2nd-project",
       "arn": "arn:aws:codestar:us-west-2:123456789012:project/my-2nd-project",
       "description": "My second CodeStar project.",
       "createdTimeStamp": 1572547510.128,
       "status": {
           "state": "CreateComplete"
       }
   }
   ```

프로젝트가 삭제된 경우 이 명령은 다음과 유사한 출력을 반환합니다.

```
   An error occurred (ProjectNotFoundException) when calling the DescribeProject
operation: The project ID was not found: my-2nd-project. Make sure that the
project ID is correct and then try again.
```

3. list-projects 명령을 실행하고 삭제된 프로젝트가 AWS 계정과 연결된 프로젝트 목록에 더 이상 표시되지 않는지 확인합니다.

```
aws codestar list-projects
```

AWS CodeStar 팀 작업

개발 프로젝트를 만든 후 다른 사람과 협력하기 위해 이들에게 액세스 권한을 부여합니다. AWS CodeStar에서 각 프로젝트에는 프로젝트 팀이 존재합니다. 사용자는 여러 AWS CodeStar 프로젝트에 속할 수 있고, 각각 서로 다른 AWS CodeStar 역할(따라서 서로 다른 권한)을 가질 수 있습니다. AWS CodeStar 콘솔에서 사용자는 AWS 계정과 연결된 모든 프로젝트를 볼 수 있지만, 자신이 팀원인 프로젝트만 보고 작업할 수 있습니다.

팀원은 자신의 대화명을 선택할 수 있습니다. 또한 다른 팀원이 연락할 수 있도록 이메일 주소를 추가할 수 있습니다. 소유자가 아닌 팀원은 프로젝트에 대한 자신의 AWS CodeStar 역할을 변경할 수 없습니다.

AWS CodeStar에서 각 프로젝트에는 다음 세 가지 역할이 있습니다.

AWS CodeStar 프로젝트의 역할 및 권한

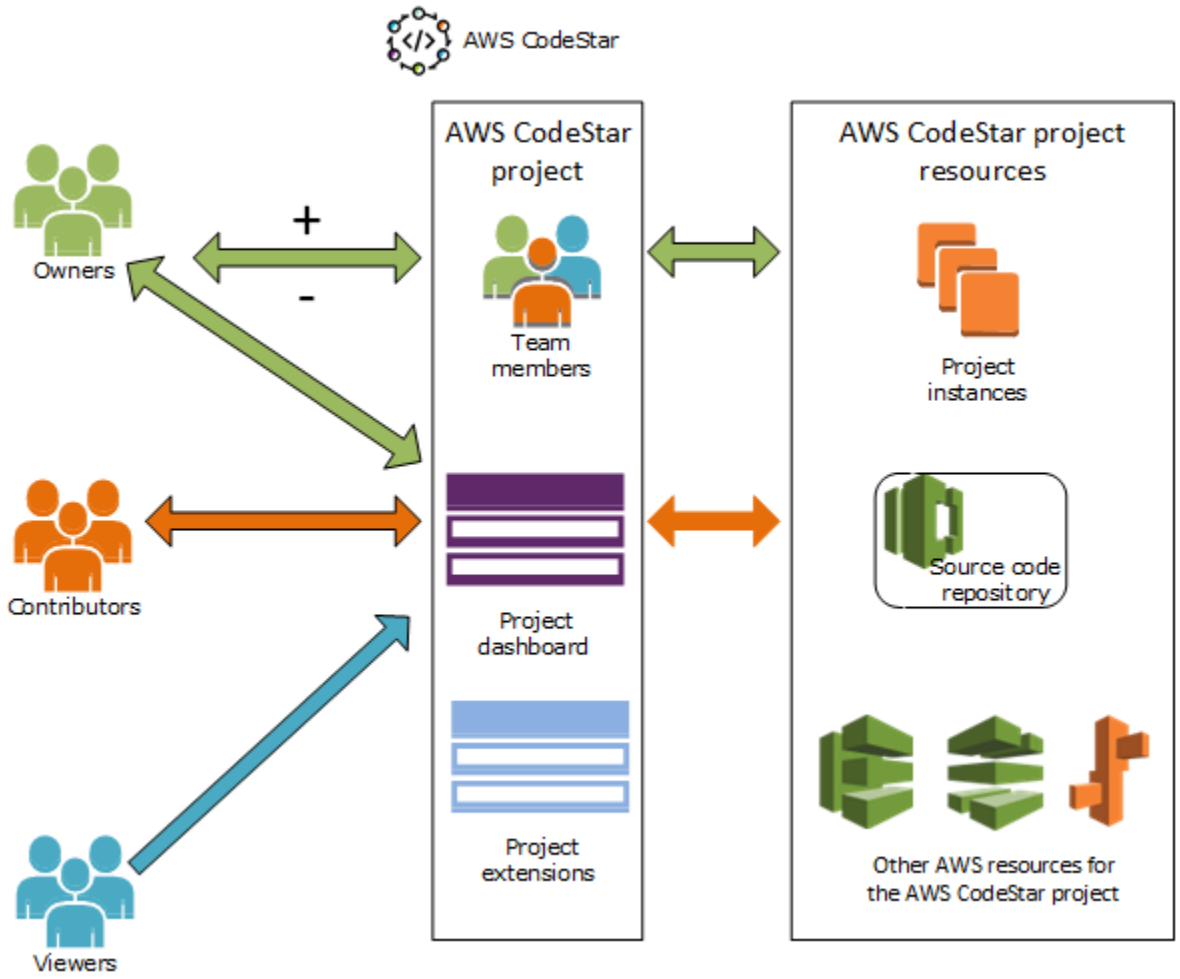
역할 이름	프로젝트 대시보드 및 상태 보기	프로젝트 리소스 추가/제거/액세스	팀원 추가/제거	프로젝트 삭제
Owner	x	x	x	x
기고자	x	x		
최종 사용자	x			

- 소유자: 다른 팀원을 추가 및 제거하고, 코드가 CodeCommit에 저장되어 있는 경우 프로젝트 리포지토리에 코드를 제공하고, 다른 팀원에게 Linux를 실행하는, 프로젝트와 연결된 모든 Amazon EC2 인스턴스에 대한 원격 액세스 권한을 부여하거나 거부하고, 프로젝트 대시보드를 구성하고, 프로젝트를 삭제할 수 있습니다.
- 기고자: JIRA 타일 등의 대시보드 리소스를 추가 및 제거하고, 코드가 CodeCommit에 저장되어 있는 경우 프로젝트 리포지토리에 코드를 제공하고, 대시보드와 완전히 상호 작용할 수 있습니다. 팀원을 추가하거나 제거하고, 리소스에 대한 원격 액세스 권한을 부여하거나 거부하고, 프로젝트를 삭제할 수는 없습니다. 대부분의 팀원에 대해 이 역할을 선택해야 합니다.
- 최종 사용자: 프로젝트 대시보드, 코드(코드가 CodeCommit에 저장되어 있는 경우)를 보고 대시보드 타일에서 프로젝트와 해당 리소스의 상태를 볼 수 있습니다.

> **Important**
>
> 프로젝트가 GitHub 리포지토리 또는 Atlassian JIRA의 문제 같은 AWS 외부의 리소스를 사용하는 경우, 해당 리소스에 대한 액세스는 AWS CodeStar가 아닌 리소스 제공자가 제어합니다. 자세한 내용은 리소스 공급자 설명서를 참조하십시오.
>
> AWS CodeStar 프로젝트에 액세스할 수 있는 사용자는 AWS CodeStar 콘솔을 사용하여 AWS 외부에 있지만 해당 프로젝트와 관련된 리소스에 액세스할 수 있습니다.
>
> AWS CodeStar는 팀원이 프로젝트에 관한 어떠한 관련 AWS Cloud9 개발 환경에도 참가하도록 자동으로 허용하지 않습니다. 팀원이 공유 환경에 참여하도록 허용하는 방법은 프로젝트 팀원과 AWS Cloud9 환경 공유 단원을 참조하십시오.

IAM 정책은 각 프로젝트 역할과 연결되어 있습니다. 이 정책은 해당 리소스를 반영하도록 프로젝트에 대해 사용자 지정되어 있습니다. 이러한 정책에 대한 자세한 내용은 AWS CodeStarID 기반 정책 예제을 참조하십시오.

다음 다이어그램에 각 역할과 AWS CodeStar 프로젝트 간의 관계가 나와 있습니다.



주제

- AWS CodeStar 프로젝트에 팀원 추가
- AWS CodeStar 팀원의 권한 관리.
- AWS CodeStar 프로젝트에서 팀원 제거

AWS CodeStar 프로젝트에 팀원 추가

AWS CodeStar 프로젝트에서 소유자 역할이 있거나 `AWSCodeStarFullAccess` 정책을 IAM 사용자에 적용한 경우, 다른 IAM 사용자를 프로젝트 팀에 추가할 수 있습니다. 이는 AWS CodeStar 역할(소유자, 기고자, 최종 사용자)을 사용자에게 적용하는 간단한 프로세스입니다. 이러한 역할은 프로젝트별 역할이며 사용자 지정됩니다. 예를 들어 프로젝트 A의 기고자 팀원은 프로젝트 B의 기고자 팀원과는 다른 리소스에 대한 권한을 가질 수 있습니다. 팀원은 한 프로젝트에서 한 가지 역할만 가질 수 있습니다. 팀원을 추가한 후 해당 팀원은 역할에 정의된 수준에서 프로젝트와 즉시 상호 작용할 수 있습니다.

AWS CodeStar 역할 및 팀 멤버십의 이점은 다음과 같습니다.

- IAM에서 팀원에 대한 권한을 수동으로 구성할 필요가 없습니다.
- 프로젝트에 대한 팀원의 액세스 수준을 쉽게 변경할 수 있습니다.
- 사용자는 팀원인 경우에만 AWS CodeStar 콘솔에서 프로젝트에 액세스할 수 있습니다.
- 프로젝트에 대한 사용자 액세스는 역할로 정의됩니다.

팀 및 AWS CodeStar 역할에 대한 자세한 내용은 AWS CodeStar 팀 작업 및 AWS CodeStar 사용자 프로필 작업 단원을 참조하십시오.

프로젝트에 팀원을 추가하려면 해당 프로젝트에 대한 AWS CodeStar 소유자 역할이 있거나 `AWSCodeStarFullAccess` 정책이 있어야 합니다.

> **Important**
>
> 팀원을 추가해도 GitHub 리포지토리 또는 Atlassian JIRA 같은 AWS 외부에 있는 리소스에 대한 팀원의 액세스에는 영향을 미치지 않습니다. 이러한 액세스 권한은 AWS CodeStar가 아닌 리소스 공급자가 제어합니다. 자세한 내용은 리소스 공급자 설명서를 참조하십시오.
> AWS CodeStar 프로젝트에 액세스할 수 있는 사용자는 AWS CodeStar 콘솔을 사용하여 AWS 외부에 있지만 해당 프로젝트와 관련된 리소스에 액세스할 수 있습니다.
> 팀원을 프로젝트에 추가해도 프로젝트의 모든 관련 AWS Cloud9 개발 환경에 대한 구성원의 참여를 자동으로 허용하지는 않습니다. 팀원이 공유 환경에 참여하도록 허용하는 방법은 프로젝트 팀원과 AWS Cloud9 환경 공유 단원을 참조하십시오.
> 연합된 사용자에게 프로젝트에 대한 액세스를 부여하려면 연합된 사용자가 위임한 역할에 AWS CodeStar 소유자, 기고자 또는 최종 사용자 관리형 정책을 직접 연결해야 합니다. 자세한 내용은 연동 사용자 액세스: AWS CodeStar 섹션을 참조하세요.

주제

- 팀원 추가(콘솔)
- 팀원 추가 및 보기(AWS CLI)

팀원 추가(콘솔)

AWS CodeStar 콘솔을 이용해 프로젝트에 팀원을 추가할 수 있습니다. 추가할 사람의 IAM 사용자가 이미 있는 경우 해당 IAM 사용자를 직접 추가할 수 있습니다. 그렇지 않다면 프로젝트에 사람을 추가할 때 해당 인물에 대한 IAM 사용자를 생성하면 됩니다.

AWS CodeStar 프로젝트에 팀원을 추가하려면(콘솔)

1. https://console.aws.amazon.com/codestar/에서 AWS CodeStar 콘솔을 엽니다.
2. 탐색 창에서 프로젝트를 선택한 후 프로젝트를 선택합니다.
3. 프로젝트의 측면 탐색 모음에서 팀을 선택합니다.
4. [Team members] 페이지에서 [Add team member]를 선택합니다.
5. 사용자 선택에서 다음 중 하나를 수행합니다.

   - 추가할 사람의 IAM 사용자가 이미 있다면, 목록에서 해당 IAM 사용자 이름을 선택합니다.

   > **Note**
   >
   > 다른 AWS CodeStar 프로젝트에 이미 추가한 사용자는 기존 AWS CodeStar 사용자 목록에 표시됩니다.

   프로젝트 역할에서 이 사용자에게 부여할 AWS CodeStar 역할(소유자, 기고자, 최종 사용자)을 선택합니다. 이는 프로젝트의 소유자만 변경할 수 있는 AWS CodeStar 프로젝트 수준 역할입니다. IAM 사용자에게 적용하면 이 역할은 AWS CodeStar 프로젝트 리소스에 액세스하는 데 필요한 모든 권한을 제공합니다. 이는 IAM에서 CodeCommit에 저장된 코드에 대한 Git 자격 증명을 만들고 관리하거나 IAM 사용자의 Amazon EC2 SSH 키를 업로드하는 데 필요한 정책을 적용합니다.

   > **Important**
   >
   > IAM 사용자의 표시 이름 또는 이메일 정보를 입력하거나 변경할 수 없습니다. 해당 사용자로 콘솔에 로그인한 경우에만 가능합니다. 자세한 내용은 AWS CodeStar 사용자 프로필에 대한 표시 정보 관리 섹션을 참조하세요.

   팀원 추가를 선택합니다.

- 프로젝트에 추가할 사람의 IAM 사용자가 없다면, 새 IAM 사용자 생성을 선택합니다. 새 IAM 사용자를 생성할 수 있는 IAM 콘솔로 리디렉션됩니다. 자세한 내용은 IAM 사용 설명서의 IAM 사용자 생성을 참조하십시오. IAM 사용자를 생성한 후 AWS CodeStar 콘솔로 돌아가 사용자 목록을 새로 고치고 드롭다운 목록에서 생성한 IAM 사용자를 선택합니다. 이 새 사용자에게 적용할 AWS CodeStar 표시 이름, 이메일 주소, 프로젝트 역할을 입력한 다음 팀원 추가를 선택합니다.

> **Note**
>
> 관리하기 쉽도록 하나 이상의 사용자에 프로젝트의 소유자 역할이 할당되어 있어야 합니다.

6. 새 팀원에게 다음 정보를 보냅니다.

   - AWS CodeStar 프로젝트의 연결 정보
   - 소스 코드가 CodeCommit에 저장되어 있는 경우 로컬 컴퓨터에서 Git 자격 증명을 사용하여 CodeCommit 리포지토리에 액세스하도록 설정하는 방법에 대한 지침입니다.
   - AWS CodeStar 사용자 프로필 작업 에 설명된 대로 사용자가 표시 이름, 이메일 주소, 퍼블릭 Amazon EC2 SSH 키를 관리하는 방법에 대한 정보입니다.
   - 일회용 암호와 연결 정보(사용자가 AWS를 처음 사용하며 해당 사용자의 IAM 사용자를 만든 경우) 암호는 사용자가 처음 로그인하면 만료됩니다. 사용자는 새 암호를 선택해야 합니다.

팀원 추가 및 보기(AWS CLI)

AWS CLI를 사용하여 프로젝트 팀에 팀원을 추가할 수 있습니다. 프로젝트의 모든 팀원에 대한 정보를 볼 수도 있습니다.

팀원을 추가하려면

1. 터미널 또는 명령 창을 엽니다.
2. associate-team-member 명령을 `--project-id`, `-user-arn` 및 `--project-role` 파라미터와 함께 실행합니다. `--remote-access-allowed` 또는 `--no-remote-access-allowed` 파라미터를 포함시켜 사용자가 프로젝트 인스턴스에 원격 액세스할 수 있는지 여부도 지정할 수 있습니다. 예:

```
aws codestar associate-team-member --project-id my-first-projec --user-arn
 arn:aws:iam:111111111111:user/Jane_Doe --project-role Contributor --remote-access-
allowed
```

이 명령은 출력을 반환하지 않습니다.

모든 팀원을 보려면(AWS CLI)

1. 터미널 또는 명령 창을 엽니다.
2. list-team-members 명령을 `--project-id` 파라미터와 함께 실행합니다. 예:

```
aws codestar list-team-members --project-id my-first-projec
```

다음과 비슷한 출력이 반환됩니다.

```
{
    "teamMembers":[

 {"projectRole":"Owner","remoteAccessAllowed":true,"userArn":"arn:aws:iam::111111111111:use
Mary_Major"},

 {"projectRole":"Contributor","remoteAccessAllowed":true,"userArn":"arn:aws:iam::1111111111
Jane_Doe"},

 {"projectRole":"Contributor","remoteAccessAllowed":true,"userArn":"arn:aws:iam::1111111111
John_Doe"},

 {"projectRole":"Viewer","remoteAccessAllowed":false,"userArn":"arn:aws:iam::111111111111:u
John_Stiles"}
 ]
}
```

AWS CodeStar 팀원의 권한 관리.

팀원의 AWS CodeStar 역할을 변경하여 이들의 권한을 변경합니다. AWS CodeStar 프로젝트에서 각 팀원은 하나의 역할에만 할당할 수 있지만, 많은 사용자가 동일한 역할에 할당할 수 있습니다. AWS CodeStar 콘솔이나 AWS CLI를 사용해 권한을 관리할 수 있습니다.

> **Important**
>
> 팀원의 역할을 변경하려면 해당 프로젝트에 대한 AWS CodeStar 소유자 역할이 있거나 `AWSCodeStarFullAccess` 정책을 적용해야 합니다.
> 팀원의 권한을 변경해도 GitHub 리포지토리 또는 Atlassian JIRA 같은 AWS 외부에 있는 리소스에 대한 팀원의 액세스에는 영향을 미치지 않습니다. 이러한 액세스 권한은 AWS CodeStar가 아닌 리소스 공급자가 제어합니다. 자세한 내용은 리소스 공급자 설명서를 참조하십시오.
> AWS CodeStar 프로젝트에 액세스할 수 있는 사용자는 AWS CodeStar 콘솔을 사용하여 AWS 외부에 있지만 해당 프로젝트와 관련된 리소스에 액세스할 수 있습니다.
> 프로젝트 팀 구성원의 역할을 변경해도 프로젝트에 대한 AWS Cloud9 개발 환경에 구성원의 참여를 자동으로 허용 또는 방지하지 않습니다. 팀원이 공유 환경에 참여하도록 허용하는 방법은 프로젝트 팀원과 AWS Cloud9 환경 공유 단원을 참조하십시오.

사용자에게 프로젝트와 연결된 모든 Amazon EC2 Linux 인스턴스에 원격으로 액세스할 권한을 부여할 수도 있습니다. 이 권한을 부여한 후 사용자는 모든 팀 프로젝트에서 자신의 AWS CodeStar 사용자 프로필과 연결할 SSH 퍼블릭 키를 업로드해야 합니다. Linux 인스턴스에 연결하려면 사용자는 로컬 컴퓨터에 구성된 SSH와 프라이빗 키가 있어야 합니다.

주제

팀 권한 관리(콘솔)

AWS CodeStar 콘솔을 이용해 팀원의 역할을 관리할 수 있습니다. 또한 팀원이 프로젝트와 연결된 Amazon EC2 인스턴스에 원격 액세스할 수 있는지 여부를 관리할 수 있습니다.

팀원의 역할을 변경하려면

1. https://console.aws.amazon.com/codestar/에서 AWS CodeStar 콘솔을 엽니다.
2. 탐색 창에서 프로젝트를 선택한 후 프로젝트를 선택합니다.
3. 프로젝트의 측면 탐색 모음에서 팀을 선택합니다.
4. 팀원 페이지에서 팀원을 선택한 다음 편집을 선택합니다.
5. 프로젝트 역할에서 이 사용자에게 부여할 AWS CodeStar 역할(소유자, 기고자, 최종 사용자)을 선택합니다.

AWS CodeStar 역할과 권한에 대한 자세한 내용은 AWS CodeStar 팀 작업 단원을 참조하십시오.

팀원 편집을 선택합니다.

팀원에 Amazon EC2 인스턴스에 대한 원격 액세스 권한을 부여하려면

1. https://console.aws.amazon.com/codestar/에서 AWS CodeStar 콘솔을 엽니다.
2. 탐색 창에서 프로젝트를 선택한 후 프로젝트를 선택합니다.
3. 프로젝트의 측면 탐색 모음에서 팀을 선택합니다.
4. 팀원 페이지에서 팀원을 선택한 다음 편집을 선택합니다.
5. 프로젝트 인스턴스에 대한 SSH 액세스 허용을 선택한 다음 팀원 편집을 선택합니다.
6. (선택 사항) 팀원에게 자신의 AWS CodeStar 사용자의 SSH 퍼블릭 키를 아직 업로드하지 않은 경우 이를 업로드해야 한다고 알립니다. 자세한 내용은 AWS CodeStar 사용자 프로필에 퍼블릭 키 추가 섹션을 참조하세요.

팀 권한 관리(AWS CLI)

AWS CLI를 사용하여 팀원에 할당된 프로젝트 역할을 관리할 수 있습니다. 동일한 AWS CLI 명령을 사용하여 팀원이 프로젝트와 연결된 Amazon EC2 인스턴스에 원격 액세스할 수 있는지 여부를 관리할 수 있습니다.

팀원의 권한을 관리하려면

1. 터미널 또는 명령 창을 엽니다.
2. update-team-member 명령을 `--project-id`, `-user-arn` 및 `--project-role` 파라미터와 함께 실행합니다. `--remote-access-allowed` 또는 `--no-remote-access-allowed` 파라미터를 포함시켜 사용자가 프로젝트 인스턴스에 원격 액세스할 수 있는지 여부도 지정할 수 있습니다. 예를 들어 John_Doe라는 IAM 사용자의 프로젝트 역할을 업데이트하고 이 사용자의 권한을 프로젝트 Amazon EC2 인스턴스에 대한 원격 액세스 권한이 없는 최종 사용자 권한으로 변경하려면:

```
aws codestar update-team-member --project-id my-first-projec --user-arn
 arn:aws:iam:111111111111:user/John_Doe --project-role Viewer --no-remote-access-
allowed
```

다음과 비슷한 출력이 반환됩니다.

```
{
 "projectRole":"Viewer",
 "remoteAccessAllowed":false,
 "userArn":"arn:aws:iam::111111111111:user/John_Doe"
}
```

AWS CodeStar 프로젝트에서 팀원 제거

AWS CodeStar 프로젝트에서 제거한 이후에도 해당 사용자는 프로젝트 리포지토리의 커밋 이력에 여전히 표시되지만, CodeCommit 리포지토리 또는 다른 프로젝트 리소스(예: 프로젝트 파이프라인)에는 더 이상 액세스할 수 없습니다. (리소스에 대한 액세스 권한을 부여하는 다른 정책이 있는 IAM 사용자는 예외입니다.) 사용자는 프로젝트 대시보드에 액세스할 수 없고 AWS CodeStar 대시보드에서 사용자에게 표시되는 프로젝트 목록에 프로젝트가 더 이상 표시되지 않습니다. AWS CodeStar 콘솔이나 AWS CLI를 사용하여 프로젝트 팀에서 팀원을 제거할 수 있습니다.

> **Important**
>
> 프로젝트에서 팀원을 제거하면 프로젝트 Amazon EC2 인스턴스에 대한 원격 액세스가 거부되지만 사용자의 활성 SSH 세션은 닫히지 않습니다.
>
> 팀원을 제거해도 GitHub 리포지토리 또는 Atlassian JIRA 같은 AWS 외부에 있는 리소스에 대한 팀원의 액세스에는 영향을 미치지 않습니다. 이러한 액세스 권한은 AWS CodeStar가 아닌 리소스 공급자가 제어합니다. 자세한 내용은 리소스 공급자 설명서를 참조하십시오.
>
> 프로젝트에서 팀 구성원을 제거해도 자동으로 팀 구성원의 관련 AWS Cloud9 개발 환경을 삭제하거나 구성원이 초대된 관련 AWS Cloud9 개발 환경의 참여를 방지하지 않습니다. 개발 환경을 삭제하는 방법은 프로젝트에서 AWS Cloud9 환경 삭제 단원을 참조하십시오. 팀원의 공유 환경 참여를 방지하는 방법은 프로젝트 팀원과 AWS Cloud9 환경 공유 단원을 참조하십시오.

프로젝트에서 팀원을 제거하려면 해당 프로젝트에 대한 AWS CodeStar 소유자 역할이 있거나 `AWSCodeStarFullAccess` 정책을 계정에 적용해야 합니다.

주제

- 팀원 제거(콘솔)
- 팀원 제거(AWS CLI)

팀원 제거(콘솔)

AWS CodeStar 콘솔을 사용하여 프로젝트 팀에서 팀원을 제거할 수 있습니다.

프로젝트에서 팀원을 제거하려면

1. https://console.aws.amazon.com/codestar/에서 AWS CodeStar 콘솔을 엽니다.
2. 탐색 창에서 프로젝트를 선택한 후 프로젝트를 선택합니다.
3. 프로젝트의 측면 탐색 모음에서 팀을 선택합니다.
4. 팀원 페이지에서 팀원을 선택한 다음 제거를 선택합니다.

팀원 제거(AWS CLI)

AWS CLI를 사용하여 프로젝트 팀에서 팀원을 제거할 수 있습니다.

팀원을 제거하려면

1. 터미널 또는 명령 창을 엽니다.
2. disassociate-team-member 명령을 `--project-id` 및 `-user-arn`와 함께 실행합니다. 예:

```
aws codestar disassociate-team-member --project-id my-first-projec --user-arn
 arn:aws:iam:111111111111:user/John_Doe
```

다음과 비슷한 출력이 반환됩니다.

```
{
    "projectId": "my-first-projec",
    "userArn": "arn:aws:iam::111111111111:user/John_Doe"
}
```

AWS CodeStar 사용자 프로필 작업

AWS CodeStar 사용자 프로필은 IAM 사용자와 연결됩니다. 이 프로필에는 사용자가 속한 모든 AWS CodeStar 프로젝트에서 사용되는 표시 이름과 이메일 주소가 포함되어 있습니다. 프로필과 연결할 SSH 퍼블릭 키를 업로드할 수 있습니다. 이 퍼블릭 키는 사용자가 속한 AWS CodeStar 프로젝트에 연결된 Amazon EC2 인스턴스에 연결할 때 사용하는 SSH 퍼블릭/프라이빗 키 페어의 일부입니다.

> **Note**
>
> 이 주제의 정보는 AWS CodeStar 사용자 프로필에 대해서만 설명합니다. 프로젝트가 GitHub 리포지토리 또는 Atlassian JIRA의 문제 같은 AWS 외부의 리소스를 사용하는 경우, 리소스 공급자는 서로 다른 설정을 이용하는 자체 사용자 프로필을 사용할 수 있습니다. 자세한 내용은 리소스 공급자 설명서를 참조하십시오.

주제

AWS CodeStar 사용자 프로필에 대한 표시 정보 관리

AWS CodeStar 콘솔이나 AWS CLI를 이용해 사용자 프로필의 표시 이름과 이메일 주소를 변경할 수 있습니다. 사용자 프로필은 프로젝트에 따라 달라지지 않습니다. IAM 사용자와 연결되며, AWS 리전에서 사용자가 속한 모든 AWS CodeStar 프로젝트에 적용됩니다. 사용자가 여러 AWS 리전의 프로젝트에 속한 경우 리전별로 별도의 사용자 프로필이 있습니다.

AWS CodeStar 콘솔에서 본인의 사용자 프로필만 관리할 수 있습니다. `AWSCodeStarFullAccess` 정책이 있는 경우 AWS CLI를 사용하여 다른 프로필을 보고 관리할 수 있습니다.

> **Note**
>
> 이 주제의 정보는 AWS CodeStar 사용자 프로필에 대해서만 설명합니다. 프로젝트가 GitHub 리포지토리 또는 Atlassian JIRA의 문제 같은 AWS 외부의 리소스를 사용하는 경우, 리소스 공급자는 서로 다른 설정을 이용하는 자체 사용자 프로필을 사용할 수 있습니다. 자세한 내용은 리소스 공급자 설명서를 참조하십시오.

주제

사용자 프로필 관리(콘솔)

사용자가 팀원으로 속한 프로젝트로 이동한 후 프로필 정보를 변경하여 AWS CodeStar 콘솔에서 사용자 프로필을 관리할 수 있습니다. 사용자 프로필은 프로젝트가 아닌 사용자에 특정하므로 사용자 프로필을 변경하면 AWS 리전 내에서 사용자가 팀원으로 속한 모든 프로젝트에 변경 사항이 표시됩니다.

> **Important**
>
> 콘솔을 이용해 사용자에 대한 표시 정보를 변경하려면 해당 IAM 사용자로 로그인해야 합니다. 다른 사용자는 프로젝트에 대한 AWS CodeStar 소유자 역할이 있거나 `AWSCodeStarFullAccess` 정책이 적용된 경우에도 표시 정보를 변경할 수 없습니다.

AWS 리전 내의 모든 프로젝트에서 표시 정보를 변경하려면

1. https://console.aws.amazon.com/codestar/에서 AWS CodeStar 콘솔을 엽니다.
2. 탐색 창에서 프로젝트를 선택하고 자신이 팀원으로 있는 프로젝트를 선택합니다.
3. 프로젝트의 측면 탐색 모음에서 팀을 선택합니다.
4. 팀원 페이지에서 IAM 사용자를 선택한 다음 편집을 선택합니다.
5. 표시 이름, 이메일 주소 또는 둘 모두를 편집한 다음 팀원 편집을 선택합니다.

> **Note**
>
> 표시 이름과 이메일 주소가 필요합니다. 자세한 내용은 AWS CodeStar의 제한 값 섹션을 참조하세요.

사용자 프로필 관리(AWS CLI)

AWS CLI를 사용하여 AWS CodeStar에서 사용자 프로필을 생성하고 관리할 수 있습니다. 또한 AWS CLI를 사용하여 사용자 프로필 정보를 보고 AWS 리전에서 AWS 계정에 대해 구성된 모든 사용자 프로필을 볼 수 있습니다.

사용자 프로필을 생성, 관리 또는 확인하려는 리전에 대해 AWS 프로필이 구성되어 있는지 확인합니다.

사용자 프로필을 생성하려면

1. 터미널 또는 명령 창을 엽니다.
2. create-user-profile 명령을 `user-arn`, `display-name` 및 `email-address` 파라미터와 함께 실행합니다. 예:

```
aws codestar create-user-profile --user-arn arn:aws:iam:111111111111:user/
John_Stiles --display-name "John Stiles" --email-address "john_stiles@example.com"
```

다음과 비슷한 출력이 반환됩니다.

```
{
 "createdTimestamp":1.491439687681E9,"
 displayName":"John Stiles",
 "emailAddress":"john.stiles@example.com",
 "lastModifiedTimestamp":1.491439687681E9,
 "userArn":"arn:aws:iam::111111111111:user/Jane_Doe"
}
```

표시 정보를 보려면

1. 터미널 또는 명령 창을 엽니다.
2. describe-user-profile 명령을 `user-arn` 파라미터와 함께 실행합니다. 예:

```
aws codestar describe-user-profile --user-arn arn:aws:iam:111111111111:user/
Mary_Major
```

다음과 비슷한 출력이 반환됩니다.

```
{
 "createdTimestamp":1.490634364532E9,
 "displayName":"Mary Major",
 "emailAddress":"mary.major@example.com",
 "lastModifiedTimestamp":1.491001935261E9,
 "sshPublicKey":"EXAMPLE=",
 "userArn":"arn:aws:iam::111111111111:user/Mary_Major"
```

```
}
```

표시 정보를 변경하려면

1. 터미널 또는 명령 창을 엽니다.
2. update-user-profile 명령을 `user-arn` 파라미터와 변경할 프로필 파라미터(예: `display-name` 또는 `email-address`)와 함께 실행합니다. 예를 들어 표시 이름이 Jane Doe인 사용자가 표시 이름을 Jane Mary Doe로 변경하려고 하는 경우:

```
aws codestar update-user-profile --user-arn arn:aws:iam:111111111111:user/Jane_Doe
 --display-name "Jane Mary Doe"
```

다음과 비슷한 출력이 반환됩니다.

```
{
 "createdTimestamp":1.491439687681E9,
 "displayName":"Jane Mary Doe",
 "emailAddress":"jane.doe@example.com",
 "lastModifiedTimestamp":1.491442730598E9,
 "sshPublicKey":"EXAMPLE1",
 "userArn":"arn:aws:iam::111111111111:user/Jane_Doe"
}
```

AWS 계정에서 AWS 리전에 모든 사용자 프로필을 나열하려면

1. 터미널 또는 명령 창을 엽니다.
2. aws codestar list-user-profiles 명령을 실행합니다. 예:

```
aws codestar list-user-profiles
```

다음과 비슷한 출력이 반환됩니다.

```
{
  "userProfiles":[
 {
  "displayName":"Jane Doe",
  "emailAddress":"jane.doe@example.com",
  "sshPublicKey":"EXAMPLE1",
```

```
  "userArn":"arn:aws:iam::111111111111:user/Jane_Doe"
 },
 {
  "displayName":"John Doe",
  "emailAddress":"john.doe@example.com",
  "sshPublicKey":"EXAMPLE2",
  "userArn":"arn:aws:iam::111111111111:user/John_Doe"
 },
 {
  "displayName":"Mary Major",
  "emailAddress":"mary.major@example.com",
  "sshPublicKey":"EXAMPLE=",
  "userArn":"arn:aws:iam::111111111111:user/Mary_Major"
 },
 {
  "displayName":"John Stiles",
  "emailAddress":"john.stiles@example.com",
  "sshPublicKey":"",
  "userArn":"arn:aws:iam::111111111111:user/John_Stiles"
 }
  ]
}
```

AWS CodeStar 사용자 프로필에 퍼블릭 키 추가

퍼블릭 SSH 키를 사용자가 생성하고 관리하는 퍼블릭-프라이빗 키 페어의 일부로 업로드할 수 있습니다. 이 SSH 퍼블릭-프라이빗 키 페어를 사용하여 Linux를 실행하는 Amazon EC2 인스턴스에 액세스합니다. 프로젝트 소유자가 원격 액세스 권한을 부여한 경우 프로젝트에 연결된 인스턴스에만 액세스할 수 있습니다. AWS CodeStar 콘솔을 사용하거나 퍼블릭 키를 AWS CLI 관리할 수 있습니다.

Important

프로젝트 소유자는 AWS CodeStar 프로젝트 소유자, 기여자 및 최종 사용자에게 프로젝트의 Amazon EC2 인스턴스에 대한 SSH 액세스 권한을 부여할 수 있지만, 개인 (소유자, 기여자 또는 최종 사용자) 만 SSH 키를 설정할 수 있습니다. 이 작업을 하려면 사용자는 개인 소유자, 기고자 또는 최종 사용자로 로그인해야 합니다.
AWS CodeStar 환경의 SSH 키는 관리하지 않습니다. AWS Cloud9

주제

퍼블릭 키 관리(콘솔)

콘솔에서 공개-개인 키 쌍을 생성할 수는 없지만 로컬에서 만든 다음 콘솔을 통해 사용자 프로필의 일부로 추가 또는 관리할 수 있습니다. AWS CodeStar

퍼블릭 SSH 키를 관리하려면

1. 터미널 또는 Bash 에뮬레이터 창에서 ssh-keygen 명령을 실행하여 로컬 컴퓨터에 SSH 퍼블릭-프라이빗 키 페어를 생성합니다. Amazon EC2에서 허용되는 모든 형식으로 키를 생성할 수 있습니다. 허용되는 형식에 대한 자세한 내용은 Amazon EC2로 사용자의 퍼블릭 키 가져오기를 참조하십시오. OpenSSH 형식의 SSH-2 RSA이고 2048비트를 포함하는 키를 생성하는 것이 좋습니다. 퍼블릭 키는 .pub 확장명을 가진 파일로 저장됩니다.
2. https://console.aws.amazon.com/codestar/ 에서 AWS CodeStar 콘솔을 엽니다.

   사용자가 팀원으로 속한 프로젝트를 선택합니다.
3. 탐색 창에서 팀을 선택합니다.
4. 팀원 페이지에서 IAM 사용자의 이름을 찾은 다음 편집을 선택합니다.
5. 팀원 편집 페이지의 원격 액세스에서 프로젝트 인스턴스에 대한 SSH 액세스 허용을 활성화합니다.
6. SSH 퍼블릭 키 상자에 퍼블릭 키를 붙여 넣은 다음 팀원 편집을 선택합니다.

> **Note**
>
> 이 필드에서 이전 키를 삭제하고 새 키를 붙여넣어서 퍼블릭 키를 변경할 수 있습니다. 이 필드의 내용을 삭제한 다음 팀원 편집을 선택하여 퍼블릭 키를 삭제할 수 있습니다.

퍼블릭 키를 변경하거나 삭제하면 사용자 프로필이 변경됩니다. 즉, 프로젝트별 변경이 아닙니다. 키는 프로필과 연결되어 있으므로 원격 액세스 권한이 부여된 모든 프로젝트에서 키가 변경되거나 삭제됩니다.

퍼블릭 키를 삭제하면 원격 액세스 권한이 부여된 모든 프로젝트에서 Linux를 실행하는 Amazon EC2 인스턴스에 대한 액세스 권한이 제거됩니다. 하지만 해당 키를 사용하여 연 SSH 세션은 닫히지 않습니다. 열려 있는 세션을 모두 닫아야 합니다.

퍼블릭 키 관리(AWS CLI)

를 사용하여 사용자 프로필의 일부로 SSH 공개 키를 관리할 수 있습니다. AWS CLI

퍼블릭 키를 관리하려면

1. 터미널 또는 Bash 에뮬레이터 창에서 ssh-keygen 명령을 실행하여 로컬 컴퓨터에 SSH 퍼블릭-프라이빗 키 페어를 생성합니다. Amazon EC2에서 허용되는 모든 형식으로 키를 생성할 수 있습니다. 허용되는 형식에 대한 자세한 내용은 Amazon EC2로 사용자의 퍼블릭 키 가져오기를 참조하십시오. OpenSSH 형식의 SSH-2 RSA이고 2048비트를 포함하는 키를 생성하는 것이 좋습니다. 퍼블릭 키는 .pub 확장명을 가진 파일로 저장됩니다.
2. AWS CodeStar 사용자 프로필에서 SSH 퍼블릭 키를 추가하거나 변경하려면 파라미터와 함께 update-user-profile 명령을 실행합니다. `--ssh-public-key` 예:

```
aws codestar update-user-profile --user-arn arn:aws:iam:111111111111:user/Jane_Doe
 --ssh-key-id EXAMPLE1
```

다음과 비슷한 출력이 반환됩니다.

```
{
 "createdTimestamp":1.491439687681E9,
 "displayName":"Jane Doe",
 "emailAddress":"jane.doe@example.com",
 "lastModifiedTimestamp":1.491442730598E9,
 "sshPublicKey":"EXAMPLE1",
 "userArn":"arn:aws:iam::111111111111:user/Jane_Doe"
}
```

프라이빗 키를 사용하여 Amazon EC2 인스턴스에 연결

Amazon EC2 키 페어를 생성했는지 확인합니다. 에서 AWS CodeStar사용자 프로필에 공개 키를 추가합니다. 키 페어를 생성하는 방법은 4단계: AWS CodeStar 프로젝트에 대한 Amazon EC2 키 페어 만

들기 단원을 참조하십시오. 사용자 프로필에 퍼블릭 키를 추가하려면 이 주제의 앞부분에 나오는 지침을 참조하십시오.

프라이빗 키를 이용해 Amazon EC2 Linux 인스턴스에 연결하려면

1. AWS CodeStar 콘솔에서 프로젝트를 열고 탐색 창에서 프로젝트를 선택합니다.
2. 프로젝트 리소스에서 유형이 Amazon EC2이고 이름이 instance로 시작하는 행의 ARN 링크를 선택합니다.
3. Amazon EC2 콘솔에서 연결을 선택합니다.
4. 인스턴스에 연결 대화 상자의 지침을 따릅니다.

   사용자 이름에는 `ec2-user`를 사용하십시오. 잘못된 사용자 이름을 사용하면 인스턴스에 연결할 수 없습니다.

자세한 내용은 Amazon EC2 사용 설명서의 다음 리소스를 참조하십시오.

- SSH를 사용하여 Linux 인스턴스에 연결
- PuTTY를 사용하여 Windows에서 Linux 인스턴스에 연결
- 를 사용하여 Linux 인스턴스에 연결 MindTerm

AWS CodeStar의 보안

AWS에서 클라우드 보안을 가장 중요하게 생각합니다. 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 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드의 보안 - AWS는 AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호합니다. AWS는 또한 안전하게 사용할 수 있는 서비스를 제공합니다. 타사 감사자는 AWS 규정 준수 프로그램의 일환으로 보안 효과를 정기적으로 테스트하고 검증합니다. AWS CodeStar에 적용되는 규정 준수 프로그램에 대한 자세한 내용을 알아보려면 규정 준수 프로그램 제공 범위 내 AWS 서비스를 참조하세요.
- 클라우드 내 보안 – 귀하의 책임은 귀하가 사용하는 AWS 서비스에 의해 결정됩니다. 또한 귀하는 데이터의 민감도, 회사 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 AWS CodeStar 사용 시 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 주제에서는 보안 및 규정 준수 목표를 충족하도록 AWS CodeStar를 구성하는 방법을 보여줍니다. 또한 AWS CodeStar 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법을 알아봅니다.

AWS CodeStar에서 사용자 지정 정책을 만들고 권한 경계를 사용하는 경우, 작업 수행에 필요한 권한만 부여하고 대상 리소스에만 권한 범위를 좁혀 최소 권한 액세스를 보장하세요. 다른 프로젝트의 구성원이 프로젝트의 리소스에 액세스하는 것을 방지하려면 조직 구성원에게 AWS CodeStar 프로젝트별로 별도의 권한을 부여하세요. 가장 좋은 방법은 각 구성원에 대한 프로젝트 계정을 만든 다음 해당 계정에 역할 기반 액세스 권한을 할당하는 것입니다.

예를 들어, AWS Control Tower with AWS Organizations와 같은 서비스를 사용하여 DevOps 그룹의 각 개발자 역할에 대한 계정을 프로비저닝할 수 있습니다. 그런 다음 해당 계정에 권한을 할당할 수 있습니다. 전체 권한은 계정에 적용되지만 사용자는 프로젝트 외부 리소스에 대한 액세스 권한이 제한됩니다.

다중 계정 전략을 사용하여 AWS 리소스에 대한 최소 권한 액세스를 관리하는 방법에 대한 자세한 내용은 AWSControl Tower 사용 설명서의 랜딩 존을 위한 AWS 다중 계정 전략을 참조하십시오.

주제

- 데이터 보호 기능 AWS CodeStar

- Identity 및 Access Management에 대한 AWS CodeStar
- AWS CloudTrail을 사용하여 AWS CodeStar API 직접 호출 로깅
- AWS CodeStar의 규정 준수 확인
- AWS CodeStar의 복원성
- 의 인프라 보안 AWS CodeStar

데이터 보호 기능 AWS CodeStar

AWS 공동 책임 모델 의 데이터 보호에 적용됩니다 AWS CodeStar. 이 모델에 설명된 대로 AWS 는 모든 데이터를 실행하는 글로벌 인프라를 보호하는 역할을 AWS 클라우드합니다. 사용자는 인프라에서 호스팅되는 콘텐츠를 관리해야 합니다. 사용하는 AWS 서비스 의 보안 구성과 관리 작업에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 대한 자세한 내용은 데이터 프라이버시를 참조하십시오FAQ. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 공동 책임 모델 및AWS 보안 GDPR 블로그의 블로그 게시물을 참조하십시오.

데이터 보호를 위해 AWS 계정 자격 증명을 보호하고 개별 사용자에게 AWS IAM Identity Center 또는 AWS Identity and Access Management (IAM) 를 설정하는 것이 좋습니다. 이렇게 하면 개별 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정마다 다단계 인증 (MFA) 을 사용하십시오.
- SSLTLS/를 사용하여 AWS 리소스와 통신하세요. TLS1.2가 필요하고 TLS 1.3을 권장합니다.
- API를 사용하여 사용자 활동 로깅을 설정합니다 AWS CloudTrail.
- 포함된 모든 기본 보안 제어와 함께 AWS 암호화 솔루션을 사용하십시오 AWS 서비스.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용하세요.
- 명령줄 인터페이스 또는 API an을 AWS 통해 액세스할 때 FIPS 140-3개의 검증된 암호화 모듈이 필요한 경우 엔드포인트를 사용하십시오. FIPS 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 연방 정보 처리 표준 () 140-3을 참조하십시오. FIPS

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 양식 필드에 입력하지 않는 것이 좋습니다. 여기에는 콘솔, API AWS CLI, CodeStar 또는 다른 사용자와 AWS 서비스 함께 작업하는 경우가 포함됩니다. AWS SDKs 이름에 사용되는 태그 또는 자유 형식 텍스트 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL a를 제공하는 경우 해당 서버에 대한 요청을 URL 검증하기 위해 자격 증명 정보를 에 포함하지 않는 것이 좋습니다.

의 데이터 암호화 AWS CodeStar

기본적으로 프로젝트에 대해 저장하는 정보를 AWS CodeStar 암호화합니다. 프로젝트 이름, 설명 및 사용자 이메일과 같이 프로젝트 ID 이외의 모든 사항이 유휴 상태에서 암호화됩니다. 프로젝트에 IDs 개인 정보를 넣지 마세요. AWS CodeStar 또한 기본적으로 전송 중인 정보를 암호화합니다. 유휴 상태 암호화나 전송 중 암호화를 위해 고객이 수행할 작업은 없습니다.

Identity 및 Access Management에 대한 AWS CodeStar

AWS Identity and Access Management (IAM) 는 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어할 수 AWS 서비스 있도록 도와줍니다. IAM관리자는 AWS CodeStar 리소스를 사용할 수 있는 인증 (로그인) 및 권한 부여 (권한 보유) 를 받을 수 있는 사용자를 제어합니다. IAM추가 비용 없이 사용할 AWS 서비스 수 있습니다.

주제

고객

AWS Identity and Access Management (IAM) 를 사용하는 방법은 수행하는 작업에 따라 다릅니다. AWS CodeStar

서비스 사용자 - AWS CodeStar 서비스를 사용하여 작업을 수행하는 경우 관리자가 필요한 자격 증명과 권한을 제공합니다. 더 많은 AWS CodeStar 기능을 사용하여 작업을 수행함에 따라 추가 권한이 필요할 수 있습니다. 액세스 권한 관리 방식을 이해하면 적절한 권한을 관리자에게 요청할 수 있습니다. 에서 AWS CodeStar 기능에 액세스할 수 없는 경우 을 참조하십시오AWS CodeStarID 및 액세스 문제 해결.

서비스 관리자 — 회사에서 AWS CodeStar 리소스를 담당하고 있다면 전체 액세스 권한이 있을 것입니다 AWS CodeStar. 서비스 사용자가 액세스해야 하는 AWS CodeStar 기능과 리소스를 결정하는 것

은 여러분의 몫입니다. 그런 다음 IAM 관리자에게 서비스 사용자의 권한을 변경해 달라는 요청을 제출해야 합니다. 이 페이지의 정보를 검토하여 의 기본 개념을 IAM 이해하십시오. 회사에서 wth를 사용하는 방법에 대한 자세한 내용은 IAM AWS CodeStar 을 참조하십시오의 AWS CodeStar 작동 방식 IAM.

IAM관리자 — IAM 관리자인 경우 액세스 관리를 위한 정책을 작성하는 방법에 대해 자세히 알아보는 것이 좋습니다 AWS CodeStar. 에서 IAM 사용할 수 있는 AWS CodeStar ID 기반 정책의 예를 보려면 을 참조하십시오. AWS CodeStarID 기반 정책 예제

자격 증명을 통한 인증

인증은 ID 자격 증명을 AWS 사용하여 로그인하는 방법입니다. IAM사용자로서 또는 역할을 위임하여 인증 (로그인 AWS) 을 받아야 합니다. AWS 계정 루트 사용자 IAM

ID 소스를 통해 제공된 자격 증명을 사용하여 페더레이션 ID로 로그인할 수 있습니다. AWS AWS IAM Identity Center (IAMID 센터) 사용자, 회사의 싱글 사인온 인증, Google 또는 Facebook 자격 증명이 페더레이션 ID의 예입니다. 페더레이션 ID로 로그인하는 경우 관리자는 이전에 역할을 사용하여 ID 페더레이션을 설정했습니다. IAM 페더레이션을 AWS 사용하여 액세스하는 경우 간접적으로 역할을 수임하는 것입니다.

사용자 유형에 따라 AWS Management Console 또는 AWS 액세스 포털에 로그인할 수 있습니다. 로그인에 대한 자세한 내용은 AWS 로그인 사용 설명서의 내 로그인 방법을 참조하십시오. AWS AWS 계정

AWS 프로그래밍 방식으로 액세스하는 경우 자격 증명을 사용하여 요청에 암호로 서명할 수 있는 소프트웨어 개발 키트 (SDKCLI) 와 명령줄 인터페이스 () 가 AWS 제공됩니다. AWS 도구를 사용하지 않는 경우 요청에 직접 서명해야 합니다. 권장 방법을 사용하여 직접 요청에 서명하는 방법에 대한 자세한 내용은 사용 IAM설명서의 AWS API요청 서명을 참조하십시오.

사용하는 인증 방법에 상관없이 추가 보안 정보를 제공해야 할 수도 있습니다. 예를 들어, 계정 보안을 강화하기 위해 다단계 인증 (MFA) 을 사용할 AWS 것을 권장합니다. 자세한 내용은 사용 설명서의 다단계 인증 및 사용 AWS IAM Identity Center 설명서의 다단계 인증 사용 (MFA) 을 IAM 참조하십시오. AWS

AWS 계정 루트 사용자

계정을 AWS 계정만들 때는 먼저 계정의 모든 AWS 서비스 리소스에 대한 완전한 액세스 권한을 가진 하나의 로그인 ID로 시작합니다. 이 ID를 AWS 계정 루트 사용자라고 하며, 계정을 만들 때 사용한 이메일 주소와 비밀번호로 로그인하여 액세스할 수 있습니다. 일상적인 작업에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 보안 인증 정보를 보호하고 루트 사용자만 수행할 수 있는 작업을 수행하는 데 사용합니다. 루트 사용자로 로그인해야 하는 작업의 전체 목록은 사용 설명서의 루트 사용자 자격 증명이 필요한 작업을 참조하십시오. IAM

IAM사용자 및 그룹

사용자는 단일 IAM사용자 또는 애플리케이션에 대한 특정 권한을 AWS 계정 가진 사용자 내의 ID입니다. 가능하면 암호 및 액세스 키와 같은 장기 자격 증명을 가진 IAM 사용자를 만드는 대신 임시 자격 증명을 사용하는 것이 좋습니다. 하지만 특정 사용 사례에서 IAM 사용자의 장기 자격 증명이 필요한 경우에는 액세스 키를 교체하는 것이 좋습니다. 자세한 내용은 사용 설명서의 장기 자격 증명이 필요한 사용 사례에 대한 정기적인 액세스 키 IAM 교체를 참조하십시오.

IAM그룹은 IAM 사용자 컬렉션을 지정하는 ID입니다. 사용자는 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 이름을 지정한 IAMAdmins그룹을 만들고 해당 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 보안 인증 정보를 가지고 있지만, 역할은 임시 보안 인증만 제공합니다. 자세히 알아보려면 사용 설명서의 역할 대신 IAM 사용자를 만드는 시기를 참조하십시오. IAM

IAM역할

IAM역할은 특정 권한을 AWS 계정 가진 사용자 내의 ID입니다. IAM사용자와 비슷하지만 특정인과 관련이 있는 것은 아닙니다. 역할을 AWS Management Console 전환하여 에서 일시적으로 IAM 역할을 맡을 수 있습니다. AWS CLI or AWS API 작업을 호출하거나 사용자 지정을 사용하여 역할을 수임할 수 URL 있습니다. 역할 사용 방법에 대한 자세한 내용은 사용 IAM설명서의 IAM역할 사용을 참조하십시오.

IAM임시 자격 증명이 있는 역할은 다음과 같은 상황에서 유용합니다.

- 페더레이션 사용자 액세스 - 페더레이션 ID에 권한을 부여하려면 역할을 생성하고 해당 역할의 권한을 정의합니다. 페더레이션 ID가 인증되면 역할이 연결되고 역할에 정의된 권한이 부여됩니다. 페더레이션을 위한 역할에 대한 자세한 내용은 IAM사용 설명서의 타사 ID 제공자를 위한 역할 생성을 참조하십시오. IAMIdentity Center를 사용하는 경우 권한 집합을 구성합니다. ID가 인증된 후 액세스할 수 있는 대상을 제어하기 위해 IAM Identity Center는 권한 집합을 역할의 상관 관계와 연결합니다. IAM 권한 세트에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 권한 세트를 참조하세요.
- 임시 IAM 사용자 권한 - IAM 사용자 또는 역할이 역할을 맡아 특정 작업에 대해 일시적으로 다른 권한을 부여받을 수 있습니다. IAM
- 계정 간 액세스 - IAM 역할을 사용하여 다른 계정의 사용자 (신뢰할 수 있는 사용자) 가 계정의 리소스에 액세스하도록 허용할 수 있습니다. 역할은 계정 간 액세스를 부여하는 기본적인 방법입니다. 하

지만 일부 AWS 서비스경우에는 역할을 프록시로 사용하는 대신 정책을 리소스에 직접 연결할 수 있습니다. 계정 간 액세스에 대한 역할과 리소스 기반 정책 간의 차이점을 알아보려면 사용 설명서의 교차 계정 리소스 액세스를 참조하십시오. IAM IAM

- 서비스 간 액세스 — 일부는 다른 기능을 AWS 서비스 사용합니다. AWS 서비스예를 들어, 서비스를 호출하면 해당 서비스가 Amazon에서 애플리케이션을 EC2 실행하거나 Amazon S3에 객체를 저장하는 것이 일반적입니다. 서비스는직접적으로 호출하는 보안 주체의 권한을 사용하거나, 서비스 역할을 사용하거나, 또는 서비스 연결 역할을 사용하여 이 작업을 수행할 수 있습니다.
  - 전달 액세스 세션 (FAS) — IAM 사용자 또는 역할을 사용하여 작업을 수행하는 AWS경우 보안 주체로 간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS전화를 거는 주체의 권한을 다운스트림 서비스에 AWS 서비스 요청하라는 요청과 결합하여 사용합니다. AWS 서비스 FAS요청은 다른 서비스 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 서비스가 수신한 경우에만 이루어집니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS요청 시 적용되는 정책 세부 정보는 전달 액세스 세션을 참조하십시오.
  - 서비스 역할 - 서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하는 것으로 간주하는 IAM 역할입니다. IAM관리자는 내부에서 IAM 서비스 역할을 만들고, 수정하고, 삭제할 수 있습니다. 자세한 내용은 사용 설명서의 역할 만들기를 참조하여 권한을 위임하십시오IAM. AWS 서비스
  - 서비스 연결 역할 - 서비스 연결 역할은 에 연결된 서비스 역할 유형입니다. AWS 서비스서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은 사용자에게 AWS 계정 표시되며 해당 서비스가 소유합니다. IAM관리자는 서비스 연결 역할에 대한 권한을 볼 수 있지만 편집할 수는 없습니다.
- Amazon에서 실행 중인 애플리케이션 EC2 — IAM 역할을 사용하여 EC2 인스턴스에서 실행되고 AWS API 요청을 보내는 애플리케이션의 임시 자격 증명을 관리할 수 있습니다. AWS CLI EC2인스턴스 내에 액세스 키를 저장하는 것보다 이 방법이 더 좋습니다. EC2인스턴스에 AWS 역할을 할당하고 모든 애플리케이션에서 사용할 수 있게 하려면 인스턴스에 연결된 인스턴스 프로필을 만들어야 합니다. 인스턴스 프로필에는 역할이 포함되며, 이를 통해 EC2 인스턴스에서 실행 중인 프로그램이 임시 자격 증명을 얻을 수 있습니다. 자세한 내용은 사용 설명서의 IAM역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여를 IAM 참조하십시오.

IAM역할을 사용할지 IAM 사용자를 사용할지 알아보려면 사용 설명서의 IAM 역할 생성 시기 (사용자 대신) 를 IAM참조하십시오.

정책을 사용하여 액세스 관리

정책을 만들고 이를 AWS ID 또는 리소스에 AWS 연결하여 액세스를 제어할 수 있습니다. 정책은 ID 또는 리소스와 연결될 때 AWS 해당 권한을 정의하는 객체입니다. AWS 주도자 (사용자, 루트 사용자 또

는 역할 세션) 가 요청할 때 이러한 정책을 평가합니다. 정책에서 권한은 요청이 허용되거나 거부되는지를 결정합니다. 대부분의 정책은 JSON 문서로 AWS 저장됩니다. JSON정책 문서의 구조 및 내용에 대한 자세한 내용은 IAM사용 설명서의 JSON 정책 개요를 참조하십시오.

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

기본적으로, 사용자와 역할에는 어떠한 권한도 없습니다. IAM관리자는 IAM 정책을 생성하여 필요한 리소스에서 작업을 수행할 수 있는 권한을 사용자에게 부여할 수 있습니다. 그러면 관리자가 역할에 IAM 정책을 추가할 수 있으며, 사용자는 역할을 수임할 수 있습니다.

IAM정책은 작업을 수행하는 데 사용하는 방법에 관계없이 작업에 대한 권한을 정의합니다. 예를 들어, `iam:GetRole` 작업을 허용하는 정책이 있다고 가정합니다. 해당 정책을 사용하는 사용자는 AWS Management Console, AWS CLI, 또는 에서 역할 정보를 가져올 수 AWS API 있습니다.

자격 증명 기반 정책

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자와 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. ID 기반 정책을 만드는 방법을 알아보려면 사용 설명서의 IAM정책 생성을 참조하십시오. IAM

보안 인증 기반 정책은 인라인 정책 또는 관리형 정책으로 한층 더 분류할 수 있습니다. 인라인 정책은 단일 사용자, 그룹 또는 역할에 직접 포함됩니다. 관리형 정책은 내 여러 사용자, 그룹 및 역할에 연결할 수 있는 독립형 정책입니다. AWS 계정관리형 정책에는 AWS 관리형 정책과 고객 관리형 정책이 포함됩니다. 관리형 정책과 인라인 정책 중에서 선택하는 방법을 알아보려면 IAM사용 설명서의 관리형 정책과 인라인 정책 중 선택을 참조하십시오.

리소스 기반 정책

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

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

액세스 제어 목록 () ACLs

액세스 제어 목록 (ACLs) 은 리소스에 액세스할 수 있는 권한을 가진 주체 (계정 구성원, 사용자 또는 역할) 를 제어합니다. ACLs정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 JSON 비슷합니다.

지원하는 서비스의 VPC 예로는 Amazon S3와 Amazon이 ACLs 있습니다. AWS WAF자세한 내용은 Amazon 심플 스토리지 서비스 개발자 안내서의 액세스 제어 목록 (ACL) 개요를 참조하십시오. ACLs

기타 정책 유형

AWS 일반적이지 않은 추가 정책 유형을 지원합니다. 이러한 정책 타입은 더 일반적인 정책 유형에 따라 사용자에게 부여되는 최대 권한을 설정할 수 있습니다.

- 권한 경계 - 권한 경계는 ID 기반 정책이 IAM 엔티티 (IAM사용자 또는 역할) 에 부여할 수 있는 최대 권한을 설정하는 고급 기능입니다. 개체에 대한 권한 경계를 설정할 수 있습니다. 그 결과로 얻는 권한은 객체의 자격 증명 기반 정책과 그 권한 경계의 교집합입니다. `Principal` 필드에서 사용자나 역할을 지정하는 리소스 기반 정책은 권한 경계를 통해 제한되지 않습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 권한 경계에 대한 자세한 내용은 사용 IAM설명서의 IAM 엔티티에 대한 권한 경계를 참조하십시오.
- 서비스 제어 정책 (SCPs) - SCPs 조직 또는 OU (조직 구성 단위) 에 대한 최대 권한을 지정하는 JSON AWS Organizations정책입니다. AWS Organizations 기업이 소유한 여러 AWS 계정 개를 그룹화하고 중앙에서 관리하는 서비스입니다. 조직의 모든 기능을 사용하도록 설정하면 일부 또는 모든 계정에 서비스 제어 정책 (SCPs) 을 적용할 수 있습니다. 각 항목을 포함하여 구성원 계정의 엔티티에 대한 권한을 SCP AWS 계정 루트 사용자제한합니다. Organizations 및 SCPs 에 대한 자세한 내용은 AWS Organizations 사용 설명서의 서비스 제어 정책을 참조하십시오.
- 세션 정책 – 세션 정책은 역할 또는 페더레이션 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 결과적으로 얻는 세션의 권한은 사용자 또는 역할의 보안 인증 기반 정책의 교차와 세션 정책입니다. 또한 권한을 리소스 기반 정책에서 가져올 수도 있습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 자세한 내용은 IAM사용 설명서의 세션 정책을 참조하십시오.

여러 정책 유형

여러 정책 유형이 요청에 적용되는 경우, 결과 권한은 이해하기가 더 복잡합니다. 여러 정책 유형이 관련된 경우 요청을 허용할지 여부를 AWS 결정하는 방법을 알아보려면 IAM사용 설명서의 정책 평가 로직을 참조하십시오.

의 AWS CodeStar 작동 방식 IAM

액세스를 관리하는 IAM 데 사용하기 전에 먼저 사용할 수 있는 IAM 기능을 이해해야 AWS CodeStar 합니다. AWS CodeStar AWS CodeStar 및 기타 AWS 서비스가 어떻게 작동하는지 자세히 알아보려면 IAM사용 IAM 설명서에서 함께 작동하는AWS 서비스를 참조하십시오. IAM

주제

- AWS CodeStarID 기반 정책
- AWS CodeStar 리소스 기반 정책
- AWS CodeStar 태그 기반 권한 부여
- AWS CodeStar IAM역할
- IAM사용자 액세스: AWS CodeStar
- 연동 사용자 액세스: AWS CodeStar
- 다음과 같은 임시 자격 증명 사용 AWS CodeStar
- 서비스 연결 역할
- 서비스 역할

AWS CodeStarID 기반 정책

IAMID 기반 정책을 사용하면 허용 또는 거부된 작업과 리소스, 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. AWS CodeStar 사용자를 대신하여 여러 ID 기반 정책을 AWS CodeStar 생성하여 프로젝트 범위 내에서 리소스를 만들고 관리할 수 있도록 합니다. AWS CodeStar AWS CodeStar 특정 작업, 리소스 및 조건 키를 지원합니다. JSON정책에서 사용하는 모든 요소에 대해 알아보려면 사용 IAM설명서의 IAMJSON정책 요소 참조를 참조하십시오.

작업

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

정책 `Action` 요소는 JSON 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 작업을 설명합니다. 정책 작업은 일반적으로 관련 AWS API 작업과 이름이 같습니다. 일치하는 작업이 없는 권한 전용 작업과 같은 몇 가지 예외가 있습니다. API 정책에서 여러 작업이 필요한 몇 가지 작업도 있습니다. 이러한 추가 작업을 일컬어 종속 작업이라고 합니다.

연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함하십시오.

정책 조치는 조치 앞에 다음 접두사를 AWS CodeStar 사용합니다. `codestar:` 예를 들어, 지정된 IAM 사용자가 프로젝트 설명과 같은 AWS CodeStar 프로젝트 속성을 편집할 수 있도록 허용하려면 다음 정책 설명을 사용할 수 있습니다.

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codestar:UpdateProject"
      ],
      "Resource" : "arn:aws:codestar:us-east-2:project/my-first-projec"
    }
  ]
}
```

정책 문에는 `Action` 또는 `NotAction` 요소가 포함되어야 합니다. AWS CodeStar 이 서비스로 수행할 수 있는 작업을 설명하는 자체 작업 세트를 정의합니다.

명령문 하나에 여러 태스크를 지정하려면 다음과 같이 쉼표로 구분합니다.

```
"Action": [
      "codestar:action1",
      "codestar:action2"
```

와일드카드(*)를 사용하여 여러 작업을 지정할 수 있습니다. 예를 들어, `List`라는 단어로 시작하는 모든 태스크를 지정하려면 다음 태스크를 포함합니다.

```
"Action": "codestar:List*"
```

AWS CodeStar 작업 목록을 보려면 IAM사용 설명서의 정의된 AWS CodeStar 작업을 참조하십시오.

리소스

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

`Resource`JSON정책 요소는 작업이 적용되는 하나 또는 여러 개의 객체를 지정합니다. 문장에는 `Resource`또는 `NotResource`요소가 반드시 추가되어야 합니다. Amazon 리소스 이름 (ARN) 을 사용

하여 리소스를 지정하는 것이 가장 좋습니다. 리소스 수준 권한이라고 하는 특정 리소스 유형을 지원하는 작업에 대해 이 태스크를 수행할 수 있습니다.

작업 나열과 같이 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

```
"Resource": "*"
```

AWS CodeStar 프로젝트 리소스는 ARN 다음과 같습니다.

```
arn:aws:codestar:region:account:project/resource-specifier
```

형식에 대한 자세한 내용은 Amazon 리소스 이름 (ARNs) 및 AWS 서비스 네임스페이스를 참조하십시오. ARNs

예를 들어 다음은 해당 111111111111 지역의 AWS 계정에 *my-first-projec* 등록된 AWS CodeStar 프로젝트 이름을 지정합니다. AWS us-east-2

```
arn:aws:codestar:us-east-2:111111111111:project/my-first-projec
```

다음은 해당 111111111111 AWS 지역의 AWS 계정에 my-proj 등록된 이름으로 시작하는 모든 AWS CodeStar 프로젝트를 지정합니다us-east-2.

```
arn:aws:codestar:us-east-2:111111111111:project/my-proj*
```

프로젝트 목록과 같은 일부 AWS CodeStar 작업은 리소스에서 수행할 수 없습니다. 이러한 경우, 와일드카드(*)를 사용해야 합니다.

```
"LisProjects": "*"
```

AWS CodeStar 리소스 유형 및 해당 ARNs 유형의 목록을 보려면 IAM사용 설명서의 리소스 정의 기준을 참조하십시오. AWS CodeStar 각 리소스에 어떤 작업을 지정할 수 있는지 알아보려면 작업 정의 기준을 참조하십시오 AWS CodeStar. ARN

조건 키

AWS CodeStar 서비스별 조건 키는 제공하지 않지만 일부 글로벌 조건 키 사용은 지원합니다. 모든 AWS 글로벌 조건 키를 보려면 IAM사용 설명서의AWS 글로벌 조건 컨텍스트 키를 참조하십시오.

예시

AWS CodeStar ID 기반 정책의 예를 보려면 을 참조하십시오. AWS CodeStarID 기반 정책 예제

AWS CodeStar 리소스 기반 정책

AWS CodeStar 리소스 기반 정책을 지원하지 않습니다.

AWS CodeStar 태그 기반 권한 부여

AWS CodeStar 프로젝트에 태그를 첨부하거나 요청에 태그를 전달할 수 있습니다. AWS CodeStar 태그에 근거하여 액세스를 제어하려면 `codestar:ResourceTag/key-name`, `aws:RequestTag/key-name` 또는 `aws:TagKeys` 조건 키를 사용하여 정책의 조건 요소에 태그 정보를 제공합니다. AWS CodeStar 리소스 태깅에 대한 자세한 내용은 을 참조하십시오the section called “프로젝트 태그 작업”.

프로젝트의 태그를 기반으로 AWS CodeStar 프로젝트에 대한 액세스를 제한하는 ID 기반 정책의 예를 보려면 을 참조하십시오. 태그 기반으로 AWS CodeStar 프로젝트 보기

AWS CodeStar IAM역할

IAM역할은 AWS 계정에서 특정 권한을 가진 엔티티입니다.

사용자, 페더레이션 IAM사용자, 루트 사용자 또는 수임된 AWS CodeStar 역할로 사용할 수 있습니다. 적절한 권한을 가진 모든 사용자 유형은 AWS 리소스에 대한 프로젝트 권한을 관리할 수 있지만 사용자에 대한 IAM 프로젝트 권한은 자동으로 AWS CodeStar 관리합니다. IAM정책과 역할은 프로젝트 역할을 기반으로 해당 사용자에게 권한과 액세스 권한을 부여합니다. IAM콘솔을 사용하여 IAM 사용자에게 권한 AWS CodeStar 및 기타 권한을 할당하는 다른 정책을 만들 수 있습니다.

예를 들어 사용자에게 AWS CodeStar 프로젝트를 보는 것은 허용하나 변경하는 것은 허용하지 않도록 하고자 할 수 있습니다. 이 경우 뷰어 역할을 가진 AWS CodeStar 프로젝트에 IAM 사용자를 추가합니다. 모든 AWS CodeStar 프로젝트에는 프로젝트에 대한 액세스를 제어하는 데 도움이 되는 일련의 정책이 있습니다. 또한 액세스할 수 있는 사용자를 제어할 수 AWS CodeStar있습니다.

AWS CodeStar 액세스는 사용자와 페더레이션 IAM 사용자에 따라 다르게 처리됩니다. IAM사용자만 팀에 추가할 수 있습니다. IAM사용자에게 프로젝트 권한을 부여하려면 사용자를 프로젝트 팀에 추가하고 사용자에게 역할을 할당합니다. 연동 사용자에게 프로젝트 권한을 부여하려면 AWS CodeStar 프로젝트 역할의 관리형 정책을 연동 사용자의 역할에 수동으로 연결합니다.

이 표는 각 유형의 액세스에 적용 가능한 도구를 요약한 것입니다.

권한 기능	IAM사용자	페더레이션 사용자	루트 사용자
SSHAmazon EC2 및 Elastic Beanstalk 프로젝트의 원격 액세스를 위한 키 관리	✓		
AWS CodeCommit SSH액세스	✓		
IAM사용자 권한 관리: AWS CodeStar	✓		
수동으로 관리하는 프로젝트 권한		✓	✓
사용자는 팀원으로 프로젝트에 추가 가능	✓		

IAM사용자 액세스: AWS CodeStar

프로젝트에 사용자를 추가하고 IAM 사용자의 역할을 선택하면 적절한 정책이 IAM 사용자에게 자동으로 AWS CodeStar 적용됩니다. IAM사용자의 경우 에서 정책 또는 권한을 직접 연결하거나 관리할 필요가 없습니다IAM. AWS CodeStar 프로젝트에 IAM 사용자를 추가하는 방법에 대한 자세한 내용은 을 참조하십시오AWS CodeStar 프로젝트에 팀원 추가 . AWS CodeStar 프로젝트에서 IAM 사용자를 제거하는 방법에 대한 자세한 내용은 을 참조하십시오AWS CodeStar 프로젝트에서 팀원 제거 .

사용자에게 인라인 정책 연결 IAM

프로젝트에 사용자를 추가하면 사용자 역할과 일치하는 프로젝트의 관리형 정책을 AWS CodeStar 자동으로 연결합니다. 프로젝트의 AWS CodeStar 관리형 정책을 IAM 사용자에게 수동으로 연결해서는 안 됩니다. 를 제외하고는 AWS CodeStar 프로젝트에서 IAM 사용자 권한을 변경하는 정책을 첨부하지 않는 것이 좋습니다. `AWSCodeStarFullAccess` 자체 정책을 만들어 첨부하려는 경우 사용 IAM설명서의 IAMID 권한 추가 및 제거를 참조하십시오.

연동 사용자 액세스: AWS CodeStar

IAM사용자를 만들거나 루트 IAM 사용자를 사용하는 대신 기업 사용자 디렉토리, 웹 ID 공급자 또는 역할을 맡는 사용자의 사용자 ID를 사용할 수 있습니다. AWS Directory Service이러한 사용자를 연동 사용자라고 합니다.

AWS CodeStar 프로젝트 AWS CodeStar 수준 정책 및 권한에 설명된 관리형 정책을 사용자 역할에 수동으로 연결하여 연동 사용자에게 프로젝트 액세스 권한을 부여하십시오. IAM 프로젝트 리소스 및 역할을 AWS CodeStar 만든 후 소유자, 기여자 또는 최종 사용자 정책을 연결합니다. IAM

사전 조건:

- 자격 증명 공급자를 설정해야 합니다. 예를 들어 SAML ID 공급자를 설정하고 공급자를 통해 AWS 인증을 설정할 수 있습니다. ID 제공자 설정에 대한 자세한 내용은 ID 제공자 생성을 IAM 참조하십시오. SAML페더레이션에 대한 자세한 내용은 SAML2.0 기반 페더레이션에 대한 정보를 참조하십시오.
- 자격 증명 공급자를 통해 액세스를 요청할 때는, 연합된 사용자가 위임할 역할을 생성해야 합니다. 연동 사용자가 역할을 수임할 수 있도록 허용하는 역할에는 STS 신뢰 정책을 첨부해야 합니다. 자세한 내용은 사용 설명서의 연동 IAM사용자 및 역할을 참조하십시오.
- 프로젝트를 생성했고 AWS CodeStar 프로젝트 ID를 알고 있어야 합니다.

자격 증명 공급자를 위한 역할 생성에 대한 자세한 내용은 타사 자격 증명 공급자(연동)를 위한 역할 생성을 참조하십시오.

AWSCodeStarFullAccess 관리형 정책을 연동 사용자 역할에 연결

`AWSCodeStarFullAccess` 관리형 정책을 연결해 연합된 사용자에게 프로젝트 생성 권한을 부여합니다. 이 단계를 수행하려면 루트 사용자, 계정의 관리자 사용자 또는 연결된 `AdministratorAccess` 관리형 정책 또는 이와 동등한 권한을 가진 IAM 사용자 또는 연동 사용자로 콘솔에 로그인해야 합니다.

> **Note**
>
> 프로젝트를 생성해도 프로젝트 소유자 권한은 자동으로 적용되지 않습니다. 계정에 대한 관리 권한이 있는 역할을 사용하여 프로젝트의 AWS CodeStar 뷰어/기여자/소유자 관리형 정책을 연동 사용자의 역할에 연결합니다.에 설명된 대로 소유자 관리형 정책을 연결하십시오.

1. IAM콘솔을 엽니다. 탐색 창에서 Policies를 선택합니다.
2. 검색 필드에 `AWSCodeStarFullAccess`를 입력합니다. 정책 이름이 표시되며, 정책 유형은 AWS 관리형입니다. 정책을 확장하면 정책 설명의 권한을 확인할 수 있습니다.
3. 정책 옆에 있는 동그라미를 선택하고, 정책 작업에서 연결을 선택합니다.

4. 요약 페이지에서 연결된 개체 탭을 선택합니다. 연결을 선택합니다.
5. 정책 연결 페이지에서 검색 필드에 연합된 사용자의 역할을 입력해 필터링합니다. 역할 이름 옆에 있는 확인란을 선택하고 정책 연결을 선택합니다. 연결된 개체 탭에 새로운 연결이 표시됩니다.

프로젝트의 AWS CodeStar 뷰어/기여자/소유자 관리형 정책을 연동 사용자의 역할에 연결합니다.

관련된 소유자, 기고자, 최종 사용자 관리형 정책을 사용자 역할에 연결해 연합된 사용자 액세스를 프로젝트에 부여합니다. 관리형 정책은 적절한 수준의 권한을 제공합니다. IAM사용자와 달리 연동 사용자에 대한 관리형 정책은 수동으로 연결 및 분리해야 합니다. 이는 에서 팀 구성원에게 프로젝트 권한을 할당하는 것과 같습니다. AWS CodeStar이 단계를 수행하려면 루트 사용자, 계정의 관리자 사용자 또는 관련 `AdministratorAccess` 관리형 정책 또는 이와 동등한 정책을 사용하는 IAM 사용자 또는 연동 사용자로 콘솔에 로그인해야 합니다.

사전 조건:

- 연합된 사용자가 위임하는 역할을 만들거나 기존 역할이 있어야 합니다.
- 자신이 부여할 권한 수준을 알고 있어야 합니다. 소유자, 기고자, 최종 사용자 역할에 연결된 관리형 정책은 프로젝트에 역할 기반 권한을 제공합니다.
- AWS CodeStar 프로젝트가 생성되어 있어야 합니다. 프로젝트를 생성할 IAM 때까지는 관리형 정책을 에서 사용할 수 없습니다.

1. IAM콘솔을 엽니다. 탐색 창에서 Policies를 선택합니다.
2. 검색 필드에 프로젝트 ID를 입력합니다. 프로젝트와 일치하는 정책 이름이 표시되며, 정책 유형은 고객 관리형입니다. 정책을 확장하면 정책 설명의 권한을 확인할 수 있습니다.
3. 다음 관리형 정책 중 하나를 선택하십시오. 정책 옆에 있는 동그라미를 선택하고, 정책 작업에서 연결을 선택합니다.
4. 요약 페이지에서 연결된 개체 탭을 선택합니다. 연결을 선택합니다.
5. 정책 연결 페이지에서 검색 필드에 연합된 사용자의 역할을 입력해 필터링합니다. 역할 이름 옆에 있는 확인란을 선택하고 정책 연결을 선택합니다. 연결된 개체 탭에 새로운 연결이 표시됩니다.

연동 사용자 역할에서 AWS CodeStar 관리형 정책을 분리합니다.

AWS CodeStar 프로젝트를 삭제하기 전에 연동 사용자 역할에 연결한 모든 관리형 정책을 수동으로 분리해야 합니다. 이 단계를 수행하려면 루트 사용자, 계정의 관리자 사용자 또는 연결된 `AdministratorAccess` 관리형 정책 또는 이와 동등한 권한을 가진 IAM 사용자 또는 페더레이션 사용자로 콘솔에 로그인해야 합니다.

1. IAM콘솔을 엽니다. 탐색 창에서 Policies를 선택합니다.
2. 검색 필드에 프로젝트 ID를 입력합니다.
3. 정책 옆에 있는 동그라미를 선택하고, 정책 작업에서 연결을 선택합니다.
4. 요약 페이지에서 연결된 개체 탭을 선택합니다.
5. 검색 필드에 연합된 사용자의 역할을 입력해 필터링합니다. 분리를 선택합니다.

연동 사용자 역할에 AWS Cloud9 관리형 정책 연결

AWS Cloud9 개발 환경을 사용하는 경우 `AWSCloud9User` 관리형 정책을 사용자 역할에 연결하여 연동 사용자에게 액세스 권한을 부여하세요. IAM사용자와 달리 연동 사용자에 대한 관리형 정책은 수동으로 연결 및 분리해야 합니다. 이 단계를 수행하려면 루트 사용자, 계정의 관리자 사용자 또는 연결된 `AdministratorAccess` 관리형 정책 또는 이와 동등한 정책을 사용하는 IAM 사용자 또는 페더레이션 사용자로 콘솔에 로그인해야 합니다.

사전 조건:

- 연합된 사용자가 위임하는 역할을 만들거나 기존 역할이 있어야 합니다.
- 자신이 부여할 권한 수준을 알고 있어야 합니다.
  - `AWSCloud9User` 관리형 정책이 있으면 사용자는 다음과 같은 작업을 할 수 있습니다.
    - 자체 AWS Cloud9 개발 환경을 만드세요.
    - 환경 관련 정보를 확인합니다.
    - 환경 설정을 변경합니다.
  - `AWSCloud9Administrator` 관리형 정책이 있으면 사용자는 자신이나 타인을 위해 다음과 같은 작업을 할 수 있습니다.
    - 환경을 생성합니다.
    - 환경 관련 정보를 확인합니다.
    - 환경을 삭제합니다.
    - 환경 설정을 변경합니다.

1. IAM콘솔을 엽니다. 탐색 창에서 Policies를 선택합니다.
2. 검색 필드에 정책 이름을 입력합니다. 관리형 정책이 표시되며, 정책 유형은 AWS 관리형입니다. 정책을 확장하면 정책 설명의 권한을 확인할 수 있습니다.
3. 다음 관리형 정책 중 하나를 선택하십시오. 정책 옆에 있는 동그라미를 선택하고, 정책 작업에서 연결을 선택합니다.

4. 요약 페이지에서 연결된 개체 탭을 선택합니다. 연결을 선택합니다.
5. 정책 연결 페이지에서 검색 필드에 연합된 사용자의 역할을 입력해 필터링합니다. 역할 이름 옆에 있는 확인란을 선택하고 정책 연결을 선택합니다. 연결된 개체 탭에 새로운 연결이 표시됩니다.

연동 사용자 역할에서 AWS Cloud9 관리형 정책을 분리합니다.

AWS Cloud9 개발 환경을 사용하는 경우 액세스 권한을 부여하는 정책을 분리하여 연동 사용자의 액세스 권한을 제거할 수 있습니다. 이 단계를 수행하려면 루트 사용자, 계정의 관리자 사용자 또는 연결된 `AdministratorAccess` 관리형 정책 또는 이와 동등한 정책을 사용하는 IAM 사용자 또는 연동 사용자로 콘솔에 로그인해야 합니다.

1. IAM콘솔을 엽니다. 탐색 창에서 Policies를 선택합니다.
2. 검색 필드에 프로젝트 이름을 입력합니다.
3. 정책 옆에 있는 동그라미를 선택하고, 정책 작업에서 연결을 선택합니다.
4. 요약 페이지에서 연결된 개체 탭을 선택합니다.
5. 검색 필드에 연합된 사용자의 역할을 입력해 필터링합니다. 분리를 선택합니다.

다음과 같은 임시 자격 증명 사용 AWS CodeStar

임시 자격 증명을 사용하여 페더레이션으로 로그인하거나, 역할을 수임하거나, 계정 간 IAM 역할을 수임할 수 있습니다. AssumeRole또는 GetFederationToken와 같은 AWS STS API 작업을 호출하여 임시 보안 자격 증명을 얻을 수 있습니다.

AWS CodeStar 임시 자격 증명 사용을 지원하지만 페더레이션 액세스에는 AWS CodeStar 팀원 기능이 작동하지 않습니다. AWS CodeStar 팀원 기능은 IAM 사용자를 팀원으로 추가하는 것만 지원합니다.

서비스 연결 역할

서비스 연결 역할을 사용하면 AWS 서비스가 다른 서비스의 리소스에 액세스하여 사용자를 대신하여 작업을 완료할 수 있습니다. 서비스 연결 역할은 IAM 계정에 표시되며 서비스에서 소유합니다. 관리자는 서비스 연결 역할의 권한을 볼 수는 있지만 편집할 수는 없습니다.

AWS CodeStar 서비스 연결 역할을 지원하지 않습니다.

서비스 역할

이 기능을 사용하면 서비스가 사용자를 대신하여 서비스 역할을 수임할 수 있습니다. 이 역할을 사용하면 서비스가 다른 서비스의 리소스에 액세스해 사용자를 대신해 작업을 완료할 수 있습니다. 서비스 역할은 계정에 표시되며 해당 IAM 계정에서 소유합니다. 즉, 관리자가 이 역할에 대한 권한을 변경할 수 있습니다. 그러나 권한을 변경하면 서비스의 기능이 손상될 수 있습니다.

AWS CodeStar 서비스 역할을 지원합니다. AWS CodeStar 프로젝트의 리소스를 만들고 관리할 때 서비스 역할을 사용합니다. aws-codestar-service-role 자세한 내용은 IAM사용 설명서의 역할 용어 및 개념을 참조하십시오.

> **Important**
>
> 이 서비스 역할을 만들려면 관리 사용자나 루트 계정으로 로그인해야 합니다. 자세한 내용은 사용 설명서의 최초 액세스 전용: 루트 사용자 자격 증명 및 첫 번째 관리자 및 그룹 생성을 참조하십시오. IAM

이 역할은 에서 AWS CodeStar프로젝트를 처음 만들 때 자동으로 생성됩니다. 서비스 역할은 사용자를 대신해 다음을 수행합니다.

- 프로젝트를 생성할 때 선택한 리소스를 생성합니다.
- AWS CodeStar 프로젝트 대시보드에 해당 리소스에 대한 정보를 표시합니다.

또한 프로젝트의 리소스를 관리하는 경우 사용자를 대신해 이를 수행합니다. 이 정책 설명의 예제는 AWSCodeStarServiceRole 정책 단원을 참조하십시오.

또한 프로젝트 유형에 따라 여러 프로젝트별 서비스 역할을 AWS CodeStar 생성합니다. AWS CloudFormation 각 프로젝트 유형에 대해 툴체인 역할이 생성됩니다.

- AWS CloudFormation 역할을 통해 AWS CodeStar 프로젝트에 사용할 AWS CloudFormation 스택을 만들고 수정할 수 있습니다. AWS CodeStar
- 툴체인 역할을 통해 다른 AWS 서비스에 AWS CodeStar 액세스하여 프로젝트의 리소스를 만들고 수정할 수 있습니다. AWS CodeStar

AWS CodeStar 프로젝트 수준 정책 및 권한

프로젝트를 만들 때 프로젝트 리소스를 관리하는 데 필요한 IAM 역할과 정책을 AWS CodeStar 만듭니다. 정책은 다음 세 가지 범주로 나뉩니다.

- IAM프로젝트 팀 구성원을 위한 정책.
- IAM작업자 역할 정책.
- IAM런타임 실행 역할에 대한 정책.

IAM팀 구성원을 위한 정책

프로젝트를 생성할 때 는 프로젝트에 대한 소유자, 기여자 및 뷰어 액세스를 위한 세 가지 고객 관리 정책을 AWS CodeStar 생성합니다. 모든 AWS CodeStar 프로젝트에는 이러한 세 가지 액세스 수준에 대한 IAM 정책이 포함되어 있습니다. 이러한 액세스 수준은 프로젝트별로 다르며 표준 이름을 가진 IAM 관리형 정책에 의해 정의됩니다.*project-id* AWS CodeStar 프로젝트의 ID입니다 (예: *my-first-projec*):

- `CodeStar_`*project-id*`_Owner`
- `CodeStar_`*project-id*`_Contributor`
- `CodeStar_`*project-id*`_Viewer`

> **Important**
>
> 이 정책은 에 따라 변경될 수 AWS CodeStar있습니다. 수동으로 편집하면 안 됩니다. 권한을 추가하거나 변경하려면 IAM 사용자에게 추가 정책을 추가하세요.

프로젝트에 팀 구성원 (IAM사용자) 을 추가하고 액세스 수준을 선택하면 해당 정책이 사용자에게 연결되어 IAM 사용자에게 프로젝트 리소스에 대한 작업을 수행할 수 있는 적절한 권한 집합이 부여됩니다. 대부분의 경우 에서 IAM 정책이나 권한을 직접 연결하거나 관리할 필요가 없습니다. AWS CodeStar 액세스 수준 정책을 IAM 사용자에게 수동으로 연결하는 것은 권장되지 않습니다. 꼭 필요한 경우 AWS CodeStar 액세스 수준 정책을 보완하기 위해 자체 관리형 또는 인라인 정책을 만들어 사용자에게 고유한 수준의 권한을 적용할 수 있습니다. IAM

정책의 범위는 프로젝트 리소스 및 특정 작업으로 엄격하게 지정됩니다. 새 리소스가 인프라 스택에 추가되면 지원되는 리소스 유형 중 하나인 경우 새 리소스에 액세스할 수 있는 권한을 포함하도록 팀원 정책을 AWS CodeStar 업데이트하려고 시도합니다.

> **Note**
>
> 프로젝트의 액세스 수준 정책은 해당 AWS CodeStar 프로젝트에만 적용됩니다. 이렇게 하면 사용자가 역할에 따라 결정된 수준에서만 권한이 있는 AWS CodeStar 프로젝트를 보고 상호 작용할 수 있습니다. AWS CodeStar 프로젝트를 만드는 사용자에게만 프로젝트와 상관없이 모든 AWS CodeStar 리소스에 대한 액세스를 허용하는 정책을 적용해야 합니다.

모든 AWS CodeStar 액세스 수준 정책은 액세스 수준이 연결된 프로젝트와 관련된 AWS 리소스에 따라 달라집니다. 다른 AWS 서비스와 달리, 이러한 정책은 프로젝트가 만들어질 때와 프로젝트 리소스가 변경됨에 따라 업데이트될 때 사용자 지정됩니다. 따라서 정식 소유자, 기고자 또는 최종 사용자 관리형 정책은 없습니다.

AWS CodeStar 소유자 역할 정책

`CodeStar_project-id_Owner`고객 관리형 정책을 사용하면 사용자가 AWS CodeStar 프로젝트의 모든 작업을 제한 없이 수행할 수 있습니다. 이는 사용자가 팀원을 추가하거나 제거할 수 있도록 허용하는 유일한 정책입니다. 정책의 내용은 프로젝트와 연결된 리소스에 따라 다릅니다. 예제는 AWS CodeStar 소유자 역할 정책 단원을 참조하세요.

이 정책을 사용하는 IAM 사용자는 프로젝트의 모든 AWS CodeStar 작업을 수행할 수 있지만 `AWSCodeStarFullAccess` 정책을 사용하는 IAM 사용자와 달리 사용자는 프로젝트를 만들 수 없습니다. `codestar:*`권한 범위는 특정 리소스 (해당 AWS CodeStar 프로젝트 ID와 연결된 프로젝트) 로 제한됩니다.

AWS CodeStar 기여자 역할 정책

`CodeStar_project-id_Contributor` 고객 관리형 정책은 사용자가 프로젝트에 참가하고 프로젝트 대시보드를 변경하는 것은 허용하지만, 팀원을 추가하거나 제거하는 것은 허용하지 않습니다. 정책의 내용은 프로젝트와 연결된 리소스에 따라 다릅니다. 예제는 AWS CodeStar 기고자 역할 정책 단원을 참조하세요.

AWS CodeStar 시청자 역할 정책

`CodeStar_project-id_Viewer` 고객 관리형 정책은 사용자가 AWS CodeStar에서 프로젝트를 보는 것은 허용하지만, 리소스를 변경하거나 팀원을 추가 또는 제거하는 것은 허용하지 않습니다. 정책의 내용은 프로젝트와 연결된 리소스에 따라 다릅니다. 예제는 AWS CodeStar 뷰어 역할 정책 단원을 참조하세요.

IAM작업자 역할 정책

2018년 PDT 12월 6일 이후에 AWS CodeStar 프로젝트를 생성하면 두 개의 작업자 역할이 AWS CodeStar `CodeStar-`*`project-id`*`-ToolChain` 생성되고`CodeStar-`*`project-id`*`-CloudFormation`, 작업자 역할은 서비스에 전달하기 위해 AWS CodeStar 생성하는 프로젝트별 IAM 역할입니다. 서비스가 프로젝트 컨텍스트에서 리소스를 만들고 작업을 실행할 수 있도록 권한을 부여합니다 AWS CodeStar . 툴체인 작업자 역할은 CodeBuild, CodeDeploy, 등의 툴체인 서비스와 설정된 신뢰 관계를 갖습니다. CodePipeline 프로젝트 팀원(소유자 및 기고자)에게는 작업자 역할을 신뢰할 수 있는 다운스트림 서비스로 전달할 수 있는 액세스 권한이 부여됩니다. 이 역할에 대한 인라인 정책 설명의 예제는 AWS CodeStar 툴체인 작업자 역할 정책 (2018년 PDT 12월 6일 이후) 단원을 참조하십시오.

CloudFormation 작업자 역할에는 에서 지원하는 AWS CloudFormation선택된 리소스에 대한 권한뿐만 아니라 애플리케이션 스택에서 IAM 사용자, 역할 및 정책을 생성할 수 있는 권한이 포함됩니다. 또한 신뢰 관계도 설정되어 있습니다 AWS CloudFormation. 권한 상승 및 파괴적인 조치의 위험을 줄이기 위해 AWS CloudFormation 역할 정책에는 인프라 스택에서 생성되는 모든 IAM 엔티티 (사용자 또는 역할) 에 대해 프로젝트별 권한 경계를 요구하는 조건이 포함되어 있습니다. 이 역할에 대한 인라인 정책 설명의 예제는 AWS CloudFormation 작업자 역할 정책 단원을 참조하십시오.

2018년 PDT AWS CodeStar 12월 6일 이전에 만든 AWS CodeStar 프로젝트의 경우,, CloudWatch Events와 같은 CodePipeline 툴체인 리소스에 대한 개별 작업자 역할을 생성하고 제한된 리소스 세트를 지원하는 작업자 역할도 생성합니다. CodeBuild AWS CloudFormation 이러한 각 역할은 해당 서비스와 신뢰 관계를 수립합니다. 프로젝트 팀원(소유자 및 기고자) 및 일부 다른 작업자 역할에는 신뢰할 수 있는 다운스트림 서비스로 역할을 전달할 수 있는 액세스 권한이 부여됩니다. 작업자 역할의 권한은 인라인 정책에 정의되어 있는데 이 역할이 프로젝트 리소스 세트에 대해 수행할 수 있는 기본 작업 세트로 범위가 지정됩니다. 이러한 권한은 정적입니다. 생성 시 프로젝트에 포함된 리소스에 대한 권한을 포함하지만, 프로젝트에 새로운 리소스가 추가되더라도 업데이트되지 않습니다. 이러한 정책 설명의 예제는 다음을 참조하십시오.

- AWS CloudFormation 근로자 역할 정책 (2018년 PDT 12월 6일 이전)
- AWS CodePipeline 근로자 역할 정책 (2018년 PDT 12월 6일 이전)
- AWS CodeBuild 근로자 역할 정책 (2018년 PDT 12월 6일 이전)
- 아마존 CloudWatch 이벤트 워커 역할 정책 (2018년 12월 6일 이전PDT)

IAM실행 역할 정책

2018년 12월 6일 이후에 만든 프로젝트의 PDT 경우 애플리케이션 스택의 샘플 프로젝트에 대한 일반 실행 역할을 AWS CodeStar 생성합니다. 이 역할의 범위는 권한 경계 정책을 사용하여 프로젝트 리소스로 좁혀집니다. 샘플 프로젝트를 확장하면서 추가 IAM 역할을 만들 수 있으며 권한 에스컬레이션을 방지하기 위해 AWS CloudFormation 역할 정책에 따라 권한 경계를 사용하여 이러한 역할의 범위를 축소해야 합니다. 자세한 내용은 프로젝트에 IAM 역할 추가 단원을 참조하십시오.

PDT2018년 12월 6일 이전에 생성된 Lambda 프로젝트의 경우, 프로젝트 스택의 리소스에서 작업을 수행할 권한이 있는 인라인 정책이 첨부된 Lambda 실행 역할을 AWS CodeStar 생성합니다. AWS SAM 새 리소스가 SAM 템플릿에 추가되면 지원되는 리소스 유형 중 하나인 경우 새 리소스에 대한 권한을 포함하도록 Lambda 실행 역할 정책을 AWS CodeStar 업데이트하려고 시도합니다.

IAM 권한 경계

2018년 PDT 12월 6일 이후 프로젝트를 생성하면 고객 관리형 정책을 AWS CodeStar 생성하고 이 정책을 프로젝트 내 IAM 역할에 IAM대한 권한 경계로 할당합니다. AWS CodeStar애플리케이션 스택에서 만든 모든 IAM 엔티티에 권한 경계가 있어야 합니다. 권한 경계는 역할이 가질 수 있는 최대 권한을 제어하지만, 역할에 권한을 제공하지는 않습니다. 권한 정책은 역할의 권한을 정의합니다. 즉 역할에 추가된 권한 수가 몇 개이든, 역할을 사용하는 사람은 권한 경계에 포함된 작업 이상을 수행할 수 없습니다. 권한 정책 및 권한 경계를 평가하는 방법에 대한 자세한 내용은 사용 IAM설명서의 정책 평가 로직을 참조하십시오.

AWS CodeStar 프로젝트별 권한 경계를 사용하여 프로젝트 외부 리소스로 권한이 에스컬레이션되는 것을 방지합니다. AWS CodeStar 권한 경계에는 프로젝트 리소스에 ARNs 대한 내용이 포함됩니다. 이 정책 설명의 예제는 AWS CodeStar 권한 경계 정책 단원을 참조하십시오.

애플리케이션 스택 (`template.yml`) 을 통해 지원되는 리소스를 프로젝트에 추가하거나 제거하면 AWS CodeStar 변환이 이 정책을 업데이트합니다.

기존 프로젝트에 IAM 권한 경계 추가

2018년 PDT 12월 6일 이전에 만든 AWS CodeStar 프로젝트가 있는 경우 프로젝트의 IAM 역할에 권한 경계를 수동으로 추가해야 합니다. 모범 사례로써 프로젝트 외부의 리소스로의 권한 상승을 방지하도록 프로젝트의 리소스만 포함하는 프로젝트별 경계를 사용하는 것이 좋습니다. 프로젝트가 진행됨에 따라 AWS CodeStar 업데이트되는 관리 권한 경계를 사용하려면 다음 단계를 따르세요.

1. AWS CloudFormation 콘솔에 로그인하고 프로젝트의 툴체인 스택용 템플릿을 찾으세요. 이 템플릿 이름은 `awscodestar-`*`project-id`*입니다.
2. 해당 템플릿을 선택하고 작업을 선택한 다음 Designer에서 템플릿 보기/편집을 선택합니다.

3. Resources 섹션을 찾은 후 섹션 상단에 다음 조각을 포함시킵니다.

```
 PermissionsBoundaryPolicy:
   Description: Creating an IAM managed policy for defining the permissions boundary
for an AWS CodeStar project
   Type: AWS::IAM::ManagedPolicy
   Properties:
     ManagedPolicyName: !Sub 'CodeStar_${ProjectId }_PermissionsBoundary'
     Description: 'IAM policy to define the permissions boundary for IAM entities
created in an AWS CodeStar project'
     PolicyDocument:
       Version: '2012-10-17'
       Statement:
       - Sid: '1'
         Effect: Allow
         Action: ['*']
         Resource:
           - !Sub 'arn:${AWS::Partition}:cloudformation:${AWS::Region}:
${AWS::AccountId}:stack/awscodestar-${ProjectId}-*'
```

AWS CloudFormation 콘솔에서 스택을 업데이트하려면 추가 IAM 권한이 필요할 수 있습니다.

4. (선택 사항) 애플리케이션별 IAM 역할을 생성하려면 이 단계를 완료하십시오. IAM콘솔에서 프로젝트 AWS CloudFormation 역할에 연결된 인라인 정책을 업데이트하여 다음 스니펫을 포함하세요. 정책을 업데이트하려면 추가 IAM 리소스가 필요할 수 있습니다.

```
      {
           "Action": [
               "iam:PassRole"
           ],
           "Resource": "arn:aws:iam::{AccountId}:role/CodeStar-{ProjectId}*",
           "Effect": "Allow"
       },
       {
           "Action": [
               "iam:CreateServiceLinkedRole",
               "iam:GetRole",
               "iam:DeleteRole",
               "iam:DeleteUser"
           ],
```

```
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "iam:AttachRolePolicy",
                "iam:AttachUserPolicy",
                "iam:CreateRole",
                "iam:CreateUser",
                "iam:DeleteRolePolicy",
                "iam:DeleteUserPolicy",
                "iam:DetachUserPolicy",
                "iam:DetachRolePolicy",
                "iam:PutUserPermissionsBoundary",
                "iam:PutRolePermissionsBoundary"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PermissionsBoundary": "arn:aws:iam::{AccountId}:policy/
CodeStar_{ProjectId}_PermissionsBoundary"
                }
            },
            "Effect": "Allow"
        }
```

5. 적절한 권한으로 권한 경계를 AWS CodeStar 업데이트하도록 프로젝트 파이프라인을 통해 변경 사항을 푸시하세요.

자세한 내용은 프로젝트에 IAM 역할 추가 단원을 참조하십시오.

AWS CodeStarID 기반 정책 예제

기본적으로 IAM 사용자 및 역할에는 리소스를 만들거나 수정할 AWS CodeStar 권한이 없습니다. 또한 AWS Management Console AWS CLI, 또는 를 사용하는 작업도 수행할 수 없습니다 AWS API. 관리자는 필요한 지정된 리소스에서 특정 API 작업을 수행할 수 있는 권한을 사용자 및 역할에 부여하는 IAM 정책을 만들어야 합니다. 그런 다음 관리자는 해당 권한이 필요한 IAM 사용자 또는 그룹에 해당 정책을 연결해야 합니다.

이 예제 JSON 정책 문서를 사용하여 IAM ID 기반 정책을 만드는 방법을 알아보려면 사용 IAM설명서의 JSON 탭에서 정책 생성을 참조하십시오.

주제

정책 모범 사례

ID 기반 정책은 누군가가 계정의 AWS CodeStar 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부를 결정합니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. ID 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따릅니다.

- AWS 관리형 정책으로 시작하여 최소 권한 권한으로 이동 — 사용자와 워크로드에 권한을 부여하려면 여러 일반적인 사용 사례에 권한을 부여하는 AWS 관리형 정책을 사용하세요. 해당 내용은 에서 사용할 수 있습니다. AWS 계정사용 사례에 맞는 AWS 고객 관리형 정책을 정의하여 권한을 더 줄이는 것이 좋습니다. 자세한 내용은 IAM사용 설명서의 AWS 관리형 정책 또는 작업 기능에 대한AWS 관리형 정책을 참조하십시오.
- 최소 권한 적용 — IAM 정책으로 권한을 설정하는 경우 작업 수행에 필요한 권한만 부여하십시오. 이렇게 하려면 최소 권한으로 알려진 특정 조건에서 특정 리소스에 대해 수행할 수 있는 작업을 정의합니다. 를 사용하여 권한을 IAM 적용하는 방법에 대한 자세한 내용은 사용 설명서의 정책 및 권한을 참조하십시오. IAM IAM
- IAM정책의 조건을 사용하여 액세스를 더욱 제한할 수 있습니다. - 정책에 조건을 추가하여 작업 및 리소스에 대한 액세스를 제한할 수 있습니다. 예를 들어, 를 사용하여 모든 요청을 전송하도록 지정하는 정책 조건을 작성할 수 SSL 있습니다. 예를 AWS 서비스들어 특정 작업을 통해 서비스 작업을 사용하는 경우 조건을 사용하여 서비스 작업에 대한 액세스 권한을 부여할 수도 AWS CloudFormation있습니다. 자세한 내용은 IAM사용 설명서의 IAMJSON정책 요소: 조건을 참조하십시오.
- IAMAccess Analyzer를 사용하여 IAM 정책을 검증하여 안전하고 기능적인 권한을 보장합니다. IAM Access Analyzer는 새 정책과 기존 정책을 검증하여 정책이 IAM 정책 언어 (JSON) 및 IAM 모범 사례를 준수하는지 확인합니다. IAMAccess Analyzer는 안전하고 기능적인 정책을 작성하는 데 도움이 되는 100개 이상의 정책 검사와 실행 가능한 권장 사항을 제공합니다. 자세한 내용은 사용 설명서의 IAMAccess Analyzer 정책 검증을 참조하십시오. IAM
- 다단계 인증 필요 (MFA) - 사용자 또는 루트 IAM 사용자가 필요한 시나리오가 있는 경우 보안을 강화하려면 이 기능을 MFA 켜십시오. AWS 계정 API작업 호출 MFA 시기를 요구하려면 정책에 MFA 조건을 추가하세요. 자세한 내용은 IAM사용 설명서의 MFA -보호된 API 액세스 구성을 참조하십시오.

의 모범 사례에 IAM 대한 자세한 내용은 IAM사용 설명서의 보안 모범 사례를 참조하십시오. IAM

AWSCodeStarServiceRole 정책

`aws-codestar-service-role`정책은 다른 서비스와 함께 작업을 수행할 수 AWS CodeStar 있는 서비스 역할에 연결됩니다. 에 처음 로그인할 AWS CodeStar때 서비스 역할을 생성합니다. 한 번만 생성하면 됩니다. 서비스 역할이 생성된 후 정책이 자동으로 연결됩니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Sid": "ProjectEventRules",
            "Effect": "Allow",
            "Action": [
                "events:PutTargets",
                "events:RemoveTargets",
                "events:PutRule",
                "events:DeleteRule",
                "events:DescribeRule"
            ],
            "Resource": [
                "arn:aws:events:*:*:rule/awscodestar-*"
            ]
        },
        {
            "Sid": "ProjectStack",
            "Effect": "Allow",
            "Action": [
                "cloudformation:*Stack*",
                "cloudformation:CreateChangeSet",
                "cloudformation:ExecuteChangeSet",
                "cloudformation:DeleteChangeSet",
                "cloudformation:GetTemplate"
            ],
            "Resource": [
                "arn:aws:cloudformation:*:*:stack/awscodestar-*",
                "arn:aws:cloudformation:*:*:stack/awseb-*",
                "arn:aws:cloudformation:*:*:stack/aws-cloud9-*",
                "arn:aws:cloudformation:*:aws:transform/CodeStar*"
            ]
        },
        {
            "Sid": "ProjectStackTemplate",
            "Effect": "Allow",
            "Action": [
                "cloudformation:GetTemplateSummary",
                "cloudformation:DescribeChangeSet"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ProjectQuickstarts",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
```

```
            ],
            "Resource": [
                "arn:aws:s3:::awscodestar-*/*"
            ]
        },
        {
            "Sid": "ProjectS3Buckets",
            "Effect": "Allow",
            "Action": [
                "s3:*"
            ],
            "Resource": [
                "arn:aws:s3:::aws-codestar-*",
                "arn:aws:s3:::elasticbeanstalk-*"
            ]
        },
        {
            "Sid": "ProjectServices",
            "Effect": "Allow",
            "Action": [
                "codestar:*",
                "codecommit:*",
                "codepipeline:*",
                "codedeploy:*",
                "codebuild:*",
                "autoscaling:*",
                "cloudwatch:Put*",
                "ec2:*",
                "elasticbeanstalk:*",
                "elasticloadbalancing:*",
                "iam:ListRoles",
                "logs:*",
                "sns:*",
                "cloud9:CreateEnvironmentEC2",
                "cloud9:DeleteEnvironment",
                "cloud9:DescribeEnvironment*",
                "cloud9:ListEnvironments"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ProjectWorkerRoles",
            "Effect": "Allow",
            "Action": [
```

```
                "iam:AttachRolePolicy",
                "iam:CreateRole",
                "iam:DeleteRole",
                "iam:DeleteRolePolicy",
                "iam:DetachRolePolicy",
                "iam:GetRole",
                "iam:PassRole",
                "iam:GetRolePolicy",
                "iam:PutRolePolicy",
                "iam:SetDefaultPolicyVersion",
                "iam:CreatePolicy",
                "iam:DeletePolicy",
                "iam:AddRoleToInstanceProfile",
                "iam:CreateInstanceProfile",
                "iam:DeleteInstanceProfile",
                "iam:RemoveRoleFromInstanceProfile"
            ],
            "Resource": [
                "arn:aws:iam::*:role/CodeStarWorker*",
                "arn:aws:iam::*:policy/CodeStarWorker*",
                "arn:aws:iam::*:instance-profile/awscodestar-*"
            ]
        },
        {
            "Sid": "ProjectTeamMembers",
            "Effect": "Allow",
            "Action": [
                "iam:AttachUserPolicy",
                "iam:DetachUserPolicy"
            ],
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                    "iam:PolicyArn": [
                        "arn:aws:iam::*:policy/CodeStar_*"
                    ]
                }
            }
        },
        {
            "Sid": "ProjectRoles",
            "Effect": "Allow",
            "Action": [
                "iam:CreatePolicy",
```

```
                "iam:DeletePolicy",
                "iam:CreatePolicyVersion",
                "iam:DeletePolicyVersion",
                "iam:ListEntitiesForPolicy",
                "iam:ListPolicyVersions",
                "iam:GetPolicy",
                "iam:GetPolicyVersion"
            ],
            "Resource": [
                "arn:aws:iam::*:policy/CodeStar_*"
            ]
        },
        {
            "Sid": "InspectServiceRole",
            "Effect": "Allow",
            "Action": [
                "iam:ListAttachedRolePolicies"
            ],
            "Resource": [
                "arn:aws:iam::*:role/aws-codestar-service-role",
                "arn:aws:iam::*:role/service-role/aws-codestar-service-role"
            ]
        },
        {
            "Sid": "IAMLinkRole",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": "cloud9.amazonaws.com"
                }
            }
        },
        {
            "Sid": "DescribeConfigRuleForARN",
            "Effect": "Allow",
            "Action": [
                "config:DescribeConfigRules"
            ],
            "Resource": [
                "*"
```

```
            ]
        },
        {
            "Sid": "ProjectCodeStarConnections",
            "Effect": "Allow",
            "Action": [
                "codestar-connections:UseConnection",
                "codestar-connections:GetConnection"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ProjectCodeStarConnectionsPassConnections",
            "Effect": "Allow",
            "Action": "codestar-connections:PassConnection",
            "Resource": "*",
            "Condition": {
                "StringEqualsIfExists": {
                    "codestar-connections:PassedToService":
 "codepipeline.amazonaws.com"
                }
            }
        }
    ]
}
```

AWSCodeStarFullAccess 정책

AWS CodeStar설정지침에서 IAM 사용자에게 이름이 지정된 AWSCodeStarFullAccess 정책을 첨부했습니다. 이 정책 설명을 통해 사용자는 AWS CodeStar AWS 계정과 관련된 모든 가용 AWS CodeStar 리소스를 사용하여 가능한 모든 작업을 수행할 수 있습니다. 여기에는 프로젝트를 만들고 삭제하는 작업이 포함됩니다. 다음 예제는 대표적인 AWSCodeStarFullAccess 정책의 코드 조각입니다. 실제 정책은 새 AWS CodeStar 프로젝트를 시작할 때 선택하는 템플릿에 따라 달라집니다.

AWS CloudFormation 대상 스택 cloudformation::DescribeStacks 없이 호출하는 경우 cloudformation::ListStacks 권한이 필요합니다.

권한 세부 정보

이 정책에는 다음을 수행할 수 있는 권한이 포함되어 있습니다.

- ec2EC2—인스턴스에 대한 정보를 검색하여 AWS CodeStar 프로젝트를 생성합니다.

- `cloud9`—환경에 대한 AWS Command Line Interface 정보를 검색합니다.
- `cloudformation` AWS CodeStar —프로젝트 스택에 대한 정보를 검색합니다.
- `codestar`—프로젝트 내에서 작업을 수행합니다. AWS CodeStar

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeStarEC2",
      "Effect": "Allow",
      "Action": [
        "codestar:*",
        "ec2:DescribeKeyPairs",
        "ec2:DescribeVpcs",
        "ec2:DescribeSubnets",
        "cloud9:DescribeEnvironment*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CodeStarCF",
      "Effect": "Allow",
      "Action": [
        "cloudformation:DescribeStack*",
        "cloudformation:ListStacks*",
        "cloudformation:GetTemplateSummary"
      ],
      "Resource": [
        "arn:aws:cloudformation:*:*:stack/awscodestar-*"
      ]
    }
  ]
}
```

모든 사용자에게 이렇게 많은 액세스 권한을 부여하고 싶지 않을 수 있습니다. 대신 에서 관리하는 프로젝트 역할을 사용하여 프로젝트 수준 권한을 추가할 수 있습니다. AWS CodeStar역할은 AWS CodeStar 프로젝트에 대한 특정 수준의 액세스 권한을 부여하며 이름은 다음과 같습니다.

- 소유자
- 기고자

- 뷰어

AWS CodeStar 소유자 역할 정책

AWS CodeStar 소유자 역할 정책은 사용자가 AWS CodeStar 프로젝트의 모든 작업을 제한 없이 수행할 수 있도록 허용합니다. AWS CodeStar 소유자 액세스 수준을 가진 프로젝트 팀 구성원에게 CodeStar_*project-id*_Owner 정책을 적용합니다.

```
...
{
  "Effect": "Allow",
  "Action": [
    ...
    "codestar:*",
    ...
  ],
  "Resource": [
    "arn:aws:codestar:us-east-2:111111111111:project/project-id",
    "arn:aws:iam::account-id:policy/CodeStar_project-id_Owner"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "codestar:DescribeUserProfile",
    "codestar:ListProjects",
    "codestar:ListUserProfiles",
    "codestar:VerifyServiceRole",
    ...
  ],
  "Resource": [
    "*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "codestar:*UserProfile",
    ...
  ],
  "Resource": [
    "arn:aws:iam::account-id:user/user-name"
```

```
  ]
}
...
```

AWS CodeStar 기고자 역할 정책

AWS CodeStar 기여자 역할 정책을 통해 사용자는 프로젝트에 기여하고 프로젝트 대시보드를 변경할 수 있습니다. AWS CodeStar 기여자 액세스 수준을 가진 프로젝트 팀 구성원에게 CodeStar_*project-id*_Contributor 정책을 적용합니다. 기고자 액세스 권한이 있는 사용자는 프로젝트에 참여하고 프로젝트 대시보드를 변경할 수 있지만, 팀원을 추가하거나 제거할 수는 없습니다.

```
...
{
  "Effect": "Allow",
  "Action": [
    ...
    "codestar:Describe*",
    "codestar:Get*",
    "codestar:List*",
    "codestar:PutExtendedAccess",
    ...
  ],
  "Resource": [
    "arn:aws:codestar:us-east-2:111111111111:project/project-id",
    "arn:aws:iam::account-id:policy/CodeStar_project-id_Contributor"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "codestar:DescribeUserProfile",
    "codestar:ListProjects",
    "codestar:ListUserProfiles",
    "codestar:VerifyServiceRole",
    ...
  ],
  "Resource": [
    "*"
  ]
},
{
  "Effect": "Allow",
```

```
  "Action": [
    "codestar:*UserProfile",
    ...
  ],
  "Resource": [
    "arn:aws:iam::account-id:user/user-name"
  ]
}
...
```

AWS CodeStar 뷰어 역할 정책

AWS CodeStar 뷰어 역할 정책은 사용자가 에서 프로젝트를 볼 수 있도록 허용합니다 AWS CodeStar. AWS CodeStar 뷰어 액세스 수준을 가진 프로젝트 팀 구성원에게 CodeStar_*project-id*_Viewer 정책을 적용합니다. 뷰어 액세스 권한이 있는 사용자는 에서 AWS CodeStar 프로젝트를 볼 수 있지만 리소스를 변경하거나 팀 구성원을 추가 또는 제거할 수는 없습니다.

```
...
{
  "Effect": "Allow",
  "Action": [
    ...
    "codestar:Describe*",
    "codestar:Get*",
    "codestar:List*",
    ...
  ],
  "Resource": [
    "arn:aws:codestar:us-east-2:111111111111:project/project-id",
    "arn:aws:iam::account-id:policy/CodeStar_project-id_Viewer"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "codestar:DescribeUserProfile",
    "codestar:ListProjects",
    "codestar:ListUserProfiles",
    "codestar:VerifyServiceRole",
    ...
  ],
  "Resource": [
    "*"
```

```
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "codestar:*UserProfile",
    ...
  ],
  "Resource": [
    "arn:aws:iam::account-id:user/user-name"
  ]
}
...
```

AWS CodeStar 툴체인 작업자 역할 정책 (2018년 PDT 12월 6일 이후)

2018년 PDT 12월 6일 이후에 만든 AWS CodeStar 프로젝트의 경우 작업자 역할에 대한 인라인 정책을 만들어 다른 AWS 서비스에 프로젝트 리소스를 생성합니다. AWS CodeStar 정책의 내용은 생성 중인 프로젝트의 유형에 따라 다릅니다. 다음은 정책의 예입니다. 자세한 내용은 IAM작업자 역할 정책 단원을 참조하십시오.

```
{
  "Statement": [
    {
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetBucketVersioning",
        "s3:PutObject*",
        "codecommit:CancelUploadArchive",
        "codecommit:GetBranch",
        "codecommit:GetCommit",
        "codecommit:GetUploadArchiveStatus",
        "codecommit:GitPull",
        "codecommit:UploadArchive",
        "codebuild:StartBuild",
        "codebuild:BatchGetBuilds",
        "codebuild:StopBuild",
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "cloudformation:DescribeStacks",
        "cloudformation:DescribeChangeSet",
```

```
        "cloudformation:CreateChangeSet",
        "cloudformation:DeleteChangeSet",
        "cloudformation:ExecuteChangeSet",
        "codepipeline:StartPipelineExecution",
        "lambda:ListFunctions",
        "lambda:InvokeFunction",
        "sns:Publish"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Encrypt",
        "kms:Decrypt"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

AWS CloudFormation 작업자 역할 정책

2018년 12월 6일 이후에 만든 AWS CodeStar 프로젝트의 PDT 경우 프로젝트에 AWS CloudFormation 필요한 리소스를 AWS CodeStar 생성하는 작업자 역할에 대한 인라인 정책을 만듭니다. AWS CodeStar 정책의 내용은 프로젝트에 필요한 리소스의 유형에 따라 다릅니다. 다음은 정책의 예입니다. 자세한 내용은 IAM작업자 역할 정책 단원을 참조하십시오.

```
{
{
    "Statement": [
        {
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::aws-codestar-region-id-account-id-project-id",
                "arn:aws:s3:::aws-codestar-region-id-account-id-project-id/*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "apigateway:DELETE",
                "apigateway:GET",
                "apigateway:PATCH",
                "apigateway:POST",
                "apigateway:PUT",
                "codedeploy:CreateApplication",
                "codedeploy:CreateDeployment",
                "codedeploy:CreateDeploymentConfig",
                "codedeploy:CreateDeploymentGroup",
                "codedeploy:DeleteApplication",
                "codedeploy:DeleteDeployment",
                "codedeploy:DeleteDeploymentConfig",
                "codedeploy:DeleteDeploymentGroup",
                "codedeploy:GetDeployment",
                "codedeploy:GetDeploymentConfig",
                "codedeploy:GetDeploymentGroup",
                "codedeploy:RegisterApplicationRevision",
                "codestar:SyncResources",
                "config:DeleteConfigRule",
                "config:DescribeConfigRules",
                "config:ListTagsForResource",
                "config:PutConfigRule",
                "config:TagResource",
                "config:UntagResource",
                "dynamodb:CreateTable",
                "dynamodb:DeleteTable",
```

```
                "dynamodb:DescribeContinuousBackups",
                "dynamodb:DescribeTable",
                "dynamodb:DescribeTimeToLive",
                "dynamodb:ListTagsOfResource",
                "dynamodb:TagResource",
                "dynamodb:UntagResource",
                "dynamodb:UpdateContinuousBackups",
                "dynamodb:UpdateTable",
                "dynamodb:UpdateTimeToLive",
                "ec2:AssociateIamInstanceProfile",
                "ec2:AttachVolume",
                "ec2:CreateSecurityGroup",
                "ec2:createTags",
                "ec2:DescribeIamInstanceProfileAssociations",
                "ec2:DescribeInstances",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DetachVolume",
                "ec2:DisassociateIamInstanceProfile",
                "ec2:ModifyInstanceAttribute",
                "ec2:ModifyInstanceCreditSpecification",
                "ec2:ModifyInstancePlacement",
                "ec2:MonitorInstances",
                "ec2:ReplaceIamInstanceProfileAssociation",
                "ec2:RunInstances",
                "ec2:StartInstances",
                "ec2:StopInstances",
                "ec2:TerminateInstances",
                "events:DeleteRule",
                "events:DescribeRule",
                "events:ListTagsForResource",
                "events:PutRule",
                "events:PutTargets",
                "events:RemoveTargets",
                "events:TagResource",
                "events:UntagResource",
                "kinesis:AddTagsToStream",
                "kinesis:CreateStream",
                "kinesis:DecreaseStreamRetentionPeriod",
                "kinesis:DeleteStream",
                "kinesis:DescribeStream",
                "kinesis:IncreaseStreamRetentionPeriod",
                "kinesis:RemoveTagsFromStream",
                "kinesis:StartStreamEncryption",
```

```
                "kinesis:StopStreamEncryption",
                "kinesis:UpdateShardCount",
                "lambda:CreateAlias",
                "lambda:CreateFunction",
                "lambda:DeleteAlias",
                "lambda:DeleteFunction",
                "lambda:DeleteFunctionConcurrency",
                "lambda:GetFunction",
                "lambda:GetFunctionConfiguration",
                "lambda:ListTags",
                "lambda:ListVersionsByFunction",
                "lambda:PublishVersion",
                "lambda:PutFunctionConcurrency",
                "lambda:TagResource",
                "lambda:UntagResource",
                "lambda:UpdateAlias",
                "lambda:UpdateFunctionCode",
                "lambda:UpdateFunctionConfiguration",
                "s3:CreateBucket",
                "s3:DeleteBucket",
                "s3:DeleteBucketWebsite",
                "s3:PutAccelerateConfiguration",
                "s3:PutAnalyticsConfiguration",
                "s3:PutBucketAcl",
                "s3:PutBucketCORS",
                "s3:PutBucketLogging",
                "s3:PutBucketNotification",
                "s3:PutBucketPublicAccessBlock",
                "s3:PutBucketVersioning",
                "s3:PutBucketWebsite",
                "s3:PutEncryptionConfiguration",
                "s3:PutInventoryConfiguration",
                "s3:PutLifecycleConfiguration",
                "s3:PutMetricsConfiguration",
                "s3:PutReplicationConfiguration",
                "sns:CreateTopic",
                "sns:DeleteTopic",
                "sns:GetTopicAttributes",
                "sns:ListSubscriptionsByTopic",
                "sns:ListTopics",
                "sns:SetSubscriptionAttributes",
                "sns:Subscribe",
                "sns:Unsubscribe",
                "sqs:CreateQueue",
```

```
                "sqs:DeleteQueue",
                "sqs:GetQueueAttributes",
                "sqs:GetQueueUrl",
                "sqs:ListQueueTags",
                "sqs:TagQueue",
                "sqs:UntagQueue"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "lambda:AddPermission",
                "lambda:RemovePermission"
            ],
            "Resource": [
                "arn:aws:lambda:region-id:account-id:function:awscodestar-*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::account-id:role/CodeStar-project-id*"
            ],
            "Effect": "Allow"
        },
        {
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "codedeploy.amazonaws.com"
                }
            },
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::account-id:role/CodeStarWorker-project-id-CodeDeploy"
            ],
            "Effect": "Allow"
        },
        {
```

```
            "Action": [
                "cloudformation:CreateChangeSet"
            ],
            "Resource": [
                "arn:aws:cloudformation:region-id:aws:transform/Serverless-2016-10-31",
                "arn:aws:cloudformation:region-id:aws:transform/CodeStar"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "iam:CreateServiceLinkedRole",
                "iam:GetRole",
                "iam:DeleteRole",
                "iam:DeleteUser"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Condition": {
                "StringEquals": {
                    "iam:PermissionsBoundary": "arn:aws:iam::account-id:policy/
CodeStar_project-id_PermissionsBoundary"
                }
            },
            "Action": [
                "iam:AttachRolePolicy",
                "iam:AttachUserPolicy",
                "iam:CreateRole",
                "iam:CreateUser",
                "iam:DeleteRolePolicy",
                "iam:DeleteUserPolicy",
                "iam:DetachUserPolicy",
                "iam:DetachRolePolicy",
                "iam:PutUserPermissionsBoundary",
                "iam:PutRolePermissionsBoundary"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "kms:CreateKey",
```

```
                "kms:CreateAlias",
                "kms:DeleteAlias",
                "kms:DisableKey",
                "kms:EnableKey",
                "kms:UpdateAlias",
                "kms:TagResource",
                "kms:UntagResource"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Condition": {
                "StringEquals": {
                    "ssm:ResourceTag/awscodestar:projectArn":
 "arn:aws:codestar:project-id:account-id:project/project-id"
                }
            },
            "Action": [
                "ssm:GetParameter*"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

AWS CloudFormation 근로자 역할 정책 (2018년 PDT 12월 6일 이전)

2018년 12월 6일 이전에 만든 AWS CodeStar 프로젝트라면 AWS CloudFormation 작업자 PDT 역할에 대한 인라인 정책을 AWS CodeStar 만드세요. 다음은 정책 설명의 예제입니다.

```
{
    "Statement": [
        {
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::aws-codestar-us-east-1-account-id-project-id-pipe",
```

```
                "arn:aws:s3:::aws-codestar-us-east-1-account-id-project-id-pipe/*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "codestar:SyncResources",
                "lambda:CreateFunction",
                "lambda:DeleteFunction",
                "lambda:AddPermission",
                "lambda:UpdateFunction",
                "lambda:UpdateFunctionCode",
                "lambda:GetFunction",
                "lambda:GetFunctionConfiguration",
                "lambda:UpdateFunctionConfiguration",
                "lambda:RemovePermission",
                "lambda:listTags",
                "lambda:TagResource",
                "lambda:UntagResource",
                "apigateway:*",
                "dynamodb:CreateTable",
                "dynamodb:DeleteTable",
                "dynamodb:DescribeTable",
                "kinesis:CreateStream",
                "kinesis:DeleteStream",
                "kinesis:DescribeStream",
                "sns:CreateTopic",
                "sns:DeleteTopic",
                "sns:ListTopics",
                "sns:GetTopicAttributes",
                "sns:SetTopicAttributes",
                "s3:CreateBucket",
                "s3:DeleteBucket",
                "config:DescribeConfigRules",
                "config:PutConfigRule",
                "config:DeleteConfigRule",
                "ec2:*",
                "autoscaling:*",
                "elasticloadbalancing:*",
                "elasticbeanstalk:*"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
```

```
        {
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::account-id:role/CodeStarWorker-project-id-Lambda"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "cloudformation:CreateChangeSet"
            ],
            "Resource": [
                "arn:aws:cloudformation:us-east-1:aws:transform/Serverless-2016-10-31",
                "arn:aws:cloudformation:us-east-1:aws:transform/CodeStar"
            ],
            "Effect": "Allow"
        }
    ]
}
```

AWS CodePipeline 근로자 역할 정책 (2018년 PDT 12월 6일 이전)

2018년 12월 6일 이전에 만든 AWS CodeStar 프로젝트라면 CodePipeline 작업자 PDT 역할에 대한 인라인 정책을 AWS CodeStar 만드세요. 다음은 정책 설명의 예제입니다.

```
{
    "Statement": [
        {
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion",
                "s3:GetBucketVersioning",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::aws-codestar-us-east-1-account-id-project-id-pipe",
                "arn:aws:s3:::aws-codestar-us-east-1-account-id-project-id-pipe/*"
            ],
            "Effect": "Allow"
        },
        {
```

```
            "Action": [
                "codecommit:CancelUploadArchive",
                "codecommit:GetBranch",
                "codecommit:GetCommit",
                "codecommit:GetUploadArchiveStatus",
                "codecommit:UploadArchive"
            ],
            "Resource": [
                "arn:aws:codecommit:us-east-1:account-id:project-id"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "codebuild:StartBuild",
                "codebuild:BatchGetBuilds",
                "codebuild:StopBuild"
            ],
            "Resource": [
                "arn:aws:codebuild:us-east-1:account-id:project/project-id"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "cloudformation:DescribeStacks",
                "cloudformation:DescribeChangeSet",
                "cloudformation:CreateChangeSet",
                "cloudformation:DeleteChangeSet",
                "cloudformation:ExecuteChangeSet"
            ],
            "Resource": [
                "arn:aws:cloudformation:us-east-1:account-id:stack/awscodestar-project-
id-lambda/*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::account-id:role/CodeStarWorker-project-id-CloudFormation"
            ],
```

```
            "Effect": "Allow"
        }
    ]
}
```

AWS CodeBuild 근로자 역할 정책 (2018년 PDT 12월 6일 이전)

2018년 12월 6일 이전에 만든 AWS CodeStar 프로젝트라면 CodeBuild 작업자 PDT 역할에 대한 인라인 정책을 AWS CodeStar 만드세요. 다음은 정책 설명의 예제입니다.

```
{
    "Statement": [
        {
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::aws-codestar-us-east-1-account-id-project-id-pipe",
                "arn:aws:s3:::aws-codestar-us-east-1-account-id-project-id-pipe/*",
                "arn:aws:s3:::aws-codestar-us-east-1-account-id-project-id-app",
                "arn:aws:s3:::aws-codestar-us-east-1-account-id-project-id-app/*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "codecommit:GitPull"
            ],
            "Resource": [
                "arn:aws:codecommit:us-east-1:account-id:project-id"
            ],
            "Effect": "Allow"
```

```
        },
        {
            "Action": [
                "kms:GenerateDataKey*",
                "kms:Encrypt",
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:account-id:alias/aws/s3"
            ],
            "Effect": "Allow"
        }
    ]
}
```

아마존 CloudWatch 이벤트 워커 역할 정책 (2018년 12월 6일 이전PDT)

2018년 PDT 12월 6일 이전에 만든 AWS CodeStar 프로젝트라면 CloudWatch 이벤트 작업자 AWS CodeStar 역할을 위한 인라인 정책을 만드세요. 다음은 정책 설명의 예제입니다.

```
{
    "Statement": [
        {
            "Action": [
                "codepipeline:StartPipelineExecution"
            ],
            "Resource": [
                "arn:aws:codepipeline:us-east-1:account-id:project-id-Pipeline"
            ],
            "Effect": "Allow"
        }
    ]
}
```

AWS CodeStar 권한 경계 정책

2018년 PDT 12월 6일 이후에 AWS CodeStar 프로젝트를 만드는 경우 프로젝트에 대한 권한 경계 정책을 AWS CodeStar 생성합니다. 이 정책은 프로젝트 외부의 리소스로의 권한 상승을 방지합니다. 이 정책은 프로젝트가 진행됨에 따라 업데이트되는 동적 정책입니다. 정책의 내용은 생성 중인 프로젝트의 유형에 따라 다릅니다. 다음은 정책의 예입니다. 자세한 내용은 IAM 권한 경계 단원을 참조하십시오.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "1",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::*/AWSLogs/*/Config/*"
      ]
    },
    {
      "Sid": "2",
      "Effect": "Allow",
      "Action": [
        "*"
      ],
      "Resource": [
        "arn:aws:codestar:us-east-1:account-id:project/project-id",
        "arn:aws:cloudformation:us-east-1:account-id:stack/awscodestar-project-id-
lambda/eefbbf20-c1d9-11e8-8a3a-500c28b4e461",
        "arn:aws:cloudformation:us-east-1:account-id:stack/awscodestar-project-
id/4b80b3f0-c1d9-11e8-8517-500c28b236fd",
        "arn:aws:codebuild:us-east-1:account-id:project/project-id",
        "arn:aws:codecommit:us-east-1:account-id:project-id",
        "arn:aws:codepipeline:us-east-1:account-id:project-id-Pipeline",
        "arn:aws:execute-api:us-east-1:account-id:7rlst5mrgi",
        "arn:aws:iam::account-id:role/CodeStarWorker-project-id-CloudFormation",
        "arn:aws:iam::account-id:role/CodeStarWorker-project-id-CloudWatchEventRule",
        "arn:aws:iam::account-id:role/CodeStarWorker-project-id-CodeBuild",
        "arn:aws:iam::account-id:role/CodeStarWorker-project-id-CodePipeline",
        "arn:aws:iam::account-id:role/CodeStarWorker-project-id-Lambda",
        "arn:aws:lambda:us-east-1:account-id:function:awscodestar-project-id-lambda-
GetHelloWorld-KFKTXYNH9573",
        "arn:aws:s3:::aws-codestar-us-east-1-account-id-project-id-app",
        "arn:aws:s3:::aws-codestar-us-east-1-account-id-project-id-pipe"
      ]
    },
    {
      "Sid": "3",
      "Effect": "Allow",
```

```
      "Action": [
        "apigateway:GET",
        "config:Describe*",
        "config:Get*",
        "config:List*",
        "config:Put*",
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:DescribeLogGroups",
        "logs:PutLogEvents"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

프로젝트의 리소스 나열

이 예시에서는 AWS 계정의 지정된 IAM 사용자에게 AWS CodeStar 프로젝트의 리소스를 나열할 수 있는 액세스 권한을 부여하려고 합니다.

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codestar:ListResources",
      ],
      "Resource" : "arn:aws:codestar:us-east-2:project/my-first-projec"
    }
  ]
}
```

AWS CodeStar 콘솔 사용

AWS CodeStar 콘솔에 액세스하는 데 특정 권한이 필요하지는 않지만 `AWSCodeStarFullAccess` 정책 또는 AWS CodeStar 프로젝트 수준 역할 (소유자, 기여자 또는 뷰어) 중 하나가 없으면 유용한 작업을 수행할 수 없습니다. `AWSCodeStarFullAccess`에 대한 자세한 내용은 AWSCodeStarFullAccess

정책 단원을 참조하세요. 프로젝트 수준 정책에 대한 자세한 내용은 IAM팀 구성원을 위한 정책 단원을 참조하십시오.

또는 에만 전화를 거는 사용자에게는 최소 콘솔 권한을 허용할 필요가 없습니다. AWS CLI AWS API 대신 수행하려는 작업과 일치하는 API 작업에만 액세스를 허용하세요.

사용자가 자신이 권한을 볼 수 있도록 허용

이 예제에서는 IAM 사용자가 자신의 사용자 ID에 연결된 인라인 및 관리형 정책을 볼 수 있도록 하는 정책을 만드는 방법을 보여줍니다. 이 정책에는 콘솔에서 또는 OR를 사용하여 프로그래밍 방식으로 이 작업을 완료할 수 있는 AWS CLI 권한이 포함됩니다. AWS API

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
```

```
}
```

AWS CodeStar 프로젝트 업데이트

이 예시에서는 AWS 계정의 지정된 IAM 사용자에게 프로젝트 설명과 같은 AWS CodeStar 프로젝트 속성을 편집할 수 있는 액세스 권한을 부여하려고 합니다.

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codestar:UpdateProject"
      ],
      "Resource" : "arn:aws:codestar:us-east-2:project/my-first-projec"
    }
  ]
}
```

프로젝트에 팀원 추가

이 예시에서는 지정된 IAM 사용자에게 AWS CodeStar 프로젝트 ID를 사용하여 프로젝트에 팀 구성원을 추가할 수 있는 권한을 부여하려고 합니다.*my-first-projec*하지만 해당 사용자에게 팀 구성원을 제거할 수 있는 권한을 명시적으로 거부하려면:

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codestar:AssociateTeamMember",
      ],
      "Resource" : "arn:aws:codestar:us-east-2:project/my-first-projec"
    },
    {
      "Effect" : "Deny",
      "Action" : [
        "codestar:DisassociateTeamMember",
      ],
```

```
      "Resource" : "arn:aws:codestar:us-east-2:project/my-first-projec"
    }
      ]

  ]
}
```

계정과 관련된 사용자 프로필 목록 AWS

이 예시에서는 이 정책이 연결된 IAM 사용자가 AWS 계정과 관련된 모든 AWS CodeStar 사용자 프로필을 나열하도록 허용합니다.

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

태그 기반으로 AWS CodeStar 프로젝트 보기

ID 기반 정책의 조건을 사용하여 태그를 기반으로 AWS CodeStar 프로젝트에 대한 액세스를 제어할 수 있습니다. 이 예제에서는 프로젝트 보기를 허용하는 정책을 생성하는 방법을 보여줍니다. 그러나 프로젝트 태그 `Owner`에 해당 사용자의 사용자 이름 값이 있는 경우에만 권한이 부여됩니다. 이 정책은 콘솔에서 이 작업을 완료하는 데 필요한 권한도 부여합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ListProjectsInConsole",
            "Effect": "Allow",
            "Action": "codestar:ListProjects",
            "Resource": "*"
        },
```

```
        {
            "Sid": "ViewProjectIfOwner",
            "Effect": "Allow",
            "Action": "codestar:GetProject,
            "Resource": "arn:aws:codestar:*:*:project/*",
            "Condition": {
                "StringEquals": {"codestar:ResourceTag/Owner": "${aws:username}"}
            }
        }
    ]
}
```

계정의 IAM 사용자에게 이 정책을 첨부할 수 있습니다. 라는 사용자가 프로젝트를 `richard-roe` 보려고 하는 경우 AWS CodeStar 프로젝트에 `Owner=richard-roe` 또는 `owner=richard-roe` 태그를 지정해야 합니다. 그렇지 않으면 액세스가 거부됩니다. 조건 키 이름은 대소문자를 구분하지 않기 때문에 조건 태그 키 `Owner`는 `Owner` 및 `owner` 모두와 일치합니다. 자세한 내용은 IAM사용 설명서의 IAM JSON 정책 요소: 조건을 참조하십시오.

AWS CodeStarAWS 관리형 정책 업데이트

이 서비스가 이러한 변경 사항을 추적하기 시작한 AWS CodeStar 이후의 AWS 관리형 정책 업데이트에 대한 세부 정보를 볼 수 있습니다. 이 페이지의 변경 사항에 대한 자동 알림을 받으려면 AWS CodeStar 문서 기록 페이지에서 RSS 피드를 구독하십시오.

변경 사항	설명	날짜
AWSCodeStarFullAccess정책 — AWSCodeStarFullAccess 정책 업데이트	AWS CodeStar 액세스 역할 정책이 업데이트되었습니다. 정책의 결과는 동일하지만 ListStacks 클라우드포메이션에는 DescribeStacks 추가가 필요하며 이는 이미 필요합니다.	2023년 3월 24일
AWSCodeStarServiceRole 정책 — 정책 업데이트 AWSCodeStarServiceRole	정책 설명의 중복 작업을 수정하도록 AWS CodeStar 서비스 역할 정책이 업데이트되었습니다.	2021년 9월 23일

변경 사항	설명	날짜
	서비스 역할 정책은 AWS CodeStar 서비스가 사용자를 대신하여 작업을 수행하도록 허용합니다.	
AWS CodeStar 변경 내용 추적을 시작했습니다.	AWS CodeStar AWS 관리형 정책의 변경 사항 추적을 시작했습니다.	2021년 9월 23일

AWS CodeStarID 및 액세스 문제 해결

다음 정보를 사용하면 및 를 사용할 때 발생할 수 있는 일반적인 문제를 AWS CodeStar 진단하고 해결하는 데 도움이 IAM 됩니다.

주제

- 다음과 같은 작업을 수행할 권한이 없습니다. AWS CodeStar
- 저는 iam을 수행할 권한이 없습니다. PassRole
- 내 AWS 계정 외부의 사용자가 내 AWS CodeStar 리소스에 액세스할 수 있도록 허용하고 싶습니다.

다음과 같은 작업을 수행할 권한이 없습니다. AWS CodeStar

작업을 수행할 권한이 없다는 AWS Management Console 메시지가 표시되면 관리자에게 도움을 요청하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

다음 예제 오류는 `mateojackson` IAM 사용자가 콘솔을 사용하여 콘솔에 대한 세부 정보를 보려고 할 때 발생합니다.*`widget`* 하지만 `codestar:`*`GetWidget`* 권한이 없습니다.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 codestar:GetWidget on resource: my-example-widget
```

이 경우 Mateo는 *`my-example-widget`* 작업을 사용하여 `codestar:`*`GetWidget`* 리소스에 액세스하도록 허용하는 정책을 업데이트하라고 관리자에게 요청합니다.

저는 iam을 수행할 권한이 없습니다. PassRole

작업을 수행할 권한이 없다는 오류가 발생하는 경우 역할을 넘길 수 있도록 정책을 업데이트해야 합니다. `iam:PassRole` AWS CodeStar

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

다음 예제 오류는 라는 IAM 사용자가 콘솔을 사용하여 에서 작업을 `marymajor` 수행하려고 할 때 발생합니다. AWS CodeStar 하지만 작업을 수행하려면 서비스 역할이 부여한 권한이 서비스에 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 자격 증명을 제공한 사람입니다.

내 AWS 계정 외부의 사용자가 내 AWS CodeStar 리소스에 액세스할 수 있도록 허용하고 싶습니다.

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

자세히 알아보려면 다음을 참조하세요.

- 이러한 기능의 AWS CodeStar 지원 여부를 알아보려면 을 참조하십시오. 의 AWS CodeStar 작동 방식 IAM
- 소유한 리소스에 대한 액세스 권한을 AWS 계정 부여하는 방법을 알아보려면 사용 설명서에서 AWS 계정 자신이 소유한 다른 IAM 사용자의 액세스 권한 제공을 IAM 참조하십시오.
- 제3자에게 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM사용 설명서의 제3자가 AWS 계정 소유한 리소스에 대한 액세스 제공을 참조하십시오. AWS 계정
- ID 페더레이션을 통해 액세스를 제공하는 방법을 알아보려면 사용 설명서의 외부 인증된 사용자에게 액세스 제공 (ID 페더레이션) 을 IAM 참조하십시오.

- 계정 간 액세스에 대한 역할 사용과 리소스 기반 정책의 차이점을 알아보려면 사용 설명서의 계정 간 리소스 액세스를 참조하십시오. IAM IAM

AWS CloudTrail을 사용하여 AWS CodeStar API 직접 호출 로깅

AWS CodeStar는 AWS CodeStar에서 사용자, 역할, 또는 AWS 서비스가 수행한 작업에 대한 레코드를 제공하는 서비스인 AWS CloudTrail와 통합됩니다. CloudTrail은 AWS CodeStar에 대한 모든 API 직접 호출을 이벤트로 캡처합니다. 캡처되는 직접 호출에는 AWS CodeStar 콘솔에서 수행한 직접 호출과 AWS CodeStar API 작업에 대한 코드 직접 호출이 포함됩니다. 추적을 생성하면 AWS CodeStar 이벤트를 포함한 CloudTrail 이벤트를 지속적으로 S3 버킷에 배포할 수 있습니다. 추적을 구성하지 않은 경우에도 CloudTrail 콘솔의 이벤트 기록에서 최신 이벤트를 볼 수 있습니다. CloudTrail에서 수집한 정보를 사용하여 AWS CodeStar에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 기타 세부 정보를 확인할 수 있습니다.

CloudTrail에 대한 자세한 내용은 AWS CloudTrail 사용 설명서를 참조하세요.

CloudTrail의 AWS CodeStar 정보

CloudTrail은 계정 생성 시 AWS 계정에서 사용되도록 설정됩니다. AWS CodeStar에서 활동이 발생하면 해당 활동이 이벤트 기록의 다른 AWS 서비스 이벤트와 함께 CloudTrail 이벤트에 기록됩니다. AWS 계정에서 최신 이벤트를 확인, 검색 및 다운로드할 수 있습니다. 자세한 내용은 CloudTrail 이벤트 기록을 사용하여 이벤트 보기를 참조하세요.

AWS CodeStar에 대한 이벤트를 포함하여 AWS 계정에 이벤트를 지속적으로 기록하려면 추적을 생성합니다. 콘솔에서 추적을 생성하면 기본적으로 모든 AWS 리전에 추적이 적용됩니다. 추적은 AWS 파티션에 있는 모든 리전의 이벤트를 로깅하고 지정된 S3 버킷으로 로그 파일을 전송합니다. 다른 AWS 서비스를 구성하여 CloudTrail 로그에 수집된 이벤트 데이터를 추가로 분석하고 조치를 취할 수 있습니다. 자세한 내용은 다음을 참조하세요.

- 추적 생성 개요
- CloudTrail 지원 서비스 및 통합
- CloudTrail에 대한 Amazon SNS 알림 구성
- 여러 리전에서 CloudTrail 로그 파일 받기 및 여러 계정에서 CloudTrail 로그 파일 받기

모든 AWS CodeStar 작업은 CloudTrail에서 로깅되고 AWS CodeStar API 참조에 기록됩니다. 예를 들어 `DescribeProject`, `UpdateProject` 및 `AssociateTeamMember` 작업을 직접적으로 호출하면 CloudTrail 로그 파일에 항목이 생성됩니다.

모든 이벤트 및 로그 항목에는 요청을 생성한 사용자에 대한 정보가 들어 있습니다. 자격 증명 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청을 루트로 했는지 아니면 IAM 사용자 보안 인증 정보로 했는지 여부.
- 역할 또는 연합된 사용자에 대한 임시 보안 자격 증명을 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청했는지 여부.

자세한 내용은 CloudTrail userIdentity 요소를 참조하세요.

AWS CodeStar 로그 파일 항목 이해

CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함될 수 있습니다. 이벤트는 모든 소스의 단일 요청을 나타내며 요청된 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보를 포함합니다. CloudTrail 로그 파일은 퍼블릭 API 직접 호출의 주문 스택 트레이스가 아니므로 특정 순서로 표시되지 않습니다.

다음은 AWS CodeStar에서 직접적으로 호출되는 `CreateProject` 작업을 보여 주는 CloudTrail 로그 항목을 나타낸 예제입니다.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAJLIN2OF3UBEXAMPLE:role-name",
    "arn": "arn:aws:sts::account-ID:assumed-role/role-name/role-session-name",
    "accountId": "account-ID",
    "accessKeyId": "ASIAJ44LFQS5XEXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2017-06-04T23:56:57Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAJLIN2OF3UBEXAMPLE",
        "arn": "arn:aws:iam::account-ID:role/service-role/role-name",
        "accountId": "account-ID",
        "userName": "role-name"
      }
    },
    "invokedBy": "codestar.amazonaws.com"
  },
```

```
  "eventTime": "2017-06-04T23:56:57Z",
  "eventSource": "codestar.amazonaws.com",
  "eventName": "CreateProject",
  "awsRegion": "region-ID",
  "sourceIPAddress": "codestar.amazonaws.com",
  "userAgent": "codestar.amazonaws.com",
  "requestParameters": {
    "clientRequestToken": "arn:aws:cloudformation:region-ID:account-ID:stack/stack-
name/additional-ID",
    "id": "project-ID",
    "stackId": "arn:aws:cloudformation:region-ID:account-ID:stack/stack-
name/additional-ID",
    "description": "AWS CodeStar created project",
    "name": "project-name",
    "projectTemplateId": "arn:aws:codestar:region-ID::project-template/project-
template-name"
  },
  "responseElements": {
    "projectTemplateId": "arn:aws:codestar:region-ID::project-template/project-
template-name",
    "arn": "arn:aws:codestar:us-east-1:account-ID:project/project-ID",
    "clientRequestToken": "arn:aws:cloudformation:region-ID:account-ID:stack/stack-
name/additional-ID",
    "id": "project-ID"
  },
  "requestID": "7d7556d0-4981-11e7-a3bc-dd5daEXAMPLE",
  "eventID": "6b0d6e28-7a1e-4a73-981b-c8fdbEXAMPLE",
  "eventType": "AwsApiCall",
  "recipientAccountId": "account-ID"
}
```

AWS CodeStar의 규정 준수 확인

AWS CodeStar는 AWS 규정 준수 프로그램의 범위에 속하지 않습니다.

특정 규정 준수 프로그램 범위에 속하는 AWS 서비스의 목록은 규정 준수 프로그램 제공 AWS 범위 내 서비스 페이지에서 확인하세요. 일반 정보는 AWS 규정 준수 프로그램을 참조하세요.

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

AWS CodeStar의 복원성

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 중심으로 구축됩니다. AWS 리전은 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 가용 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 정보는 AWS 글로벌 인프라를 참조하세요.

의 인프라 보안 AWS CodeStar

관리형 서비스로서 AWS 글로벌 네트워크 보안으로 AWS CodeStar 보호됩니다. AWS 보안 서비스 및 인프라 AWS 보호 방법에 대한 자세한 내용은 AWS 클라우드 보안을 참조하십시오. 인프라 보안 모범 사례를 사용하여 AWS 환경을 설계하려면 Security Pillar AWS Well-Architected Framework의 인프라 보호를 참조하십시오.

AWS 게시된 API 호출을 사용하여 네트워크를 통해 CodeStar 액세스합니다. 고객은 다음을 지원해야 합니다.

- 전송 계층 보안 (TLS). TLS1.2가 필요하고 TLS 1.3을 권장합니다.
- (임시 디피-헬만) 또는 (타원 곡선 임시 디피-헬만PFS) 와 같이 완벽한 순방향 기밀성 DHE () 을 갖춘 암호 제품군. ECDHE Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

또한 액세스 키 ID와 보안 주체와 연결된 비밀 액세스 키를 사용하여 요청에 서명해야 합니다. IAM 또는 AWS Security Token Service(AWS STS)를 사용하여 임시 보안 인증을 생성하여 요청에 서명할 수 있습니다.

기본적으로 서비스 트래픽을 격리하지 AWS CodeStar 않습니다. AmazonEC2, API Gateway 또는 Elastic Beanstalk를 통해 액세스 설정을 수동으로 수정하지 않는 한, 를 사용하여 만든 프로젝트는 퍼블릭 인터넷에 AWS CodeStar 공개됩니다. 이는 의도적인 것입니다. AmazonEC2, API Gateway 또는 Elastic Beanstalk의 액세스 설정을 모든 인터넷 액세스 차단을 포함하여 원하는 수준으로 수정할 수 있습니다.

AWS CodeStar 는 기본적으로 VPC endpoints (AWS PrivateLink) 를 지원하지 않지만 프로젝트 리소스에서 직접 지원을 구성할 수 있습니다.

AWS CodeStar의 제한 값

다음 표는 AWS CodeStar의 제한을 설명합니다. AWS CodeStar는 프로젝트 리소스에 대한 다른 AWS 서비스에 의존합니다. 일부 서비스 제한이 변경될 수 있습니다. 변경 가능한 제한에 대한 자세한 내용은 AWS 서비스 제한 단원을 참조하십시오.

프로젝트 수	AWS 계정의 최대 프로젝트 수는 333개입니다. 실제 제한은 다른 서비스 종속성 수준(예: AWS 계정에 대해 허용되는 CodePipeline의 최대 파이프라인 수)에 따라 달라집니다.
IAM 사용자가 속할 수 있는 AWS CodeStar 프로젝트 수	개별 IAM 사용자당 최대 10개입니다.
프로젝트 ID	프로젝트 ID는 AWS 계정 내에서 고유해야 합니다. 프로젝트 ID는 2자 이상이어야 하며 15자를 초과할 수 없습니다. 허용되는 문자는 다음과 같습니다. a부터 z까지의 문자 0부터 9까지의 숫자 특수 문자 -(마이너스 부호) 대문자, 공백, .(마침표), @(at 기호), _(밑줄) 등과 같은 다른 문자는 허용되지 않습니다.
프로젝트 이름	프로젝트 이름은 100자를 초과할 수 없으며, 공백으로 시작하거나 끝날 수 없습니다.
제품 설명	0~1,024자의 문자 조합이면 됩니다. 프로젝트 설명은 선택 사항입니다.
AWS CodeStar 프로젝트의 팀원	100
사용자 프로필의 표시 이름	1~100자의 문자 조합이면 됩니다. 표시 이름은 하나 이상의 문자를 포함해야 합니다. 해당 문자

	가 공백일 수 없습니다. 표시 이름은 공백으로 시작하거나 끝날 수 없습니다.
사용자 프로필의 이메일 주소	이메일 주소는 @를 포함하고 유효한 도메인 확장명으로 끝나야 합니다.
AWS CodeStar에 대한 연동된 액세스, 루트 계정 액세스 또는 임시 액세스	AWS CodeStar는 연합된 사용자 및 임시 액세스 자격 증명 사용을 지원합니다. 루트 계정으로 AWS CodeStar를 사용하는 것은 권장되지 않습니다.
IAM 역할	관리형 정책의 최대 5,120자가 IAM 역할에 연결됩니다.

문제 해결 AWS CodeStar

다음은 AWS CodeStar에서 일반적으로 발생하는 문제를 해결하는 데 유용한 정보입니다.

주제

프로젝트 만들기 실패: 프로젝트가 만들어지지 않음

문제: 프로젝트를 만들려고 하면 만들기에 실패했다는 메시지가 나타납니다.

수정 방법: 가장 일반적인 실패 이유는 다음과 같습니다.

- 해당 ID의 프로젝트가 이미 AWS 계정에 존재하며, 다른 AWS 지역에 있을 수 있습니다.
- 로그인할 때 사용한 IAM 사용자에게는 프로젝트 생성에 필요한 권한이 AWS Management Console 없습니다.

- AWS CodeStar 서비스 역할에 필요한 권한이 하나 이상 누락되었습니다.
- 프로젝트의 하나 이상의 리소스에 대한 최대 한도 (예: IAM, Amazon S3 버킷 또는 파이프라인의 고객 관리형 정책 제한) 에 도달했습니다. CodePipeline

프로젝트를 만들기 전에 `AWSCodeStarFullAccess` 정책이 IAM 사용자에게 적용되었는지 확인하십시오. 자세한 정보는 AWSCodeStarFullAccess 정책을 참조하세요.

프로젝트를 만들 때 ID가 고유하고 AWS CodeStar 요구 사항을 충족해야 합니다. 자신을 대신하여 AWS 리소스를 관리할 수AWS CodeStar 있는 권한 요청 확인란을 선택했는지 확인하십시오.

다른 문제를 해결하려면 AWS CloudFormation 콘솔을 열고 만들려고 했던 프로젝트의 스택을 선택한 다음 이벤트 탭을 선택합니다. 프로젝트 하나에 두 개 이상의 스택이 있을 수 있습니다. 스택 이름은 `awscodestar-`로 시작하고 뒤에 프로젝트 ID가 옵니다. 스택이 [Deleted] 필터 보기 아래에 있을 수 있습니다. 스택 이벤트의 모든 실패 메시지를 검토하고 그러한 실패의 원인으로 나열된 문제를 해결합니다.

프로젝트 만들기: 프로젝트를 만들 때 Amazon EC2 구성을 편집하려고 하면 오류가 나타납니다.

문제: 프로젝트를 만드는 중에 Amazon EC2 구성 옵션을 편집하면, 오류 메시지 또는 회색으로 표시된 옵션이 나타나 프로젝트 만들기를 계속할 수 없습니다.

수정 방법: 오류 메시지가 나타나는 가장 일반적인 이유는 다음과 같습니다.

- AWS CodeStar 프로젝트 템플릿의 VPC (기본 VPC 또는 Amazon EC2 구성 편집 시 사용된 VPC) 에는 전용 인스턴스 테넌시가 있으며, 전용 인스턴스에는 인스턴스 유형이 지원되지 않습니다. 다른 인스턴스 유형 또는 다른 Amazon VPC를 선택합니다.
- AWS 계정에는 Amazon VPC가 없습니다. 기본 VPC를 삭제한 후 다른 VPC를 생성하지 않았을 수 있습니다. https://console.aws.amazon.com/vpc/ 에서 Amazon VPC 콘솔을 열고 내 VPC를 선택한 다음 하나 이상의 VPC가 구성되어 있는지 확인합니다. 없는 경우 하나를 만듭니다. 자세한 내용은 Amazon Virtual Private Cloud 시작 안내서의 Amazon VPC 개요 단원을 참조하세요.
- Amazon VPC에 서브넷이 없습니다. 다른 VPC를 선택하거나 VPC에 대한 서브넷을 생성합니다. 자세한 내용은 VPC 및 서브넷 기본 사항을 참조하십시오.

프로젝트 삭제: AWS CodeStar 프로젝트가 삭제되었지만 리소스가 아직 남아 있습니다.

문제: AWS CodeStar 프로젝트가 삭제되었지만 해당 프로젝트를 위해 생성된 리소스가 여전히 존재합니다. 기본적으로 프로젝트가 AWS CodeStar 삭제되면 프로젝트 리소스가 삭제됩니다. Amazon S3 버킷과 같은 일부 리소스는 사용자가 리소스 삭제 확인란을 선택하더라도 버킷에 데이터가 포함되어 있을 수 있으므로 보존됩니다.

가능한 해결 방법:AWS CloudFormation 콘솔을 열고 프로젝트를 만드는 데 사용된 AWS CloudFormation 스택을 하나 이상 찾으십시오. 스택 이름은 `awscodestar-`로 시작하고 뒤에 프로젝트 ID가 옵니다. 스택이 [Deleted] 필터 보기 아래에 있을 수 있습니다. 스택과 관련된 이벤트를 검토하여 해당 프로젝트에 대해 만든 리소스를 찾습니다. AWS CodeStar 프로젝트를 생성한 AWS 지역의 각 리소스에 대한 콘솔을 연 다음 리소스를 수동으로 삭제하십시오.

남아 있을 수 있는 프로젝트 리소스는 다음과 같습니다.

- Amazon S3에 있는 하나 이상의 프로젝트 버킷입니다. 다른 프로젝트 리소스와 달리 Amazon S3의 프로젝트 버킷은 프로젝트와 함께 AWS CodeStar 관련 AWS 리소스 삭제 확인란을 선택해도 삭제되지 않습니다.

  https://console.aws.amazon.com/s3/에서 S3 콘솔을 엽니다.
- 내 프로젝트의 소스 리포지토리. CodeCommit

  https://console.aws.amazon.com/codecommit/ 에서 CodeCommit 콘솔을 엽니다.
- 에서 프로젝트를 위한 파이프라인 CodePipeline.

  https://console.aws.amazon.com/codepipeline/ 에서 CodePipeline 콘솔을 엽니다.
- 의 애플리케이션 및 관련 배포 그룹 CodeDeploy

  https://console.aws.amazon.com/codedeploy/ 에서 CodeDeploy 콘솔을 엽니다.
- AWS Elastic Beanstalk의 애플리케이션 및 관련 환경입니다.

  https://console.aws.amazon.com/elasticbeanstalk/에서 Elastic Beanstalk 콘솔을 엽니다.
- AWS Lambda의 함수입니다.

  https://console.aws.amazon.com/lambda/ 에서 AWS Lambda 콘솔을 엽니다.
- API Gateway에 있는 하나 이상의 API입니다.

https://console.aws.amazon.com/apigateway/에서 Amazon API Gateway 콘솔을 엽니다.

- IAM에 있는 하나 이상의 IAM 정책 또는 역할입니다.

  에 AWS Management Console 로그인하고 https://console.aws.amazon.com/iam/ 에서 IAM 콘솔을 엽니다.

- Amazon EC2에 있는 인스턴스입니다.

  https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.

- 에 하나 이상의 개발 환경이 있습니다. AWS Cloud9

  개발 환경을 보고, 액세스하고, 관리하려면 https://console.aws.amazon.com/cloud9/ 에서 AWS Cloud9 콘솔을 여십시오.

프로젝트에서 외부 리소스 AWS (예: Atlassian JIRA의 GitHub 리포지토리 또는 이슈) 를 사용하는 경우 CodeStar 프로젝트와 함께 관련 리소스 삭제 상자를 선택해도 해당 AWS 리소스는 삭제되지 않습니다.

팀 관리 실패: 프로젝트의 팀에 IAM 사용자를 추가할 수 없습니다. AWS CodeStar

문제: 프로젝트에 사용자를 추가하려고 하면 추가에 실패했다는 오류 메시지가 나타납니다.

수정 방법: 이 오류가 발생하는 가장 일반적인 이유는 해당 사용자가 IAM의 사용자에게 적용할 수 있는 관리형 정책의 한도에 도달했기 때문입니다. 사용자를 추가하려고 시도한 AWS CodeStar 프로젝트에 소유자 역할이 없거나 IAM 사용자가 없거나 삭제된 경우에도 이 오류가 발생할 수 있습니다.

해당 AWS CodeStar 프로젝트의 소유자인 사용자로 로그인했는지 확인하세요. 자세한 정보는 AWS CodeStar 프로젝트에 팀원 추가 을 참조하세요.

다른 문제를 해결하려면 IAM 콘솔을 열고, 추가하려 한 사용자를 선택한 다음, 해당 IAM 사용자에게 적용되는 관리형 정책의 수를 확인합니다.

자세한 내용은 IAM 개체 및 객체에 대한 제한을 참조하십시오. 변경할 수 있는 제한은 AWS 서비스 제한을 참조하십시오.

액세스 실패: 페더레이션 사용자는 프로젝트에 접근할 수 없습니다. AWS CodeStar

문제: 페더레이션 사용자는 콘솔에서 프로젝트를 볼 수 없습니다. AWS CodeStar

수정 방법: 연합된 사용자로 로그인했다면, 위임 받는 역할에 적절한 관리형 정책이 연결돼 있는지 확인하십시오. 자세한 정보는 프로젝트의 AWS CodeStar 뷰어/기여자/소유자 관리형 정책을 연동 사용자의 역할에 연결합니다.을 참조하세요.

정책을 수동으로 연결하여 페더레이션 사용자를 AWS Cloud9 환경에 추가하십시오. 연동 사용자 역할에 AWS Cloud9 관리형 정책 연결를 참조하세요.

액세스 실패: 페더레이션 사용자는 환경에 액세스하거나 환경을 만들 수 없습니다. AWS Cloud9

문제: 페더레이션 사용자는 콘솔에서 AWS Cloud9 환경을 보거나 만들 수 없습니다. AWS Cloud9

수정 방법: 연합된 사용자로 로그인했다면, 연합된 사용자의 역할에 적절한 관리형 정책이 연결돼 있는지 확인하십시오.

페더레이션된 사용자 역할에 정책을 수동으로 연결하여 페더레이션 사용자를 AWS Cloud9 환경에 추가합니다. 연동 사용자 역할에 AWS Cloud9 관리형 정책 연결를 참조하세요.

액세스 실패: 페더레이션 사용자는 프로젝트를 만들 수 있지만 AWS CodeStar 프로젝트 리소스를 볼 수는 없습니다.

문제: 연합된 사용자가 프로젝트를 생성할 수 있지만, 프로젝트 파이프라인 같은 프로젝트 리소스를 볼 수는 없습니다.

가능한 해결 방법: **`AWSCodeStarFullAccess`**관리형 정책을 연결한 경우 에서 AWS CodeStar프로젝트를 만들 수 있는 권한이 있습니다. 하지만 모든 프로젝트 리소스에 액세스하려면 소유자 관리형 정책을 연결해야 합니다.

프로젝트 리소스를 AWS CodeStar 만든 후에는 소유자, 기여자 및 뷰어 관리 정책에서 모든 프로젝트 리소스에 대한 프로젝트 권한을 사용할 수 있습니다. 모든 리소스에 액세스하려면, 소유자 정책을 역할을 직접 연결해야 합니다. 3단계: 사용자의 IAM 권한 구성를 참조하세요.

서비스 역할 문제: 서비스 역할을 만들 수 없습니다.

문제: 에서 프로젝트를 만들려고 AWS CodeStar하면 서비스 역할을 만들라는 메시지가 표시됩니다. 프로젝트를 만드는 옵션을 선택하면 오류가 나타납니다.

가능한 해결 방법: 이 오류가 발생하는 가장 일반적인 이유는 서비스 역할을 만들 수 있는 충분한 권한이 없는 계정으로 로그인했기 때문입니다. AWS AWS CodeStar 서비스 역할 (`aws-codestar-service-role`) 을 만들려면 관리자 또는 루트 계정으로 로그인해야 합니다. 콘솔에서 로그아웃한 후, 적용되는 `AdministratorAccess` 관리형 정책이 있는 IAM 사용자로 다시 로그인합니다.

서비스 역할 문제: 서비스 역할이 유효하지 않거나 없습니다.

문제: AWS CodeStar 콘솔을 열면 AWS CodeStar 서비스 역할이 없거나 유효하지 않다는 메시지가 표시됩니다.

수정 방법: 이 오류가 발생하는 가장 일반적인 이유는 관리 사용자가 서비스 역할(`aws-codestar-service-role`)을 편집했거나 삭제했기 때문입니다. 서비스 역할이 삭제된 경우 이를 만들라는 메시지가 나타납니다. 역할을 만들려면 관리 사용자 또는 루트 계정으로 로그인해야 합니다. 역할이 편집된 경우 더 이상 유효하지 않습니다. 관리 사용자로 IAM 콘솔에 로그인하고, 역할 목록에서 해당 서비스 역할을 찾은 후, 이를 삭제합니다. AWS CodeStar 콘솔로 전환하고 지침에 따라 서비스 역할을 생성합니다.

프로젝트 역할 문제: AWS CodeStar 프로젝트 내 인스턴스의 AWS Elastic Beanstalk 상태 확인이 실패합니다.

문제: 2017년 9월 22일 이전에 Elastic Beanstalk를 포함하는 AWS CodeStar 프로젝트를 생성한 경우 Elastic Beanstalk 상태 확인이 실패할 수 있습니다. 프로젝트 생성 후 Elastic Beanstalk 구성을 한 번도 변경하지 않았다면, 정상 상태 확인이 실패하고 회색 상태가 보고됩니다. 정상 상태 확인이 실패하더라도, 애플리케이션은 정상적으로 실행됩니다. 프로젝트 생성 후 Elastic Beanstalk 구성을 한 번도 변경하지 않았다면, 정상 상태 확인이 실패하고 애플리케이션이 올바르게 작동하지 않을 수 있습니다.

수정 방법: 하나 이상의 IAM 역할에 필수 IAM 정책 설명이 누락되어 있습니다. 누락된 정책을 AWS 계정의 영향받는 역할에 추가하십시오.

1. AWS Management Console 로그인하고 https://console.aws.amazon.com/iam/ 에서 IAM 콘솔을 엽니다.

   (이렇게 할 수 없는 경우 AWS 계정 관리자에게 도움을 요청하십시오.)

2. 탐색 창에서 역할을 선택합니다.
3. 역할 목록에서 ***Project-ID*** -EB를 선택합니다CodeStarWorker. 여기서 Project-ID는 ***### ## ### ###*** 중 하나의 ID입니다. (목록에서 역할을 쉽게 찾을 수 없다면, 검색 상자에 역할 이름 일부 또는 전부를 입력하십시오.)
4. 권한 탭에서 정책 연결을 선택합니다.
5. 정책 목록에서 및 을 선택합니다. AWSElasticBeanstalkEnhancedHealthAWSElasticBeanstalkService (목록에서 정책을 쉽게 찾을 수 없다면, 검색 상자에 정책 이름 일부 또는 전부를 입력하십시오.)
6. 정책 연결을 선택하세요.
7. 이름이 ***Project-ID CodeStarWorker*** -EB 패턴을 따르는 영향을 받는 각 역할에 대해 3~6단계를 반복합니다.

프로젝트 역할 문제: 프로젝트 역할이 유효하지 않거나 없습니다.

문제: 프로젝트에 사용자를 추가하려고 하면 프로젝트 역할에 대한 정책이 없거나 유효하지 않아 추가에 실패했다는 오류 메시지가 나타납니다.

수정 방법: 이 오류가 발생하는 가장 일반적인 이유는 IAM에서 하나 이상의 프로젝트 정책이 편집되었거나 삭제되었기 때문입니다. 프로젝트 정책은 프로젝트별로 고유하며 다시 AWS CodeStar 만들 수 없습니다. 이 프로젝트는 사용할 수 없습니다. 에서 AWS CodeStar프로젝트를 만든 다음 데이터를 새 프로젝트로 마이그레이션하십시오. 사용할 수 없는 프로젝트의 리포지토리에서 프로젝트 코드를 복제한 다음, 새 프로젝트의 리포지토리로 이 코드를 푸시합니다. 이전 프로젝트의 팀 wiki 정보를 새 프로젝트에 복사합니다. 새 프로젝트에 사용자를 추가합니다. 데이터와 설정을 모두 마이그레이션했으면 사용할 수 없는 프로젝트를 삭제합니다.

프로젝트 확장명: JIRA에 연결할 수 없습니다.

문제: Atlassian JIRA 확장 프로그램을 사용하여 AWS CodeStar 프로젝트를 JIRA 인스턴스에 연결하려고 하면 다음과 같은 메시지가 표시됩니다. “URL이 유효한 JIRA URL이 아닙니다. URL이 올바른지 확인하십시오."라는 메시지가 나타납니다.

수정 방법:

- JIRA URL이 올바른지 확인한 후 다시 연결을 시도합니다.
- 퍼블릭 인터넷에서 자체 호스팅된 JIRA 인스턴스에 액세스할 수 없습니다. 네트워크 관리자에게 문의하여 퍼블릭 인터넷에서 JIRA 인스턴스에 액세스할 수 있는지 확인한 후 다시 연결을 시도합니다.

GitHub: 리포지토리의 커밋 기록, 이슈 또는 코드에 액세스할 수 없습니다.

문제: 코드를 저장하는 프로젝트의 대시보드에서 커밋 기록 및 GitHub이슈 타일에 연결 오류가 표시되거나 이러한 타일에서 열기 GitHub 또는 이슈 만들기를 선택하면 오류가 표시됩니다. GitHub

가능한 원인:

- AWS CodeStar 프로젝트가 더 이상 GitHub 리포지토리에 액세스하지 못할 수 있습니다.
- 리포지토리가 삭제되었거나 에서 GitHub 이름이 변경되었을 수 있습니다.

AWS CloudFormation: 누락된 권한 때문에 스택 생성이 취소됨

리소스를 `template.yml` 파일에 추가하고 나면, AWS CloudFormation 스택 업데이트를 확인해 오류 메시지가 없는지 확인하십시오. 특정 기준이 충족되지 않으면(예: 필수 리소스 권한이 누락됨) 스택 업데이트가 실패합니다.

> **Note**
>
> 2019년 5월 2일부터 모든 기존 프로젝트의 AWS CloudFormation 작업자 역할 정책이 업데이트되었습니다. 이로써 사용자의 프로젝트 보안 강화를 위해 프로젝트 파이프라인에 부여된 액세스 권한 범위가 줄어듭니다.

문제를 해결하려면 프로젝트 파이프라인의 AWS CodeStar 대시보드 보기에서 실패 상태를 확인하세요.

그런 다음, 파이프라인의 배포 단계에서 CloudFormation링크를 선택하여 콘솔에서 장애를 해결하세요. AWS CloudFormation 스택 생성 세부 정보를 보려면, 프로젝트의 이벤트 목록을 확장해 오류 메시지를 확인하십시오. 메시지는 어떤 권한이 누락되었는지를 표시합니다. AWS CloudFormation 작업자 역할 정책을 수정하고 파이프라인을 다시 실행합니다.

AWS CloudFormation Lambda 실행 PassRole 역할에서 iam:을 수행할 권한이 없습니다.

2018년 12월 6일 (PDT) 이전에 Lambda 함수를 생성하는 프로젝트를 생성한 경우 다음과 같은 오류가 AWS CloudFormation 표시될 수 있습니다.

```
User: arn:aws:sts::id:assumed-role/CodeStarWorker-project-id-CloudFormation/
AWSCloudFormation is not authorized to perform: iam:PassRole on resource:
 arn:aws:iam::id:role/CodeStarWorker-project-id-Lambda (Service: AWSLambdaInternal;
 Status Code: 403; Error Code: AccessDeniedException; Request ID: id)
```

이 오류는 AWS CloudFormation 작업자 역할에 새 Lambda 함수를 프로비저닝하기 위한 역할을 전달할 권한이 없기 때문에 발생합니다.

이 오류를 수정하려면 다음 스니펫으로 AWS CloudFormation 작업자 역할 정책을 업데이트해야 합니다.

```
{
        "Action":[ "iam:PassRole" ],
        "Resource": [
           "arn:aws:iam::account-id:role/CodeStarWorker-project-id-Lambda",
           ],

        "Effect": "Allow"
}
```

정책을 업데이트한 후 파이프라인을 다시 실행합니다.

또는 기존 프로젝트에 IAM 권한 경계 추가에 설명된 대로 프로젝트에 권한 경계를 추가하여 Lambda 함수에 대한 사용자 지정 역할을 사용할 수 있습니다.

리포지토리에 대한 연결을 생성할 수 없습니다. GitHub

문제:

GitHub 리포지토리 연결에는 AWS Connector 형식이 사용되므로 연결을 만들려면 리포지토리에 대한 GitHub 조직 소유자 권한 또는 관리자 권한이 필요합니다.

가능한 해결 방법: GitHub 리포지토리의 권한 수준에 대한 자세한 내용은 https://docs.github.com/en/free-pro-team @latest /github/ -/setting-up-and-managing-ation을 참조하십시오. organizations-and-teams permission-levels-for-an

AWS CodeStar 사용 설명서 릴리스 정보

다음 표에서는 AWS CodeStar 사용 설명서의 각 릴리스에서 변경된 중요 사항에 대해 설명합니다. 이 설명서에 대한 업데이트 알림을 받으려면 RSS 피드를 구독하면 됩니다.

변경 사항	설명	날짜
액세스 정책 업데이트	AWS CodeStar 액세스 역할 정책이 업데이트되었습니다. 정책의 결과는 같지만 클라우드포메이션에는 이미 필요한 DescriveStacks와 함께 ListStacks가 필요합니다. 업데이트된 정책을 참조하려면 AWS CodeStarFullAccess 정책을 참조하십시오.	2023년 3월 24일
서비스 역할 정책 업데이트	AWS CodeStar 액세스 역할 정책이 업데이트되었습니다. 업데이트된 정책을 참조하려면 AWSCodeStarServiceRole 정책을 참조하십시오.	2021년 9월 23일
GitHub 소스 리포지토리가 있는 프로젝트에 연결 리소스 사용	콘솔을 사용하여 AWS CodeStar에서 GitHub 리포지토리로 프로젝트를 생성하면 연결 리소스가 GitHub 작업을 관리하는 데 사용됩니다. 연결은 GitHub 앱을 사용하는 반면, 이전 GitHub 인증에서는 OAuth를 사용했습니다. GitHub 연결을 사용하는 프로젝트를 생성하는 방법을 보여주는 자습서는 자습서: GitHub 소스 리포지토리로 프로젝트 생성을 참조하십시오. 또한 자습서에서는 프로젝트 소스 리	2021년 4월 27일

	포지토리에 대한 Pull 요청을 만들고, 검토하고, 병합하는 방법도 보여줍니다.	
AWS CodeStar가 미국 서부(캘리포니아 북부) 리전에서 AWS Cloud9을 지원합니다.	AWS CodeStar가 이제 미국 서부(캘리포니아 북부) 리전에서 AWS Cloud9 사용을 지원합니다. 자세한 내용은 Cloud9 설정을 참조하세요.	2021년 2월 16일
새로운 콘솔 경험을 반영하도록 설명서를 업데이트하십시오.	2020년 8월 12일에 AWS CodeStar 서비스가 AWS 콘솔의 새로운 사용자 환경으로 이동했습니다. 사용자 가이드를 새로운 콘솔 환경과 일치하도록 업데이트했습니다.	2020년 8월 12일
AWS CodeStar 프로젝트는 AWS CodeStar CLI로 생성할 수 있습니다.	AWS CodeStar 프로젝트는 CLI 명령으로 생성할 수 있습니다. AWS CodeStar는 소스 코드와 사용자가 제공한 도구 체인 템플릿을 이용해 프로젝트와 인프라를 생성합니다. AWS CodeStar(AWS CLI)에서 프로젝트 생성 단원을 참조하십시오.	2018년 10월 24일

이제 모든AWS CodeStar 프로젝트 템플릿에는 인프라 업데이트를 위한 AWS CloudFormation 파일이 포함됩니다.	AWS CodeStar는 AWS CloudFormation과 함께 작동해 사용자가 코드를 사용해 클라우드에서 지원 서비스와 서버 또는 서버리스 플랫폼을 생성하게 합니다. 이제 AWS CloudFormation 파일을 모든 AWS CodeStar 프로젝트 템플릿 유형(Lambda, EC2 또는 Elastic Beanstalk 컴퓨팅 플랫폼을 이용하는 템플릿)에서 이용할 수 있습니다. 파일은 소스 리포지토리의 `template.yml` 에 저장됩니다. 파일을 확인하고 수정해 프로젝트에 리소스를 추가할 수 있습니다. 자세한 내용은 프로젝트 템플릿을 참조하십시오.	2018년 8월 3일
RSS에서 AWS CodeStar 사용 설명서 업데이트 알림 이용 가능	AWS CodeStar 사용 설명서의 HTML 버전이 설명서 업데이트 릴리스 정보 페이지에 명시된 업데이트의 RSS 피드를 지원합니다. RSS 피드에는 2018년 6월 30일 이후의 업데이트가 포함됩니다. 이전에 발표한 업데이트도 설명서 업데이트 릴리스 정보 페이지에서 이용할 수 있습니다. 피드를 구독하려면 상단 메뉴판에서 RSS 버튼을 누릅니다.	2018년 6월 30일

다음 표에서는 2018년 6월 30일 이전 AWS CodeStar 사용 설명서의 각 릴리스에서 변경된 중요 사항에 대해 설명합니다.

변경 사항	설명	변경 날짜
이제 GitHub에서도 AWS CodeStar 사용 설명서를 사용할 수 있습니다.	이제 GitHub에서도 이 설명서를 사용할 수 있습니다. 또한 GitHub를 이용해 이 설명서의 콘텐츠에 대한 피드백과 변경 요구사항을 제출할 수도 있습니다. 자세한 내용을 확인하려면 설명서의 탐색 모음에서 GitHub에서 편집 아이콘을 선택하거나, GitHub 웹사이트의 awsdocs/aws-codestar-user-guide 리포지토리를 참조하십시오.	2018년 2월 22일
아시아 태평양(서울)에서 AWS CodeStar 구매 가능	AWS CodeStar는 이제 아시아 태평양(서울) 리전에서 사용할 수 있습니다. 자세한 내용은 AWS CodeStar의 Amazon Web Services 일반 참조 섹션을 참조하세요.	2018년 2월 14일
이제 아시아 태평양(도쿄) 및 캐나다(중부)에서 AWS CodeStar 구매 가능	이제 아시아 태평양(도쿄) 및 캐나다(중부) 리전에서 AWS CodeStar 구매 가능 자세한 내용은 AWS CodeStar의 Amazon Web Services 일반 참조 섹션을 참조하세요.	2017년 12월 20일
AWS CodeStar가 이제 AWS Cloud9을 지원합니다.	AWS CodeStar가 이제 웹 브라우저 기반 온라인 IDE인 AWS Cloud9을 프로젝트 코드와 함께 사용하는 기능을 지원합니다. 자세한 내용은 AWS CodeStar와 함께 AWS Cloud9 사용 섹션을 참조하세요. 지원되는 AWS 리전 목록은 Amazon Web Services 일반 참조의 AWS Cloud9 섹션을 참조하세요.	2017년 11월 30일
AWS CodeStar가 GitHub 지원	AWS CodeStar가 이제 GitHub에 프로젝트 코드를 저장하는 기능을 지원합니다. 자세한 내용은 프로젝트 만들기를 참조하십시오.	2017년 10월 12일
이제 미국 서부(캘리포니아 북부) 및 유럽(런던)에서 AWS CodeStar 구매 가능	이제 미국 서부(캘리포니아 북부) 및 유럽(런던) 리전에서 AWS CodeStar 구매 가능 자세한 내용은 AWS CodeStar의 Amazon Web Services 일반 참조 섹션을 참조하세요.	2017년 8월 17일
이제 아시아 태평양(시드니), 아시아 태평양(싱가포르) 및	이제 아시아 태평양(시드니), 아시아 태평양(싱가포르) 및 유럽(프랑크푸르트) 리전에서 AWS CodeStar 구매 가능 자세	2017년 7월 25일

변경 사항	설명	변경 날짜
유럽(프랑크푸르트)에서 AWS CodeStar 구매 가능	한 내용은 AWS CodeStar의 Amazon Web Services 일반 참조 섹션을 참조하세요.	
AWS CloudTrail가 이제 AWS CodeStar을 지원합니다.	AWS CodeStar가 CloudTrail과 통합되었습니다. 이 서비스는 AWS 계정에서 AWS CodeStar에 의해 또는 이를 대신하여 수행된 API 직접 호출을 캡처하고 사용자가 지정하는 Amazon S3 버킷에 로그 파일을 전송합니다. 자세한 내용은 AWS CloudTrail을 사용하여 AWS CodeStar API 직접 호출 로깅 섹션을 참조하세요.	2017년 6월 14일
최초 릴리스	이 설명서는 AWS CodeStar 사용 설명서의 첫 번째 릴리스입니다.	2017년 4월 19일

AWS 용어집

최신 AWS 용어는 AWS 용어집 참조서의 AWS 용어집을 참조하세요.