



사용자 가이드

개발자 도구 콘솔



개발자 도구 콘솔: 사용자 가이드

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

상표 및 브랜드 디자인은 타사 제품 또는 서비스와 함께, 당사 브랜드 이미지를 떨어뜨리거나 고객에게 혼동을 일으킬 수 있는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon 계열사, 관련 업체 또는 Amazon의 지원 업체 여부에 상관없이 해당 소유자의 자산입니다.

Table of Contents

개발자 도구 콘솔이란 무엇입니까?	1
를 처음 사용하십니까?	3
개발자 도구 콘솔의 기능	3
알림이란 무엇입니까?	3
알림으로 할 수 있는 작업은 무엇입니까?	4
알림은 어떻게 작동합니까?	4
알림을 시작하려면 어떻게 해야 합니까?	4
알림 관련 개념	4
설정	12
알림 시작하기	18
알림 규칙 사용	25
알림 규칙 대상 사용	37
알림과 AWS Chatbot 간의 통합 구성	46
AWS CloudTrail을 사용하여 AWS CodeStar Notifications API 호출 로깅	50
문제 해결	54
할당량	56
연결이란 무엇입니까?	57
연결로 어떤 작업을 할 수 있습니까?	57
어떤 서드 파티 공급자에 대한 연결을 생성할 수 있나요?	57
연결과 AWS 서비스 통합되는 것은 무엇인가요?	58
연결은 어떻게 작동합니까?	58
연결을 시작하려면 어떻게 해야 합니까?	62
연결 관련 개념	63
AWS CodeStar 연결, 지원되는 공급자 및 버전	63
AWS CodeStar Connections와 제품 및 서비스 통합	65
연결 설정	67
연결 시작하기	70
연결 관련 작업	76
호스트 작업	127
연결된 리포지토리의 동기화 구성 작업	137
CloudTrail을 사용하여 연결 API 호출 로깅	147
VPC 엔드포인트(AWS PrivateLink)	149
연결 문제 해결	153
할당량	164

허용 목록에 추가할 IP 주소	164
보안	167
알림 콘텐츠 및 보안 이해	167
데이터 보호	168
자격 증명 및 액세스 관리	169
고객	170
자격 증명을 통한 인증	171
정책을 사용한 액세스 관리	173
개발자 도구 콘솔의 기능이 IAM에서 작동하는 방식	174
AWS CodeConnections 권한 참조	180
자격 증명 기반 정책 예시	195
태그를 사용하여 AWS CodeStar Connections 리소스에 대한 액세스를 제어합니다.	208
콘솔 사용	209
사용자가 자신이 권한을 볼 수 있도록 허용	210
문제 해결	211
AWS CodeStar Notifications에 서비스 연결 역할 사용	213
AWS CodeConnections의 서비스 연결 역할 사용	218
AWS 관리형 정책	220
규정 준수 확인	222
복원성	223
인프라 보안	223
여러 리전에 걸친 AWS CodeConnections 리소스 간 트래픽	224
사용 설명서 기록	225
AWS 용어집	230
.....	ccxxxi

개발자 도구 콘솔이란 무엇입니까?

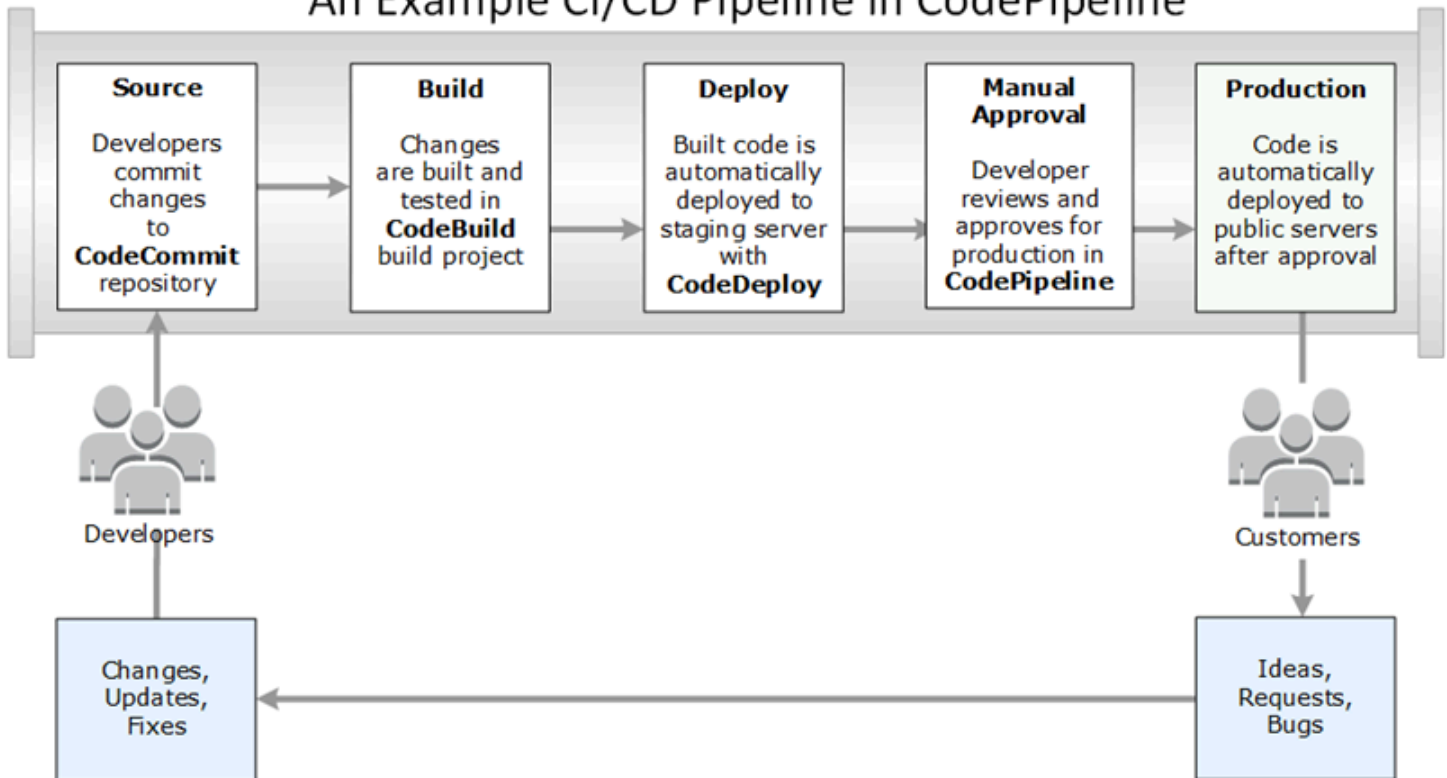
개발자 도구 콘솔에는 개별적으로 또는 팀으로 소프트웨어 개발을 돕기 위해 개별적으로 또는 집합적으로 사용할 수 있는 일련의 서비스 및 기능이 있습니다. 개발자 도구를 사용하면 소프트웨어를 안전하게 저장, 빌드, 테스트 및 배포할 수 있습니다. 개별 또는 집합적으로 사용되는 이러한 도구는 DevOps, CI/CD(지속적 통합 및 지속적 전달)를 지원합니다.

개발자 도구 콘솔에는 다음 서비스가 포함되어 있습니다.

- [AWS CodeCommit](#)은 프라이빗 Git 리포지토리를 호스팅하는 완전 관리형 소스 제어 서비스입니다. 리포지토리를 사용하여 AWS 클라우드의 자산(문서, 소스 코드 및 이진 파일 등)을 비공개로 저장하고 관리할 수 있습니다. 또한 첫 번째 커밋부터 마지막 변경 내용까지 프로젝트 기록을 저장합니다. 코드 품질을 보장하기 위해 코드에 주석을 달고 풀 요청을 생성하여 리포지토리의 코드를 공동으로 작업할 수 있습니다.
- [AWS CodeBuild](#)는 소스 코드를 컴파일하고 단위 테스트를 실행하며 배포할 준비가 완료된 아티팩트를 생성하는 완전 관리형 빌드 서비스입니다. 이 서비스는 Apache Maven, Gradle 등과 같은 널리 사용되는 프로그래밍 언어 및 빌드 도구에 맞게 사전 패키징된 빌드 환경을 제공합니다. CodeBuild에서 빌드 환경을 사용자 지정하여 사용자 고유의 빌드 도구를 사용할 수도 있습니다.
- [AWS CodeDeploy](#)는 Amazon EC2, AWS Lambda 및 온프레미스 서버와 같은 서비스를 계산하기 위해 소프트웨어 배포를 자동화하는 완전 관리형 배포 서비스입니다. 새로운 기능을 신속하게 출시하고, 애플리케이션을 배포하는 동안 가동 중지를 방지하며, 애플리케이션 업데이트의 복잡성을 처리할 수 있습니다.
- [AWS CodePipeline](#)은 소프트웨어 출시에 필요한 단계를 모델링, 시각화 및 자동화하는 데 사용할 수 있는 지속적 통합 및 지속적 전달 서비스입니다. 소프트웨어 릴리스 프로세스를 구성하는 여러 단계를 신속하게 모델링하고 구성할 수 있습니다. 사용자가 정의한 릴리스 프로세스 모델에 따라 코드가 변경될 때마다 코드를 빌드, 테스트 및 배포합니다.

다음은 개발자 도구 콘솔의 서비스를 함께 사용하여 소프트웨어를 개발하는 방법에 대한 예입니다.

An Example CI/CD Pipeline in CodePipeline



이 예에서 개발자는 CodeCommit에서 리포지토리를 생성하고 이를 사용하여 코드를 개발하고 공동 작업합니다. CodeBuild에서 빌드 프로젝트를 생성하여 코드를 빌드 및 테스트하고 CodeDeploy를 사용하여 코드를 테스트 및 프로덕션 환경에 배포합니다. 개발자들은 빠르게 반복하기를 원하므로 CodePipeline에서 파이프라인을 생성하여 CodeCommit 리포지토리의 변경 사항을 감지합니다. 이러한 변경 사항이 빌드되고 테스트가 실행되며 성공적으로 빌드되고 테스트된 코드가 테스트 서버에 배포됩니다. 팀은 테스트 단계를 파이프라인에 추가하여 스테이징 서버에서 통합 또는 로드 테스트와 같은 더 많은 테스트를 실행합니다. 이러한 테스트가 성공적으로 완료되면, 팀 구성원이 결과를 검토하여 만족하는 경우 수동으로 프로덕션에 대한 변경 사항을 승인합니다. CodePipeline은 테스트되고 승인된 코드를 프로덕션 인스턴스에 배포합니다.

이는 개발자 도구 콘솔에서 사용 가능한 서비스를 하나 이상 사용하여 소프트웨어를 개발하는 데 도움이 되는 간단한 예입니다. 각 서비스는 사용자의 요구에 맞게 사용자 지정할 수 있습니다. AWS 및 기타 타사 도구의 다른 제품 및 서비스와의 많은 통합을 제공합니다. 자세한 정보는 다음 주제를 참조하세요.

- CodeCommit: [제품 및 서비스 통합](#)
- CodeBuild: [Jenkins에 CodeBuild 사용](#)
- CodeDeploy: [제품 및 서비스 통합](#)

- CodePipeline: [제품 및 서비스 통합](#)

를 처음 사용하십니까?

개발자 도구 콘솔에서 사용할 수 있는 서비스 중 하나 이상을 처음 사용하는 경우 다음 주제를 읽어 보는 것이 좋습니다.

- [CodeCommit 시작하기](#)
- [CodeBuild 시작하기](#), [개념](#)
- [CodeDeploy 시작하기](#), [기본 구성 요소](#)
- [CodePipeline 시작하기](#), [개념](#)

개발자 도구 콘솔의 기능

개발자 도구 콘솔에는 다음 기능이 포함되어 있습니다.

- 개발자 도구 콘솔에는 AWS CodeBuild, AWS CodeCommit, AWS CodeDeploy 및 AWS CodePipeline에서 이벤트를 구독하는 데 사용할 수 있는 알림 관리자 기능이 포함되어 있습니다. 이 기능에는 자체 API인 AWS CodeStar Notifications가 있습니다. 알림 기능을 사용하여 사용자에게 작업에 가장 중요한 리포지토리, 빌드 프로젝트, 배포 애플리케이션 및 파이프라인의 이벤트를 신속하게 알릴 수 있습니다. 알림 관리자는 사용자가 리포지토리, 빌드, 배포 또는 파이프라인에서 발생하는 이벤트를 인식하여 변경 승인 또는 오류 수정과 같은 조치를 신속하게 수행할 수 있도록 합니다. 자세한 정보는 [알림이란 무엇입니까?](#) 섹션을 참조하세요.
- 개발자 도구 콘솔에는 AWS 리소스를 서드 파티 소스 코드 공급자와 연결하는 데 사용할 수 있는 연결 기능이 포함되어 있습니다. 이 기능에는 자체 API인 AWS CodeStar Connections가 있습니다. 연결 기능을 사용하여 서드 파티 공급자와의 인증된 연결을 설정하고 다른 AWS 서비스와 연결 리소스를 사용할 수 있습니다. 자세한 정보는 [연결이란 무엇입니까?](#) 섹션을 참조하세요.

알림이란 무엇입니까?

개발자 도구 콘솔의 알림 기능은 AWS CodeBuild, AWS CodeCommit, AWS CodeDeploy, AWS CodePipeline의 이벤트를 구독하기 위한 알림 관리자입니다. 이 기능에는 자체 API인 AWS CodeStar Notifications가 있습니다. 알림 기능을 사용하여 사용자에게 작업에 가장 중요한 리포지토리, 빌드 프로젝트, 배포 애플리케이션 및 파이프라인의 이벤트를 신속하게 알릴 수 있습니다. 알림 관리자는 사용

자가 리포지토리, 빌드, 배포 또는 파이프라인에서 발생하는 이벤트를 인식하여 변경 승인 또는 오류 수정과 같은 조치를 신속하게 수행할 수 있도록 합니다.

알림으로 할 수 있는 작업은 무엇입니까?

알림 기능을 통해 알림 규칙을 생성 및 관리하여 다음과 같은 리소스의 중요 변경 사항을 사용자에게 알릴 수 있습니다.

- CodeBuild 빌드 프로젝트의 빌드 성공 및 실패
- CodeDeploy 애플리케이션의 배포 성공 및 실패
- CodeCommit 리포지토리의 코드 설명을 포함한 풀 요청 생성 및 업데이트
- CodePipeline의 수동 승인 상태 및 파이프라인 실행

Amazon SNS 주제를 구독한 사용자 이메일 주소로 이동하도록 알림을 설정할 수 있습니다. 또한 이 기능을 [AWS Chatbot](#)과 통합하고 Slack 채널, Microsoft Teams 채널 또는 Amazon Chime 채팅룸에 알림을 전달할 수도 있습니다.

알림은 어떻게 작동합니까?

리포지토리, 빌드 프로젝트, 애플리케이션 또는 파이프라인과 같이 지원되는 리소스에 대한 알림 규칙을 구성할 때 알림 기능은 지정한 이벤트를 모니터링하는 Amazon EventBridge 규칙을 생성합니다. 해당되는 유형의 이벤트가 발생하면 알림 규칙에서 해당 규칙에 대해 대상으로 지정된 Amazon SNS 주제로 알림을 보냅니다. 이 대상의 구독자는 이벤트에 관한 알림을 수신합니다.

알림을 시작하려면 어떻게 해야 합니까?

시작하려면 다음과 같은 유용한 주제를 검토하십시오.

- 알림과 관련한 [개념](#)을 자세히 알아봅니다.
- [필요한 리소스](#)를 설정하여 알림 작업을 시작합니다.
- [첫 번째 알림 규칙](#)을 시작하고 첫 번째 알림을 수신합니다.

알림 관련 개념

개념 및 용어를 이해하면 알림 설정 및 사용이 더 쉬워집니다. 알림을 사용할 때 알아야 할 몇 가지 개념은 다음과 같습니다.

주제

- [알림](#)
- [알림 규칙](#)
- [이벤트](#)
- [세부 정보 유형](#)
- [대상](#)
- [알림 및 AWS CodeStar Notifications](#)
- [리포지토리의 알림 규칙에 대한 이벤트](#)
- [빌드 프로젝트의 알림 규칙에 대한 이벤트](#)
- [배포 애플리케이션의 알림 규칙 이벤트](#)
- [파이프라인의 알림 규칙에 대한 이벤트](#)

알림

알림은 사용자 및 개발자가 사용하는 리소스에서 발생하는 이벤트에 관한 정보가 포함된 메시지입니다. 알림을 설정하면 생성한 알림 규칙에 따라 빌드 프로젝트, 리포지토리, 배포 애플리케이션 또는 파이프라인과 같은 리소스 사용자가 지정한 이벤트 유형에 관한 이메일을 수신합니다.

AWS CodeCommit에 대한 알림에는 세션 태그의 사용을 통해 표시 이름이나 이메일 주소와 같은 사용자 자격 증명 정보가 포함될 수 있습니다. CodeCommit에서는 세션 태그의 사용을 지원합니다. 세션 태그는 IAM 역할을 맡거나, 임시 자격 증명을 사용하거나, AWS Security Token Service(AWS STS)에서 사용자를 연동할 때 전달하는 키-값 페어 속성입니다. IAM 사용자에게 태그를 연결할 수도 있습니다. CodeCommit은 해당 태그가 있는 경우 `displayName` 및 `emailAddress`에 대한 값을 알림 내용에 포함합니다. 자세한 내용은 [태그를 사용하여 CodeCommit에 추가 자격 증명 정보 제공](#)을 참조하세요.

Important

알림에는 빌드 상태, 배포 상태, 의견을 보유한 코드 라인 및 파이프라인 승인과 같은 프로젝트 별 정보가 포함됩니다. 새 기능이 추가되면 알림 내용이 변경될 수 있습니다. 보안 모범 사례로 알림 규칙 및 Amazon SNS 주제 구독자의 대상을 정기적으로 검토해야 합니다. 자세한 내용은 [알림 콘텐츠 및 보안 이해](#) 섹션을 참조하세요.

알림 규칙

알림 규칙은 알림이 발송되는 시기 및 위치를 지정하도록 생성한 AWS 리소스입니다. 이는 다음을 정의합니다.

- 알림이 생성되는 조건. 이러한 조건은 선택한 이벤트를 기반으로 하며, 리소스 유형에 따라 다릅니다. 지원되는 리소스 유형으로는 AWS CodeBuild의 빌드 프로젝트, AWS CodeDeploy의 배포 애플리케이션, AWS CodePipeline의 파이프라인 및 AWS CodeCommit의 리포지토리가 있습니다.
- 알림이 발송되는 대상. 알림 규칙에 대해 최대 10개의 대상을 지정할 수 있습니다.

알림 규칙의 범위는 개별 빌드 프로젝트, 배포 애플리케이션, 파이프라인 및 리포지토리입니다. 알림 규칙에는 사용자가 정의한 친숙한 이름 및 Amazon 리소스 이름(ARN)이 둘 다 있습니다. 알림 규칙은 리소스가 존재하는 AWS 리전에서만 생성되어야 합니다. 예를 들어 미국 동부(오하이오) 리전에서 프로젝트를 빌드하는 경우 알림 규칙도 미국 동부(오하이오) 리전에서 생성해야 합니다.

리소스당 최대 10개의 알림 규칙을 정의할 수 있습니다.

이벤트

이벤트는 모니터링하고자 하는 리소스의 상태 변경입니다. 각 리소스에는 선택할 수 있는 이벤트 유형 목록이 있습니다. 리소스에 알림 규칙을 설정할 때 알림이 전송되도록 하는 이벤트를 지정합니다. 예를 들어 CodeCommit에서 리포지토리에 대한 알림을 설정하고 [풀 요청(Pull request)]과 [브랜치 및 태그(Branches and tags)] 모두에서 [생성됨(Created)]을 선택하면 해당 리포지토리의 사용자가 풀 요청, 브랜치 또는 Git 태그를 생성할 때마다 알림이 전송됩니다.

세부 정보 유형

알림 규칙을 생성할 때 알림에 포함되는 세부 정보 또는 세부 유형 수준을 선택할 수 있습니다(전체 또는 기본). 전체 설정(기본값)에는 특정 이벤트에 대해 서비스에서 제공하는 개선된 정보를 비롯하여 알림의 이벤트에 사용할 수 있는 모든 정보가 포함됩니다. 기본 설정에는 사용 가능한 정보의 하위 집합만 포함됩니다.

다음 표에서는 특정 이벤트 유형에 사용할 수 있는 개선된 정보를 나열하고 세부 유형 간의 차이점을 설명합니다.

서비스	이벤트	전체에 포함되는 항목	기본에 포함되지 않는 항목
CodeCommit	커밋에 대한 의견 풀 요청에 대한 의견	모든 이벤트 세부 정보 및 의견의 내용(답글 또는 의견 스레드 포함). 또한 줄 번호와 의	의견의 내용. 행 번호, 코드 줄 또는 주석 스레드

서비스	이벤트	전체에 포함되는 항목	기본에 포함되지 않는 항목
		견이 만들어진 코드 줄을 포함합니다.	
CodeCommit	풀 요청이 생성됨	대상 분기와 관련하여 풀 요청에서 추가, 수정 또는 삭제된 모든 이벤트 세부 정보 및 파일 수입입니다.	풀 요청 원본 분기가 파일을 추가, 수정 또는 삭제했는지 여부에 대한 파일 목록이나 세부 정보는 없습니다.
CodePipeline	수동 승인 필요	모든 이벤트 세부 정보 및 사용자 지정 데이터 (구성된 경우). 알림에는 파이프라인의 필수 승인에 대한 링크도 포함됩니다.	사용자 지정 데이터 또는 링크가 없습니다.
CodePipeline	작업 실행 실패 파이프라인 실행 실패 스테이지 실행 실패	모든 이벤트 세부 정보 및 실패에 대한 오류 메시지의 내용입니다.	오류 메시지 내용이 없습니다.

대상

대상은 알림 규칙으로부터 알림을 수신하는 위치입니다. 허용되는 대상 유형은 Slack 또는 Microsoft Teams 채널에 대해 구성된 Amazon SNS 주제 및 AWS Chatbot 클라이언트입니다. 대상을 구독하는 모든 사용자는 알림 규칙에서 지정한 이벤트에 관해 알림을 수신합니다.

알림 범위를 확장하고자 하는 경우 알림이 Amazon Chime 채팅룸으로 발송되도록 알림과 AWS Chatbot 간의 통합을 수동으로 구성할 수 있습니다. 그런 다음 해당 AWS Chatbot 클라이언트에 대해 구성된 Amazon SNS 주제를 알림 규칙의 대상으로 선택할 수 있습니다. 자세한 내용은 [AWS Chatbot 및 Amazon Chime과 알림을 통합하려면](#) 섹션을 참조하세요.

AWS Chatbot 클라이언트를 대상으로 사용하는 경우 먼저 AWS Chatbot에서 해당 클라이언트를 생성해야 합니다. AWS Chatbot 클라이언트를 알림 규칙의 대상으로 선택하면 Slack 또는 Microsoft Teams

채널로 알림을 보내는 데 필요한 모든 정책이 포함된 Amazon SNS 주제가 해당 AWS Chatbot 클라이언트에 대해 구성됩니다. AWS Chatbot 클라이언트에 대한 기존 Amazon SNS 주제를 구성할 필요가 없습니다.

알림 규칙 생성 중 Amazon SNS 주제를 대상으로 생성하도록 선택할 수 있습니다(권장). 알림 규칙과 동일한 AWS 리전의 기존 Amazon SNS 주제를 선택할 수도 있지만 필요한 정책으로 구성해야 합니다. 대상에 사용하는 Amazon SNS 주제는 AWS 계정에 있어야 합니다. 또한 알림 규칙 및 규칙이 생성된 AWS 리소스와 동일한 AWS 리전에 있어야 합니다.

예를 들어 미국 동부(오하이오) 리전에서 리포지토리에 대한 알림 규칙을 생성한 경우 Amazon SNS 주제 또한 해당 리전에 존재해야 합니다. 알림 규칙을 만드는 과정에서 Amazon SNS 주제를 생성하는 경우 주제는 주제에 이벤트를 게시할 수 있도록 허용하는 데 필요한 정책으로 구성됩니다. 이는 대상 및 알림 규칙 작업에 가장 적합한 방법입니다. 이미 존재하는 주제를 사용하거나 수동으로 만들도록 선택한 경우 필요한 권한으로 주제를 구성해야 사용자가 알림을 받을 수 있습니다. 자세한 내용은 [알림에 대한 Amazon SNS 주제 구성](#) 섹션을 참조하세요.

Note

새 항목을 생성하는 대신 기존 Amazon SNS 항목을 사용하려면 [대상(Targets)]에서 해당 ARN을 선택합니다. 주제에 적절한 액세스 정책이 있는지 여부와 구독자 목록에 리소스에 대한 정보를 볼 수 있는 사용자만 포함되어 있는지를 확인합니다. Amazon SNS 주제가 2019년 11월 5일 이전에 CodeCommit 알림에 사용된 주제인 경우 CodeCommit이 AWS CodeStar Notifications에 필요한 권한과 다른 권한을 포함하는 주제를 게시하도록 허용하는 정책이 포함됩니다. 이러한 주제는 사용하지 않는 것이 좋습니다. 해당 환경에 대해 생성된 정책을 사용하려면 이미 존재하는 정책 외에 AWS CodeStar Notifications에 대한 필수 정책을 추가해야 합니다. 자세한 정보는 [알림에 대한 Amazon SNS 주제 구성](#) 및 [알림 콘텐츠 및 보안 이해](#) 섹션을 참조하세요.

알림 및 AWS CodeStar Notifications

개발자 도구 콘솔의 기능인 알림에는 자체적인 API인 AWS CodeStar Notifications가 있습니다. 또한 자체 AWS 리소스 유형(알림 규칙), 권한 및 이벤트도 있습니다. 알림 규칙에 대한 이벤트는 AWS CloudTrail에 기록됩니다. API 작업은 IAM 정책을 통해 허용되거나 거부될 수 있습니다.

리포지토리의 알림 규칙에 대한 이벤트

범주	이벤트	이벤트 ID
설명	커밋 시 풀 요청 시	codecommit-repository- comments-on-commits codecommit-repository- comments-on-pull-reques ts
승인	상태 변경 규칙 재정의	codecommit-repository- approvals-status-change d codecommit-repository- approvals-rule-override
풀 요청	생성 완료 소스 업데이트 상태 변경 병합	codecommit-repository- pull-request-created codecommit-repository- pull-request-source-upd ated codecommit-repository- pull-request-status-cha nged codecommit-repository- pull-request-merged
브랜치 및 태그	생성 완료 Deleted Updated	codecommit-repository- branches-and-tags-creat ed codecommit-repository- branches-and-tags-delet ed

범주	이벤트	이벤트 ID
		codecommit-repository-branches-and-tags-updated

빌드 프로젝트의 알림 규칙에 대한 이벤트

범주	이벤트	이벤트 ID
빌드 상태	실패함	codebuild-project-build-state-failed
	성공	
	진행 중	codebuild-project-build-state-succeeded
	중지됨	codebuild-project-build-state-in-progress
빌드 단계	결함	codebuild-project-build-phase-failure
	성공	codebuild-project-build-phase-success

배포 애플리케이션의 알림 규칙 이벤트

범주	이벤트	이벤트 ID
배포	실패함	codedeploy-application-deployment-failed
	성공	
	시작됨	codedeploy-application-deployment-succeeded

범주	이벤트	이벤트 ID
		codedeploy-application-deployment-started

파이프라인의 알림 규칙에 대한 이벤트

범주	이벤트	이벤트 ID
작업 실행	성공	codepipeline-pipeline-action-execution-succeeded
	실패함	
	취소됨	codepipeline-pipeline-action-execution-failed
	시작됨	codepipeline-pipeline-action-execution-canceled
단계 실행	시작됨	codepipeline-pipeline-stage-execution-started
	성공	
	재개	codepipeline-pipeline-stage-execution-succeeded
	취소됨	codepipeline-pipeline-stage-execution-resumed
	실패함	codepipeline-pipeline-stage-execution-canceled
파이프라인 실행	실패함	codepipeline-pipeline-pipeline-execution-failed
	취소됨	

범주	이벤트	이벤트 ID
	시작됨	codepipeline-pipeline-pipeline-execution-canceled
	재개	
	성공	codepipeline-pipeline-pipeline-execution-started
	대체됨	codepipeline-pipeline-pipeline-execution-resumed
		codepipeline-pipeline-pipeline-execution-succeeded
		codepipeline-pipeline-pipeline-execution-superseded
수동 승인	실패함	codepipeline-pipeline-manual-approval-failed
	필요	
	성공	codepipeline-pipeline-manual-approval-needed codepipeline-pipeline-manual-approval-succeeded

설정

IAM 사용자 또는 역할에 AWS CodeBuild, AWS CodeCommit, AWS CodeDeploy 또는 AWS CodePipeline에 대한 관리형 정책이 적용된 경우 정책에서 제공하는 역할 및 권한의 제한 내에서 알림 작업에 필요한 권한을 갖게 됩니다. 예를 들어, `AWSCodeBuildAdminAccess`, `AWSCodeCommitFullAccess`, `AWSCodeDeployFullAccess` 또는 `AWSCodePipeline_FullAccess` 관리형 정책이 적용된 사용자는 알림에 대한 전체 관리 액세스 권한을 가집니다.

정책 예제를 포함한 자세한 내용은 [ID 기반 정책](#)을 참조하십시오.

이러한 정책 중 하나가 IAM 사용자 또는 역할에 적용되고 CodeBuild의 빌드 프로젝트, CodeCommit의 리포지토리, CodeDeploy의 배포 애플리케이션 또는 CodePipeline의 파이프라인이 있으면 첫 번째 알

림 규칙을 생성할 수 있습니다. 계속해서 [알림 시작하기](#)로 이동하십시오. 그렇지 않은 경우 다음 주제를 참조하십시오.

- CodeBuild: [CodeBuild 시작하기](#)
- CodeCommit: [CodeCommit 시작하기](#)
- CodeDeploy: [자습서](#)
- CodePipeline: [CodePipeline 시작하기](#)

IAM 사용자, 그룹 또는 역할의 알림에 대한 관리 권한을 직접 관리하려면 이 주제의 절차에 따라 서비스를 사용하는 데 필요한 권한 및 리소스를 설정합니다.

알림 전용 주제를 생성하는 대신 이전에 생성된 Amazon SNS 주제를 알림에 사용하려면 이벤트를 해당 주제에 게시할 수 있도록 허용하는 정책을 적용하여 Amazon SNS 주제를 알림 규칙의 대상으로 사용하도록 구성해야 합니다.

Note

다음 절차를 수행하려면 관리자 권한을 보유한 계정으로 로그인해야 합니다. 자세한 내용은 [첫 번째 IAM 관리자 및 그룹 생성](#)을 참조하세요.

주제

- [알림에 대한 관리자 액세스를 위한 정책 생성 및 적용](#)
- [알림에 대한 Amazon SNS 주제 구성](#)
- [대상인 Amazon SNS 주제에 대한 사용자 구독](#)

알림에 대한 관리자 액세스를 위한 정책 생성 및 적용

IAM 사용자로 로그인하거나 알림을 생성하려는 서비스(AWS CodeBuild, AWS CodeCommit, AWS CodeDeploy 또는 AWS CodePipeline)에 대한 액세스 권한이 있는 역할을 사용하여 알림을 관리할 수 있습니다. 자체 정책을 생성하여 사용자 또는 그룹에 적용할 수도 있습니다.

다음 절차에서는 알림을 관리하고 IAM 사용자를 추가할 권한이 있는 IAM 그룹을 구성하는 방법을 보여줍니다. 그룹을 설정하지 않고자 하는 경우 이 정책을 직접 IAM 사용자 또는 사용자가 수입할 수 있는 IAM 역할에 적용할 수 있습니다. 또한 정책 범위에 따라 알림 기능에 대한 적절한 정책 액세스를 포함하는 CodeBuild, CodeCommit, CodeDeploy 또는 CodePipeline에 대한 관리형 정책을 사용할 수 있습니다.

아래 정책에서 이 정책에 대한 이름(예: `AWSCodeStarNotificationsFullAccess`)과 설명(선택 사항)을 입력합니다. 설명은 정책의 목적을 기억하는 데 도움이 됩니다(예: **This policy provides full access to AWS CodeStar Notifications.**).

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 정책(Policies)을 선택합니다.

정책(Policies)을 처음으로 선택하는 경우 관리형 정책 소개 페이지가 나타납니다. 시작하기(Get Started)를 선택합니다.

3. 페이지 상단에서 정책 생성을 선택합니다.
4. 정책 편집기 섹션에서 JSON 옵션을 선택합니다.
5. 다음 JSON 정책 문서를 입력합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSCodeStarNotificationsFullAccess",
      "Effect": "Allow",
      "Action": [
        "codestar-notifications:CreateNotificationRule",
        "codestar-notifications>DeleteNotificationRule",
        "codestar-notifications:DescribeNotificationRule",
        "codestar-notifications:ListNotificationRules",
        "codestar-notifications:UpdateNotificationRule",
        "codestar-notifications:Subscribe",
        "codestar-notifications:Unsubscribe",
        "codestar-notifications>DeleteTarget",
        "codestar-notifications:ListTargets",
        "codestar-notifications:ListTagsForResource",
        "codestar-notifications:TagResource",
        "codestar-notifications:UntagResource"
      ],
      "Resource": "*"
    }
  ]
}
```

6. 다음(Next)을 선택합니다.

Note

언제든지 시각적 편집기 옵션과 JSON 편집기 옵션을 서로 전환할 수 있습니다. 그러나 변경을 적용하거나 시각적 편집기에서 다음을 선택한 경우 IAM은 시각적 편집기에 최적화 되도록 정책을 재구성할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [정책 재구성](#)을 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 생성(Create policy)을 선택하고 새로운 정책을 저장합니다.

알림에 대한 Amazon SNS 주제 구성

알림을 설정하는 가장 쉬운 방법은 알림 규칙을 생성할 때 Amazon SNS 주제를 생성하는 것입니다. 다음 요구 사항을 충족하는 경우 기존 Amazon SNS 주제를 사용할 수 있습니다.

- 알림 규칙을 생성하고자 하는 리소스(빌드 프로젝트, 배포 애플리케이션, 리포지토리 또는 파이프라인)와 동일한 AWS 리전에서 생성된 경우
- 2019년 11월 5일 이전에 CodeCommit에 대한 알림을 보내는 데 사용되지 않은 경우 이 경우 해당 기능을 사용하도록 설정한 정책 설명이 포함됩니다. 이 주제를 사용하도록 선택할 수 있지만 절차에 지정된 대로 추가 정책을 추가해야 합니다. 2019년 11월 5일 이전에 알림에 대해 하나 이상의 리포지토리가 구성된 경우 기존 정책 설명을 제거하면 안 됩니다.
- AWS CodeStar Notifications가 알림을 주제에 게시할 수 있도록 허용하는 정책이 있는 경우

AWS CodeStar Notifications 알림 규칙의 대상으로 Amazon SNS 주제를 구성하는 방법

1. AWS Management Console에 로그인하고 <https://console.aws.amazon.com/sns/v3/home>에서 Amazon SNS 콘솔을 엽니다.
2. 탐색 모음에서 주제를 선택하고 구성할 주제를 선택한 다음 편집을 선택합니다.
3. Access policy(액세스 정책)를 확장한 다음 고급을 선택합니다.
4. JSON 편집기에서 다음 설명을 정책에 추가합니다. 주제 ARN, AWS 리전, AWS 계정 ID 및 주제 이름을 포함합니다.

```
{
```

```

    "Sid": "AWSCodeStarNotifications_publish",
    "Effect": "Allow",
    "Principal": {
      "Service": [
        "codestar-notifications.amazonaws.com"
      ]
    },
    "Action": "SNS:Publish",
    "Resource": "arn:aws:sns:us-east-2:123456789012:codestar-notifications-MyTopicForNotificationRules"
  }

```

정책 설명은 다음과 같습니다.

```

{
  "Version": "2008-10-17",
  "Id": "__default_policy_ID",
  "Statement": [
    {
      "Sid": "__default_statement_ID",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "SNS:GetTopicAttributes",
        "SNS:SetTopicAttributes",
        "SNS:AddPermission",
        "SNS:RemovePermission",
        "SNS:DeleteTopic",
        "SNS:Subscribe",
        "SNS:ListSubscriptionsByTopic",
        "SNS:Publish",
        "SNS:Receive"
      ],
      "Resource": "arn:aws:sns:us-east-2:123456789012:codestar-notifications-MyTopicForNotificationRules",
      "Condition": {
        "StringEquals": {
          "AWS:SourceOwner": "123456789012"
        }
      }
    }
  ],
}

```

```
{
  "Sid": "AWSCodeStarNotifications_publish",
  "Effect": "Allow",
  "Principal": {
    "Service": [
      "codestar-notifications.amazonaws.com"
    ]
  },
  "Action": "SNS:Publish",
  "Resource": "arn:aws:sns:us-east-2:123456789012:codestar-notifications-MyTopicForNotificationRules"
}
```

5. Save changes(변경 사항 저장)를 선택합니다.
6. AWS KMS로 암호화된 Amazon SNS 주제를 사용하여 알림을 전송하려면 AWS KMS key 정책에 다음 명령문을 추가하여 이벤트 소스(AWS CodeStar Notifications)와 암호화된 주제 사이의 호환성도 사용 설정해야 합니다. AWS 리전(이 예에서는 us-east-2)을 키가 생성된 AWS 리전으로 바꿉니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "codestar-notifications.amazonaws.com"
      },
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "sns.us-east-2.amazonaws.com"
        }
      }
    }
  ]
}
```

자세한 내용은 AWS Key Management Service 개발자가이드에서 [저장 시 암호화](#) 및 [AWS KMS에 정책 조건 사용](#) 을 참조하세요.

대상인 Amazon SNS 주제에 대한 사용자 구독

사용자가 알림을 수신하려면 먼저 알림 규칙의 대상이 되는 Amazon SNS 주제를 구독해야 합니다. 사용자가 이메일 주소로 구독하는 경우 알림을 받기 전에 구독을 확인해야 합니다. Slack 채널, Microsoft Teams 채널 또는 Amazon Chime 채팅룸에서 사용자에게 알림을 보내려면 [알림과 AWS Chatbot 간의 통합 구성](#) 섹션을 참조하세요.

알림에 사용되는 Amazon SNS 주제에 대해 사용자를 구독하려면

1. AWS Management Console에 로그인하고 <https://console.aws.amazon.com/sns/v3/home>에서 Amazon SNS 콘솔을 엽니다.
2. 탐색 모음에서 주제를 선택하고 사용자를 구독하려는 주제를 선택합니다.
3. 구독에서 구독 생성을 선택합니다.
4. 프로토콜에서 이메일을 선택합니다. 엔드포인트에 이메일 주소를 입력한 다음 구독 생성을 선택합니다.

알림 시작하기

알림을 시작하는 가장 쉬운 방법은 빌드 프로젝트, 배포 애플리케이션, 파이프라인 또는 리포지토리 중 하나에 알림 규칙을 설정하는 것입니다.

Note

처음 알림 규칙을 생성하면 해당 계정에 서비스 연결 역할이 생성됩니다. 자세한 내용은 [AWS CodeStar Notifications에 서비스 연결 역할 사용](#) 섹션을 참조하세요.

주제

- [사전 조건](#)
- [리포지토리에 대한 알림 규칙 생성](#)
- [빌드 프로젝트에 대한 알림 규칙 생성](#)
- [배포 애플리케이션에 대한 알림 규칙 생성](#)

- [파이프라인에 대한 알림 규칙 생성](#)

사전 조건

[설정](#)의 단계를 수행합니다. 또한 알림 규칙을 만들 리소스가 필요합니다.

- [CodeBuild에서 빌드 프로젝트를 생성](#)하거나 기존 프로젝트를 사용합니다.
- [애플리케이션을 생성](#)하거나 기존 배포 애플리케이션을 사용합니다.
- [CodePipeline에서 파이프라인을 생성](#)하거나 기존 파이프라인을 사용합니다.
- [AWS CodeCommit 리포지토리를 생성](#)하거나 기존 리포지토리를 사용합니다.

리포지토리에 대한 알림 규칙 생성

알림 규칙을 생성하여 중요한 리포지토리의 이벤트에 관한 알림을 보낼 수 있습니다. 다음 단계는 단일 리포지토리 이벤트에 대한 알림 규칙을 설정하는 방법을 보여줍니다. 이러한 단계는 AWS 계정에 리포지토리가 구성되어 있다는 가정하에 작성되었습니다.

Important

2019년 11월 5일 이전에 CodeCommit에서 알림을 설정한 경우 해당 알림에 사용되는 Amazon SNS 주제에는 CodeCommit이 AWS CodeStar Notifications에 필요한 권한과 다른 권한을 포함하는 주제에 게시할 수 있는 정책이 포함됩니다. 이러한 주제는 사용하지 않는 것이 좋습니다. 해당 환경에 대해 생성된 정책을 사용하려면 이미 존재하는 정책 외에 AWS CodeStar Notifications에 대한 필수 정책을 추가해야 합니다. 자세한 정보는 [알림에 대한 Amazon SNS 주제 구성 및 알림 콘텐츠 및 보안 이해](#) 섹션을 참조하세요.

1. <https://console.aws.amazon.com/codecommit/>에서 CodeCommit 콘솔을 엽니다.
2. 목록에서 리포지토리를 선택하고 엽니다.
3. Notify(알림)를 선택하고 Create notification rule(알림 규칙 생성)을 선택합니다. 설정을 선택하고, 알림을 선택한 다음 Create notification rule(알림 규칙 생성)을 선택합니다.
4. 알림 이름에 규칙에 대한 이름을 입력합니다.
5. 알림에 포함된 Amazon EventBridge에 제공된 정보만 원하는 경우 [세부 정보 유형(Detail type)]에서 [기본(Basic)]을 선택합니다. Amazon EventBridge에 제공된 정보와 리소스 서비스 또는 알림 관리자가 제공할 수 있는 정보를 포함하려는 경우 [전체(Full)]를 선택합니다.

자세한 내용은 [알림 콘텐츠 및 보안 이해](#) 섹션을 참조하세요.

6. Events that trigger notifications(알림을 트리거하는 이벤트)의 Branches and tags(브랜치 및 태그)에서 생성됨을 선택합니다.
7. 대상에서 SNS 주제 생성을 선택합니다.

Note

알림 규칙을 만드는 과정에서 주제를 생성하면 CodeCommit이 주제에 이벤트를 게시하도록 허용하는 정책이 적용됩니다. 알림 규칙에 대해 생성된 주제를 사용하면 이 리포지토리에 대한 알림을 받기를 원하는 사용자만 구독할 수 있습니다.

codestar-notifications- 접두사 뒤에 주제 이름을 입력한 다음 제출을 선택합니다.

Note

새 항목을 생성하는 대신 기존 Amazon SNS 항목을 사용하려면 [대상(Targets)]에서 해당 ARN을 선택합니다. 주제에 적절한 액세스 정책이 있는지 여부와 구독자 목록에 리소스에 대한 정보를 볼 수 있는 사용자만 포함되어 있는지를 확인합니다. Amazon SNS 주제가 2019년 11월 5일 이전에 CodeCommit 알림에 사용된 주제인 경우 CodeCommit이 AWS CodeStar Notifications에 필요한 권한과 다른 권한을 포함하는 주제를 게시하도록 허용하는 정책이 포함됩니다. 이러한 주제는 사용하지 않는 것이 좋습니다. 해당 환경에 대해 생성된 정책을 사용하려면 이미 존재하는 정책 외에 AWS CodeStar Notifications에 대한 필수 정책을 추가해야 합니다. 자세한 정보는 [알림에 대한 Amazon SNS 주제 구성 및 알림 콘텐츠 및 보안 이해](#) 섹션을 참조하세요.

8. 제출을 선택한 다음 알림 규칙을 검토합니다.
9. 방금 생성한 Amazon SNS 주제에 대한 이메일 주소를 구독합니다. 자세한 내용은 [알림에 사용되는 Amazon SNS 주제에 대해 사용자를 구독하려면](#) 섹션을 참조하세요.
10. 리포지토리로 이동하여 기본 브랜치에서 테스트 브랜치를 생성합니다.
11. 브랜치를 생성하면 알림 규칙이 모든 주제 가입자에게 해당 이벤트에 대한 정보와 함께 알림을 보냅니다.

빌드 프로젝트에 대한 알림 규칙 생성

알림 규칙을 생성하여 중요한 빌드 프로젝트의 이벤트에 관한 알림을 보낼 수 있습니다. 다음 단계는 단일 빌드 프로젝트 이벤트에 대한 알림 규칙을 설정하는 방법을 보여줍니다. 이러한 단계는 AWS 계정에 빌드 프로젝트가 구성되어 있다는 가정하에 작성되었습니다.

1. <https://console.aws.amazon.com/codebuild/>에서 CodeBuild 콘솔을 엽니다.
2. 목록에서 빌드 프로젝트를 선택하고 엽니다.
3. Notify(알림)를 선택하고 Create notification rule(알림 규칙 생성)을 선택합니다. 설정을 선택한 다음 Create notification rule(알림 규칙 생성)을 선택할 수도 있습니다.
4. 알림 이름에 규칙에 대한 이름을 입력합니다.
5. 알림에 포함된 Amazon EventBridge에 제공된 정보만 원하는 경우 [세부 정보 유형(Detail type)]에서 [기본(Basic)]을 선택합니다. Amazon EventBridge에 제공된 정보와 리소스 서비스 또는 알림 관리자가 제공할 수 있는 정보를 포함하려는 경우 [전체(Full)]를 선택합니다.

자세한 내용은 [알림 콘텐츠 및 보안 이해](#) 섹션을 참조하세요.

6. Events that trigger notifications(알림을 트리거하는 이벤트)의 Build phase(빌드 단계)에서 성공을 선택합니다.
7. 대상에서 SNS 주제 생성을 선택합니다.

Note

알림 규칙을 만드는 과정에서 주제를 생성하면 CodeBuild가 주제에 이벤트를 게시하도록 허용하는 정책이 적용됩니다. 알림 규칙에 대해 생성된 주제를 사용하면 이 빌드 프로젝트에 대한 알림을 받기를 원하는 사용자만 구독할 수 있습니다.

codestar-notifications- 접두사 뒤에 주제 이름을 입력한 다음 제출을 선택합니다.

Note

새 항목을 생성하는 대신 기존 Amazon SNS 항목을 사용하려면 [대상(Targets)]에서 해당 ARN을 선택합니다. 주제에 적절한 액세스 정책이 있는지 여부와 구독자 목록에 리소스에 대한 정보를 볼 수 있는 사용자만 포함되어 있는지를 확인합니다. Amazon SNS 주제가 2019년 11월 5일 이전에 CodeCommit 알림에 사용된 주제인 경우 CodeCommit이 AWS CodeStar Notifications에 필요한 권한과 다른 권한을 포함하는 주제를 게시하도록 허용하

는 정책이 포함됩니다. 이러한 주제는 사용하지 않는 것이 좋습니다. 해당 환경에 대해 생성된 정책을 사용하려면 이미 존재하는 정책 외에 AWS CodeStar Notifications에 대한 필수 정책을 추가해야 합니다. 자세한 정보는 [알림에 대한 Amazon SNS 주제 구성 및 알림 콘텐츠 및 보안 이해](#) 섹션을 참조하세요.

8. 제출을 선택한 다음 알림 규칙을 검토합니다.
9. 방금 생성한 Amazon SNS 주제에 대한 이메일 주소를 구독합니다. 자세한 내용은 [알림에 사용되는 Amazon SNS 주제에 대해 사용자를 구독하려면](#) 섹션을 참조하세요.
10. 빌드 프로젝트로 이동한 다음 빌드를 시작합니다.
11. 빌드 단계가 성공적으로 완료되면 알림 규칙이 모든 주제 가입자에게 해당 이벤트에 대한 정보와 함께 알림을 보냅니다.

배포 애플리케이션에 대한 알림 규칙 생성

알림 규칙을 생성하여 중요한 배포 애플리케이션의 이벤트에 관한 알림을 보낼 수 있습니다. 다음 단계는 단일 빌드 프로젝트 이벤트에 대한 알림 규칙을 설정하는 방법을 보여줍니다. 이러한 단계는 AWS 계정에 배포 애플리케이션이 구성되어 있다는 가정하에 작성되었습니다.

1. <https://console.aws.amazon.com/codedeploy/>에서 CodeDeploy 콘솔을 엽니다.
2. 목록에서 애플리케이션을 선택하고 엽니다.
3. Notify(알림)를 선택하고 Create notification rule(알림 규칙 생성)을 선택합니다. 설정을 선택한 다음 Create notification rule(알림 규칙 생성)을 선택할 수도 있습니다.
4. 알림 이름에 규칙에 대한 이름을 입력합니다.
5. 알림에 포함된 Amazon EventBridge에 제공된 정보만 원하는 경우 [세부 정보 유형(Detail type)]에서 [기본(Basic)]을 선택합니다. Amazon EventBridge에 제공된 정보와 리소스 서비스 또는 알림 관리자가 제공할 수 있는 정보를 포함하려는 경우 [전체(Full)]를 선택합니다.

자세한 내용은 [알림 콘텐츠 및 보안 이해](#) 섹션을 참조하세요.

6. Events that trigger notifications(알림을 트리거하는 이벤트)의 배포에서 성공을 선택합니다.
7. 대상에서 SNS 주제를 생성을 선택합니다.

Note

알림 규칙을 만드는 과정에서 주제를 생성하면 CodeDeploy에서 주제에 이벤트를 게시하도록 허용하는 정책이 적용됩니다. 알림 규칙에 대해 생성된 주제를 사용하면 이 배포 애플리케이션에 대한 알림을 받기를 원하는 사용자만 구독할 수 있습니다.

codestar-notifications- 접두사 뒤에 주제 이름을 입력한 다음 제출을 선택합니다.

Note

새 항목을 생성하는 대신 기존 Amazon SNS 항목을 사용하려면 [대상(Targets)]에서 해당 ARN을 선택합니다. 주제에 적절한 액세스 정책이 있는지 여부와 구독자 목록에 리소스에 대한 정보를 볼 수 있는 사용자만 포함되어 있는지를 확인합니다. Amazon SNS 주제가 2019년 11월 5일 이전에 CodeCommit 알림에 사용된 주제인 경우 CodeCommit이 AWS CodeStar Notifications에 필요한 권한과 다른 권한을 포함하는 주제를 게시하도록 허용하는 정책이 포함됩니다. 이러한 주제는 사용하지 않는 것이 좋습니다. 해당 환경에 대해 생성된 정책을 사용하려면 이미 존재하는 정책 외에 AWS CodeStar Notifications에 대한 필수 정책을 추가해야 합니다. 자세한 정보는 [알림에 대한 Amazon SNS 주제 구성 및 알림 콘텐츠 및 보안 이해](#) 섹션을 참조하세요.

8. 제출을 선택한 다음 알림 규칙을 검토합니다.
9. 방금 생성한 Amazon SNS 주제에 대한 이메일 주소를 구독합니다. 자세한 내용은 [알림에 사용되는 Amazon SNS 주제에 대해 사용자를 구독하려면](#) 섹션을 참조하세요.
10. 배포 애플리케이션으로 이동하여 배포를 시작합니다.
11. 배포가 성공하면 알림 규칙이 모든 주제 구독자에게 이벤트에 대한 정보와 함께 알림을 보냅니다.

파이프라인에 대한 알림 규칙 생성

알림 규칙을 생성하여 중요한 파이프라인의 이벤트에 관한 알림을 보낼 수 있습니다. 다음 단계는 단일 파이프라인 이벤트에 대한 알림 규칙을 설정하는 방법을 보여줍니다. 이러한 단계는 AWS 계정에 파이프라인이 구성되어 있다는 가정하에 작성되었습니다.

1. <https://console.aws.amazon.com/codepipeline/>에서 CodePipeline 콘솔을 엽니다.
2. 목록에서 파이프라인을 선택하고 엽니다.

3. Notify(알림)를 선택하고 Create notification rule(알림 규칙 생성)을 선택합니다. 설정을 선택한 다음 Create notification rule(알림 규칙 생성)을 선택할 수도 있습니다.
4. 알림 이름에 규칙에 대한 이름을 입력합니다.
5. 알림에 포함된 Amazon EventBridge에 제공된 정보만 원하는 경우 [세부 정보 유형(Detail type)]에서 [기본(Basic)]을 선택합니다. Amazon EventBridge에 제공된 정보와 리소스 서비스 또는 알림 관리자가 제공할 수 있는 정보를 포함하려는 경우 [전체(Full)]를 선택합니다.

자세한 내용은 [알림 콘텐츠 및 보안 이해](#) 섹션을 참조하세요.

6. Events that trigger notifications(알림을 트리거하는 이벤트)의 Action execution(작업 실행)에서 시작됨을 선택합니다.
7. 대상에서 SNS 주제 생성을 선택합니다.

Note

알림 규칙을 만드는 과정에서 주제를 생성하면 CodePipeline이 주제에 이벤트를 게시하도록 허용하는 정책이 적용됩니다. 알림 규칙에 대해 생성된 주제를 사용하면 이 파이프라인에 대한 알림을 받기를 원하는 사용자만 구독할 수 있습니다.

codestar-notifications- 접두사 뒤에 주제 이름을 입력한 다음 제출을 선택합니다.

Note

새 항목을 생성하는 대신 기존 Amazon SNS 항목을 사용하려면 [대상(Targets)]에서 해당 ARN을 선택합니다. 주제에 적절한 액세스 정책이 있는지 여부와 구독자 목록에 리소스에 대한 정보를 볼 수 있는 사용자만 포함되어 있는지를 확인합니다. Amazon SNS 주제가 2019년 11월 5일 이전에 CodeCommit 알림에 사용된 주제인 경우 CodeCommit이 AWS CodeStar Notifications에 필요한 권한과 다른 권한을 포함하는 주제를 게시하도록 허용하는 정책이 포함됩니다. 이러한 주제는 사용하지 않는 것이 좋습니다. 해당 환경에 대해 생성된 정책을 사용하려면 이미 존재하는 정책 외에 AWS CodeStar Notifications에 대한 필수 정책을 추가해야 합니다. 자세한 정보는 [알림에 대한 Amazon SNS 주제 구성 및 알림 콘텐츠 및 보안 이해](#) 섹션을 참조하세요.

8. 제출을 선택한 다음 알림 규칙을 검토합니다.
9. 방금 생성한 Amazon SNS 주제에 대한 이메일 주소를 구독합니다. 자세한 내용은 [알림에 사용되는 Amazon SNS 주제에 대해 사용자를 구독하려면](#) 섹션을 참조하세요.

10. 파이프라인으로 이동한 다음 변경 사항 배포를 선택합니다.
11. 작업이 시작되면 알림 규칙이 모든 주제 구독자에게 이벤트에 대한 정보와 함께 알림을 보냅니다.

알림 규칙 사용

알림 규칙에서는 사용자가 알림을 받을 이벤트를 구성하고 해당 알림을 받을 대상을 지정합니다. Amazon SNS를 통해 직접 또는 Slack 또는 Microsoft Teams 채널에 대해 구성된 AWS Chatbot 클라이언트를 통해 사용자에게 알림을 보낼 수 있습니다. 알림 범위를 확장하고자 하는 경우 알림이 Amazon Chime 채팅룸으로 발송되도록 알림과 AWS Chatbot 간의 통합을 수동으로 구성할 수 있습니다. 자세한 정보는 [대상](#) 및 [AWS Chatbot 및 Amazon Chime과 알림을 통합하려면](#) 섹션을 참조하세요.

Create notification rule

Notification rules set up a subscription to events that happen with your resources. When these events occur, you will receive notifications sent to the targets you designate. You can manage your notification preferences in Settings. [Info](#)

Notification rule settings

Notification name

Detail type

Choose the level of detail you want in notifications. [Learn more about notifications and security](#)

Full
Includes any supplemental information about events provided by the resource or the notifications feature.

Basic
Includes only information provided in resource events.

Events that trigger notifications

Comments

On commits
 On pull requests

Approvals

Status changed
 Rule override

Pull request

Source updated
 Created
 Status changed
 Merged

Branches and tags

Created
 Deleted
 Updated

Targets

Choose a target type for the notification rule. SNS topics can be created specifically for use with the notification rule, or existing topics can be modified for use with notifications. AWS Chatbot clients for Slack integration must be created before you can choose them as a target type. [Learn more](#)

개발자 도구 콘솔을 사용하거나 AWS CLI를 사용하여 알림 규칙을 생성 및 관리할 수 있습니다.

주제

- [알림 규칙 생성](#)

- [알림 규칙 보기](#)
- [알림 규칙 편집](#)
- [알림 규칙에 대한 알림 사용 또는 사용 중지](#)
- [알림 규칙 삭제](#)

알림 규칙 생성

개발자 도구 콘솔을 사용하거나 AWS CLI를 사용하여 알림 규칙을 생성할 수 있습니다. 규칙을 생성하는 중 알림 규칙의 대상으로 사용할 Amazon SNS 주제를 생성할 수 있습니다. AWS Chatbot 클라이언트를 대상으로 사용하려면 규칙을 만들기 전에 해당 클라이언트를 만들어야 합니다. 자세한 내용은 [Slack 채널에 대한 AWS Chatbot 클라이언트 구성](#) 섹션을 참조하세요.

알림 규칙을 생성하려면(콘솔)

1. <https://console.aws.amazon.com/codesuite/settings/notifications>에서 AWS 개발자 도구 콘솔을 엽니다.
2. 탐색 모음을 사용하여 리소스로 이동합니다.
 - CodeBuild의 경우 [빌드(Build)]를 선택하고, [빌드 프로젝트(Build projects)]를 선택한 다음 빌드 프로젝트를 선택합니다.
 - CodeCommit의 경우 [소스(Source)]를 선택하고, [리포지토리(Repositories)]를 선택한 다음 리포지토리를 선택합니다.
 - CodeDeploy의 경우 [애플리케이션(Applications)]을 선택하고 애플리케이션을 선택합니다.
 - CodePipeline의 경우 [파이프라인(Pipeline)]을 선택하고, [파이프라인(Pipelines)]을 선택한 다음 파이프라인을 선택합니다.
3. 리소스 페이지에서 Notify(알림)를 선택하고 Create notification rule(알림 규칙 생성)을 선택합니다. 리소스에 대한 설정 페이지로 이동하고, 알림 또는 Notification rules(알림 규칙)로 이동한 다음 Create notification rule(알림 규칙 생성)을 선택할 수도 있습니다.
4. 알림 이름에 규칙에 대한 이름을 입력합니다.
5. 알림에 포함된 Amazon EventBridge에 제공된 정보만 원하는 경우 [세부 정보 유형(Detail type)]에서 [기본(Basic)]을 선택합니다. Amazon EventBridge에 제공된 정보와 리소스 서비스 또는 알림 관리자가 제공할 수 있는 정보를 포함하려는 경우 [전체(Full)]를 선택합니다.

자세한 내용은 [알림 콘텐츠 및 보안 이해](#) 섹션을 참조하세요.
6. Events that trigger notifications(알림을 트리거하는 이벤트)에서 알림을 보내고자 하는 이벤트를 선택합니다. 리소스에 대한 이벤트 유형은 다음을 참조하십시오.

- CodeBuild: [빌드 프로젝트의 알림 규칙에 대한 이벤트](#)
- CodeCommit: [리포지토리의 알림 규칙에 대한 이벤트](#)
- CodeDeploy: [배포 애플리케이션의 알림 규칙 이벤트](#)
- CodePipeline: [파이프라인의 알림 규칙에 대한 이벤트](#)

7. 대상에서 다음 중 하나를 수행합니다.

- 알림과 함께 사용할 리소스를 이미 구성한 경우 대상 유형 선택에서 AWS Chatbot(Slack), AWS Chatbot(Microsoft Teams) 또는 SNS 주제를 선택합니다. 대상 선택에서 클라이언트의 이름(AWS Chatbot에 구성된 Slack 또는 Microsoft Teams 클라이언트의 경우) 또는 Amazon SNS 주제의 Amazon 리소스 이름(ARN)(알림에 필요한 정책으로 이미 구성된 Amazon SNS 주제의 경우)을 선택합니다.
- 알림과 함께 사용할 리소스를 구성하지 않은 경우 Create target(대상 생성)을 선택한 다음 SNS 주제를 선택합니다. codestar-notifications- 뒤에 주제 이름을 입력한 다음 생성을 선택합니다.

Note

- 알림 규칙을 만드는 과정에서 Amazon SNS 주제를 생성하면 알림 기능이 주제에 이벤트를 게시할 수 있도록 허용하는 정책이 적용됩니다. 알림 규칙에 대해 생성된 주제를 사용하면 이 리소스에 대한 알림을 받기를 원하는 사용자만 구독할 수 있습니다.
- 알림 규칙 생성 중에는 AWS Chatbot 클라이언트를 생성할 수 없습니다. AWS Chatbot(Slack) 또는 AWS Chatbot(Microsoft Teams)을 선택하면 AWS Chatbot에서 클라이언트를 구성하도록 지시하는 버튼이 표시됩니다. 이 옵션을 선택하면 AWS Chatbot 콘솔이 열립니다. 자세한 내용은 [Slack 채널에 대한 AWS Chatbot 클라이언트 구성](#) 섹션을 참조하세요.
- 기존 Amazon SNS 주제를 대상으로 사용하려면 해당 주제에 대해 존재할 수 있는 다른 정책 외에 AWS CodeStar Notifications에 필요한 정책을 추가해야 합니다. 자세한 정보는 [알림에 대한 Amazon SNS 주제 구성 및 알림 콘텐츠 및 보안 이해](#) 섹션을 참조하세요.

8. 제출을 선택한 다음 알림 규칙을 검토합니다.

Note

사용자가 알림을 받으려면 규칙의 대상으로 지정한 Amazon SNS 주제를 구독하고 구독을 확인해야 합니다. 자세한 내용은 [알림에 사용되는 Amazon SNS 주제에 대해 사용자를 구독하려면](#) 섹션을 참조하세요.

알림 규칙을 생성하려면(AWS CLI)

1. 터미널 또는 명령 프롬프트에서 `create-notification rule` 명령을 실행하여 JSON 스킴레톤을 생성합니다.

```
aws codestar-notifications create-notification-rule --generate-cli-skeleton
> rule.json
```

원하는 대로 파일 이름을 지정할 수 있습니다. 이 예에서는 *rule.json*으로 파일 이름을 지정합니다.

2. 일반 텍스트 편집기에서 JSON 파일을 열고 규칙에 대해 원하는 리소스, 이벤트 유형, Amazon SNS 대상을 포함하도록 편집합니다.

다음 예는 ID가 *123456789012*인 AWS 계정의 *MyDemoRepo*라는 리포지토리에 대한 **MyNotificationRule**이라는 알림 규칙을 보여줍니다. 전체 세부 정보 유형의 알림은 브랜치 및 태그가 생성될 때 *MyNotificationTopic*이라는 Amazon SNS 주제로 전송됩니다.

```
{
  "Name": "MyNotificationRule",
  "EventIds": [
    "codecommit-repository-branches-and-tags-created"
  ],
  "Resource": "arn:aws:codecommit:us-east-1:123456789012:MyDemoRepo",
  "Targets": [
    {
      "TargetType": "SNS",
      "TargetAddress": "arn:aws:sns:us-east-1:123456789012:MyNotificationTopic"
    }
  ],
  "Status": "ENABLED",
  "DetailType": "FULL"
```

```
}

```

파일을 저장합니다.

3. 터미널 또는 명령줄에서 `create-notification-rule` 명령을 다시 실행하여 조금 전 편집한 파일을 사용해 알림 규칙을 생성합니다.

```
aws codestar-notifications create-notification-rule --cli-input-json
file://rule.json

```

4. 성공한 경우 명령에서 다음과 유사한 알림 규칙의 ARN을 반환합니다.

```
{
  "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
}
```

알림 규칙의 이벤트 유형을 나열하려면(AWS CLI)

1. 터미널 또는 명령 프롬프트에서 `list-event-types` 명령을 실행합니다. `--filters` 옵션을 사용하여 응답을 특정 리소스 유형이나 다른 속성으로 제한할 수 있습니다. 예를 들어, 다음은 CodeDeploy 애플리케이션의 이벤트 유형 목록을 반환합니다.

```
aws codestar-notifications list-event-types --filters
Name=SERVICE_NAME,Value=CodeDeploy

```

2. 다음과 비슷한 출력이 생성됩니다.

```
{
  "EventTypes": [
    {
      "EventTypeId": "codedeploy-application-deployment-succeeded",
      "ServiceName": "CodeDeploy",
      "EventTypeName": "Deployment: Succeeded",
      "ResourceType": "Application"
    },
    {
      "EventTypeId": "codedeploy-application-deployment-failed",
      "ServiceName": "CodeDeploy",
      "EventTypeName": "Deployment: Failed",
      "ResourceType": "Application"
    }
  ]
}
```

```

    },
    {
      "EventTypeId": "codedeploy-application-deployment-started",
      "ServiceName": "CodeDeploy",
      "EventTypeName": "Deployment: Started",
      "ResourceType": "Application"
    }
  ]
}

```

알림 규칙에 태그를 추가하려면(AWS CLI)

1. 터미널 또는 명령 프롬프트에서 `tag-resource` 명령을 실행합니다. 예를 들어 다음 명령을 사용하여 `Team`이라는 이름과 `Li_Juan` 값이 있는 태그 키-값 페어를 추가합니다.

```
aws codestar-notifications tag-resource --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/fe1efd35-EXAMPLE --tags Team=Li_Juan
```

2. 다음과 비슷한 출력이 생성됩니다.

```

{
  "Tags": {
    "Team": "Li_Juan"
  }
}

```

알림 규칙 보기

개발자 도구 콘솔 또는 AWS CLI를 사용하여 AWS 리전에 있는 모든 리소스에 대한 알림 규칙을 모두 볼 수 있습니다. 각 알림 규칙의 세부 정보도 볼 수 있습니다. 알림 규칙을 생성하는 프로세스와 달리 리소스에 대한 리소스 페이지로 이동할 필요가 없습니다.

알림 규칙을 보려면(콘솔)

1. <https://console.aws.amazon.com/codesuite/settings/notifications>에서 AWS 개발자 도구 콘솔을 엽니다.
2. 탐색 창에서 설정을 선택한 다음 Notification rules(알림 규칙)를 선택합니다.
3. Notification rules(알림 규칙)에서 현재 로그인한 AWS 리전의 AWS 계정에 있는 리소스에 대해 구성된 규칙 목록을 검토합니다. 선택기를 사용하여 AWS 리전을 변경합니다.

- 알림 규칙의 세부 정보를 보려면 목록에서 규칙을 선택한 다음 세부 정보 보기를 선택합니다. 목록에서 단순히 이름을 선택할 수도 있습니다.

알림 규칙 목록을 보려면(AWS CLI)

- 터미널 또는 명령 프롬프트에서 `list-notification-rules` 명령을 실행하여 지정된 AWS 리전에 대한 모든 알림 규칙을 봅니다.

```
aws codestar-notifications list-notification-rules --region us-east-1
```

- 성공한 경우 이 명령에서 다음과 유사하게 AWS 리전의 각 알림 규칙에 대한 ID 및 ARN을 반환합니다.

```
{
  "NotificationRules": [
    {
      "Id": "dc82df7a-EXAMPLE",
      "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/dc82df7a-EXAMPLE"
    },
    {
      "Id": "8d1f0983-EXAMPLE",
      "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/8d1f0983-EXAMPLE"
    }
  ]
}
```

알림 규칙의 세부 정보를 보려면(AWS CLI)

- 터미널 또는 명령 프롬프트에서 `describe-notification-rule` 명령을 실행하여 알림 규칙의 ARN을 지정합니다.

```
aws codestar-notifications describe-notification-rule --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/dc82df7a-EXAMPLE
```

- 이 작업이 성공하면 다음과 비슷한 출력이 반환됩니다.

```
{
  "LastModifiedTimestamp": 1569199844.857,
```

```

"EventTypes": [
  {
    "ServiceName": "CodeCommit",
    "EventTypeName": "Branches and tags: Created",
    "ResourceType": "Repository",
    "EventTypeId": "codecommit-repository-branches-and-tags-created"
  }
],
>Status": "ENABLED",
"DetailType": "FULL",
"Resource": "arn:aws:codecommit:us-east-1:123456789012:MyDemoRepo",
"Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/dc82df7a-EXAMPLE",
"Targets": [
  {
    "TargetStatus": "ACTIVE",
    "TargetAddress": "arn:aws:sns:us-east-1:123456789012:MyNotificationTopic",
    "TargetType": "SNS"
  }
],
>Name": "MyNotificationRule",
"CreatedTimestamp": 1569199844.857,
"CreatedBy": "arn:aws:iam::123456789012:user/Mary_Major"
}

```

알림 규칙에 대한 태그 목록을 보려면(AWS CLI)

1. 터미널 또는 명령 프롬프트에서 `list-tags-for-resource` 명령을 실행하여 지정된 알림 규칙 ARN에 대한 모든 태그를 봅니다.

```
aws codestar-notifications list-tags-for-resource --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/fe1efd35-EXAMPLE
```

2. 이 명령이 성공하면 다음과 비슷한 출력이 반환됩니다.

```

{
  "Tags": {
    "Team": "Li_Juan"
  }
}

```

알림 규칙 편집

알림 규칙을 편집하여 이름, 알림을 보내는 이벤트, 세부 정보 유형, 알림을 보낼 대상을 변경할 수 있습니다. 개발자 도구 콘솔을 사용하거나 AWS CLI를 사용하여 알림 규칙을 편집할 수 있습니다.

알림 규칙을 편집하려면(콘솔)

1. <https://console.aws.amazon.com/codesuite/settings/notifications>에서 AWS 개발자 도구 콘솔을 엽니다.
2. 탐색 창에서 설정을 선택한 다음 Notification rules(알림 규칙)를 선택합니다.
3. Notification rules(알림 규칙)에서 현재 로그인한 AWS 리전의 AWS 계정에 있는 리소스에 대해 구성된 규칙을 검토합니다. 선택기를 사용하여 AWS 리전을 변경합니다.
4. 목록에서 규칙을 선택한 다음 편집을 선택합니다. 내용을 변경하고 제출을 선택합니다.

알림 규칙을 편집하려면(AWS CLI)

1. 터미널 또는 명령 프롬프트에서 [describe-notification-rule 명령](#)을 실행하여 알림 규칙의 구조를 봅니다.
2. update-notification rule 명령을 실행하여 JSON 스킴레톤을 생성하고 파일로 저장합니다.

```
aws codestar-notifications update-notification-rule --generate-cli-skeleton
> update.json
```

원하는 대로 파일 이름을 지정할 수 있습니다. 이 예에서는 *update.json* 파일입니다.

3. 일반 텍스트 편집기에서 JSON 파일을 열고 규칙을 변경합니다.

다음 예는 ID가 *123456789012*인 AWS 계정의 *MyDemoRepo*라는 리포지토리에 대한 **MyNotificationRule**이라는 알림 규칙을 보여줍니다. 알림은 브랜치 및 태그가 생성될 때 *MyNotificationTopic*이라는 Amazon SNS 주제로 전송됩니다. 규칙 이름이 *MyNewNotificationRule*로 변경됩니다.

```
{
  "Name": "MyNewNotificationRule",
  "EventTypeId": "codecommit-repository-branches-and-tags-created",
  "Resource": "arn:aws:codecommit:us-east-1:123456789012:MyDemoRepo",
  "Targets": [
```

```
{
  "TargetType": "SNS",
  "TargetAddress": "arn:aws:sns:us-
east-1:123456789012:MyNotificationTopic"
},
"Status": "ENABLED",
"DetailType": "FULL"
}
```

파일을 저장합니다.

4. 터미널 또는 명령줄에서 `update-notification-rule` 명령을 다시 실행하여 조금 전 편집한 파일을 사용해 알림 규칙을 업데이트합니다.

```
aws codestar-notifications update-notification-rule --cli-input-json
file://update.json
```

5. 성공한 경우 명령에서 다음과 유사한 알림 규칙의 Amazon 리소스 이름(ARN)을 반환합니다.

```
{
  "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
}
```

알림 규칙에서 대상을 제거하려면(AWS CLI)

1. 터미널 또는 명령 프롬프트에서 `untag-resource` 명령을 실행합니다. 예를 들어 다음 명령은 `Team`이라는 이름의 태그를 제거합니다.

```
aws codestar-notifications untag-resource --arn arn:aws:codestar-notifications:us-
east-1:123456789012:notificationrule/fe1efd35-EXAMPLE --tag-keys Team
```

2. 성공한 경우 이 명령은 아무 것도 반환하지 않습니다.

다음 사항도 참조하세요.

- [알림 규칙에 대한 대상 추가 또는 제거](#)
- [알림 규칙에 대한 알림 사용 또는 사용 중지](#)
- [이벤트](#)

알림 규칙에 대한 알림 사용 또는 사용 중지

알림 규칙을 생성하면 알림이 기본적으로 활성화됩니다. 알림을 보내지 못하도록 규칙을 삭제할 필요가 없습니다. 알림 상태를 간단히 변경할 수 있습니다.

알림 규칙에 대한 알림 상태를 변경하려면(콘솔)

1. <https://console.aws.amazon.com/codesuite/settings/notifications>에서 AWS 개발자 도구 콘솔을 엽니다.
2. 탐색 창에서 설정을 선택한 다음 Notification rules(알림 규칙)를 선택합니다.
3. Notification rules(알림 규칙)에서 현재 로그인한 AWS 리전의 AWS 계정에 있는 리소스에 대해 구성된 규칙을 검토합니다. 선택기를 사용하여 AWS 리전을 변경합니다.
4. 활성화 또는 비활성화할 알림 규칙을 찾아서 선택한 다음 세부 정보를 표시합니다.
5. Notification status(알림 상태)에서 슬라이더를 선택하여 규칙의 상태를 변경합니다.
 - Sending notifications(알림 전송): 기본값입니다.
 - Notifications paused(알림 일시 중지): 지정된 대상에 알림이 전송되지 않습니다.

알림 규칙에 대한 알림 상태를 변경하려면(AWS CLI)

1. [알림 규칙을 편집하려면\(AWS CLI\)](#)의 단계에 따라 알림 규칙에 대한 JSON을 가져옵니다.
2. Status 필드를 ENABLED(기본값) 또는 DISABLED(알림 없음)로 편집한 다음 update-notification-rule 명령을 실행하여 상태를 변경합니다.

```
"Status": "ENABLED"
```

알림 규칙 삭제

리소스에 대해 10개의 알림 규칙만 구성할 수 있으므로 더 이상 필요하지 않은 규칙은 삭제하는 것이 좋습니다. 개발자 도구 콘솔을 사용하거나 AWS CLI를 사용하여 알림 규칙을 삭제할 수 있습니다.

Note

알림 규칙을 삭제한 경우 실행을 취소할 수는 없지만 다시 생성할 수는 있습니다. 알림 규칙을 삭제해도 대상은 삭제되지 않습니다.

알림 규칙을 삭제하려면(콘솔)

1. <https://console.aws.amazon.com/codesuite/settings/notifications>에서 AWS 개발자 도구 콘솔을 엽니다.
2. 탐색 창에서 설정을 선택한 다음 Notification rules(알림 규칙)를 선택합니다.
3. Notification rules(알림 규칙)에서 현재 로그인한 AWS 리전의 AWS 계정에 있는 리소스에 대해 구성된 규칙을 검토합니다. 선택기를 사용하여 AWS 리전을 변경합니다.
4. 알림 규칙을 선택한 다음 삭제를 선택합니다.
5. **delete**를 입력한 다음 삭제를 선택합니다.

알림 규칙을 삭제하려면(AWS CLI)

1. 터미널 또는 명령 프롬프트에서 delete-notification-rule 명령을 실행하여 알림 규칙의 ARN을 지정합니다.

```
aws codestar-notifications delete-notification-rule --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/dc82df7a-EXAMPLE
```

2. 성공한 경우 명령에서 다음과 유사한 삭제된 알림 규칙의 ARN을 반환합니다.

```
{
  "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/dc82df7a-EXAMPLE"
}
```

알림 규칙 대상 사용

알림 규칙 대상은 알림 규칙의 이벤트 조건이 충족될 때 알림을 보내고자 하는 위치를 정의한 목적지입니다. Slack 또는 Microsoft Teams 채널에 대해 구성된 Amazon SNS 주제와 AWS Chatbot 클라이언트 중에서 선택할 수 있습니다. 알림 규칙 생성 중 Amazon SNS 주제를 대상으로 만들 수 있습니다(권장). 알림 규칙과 동일한 AWS 리전의 기존 Amazon SNS 주제를 선택할 수도 있지만 필요한 정책으로 구성해야 합니다. AWS Chatbot 클라이언트를 대상으로 사용하는 경우 먼저 AWS Chatbot에서 해당 클라이언트를 생성해야 합니다.

알림 범위를 확장하고자 하는 경우 알림이 Amazon Chime 채팅룸으로 발송되도록 알림과 AWS Chatbot 간의 통합을 수동으로 구성할 수 있습니다. 그런 다음 해당 AWS Chatbot 클라이언트에 대해

구성된 Amazon SNS 주제를 알림 규칙의 대상으로 선택할 수 있습니다. 자세한 내용은 [AWS Chatbot 및 Amazon Chime과 알림을 통합하려면](#) 섹션을 참조하세요.

개발자 도구 콘솔을 사용하거나 AWS CLI를 사용하여 알림 대상을 관리할 수 있습니다. 콘솔 또는 AWS CLI를 사용하여 Amazon SNS 주제 및 AWS Chatbot 클라이언트를 [대상](#)으로 생성 및 구성할 수 있습니다. 또한 대상으로 구성하는 Amazon SNS 주제와 AWS Chatbot 간의 통합을 구성할 수 있습니다. 이렇게 하면 Amazon Chime 채팅룸에 알림을 보낼 수 있습니다. 자세한 내용은 [알림과 AWS Chatbot 간의 통합 구성](#) 섹션을 참조하세요.

주제

- [알림 규칙 대상 생성 또는 구성](#)
- [알림 규칙 대상 보기](#)
- [알림 규칙에 대한 대상 추가 또는 제거](#)
- [알림 규칙 대상 삭제](#)

알림 규칙 대상 생성 또는 구성

알림 규칙 대상은 Slack 또는 Microsoft Teams 채널에 대해 구성된 Amazon SNS 주제 또는 AWS Chatbot 클라이언트입니다.

AWS Chatbot 클라이언트를 대상으로 선택하려면 먼저 클라이언트를 생성해야 합니다. AWS Chatbot 클라이언트를 알림 규칙의 대상으로 선택하면 Slack 또는 Microsoft Teams 채널로 알림을 보내는 데 필요한 모든 정책이 포함된 Amazon SNS 주제가 해당 AWS Chatbot 클라이언트에 대해 구성됩니다. AWS Chatbot 클라이언트에 대한 기존 Amazon SNS 주제를 구성할 필요가 없습니다.

알림 규칙을 생성할 때 개발자 도구 콘솔에서 Amazon SNS 알림 규칙 대상을 생성할 수 있습니다. 해당 주제로 알림을 보낼 수 있도록 허용하는 정책이 적용됩니다. 이 방법은 알림 규칙의 대상을 생성하는 가장 쉬운 방법입니다. 자세한 내용은 [알림 규칙 생성](#) 섹션을 참조하세요.

기존 Amazon SNS 주제를 사용하는 경우 리소스가 해당 주제에 알림을 보낼 수 있도록 허용하는 액세스 정책을 사용하여 주제를 구성해야 합니다. 예시는 [알림에 대한 Amazon SNS 주제 구성](#)에서 확인하세요.

Note

새 항목을 생성하는 대신 기존 Amazon SNS 항목을 사용하려면 [대상(Targets)]에서 해당 ARN을 선택합니다. 주제에 적절한 액세스 정책이 있는지 여부와 구독자 목록에 리소스에 대한 정보를 볼 수 있는 사용자만 포함되어 있는지를 확인합니다. Amazon SNS 주제가 2019년

11월 5일 이전에 CodeCommit 알림에 사용된 주제인 경우 CodeCommit이 AWS CodeStar Notifications에 필요한 권한과 다른 권한을 포함하는 주제를 게시하도록 허용하는 정책이 포함됩니다. 이러한 주제는 사용하지 않는 것이 좋습니다. 해당 환경에 대해 생성된 정책을 사용하려면 이미 존재하는 정책 외에 AWS CodeStar Notifications에 대한 필수 정책을 추가해야 합니다. 자세한 정보는 [알림에 대한 Amazon SNS 주제 구성](#) 및 [알림 콘텐츠 및 보안 이해](#) 섹션을 참조하세요.

알림 범위를 확장하고자 하는 경우 알림이 Amazon Chime 채팅룸으로 발송되도록 알림과 AWS Chatbot 간의 통합을 수동으로 구성할 수 있습니다. 자세한 정보는 [대상](#) 및 [AWS Chatbot 및 Amazon Chime과 알림을 통합하려면](#) 섹션을 참조하세요.

기존 Amazon SNS 주제를 구성하여 알림 규칙 대상으로 사용하려면(콘솔)

1. AWS Management Console에 로그인하고 <https://console.aws.amazon.com/sns/v3/home>에서 Amazon SNS 콘솔을 엽니다.
2. 탐색 모음에서 [Topics]를 선택합니다. 주제를 선택한 다음 편집을 선택합니다.
3. Access policy(액세스 정책)를 확장한 다음 고급을 선택합니다.
4. JSON 편집기에서 다음 설명을 정책에 추가합니다. 주제 ARN, AWS 리전, AWS 계정 ID 및 주제 이름을 포함합니다.

```
{
  "Sid": "AWSCodeStarNotifications_publish",
  "Effect": "Allow",
  "Principal": {
    "Service": [
      "codestar-notifications.amazonaws.com"
    ]
  },
  "Action": "SNS:Publish",
  "Resource": "arn:aws:sns:us-east-2:123456789012:codestar-notifications-MyTopicForNotificationRules"
}
```

정책 설명은 다음과 같습니다.

```
{
  "Version": "2008-10-17",
  "Id": "__default_policy_ID",
```


```
"Statement": [
  {
    "Sid": "__default_statement_ID",
    "Effect": "Allow",
    "Principal": {
      "AWS": "*"
    },
    "Action": [
      "SNS:GetTopicAttributes",
      "SNS:SetTopicAttributes",
      "SNS:AddPermission",
      "SNS:RemovePermission",
      "SNS:DeleteTopic",
      "SNS:Subscribe",
      "SNS:ListSubscriptionsByTopic",
      "SNS:Publish",
      "SNS:Receive"
    ],
    "Resource": "arn:aws:sns:us-east-2:123456789012:codestar-notifications-MyTopicForNotificationRules",
    "Condition": {
      "StringEquals": {
        "AWS:SourceOwner": "123456789012"
      }
    }
  },
  {
    "Sid": "AWSCodeStarNotifications_publish",
    "Effect": "Allow",
    "Principal": {
      "Service": [
        "codestar-notifications.amazonaws.com"
      ]
    },
    "Action": "SNS:Publish",
    "Resource": "arn:aws:sns:us-east-2:123456789012:codestar-notifications-MyTopicForNotificationRules"
  }
]
```

5. Save changes(변경 사항 저장)를 선택합니다.

6. 구독에서 주제 구독자 목록을 검토합니다. 이 알림 규칙 대상에 대해 적절하게 구독자를 추가, 편집 또는 삭제합니다. 구독자 목록에 리소스에 대한 정보를 볼 수 있는 사용자만 포함되어 있는지 확인합니다. 자세한 내용은 [알림 콘텐츠 및 보안 이해](#) 섹션을 참조하세요.

Slack을 사용하여 대상으로 사용할 AWS Chatbot 클라이언트를 만들려면

1. AWS Chatbot 관리자 가이드에서 [Slack을 사용한 AWS Chatbot 설정](#)의 지침을 따르세요. 이 경우 알림과의 통합을 최적화하려면 다음 선택 사항을 고려하십시오.
 - IAM 역할을 만들 때 이 역할의 목적을 쉽게 식별할 수 있는 역할 이름(예: **AWSCodeStarNotifications-Chatbot-Slack-Role**)을 선택하는 것이 좋습니다. 이 기능은 향후 역할의 목적을 식별하는 데 도움이 될 수 있습니다.
 - SNS 주제에서는 주제나 AWS 리전을 선택할 필요가 없습니다. AWS Chatbot 클라이언트를 [대상](#)으로 선택하면 알림 규칙 생성 프로세스 중 AWS Chatbot 클라이언트에 대해 필요한 모든 권한이 있는 Amazon SNS 주제가 생성되고 구성됩니다.
2. 클라이언트 생성 프로세스를 완료합니다. 그러면 알림 규칙을 생성할 때 이 클라이언트를 대상으로 선택할 수 있습니다. 자세한 내용은 [알림 규칙 생성](#) 섹션을 참조하세요.

 Note

Amazon SNS 주제가 구성된 후에는 AWS Chatbot 클라이언트에서 제거하지 마세요. 이렇게 하면 Slack으로 알림이 전송되지 않습니다.

Microsoft Teams를 사용하여 대상으로 사용할 AWS Chatbot 클라이언트를 만들려면

1. AWS Chatbot 관리자 가이드에서 [Microsoft Teams를 사용한 AWS Chatbot 설정](#)의 지침을 따르세요. 이 경우 알림과의 통합을 최적화하려면 다음 선택 사항을 고려하십시오.
 - IAM 역할을 만들 때 이 역할의 목적을 쉽게 식별할 수 있는 역할 이름(예: **AWSCodeStarNotifications-Chatbot-Microsoft-Teams-Role**)을 선택하는 것이 좋습니다. 이 기능은 향후 역할의 목적을 식별하는 데 도움이 될 수 있습니다.
 - SNS 주제에서는 주제나 AWS 리전을 선택할 필요가 없습니다. AWS Chatbot 클라이언트를 [대상](#)으로 선택하면 알림 규칙 생성 프로세스 중 AWS Chatbot 클라이언트에 대해 필요한 모든 권한이 있는 Amazon SNS 주제가 생성되고 구성됩니다.
2. 클라이언트 생성 프로세스를 완료합니다. 그러면 알림 규칙을 생성할 때 이 클라이언트를 대상으로 선택할 수 있습니다. 자세한 내용은 [알림 규칙 생성](#) 섹션을 참조하세요.

Note

Amazon SNS 주제가 구성된 후에는 AWS Chatbot 클라이언트에서 제거하지 마세요. 이렇게 하면 Microsoft Teams로 알림이 전송되지 않습니다.

알림 규칙 대상 보기

Amazon SNS 콘솔이 아닌 개발자 도구 콘솔을 사용하여 AWS 리전에 있는 모든 리소스에 대한 알림 규칙 대상을 모두 볼 수 있습니다. 알림 규칙 대상의 세부 정보도 볼 수 있습니다.

알림 규칙 대상을 보려면(콘솔)

1. <https://console.aws.amazon.com/codesuite/settings/notifications>에서 AWS 개발자 도구 콘솔을 엽니다.
2. 탐색 창에서 설정을 선택한 다음 Notification rules(알림 규칙)를 선택합니다.
3. Notification rule targets(알림 규칙 대상)에서 현재 로그인한 AWS 리전의 AWS 계정에 있는 알림 규칙에서 사용되는 대상 목록을 검토합니다. 선택기를 사용하여 AWS 리전을 변경합니다. 대상 상태가 연결할 수 없음으로 표시되면 조사가 필요할 수 있습니다. 자세한 내용은 [문제 해결](#) 섹션을 참조하세요.

알림 규칙 대상 목록을 보려면(AWS CLI)

1. 터미널 또는 명령 프롬프트에서 list-targets 명령을 실행하여 지정된 AWS 리전에 대한 모든 알림 규칙 대상 목록을 봅니다.

```
aws codestar-notifications list-targets --region us-east-2
```

2. 성공한 경우 이 명령에서 다음과 유사하게 AWS 리전의 각 알림 규칙에 대한 ID 및 ARN을 반환합니다.

```
{
  "Targets": [
    {
      "TargetAddress": "arn:aws:sns:us-
east-2:123456789012:MySNSTopicForNotificationRules",
      "TargetType": "SNS",
      "TargetStatus": "ACTIVE"
    }
  ]
}
```

```

    },
    {
      "TargetAddress": "arn:aws:chatbot::123456789012:chat-configuration/
slack-channel/MySlackChannelClientForMyDevTeam",
      "TargetStatus": "ACTIVE",
      "TargetType": "AWSChatbotSlack"
    },
    {
      "TargetAddress": "arn:aws:sns:us-
east-2:123456789012:MySNSTopicForNotificationsAboutMyDemoRepo",
      "TargetType": "SNS",
      "TargetStatus": "ACTIVE"
    }
  ]
}

```

알림 규칙에 대한 대상 추가 또는 제거

알림 규칙을 편집하여 알림을 보낼 대상을 변경할 수 있습니다. 개발자 도구 콘솔 또는 AWS CLI를 사용하여 알림 규칙의 대상을 변경할 수 있습니다.

알림 규칙에 대한 대상을 변경하려면(콘솔)

1. <https://console.aws.amazon.com/codesuite/settings/notifications>에서 AWS 개발자 도구 콘솔을 엽니다.
2. 탐색 창에서 설정을 선택한 다음 Notification rules(알림 규칙)를 선택합니다.
3. Notification rules(알림 규칙)에서 현재 로그인한 AWS 리전의 AWS 계정에 있는 리소스에 대해 구성된 규칙 목록을 검토합니다. 선택기를 사용하여 AWS 리전을 변경합니다.
4. 규칙을 선택한 다음 편집을 선택합니다.
5. 대상에서 다음 중 하나를 수행합니다.
 - 다른 대상을 추가하려면 대상 추가를 선택한 다음 목록에서 추가하고자 하는 Amazon SNS 주제 또는 AWS Chatbot(Slack) 또는 AWS Chatbot(Microsoft Teams) 클라이언트를 선택합니다. SNS 주제 생성을 선택하여 주제를 생성하고 대상으로 추가할 수도 있습니다. 알림 규칙의 대상은 최대 10개까지 보유할 수 있습니다.
 - 대상을 제거하려면 제거하고자 하는 대상 옆에 있는 Remove target(대상 제거)을 선택합니다.
6. Submit(제출)을 선택합니다.

알림 규칙에 대한 대상을 추가하려면(AWS CLI)

1. 터미널 또는 명령 프롬프트에서 `subscribe` 명령을 실행하여 대상을 추가합니다. 예를 들어 다음 명령은 Amazon SNS 주제를 알림 규칙의 대상으로 추가합니다.

```
aws codestar-notifications subscribe --arn arn:aws:codestar-
notifications:us-east-1:123456789012:notificationrule/dc82df7a-
EXAMPLE --target TargetType=SNS,TargetAddress=arn:aws:sns:us-
east-1:123456789012:MyNotificationTopic
```

2. 성공한 경우 명령에서 다음과 유사한 업데이트된 알림 규칙의 ARN을 반환합니다.

```
{
  "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
}
```

알림 규칙에서 대상을 제거하려면(AWS CLI)

1. 터미널 또는 명령 프롬프트에서 `unsubscribe` 명령을 실행하여 대상을 제거합니다. 예를 들어 다음 명령은 알림 규칙의 대상으로 Amazon SNS 주제를 제거합니다.

```
aws codestar-notifications unsubscribe --arn arn:aws:codestar-
notifications:us-east-1:123456789012:notificationrule/dc82df7a-
EXAMPLE --target TargetType=SNS,TargetAddress=arn:aws:sns:us-
east-1:123456789012:MyNotificationTopic
```

2. 성공한 경우 명령에서 다음과 유사한 업데이트된 알림 규칙의 ARN과 제거된 대상 관련 정보를 반환합니다.

```
{
  "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
  "TargetAddress": "arn:aws:sns:us-east-1:123456789012:MyNotificationTopic"
}
```

다음 사항도 참조하세요.

- [알림 규칙 편집](#)

- [알림 규칙에 대한 알림 사용 또는 사용 중지](#)

알림 규칙 대상 삭제

더 이상 필요하지 않은 대상은 삭제할 수 있습니다. 리소스에는 10개의 알림 규칙 대상만 구성될 수 있습니다. 따라서 필요하지 않은 대상을 삭제하면 알림 규칙에 추가하고자 하는 다른 대상을 생성할 공간이 마련될 수 있습니다.

Note

알림 규칙 대상을 삭제하면 대상으로 사용하도록 구성된 모든 알림 규칙에서 대상이 제거되지만 대상 자체가 삭제되지는 않습니다.

알림 규칙 대상을 삭제하려면(콘솔)

1. <https://console.aws.amazon.com/codesuite/settings/notifications>에서 AWS 개발자 도구 콘솔을 엽니다.
2. 탐색 창에서 설정을 선택한 다음 Notification rules(알림 규칙)를 선택합니다.
3. Notification rule targets(알림 규칙 대상)에서 현재 로그인한 AWS 리전의 AWS 계정에 있는 리소스에 대해 구성된 대상 목록을 검토합니다. 선택기를 사용하여 AWS 리전을 변경합니다.
4. 알림 규칙 대상을 선택한 다음 삭제를 선택합니다.
5. **delete**를 입력한 다음 삭제를 선택합니다.

알림 규칙 대상을 삭제하려면(AWS CLI)

1. 터미널 또는 명령 프롬프트에서 delete-target 명령을 실행하여 대상의 ARN을 지정합니다. 예를 들어 다음 명령은 Amazon SNS 주제를 사용하는 대상을 삭제합니다.

```
aws codestar-notifications delete-target --target-address arn:aws:sns:us-east-1:123456789012:MyNotificationTopic
```

2. 성공한 경우 이 명령은 아무 것도 반환하지 않습니다. 실패한 경우 이 명령은 오류를 반환합니다. 가장 일반적인 오류는 주제가 하나 이상의 알림 규칙에 대한 대상이 되는 것입니다.

```
An error occurred (ValidationException) when calling the DeleteTarget operation: Unsubscribe target before deleting.
```

--force-unsubscribe-all 파라미터를 사용하여 대상으로 사용하도록 구성된 모든 알림 규칙에서 대상을 제거한 다음 대상을 삭제할 수 있습니다.

```
aws codestar-notifications delete-target --target-address arn:aws:sns:us-east-1:123456789012:MyNotificationTopic --force-unsubscribe-all
```

알림과 AWS Chatbot 간의 통합 구성

AWS Chatbot은 DevOps 및 소프트웨어 개발 팀이 Amazon Chime 채팅룸 및 Slack 채널 및 Microsoft Teams 채널을 사용하여 AWS 클라우드의 운영 이벤트를 모니터링 및 대응할 수 있도록 하는 AWS 서비스입니다. 알림 규칙 대상과 AWS Chatbot 사이의 통합을 구성하여 이벤트 관련 알림이 선택한 Amazon Chime 룸 또는 Slack 채널 또는 Microsoft Teams 채널에 표시되도록 할 수 있습니다. 자세한 내용은 [AWS Chatbot 문서](#)를 참조하세요.

AWS Chatbot 통합을 구성하기 전에 알림 규칙 및 규칙 대상을 구성해야 합니다. 자세한 정보는 [설정](#) 및 [알림 규칙 생성](#) 섹션을 참조하세요. 또한 AWS Chatbot에서 Slack 채널, Microsoft Teams 채널 또는 Amazon Chime 채팅룸을 구성해야 합니다. 자세한 내용은 이러한 서비스에 대한 설명서를 참조하십시오.

주제

- [Slack 채널에 대한 AWS Chatbot 클라이언트 구성](#)
- [Microsoft Teams 채널에 대한 AWS Chatbot 클라이언트 구성](#)
- [Slack 또는 Amazon Chime에 대해 수동으로 클라이언트 구성](#)

Slack 채널에 대한 AWS Chatbot 클라이언트 구성

AWS Chatbot 클라이언트를 대상으로 사용하는 알림 규칙을 생성할 수 있습니다. Slack 채널에 대한 클라이언트를 생성하는 경우 이 클라이언트를 워크플로에서 직접 대상으로 사용하여 알림 규칙을 생성할 수 있습니다. 이는 Slack 채널에 표시되는 알림을 설정하는 가장 쉬운 방법입니다.

Slack을 사용하여 대상으로 사용할 AWS Chatbot 클라이언트를 만들려면

1. AWS Chatbot 관리자 가이드에서 [Slack을 사용한 AWS Chatbot 설정](#)의 지침을 따르세요. 이 경우 알림과의 통합을 최적화하려면 다음 선택 사항을 고려하십시오.

- IAM 역할을 만들 때 이 역할의 목적을 쉽게 식별할 수 있는 역할 이름(예: **AWSCodeStarNotifications-Chatbot-Slack-Role**)을 선택하는 것이 좋습니다. 이 기능은 향후 역할의 목적을 식별하는 데 도움이 될 수 있습니다.
 - SNS 주제에서는 주제나 AWS 리전을 선택할 필요가 없습니다. AWS Chatbot 클라이언트를 **대상**으로 선택하면 알림 규칙 생성 프로세스 중 AWS Chatbot 클라이언트에 대해 필요한 모든 권한이 있는 Amazon SNS 주제가 생성되고 구성됩니다.
2. 클라이언트 생성 프로세스를 완료합니다. 그러면 알림 규칙을 생성할 때 이 클라이언트를 대상으로 선택할 수 있습니다. 자세한 내용은 [알림 규칙 생성](#) 섹션을 참조하세요.

Note

Amazon SNS 주제가 구성된 후에는 AWS Chatbot 클라이언트에서 제거하지 마세요. 이렇게 하면 Slack으로 알림이 전송되지 않습니다.

Microsoft Teams 채널에 대한 AWS Chatbot 클라이언트 구성

AWS Chatbot 클라이언트를 대상으로 사용하는 알림 규칙을 생성할 수 있습니다. Microsoft Teams 채널에 대한 클라이언트를 생성하는 경우 이 클라이언트를 워크플로에서 직접 대상으로 사용하여 알림 규칙을 생성할 수 있습니다. 이는 Microsoft Teams 채널에 표시되는 알림을 설정하는 가장 쉬운 방법입니다.

Microsoft Teams를 사용하여 대상으로 사용할 AWS Chatbot 클라이언트를 만들려면

1. AWS Chatbot 관리자 가이드에서 [Microsoft Teams를 사용한 AWS Chatbot 설정](#)의 지침을 따르세요. 이 경우 알림과의 통합을 최적화하려면 다음 선택 사항을 고려하십시오.
 - IAM 역할을 만들 때 이 역할의 목적을 쉽게 식별할 수 있는 역할 이름(예: **AWSCodeStarNotifications-Chatbot-Microsoft-Teams-Role**)을 선택하는 것이 좋습니다. 이 기능은 향후 역할의 목적을 식별하는 데 도움이 될 수 있습니다.
 - SNS 주제에서는 주제나 AWS 리전을 선택할 필요가 없습니다. AWS Chatbot 클라이언트를 **대상**으로 선택하면 알림 규칙 생성 프로세스 중 AWS Chatbot 클라이언트에 대해 필요한 모든 권한이 있는 Amazon SNS 주제가 생성되고 구성됩니다.
2. 클라이언트 생성 프로세스를 완료합니다. 그러면 알림 규칙을 생성할 때 이 클라이언트를 대상으로 선택할 수 있습니다. 자세한 내용은 [알림 규칙 생성](#) 섹션을 참조하세요.

Note

Amazon SNS 주제가 구성된 후에는 AWS Chatbot 클라이언트에서 제거하지 마세요. 이렇게 하면 Microsoft Teams로 알림이 전송되지 않습니다.

Slack 또는 Amazon Chime에 대해 수동으로 클라이언트 구성

알림과 Slack 또는 Amazon Chime 간의 통합을 직접 생성하도록 선택할 수 있습니다. 이 방법은 Amazon Chime 채팅룸에 대한 알림을 구성하는 데 사용할 수 있는 유일한 방법입니다. 이 통합을 수동으로 구성할 때 이전에 알림 규칙의 대상으로 구성한 Amazon SNS 주제를 사용하는 AWS Chatbot 클라이언트를 생성합니다.

AWS Chatbot 및 Slack과 알림을 수동으로 통합하려면


1. <https://console.aws.amazon.com/codesuite/settings/notifications>에서 AWS 개발자 도구 콘솔을 엽니다.
2. 설정을 선택하고 Notification rules(알림 규칙)를 선택합니다.
3. 알림 규칙 대상에서 대상을 찾아 복사합니다.

Note

대상과 동일한 Amazon SNS 주제를 사용하도록 하나 이상의 알림 규칙을 구성할 수 있습니다. 이렇게 하면 메시지 통합에 도움이 되지만, 구독 목록이 하나의 알림 규칙 또는 리소스에 특정한 경우 의도하지 않은 결과가 발생할 수 있습니다.


4. <https://console.aws.amazon.com/chatbot/>에서 AWS Chatbot 콘솔을 엽니다.
5. Configure new client(새 클라이언트 구성)를 선택한 다음 Slack을 선택합니다.
6. 구성(Configure)을 선택합니다.
7. Slack 작업 영역에 로그인합니다.
8. 선택 사항을 확인하라는 메시지가 나타나면 허용을 선택합니다.
9. Configure new client(새 클라이언트 구성)를 선택합니다.
10. 구성 세부 정보의 구성 이름에 클라이언트 이름을 입력합니다. 알림 규칙을 생성할 때 AWS Chatbot(Slack) 대상 유형에 대해 사용 가능한 대상 목록에 나타나는 이름입니다.

11. Slack 채널 구성(Configure Slack Channel)의 채널 유형(Channel type)에서 통합하고자 하는 채널 유형에 따라 퍼블릭(Public) 또는 프라이빗(Private)을 선택합니다.
 - Public channel(퍼블릭 채널)의 목록에서 Slack 채널의 이름을 선택합니다.
 - Private channel ID(프라이빗 채널 ID)에 채널 코드 또는 URL을 입력합니다.
12. [IAM 권한(IAM permissions)]의 [역할(Role)]에서 [템플릿을 사용하여 역할 생성(Create an IAM role using a template)]을 선택합니다. 정책 템플릿에서 Notification permissions(알림 권한)를 선택합니다. 역할 이름에 이 역할의 이름(예: **AWSCodeStarNotifications-Chatbot-Slack-Role**)을 입력합니다. 정책 템플릿에서 Notification permissions(알림 권한)를 선택합니다.
13. SNS 주제의 SNS Region(SNS 리전)에서 알림 규칙 대상을 생성한 AWS 리전을 선택합니다. [SNS 주제(SNS topics)]에서 알림 규칙 대상으로 구성된 Amazon SNS 주제의 이름을 선택합니다.

 Note

이 클라이언트를 대상으로 사용하여 알림 규칙을 생성하는 경우에는 이 단계가 필요하지 않습니다.

14. 구성(Configure)을 선택합니다.

 Note

프라이빗 채널과의 통합을 구성한 경우 해당 채널에 알림이 표시되기 전에 AWS Chatbot을 채널에 초대해야 합니다. 자세한 내용은 [AWS Chatbot 문서](#)를 참조하세요.

15. (선택 사항) 통합을 테스트하려면 Amazon SNS 주제를 대상으로 사용하도록 구성된 알림 규칙에 대한 이벤트 유형과 일치하는 리소스를 변경합니다. 예를 들어 폴 요청에 의견이 작성되었을 때 알림을 보내도록 알림 규칙이 구성된 경우, 폴 요청에 의견을 작성한 다음 브라우저의 Slack 채널을 확인하여 알림이 표시되는지 확인합니다.

AWS Chatbot 및 Amazon Chime과 알림을 통합하려면

1. <https://console.aws.amazon.com/codesuite/settings/notifications>에서 AWS 개발자 도구 콘솔을 엽니다.
2. 설정을 선택하고 Notification rules(알림 규칙)를 선택합니다.
3. 알림 규칙 대상에서 대상을 찾아 복사합니다.

Note

대상과 동일한 Amazon SNS 주제를 사용하도록 하나 이상의 알림 규칙을 구성할 수 있습니다. 이는 메시지 통합에 도움이 되지만, 구독 목록이 하나의 알림 규칙 또는 리소스에 특정한 경우 의도하지 않은 결과가 발생할 수 있습니다.

4. Amazon Chime에서 통합에 대해 구성하고자 하는 채팅룸을 엽니다.
5. 오른쪽 상단 모서리에 있는 기어 아이콘을 선택하고 Manage webhooks(Webhook 관리)를 선택합니다.
6. Manage webhooks(Webhook 관리) 대화 상자에서 새로 만들기를 선택하고 Webhook 이름을 입력한 후 생성을 선택합니다.
7. Webhook이 나타나는지 확인한 다음 Copy webhook URL(Webhook URL 복사)을 선택합니다.
8. <https://console.aws.amazon.com/chatbot/>에서 AWS Chatbot 콘솔을 엽니다.
9. [새 클라이언트 구성(Configure new client)]을 선택한 다음 [Amazon Chime]을 선택합니다.
10. 구성 세부 정보의 구성 이름에 클라이언트 이름을 입력합니다.
11. Webhook URL(Webhook URL)에 URL을 붙여넣습니다. Webhook description(Webhook 설명)에 선택적 설명을 입력합니다.
12. [IAM 권한(IAM permissions)]의 [역할(Role)]에서 [템플릿을 사용하여 역할 생성(Create an IAM role using a template)]을 선택합니다. 정책 템플릿에서 Notification permissions(알림 권한)를 선택합니다. 역할 이름에 이 역할의 이름(예: **AWSCodeStarNotifications-Chatbot-Chime-Role**)을 입력합니다.
13. SNS 주제의 SNS Region(SNS 리전)에서 알림 규칙 대상을 생성한 AWS 리전을 선택합니다. [SNS 주제(SNS topics)]에서 알림 규칙 대상으로 구성된 Amazon SNS 주제의 이름을 선택합니다.
14. 구성(Configure)을 선택합니다.
15. (선택 사항) 통합을 테스트하려면 Amazon SNS 주제를 대상으로 사용하도록 구성된 알림 규칙에 대한 이벤트 유형과 일치하는 리소스를 변경합니다. 예를 들어 폴 요청에 의견이 작성되었을 때 알림을 보내도록 알림 규칙이 구성된 경우, 폴 요청에 의견을 작성한 다음 Amazon Chime 채팅룸을 확인하여 알림이 표시되는지 확인합니다.

AWS CloudTrail을 사용하여 AWS CodeStar Notifications API 호출 로깅

AWS CodeStar Notifications는 사용자, 역할 또는 AWS 서비스가 수행한 작업에 대한 레코드를 제공하는 서비스인 AWS CloudTrail과 통합됩니다. CloudTrail은 알림을 제공하기 위해 API 호출을 이벤트로 캡처합니다. 캡처되는 호출에는 개발자 도구 콘솔에서 수행한 호출과 AWS CodeStar Notifications

API 작업에 대한 코드 호출이 포함됩니다. 추적을 생성하면 알림 이벤트를 포함한 CloudTrail 이벤트를 지속적으로 Amazon S3 버킷에 배포할 수 있습니다. 추적을 구성하지 않은 경우에도 CloudTrail 콘솔의 이벤트 기록에서 최신 이벤트를 볼 수 있습니다. CloudTrail에서 수집한 정보를 사용하여 AWS CodeStar Notifications에 대해 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 기타 세부 정보를 확인할 수 있습니다.

자세한 정보는 [AWS CloudTrail 사용 설명서](#)를 참조하세요.

CloudTrail의 AWS CodeStar Notifications 정보

CloudTrail은 계정 생성 시 AWS 계정에서 사용되도록 설정됩니다. AWS CodeStar Notifications에서 활동이 발생하면 해당 활동이 [이벤트 기록(Event history)]의 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 보기](#)를 참조하세요.

AWS CodeStar Notifications 이벤트를 비롯하여 AWS 계정 계정의 이벤트의 진행 중인 기록을 보유하려면 추적을 생성합니다. CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 콘솔에서 추적을 생성하면 기본적으로 모든 AWS 리전에 추적이 적용됩니다. 추적은 AWS 파티션에 있는 모든 리전의 이벤트를 로깅하고 지정된 Amazon S3 버킷으로 로그 파일을 전송합니다. 또는 CloudTrail 로그에서 수집된 이벤트 데이터를 추가 분석 및 처리하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 내용은 다음 자료를 참조하세요.

- [추적 생성 개요](#)
- [CloudTrail 지원 서비스 및 통합](#)
- [CloudTrail에 대한 SNS 알림 구성](#)
- [여러 리전에서 CloudTrail 로그 파일 받기](#) 및 [여러 계정에서 CloudTrail 로그 파일 받기](#)

모든 AWS CodeStar Notifications 작업은 CloudTrail에 의해 로깅되며 [AWS CodeStar Notifications API ##](#)에 설명되어 있습니다. 예를 들어 CreateNotificationRule, Subscribe, ListEventTypes 작업을 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 AWS Identity and Access Management(IAM) 사용자 자격 증명으로 했는지.
- 역할 또는 페더레이션 사용자에게 대한 임시 보안 자격 증명을 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청했는지 여부.

자세한 내용은 [CloudTrail userIdentity 요소](#)를 참조하세요.

로그 파일 항목 이해

추적이란 지정한 Amazon S3 버킷에 이벤트를 로그 파일로 입력할 수 있게 하는 구성입니다.

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 호출의 주문 스택 트레이스가 아니므로 특정 순서로 표시되지 않습니다.

다음 예제는 CreateNotificationRule 및 Subscribe 작업을 모두 포함한 알림 규칙의 생성을 보여주는 CloudTrail 로그 항목을 표시합니다.

Note

알림 로그 파일 항목의 이벤트 중 일부는 서비스 연결 역할인 AWSServiceRoleForCodeStarNotifications에서 발생할 수 있습니다.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/Mary_Major",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "Mary_Major"
  },
  "eventTime": "2019-10-07T21:34:41Z",
  "eventSource": "events.amazonaws.com",
  "eventName": "CreateNotificationRule",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "codestar-notifications.amazonaws.com",
  "userAgent": "codestar-notifications.amazonaws.com",
  "requestParameters": {
    "description": "This rule is used to route CodeBuild, CodeCommit, CodePipeline, and other Developer Tools notifications to AWS CodeStar Notifications",
    "name": "awscodestarnotifications-rule",
    "eventPattern": "{\"source\": [\"aws.codebuild\", \"aws.codecommit\", \"aws.codepipeline\"]}"
  },
  "responseElements": {
```



```

    "ruleArn": "arn:aws:events:us-east-1:123456789012:rule/
awscodestarnotifications-rule"
  },
  "requestID": "ff1f309a-EXAMPLE",
  "eventID": "93c82b07-EXAMPLE",
  "eventType": "AwsApiCall",
  "apiVersion": "2015-10-07",
  "recipientAccountId": "123456789012"
}

```

```

{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/Mary_Major",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "Mary_Major"
  },
  "eventTime": "2019-10-07T21:34:41Z",
  "eventSource": "events.amazonaws.com",
  "eventName": "Subscribe",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "codestar-notifications.amazonaws.com",
  "userAgent": "codestar-notifications.amazonaws.com",
  "requestParameters": {
    "targets": [
      {
        "arn": "arn:aws:codestar-notifications:us-east-1:::",
        "id": "codestar-notifications-events-target"
      }
    ]
  },
  "rule": "awscodestarnotifications-rule"
},
"responseElements": {
  "failedEntryCount": 0,
  "failedEntries": []
},
"requestID": "9466cbda-EXAMPLE",
"eventID": "2f79fdad-EXAMPLE",
"eventType": "AwsApiCall",
"apiVersion": "2015-10-07",

```

```
"recipientAccountId": "123456789012"
}
```

문제 해결

다음 정보는 알림과 관련된 일반적인 문제를 해결하는 데 도움이 됩니다.

주제

- [리소스에 대한 알림 규칙을 생성하려 할 때 권한 오류가 발생함](#)
- [알림 규칙을 볼 수 없음](#)
- [알림 규칙을 생성할 수 없음](#)
- [액세스할 수 없는 리소스에 대한 알림을 수신하고 있음](#)
- [Amazon SNS 알림이 수신되지 않음](#)
- [이벤트에 관한 중복 알림을 수신하고 있음](#)
- [알림 대상 상태가 연결할 수 없음으로 표시되는 이유](#)
- [알림 및 리소스에 대한 할당량을 늘리려는 경우](#)

리소스에 대한 알림 규칙을 생성하려 할 때 권한 오류가 발생함

적절한 권한이 있는지 확인하십시오. 자세한 내용은 [자격 증명 기반 정책 예시](#) 섹션을 참조하세요.

알림 규칙을 볼 수 없음

문제: 개발자 도구 콘솔을 사용할 경우 [설정(Settings)]에서 [알림(Notifications)]을 선택하면 권한 오류가 표시됩니다.

해결 방법: 알림을 보는 데 필요한 권한이 없을 수 있습니다. CodeCommit 및 CodePipeline과 같은 AWS 개발자 도구 서비스에 대한 대부분의 관리형 정책에는 알림 권한이 포함되어 있지만 현재 알림을 지원하지 않는 서비스의 경우 알림을 표시할 권한이 포함되어 있지 않습니다. 또는 IAM 사용자 또는 역할에 알림을 보도록 허용하지 않는 사용자 지정 정책이 적용되었을 수 있습니다. 자세한 내용은 [자격 증명 기반 정책 예시](#) 섹션을 참조하세요.

알림 규칙을 생성할 수 없음

알림 규칙을 만드는 데 필요한 권한이 없을 수 있습니다. 자세한 내용은 [자격 증명 기반 정책 예시](#) 섹션을 참조하세요.

액세스할 수 없는 리소스에 대한 알림을 수신하고 있음

알림 규칙을 생성하고 대상을 추가할 때 알림 기능은 수신자가 리소스에 액세스할 수 있는지 여부를 확인하지 않습니다. 액세스할 수 없는 리소스에 관한 알림을 수신하는 것은 가능합니다. 직접 제거할 수 없는 경우 대상의 구독 목록에서 제거하도록 요청하십시오.

Amazon SNS 알림이 수신되지 않음

Amazon SNS 주제 관련 문제를 해결하려면 다음을 확인합니다.

- Amazon SNS 주제가 알림 규칙과 동일한 AWS 리전에서 생성되었는지 확인합니다.
- 이메일 별칭이 올바른 주제를 구독하고 구독을 확인했는지 확인해야 합니다. 자세한 내용은 [엔드포인트를 Amazon SNS 주제에 구독 설정](#)을 참조하세요.
- 주제 정책이 AWS CodeStar Notifications에서 해당 주제에 대한 알림 푸시를 허용하도록 편집되었는지 확인합니다. 이 주제 정책에 다음과 유사한 문이 포함되어야 합니다.

```
{
  "Sid": "AWSCodeStarNotifications_publish",
  "Effect": "Allow",
  "Principal": {
    "Service": [
      "codestar-notifications.amazonaws.com"
    ]
  },
  "Action": "SNS:Publish",
  "Resource": "arn:aws:sns:us-east-1:123456789012:MyNotificationTopicName",
  "Condition": {
    "StringEquals": {
      "aws:SourceAccount": "123456789012"
    }
  }
}
```

자세한 내용은 [알림에 대한 Amazon SNS 주제 구성](#) 섹션을 참조하세요.

이벤트에 관한 중복 알림을 수신하고 있음

여러 알림을 수신하는 가장 일반적인 이유는 다음과 같습니다.

- 동일한 이벤트 유형을 포함하는 여러 알림 규칙이 리소스에 대해 구성되어 있고 해당 규칙의 대상인 Amazon SNS 주제에 구독한 상태입니다. 이 문제를 해결하려면 주제 중 하나에 대한 구독을 취소하거나 중복을 피하도록 알림 규칙을 편집합니다.
- 하나 이상의 알림 규칙 대상이 AWS Chatbot과 통합되어 있고, 이메일 수신함과 Slack 채널, Microsoft Teams 채널 또는 Amazon Chime 채팅룸 모두에서 알림을 수신하고 있습니다. 이 문제를 해결하려면 규칙의 대상인 Amazon SNS 주제의 이메일 주소를 구독 해제하거나, Slack 채널, Microsoft Teams 채널 또는 Amazon Chime 채팅룸을 사용하여 알림을 확인합니다.

알림 대상 상태가 연결할 수 없음으로 표시되는 이유

대상에는 활성 및 연결할 수 없음이라는 두 가지 상태가 있습니다. 연결할 수 없음은 알림이 대상에 전송되었으나 전달이 성공하지 못했음을 나타냅니다. 알림은 해당 대상에 계속 전송되며 성공하면 상태가 활성으로 재설정됩니다.

다음 이유 중 하나로 인해 알림 규칙의 대상을 사용할 수 없게 될 수 있습니다.

- 리소스(Amazon SNS 주제 또는 AWS Chatbot 클라이언트)가 삭제되었습니다. 알림 규칙에 대해 다른 대상을 선택하십시오.
- Amazon SNS 주제가 암호화되고 암호화된 주제에 대한 필수 정책이 없거나 AWS KMS 키가 삭제되었습니다. 자세한 내용은 [알림에 대한 Amazon SNS 주제 구성](#) 섹션을 참조하세요.
- 알림에 필요한 정책이 Amazon SNS 주제에 없습니다. 정책이 없으면 Amazon SNS 주제로 알림을 보낼 수 없습니다. 자세한 내용은 [알림에 대한 Amazon SNS 주제 구성](#) 섹션을 참조하세요.
- 대상(Amazon SNS 또는 AWS Chatbot)에 대한 지원 서비스에 문제가 있는 것 같습니다.

알림 및 리소스에 대한 할당량을 늘리려는 경우

현재는 할당량을 변경할 수 없습니다. [알림의 할당량](#)을(를) 참조하세요.

알림의 할당량

다음 표에는 개발자 도구 콘솔의 알림에 대한 할당량(한도라고도 함)이 나열되어 있습니다. 변경할 수 있는 할당량에 대한 자세한 내용은 [AWS service quotas](#)(AWS 서비스 할당량)를 참조하세요.

리소스	기본 한도
AWS 계정의 최대 알림 규칙 수	1000

리소스	기본 한도
알림 규칙에 대한 최대 대상 수	10
리소스에 대한 최대 알림 규칙 수	10

연결이란 무엇입니까?

개발자 도구 콘솔의 연결 기능을 사용하여 외부 코드 리포지토리와 같은 AWS 리소스를 연결할 AWS CodePipeline 수 있습니다. 이 기능에는 [AWS CodeStar 연결](#) API 참조라는 자체 API가 있습니다. 각 연결은 타사 리포지토리 (예:) 에 연결하기 위해 AWS 서비스에 제공할 수 있는 리소스입니다 BitBucket. 예를 들어 타사 코드 리포지토리의 코드가 변경될 때 파이프라인이 CodePipeline 트리거 되도록 연결을 추가할 수 있습니다. 각 연결은 이름이 지정되고 연결을 참조하는 데 사용되는 고유한 Amazon 리소스 이름(ARN)과 연결됩니다.

연결로 어떤 작업을 할 수 있습니까?

다음과 같은 개발자 도구에서 연결을 사용하여 서드 파티 공급자 리소스를 AWS 리소스와 통합할 수 있습니다.

- Bitbucket과 같은 타사 공급자에 연결하고 타사 연결을 AWS 리소스 (예:) 와의 소스 통합으로 사용하십시오. CodePipeline
- 타사 공급자를 위해 CodeBuild 빌드 프로젝트, CodeDeploy 애플리케이션, 파이프라인의 리소스 전반에서 연결에 CodePipeline 대한 액세스를 균일하게 관리하세요.
- 저장된 비밀번호나 매개변수를 참조할 필요 없이 스택 템플릿의 연결 ARN을 사용하여 프로젝트, CodeDeploy 애플리케이션 CodePipeline, 파이프라인을 CodeBuild 빌드할 수 있습니다.

어떤 서드 파티 공급자에 대한 연결을 생성할 수 있나요?

연결을 통해 AWS 리소스를 다음과 같은 타사 리포지토리와 연결할 수 있습니다.

- Bitbucket Cloud
- GitHub
- GitHub 엔터프라이즈 클라우드
- GitHub 엔터프라이즈 서버

- GitLab
- GitLab 자체 관리형 설치 (엔터프라이즈 에디션 또는 커뮤니티 에디션용)

연결 워크플로에 대한 개요는 [연결을 생성하거나 업데이트하는 워크플로](#) 섹션을 참조하세요.

클라우드 제공자 유형 (예:) 에 대한 연결을 생성하는 단계는 설치된 제공자 유형 (예: GitHub Enterprise Server) 의 단계와 다릅니다. GitHub 공급자 유형별로 연결을 생성하는 전체 단계는 [연결 관련 작업](#) 섹션을 참조하세요.

Note

유럽 (밀라노) AWS 리전에서 연결을 사용하려면 다음을 수행해야 합니다.

1. 리전별 앱 설치
2. 리전 활성화

이 리전별 앱은 유럽(밀라노) 리전 내 연결을 지원합니다. 서드 파티 제공업체 사이트에 게시되며 다른 리전의 연결을 지원하는 기존 앱과는 별개입니다. 이 앱을 설치하면 서드 파티 제공업체가 이 리전에서만 서비스와 데이터를 공유할 수 있는 권한을 부여하게 되며 앱을 제거하여 언제든지 권한을 취소할 수 있습니다.

리전을 활성화하지 않으면 서비스에서 데이터를 처리하거나 저장하지 않습니다. 이 리전을 활성화하면 데이터를 처리하고 저장할 수 있는 권한이 서비스에 부여됩니다.

리전이 활성화되지 않았더라도 리전별 앱이 설치된 상태로 유지되면 서드 파티 제공업체가 서비스와 데이터를 공유할 수 있으므로 리전을 비활성화한 후에는 앱을 제거해야 합니다. 자세한 내용은 [리전 활성화](#)를 참조하세요.

연결과 AWS 서비스 통합되는 것은 무엇인가요?

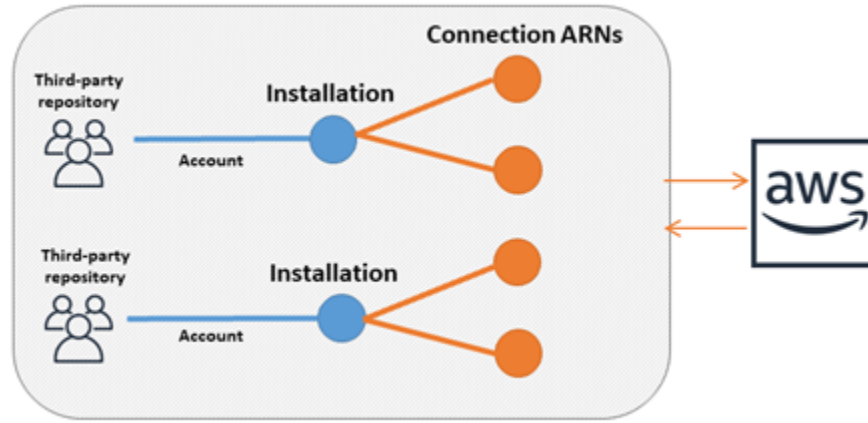
연결을 사용하여 서드 파티 리포지토리를 다른 AWS 서비스와 통합할 수 있습니다. 연결에 대한 서비스 통합을 보려면 [AWS CodeStar Connections와 제품 및 서비스 통합](#) 섹션을 참조하세요.

연결은 어떻게 작동합니까?

연결을 생성하려면 먼저 서드 파티 계정에 AWS 인증 앱을 설치하거나 액세스 권한을 제공해야 합니다. 연결을 설치한 후 해당 설치를 사용하도록 업데이트할 수 있습니다. 연결을 생성할 때 서드 파티 계정의 AWS 리소스에 대한 액세스 권한을 제공합니다. 이렇게 하면 연결을 통해 리소스를 대신하여 타

사 계정의 소스 리포지토리와 같은 콘텐츠에 액세스할 수 AWS 있습니다. 그런 다음 해당 연결을 다른 사람과 AWS 서비스 공유하여 리소스 간에 안전한 OAuth 연결을 제공할 수 있습니다.

GitHubEnterprise Server와 같은 설치된 제공자 유형에 대한 연결을 생성하려면 먼저 를 사용하여 호스트 리소스를 생성합니다. AWS Management Console



연결은 연결을 AWS 계정 생성한 사람이 소유합니다. 연결은 연결 ID가 포함된 ARN으로 식별됩니다. 연결 ID는 변경하거나 다시 매핑할 수 없는 UUID입니다. 연결을 삭제하고 다시 설정하면 새 연결 ID가 생성됨에 따라 새 연결 ARN이 생성됩니다. 즉, 연결 ARN은 절대 재사용되지 않습니다.

새로 생성된 연결은 Pending 상태입니다. 연결 설정을 완료하고 연결을 Pending 상태에서 Available 상태로 전환하려면 서드 파티 핸드셰이크(OAuth 흐름) 프로세스가 필요합니다. 이 작업이 완료되면 다음과 같은 AWS 서비스에서 연결을 사용할 수 있게 CodePipeline 됩니다. Available

새로 생성된 호스트는 Pending 상태입니다. 호스트 설정을 완료하고 호스트를 Pending 상태에서 Available 상태로 전환하려면 서드 파티 등록 프로세스가 필요합니다. 이 프로세스가 완료되면 호스트는 Available 상태가 되며 설치된 공급자 유형으로의 연결에 사용할 수 있습니다.

연결 워크플로에 대한 개요는 [연결을 생성하거나 업데이트하는 워크플로](#) 섹션을 참조하세요. 설치된 공급자의 호스트 생성 워크플로에 대한 개요는 [호스트 생성 또는 업데이트 워크플로우](#) 섹션을 참조하십시오. 공급자 유형별로 연결을 생성하는 전체 단계는 [연결 관련 작업](#) 섹션을 참조하세요.

AWS CodeStar Connections의 글로벌 리소스

연결은 글로벌 리소스입니다. 즉, 리소스가 모든 AWS 리전에 복제됩니다.

연결 ARN 형식은 생성된 리전 이름을 반영하지만 리소스의 사용 범위는 특정 리전으로만 제한되지 않습니다. 연결 리소스가 만들어진 리전은 연결 리소스 데이터 업데이트가 제어되는 리전입니다. 연결 리소스 데이터에 대한 업데이트를 제어하는 API 작업의 예로는 연결 생성, 설치 업데이트, 연결 삭제, 연결 태그 지정 등이 있습니다.

연결을 위한 호스트 리소스는 전역적으로 사용할 수 있는 리소스가 아닙니다. 호스트 리소스는 생성된 리전에서만 사용됩니다.

- 연결을 한 번만 생성하면 모든 AWS 리전에서 해당 연결을 사용할 수 있습니다.
- 연결이 생성된 리전에 문제가 있는 경우 연결 리소스 데이터를 제어하는 API에 영향을 미치지만 다른 모든 리전에서 연결을 정상적으로 사용할 수 있습니다.
- 콘솔 또는 CLI에 연결 리소스를 나열하면 목록에 모든 리전의 계정과 연결된 모든 연결 리소스가 표시됩니다.
- 콘솔 또는 CLI에 호스트 리소스를 나열하면 선택한 리전의 계정과 연결된 호스트 리소스만 목록에 표시됩니다.
- 연결된 호스트 리소스와의 연결을 CLI에 나열하거나 표시할 경우, 출력에서는 구성된 CLI 리전에 관계없이 호스트 ARN 반환합니다.

호스트 생성 또는 업데이트 워크플로우

설치된 공급자에 대한 연결을 생성할 때는 먼저 호스트를 생성합니다.

호스트는 다음 상태를 가질 수 있습니다.

- Pending - pending 호스트는 생성된 호스트이므로 설정(available로 이동)해야 사용할 수 있습니다.
- Available - 연결에 available 호스트를 사용하거나 전달할 수 있습니다.

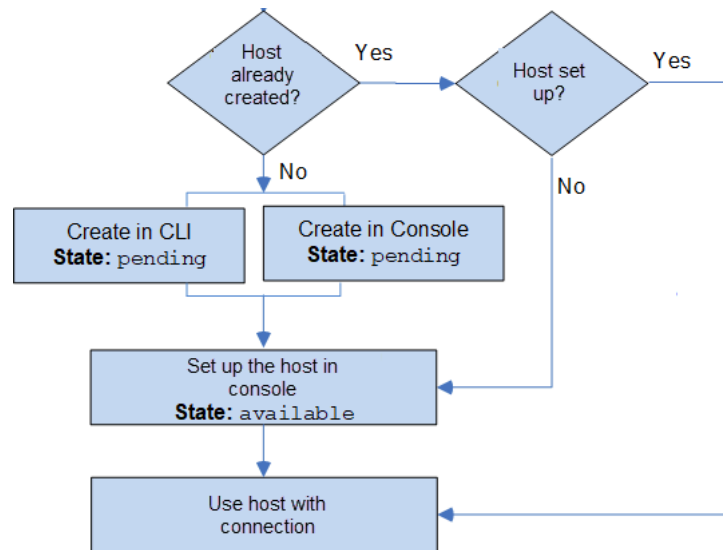
워크플로우: CLI, SDK 또는 AWS CloudFormation을 사용하여 연결 생성 또는 업데이트

[CreateHostAPI](#)를 사용하여 AWS Command Line Interface (AWS CLI), SDK 또는 AWS CloudFormation를 사용하여 호스트를 생성합니다. 생성된 호스트는 pending 상태입니다. 콘솔의 콘솔 설정 옵션을 사용하여 프로세스를 완료합니다.

워크플로우: 콘솔을 사용하여 호스트 생성 또는 업데이트

설치된 제공자 유형 (예: GitHub Enterprise Server 또는 GitLab 자체 관리) 에 대한 연결을 생성하는 경우 먼저 호스트를 생성합니다. Bitbucket과 같은 클라우드 공급자 유형에 연결하는 경우 호스트 생성 단계를 건너뛰고 연결 생성을 계속 진행합니다.

콘솔을 사용하여 호스트를 설정하고 그 상태를 pending에서 available로 변경합니다.



연결을 생성하거나 업데이트하는 워크플로

연결을 생성할 때 서드 파티 공급자와의 인증 핸드셰이크에 대한 기존 설치도 생성하거나 사용합니다.

연결 상태는 다음이 될 수 있습니다.

- Pending - pending 연결은 먼저 완료(available로 전환)해야 사용할 수 있습니다.
- Available - available 연결은 계정의 다른 리소스 및 사용자에게 사용하거나 전달할 수 있습니다.
- Error - error 상태의 연결은 자동으로 다시 시도됩니다. available 상태가 될 때까지 사용할 수 없습니다.

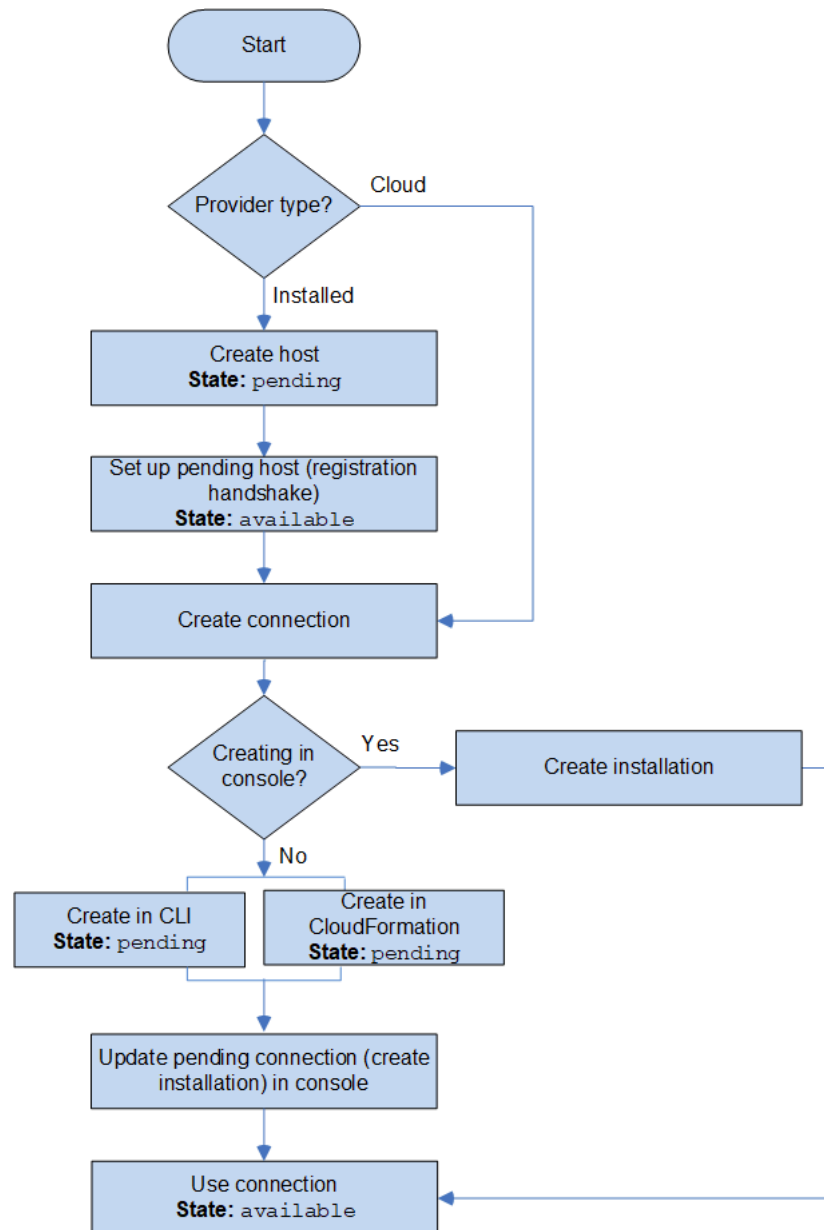
워크플로: CLI, SDK 또는 AWS CloudFormation을 사용하여 연결 생성 또는 업데이트

[CreateConnection](#) API를 사용하여 AWS Command Line Interface (AWS CLI), SDK 또는 을 사용하여 연결을 생성합니다. AWS CloudFormation 생성된 연결은 pending 상태입니다. 콘솔의 [보류 중인 연결 설정(Set up pending connection)] 옵션을 사용하여 프로세스를 완료합니다. 연결에 대한 설치를 생성하거나 기존 설치를 사용하라는 메시지가 콘솔에 표시됩니다. 그런 다음 콘솔을 사용하여 핸드셰이크를 완료하고 콘솔에서 [연결 완료(Complete connection)]를 선택하여 연결을 available 상태로 전환합니다.

워크플로: 콘솔을 사용하여 연결 생성 또는 업데이트

설치된 공급자 유형 (예: GitHub Enterprise Server) 에 대한 연결을 생성하는 경우 먼저 호스트를 생성합니다. Bitbucket과 같은 클라우드 공급자 유형에 연결하는 경우 호스트 생성 단계를 건너뛰고 연결 생성을 계속 진행합니다.

콘솔을 사용하여 연결을 생성하거나 업데이트하려면 콘솔의 CodePipeline 편집 작업 페이지를 사용하여 타사 제공자를 선택합니다. 설치를 생성하거나 연결에 대한 기존 설치를 사용하고, 콘솔을 통해 연결을 생성하라는 메시지가 콘솔에 표시됩니다. 콘솔이 핸드셰이크를 완료하고 연결을 pending 상태에서 available 상태로 자동으로 전환합니다.



연결을 시작하려면 어떻게 해야 하나요?

시작하려면 다음과 같은 유용한 주제를 검토하십시오.

- 연결과 관련한 [개념](#)을 자세히 알아봅니다.
- [필요한 리소스](#)를 설정하여 연결 작업을 시작합니다.

- [첫 번째 연결](#)을 시작하고 리소스에 연결합니다.

연결 관련 개념

개념 및 용어를 이해하면 연결 기능의 설정과 사용이 더 쉬워집니다. 다음은 개발자 도구 콘솔에서 연결을 사용할 때 알아야 할 개념입니다.

설치

서드 파티 계정에 있는 AWS 앱의 인스턴스입니다. AWS CodeStar Connector 앱을 설치하면 AWS가 서드 파티 계정 내의 리소스에 액세스할 수 있습니다. 설치하는 서드 파티 공급자의 웹 사이트에서만 편집할 수 있습니다.

ODBC

서드 파티 소스 리포지토리를 AWS 서비스에 연결하는 데 사용되는 AWS 리소스입니다.

서드 파티 리포지토리

AWS에 속하지 않은 서비스 또는 회사에서 제공하는 리포지토리입니다. 예를 들어 BitBucket 리포지토리는 서드 파티 리포지토리입니다.

공급자 유형

연결하려는 서드 파티 소스 리포지토리를 제공하는 서비스 또는 회사입니다. AWS 리소스를 외부 공급자 유형에 연결합니다. 소스 리포지토리가 네트워크와 인프라에 설치되는 공급자 유형을 설치된 공급자 유형이라고 합니다. 예를 들어 GitHub Enterprise Server는 설치된 공급자 유형입니다.

호스트

서드 파티 공급자가 설치된 인프라를 나타내는 리소스입니다. 연결은 호스트를 사용하여 GitHub 엔터프라이즈 서버와 같이 서드 파티 공급자가 설치된 서버를 나타냅니다. 해당 공급자 유형에 대한 모든 연결에 대해 하나의 호스트를 만듭니다.

Note

콘솔을 사용하여 GitHub Enterprise Server에 대한 연결을 생성하면 콘솔은 프로세스의 일부로 호스트 리소스를 생성합니다.

AWS CodeStar 연결 지원 제공자 및 버전

이 장에서는 AWS CodeStar Connections가 지원하는 공급자 및 버전에 대한 정보를 제공합니다.

주제

- [Bitbucket에 지원되는 공급자 유형](#)
- [지원되는 프로바이더 유형 및 엔터프라이즈 클라우드 GitHub GitHub](#)
- [GitHub 엔터프라이즈 서버에 지원되는 공급자 유형 및 버전](#)
- [지원되는 제공자 유형: GitLab](#)
- [GitLab 자체 관리용으로 지원되는 제공자 유형](#)

Bitbucket에 지원되는 공급자 유형

Atlassian Bitbucket Cloud와 함께 AWS CodeStar 앱을 사용할 수 있습니다.

Bitbucket Server와 같은 설치된 Bitbucket 공급자 유형은 지원되지 않습니다.

지원되는 프로바이더 유형 및 엔터프라이즈 클라우드 GitHub GitHub

GitHub 앱용 AWS 커넥터를 GitHub 엔터프라이즈 GitHub 클라우드와 함께 사용할 수 있습니다.

GitHub 엔터프라이즈 서버에 지원되는 공급자 유형 및 버전

지원되는 버전의 GitHub Enterprise Server와 함께 AWS CodeStar 앱을 사용할 수 있습니다. 지원되는 버전 목록은 <https://enterprise.github.com/releases/> 섹션을 참조하세요.

Important

AWS CodeStar 연결은 더 이상 사용되지 않는 GitHub 엔터프라이즈 서버 버전을 지원하지 않습니다. 예를 들어 릴리스의 알려진 문제로 인해 AWS CodeStar Connections는 GitHub 엔터프라이즈 서버 버전 2.22.0을 지원하지 않습니다. 연결하려면 버전 2.22.1 또는 사용 가능한 최신 버전으로 업그레이드하세요.

지원되는 제공자 유형: GitLab

와 연결을 사용할 수 GitLab 있습니다. 자세한 정보는 [에 대한 연결 생성 GitLab](#)을 참조하세요.

GitLab 자체 관리용으로 지원되는 제공자 유형

GitLab 자체 관리형 설치 (Enterprise Edition 또는 Community Edition용) 를 통해 연결을 사용할 수 있습니다. 자세한 내용은 [GitLab 자체 관리형 연결을 생성하세요.](#)을(를) 참조하세요.

AWS CodeStar Connections와 제품 및 서비스 통합

기본적으로 AWS CodeStar Connections는 여러 개의 AWS 서비스와 파트너 제품 및 서비스와 통합됩니다. 다음 섹션의 정보를 이용해, 사용 중인 제품 및 서비스와 통합할 연결을 구성할 수 있습니다.

다음 표에는 이 서비스를 이용할 때 참조할 수 있는 관련 리소스가 나와 있습니다.

주제

- [Amazon CodeGuru Reviewer](#)
- [Amazon CodeWhisperer](#)
- [Amazon SageMaker](#)
- [AWS App Runner](#)
- [AWS CloudFormation](#)
- [AWS CodePipeline](#)
- [AWS CodeStar](#)
- [서비스 카탈로그](#)
- [AWS Proton](#)

Amazon CodeGuru Reviewer

[CodeGuru Reviewer](#)는 리포지토리 코드를 모니터링하는 서비스입니다. 연결을 사용하여 검토할 코드가 있는 서드 파티 리포지토리를 연결할 수 있습니다. 코드 개선 권장 사항을 생성하기 위해 GitHub 리포지토리의 소스 코드를 모니터링하도록 CodeGuru Reviewer를 구성하는 방법을 알아보는 자습서는 Amazon CodeGuru Reviewer 사용 설명서의 [자습서: GitHub 리포지토리에서 소스 코드 모니터링](#)을 참조하세요.

Amazon CodeWhisperer

[Amazon CodeWhisperer](#)는 리포지토리 코드를 검토하는 서비스입니다. CodeWhisperer는 실시간으로 코드를 검토하고 코드 권장 사항을 제공합니다. 연결을 사용하여 데이터 소스에 액세스하는 사용자 지정을 CodeWhisperer에서 구성하는 단계는 Amazon CodeWhisperer 사용 설명서의 [사용자 지정 생성](#)을 참조하세요.

Amazon SageMaker

[Amazon SageMaker](#)는 기계 학습 언어 모델을 구축, 훈련 및 배포하는 서비스입니다. GitHub 리포지토리에 대한 연결을 구성하는 자습서는 Amazon SageMaker 개발자 안내서의 [서드 파티 Git 리포지토리를 사용한 SageMaker MLOps 프로젝트 안내](#)를 참조하세요.

AWS App Runner

[AWS App Runner](#)는 소스 코드 또는 컨테이너 이미지에서 AWS 클라우드의 확장 가능하고 안전한 웹 애플리케이션으로 직접 빠르고 간단하며 비용 효율적으로 배포할 수 있도록 지원하는 서비스입니다. App Runner 자동 통합 및 전달 파이프라인을 사용하여 리포지토리에서 애플리케이션 코드를 배포할 수 있습니다. 연결을 사용하여 프라이빗 GitHub 리포지토리에서 App Runner 서비스에 소스 코드를 배포할 수 있습니다. 자세한 내용은 AWS App Runner 개발자 안내서의 [소스 코드 리포지토리 공급자](#)를 참조하세요.

AWS CloudFormation

[AWS CloudFormation](#)은 AWS 리소스를 모델링하고 설정하여 리소스 관리 시간을 줄이고 AWS에서 실행되는 애플리케이션에 더 많은 시간을 사용하도록 해 주는 서비스입니다. 필요한 모든 AWS 리소스(예: Amazon EC2 인스턴스 또는 Amazon RDS DB 인스턴스)를 설명하는 템플릿을 생성하면 CloudFormation이 해당 리소스의 프로비저닝과 구성을 담당합니다. 자세한 내용을 알아보려면 CloudFormation 명령줄 인터페이스 사용 설명서의 [CloudFormation 익스텐션을 게시하기 위해 계정 등록](#)을 참조하세요.

AWS CodePipeline

[CodePipeline](#)은 소프트웨어 릴리스에 필요한 단계를 모델링, 시각화 및 자동화하는 데 사용할 수 있는 지속적 전달 서비스입니다. 연결을 사용하여 CodePipeline 소스 작업에 대한 서드 파티 리포지토리를 구성할 수 있습니다.

자세히 알아보기:

- CodeStarSourceConnection 작업에 대한 자세한 내용은 CodePipeline 작업 구성 참조 페이지를 참조하세요. 구성 파라미터와 예제 JSON/YAML 코드 조각은 AWS CodePipeline 사용 설명서의 [CodeStarSourceConnection](#)을 참조하세요.
- 타사 소스 리포지토리에서 파이프라인을 생성하는 시작하기 자습서를 보려면 [연결 시작하기](#) 섹션을 참조하세요.

AWS CodeStar

[AWS CodeStar](#)는 AWS에서 소프트웨어 개발 프로젝트를 생성, 관리, 작업하기 위한 클라우드 기반 서비스입니다. AWS CodeStar 프로젝트를 통해 AWS에서 애플리케이션을 빠르게 개발, 빌드, 배포할 수 있습니다. 연결을 사용하여 AWS CodeStar 프로젝트의 파이프라인에 대한 서드 파티 리포지토리를 구성할 수 있습니다. GitHub 리포지토리에 연결하여 AWS CodeStar 프로젝트를 생성하는 자습서는 [AWS CodeStar 사용 설명서의 리포지토리에 대한 링크 생성](#)을 참조하세요.

서비스 카탈로그

[Service Catalog](#)를 사용하는 조직은 AWS에서 사용이 승인된 제품을 생성하고 관리할 수 있습니다.

GitHub, GitHub Enterprise 또는 BitBucket과 같은 외부 리포지토리 제공업체와 AWS 계정 간의 연결을 승인하면 해당 연결을 사용하여 Service Catalog 제품을 서드 파티 리포지토리를 통해 관리되는 템플릿 파일과 동기화할 수 있습니다.

자세한 내용은 Service Catalog 사용 설명서의 [GitHub, GitHub Enterprise 또는 Bitbucket의 템플릿 파일과 Service Catalog 제품 동기화](#)를 참조하세요.

AWS Proton

[AWS Proton](#)는 클라우드 인프라에 배포하기 위한 클라우드 기반 서비스입니다. 연결을 사용하여 AWS Proton용 템플릿의 리소스에 대한 타사 리포지토리로 연결되는 링크를 생성할 수 있습니다. 자세한 내용은 AWS Proton 사용 설명서의 [리포지토리에 대한 링크 생성](#)을 참조하세요.

연결 설정

개발자 도구 콘솔에서 연결 기능을 생성하고 사용하도록 설정하려면 이 섹션의 작업을 완료합니다.

주제

- [AWS에 가입](#)
- [연결 생성 권한이 있는 정책 생성 및 적용](#)

AWS에 가입

AWS 계정에 등록

AWS 계정이 없는 경우 다음 절차에 따라 계정을 생성합니다.

AWS 계정에 가입하려면

1. <https://portal.aws.amazon.com/billing/signup>을 엽니다.
2. 온라인 지시 사항을 따릅니다.

가입 절차 중 전화를 받고 전화 키패드로 확인 코드를 입력하는 과정이 있습니다.

AWS 계정에 가입하면 AWS 계정 루트 사용자 항목이 생성됩니다. 루트 사용자에게 계정의 모든 AWS 서비스 및 리소스에 대한 액세스 권한이 있습니다. 보안 모범 사례는 [관리 사용자에게 관리자 액세스 권한을 할당하고](#), 루트 사용자만 [루트 사용자 액세스 권한이 필요한 작업을 수행하는 것](#)입니다.

가입 프로세스가 완료되면 AWS가 확인 이메일을 전송합니다. 언제든지 <https://aws.amazon.com/>으로 이동하고 내 계정을 선택하여 현재 계정 활동을 보고 계정을 관리할 수 있습니다.

관리 사용자 생성

AWS 계정에 가입하고, AWS 계정 루트 사용자에게 보안 조치를 한 다음, AWS IAM Identity Center를 활성화하여 일상적인 작업에 루트 사용자를 사용하지 않도록 관리 사용자를 생성합니다.

귀하의 AWS 계정 루트 사용자 보호

1. 루트 사용자를 선택하고 AWS 계정 이메일 주소를 입력하여 [AWS Management Console](#)에 계정 소유자로 로그인합니다. 다음 페이지에서 암호를 입력합니다.

루트 사용자를 사용하여 로그인하는 데 도움이 필요하다면 AWS 로그인 사용 설명서의 [루트 사용자 로 로그인](#)을 참조하세요.

2. 루트 사용자에게 다중 인증(MFA)을 활성화합니다.

지침은 IAM 사용 설명서의 [AWS 계정 루트 사용자용 가상 MFA 디바이스 활성화\(콘솔\)](#) 섹션을 참조하세요.

관리 사용자 생성

1. IAM Identity Center를 활성화합니다.

지침은 AWS IAM Identity Center 사용 설명서의 [AWS IAM Identity Center 활성화](#)를 참조하세요.

2. IAM Identity Center에서 관리 사용자에게 관리 액세스 권한을 부여합니다.

IAM Identity Center 디렉터리를 ID 소스로 사용하는 방법에 대한 자습서는 AWS IAM Identity Center 사용 설명서의 [Configure user access with the default IAM Identity Center 디렉터리](#)를 참조하세요.

관리 사용자로 로그인

- IAM 자격 증명 센터 사용자로 로그인하려면 IAM 자격 증명 센터 사용자를 생성할 때 이메일 주소로 전송된 로그인 URL을 사용합니다.

IAM Identity Center 사용자로 로그인하는 데 도움이 필요한 경우 AWS 로그인 사용 설명서의 [AWS 액세스 포털에 로그인](#)을 참조하세요.

연결 생성 권한이 있는 정책 생성 및 적용

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. 시작하기를 선택합니다.
3. 페이지 상단에서 정책 생성을 선택합니다.
4. 정책 편집기 섹션에서 JSON 옵션을 선택합니다.
5. 다음 JSON 정책 문서를 입력합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codestar-connections:CreateConnection",
        "codestar-connections>DeleteConnection",
        "codestar-connections:GetConnection",
        "codestar-connections:ListConnections",
        "codestar-connections:GetInstallationUrl",
        "codestar-connections:GetIndividualAccessToken",

```

```

        "codestar-connections:ListInstallationTargets",
        "codestar-connections:StartOAuthHandshake",
        "codestar-connections:UpdateConnectionInstallation",
        "codestar-connections:UseConnection"
    ],
    "Resource": [
        "*"
    ]
}
]
}

```

6. 다음을 선택합니다.

Note

언제든지 시각적 편집기 옵션과 JSON 편집기 옵션 간에 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 재구성](#)을 참조하십시오.

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.

연결 시작하기

연결을 시작하는 가장 쉬운 방법은 서드 파티 소스 리포지토리를 AWS 리소스에 연결하는 연결을 설정하는 것입니다. CodeCommit과 같은 AWS 소스에 파이프라인을 연결하려면, 파이프라인을 소스 작업으로 연결합니다. 그러나 외부 리포지토리가 있는 경우 리포지토리를 파이프라인과 연결하는 연결을 생성해야 합니다. 이 자습서에서는 Bitbucket 리포지토리 및 파이프라인과의 연결을 설정합니다.

이 섹션에서는 다음에 대한 연결을 사용합니다.

- **AWS CodePipeline:** 이 단계에서는 Bitbucket 리포지토리를 파이프라인 소스로 사용하여 파이프라인을 생성합니다.
- **[Amazon CodeGuru Reviewer](#):** 다음으로 Bitbucket 리포지토리를 CodeGuru Reviewer의 피드백 및 분석 도구에 연결합니다.

주제

- [필수 조건](#)
- [1단계: 소스 파일 편집](#)
- [2단계: 파이프라인 생성](#)
- [3단계: 리포지토리를 CodeGuru Reviewer와 연결](#)

필수 조건

시작하기 전에 [설정](#)의 단계를 완료해야 합니다. 또한 AWS 서비스에 연결하고 그 연결을 통해 인증을 관리하도록 허용할 서드 파티 소스 리포지토리가 필요합니다. 예를 들어 소스 리포지토리와 통합되는 AWS 서비스에 Bitbucket 리포지토리를 연결할 수 있습니다.

- Bitbucket 계정으로 Bitbucket 리포지토리를 생성합니다.
- Bitbucket 자격 증명을 준비합니다. AWS Management Console을 사용하여 연결을 설정하는 경우 Bitbucket 자격 증명으로 로그인하라는 메시지가 표시됩니다.

1단계: 소스 파일 편집

Bitbucket 리포지토리를 생성할 때 기본 README.md 파일이 포함되는데, 이 파일을 편집합니다.

1. Bitbucket 리포지토리에 로그인하고 [소스(Source)]를 선택합니다.
2. README.md 파일을 선택하고 페이지 상단에서 [편집(Edit)]을 선택합니다. 기존 텍스트를 삭제하고 다음 텍스트를 추가합니다.

```
This is a Bitbucket repository!
```

3. 커밋을 선택합니다.

README.md 파일이 리포지토리의 루트 수준에 있는지 확인합니다.

2단계: 파이프라인 생성

이 단원에서는 다음 작업을 통해 파이프라인을 생성합니다.

- Bitbucket 리포지토리 및 작업에 대한 연결이 있는 소스 단계입니다.
- AWS CodeBuild 빌드 작업이 있는 빌드 단계입니다.

마법사를 사용하여 파이프라인을 생성하려면

1. <https://console.aws.amazon.com/codepipeline/>에서 CodePipeline 콘솔에 로그인합니다.
2. Welcome(시작) 페이지, 시작하기 페이지 또는 Pipelines(파이프라인) 페이지에서 파이프라인 생성을 선택합니다.
3. 1단계: 파이프라인 설정 선택의 파이프라인 이름에 **MyBitbucketPipeline**을 입력합니다.
4. Service role(서비스 역할)에서 New service role(새 서비스 역할)을 선택합니다.

Note

역할을 생성하지 않고 기존 CodePipeline 서비스 역할을 사용할 경우 서비스 역할 정책에 대한 `codestar-connections:UseConnection` IAM 권한을 추가해야 합니다. CodePipeline 서비스 역할에 대한 지침은 [CodePipeline 서비스 역할에 권한 추가](#)를 참조하세요.

5. 고급 설정에서 기본값을 그대로 둡니다. [아티팩트 스토어(Artifact store)]에서 [기본 위치(Default location)]를 선택하여 파이프라인에 대해 선택한 리전의 파이프라인에 대해 기본값으로 지정된 Amazon S3 아티팩트 버킷과 같은 기본 아티팩트 스토어를 사용합니다.

Note

이는 소스 코드에 대한 소스 버킷이 아닙니다. 이 파이프라인은 아티팩트 스토어입니다. S3 버킷과 같은 개별 아티팩트 스토어는 각 파이프라인에 필요합니다.

다음(Next)을 선택합니다.

6. [2단계: 소스 단계 추가(Step 2: Add source stage)] 페이지에서 소스 단계를 추가합니다.
 - a. [소스 공급자(Source provider)]에서 [Bitbucket]을 선택합니다.
 - b. [연결(Connection)]에서 [Bitbucket에 연결(Connect to Bitbucket)]을 선택합니다.
 - c. [Bitbucket에 연결(Connect to Bitbucket)] 페이지에서 [연결 이름(Connection name)]에 생성하려는 연결의 이름을 입력합니다. 이후에 이 이름을 통해 이 연결을 식별할 수 있습니다.

[Bitbucket 앱(Bitbucket apps)]에서 [새 앱 설치(Install a new app)]를 선택합니다.
 - d. 앱 설치 페이지에서 AWS CodeStar 앱이 Bitbucket 계정에 연결하려고 한다는 메시지가 표시 됩니다. [Grant access(액세스 권한 부여)]를 선택합니다. 연결을 승인하면 Bitbucket의 리포지토리가 검색되어 AWS 리소스와 연결할 리포지토리를 선택할 수 있습니다.

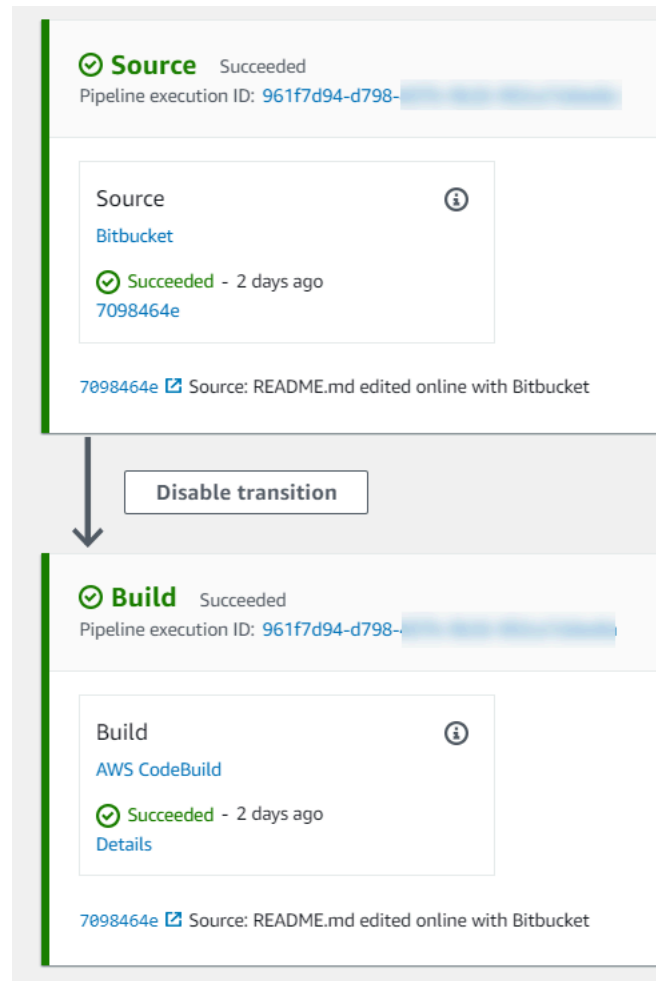
- e. 새 설치의 연결 ID가 표시됩니다. Complete connection(연결 완료)을 선택합니다. CodePipeline 콘솔로 돌아가게 됩니다.
- f. [리포지토리 이름(Repository name)]에서 Bitbucket 리포지토리의 이름을 선택합니다.
- g. [브랜치 이름(Branch name)]에서 리포지토리의 브랜치를 선택합니다.
- h. 소스 코드 변경 시 파이프라인 시작 옵션을 선택합니다.
- i. 출력 아티팩트 형식에서 다음 중 하나를 선택합니다. CodePipeline 기본값
 - 파이프라인의 아티팩트에 기본 zip 형식을 사용하려면 CodePipeline 기본값을 선택합니다.
 - 파이프라인의 아티팩트 리포지토리에 대한 Git 메타데이터를 포함하려면 전체 복제를 선택합니다. 이는 CodeBuild 작업에만 지원됩니다.

다음(Next)을 선택합니다.

7. Add build stage(빌드 스테이지 추가)에서 빌드 스테이지를 추가합니다.
 - a. 빌드 공급자에서 AWS CodeBuild를 선택합니다. 리전이 파이프라인 리전으로 기본 설정되도록 합니다.
 - b. 프로젝트 만들기를 선택합니다.
 - c. 프로젝트 이름에 이 빌드 프로젝트의 이름을 입력합니다.
 - d. 환경 이미지에서 이미지 관리를 선택합니다. [Operating system]에서 [Ubuntu]를 선택합니다.
 - e. 실행 시간에서 표준을 선택합니다. 이미지에서 aws/codebuild/standard:5.0을 선택합니다.
 - f. 서비스 역할에서 New service role(새 서비스 역할)을 선택합니다.
 - g. [Buildspec]의 [빌드 사양(Build specifications)]에서 [빌드 명령 삽입(Insert build commands)]을 선택합니다. [편집기로 전환(Switch to editor)]을 선택하고 [빌드 명령(Build commands)]에 다음을 붙여 넣습니다.

```
version: 0.2

phases:
  install:
    #If you use the Ubuntu standard image 2.0 or later, you must specify
    runtime-versions.
    #If you specify runtime-versions and use an image other than Ubuntu
    standard image 2.0, the build fails.
    runtime-versions:
      nodejs: 12
      # name: version
```

11. 성공적으로 완료된 빌드 단계에서 [세부 정보(Details)]를 선택합니다.

[실행 세부 정보(Execution details)]에서 CodeBuild 빌드 출력을 봅니다. 이 명령은 다음과 같이 README.md 파일 내용을 출력합니다.

This is a Bitbucket repository!

```

35 [Container] 2020/06/05 19:14:51 Running command cat README.md
36 This is a Bitbucket repository!
37 [Container] 2020/06/05 19:14:51 Phase complete: PRE_BUILD State: SUCCEEDED
38 [Container] 2020/06/05 19:14:51 Phase context status code: Message:
39 [Container] 2020/06/05 19:14:51 Entering phase BUILD
40 [Container] 2020/06/05 19:14:51 Phase complete: BUILD State: SUCCEEDED
41 [Container] 2020/06/05 19:14:51 Phase context status code: Message:
42 [Container] 2020/06/05 19:14:51 Entering phase POST_BUILD
43 [Container] 2020/06/05 19:14:51 Phase complete: POST_BUILD State: SUCCEEDED
44 [Container] 2020/06/05 19:14:51 Phase context status code: Message:

```

3단계: 리포지토리를 CodeGuru Reviewer와 연결

연결을 생성한 후 해당 연결을 동일한 계정의 모든 AWS 리소스에 사용할 수 있습니다. 예를 들어 파이프라인의 CodePipeline 소스 작업과 CodeGuru Reviewer의 리포지토리 커밋 분석에 동일한 Bitbucket 연결을 사용할 수 있습니다.

1. CodeGuru Reviewer 콘솔에 로그인합니다.
2. [CodeGuru Reviewer]를 선택하고 [리포지토리에 연결(Associate repository)]을 선택합니다.

한 페이지로 구성된 마법사가 열립니다.

3. [소스 공급자 선택(Select source provider)]을 선택하고 [Bitbucket]을 선택합니다.
4. Bitbucket에 연결(AWS CodeStar 연결 사용)에서 파이프라인용으로 생성한 연결을 선택합니다.
5. [리포지토리 위치(Repository location)]에서 Bitbucket 리포지토리의 이름을 선택하고 [연결(Associate)]을 선택합니다.

이어서 코드 검토를 설정할 수 있습니다. 자세한 내용은 Amazon CodeGuru Reviewer 사용 설명서에서 [Bitbucket에 연결하여 리포지토리를 CodeGuru Reviewer와 연결](#)을 참조하세요.

연결 관련 작업

연결은 AWS 리소스를 외부 코드 리포지토리에 연결하는 데 사용되는 구성입니다. 각 연결은 Bitbucket과 같은 타사 리포지토리에 AWS CodePipeline 연결하는 등 서비스에 제공할 수 있는 리소스입니다. 예를 들어 타사 코드 저장소의 코드가 변경될 때 파이프라인이 CodePipeline 트리거되도록 연결을 추가할 수 있습니다. GitHub Enterprise Server와 같은 설치된 제공자 유형에 AWS 리소스를 연결할 수도 있습니다.

설치된 제공자 유형(예: GitHub Enterprise Server)에 대한 연결을 생성하려는 경우 콘솔에서 호스트를 자동으로 생성합니다. 호스트는 공급자가 설치된 서버를 나타내기 위해 생성하는 리소스입니다. 자세한 정보는 [호스트 작업](#)을 참조하세요.

연결을 생성할 때는 콘솔의 마법사를 사용하여 타사 공급자와 함께 AWS CodeStar 앱을 설치하고 새 연결에 연결합니다. 앱을 이미 설치한 경우 AWS CodeStar 앱을 사용할 수 있습니다.

Note

유럽(밀라노) AWS 리전에서 연결을 사용하려면 다음을 수행해야 합니다.

1. 리전별 앱 설치

2. 리전 활성화

이 리전별 앱은 유럽(밀라노) 리전 내 연결을 지원합니다. 서드 파티 제공업체 사이트에 게시되며 다른 리전의 연결을 지원하는 기존 앱과는 별개입니다. 이 앱을 설치하면 서드 파티 제공업체가 이 리전에서 서비스와 데이터를 공유할 수 있는 권한을 부여하게 되며 앱을 제거하여 언제든지 권한을 취소할 수 있습니다.

리전을 활성화하지 않으면 서비스에서 데이터를 처리하거나 저장하지 않습니다. 이 리전을 활성화하면 데이터를 처리하고 저장할 수 있는 권한이 서비스에 부여됩니다.

리전이 활성화되지 않았더라도 리전별 앱이 설치된 상태로 유지되면 서드 파티 제공업체가 서비스와 데이터를 공유할 수 있으므로 리전을 비활성화한 후에는 앱을 제거해야 합니다. 자세한 내용은 [리전 활성화](#)를 참조하세요.

연결에 대한 자세한 내용은 [AWS CodeStar 연결 API 참조](#)를 참조하십시오. Bitbucket의 CodePipeline 소스 작업에 대한 자세한 내용은 AWS CodePipeline 사용 설명서를 참조하십시오 [CodestarConnectionSource](#).

AWS CodeStar 연결을 사용하는 데 필요한 권한이 포함된 정책을 생성하거나 IAM 사용자 또는 역할에 연결하려면 을 참조하십시오. AWS Identity and Access Management [AWS CodeConnections 권한 참조](#) CodePipeline 서비스 역할이 생성된 시기에 따라 AWS CodeStar 연결을 지원하도록 서비스 역할 권한을 업데이트해야 할 수 있습니다. 지침은 AWS CodePipeline 사용 설명서에서 [서비스 역할 업데이트](#)를 참조하세요.

주제

- [연결 생성](#)
- [Bitbucket에 대한 연결 생성](#)
- [에 대한 연결을 생성하십시오. GitHub](#)
- [GitHub 엔터프라이즈 서버에 대한 연결 생성](#)
- [에 대한 연결 생성 GitLab](#)
- [GitLab 자체 관리형 연결을 생성하세요.](#)
- [보류 중인 연결 업데이트](#)
- [연결 나열](#)
- [연결 삭제](#)
- [연결 리소스 태그 지정](#)
- [연결 세부 정보 보기](#)

연결 생성

다음 서드 파티 공급자 유형에 대한 연결을 생성할 수 있습니다.

- Bitbucket에 대한 연결을 생성하려면 [Bitbucket에 대한 연결 생성](#) 섹션을 참조하세요.
- GitHub 엔터프라이즈 클라우드에 대한 연결을 생성하려면 GitHub 을 참조하십시오 [에 대한 연결을 생성하십시오. GitHub](#).
- 호스트 리소스 생성을 포함하여 GitHub Enterprise Server에 대한 연결을 생성하려면 을 참조하십시오 [GitHub 엔터프라이즈 서버에 대한 연결 생성](#).
- 연결을 GitLab 생성하려면 을 참조하십시오 [에 대한 연결 생성 GitLab](#).

Bitbucket에 대한 연결 생성

AWS Management Console 또는 AWS Command Line Interface (AWS CLI) 를 사용하여 bitbucket.org 에서 호스팅되는 저장소에 대한 연결을 만들 수 있습니다.

시작하기 전:

- Bitbucket 계정이 이미 생성되어 있어야 합니다.
- bitbucket.org에 코드 리포지토리가 이미 생성되어 있어야 합니다.

Note

Bitbucket Cloud 리포지토리에 대한 연결을 생성할 수 있습니다. Bitbucket Server와 같은 설치된 Bitbucket 공급자 유형은 지원되지 않습니다. [AWS CodeStar 연결 지원 제공자 및 버전](#) 섹션을 참조하십시오.

Note

연결은 연결을 만드는 데 사용된 계정이 소유한 리포지토리에 대한 액세스 권한만 제공합니다. Bitbucket WorkSpace에 애플리케이션을 설치하는 경우 관리자 WorkSpace 권한이 필요합니다. 그렇지 않은 경우 앱 설치 옵션이 표시되지 않습니다.

주제

- [Bitbucket에 대한 연결 생성\(콘솔\)](#)

- [Bitbucket에 대한 연결 생성\(CLI\)](#)

Bitbucket에 대한 연결 생성(콘솔)

1단계: 연결 생성

1. 에 로그인하고 에서 개발자 AWS Management Console도구 콘솔을 엽니다. AWS <https://console.aws.amazon.com/codesuite/settings/connections>
2. [설정(Settings)] > [연결(Connections)]을 선택한 다음 [연결 생성(Create connection)]을 선택합니다.
3. Bitbucket 리포지토리에 대한 연결을 생성하려면 [공급자 선택(Select a provider)]에서 [Bitbucket]을 선택합니다. [연결 이름(Connection name)]에 생성하려는 연결의 이름을 입력합니다. [Bitbucket에 연결(Connect to Bitbucket)]을 선택하고 2단계로 진행합니다.

The screenshot shows the 'Create a connection' page in the AWS Management Console. The breadcrumb trail is 'Developer Tools > Connections > Create connection'. The main heading is 'Create a connection' with an 'Info' link. Below this, there are two sections: 'Select a provider' and 'Create Bitbucket connection'. In the 'Select a provider' section, there are three radio button options: 'Bitbucket' (which is selected), 'GitHub', and 'GitHub Enterprise Server'. In the 'Create Bitbucket connection' section, there is a text input field labeled 'Connection name'. At the bottom right of the form, there is an orange button labeled 'Connect to Bitbucket'.

2단계: Bitbucket에 연결

1. [Bitbucket에 연결(Connect to Bitbucket)] 설정 페이지에 연결 이름이 표시됩니다.

[Bitbucket 앱(Bitbucket apps)]에서 앱 설치를 선택하거나 [새 앱 설치(Install a new app)]을 선택하여 앱을 새로 만듭니다.

Note

각 Bitbucket WorkSpace 또는 계정마다 앱을 한 번만 설치합니다. Bitbucket 앱을 이미 설치한 경우 앱을 선택하고 이 섹션의 마지막 단계로 넘어갑니다.

2. Bitbucket의 로그인 페이지가 표시되면 자격 증명으로 로그인한 다음 계속하도록 선택합니다.
3. 앱 설치 페이지에서 앱이 Bitbucket 계정에 연결을 시도하고 있다는 메시지가 표시됩니다. AWS CodeStar

Bitbucket WorkSpace를 사용하는 경우 Authorization for(권한 부여 대상) 옵션을 WorkSpace로 변경합니다. 관리자 권한이 있는 WorkSpace만 표시됩니다.

[Grant access(액세스 권한 부여)]를 선택합니다.



AWS CodeStar requests access

This app is hosted at <https://codestar-connections.webhooks.aws>

- Read your account information
- Read your repositories and their pull requests
- Administer your repositories
- Read and modify your repositories

Authorize for

Allow AWS CodeStar to do this?

This 3rd party vendor has not provided a privacy policy or terms of use.

Atlassian's Privacy Policy is not applicable to the use of this App.

[Grant access](#) [Cancel](#)

4. [Bitbucket 앱(Bitbucket apps)]을 선택하면 새 설치의 연결 ID가 표시됩니다. 연결을 선택합니다. 생성된 연결이 연결 목록에 표시됩니다.

Connect to Bitbucket

Bitbucket connection settings [Info](#)

Connection name

Bitbucket apps
Bitbucket apps create a link for your connection with Bitbucket. To start, install a new app and save this connection.

or

Bitbucket에 대한 연결 생성(CLI)

AWS Command Line Interface (AWS CLI) 를 사용하여 연결을 만들 수 있습니다.

이렇게 하려면 create-connection 명령을 사용합니다.

Important

OR를 통해 생성된 AWS CloudFormation 연결은 기본적으로 PENDING 상태입니다. AWS CLI CLI를 사용하여 연결을 생성한 후 콘솔을 사용하여 연결을 편집하여 상태를 설정합니다. AWS CloudFormationAVAILABLE

Bitbucket에 대한 연결을 생성하려면

1. 터미널(Linux, macOS, Unix) 또는 명령 프롬프트(Windows)를 엽니다. AWS CLI 를 사용하여 연결을 --connection-name 위한 --provider-type 및 을 지정하여 create-connection 명령을 실행합니다. 이 예제에서 타사 공급자 이름은 Bitbucket이고 지정된 연결 이름은 MyConnection입니다.

```
aws codestar-connections create-connection --provider-type Bitbucket --connection-name MyConnection
```

이 명령이 제대로 실행되면 다음과 비슷한 연결 ARN 정보가 반환됩니다.

```
{
  "ConnectionArn": "arn:aws:codestar-connections:us-west-2:account_id:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f"
}
```

2. 콘솔을 사용하여 연결을 완료합니다. 자세한 정보는 [보류 중인 연결 업데이트](#)을 참조하세요.

에 대한 연결을 생성하십시오. GitHub

AWS Management Console 또는 AWS Command Line Interface (AWS CLI) 를 사용하여 연결을 만들 수 GitHub 있습니다.

시작하기 전:

- 에 이미 계정을 생성했어야 합니다 GitHub.

- 서드 파티 코드 리포지토리가 이미 생성되어 있어야 합니다.

Note

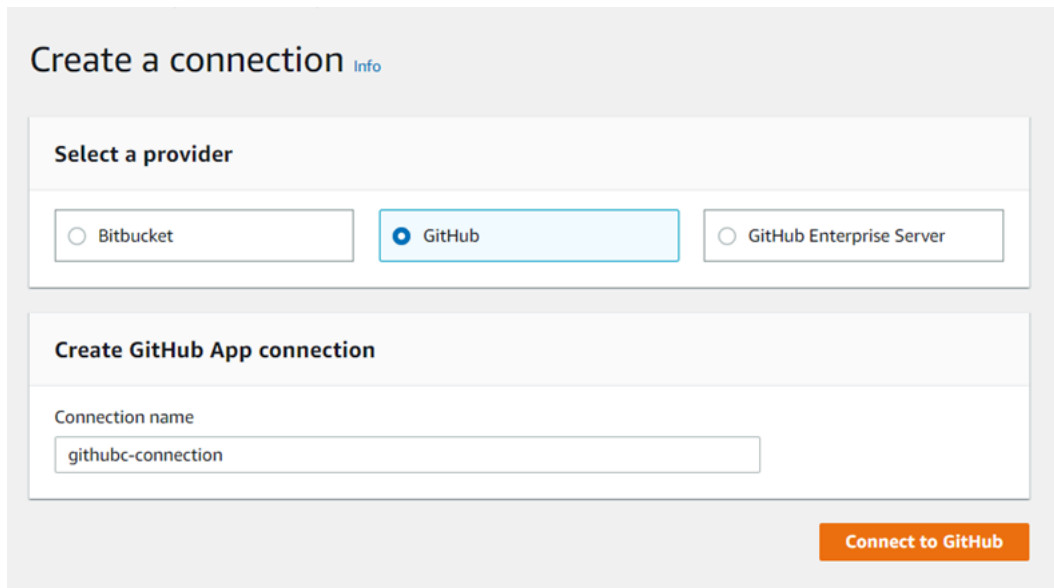
연결을 만들려면 GitHub 조직 소유자여야 합니다. 조직 소속이 아닌 리포지토리의 경우 리포지토리 소유자여야 합니다.

주제

- [GitHub\(콘솔\)에 대한 연결 만들기](#)
- [GitHub\(CLI\)에 대한 연결 생성](#)

GitHub(콘솔)에 대한 연결 만들기

1. [에 AWS Management Console](https://console.aws.amazon.com/codesuite/settings/connections)로그인하고 [에서 개발자 도구 콘솔을 엽니다](https://console.aws.amazon.com/codesuite/settings/connections)<https://console.aws.amazon.com/codesuite/settings/connections>.
2. [설정(Settings)] > [연결(Connections)]을 선택한 다음 [연결 생성(Create connection)]을 선택합니다.
3. GitHub 또는 GitHub Enterprise Cloud 리포지토리에 대한 연결을 생성하려면 공급자 선택에서 [GitHub](#)를 선택합니다. [연결 이름(Connection name)]에 생성하려는 연결의 이름을 입력합니다. [Connect to GitHub](#) (연결 대상)를 선택하고 2단계로 진행합니다.



Create a connection Info

Select a provider

Bitbucket GitHub GitHub Enterprise Server

Create GitHub App connection

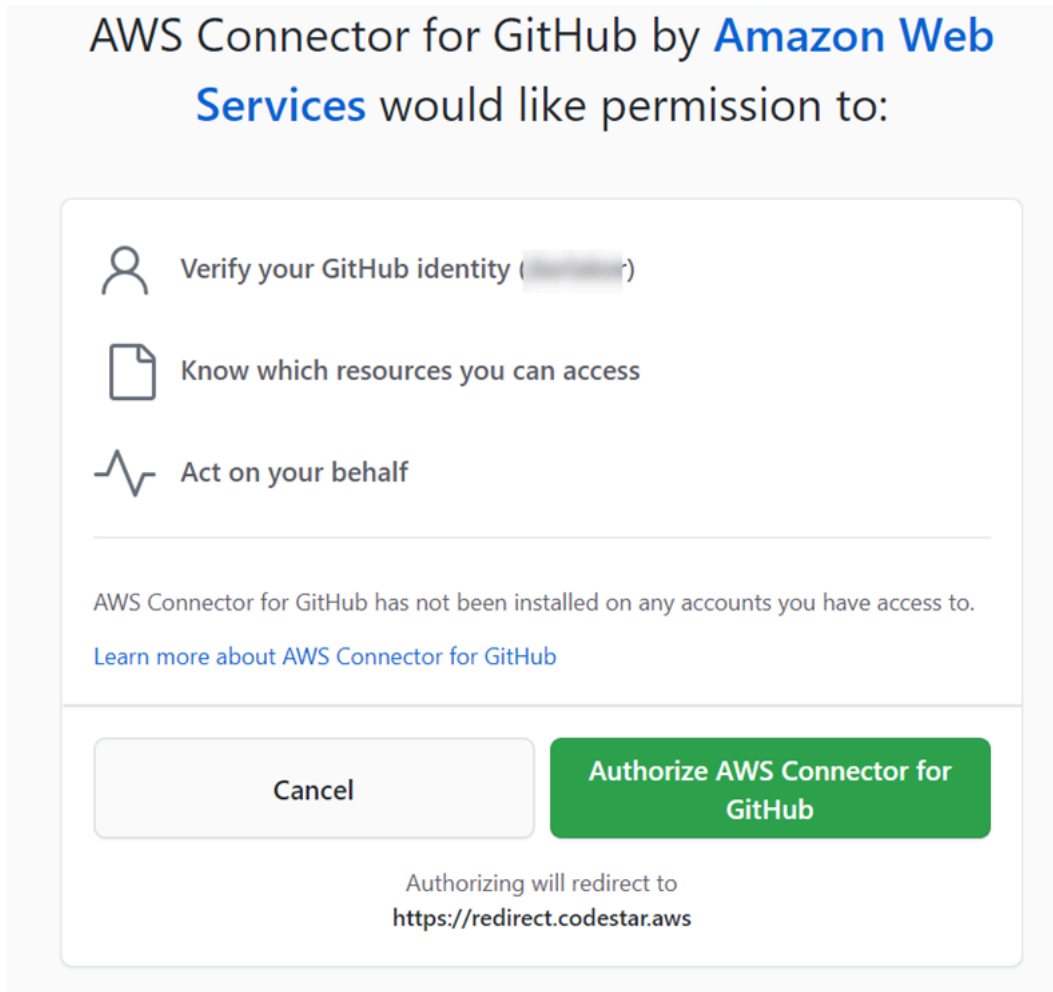
Connection name

githubc-connection

Connect to GitHub

연결을 만들려면 GitHub

1. GitHub 연결 설정에서 연결 이름은 연결 이름에 표시됩니다. [Connect to GitHub]를 선택합니다. 액세스 요청 페이지가 표시됩니다.



2. AWS 커넥터 인증을 선택합니다. GitHub 연결 페이지에 GitHub 앱 필드가 표시되고 표시됩니다.

Connect to GitHub

GitHub connection settings [Info](#)

Connection name

GitHub Apps
GitHub Apps create a link for your connection with GitHub. To start, install a new app and save this connection.

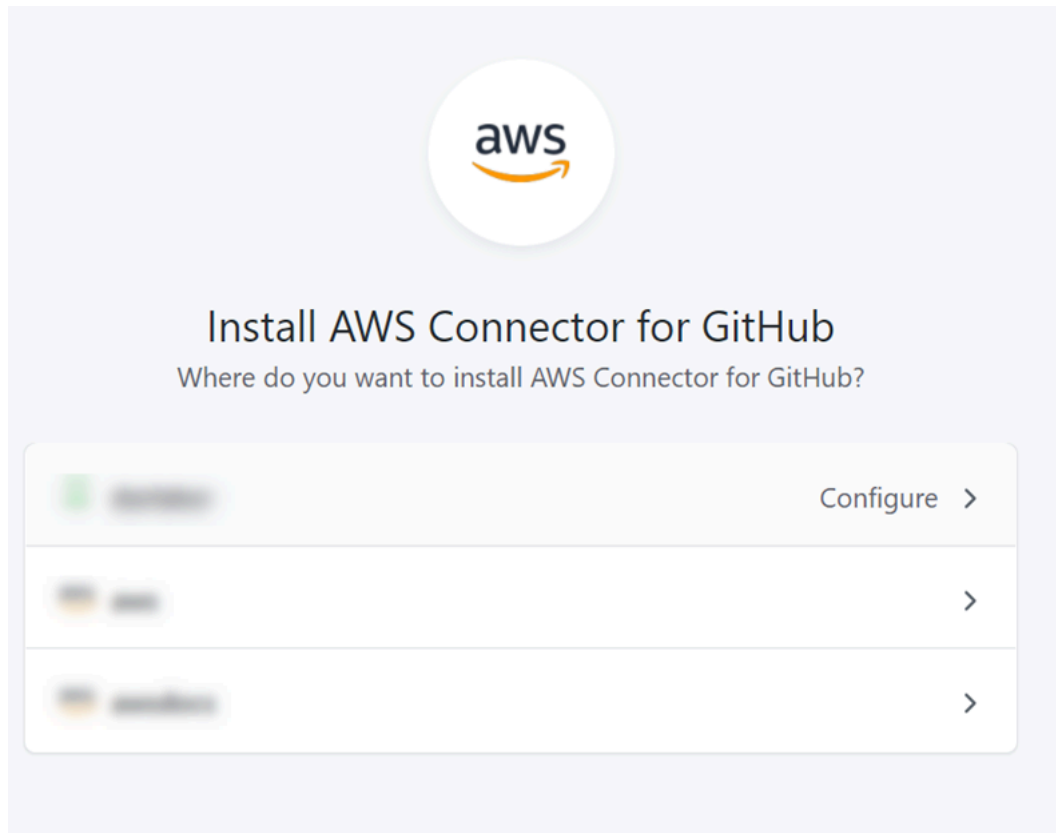
 or

3. 앱에서 GitHub 앱 설치를 선택하거나 새 앱 설치를 선택하여 앱을 생성합니다.

Note

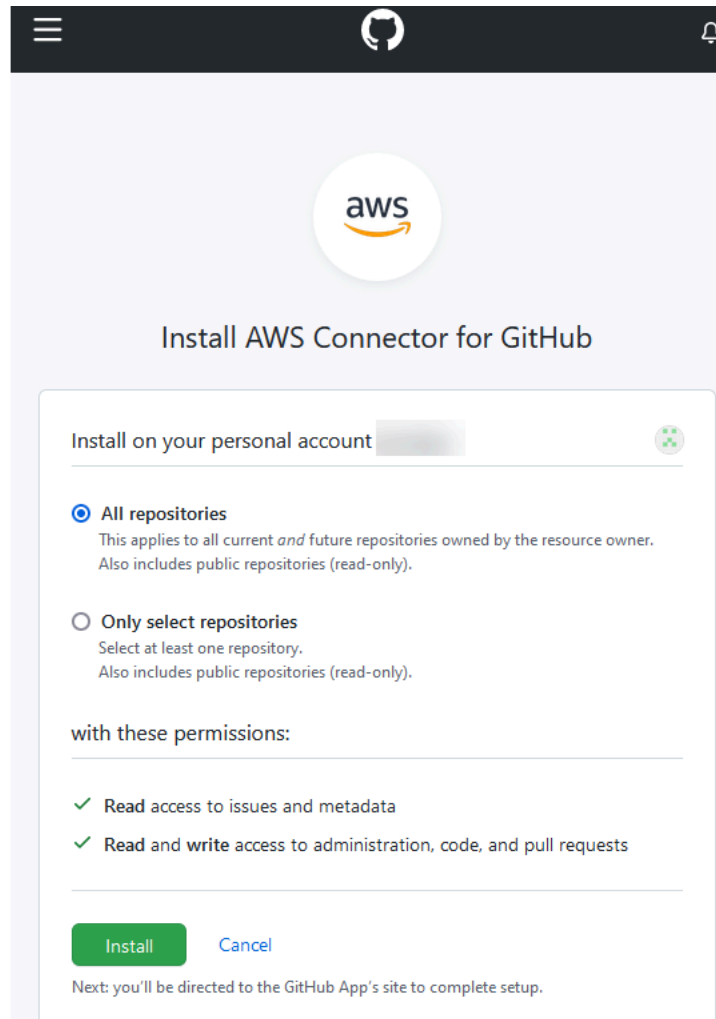
특정 공급자에 대한 모든 연결에 대해 하나의 앱을 설치합니다. GitHub 앱용 AWS 커넥터를 이미 설치한 경우 앱을 선택하고 이 단계를 건너뛰십시오.

4. AWS 커넥터 설치 대상 GitHub 페이지에서 앱을 설치할 계정을 선택합니다.

**Note**

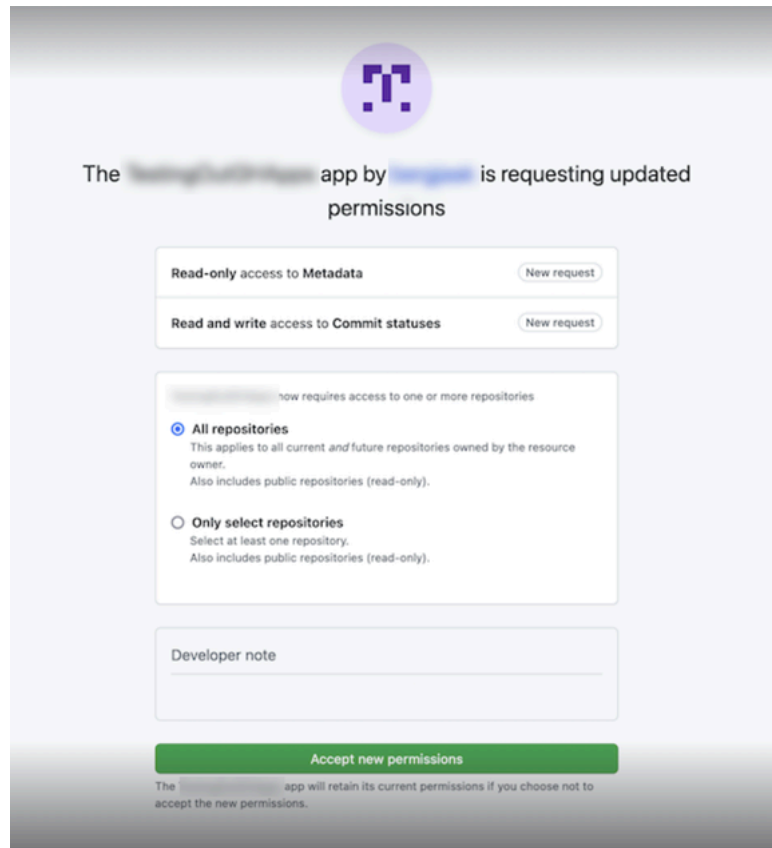
앱은 GitHub 계정당 한 번만 설치합니다. 이전에 앱을 설치한 경우 구성을 선택하여 앱 설치의 수정 페이지로 이동하거나 뒤로 버튼을 사용하여 콘솔로 돌아갈 수 있습니다.

5. AWS 커넥터 설치 GitHub 페이지에서 기본값을 그대로 두고 설치를 선택합니다.



이 단계가 끝나면 업데이트된 권한 페이지가 에 표시될 수 GitHub 있습니다.

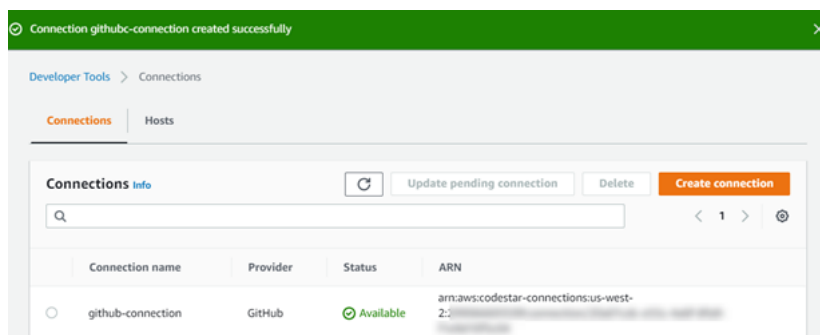
6. AWS Connector for GitHub app에 대한 권한이 업데이트되었음을 보여주는 페이지가 표시되면 새 권한 수락을 선택합니다.



7. Connect to GitHub 페이지로 돌아갑니다. 새로 설치한 연결 ID가 GitHub앱에 표시됩니다. 연결을 선택합니다.

생성한 연결 보기

- 생성된 연결이 연결 목록에 표시됩니다.



GitHub(CLI) 에 대한 연결 생성

AWS Command Line Interface (AWS CLI) 를 사용하여 GitHub 연결을 생성할 수 있습니다.

이렇게 하려면 create-connection 명령을 사용합니다.

⚠ Important

OR를 통해 생성된 AWS CloudFormation 연결은 기본적으로 PENDING 상태입니다. AWS CLI CLI를 사용하여 연결을 생성한 후 콘솔을 사용하여 연결을 편집하여 상태를 설정합니다. AWS CloudFormationAVAILABLE

연결을 생성하려면 GitHub

1. 터미널(Linux, macOS, Unix) 또는 명령 프롬프트(Windows)를 엽니다. AWS CLI 를 사용하여 연결에 --connection-name 대해 --provider-type 및 를 지정하여 create-connection 명령을 실행합니다. 이 예제에서 타사 공급자 이름은 GitHub이고 지정된 연결 이름은 MyConnection입니다.

```
aws codestar-connections create-connection --provider-type GitHub --connection-name MyConnection
```

이 명령이 제대로 실행되면 다음과 비슷한 연결 ARN 정보가 반환됩니다.

```
{
  "ConnectionArn": "arn:aws:codestar-connections:us-west-2:account_id:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f"
}
```

2. 콘솔을 사용하여 연결을 완료합니다. 자세한 정보는 [보류 중인 연결 업데이트](#)을 참조하세요.

GitHub 엔터프라이즈 서버에 대한 연결 생성

연결을 사용하여 AWS 리소스를 타사 리포지토리와 연결합니다. AWS Management Console 또는 AWS Command Line Interface (AWS CLI) 를 사용하여 GitHub 엔터프라이즈 서버에 연결할 수 있습니다.

연결은 연결 생성 시 앱 설치를 승인하는 데 사용되는 GitHub Enterprise Server 계정이 소유한 리포지토리에 대한 액세스만 제공합니다. GitHub

시작하기 전:

- GitHub 엔터프라이즈 서버 인스턴스와 그 안에 리포지토리가 이미 있어야 합니다.

- 이 섹션에 표시된 대로 GitHub 앱을 생성하고 호스트 리소스를 생성하려면 GitHub Enterprise Server 인스턴스의 관리자여야 합니다.

⚠ Important

GitHub Enterprise Server용 호스트를 설정하면 웹훅 이벤트 데이터에 대한 VPC 엔드포인트가 자동으로 생성됩니다. 2020년 11월 24일 이전에 호스트를 만들었고 VPC PrivateLink 웹훅 엔드포인트를 사용하려면 먼저 호스트를 삭제한 다음 새 호스트를 생성해야 합니다.

주제

- [GitHub엔터프라이즈 서버 연결 생성 \(콘솔\)](#)
- [GitHub엔터프라이즈 서버 \(CLI\) 에 대한 연결 생성](#)

GitHub엔터프라이즈 서버 연결 생성 (콘솔)

GitHub 엔터프라이즈 서버 연결을 생성하려면 엔터프라이즈 서버가 설치된 위치에 대한 정보를 제공하고 GitHub GitHub 엔터프라이즈 자격 증명을 사용하여 연결 생성을 승인합니다.

주제

- [GitHub엔터프라이즈 서버 연결 생성 \(콘솔\)](#)

GitHub엔터프라이즈 서버 연결 생성 (콘솔)

GitHub 엔터프라이즈 서버에 연결하려면 서버 URL과 GitHub 엔터프라이즈 자격 증명을 준비하십시오.

호스트를 생성하려면

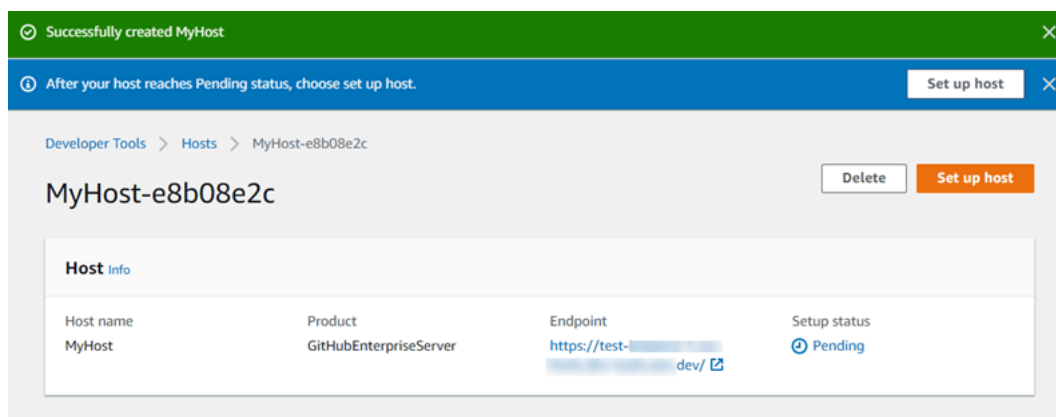
1. [에](https://console.aws.amazon.com/codesuite/settings/connections) AWS Management Console로그인하고 에서 AWS 개발자 도구 콘솔을 엽니다
2. [호스트(Hosts)] 탭에서 [호스트 생성(Create host)]을 선택합니다.
3. [호스트 이름(Host name)]에 호스트에 사용할 이름을 입력합니다.
4. 공급자 선택에서 다음 중 하나를 선택합니다.
 - GitHub 엔터프라이즈 서버

- GitLab 자체 관리형
5. [URL]에 공급자가 설치된 인프라의 엔드포인트를 입력합니다.
 6. 서버가 Amazon VPC 내에 구성되어 있는데 VPC와 연결하려는 경우 [VPC 사용(Use a VPC)]을 선택합니다. 그렇지 않은 경우 [VPC 없음]을 선택합니다.
 7. Amazon VPC로 인스턴스를 시작한 후 VPC에 연결하려는 경우 [VPC 사용(Use a VPC)]을 선택하고 다음을 완료합니다.
 - a. [VPC ID]에서 VPC ID를 선택합니다. 인스턴스가 설치된 인프라의 VPC를 선택하거나 VPN 또는 Direct Connect를 통해 인스턴스에 액세스할 수 있는 VPC를 선택해야 합니다.
 - b. 프라이빗 VPC가 구성되어 있고 퍼블릭이 아닌 인증 기관을 사용하여 TLS 검증을 수행하도록 인스턴스를 구성한 경우 TLS 인증서에 인증서 ID를 입력합니다. TLS 인증서 값은 인증서의 퍼블릭 키입니다.
 8. [호스트 생성(Create host)]을 선택합니다.
 9. 호스트 세부 정보 페이지가 표시된 후, 호스트가 생성됨에 따라 호스트 상태가 바뀝니다.

Note

호스트 설정에 VPC 구성이 포함된 경우 호스트 네트워크 구성 요소를 프로비저닝하는 데 몇 분 정도 걸립니다.

호스트가 [보류 중(Pending)] 상태가 될 때까지 기다린 다음 설치를 완료합니다. 자세한 정보는 [보류 중인 호스트 설정](#)을 참조하세요.



2단계: GitHub 엔터프라이즈 서버 연결 만들기 (콘솔)

1. 에서 개발자 도구 콘솔에 AWS Management Console 로그인하고 엽니다 <https://console.aws.amazon.com/codesuite/settings/connections>.
2. [설정(Settings)] > [연결(Connections)]을 선택한 다음 [연결 생성(Create connection)]을 선택합니다.
3. 설치된 GitHub 엔터프라이즈 서버 리포지토리에 연결하려면 GitHub 엔터프라이즈 서버를 선택합니다.

GitHub 엔터프라이즈 서버에 연결

1. [연결 이름(Connection name)]에 연결 이름을 입력합니다.

The screenshot shows the 'Create a connection' page in the AWS Management Console. The breadcrumb navigation is 'Developer Tools > Connections > Create connection'. The main heading is 'Create a connection Info'. Under 'Select a provider', there are three radio button options: 'Bitbucket', 'GitHub', and 'GitHub Enterprise Server', with the latter selected. Below this is the 'Connection Settings Info' section. It contains a 'Connection name' field with the value 'connection-ghes' and a 'URL' field with the value 'https://myserver.dev/'. There is also an unchecked checkbox for 'Use a VPC' with a note: 'If your GitHub Enterprise Server is only accessible in a VPC, configure details here. Otherwise, skip this step. Complete these steps in the same AWS Region as your VPC.' At the bottom right, there are two buttons: 'Cancel' and 'Connect to GitHub Enterprise Server'.

2. [URL]에 서버의 엔드포인트를 입력합니다.

Note

제공된 URL을 이미 사용하여 GitHub Enterprise Server 연결을 설정한 경우 해당 엔드포인트에 대해 이전에 생성한 호스트 리소스 ARN을 선택하라는 메시지가 표시됩니다.

3. (선택 사항) Amazon VPC로 서버를 시작한 후 VPC에 연결하려는 경우 VPC 사용을 선택하고 다음을 완료합니다.
 - a. [VPC ID]에서 VPC ID를 선택합니다. GitHub 엔터프라이즈 서버 인스턴스가 설치된 인프라용 VPC 또는 VPN 또는 Direct Connect를 통해 GitHub 엔터프라이즈 서버 인스턴스에 액세스할 수 있는 VPC를 선택해야 합니다.
 - b. [서브넷 ID(Subnet ID)]를 선택하고 [추가(Add)]를 선택합니다. 해당 필드에서 호스트에 사용할 서브넷 ID를 선택합니다. 서브넷은 최대 10개까지 선택할 수 있습니다.

GitHub Enterprise Server 인스턴스가 설치된 인프라의 서브넷을 선택하거나 VPN 또는 Direct Connect를 통해 설치된 GitHub Enterprise Server 인스턴스에 액세스할 수 있는 서브넷을 선택해야 합니다.

- c. [보안 그룹 ID(Security group IDs)]에서 [추가(Add)]를 선택합니다. 해당 필드에서 호스트에 사용할 보안 그룹을 선택합니다. 보안 그룹은 최대 10개까지 선택할 수 있습니다.

GitHub Enterprise Server 인스턴스가 설치된 인프라의 보안 그룹을 선택하거나 VPN 또는 Direct Connect를 통해 설치된 GitHub Enterprise Server 인스턴스에 액세스할 수 있는 보안 그룹을 선택해야 합니다.

- d. 프라이빗 VPC를 구성하고 비공개 인증 기관을 사용하여 TLS 검증을 수행하도록 GitHub Enterprise Server 인스턴스를 구성한 경우 TLS 인증서에 인증서 ID를 입력합니다. TLS 인증서 값은 인증서의 퍼블릭 키여야 합니다.

VPC ID

Choose the VPC in which your GitHub Enterprise Server is configured.

Subnet IDs

Choose the subnet or subnets for the VPC in which your GitHub Enterprise Server is configured.

Subnet ID

Security group IDs

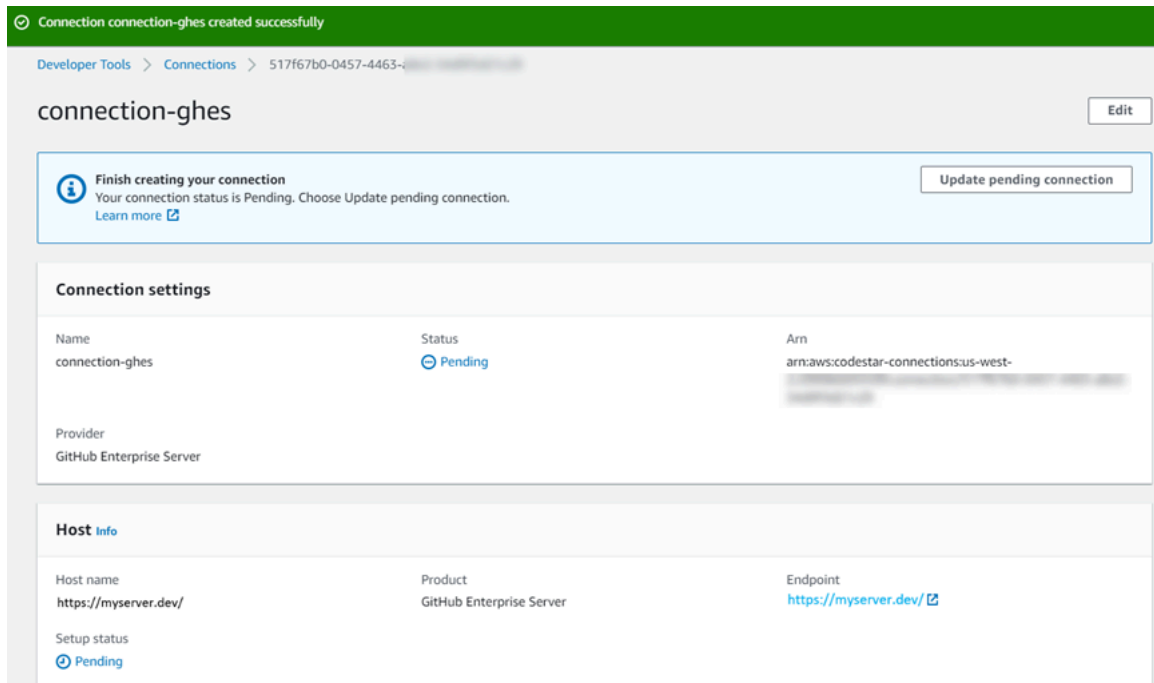
Choose the security group or groups for the VPC in which your GitHub Enterprise Server is configured.

Security group ID

TLS certificate - optional

If you have a private certificate authority behind a VPC or you are using a self-signed certificate paste the TLS certificate here.

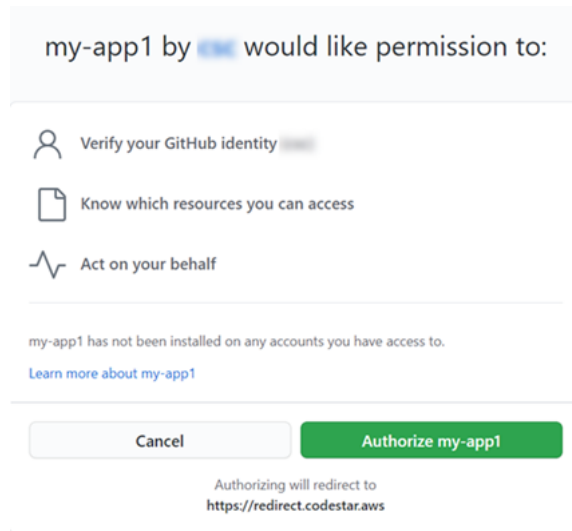
4. [GitHub 엔터프라이즈 서버에 연결] 을 선택합니다. 생성된 연결은 [대기 중(Pending)] 상태로 표시됩니다. 사용자가 제공한 서버 정보를 사용하여 연결을 위한 호스트 리소스가 생성됩니다. 호스트 이름으로 URL이 사용됩니다.
5. [보류 중인 연결을 업데이트(Update pending connection)]를 선택합니다.



6. 메시지가 표시되면 GitHub 엔터프라이즈 로그인 페이지에서 GitHub 엔터프라이즈 자격 증명으로 로그인합니다.
7. GitHub 앱 만들기 페이지에서 앱 이름을 선택합니다.

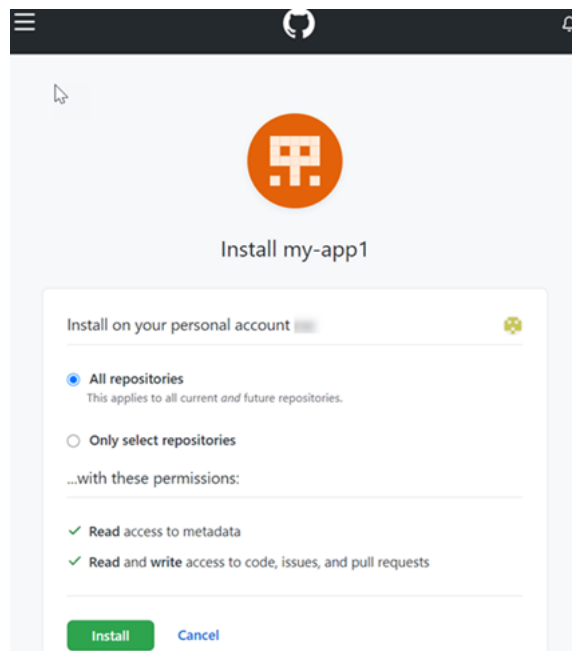


8. GitHub <app-name>인증 페이지에서 Authorize (권한 부여) 를 선택합니다.



9. 앱 설치 페이지에서 AWS CodeStar Connector 앱을 설치할 준비가 되었다는 메시지가 표시됩니다. 여러 조직이 있는 경우 앱을 설치할 조직을 선택하라는 메시지가 표시될 수 있습니다.

앱을 설치할 리포지토리 설정을 선택합니다. 설치를 선택합니다.



10. 연결 페이지에 생성된 연결이 [사용 가능(Available)] 상태로 표시됩니다.

GitHub엔터프라이즈 서버 (CLI) 에 대한 연결 생성

AWS Command Line Interface (AWS CLI) 를 사용하여 연결을 생성할 수 있습니다.

이렇게 하려면 create-host 및 create-connection 명령을 사용합니다.

⚠ Important

OR를 통해 생성된 AWS CloudFormation 연결은 기본적으로 PENDING 상태입니다. AWS CLI CLI를 사용하여 연결을 생성한 후 콘솔을 사용하여 연결을 편집하여 상태를 설정합니다. AWS CloudFormation AVAILABLE

1단계: GitHub 엔터프라이즈 서버 (CLI) 용 호스트를 만들려면

1. 터미널(Linux, macOS, Unix) 또는 명령 프롬프트(Windows)를 엽니다. AWS CLI 를 사용하여 연결에 `--name`, `--provider-type`, 및 `--provider-endpoint` 를 지정하여 `create-host` 명령을 실행합니다. 이 예제에서 서드 파티 공급자 이름은 `GitHubEnterpriseServer`이고 엔드포인트는 `my-instance.dev`입니다.

```
aws codestar-connections create-host --name MyHost --provider-type
GitHubEnterpriseServer --provider-endpoint "https://my-instance.dev"
```

이 명령이 제대로 실행되면 다음과 비슷한 호스트 Amazon 리소스 이름(ARN) 정보가 반환됩니다.

```
{
  "HostArn": "arn:aws:codestar-connections:us-west-2:account_id:host/My-
Host-28aef605"
}
```

이 단계가 끝나면 호스트는 PENDING 상태입니다.

2. 콘솔을 사용하여 호스트 설정을 완료하고 호스트를 Available 상태로 전환합니다. 자세한 정보는 [보류 중인 호스트 설정](#)을 참조하세요.

2단계: 콘솔에서 보류 중인 호스트 설정

1. 에 AWS Management Console 로그인하고 에서 개발자 도구 콘솔을 엽니다 <https://console.aws.amazon.com/codesuite/settings/connections>.
2. 콘솔을 사용하여 호스트 설정을 완료하고 호스트를 Available 상태로 전환합니다. [보류 중인 호스트 설정](#) 섹션을 참조하십시오.

3단계: GitHub 엔터프라이즈 서버 (CLI) 에 대한 연결을 만들려면

1. 터미널(Linux, macOS, Unix) 또는 명령 프롬프트(Windows)를 엽니다. AWS CLI 를 사용하여 연결을 `--connection-name` 위한 `--host-arn` 및 을 지정하여 `create-connection` 명령을 실행합니다.

```
aws codestar-connections create-connection --host-arn arn:aws:codestar-connections:us-west-2:account_id:host/MyHost-234EXAMPLE --connection-name MyConnection
```

이 명령이 제대로 실행되면 다음과 비슷한 연결 ARN 정보가 반환됩니다.

```
{
  "ConnectionArn": "arn:aws:codestar-connections:us-west-2:account_id:connection/aEXAMPLE-8aad"
}
```

2. 콘솔을 사용하여 보류 중인 연결을 설정합니다. 자세한 정보는 [보류 중인 연결 업데이트](#)을 참조하세요.

4단계: 콘솔에서 GitHub 엔터프라이즈 서버 연결을 완료하려면

1. 에서 개발자 도구 콘솔에 AWS Management Console 로그인하고 엽니다 <https://console.aws.amazon.com/codesuite/settings/connections>.
2. 콘솔을 사용하여 보류 중인 연결을 설정한 뒤 Available 상태로 연결을 전환합니다. 자세한 정보는 [보류 중인 연결 업데이트](#)을 참조하세요.

에 대한 연결 생성 GitLab

AWS Management Console 또는 AWS Command Line Interface (AWS CLI) 를 사용하여 gitlab.com에서 호스팅되는 리포지토리에 대한 연결을 생성할 수 있습니다.

Note

에서 GitLab 이 연결 설치를 승인하면 서비스에 데이터를 처리할 수 있는 권한을 부여하는 것이며 애플리케이션을 제거하여 언제든지 권한을 취소할 수 있습니다.

시작하기 전:

- 예 이미 계정을 생성했어야 합니다. GitLab

Note

연결은 연결을 만들고 권한을 부여하는 데 사용된 계정에 대한 액세스 권한만 제공합니다.

Note

소유자 역할이 있는 위치에 연결을 만든 다음 다음과 같은 리소스가 있는 저장소에서 연결을 사용할 수 CodePipeline 있습니다. GitLab 그룹 내 리포지토리의 경우 그룹 소유자가 아니어도 됩니다.

주제

- [GitLab\(콘솔\)에 대한 연결 생성](#)
- [GitLab\(CLI\)에 대한 연결 생성](#)

GitLab(콘솔)에 대한 연결 생성

1단계: 연결 생성

1. 예 로그인한 다음 에서 AWS 개발자 도구 콘솔을 엽니다 <https://console.aws.amazon.com/codesuite/settings/connections>. AWS Management Console
2. 설정을 선택한 다음 연결을 선택합니다. 연결 생성을 선택합니다.
3. GitLab 리포지토리에 대한 연결을 생성하려면 공급자 선택에서 을 선택합니다 GitLab. [연결 이름 (Connection name)]에 생성하려는 연결의 이름을 입력합니다. 연결 대상을 선택합니다 GitLab.

Developer Tools > Connections > Create connection

Create a connection [Info](#)

Select a provider

Bitbucket

GitHub

GitHub Enterprise Server

GitLab

Create GitLab connection [Info](#)

Connection name

▶ **Tags - optional**

Connect to GitLab

4. 의 로그인 페이지가 GitLab 표시되면 자격 증명으로 로그인한 다음 로그인을 선택합니다.
5. GitLab 계정 액세스를 위한 연결 승인을 요청하는 메시지와 함께 승인 페이지가 표시됩니다.

Authorize를 선택합니다.

Authorize **codestar-connections** to use your account?

An application called **codestar-connections** is requesting access to your GitLab account. This application was created by **Amazon AWS**. Please note that this application is not provided by GitLab and you should verify its authenticity before allowing access.

This application will be able to:

- **Access the authenticated user's API**
Grants complete read/write access to the API, including all groups and projects, the container registry, and the package registry.
- **Read the authenticated user's personal information**
Grants read-only access to the authenticated user's profile through the /user API endpoint, which includes username, public email, and full name. Also grants access to read-only API endpoints under /users.
- **Read Api**
Grants read access to the API, including all groups and projects, the container registry, and the package registry.
- **Allows read-only access to the repository**
Grants read-only access to repositories on private projects using Git-over-HTTP or the Repository Files API.
- **Allows read-write access to the repository**
Grants read-write access to repositories on private projects using Git-over-HTTP (not using the API).

Deny

Authorize

6. 브라우저가 연결 콘솔 페이지로 돌아갑니다. GitLab 연결 만들기의 연결 이름에 새 연결이 표시됩니다.
7. 연결 대상을 선택합니다 GitLab.

연결이 성공적으로 생성되면 성공 배너가 표시됩니다. 연결 세부 정보는 연결 설정 페이지에 나와 있습니다.

GitLab(CLI) 에 대한 연결 생성

AWS Command Line Interface (AWS CLI) 를 사용하여 연결을 생성할 수 있습니다.

이렇게 하려면 `create-connection` 명령을 사용합니다.

⚠ Important

OR를 통해 생성된 AWS CloudFormation 연결은 기본적으로 PENDING 상태입니다. AWS CLI CLI를 사용하여 연결을 생성한 후 콘솔을 사용하여 연결을 편집하여 상태를 설정합니다. AWS CloudFormation AVAILABLE

연결을 생성하려면 GitLab

1. 터미널(Linux, macOS, Unix) 또는 명령 프롬프트(Windows)를 엽니다. AWS CLI 를 사용하여 연결에 `--connection-name` 대해 `--provider-type` 및 를 지정하여 `create-connection` 명령을 실행합니다. 이 예제에서 타사 공급자 이름은 GitLab이고 지정된 연결 이름은 MyConnection입니다.

```
aws codestar-connections create-connection --provider-type GitLab --connection-name MyConnection
```

이 명령이 제대로 실행되면 다음과 비슷한 연결 ARN 정보가 반환됩니다.

```
{
  "ConnectionArn": "arn:aws:codestar-connections:us-west-2:account_id:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f"
}
```

2. 콘솔을 사용하여 연결을 완료합니다. 자세한 정보는 [보류 중인 연결 업데이트](#)을 참조하세요.

GitLab 자체 관리형 연결을 생성하세요.

자체 관리형 설치를 통해 GitLab 엔터프라이즈 에디션 또는 GitLab 커뮤니티 에디션에 대한 연결을 생성할 수 있습니다.

AWS Management Console 또는 AWS Command Line Interface (AWS CLI) 를 사용하여 GitLab 자체 관리용 연결 및 호스트를 만들 수 있습니다.

Note

GitLab 자체 관리에서 이 연결 애플리케이션을 승인하면 서비스에 데이터를 처리할 권한을 부여하는 것으로 간주되며, 애플리케이션을 제거하여 언제든지 권한을 취소할 수 있습니다.

GitLab 셀프 매니지드 연결을 생성하기 전에 다음 단계에 설명된 대로 연결에 사용할 호스트를 생성해야 합니다. 설치된 공급자의 호스트 생성 워크플로에 대한 개요는 [호스트 생성 또는 업데이트 워크플로우](#) 섹션을 참조하십시오.

VPC로 호스트를 구성할 수도 있습니다. 호스트 리소스의 네트워크 및 VPC 구성에 대한 자세한 내용은 [\(선택 사항\) 사전 요구 사항: 연결을 위한 네트워크 또는 Amazon VPC 구성 및 호스트의 VPC 구성 문제 해결](#)의 VPC 사전 요구 사항을 참조하십시오.

시작하기 전:

- 에서 이미 계정을 GitLab 생성했고 자체 관리형 설치가 포함된 GitLab Enterprise Edition 또는 GitLab Community Edition이 설치되어 있어야 합니다. 자세한 내용은 https://docs.gitlab.com/ee/subscriptions/self_managed/을 참조하십시오.

Note

연결은 연결을 만들고 권한을 부여하는 데 사용된 계정에 대한 액세스 권한만 제공합니다.

Note

소유자 역할이 있는 저장소에 대한 연결을 만든 다음 GitLab, 이 연결을 다음과 같은 CodePipeline 리소스와 함께 사용할 수 있습니다. 그룹 내 리포지토리의 경우 그룹 소유자가 아니어도 됩니다.

- 범위가 축소된 권한만 있는 GitLab 개인 액세스 토큰 (PAT) 을 이미 생성했어야 합니다: api. 자세한 내용은 https://docs.gitlab.com/ee/user/profile/personal_access_tokens.html을 참조하십시오. 관리자가 사용하는 PAT만 사용할 수 있습니다.

Note

PAT는 호스트를 인증하는 데 사용되며 연결에 달리 저장되거나 사용되지 않습니다. 호스트를 설정하려면 임시 PAT를 만든 다음 호스트를 설정한 후 PAT를 삭제하면 됩니다.

주제

- [자체 관리형 \(콘솔\) 에 대한 GitLab 연결 생성](#)
- [GitLab 자체 관리형 \(CLI\) 에 대한 연결 생성](#)

자체 관리형 (콘솔) 에 대한 GitLab 연결 생성

다음 단계를 사용하여 호스트를 생성하고 콘솔에서 GitLab 자체 관리형 호스트에 연결할 수 있습니다. VPC에서 호스트를 설정할 때 고려해야 할 사항에 대한 자세한 내용은 [\(선택 사항\) 사전 요구 사항: 연결을 위한 네트워크 또는 Amazon VPC 구성](#) 섹션을 참조하십시오.

Note

단일 GitLab 자체 관리 설치용 호스트를 만든 다음 해당 호스트에 대한 하나 이상의 GitLab 자체 관리 연결을 관리할 수 있습니다.

1단계: 호스트 생성

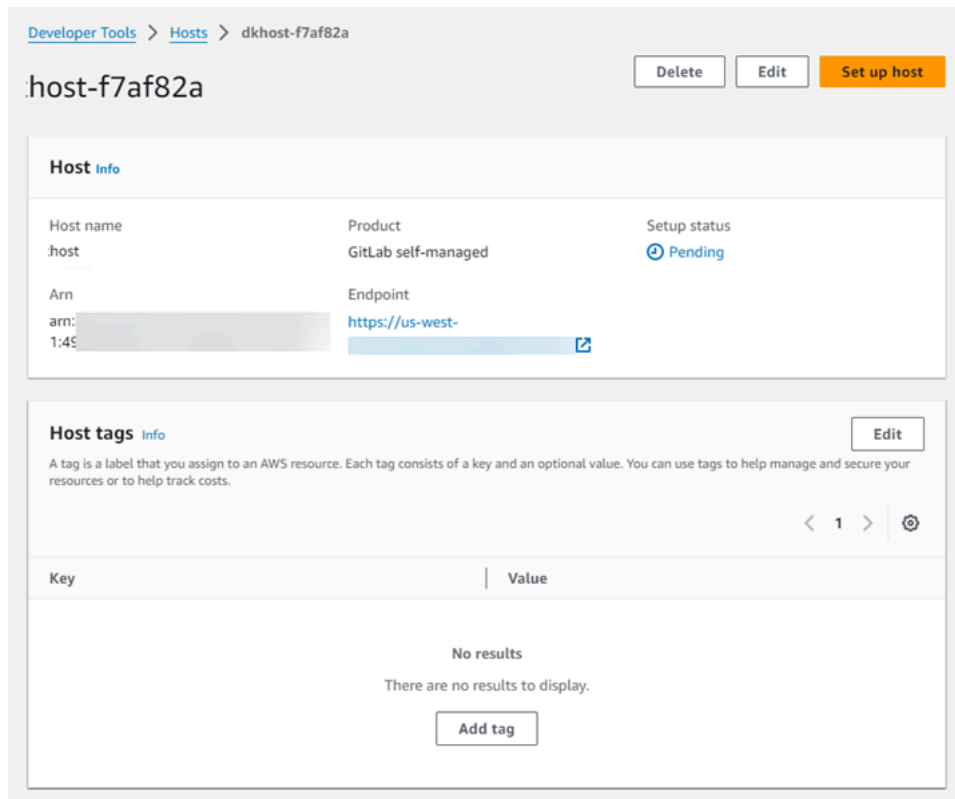
1. 에 로그인한 다음 에서 AWS 개발자 도구 콘솔을 엽니다. AWS Management Console <https://console.aws.amazon.com/codesuite/settings/connections>
2. [호스트(Hosts)] 탭에서 [호스트 생성(Create host)]을 선택합니다.
3. [호스트 이름(Host name)]에 호스트에 사용할 이름을 입력합니다.
4. 공급업체 선택에서 GitLab 자체 관리를 선택합니다.
5. [URL]에 공급자가 설치된 인프라의 엔드포인트를 입력합니다.
6. 서버가 Amazon VPC 내에 구성되어 있는데 VPC와 연결하려는 경우 [VPC 사용(Use a VPC)]을 선택합니다. 그렇지 않은 경우 [VPC 없음]을 선택합니다.

7. (선택 사항) Amazon VPC로 호스트를 시작한 후 VPC에 연결하려는 경우 VPC 사용을 선택하고 다음을 완료합니다.
 - a. [VPC ID]에서 VPC ID를 선택합니다. 호스트가 설치된 인프라의 VPC를 선택하거나 VPN 또는 Direct Connect를 통해 인스턴스에 액세스할 수 있는 VPC를 선택해야 합니다.
 - b. 프라이빗 VPC가 구성되어 있고 퍼블릭이 아닌 인증 기관을 사용하여 TLS 검증을 수행하도록 호스트를 구성한 경우 TLS 인증서에 인증서 ID를 입력합니다. TLS 인증서 값은 인증서의 퍼블릭 키입니다.
8. [호스트 생성(Create host)]을 선택합니다.
9. 호스트 세부 정보 페이지가 표시된 후, 호스트가 생성됨에 따라 호스트 상태가 바뀝니다.

Note


호스트 설정에 VPC 구성이 포함된 경우 호스트 네트워크 구성 요소를 프로비저닝하는 데 몇 분 정도 걸립니다.

호스트가 [보류 중(Pending)] 상태가 될 때까지 기다린 다음 설치를 완료합니다. 자세한 정보는 [보류 중인 호스트 설정](#)을 참조하세요.




The screenshot shows the AWS Developer Tools console for a host named 'host-f7af82a'. The breadcrumb navigation is 'Developer Tools > Hosts > dkhost-f7af82a'. At the top right, there are buttons for 'Delete', 'Edit', and 'Set up host'. The main content area is divided into two sections: 'Host Info' and 'Host tags'.

Host Info

Host name	Product	Setup status
host	GitLab self-managed	 Pending
Arn	Endpoint	
arn:aws:dkh:us-west-1:451234567890:dkhost-f7af82a	https://us-west-1.dkhost.amazonaws.com/451234567890/dkhost-f7af82a	

Host tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to help manage and secure your resources or to help track costs.

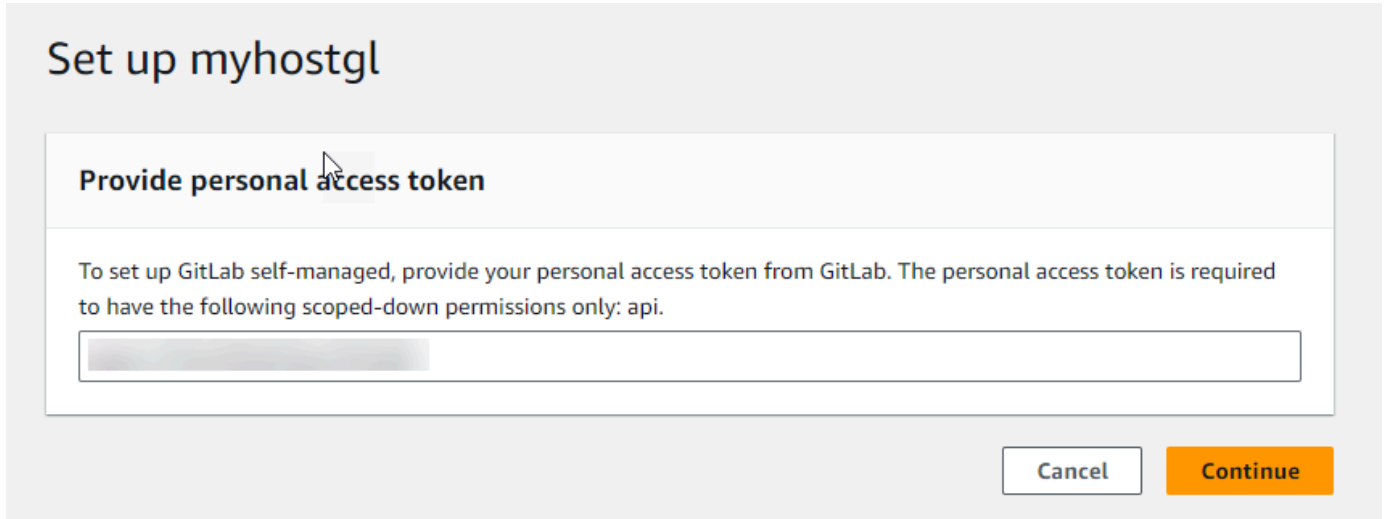
Navigation: < 1 > 

Key	Value
No results There are no results to display.	

Buttons: Edit, Add tag

2단계: 보류 중인 호스트 설정

1. 호스트 설정을 선택합니다.
2. **host_name** 설정 페이지가 표시됩니다. 개인용 액세스 토큰 제공에서 GitLab PAT에 다음과 같은 범위가 축소된 권한만 제공하십시오: api.



Set up myhostgl

Provide personal access token

To set up GitLab self-managed, provide your personal access token from GitLab. The personal access token is required to have the following scoped-down permissions only: api.

Cancel Continue

3. 호스트가 성공적으로 등록되면 호스트 세부 정보 페이지가 나타나고 호스트 상태가 [사용 가능 (Available)]으로 표시됩니다.

The screenshot shows the AWS Management Console interface for a host configuration. At the top right, there are three buttons: 'Delete', 'Edit', and 'Set up host'. The main content area is titled 'Host Info' and contains a table with the following information:

Host name	Product	Setup status
glhost	GitLab self-managed	Available
Arn	Endpoint	
[Blurred]	[Blurred]	

Below the 'Host Info' section is the 'Host tags Info' section, which includes an 'Edit' button and a description: 'A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to help manage and secure your resources or to help track costs.' At the bottom right, there are navigation arrows and a settings gear icon.

3단계: 연결 생성

1. 에 로그인한 다음 에서 AWS 개발자 도구 콘솔을 엽니다 <https://console.aws.amazon.com/codesuite/settings/connections>. AWS Management Console
2. 설정을 선택한 다음 연결을 선택합니다. 연결 생성을 선택합니다.
3. GitLab 리포지토리에 대한 연결을 생성하려면 공급자 선택에서 GitLab 자체 관리를 선택합니다. [연결 이름(Connection name)]에 생성하려는 연결의 이름을 입력합니다.

4. [URL]에 서버의 엔드포인트를 입력합니다.
5. Amazon VPC로 서버를 시작한 후 VPC에 연결하려는 경우 [VPC 사용(Use a VPC)]을 선택하고 다음을 완료합니다.
 - a. [VPC ID]에서 VPC ID를 선택합니다. 호스트가 설치된 인프라의 VPC를 선택하거나 VPN 또는 Direct Connect를 통해 호스트에 액세스할 수 있는 VPC를 선택해야 합니다.
 - b. [서브넷 ID(Subnet ID)]를 선택하고 [추가(Add)]를 선택합니다. 해당 필드에서 호스트에 사용할 서브넷 ID를 선택합니다. 서브넷은 최대 10개까지 선택할 수 있습니다.

호스트가 설치된 인프라의 서브넷을 선택하거나 VPN 또는 Direct Connect를 통해 설치된 호스트에 액세스할 수 있는 서브넷을 선택해야 합니다.

- c. [보안 그룹 ID(Security group IDs)]에서 [추가(Add)]를 선택합니다. 해당 필드에서 호스트에 사용할 보안 그룹을 선택합니다. 보안 그룹은 최대 10개까지 선택할 수 있습니다.

호스트가 설치된 인프라의 보안 그룹을 선택하거나 VPN 또는 Direct Connect를 통해 설치된 호스트에 액세스할 수 있는 보안 그룹을 선택해야 합니다.

- d. 프라이빗 VPC가 구성되어 있고 퍼블릭이 아닌 인증 기관을 사용하여 TLS 검증을 수행하도록 호스트를 구성한 경우 TLS 인증서에 인증서 ID를 입력합니다. TLS 인증서 값은 인증서의 퍼블릭 키여야 합니다.
 6. [GitLab 자체 관리형 시스템에 연결] 을 선택합니다. 생성된 연결은 [대기 중(Pending)] 상태로 표시됩니다. 사용자가 제공한 서버 정보를 사용하여 연결을 위한 호스트 리소스가 생성됩니다. 호스트 이름으로 URL이 사용됩니다.
 7. [보류 중인 연결을 업데이트(Update pending connection)]를 선택합니다.
 8. 의 로그인 페이지가 GitLab 표시되면 자격 증명으로 로그인한 다음 로그인을 선택합니다.
 9. GitLab 계정 액세스를 위한 연결 승인을 요청하는 메시지와 함께 승인 페이지가 표시됩니다. Authorize를 선택합니다.
 10. 브라우저가 연결 콘솔 페이지로 돌아갑니다. GitLab 연결 만들기에서 연결 이름에 새 연결이 표시됩니다.
 11. [GitLab 자체 관리형 시스템에 연결] 을 선택합니다.
- 연결이 성공적으로 생성되면 성공 배너가 표시됩니다. 연결 세부 정보는 연결 설정 페이지에 나와 있습니다.

GitLab 자체 관리형 (CLI) 에 대한 연결 생성

AWS Command Line Interface (AWS CLI) 를 사용하여 자체 관리용 GitLab 호스트와 연결을 생성할 수 있습니다.

이렇게 하려면 create-host 및 create-connection 명령을 사용합니다.

Important

OR를 통해 생성된 AWS CloudFormation 연결은 기본적으로 PENDING 상태입니다. AWS CLI CLI를 사용하여 연결을 생성한 후 콘솔을 사용하여 연결을 편집하여 상태를 설정합니다. AWS CloudFormationAVAILABLE

1단계: GitLab 자체 관리형 (CLI) 용 호스트 생성하기

1. 터미널(Linux, macOS, Unix) 또는 명령 프롬프트(Windows)를 엽니다. AWS CLI 를 사용하여 연결에 --name--provider-type, 및 --provider-endpoint 를 지정하여 create-host 명령을

실행합니다. 이 예제에서 서드 파티 공급자 이름은 GitLabSelfManaged이고 엔드포인트는 my-instance.dev입니다.

```
aws codestar-connections create-host --name MyHost --provider-type
GitLabSelfManaged --provider-endpoint "https://my-instance.dev"
```

이 명령이 제대로 실행되면 다음과 비슷한 호스트 Amazon 리소스 이름(ARN) 정보가 반환됩니다.

```
{
  "HostArn": "arn:aws:codestar-connections:us-west-2:account_id:host/My-
Host-28aef605"
}
```

이 단계가 끝나면 호스트는 PENDING 상태입니다.

2. 콘솔을 사용하여 호스트 설정을 완료하고 다음 단계에서 호스트를 Available 상태로 전환합니다.

2단계: 콘솔에서 보류 중인 호스트 설정

1. 에 AWS Management Console 로그인하고 에서 개발자 도구 콘솔을 엽니다 <https://console.aws.amazon.com/codesuite/settings/connections>.
2. 콘솔을 사용하여 호스트 설정을 완료하고 호스트를 Available 상태로 전환합니다. [보류 중인 호스트 설정](#) 섹션을 참조하십시오.

3단계: GitLab 자체 관리형 (CLI) 연결을 만들려면

1. 터미널(Linux, macOS, Unix) 또는 명령 프롬프트(Windows)를 엽니다. AWS CLI 를 사용하여 연결을 --connection-name 위한 --host-arn 및 을 지정하여 create-connection 명령을 실행합니다.

```
aws codestar-connections create-connection --host-arn arn:aws:codestar-
connections:us-west-2:account_id:host/MyHost-234EXAMPLE --connection-name
MyConnection
```

이 명령이 제대로 실행되면 다음과 비슷한 연결 ARN 정보가 반환됩니다.

```
{
```

```
"ConnectionArn": "arn:aws:codestar-connections:us-west-2:account_id:connection/
aEXAMPLE-8aad"
}
```

2. 콘솔을 사용하여 다음 단계에서 보류 중인 연결을 설정합니다.

4단계: 콘솔에서 GitLab 자체 관리형 연결을 완료하려면

1. 에 AWS Management Console 로그인하고 에서 개발자 도구 콘솔을 엽니다. <https://console.aws.amazon.com/codesuite/settings/connections>
2. 콘솔을 사용하여 보류 중인 연결을 설정한 뒤 Available 상태로 연결을 전환합니다. 자세한 정보는 [보류 중인 연결 업데이트](#)를 참조하세요.

보류 중인 연결 업데이트

AWS Command Line Interface (AWS CLI) 또는 를 통해 생성된 AWS CloudFormation 연결은 기본적으로 PENDING 상태입니다. AWS CLI AWS CloudFormation OR로 연결을 만든 후 콘솔을 사용하여 연결을 업데이트하여 상태를 설정합니다AVAILABLE.

Note

보류 중인 연결을 업데이트하려면 콘솔을 사용해야 합니다. AWS CLI를 사용하여 보류 중인 연결을 업데이트할 수 없습니다.

처음 콘솔을 사용하여 타사 공급자에 새 연결을 추가할 때는 연결과 관련된 설치를 사용하여 타사 공급자와의 OAuth 핸드셰이크를 완료해야 합니다.

개발자 도구 콘솔을 사용하여 보류 중인 연결을 완료할 수 있습니다.

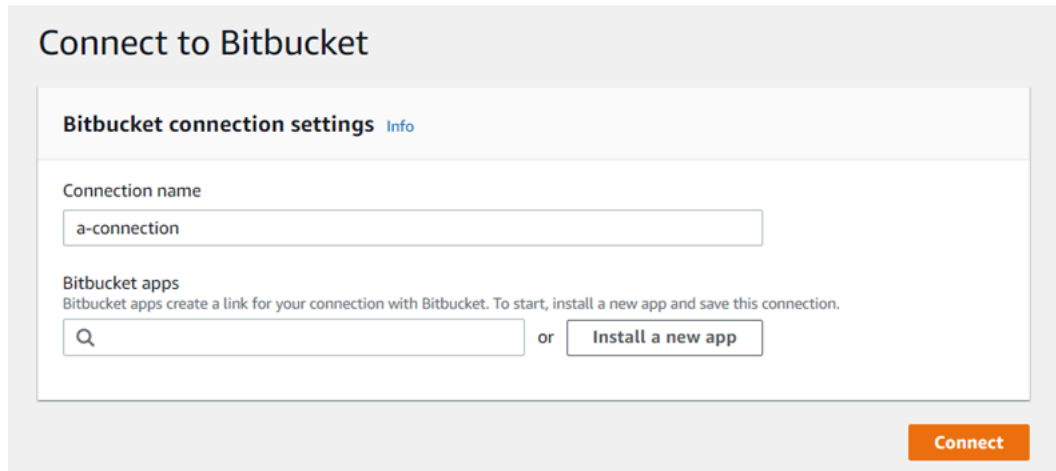
연결을 완료하려면

1. 에서 AWS 개발자 도구 콘솔을 엽니다<https://console.aws.amazon.com/codesuite/settings/connections>.
2. [설정(Settings)] > [연결(Connections)]을 선택합니다.
AWS 계정과 연결된 모든 연결의 이름이 표시됩니다.
3. 이름에서 업데이트할 보류 중인 연결의 이름을 선택합니다.

[보류 중인 연결 업데이트(Update a pending connection)]는 [보류 중(Pending)] 상태의 연결을 선택하면 활성화됩니다.

4. [보류 중인 연결을 업데이트(Update a pending connection)]를 선택합니다.
5. [Bitbucket에 연결(Connect to Bitbucket)] 페이지의 [연결 이름(Connection name)]에서 연결 이름을 확인합니다.

[Bitbucket 앱(Bitbucket apps)]에서 앱 설치를 선택하거나 [새 앱 설치(Install a new app)]를 선택하여 앱을 새로 만듭니다.



6. 앱 설치 페이지에서 앱이 Bitbucket 계정에 연결을 시도하고 있다는 메시지가 표시됩니다. AWS CodeStar [Grant access(액세스 권한 부여)]를 선택합니다.



AWS CodeStar requests access

This app is hosted at <https://codestar-connections.webhooks.aws>

- Read your account information
- Read your repositories and their pull requests
- Administer your repositories
- Read and modify your repositories

Authorize for

Allow AWS CodeStar to do this?

This 3rd party vendor has not provided a privacy policy or terms of use.

Atlassian's Privacy Policy is not applicable to the use of this App.

Grant access Cancel

7. 새 설치의 연결 ID가 표시됩니다. Complete connection(연결 완료)을 선택합니다.

연결 나열

개발자 도구 콘솔 또는 AWS Command Line Interface (AWS CLI)의 list-connections 명령을 사용하여 계정의 연결 목록을 볼 수 있습니다.

연결 나열(콘솔)

연결을 나열하려면

1. <https://console.aws.amazon.com/codesuite/settings/connections>에서 개발자 도구 콘솔을 엽니다.
2. [설정(Settings)] > [연결(Connections)]을 선택합니다.
3. 연결의 이름, 상태 및 ARN을 확인합니다.

연결 나열(CLI)

를 사용하여 타사 코드 리포지토리에 대한 연결을 AWS CLI 나열할 수 있습니다. GitHub Enterprise Server에 대한 연결과 같이 호스트 리소스에 연결된 연결의 경우 출력은 호스트 ARN을 추가로 반환합니다.

이렇게 하려면 `list-connections` 명령을 사용합니다.

연결을 나열하려면

- 터미널 (Linux, macOS 또는 Unix) 또는 명령 프롬프트 (Windows) 를 열고 AWS CLI 를 사용하여 명령을 실행합니다. `list-connections`

```
aws codestar-connections list-connections --provider-type Bitbucket
--max-results 5 --next-token: next-token
```

이 명령은 다음 출력을 반환합니다.

```
{
  "Connections": [
    {
      "ConnectionName": "my-connection",
      "ProviderType": "Bitbucket",
      "Status": "PENDING",
      "ARN": "arn:aws:codestar-connections:us-west-2:account_id:connection/
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
      "OwnerAccountId": "account_id"
    },
    {
      "ConnectionName": "my-other-connection",
      "ProviderType": "Bitbucket",
      "Status": "AVAILABLE",
      "ARN": "arn:aws:codestar-connections:us-west-2:account_id:connection/
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
      "OwnerAccountId": "account_id"
    },
  ],
  "NextToken": "next-token"
}
```

연결 삭제

개발자 도구 콘솔 또는 AWS Command Line Interface (AWS CLI)의 delete-connection 명령을 사용하여 연결을 삭제할 수 있습니다.

주제

- [연결 삭제\(콘솔\)](#)
- [연결 삭제\(CLI\)](#)

연결 삭제(콘솔)

연결 삭제

1. <https://console.aws.amazon.com/codesuite/settings/connections>에서 개발자 도구 콘솔을 엽니다.
2. [설정(Settings)] > [연결(Connections)]을 선택합니다.
3. 연결 이름에서 삭제할 연결의 이름을 선택합니다.
4. 삭제를 선택합니다.
5. 필드에 **delete**를 입력하여 확인한 후, 삭제를 선택합니다.

Important

이 작업은 실행을 취소할 수 없습니다.

연결 삭제(CLI)

AWS Command Line Interface (AWS CLI) 를 사용하여 연결을 삭제할 수 있습니다.

이렇게 하려면 delete-connection 명령을 사용합니다.

Important

명령을 실행하면 연결이 삭제됩니다. 확인 대화 상자는 표시되지 않습니다. 새 연결을 생성할 수 있지만 Amazon 리소스 이름(ARN)은 재사용되지 않습니다.

연결 삭제

- 터미널(Linux, macOS, Unix) 또는 명령 프롬프트(Windows)를 엽니다. AWS CLI 를 사용하여 delete-connection 명령을 실행하고 삭제하려는 연결의 ARN을 지정합니다.

```
aws codestar-connections delete-connection --connection-arn arn:aws:codestar-connections:us-west-2:account_id:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
```

이 명령은 아무 것도 반환하지 않습니다.

연결 리소스 태그 지정

태그는 사용자가 또는 AWS 리소스에 AWS 할당하는 사용자 지정 속성 레이블입니다. 각 AWS 태그에는 두 부분이 있습니다.

- 태그 키(예: CostCenter, Environment 또는 Project). 태그 키는 대/소문자를 구별합니다.
- 태그 값(예: 111122223333, Production 또는 팀 이름)으로 알려진 선택적 필드. 태그 값을 생략하는 것은 빈 문자열을 사용하는 것과 같습니다. 태그 키처럼 태그 값은 대/소문자를 구별합니다.

태그 키와 태그 값을 합해서 키-값 페어라고 합니다.

콘솔 또는 CLI를 사용하여 리소스에 태그를 지정할 수 있습니다.

CodeConnections에서 다음 리소스 유형에 태그를 지정할 수 있습니다.

- 연결
- 호스트

이 단계에서는 이미 최신 버전의 를 AWS CLI 설치했거나 현재 버전으로 업데이트했다고 가정합니다. 자세한 내용을 알아보려면 AWS Command Line Interface 사용자 가이드에서 [AWS CLI설치](#)를 참조하세요.

태그로 리소스를 식별, 구성 및 추적하는 것 외에도, 태그 인 AWS Identity and Access Management (IAM) 정책을 사용하여 리소스를 보고 상호 작용할 수 있는 사용자를 제어할 수 있습니다. 태그 기반 액세스 정책의 예는 [태그를 사용하여 AWS CodeStar Connections 리소스에 대한 액세스를 제어합니다](#). 단원을 참조하세요.

주제

- [리소스에 태깅\(콘솔\)](#)
- [리소스 태깅\(CLI\)](#)

리소스에 태깅(콘솔)

콘솔을 사용하여 연결 리소스에 대한 태그를 추가, 업데이트 또는 제거할 수 있습니다.

주제

- [연결 리소스에 태그 추가\(콘솔\)](#)
- [연결 리소스의 태그 보기\(콘솔\)](#)
- [연결 리소스의 태그 편집\(콘솔\)](#)
- [연결 리소스에서 태그 제거\(콘솔\)](#)

연결 리소스에 태그 추가(콘솔)

콘솔을 사용하여 기존 연결 또는 호스트에 태그를 추가할 수 있습니다.

Note

GitHubEnterprise Server와 같은 설치된 공급자에 대한 연결을 생성하고 호스트 리소스도 자동으로 생성되면 생성 중 태그가 연결에만 추가됩니다. 따라서 새 연결에 호스트를 재사용하려는 경우 호스트에 별도로 태깅할 수 있습니다. 호스트에 태그를 추가하려면 다음 단계를 따릅니다.

연결에 대한 태그를 추가하려면

1. 콘솔에 로그인합니다. 탐색 창에서 설정을 선택합니다.
2. [설정(Settings)]에서 [연결(Connections)]을 선택합니다. 연결 탭을 선택합니다.
3. 편집할 연결을 선택합니다. 연결 설정 페이지가 표시됩니다.
4. [연결 태그(Connection tags)]에서 [편집(Edit)]을 선택합니다. [연결 태그 편집(Edit Connection tags)] 페이지가 표시됩니다.
5. 키 및 값 필드에 추가할 각 태그 세트에 대한 키 페어를 입력합니다. (값 필드는 선택 사항입니다.) 예를 들어 키에 **Project**을 입력합니다. 값에는 **ProjectA**를 입력합니다.

6. (선택 사항) 행을 추가하고 태그를 더 입력하려면 태그 추가를 선택합니다.
7. 제출을 선택합니다. 태그는 연결 설정 아래에 나열됩니다.

호스트에 대한 태그를 추가하려면

1. 콘솔에 로그인합니다. 탐색 창에서 설정을 선택합니다.
2. [설정(Settings)]에서 [연결(Connections)]을 선택합니다. [호스트(Hosts)] 탭을 선택합니다.
3. 편집할 호스트를 선택합니다. 호스트 설정 페이지가 표시됩니다.
4. [호스트 태그(Host tags)]에서 [편집(Edit)]을 선택합니다. [호스트 태그(Host tags)] 페이지가 표시됩니다.
5. 키 및 값 필드에 추가할 각 태그 세트에 대한 키 페어를 입력합니다. (값 필드는 선택 사항입니다.) 예를 들어 키에 **Project**을 입력합니다. 값에는 **ProjectA**를 입력합니다.

6. (선택 사항) [태그 추가(Add tag)]를 선택하여 행을 추가하고 호스트에 대해 태그를 더 입력합니다.
7. 제출을 선택합니다. 태그는 호스트 설정 아래에 나열됩니다.

연결 리소스의 태그 보기(콘솔)

콘솔을 사용하여 기존 리소스의 태그를 볼 수 있습니다.

연결에 대한 태그를 보려면

1. 콘솔에 로그인합니다. 탐색 창에서 설정을 선택합니다.
2. [설정(Settings)]에서 [연결(Connections)]을 선택합니다. 연결 탭을 선택합니다.
3. 표시할 연결을 선택합니다. 연결 설정 페이지가 표시됩니다.
4. [연결 태그(Connection tags)]에서 [키(Key)] 및 [값(Value)] 열 아래에 연결에 대한 태그가 표시됩니다.

호스트에 대한 태그를 보려면

1. 콘솔에 로그인합니다. 탐색 창에서 설정을 선택합니다.
2. [설정(Settings)]에서 [연결(Connections)]을 선택합니다. [호스트(Hosts)] 탭을 선택합니다.
3. 표시할 호스트를 선택합니다.
4. [호스트 태그(Host tags)]에서 [키(Key)] 및 [값(Value)] 열 아래에 호스트에 대한 태그가 표시됩니다.

연결 리소스의 태그 편집(콘솔)

콘솔을 사용하여 연결 리소스에 추가된 태그를 편집할 수 있습니다.

연결에 대한 태그를 편집하려면

1. 콘솔에 로그인합니다. 탐색 창에서 설정을 선택합니다.
2. [설정(Settings)]에서 [연결(Connections)]을 선택합니다. 연결 탭을 선택합니다.
3. 편집할 연결을 선택합니다. 연결 설정 페이지가 표시됩니다.
4. [연결 태그(Connection tags)]에서 [편집(Edit)]을 선택합니다. [연결 태그(Connection tags)] 페이지가 표시됩니다.
5. 키 및 값 필드에서 필요에 따라 각 필드의 값을 업데이트합니다. 예를 들어 **Project** 키의 값에서 **ProjectA**를 **ProjectB**로 변경합니다.
6. 제출을 선택합니다.

호스트에 대한 태그를 편집하려면

1. 콘솔에 로그인합니다. 탐색 창에서 설정을 선택합니다.
2. [설정(Settings)]에서 [연결(Connections)]을 선택합니다. [호스트(Hosts)] 탭을 선택합니다.
3. 편집할 호스트를 선택합니다. 호스트 설정 페이지가 표시됩니다.
4. [호스트 태그(Host tags)]에서 [편집(Edit)]을 선택합니다. [호스트 태그(Host tags)] 페이지가 표시됩니다.
5. 키 및 값 필드에서 필요에 따라 각 필드의 값을 업데이트합니다. 예를 들어 **Project** 키의 값에서 **ProjectA**를 **ProjectB**로 변경합니다.
6. 제출을 선택합니다.

연결 리소스에서 태그 제거(콘솔)

콘솔을 사용하여 연결 리소스에서 태그를 제거할 수 있습니다. 연결된 리소스에서 태그를 제거하면 태그가 삭제됩니다.

연결의 태그를 제거하려면

1. 콘솔에 로그인합니다. 탐색 창에서 설정을 선택합니다.
2. [설정(Settings)]에서 [연결(Connections)]을 선택합니다. 연결 탭을 선택합니다.
3. 편집할 연결을 선택합니다. 연결 설정 페이지가 표시됩니다.
4. [연결 태그(Connection tags)]에서 [편집(Edit)]을 선택합니다. [연결 태그(Connection tags)] 페이지가 표시됩니다.
5. 삭제할 각 태그의 키와 값 옆에 있는 태그 제거를 선택합니다.
6. 제출을 선택합니다.

호스트의 태그를 제거하려면

1. 콘솔에 로그인합니다. 탐색 창에서 설정을 선택합니다.
2. [설정(Settings)]에서 [연결(Connections)]을 선택합니다. [호스트(Hosts)] 탭을 선택합니다.
3. 편집할 호스트를 선택합니다. 호스트 설정 페이지가 표시됩니다.
4. [호스트 태그(Host tags)]에서 [편집(Edit)]을 선택합니다. [호스트 태그(Host tags)] 페이지가 표시됩니다.
5. 삭제할 각 태그의 키와 값 옆에 있는 태그 제거를 선택합니다.
6. 제출을 선택합니다.

리소스 태깅(CLI)

CLI를 사용하여 연결 리소스에 대한 태그를 표시, 추가, 업데이트 또는 제거할 수 있습니다.

주제

- [연결 리소스에 태그 추가\(CLI\)](#)
- [연결 리소스의 태그 보기\(CLI\)](#)
- [연결 리소스의 태그 편집\(CLI\)](#)
- [연결 리소스에서 태그 제거\(CLI\)](#)

연결 리소스에 태그 추가(CLI)

를 사용하여 연결의 AWS CLI 리소스에 태그를 지정할 수 있습니다.

터미널이나 명령줄에서 태그를 추가할 리소스의 Amazon 리소스 이름(ARN)과 추가할 태그의 키와 값을 지정하여 `tag-resource` 명령을 실행합니다. 2개 이상의 태그를 추가할 수 있습니다.

연결에 대한 태그를 추가하려면

1. 리소스의 ARN을 가져옵니다. [연결 나열](#)에 나와 있는 `list-connections` 명령을 사용하여 연결 ARN을 가져옵니다.
2. 터미널 또는 명령줄에서 `tag-resource` 명령을 실행합니다.

```
## ##, ## ### ##### # ## ### ## ### ### #####, ## ## ProjectA# Project##
## ## ## ## true# ## ## ### #####. ReadOnly
```

```
aws codestar-connections tag-resource --resource-arn arn:aws:codestar-connections:us-west-2:account_id:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f --tags Key=Project,Value=ProjectA Key=IscontainerBased,Value=true
```

성공한 경우 이 명령은 아무 것도 반환하지 않습니다.

호스트에 대한 태그를 추가하려면

1. 리소스의 ARN을 가져옵니다. [호스트 나열](#)에 나와 있는 `list-hosts` 명령을 사용하여 호스트 ARN을 가져옵니다.
2. 터미널 또는 명령줄에서 `tag-resource` 명령을 실행합니다.

```
## ## ## ### ##### ##### ## 2#, ## ## ProjectA# Project ## #, ## ## true#
## ## ## #####. IscontainerBased
```

```
aws codestar-connections tag-resource --resource-arn arn:aws:codestar-
connections:us-west-2:account_id:host/My-Host-28aef605 --tags
Key=Project,Value=ProjectA Key=IscontainerBased,Value=true
```

성공한 경우 이 명령은 아무 것도 반환하지 않습니다.

연결 리소스의 태그 보기(CLI)

를 사용하여 연결 리소스의 AWS 태그를 볼 수 있습니다. AWS CLI 태그가 추가되지 않은 경우 반환되는 목록은 비어 있습니다. `list-tags-for-resource` 명령을 사용하여 연결 또는 호스트에 추가된 태그를 볼 수 있습니다.

연결에 대한 태그를 보려면

1. 리소스의 ARN을 가져옵니다. [연결 나열](#)에 나와 있는 `list-connections` 명령을 사용하여 연결 ARN을 가져옵니다.
2. 터미널 또는 명령줄에서 `list-tags-for-resource` 명령을 실행합니다. 예를 들어 연결에 대한 태그 키 및 태그 값 목록을 보려면 다음 명령을 수행합니다.

```
aws codestar-connections list-tags-for-resource --resource-arn arn:aws:codestar-
connections:us-west-2:account_id:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
```

이 명령은 리소스와 연결된 태그를 반환합니다. 이 예에서는 연결에 대해 반환된 두 키-값 페어를 보여 줍니다.

```
{
  "Tags": [
    {
      "Key": "Project",
      "Value": "ProjectA"
    },
    {
      "Key": "ReadOnly",
      "Value": "true"
    }
  ]
}
```

```
}

```

호스트에 대한 태그를 보려면

1. 리소스의 ARN을 가져옵니다. [호스트 나열](#)에 나와 있는 list-hosts 명령을 사용하여 호스트 ARN을 가져옵니다.
2. 터미널 또는 명령줄에서 list-tags-for-resource 명령을 실행합니다. 예를 들어 호스트에 대한 태그 키 및 태그 값 목록을 보려면 다음 명령을 수행합니다.

```
aws codestar-connections list-tags-for-resource --resource-arn arn:aws:codestar-connections:us-west-2:account_id:host/My-Host-28aef605
```

이 명령은 리소스와 연결된 태그를 반환합니다. 이 예에서는 호스트에 대해 반환된 두 키-값 페어를 보여 줍니다.

```
{
  "Tags": [
    {
      "Key": "IscontainerBased",
      "Value": "true"
    },
    {
      "Key": "Project",
      "Value": "ProjectA"
    }
  ]
}
```

연결 리소스의 태그 편집(CLI)

를 사용하여 리소스의 태그를 편집할 수 있습니다. AWS CLI 기존 키의 값을 변경하거나 다른 키를 추가할 수 있습니다.

터미널이나 명령줄에서 태그를 업데이트할 리소스의 ARN과 업데이트할 태그 키 및 태그 값을 지정하여 tag-resource 명령을 실행합니다.

태그를 편집할 때 지정되지 않은 모든 태그 키는 유지되지만, 키는 같지만 새 값이 있는 태그는 업데이트됩니다. edit 명령을 사용하여 추가된 새 키는 새 키-값 페어로 추가됩니다.

연결에 대한 태그를 편집하려면

1. 리소스의 ARN을 가져옵니다. [연결 나열](#)에 나와 있는 list-connections 명령을 사용하여 연결 ARN을 가져옵니다.
2. 터미널 또는 명령줄에서 tag-resource 명령을 실행합니다.

이 예제에서는 Project 키 값이 ProjectB로 변경됩니다.

```
aws codestar-connections tag-resource --resource-arn arn:aws:codestar-connections:us-west-2:account_id:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f --tags Key=Project,Value=ProjectB
```

성공한 경우 이 명령은 아무 것도 반환하지 않습니다. 연결과 연결된 태그를 확인하려면 list-tags-for-resource 명령을 실행합니다.

호스트에 대한 태그를 편집하려면

1. 리소스의 ARN을 가져옵니다. [호스트 나열](#)에 나와 있는 list-hosts 명령을 사용하여 호스트 ARN을 가져옵니다.
2. 터미널 또는 명령줄에서 tag-resource 명령을 실행합니다.

이 예제에서는 Project 키 값이 ProjectB로 변경됩니다.

```
aws codestar-connections tag-resource --resource-arn arn:aws:codestar-connections:us-west-2:account_id:host/My-Host-28aef605 --tags Key=Project,Value=ProjectB
```

성공한 경우 이 명령은 아무 것도 반환하지 않습니다. 호스트와 연결된 태그를 확인하려면 list-tags-for-resource 명령을 실행합니다.

연결 리소스에서 태그 제거(CLI)

다음 단계에 따라 AWS CLI 를 사용하여 리소스에서 태그를 제거합니다. 연결된 리소스에서 태그를 제거하면 태그가 삭제됩니다.

Note

연결 리소스를 삭제하면 삭제된 리소스에서 모든 태그 연결이 제거됩니다. 연결 리소스를 삭제하기 전에 태그를 제거할 필요가 없습니다.

터미널이나 명령줄에서 태그를 제거할 리소스의 ARN과 제거할 태그의 태그 키를 지정하여 `untag-resource` 명령을 실행합니다. 예를 들어 *Project* 및 *ReadOnly* 태그 키를 사용하는 연결에서 여러 태그를 제거하려면 다음 명령을 사용합니다.

```
aws codestar-connections untag-resource --resource-arn arn:aws:codestar-connections:us-west-2:account_id:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f --tag-keys Project ReadOnly
```

성공한 경우 이 명령은 아무 것도 반환하지 않습니다. 리소스와 연결된 태그를 확인하려면 `list-tags-for-resource` 명령을 실행합니다. 출력은 모든 태그가 제거되었음을 보여줍니다.

```
{
  "Tags": []
}
```

연결 세부 정보 보기

개발자 도구 콘솔 또는 AWS Command Line Interface (AWS CLI)의 `get-connection` 명령을 사용하여 연결에 대한 세부 정보를 볼 수 있습니다. 를 AWS CLI 사용하려면 이미 최신 버전을 AWS CLI 설치했거나 현재 버전으로 업데이트해야 합니다. 자세한 내용을 알아보려면 AWS Command Line Interface 사용자 가이드에서 [AWS CLI 설치](#)를 참조하세요.

연결을 보려면(콘솔)

1. <https://console.aws.amazon.com/codesuite/settings/connections>에서 개발자 도구 콘솔을 엽니다.
2. [설정(Settings)] > [연결(Connections)]을 선택합니다.
3. 보려는 연결 옆에 있는 버튼을 선택한 다음 [세부 정보 보기(View details)]를 선택합니다.
4. 연결에 대해 다음과 같은 정보가 나타납니다.
 - 연결 이름
 - 연결의 공급자 유형

- 연결 상태
 - 연결 ARN
 - GitHubEnterprise Server와 같은 설치된 공급자에 대해 연결을 생성한 경우 연결과 관련된 호스트 정보.
 - GitHubEnterprise Server와 같은 설치된 공급자에 대해 연결을 생성한 경우 연결을 위해 호스트와 연결된 엔드포인트 정보
5. 연결이 [보류 중(Pending)] 상태인 경우 연결을 완료하려면 [보류 중인 연결 업데이트(Update pending connection)]를 선택합니다. 자세한 내용은 [보류 중인 연결 업데이트](#)를 참조하세요.

연결을 보려면(CLI)

- 터미널 또는 명령줄에서 get-connection 명령을 실행합니다. 예를 들어 ARN 값이 `arn:aws:codestar-connections:us-west-2:account_id:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f`인 연결에 대한 세부 정보를 보려면 다음을 수행합니다.

```
aws codestar-connections get-connection --connection-arn arn:aws:codestar-connections:us-west-2:account_id:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
```

명령이 성공할 경우 연결 세부 정보가 반환됩니다.

Bitbucket 연결의 출력 예:

```
{
  "Connection": {
    "ConnectionName": "MyConnection",
    "ConnectionArn": "arn:aws:codestar-connections:us-west-2:account_id:connection/cdacd948-EXAMPLE",
    "ProviderType": "Bitbucket",
    "OwnerAccountId": "account_id",
    "ConnectionStatus": "AVAILABLE"
  }
}
```

GitHub 연결에 대한 예제 출력:

```
{
  "Connection": {
```

```

    "ConnectionName": "MyGitHubConnection",
    "ConnectionArn": "arn:aws:codestar-connections:us-
west-2:account_id:connection/ebcd4a13-EXAMPLE",
    "ProviderType": "GitHub",
    "OwnerAccountId": "account_id",
    "ConnectionStatus": "AVAILABLE"
  }
}

```

GitHub 엔터프라이즈 서버 연결의 출력 예:

```

{
  "Connection": {
    "ConnectionName": "MyConnection",
    "ConnectionArn": "arn:aws:codestar-connections:us-
west-2:account_id:connection/2d178fb9-EXAMPLE",
    "ProviderType": "GitHubEnterpriseServer",
    "OwnerAccountId": "account_id",
    "ConnectionStatus": "PENDING",
    "HostArn": "arn:aws:codestar-connections:us-west-2:account_id:host/sdfsdf-
EXAMPLE"
  }
}

```

호스트 작업

GitHub Enterprise Server와 같은 설치된 공급자 유형에 대한 연결을 생성하려면 먼저 AWS Management Console을 사용하여 호스트를 생성합니다. 호스트는 공급자가 설치된 인프라를 나타내기 위해 생성하는 리소스입니다. 그런 다음 해당 호스트를 사용하여 연결을 생성합니다. 자세한 내용은 [연결 관련 작업](#) 섹션을 참조하세요.

예를 들어 인프라를 나타낼 공급자의 서드 파티 앱을 등록할 수 있도록 연결을 위한 호스트를 생성합니다. 공급자 유형별로 하나의 호스트를 생성한 다음 해당 공급자 유형에 대한 모든 연결에 해당 호스트를 사용합니다.

콘솔을 사용하여 GitHub Enterprise Server와 같은 설치된 공급자 유형에 대한 연결을 생성할 경우, 콘솔이 자동으로 호스트 리소스를 생성합니다.

주제

- [호스트 생성](#)

- [보류 중인 호스트 설정](#)
- [호스트 나열](#)
- [호스트 편집](#)
- [호스트 삭제](#)
- [호스트 세부 정보 보기](#)

호스트 생성

AWS Management Console 또는 AWS Command Line Interface(AWS CLI)를 사용하여 인프라에 설치된 서드 파티 코드 리포지토리에 대한 연결을 생성합니다. 예를 들어 GitHub Enterprise Server가 Amazon EC2 인스턴스에서 가상 머신으로 실행될 수 있습니다. GitHub Enterprise Server에 대한 연결을 생성하기 전에 연결에 사용할 호스트를 생성합니다.

설치된 공급자의 호스트 생성 워크플로에 대한 개요는 [호스트 생성 또는 업데이트 워크플로우](#) 섹션을 참조하십시오.

시작하기 전:

- (선택 사항) VPC로 호스트를 생성하려면 네트워크 또는 Virtual Private Cloud(VPC)가 이미 생성되어 있어야 합니다.
- 인스턴스가 이미 생성되어 있어야 하며, VPC와 연결하려는 경우 VPC로 호스트를 시작한 상태여야 합니다.

Note

각 VPC는 한 번에 하나의 호스트와만 연결할 수 있습니다.

VPC로 호스트를 구성할 수도 있습니다. 호스트 리소스의 네트워크 및 VPC 구성에 대한 자세한 내용은 [\(선택 사항\) 사전 요구 사항: 연결을 위한 네트워크 또는 Amazon VPC 구성 및 호스트의 VPC 구성 문제 해결](#)의 VPC 사전 요구 사항을 참조하십시오.

콘솔을 사용하여 GitHub Enterprise Server에 대한 호스트 및 연결을 생성하려면 [GitHub 엔터프라이즈 서버 연결 생성 \(콘솔\)](#) 섹션을 참조하십시오. 콘솔이 자동으로 호스트를 생성합니다.

콘솔을 사용하여 GitLab 자체 관리형에 대한 호스트 및 연결을 생성하려면 [GitLab 자체 관리형 연결을 생성하세요](#) 섹션을 참조하십시오. 콘솔이 자동으로 호스트를 생성합니다.

(선택 사항) 사전 요구 사항: 연결을 위한 네트워크 또는 Amazon VPC 구성

인프라가 네트워크 연결로 구성된 경우 이 섹션을 건너뛸 수 있습니다.

VPC에서만 호스트에 액세스할 수 있는 경우 다음 VPC 요구 사항을 따르고 계속 진행합니다.

VPC 요구 사항

VPC로 호스트를 생성하기로 선택할 수 있습니다. 다음은 설치에 대해 설정한 VPC에 따른 일반적인 VPC 요구 사항입니다.

- 퍼블릭 및 프라이빗 서브넷을 사용하여 퍼블릭 VPC를 구성할 수 있습니다. 기본 CIDR 블록 또는 서브넷이 없는 경우 AWS 계정에 기본 VPC를 사용할 수 있습니다.
- 프라이빗 VPC가 구성되어 있고 퍼블릭이 아닌 인증 기관을 사용하여 TLS 검증을 수행하도록 GitHub Enterprise Server 인스턴스를 구성한 경우 호스트 리소스에 대한 TLS 인증서를 제공해야 합니다.
- AWS CodeStar Connections가 호스트를 생성하면 웹훅에 대한 VPC 엔드포인트(PrivateLink)가 자동으로 생성됩니다. 자세한 내용은 [AWS CodeStar Connections 및 인터페이스 VPC 엔드포인트 \(AWS PrivateLink\)](#) 섹션을 참조하세요.
- 보안 그룹 구성:
 - 호스트 생성 중에 사용되는 보안 그룹에는 네트워크 인터페이스가 GitHub Enterprise Server 인스턴스에 연결할 수 있도록 허용하는 인바운드 및 아웃바운드 규칙이 필요합니다.
 - 호스트 설정에 포함되지 않은 GitHub Enterprise Server 인스턴스에 연결된 보안 그룹에는 연결에 의해 생성된 네트워크 인터페이스로부터의 인바운드 및 아웃바운드 액세스가 필요합니다.
- VPC 서브넷들은 리전의 서로 다른 가용 영역에 상주해야 합니다. 각 가용 영역은 다른 가용 영역에서 발생한 장애를 격리시킬 수 있도록 서로 분리된 공간입니다. 각 서브넷은 단일 가용 영역 내에서만 존재해야 하며, 여러 영역으로 스케일 아웃할 수 없습니다.

VPC 및 서브넷 작업에 대한 자세한 내용은 Amazon VPC 사용 설명서에서 [IPv4의 경우, VPC 및 서브넷 크기 조정](#)을 참조하세요.

호스트 설정을 위해 제공하는 VPC 정보

다음 단계에서 연결의 호스트 리소스를 생성할 때 다음을 제공해야 합니다.

- VPC ID: GitHub Enterprise Server 인스턴스가 설치된 서버의 VPC ID이거나, VPN 또는 Direct Connect를 통해 설치된 GitHub Enterprise Server 인스턴스에 액세스할 수 있는 VPC의 ID입니다.

- 서브넷 ID 또는 ID: GitHub Enterprise Server 인스턴스가 설치된 서버의 서브넷 ID이거나, VPN 또는 Direct Connect를 통해 설치된 GitHub Enterprise Server 인스턴스에 액세스할 수 있는 서브넷의 ID입니다.
- 보안 그룹 또는 그룹: GitHub Enterprise Server 인스턴스가 설치된 서버의 보안 그룹이거나, VPN 또는 Direct Connect를 통해 설치된 GitHub Enterprise Server 인스턴스에 액세스할 수 있는 보안 그룹입니다.
- 엔드포인트: 서버 엔드포인트를 준비하고 다음 단계로 계속 진행합니다.

VPC 또는 호스트 연결 문제를 해결하는 등의 자세한 내용은 [호스트의 VPC 구성 문제 해결](#) 섹션을 참조하세요.

권한 요구 사항

AWS CodeStar Connections는 VPC 연결을 용이하게 하기 위해 호스트 생성 프로세스의 일부로 자동으로 네트워크 리소스를 생성합니다. 여기에는 호스트에서 데이터를 쿼리하는 AWS CodeStar Connections를 위한 네트워크 인터페이스, 그리고 VPC 엔드포인트 또는 PrivateLink를 사용하여 웹후크를 통해 이벤트 데이터를 AWS CodeStar Connections로 전송하는 것이 포함됩니다. 이러한 네트워크 리소스를 생성하려면 호스트를 생성하는 역할에 다음 권한이 있어야 합니다.

```
ec2:CreateNetworkInterface
ec2:CreateTags
ec2:DescribeDhcpOptions
ec2:DescribeNetworkInterfaces
ec2:DescribeSubnets
ec2>DeleteNetworkInterface
ec2:DescribeVpcs
ec2:CreateVpcEndpoint
ec2>DeleteVpcEndpoints
ec2:DescribeVpcEndpoints
```

VPC의 권한 또는 호스트 연결 문제를 해결하는 데 대한 자세한 내용은 [호스트의 VPC 구성 문제 해결](#) 섹션을 참조하세요.

웹후크 VPC 엔드포인트에 대한 자세한 내용은 [AWS CodeStar Connections 및 인터페이스 VPC 엔드포인트\(AWS PrivateLink\)](#) 섹션을 참조하세요.

주제

- [연결을 위한 호스트 생성\(콘솔\)](#)
- [연결을 위한 호스트 생성\(CLI\)](#)

연결을 위한 호스트 생성(콘솔)

GitHub Enterprise Server 또는 GitLab 자체 관리형을 통한 설치 연결의 경우 호스트를 사용하여 서드 파티 공급자가 설치된 인프라의 엔드포인트를 나타냅니다.

VPC 호스트를 설정할 때 고려해야 할 사항에 대한 자세한 내용은 [GitLab 자체 관리형 연결을 생성하세요](#) 섹션을 참조하세요.

콘솔을 사용하여 GitHub Enterprise Server에 대한 호스트 및 연결을 생성하려면 [GitHub 엔터프라이즈 서버 연결 생성 \(콘솔\)](#) 섹션을 참조하십시오. 콘솔이 자동으로 호스트를 생성합니다.

콘솔을 사용하여 GitLab 자체 관리형에 대한 호스트 및 연결을 생성하려면 [GitLab 자체 관리형 연결을 생성하세요](#) 섹션을 참조하십시오. 콘솔이 자동으로 호스트를 생성합니다.

Note

GitHub Enterprise Server 또는 GitLab 자체 관리형 계정별로 한 번만 호스트를 생성할 수 있습니다. 특정 GitHub Enterprise Server 또는 GitLab 자체 관리형 계정에 대한 모든 연결은 동일한 호스트를 사용합니다.

연결을 위한 호스트 생성(CLI)

AWS Command Line Interface(AWS CLI)를 사용하여 설치된 연결의 호스트를 생성할 수 있습니다.

Note

GitHub Enterprise Server 계정별로 한 번만 호스트를 생성합니다. 특정 GitHub Enterprise Server 계정에 대한 모든 연결은 동일한 호스트를 사용합니다.

호스트를 사용하여 서드 파티 공급자가 설치된 인프라의 엔드포인트를 나타냅니다. CLI를 통해 호스트를 생성하려면 create-host 명령을 사용합니다. 호스트 생성을 완료하고 나면 호스트가 [보류 중 (Pending)] 상태입니다. 그러면 호스트를 설정하여 [사용 가능(Available)] 상태로 전환합니다. 호스트를 사용할 수 있게 되면 연결을 생성하는 단계를 완료합니다.

Important

AWS CLI를 통해 생성된 호스트는 기본적으로 Pending 상태입니다. CLI를 사용하여 호스트를 생성한 후 콘솔을 통해 호스트를 설정하여 호스트를 상태를 Available로 전환합니다.

콘솔을 사용하여 GitHub Enterprise Server에 대한 호스트 및 연결을 생성하려면 [GitHub 엔터프라이즈 서버 연결 생성 \(콘솔\)](#) 섹션을 참조하십시오. 콘솔이 자동으로 호스트를 생성합니다.

콘솔을 사용하여 GitLab 자체 관리형에 대한 호스트 및 연결을 생성하려면 [GitLab 자체 관리형 연결을 생성하세요.](#) 섹션을 참조하십시오. 콘솔이 자동으로 호스트를 생성합니다.

보류 중인 호스트 설정

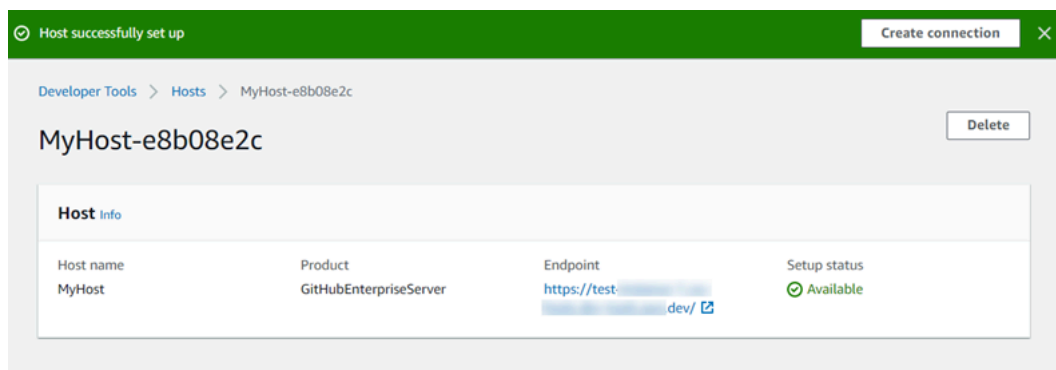
AWS Command Line Interface(AWS CLI) 또는 SDK를 통해 생성된 호스트는 기본적으로 Pending 상태입니다. 콘솔, AWS CLI 또는 SDK를 사용하여 연결을 생성한 후 콘솔을 통해 호스트를 설정하여 상태를 Available로 전환합니다.

호스트가 이미 생성되어 있어야 합니다. 자세한 내용은 [호스트 생성](#)을 참조하십시오.

보류 중인 호스트를 설정하려면

생성된 호스트는 [보류 중(Pending)] 상태입니다. 호스트를 [보류 중(Pending)] 상태에서 [사용 가능(Available)] 상태로 전환하려면 다음 단계를 완료합니다. 이 프로세스에서는 서드 파티 공급자와의 핸드셰이크를 수행하여 호스트에 AWS 연결 앱을 등록합니다.

1. 호스트가 Pending(보류 중) 상태로 전환된 후 AWS 개발자 도구 콘솔에서 Set up host(호스트 설정)를 선택합니다.
2. GitLab 자체 관리형 호스트를 만드는 경우 설정 페이지가 표시됩니다. 개인용 액세스 토큰 제공에서 GitLab PAT에 다음과 같은 범위 축소 권한인 api.man 제공하십시오.
3. 서드 파티 설치 공급자 로그인 페이지(예: GitHub Enterprise Server 로그인 페이지)에서 메시지가 표시되면 계정 자격 증명으로 로그인합니다.
4. 앱 설치 페이지의 [GitHub 앱 이름(GitHub App name)]에 호스트용으로 설치할 앱의 이름을 입력합니다. [GitHub 앱 생성(Create GitHub App)]을 선택합니다.
5. 호스트가 성공적으로 등록되면 호스트 세부 정보 페이지가 나타나고 호스트 상태가 [사용 가능(Available)]으로 표시됩니다.



- 호스트를 사용할 수 있게 되면, 이어서 연결을 생성할 수 있습니다. 성공 배너에서 [연결 생성(Create connection)]을 선택합니다. [\[연결 생성\(Create a connection\)\]](#)의 단계를 수행합니다.

호스트 나열

개발자 도구 콘솔 또는 AWS Command Line Interface(AWS CLI)의 list-connections 명령을 사용하여 계정의 연결 목록을 볼 수 있습니다.

호스트 나열(콘솔)

호스트를 나열하려면

- <https://console.aws.amazon.com/codesuite/settings/connections>에서 개발자 도구 콘솔을 엽니다.
- [호스트(Hosts)] 탭을 선택합니다. 호스트의 이름, 상태, ARN을 봅니다.

호스트 나열(CLI)

AWS CLI를 사용하여 설치된 서드 파티 공급자 연결의 호스트를 나열할 수 있습니다.

이렇게 하려면 list-hosts 명령을 사용합니다.

호스트를 나열하려면

- 터미널(Linux, macOS, Unix) 또는 명령 프롬프트(Windows)를 열고 AWS CLI를 사용하여 list-hosts 명령을 실행합니다.

```
aws codestar-connections list-hosts
```

이 명령은 다음 출력을 반환합니다.

```
{
  "Hosts": [
    {
      "Name": "My-Host",
      "HostArn": "arn:aws:codestar-connections:us-west-2:account_id:host/My-Host-28aef605",
      "ProviderType": "GitHubEnterpriseServer",
      "ProviderEndpoint": "https://my-instance.test.dev",
      "Status": "AVAILABLE"
    }
  ]
}
```

```
]
}
```

호스트 편집

Pending 상태인 호스트의 호스트 설정을 편집할 수 있습니다. 호스트 이름, URL 또는 VPC 구성을 편집할 수 있습니다.

둘 이상의 호스트에 동일한 URL을 사용할 수 없습니다.

Note

VPC 호스트를 설정할 때 고려해야 할 사항에 대한 자세한 내용은 [\(선택 사항\) 사전 요구 사항: 연결을 위한 네트워크 또는 Amazon VPC 구성](#) 섹션을 참조하세요.

호스트를 편집하려면

1. <https://console.aws.amazon.com/codesuite/settings/connections>에서 개발자 도구 콘솔을 엽니다.
2. [설정(Settings)] > [연결(Connections)]을 선택합니다.
3. [호스트(Hosts)] 탭을 선택합니다.

AWS 계정에 연결되고 선택한 AWS 리전에 생성된 호스트가 표시됩니다.

4. 호스트 이름을 편집하려면 [이름(Name)]에 새 값을 입력합니다.
5. 호스트 엔드포인트를 편집하려면 [URL]에 새 값을 입력합니다.
6. 호스트 VPC 구성을 편집하려면 [VPC ID]에 새 값을 입력합니다.
7. [호스트 편집(Edit host)]을 선택합니다.
8. 업데이트된 설정이 표시됩니다. [보류 중인 호스트 설정(Set up Pending host)]을 선택합니다.

호스트 삭제

개발자 도구 콘솔 또는 AWS Command Line Interface(AWS CLI)의 delete-host 명령을 사용하여 호스트를 삭제할 수 있습니다.

주제

- [호스트 삭제\(콘솔\)](#)

- [호스트 삭제\(CLI\)](#)

호스트 삭제(콘솔)

호스트를 삭제하려면

1. <https://console.aws.amazon.com/codesuite/settings/connections>에서 개발자 도구 콘솔을 엽니다.
2. [호스트(Hosts)] 탭을 선택합니다. [이름(Name)]에서 삭제할 호스트의 이름을 선택합니다.
3. 삭제를 선택합니다.
4. 필드에 **delete**를 입력하여 확인한 후, 삭제를 선택합니다.

⚠ Important

이 작업은 실행 취소할 수 없습니다.

호스트 삭제(CLI)

AWS Command Line Interface(AWS CLI)를 사용하여 호스트를 삭제할 수 있습니다.

이렇게 하려면 delete-host 명령을 사용합니다.

⚠ Important

호스트를 삭제하려면 먼저 호스트와 연결된 모든 연결을 삭제해야 합니다. 명령을 실행하면 호스트가 삭제됩니다. 확인 대화 상자는 표시되지 않습니다.

호스트를 삭제하려면

- 터미널(Linux, macOS, Unix) 또는 명령 프롬프트(Windows)를 엽니다. AWS CLI에서 삭제할 호스트의 Amazon 리소스 이름(ARN)을 지정하여 delete-host 명령을 실행합니다.

```
aws codestar-connections delete-host --host-arn "arn:aws:codestar-connections:us-west-2:account_id:host/My-Host-28aef605"
```

이 명령은 아무 것도 반환하지 않습니다.

호스트 세부 정보 보기

개발자 도구 콘솔 또는 AWS Command Line Interface(AWS CLI)의 get-host 명령을 사용하여 호스트의 세부 정보를 볼 수 있습니다.

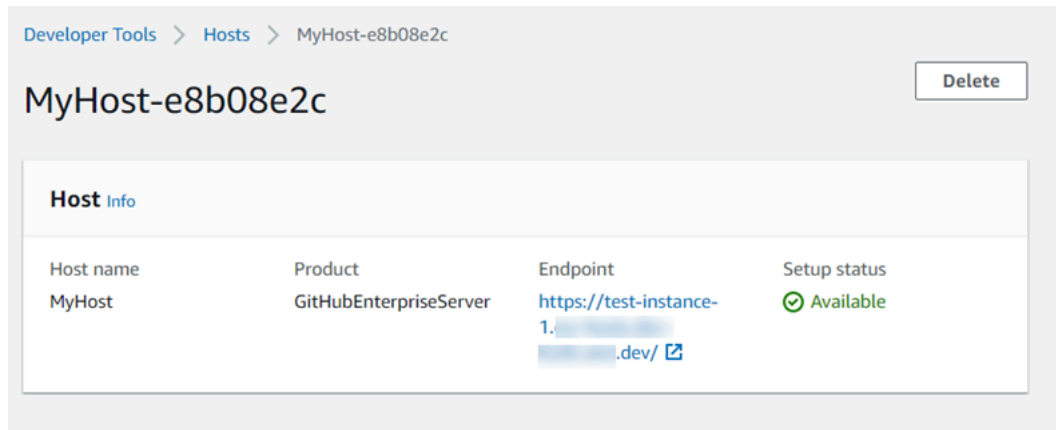
호스트 세부 정보를 보려면(콘솔)

1. AWS Management Console에 로그인하고 <https://console.aws.amazon.com/codesuite/settings/connections>에서 개발자 도구 콘솔을 엽니다.
2. [설정(Settings)] > [연결(Connections)]을 선택한 다음 [호스트(Hosts)] 탭을 선택합니다.
3. 보려는 호스트 옆에 있는 버튼을 선택한 다음 [세부 정보 보기(View details)]를 선택합니다.
4. 호스트에 대해 다음과 같은 정보가 나타납니다.

- 호스트 이름
- 연결의 공급자 유형
- 공급자가 설치된 인프라의 엔드포인트
- 호스트의 설정 상태 연결에 사용할 준비가 된 호스트는 [사용 가능(Available)] 상태로 표시됩니다. 호스트가 생성되었지만 설치가 완료되지 않은 경우 호스트가 다른 상태일 수 있습니다.

사용 가능한 상태는 다음과 같습니다.

- PENDING - 호스트가 생성 완료되었으며 호스트에 공급자 앱을 등록하여 설정을 시작할 준비가 되었습니다.
- AVAILABLE - 호스트가 생성 및 설정 완료되었으며 연결에 사용할 수 있습니다.
- ERROR - 호스트 생성 또는 등록 중에 오류가 발생했습니다.
- VPC_CONFIG_VPC_INITIALIZING - 호스트의 VPC 구성을 생성하는 중입니다.
- VPC_CONFIG_VPC_FAILED_INITIALIZATION - 호스트의 VPC 구성에서 오류가 발생하여 실패했습니다.
- VPC_CONFIG_VPC_AVAILABLE - 호스트의 VPC 구성이 설정 완료되었으며 사용할 수 있습니다.
- VPC_CONFIG_VPC_DELETING - 호스트의 VPC 구성을 삭제하는 중입니다.



- 호스트를 삭제하려면 [삭제(Delete)]를 선택합니다.
- 호스트가 [보류 중(Pending)] 상태인 경우 설정을 완료하려면 [호스트 설정(Set up host)]을 선택합니다. 자세한 내용은 [보류 중인 호스트 설정](#) 섹션을 참조하세요.

호스트 세부 정보를 보려면(CLI)

- 터미널(Linux, macOS, Unix) 또는 명령 프롬프트(Windows)를 열고 AWS CLI에서 세부 정보를 보려는 호스트의 Amazon 리소스 이름(ARN)을 지정하여 get-host 명령을 실행합니다.

```
aws codestar-connections get-host --host-arn arn:aws:codestar-connections:us-west-2:account_id:host/My-Host-28aef605
```

이 명령은 다음 출력을 반환합니다.

```
{
  "Name": "MyHost",
  "Status": "AVAILABLE",
  "ProviderType": "GitHubEnterpriseServer",
  "ProviderEndpoint": "https://test-instance-1.dev/"
}
```

연결된 리포지토리의 동기화 구성 작업

AWS CodeStar 연결에서는 연결을 사용하여 AWS 리소스를 타사 리포지토리 (예: Bitbucket Cloud, GitHub Enterprise Server 및) 에 연결합니다. GitHub GitLab CFN_STACK_SYNC 동기화 유형을 사용하면 Git 리포지토리의 콘텐츠를 AWS 동기화하여 지정된 AWS 리소스를 업데이트할 수 있는 동기화 구

성을 만들 수 있습니다. AWS CloudFormation 연결과 통합되므로 Git sync를 사용하여 동기화하는 연결된 저장소의 템플릿 및 매개 변수 파일을 관리할 수 있습니다.

연결을 생성한 후 연결 CLI 또는 AWS CloudFormation 콘솔을 사용하여 리포지토리 링크 및 동기화 구성을 생성할 수 있습니다.

- 리포지토리 링크: 리포지토리 링크는 사용자 연결과 외부 Git 리포지토리 간의 링크를 생성합니다. 리포지토리 링크를 통해 Git 동기화는 지정된 Git 리포지토리의 파일 변경 사항을 모니터링하고 동기화할 수 있습니다.
- 동기화 구성: 동기화 구성을 사용하여 Git 리포지토리의 콘텐츠를 동기화하여 지정된 AWS 리소스를 업데이트합니다.

자세한 내용은 [AWS CodeStar 연결 API 레퍼런스를](#) 참조하십시오.

AWS CloudFormation 콘솔을 사용하여 AWS CloudFormation 스택에 대한 동기화 구성을 만드는 방법을 안내하는 자습서는 [사용 CloudFormation 설명서의 AWS CloudFormation Git sync](#) 사용을 참조하십시오.

주제

- [리포지토리 링크 작업](#)
- [동기화 구성 작업](#)

리포지토리 링크 작업

리포지토리 링크는 사용자 연결과 외부 Git 리포지토리 간의 링크를 생성합니다. 리포지토리 링크를 통해 Git sync는 지정된 Git 리포지토리의 파일 변경 사항을 모니터링하고 스택에 동기화할 수 있습니다. AWS CloudFormation

리포지토리 링크에 대한 자세한 내용은 [AWS CodeStar 연결 API](#) 참조를 참조하십시오.

주제

- [리포지토리 링크 생성](#)
- [리포지토리 링크 업데이트](#)
- [리포지토리 링크 나열](#)
- [리포지토리 링크 삭제](#)
- [리포지토리 링크 세부 정보 보기](#)

리포지토리 링크 생성

AWS Command Line Interface (AWS CLI) 의 `create-repository-link` 명령을 사용하여 연결과 동기화할 외부 저장소 사이에 링크를 만들 수 있습니다.

저장소 링크를 만들려면 먼저 타사 공급자와 함께 외부 저장소를 이미 생성해야 합니다 (예:) GitHub.

리포지토리 링크를 생성하려면

1. 터미널(Linux, macOS, Unix) 또는 명령 프롬프트(Windows)를 엽니다. 를 AWS CLI 사용하여 `create-repository-link` 명령을 실행합니다. 관련 연결의 ARN, 소유자 ID, 리포지토리 이름을 지정합니다.

```
aws codestar-connections create-repository-link --connection-arn arn:aws:codestar-connections:us-east-1:account_id:connection/001f5be2-a661-46a4-b96b-4d277cac8b6e --owner-id account_id --repository-name MyRepo
```

2. 이 명령은 다음 출력을 반환합니다.

```
{
  "RepositoryLinkInfo": {
    "ConnectionArn": "arn:aws:codestar-connections:us-east-1:account_id:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
    "OwnerId": "account_id",
    "ProviderType": "GitHub",
    "RepositoryLinkArn": "arn:aws:codestar-connections:us-east-1:account_id:repository-link/be8f2017-b016-4a77-87b4-608054f70e77",
    "RepositoryLinkId": "be8f2017-b016-4a77-87b4-608054f70e77",
    "RepositoryName": "MyRepo",
    "Tags": []
  }
}
```

리포지토리 링크 업데이트

AWS Command Line Interface (AWS CLI) 의 `update-repository-link` 명령을 사용하여 지정된 저장소 링크를 업데이트할 수 있습니다.

리포지토리 링크에 대한 다음 정보를 업데이트할 수 있습니다.

- `--connection-arn`

- --owner-id
- --repository-name

리포지토리와 관련된 연결을 변경하려는 경우 리포지토리 링크를 업데이트할 수 있습니다. 다른 연결을 사용하려면 연결 ARN을 지정해야 합니다. 연결 ARN을 보는 단계는 [연결 세부 정보 보기](#)를 참조하세요.

리포지토리 링크를 업데이트하려면

1. 터미널(Linux, macOS, Unix) 또는 명령 프롬프트(Windows)를 엽니다. AWS CLI 를 사용하여 update-repository-link 명령을 실행하고 저장소 링크에 대해 업데이트할 값을 지정합니다. 예를 들어 다음 명령은 리포지토리 링크 ID와 관련된 연결을 업데이트합니다. --connection 파라미터를 사용하여 새 연결 ARN을 지정합니다.

```
aws codestar-connections update-repository-link --repository-link-id
6053346f-8a33-4edb-9397-10394b695173 --connection-arn arn:aws:codestar-
connections:us-east-1:account_id:connection/aEXAMPLE-f055-4843-adeb-4ceaefcb2167
```

2. 이 명령은 다음 출력을 반환합니다.

```
{
  "RepositoryLinkInfo": {
    "ConnectionArn": "arn:aws:codestar-connections:us-
east-1:account_id:connection/aEXAMPLE-f055-4843-adeb-4ceaefcb2167",
    "OwnerId": "owner_id",
    "ProviderType": "GitHub",
    "RepositoryLinkArn": "arn:aws:codestar-connections:us-
east-1:account_id:repository-link/6053346f-8a33-4edb-9397-10394b695173",
    "RepositoryLinkId": "6053346f-8a33-4edb-9397-10394b695173",
    "RepositoryName": "MyRepo",
    "Tags": []
  }
}
```

리포지토리 링크 나열

AWS Command Line Interface (AWS CLI) 의 list-repository-links 명령을 사용하여 계정의 저장소 링크를 나열할 수 있습니다.

리포지토리 링크를 나열하려면

1. 터미널(Linux, macOS, Unix) 또는 명령 프롬프트(Windows)를 엽니다. 를 AWS CLI 사용하여 list-repository-links 명령을 실행합니다.

```
aws codestar-connections list-repository-links
```

2. 이 명령은 다음 출력을 반환합니다.

```
{
  "RepositoryLinks": [
    {
      "ConnectionArn": "arn:aws:codestar-connections:us-
east-1:account_id:connection/001f5be2-a661-46a4-b96b-4d277cac8b6e",
      "OwnerId": "owner_id",
      "ProviderType": "GitHub",
      "RepositoryLinkArn": "arn:aws:codestar-connections:us-
east-1:account_id:repository-link/6053346f-8a33-4edb-9397-10394b695173",
      "RepositoryLinkId": "6053346f-8a33-4edb-9397-10394b695173",
      "RepositoryName": "MyRepo",
      "Tags": []
    }
  ]
}
```

리포지토리 링크 삭제

AWS Command Line Interface (AWS CLI) 의 delete-repository-link 명령을 사용하여 저장소 링크를 삭제할 수 있습니다.

리포지토리 링크를 삭제하려면 먼저 리포지토리 링크와 관련된 모든 동기화 구성을 삭제해야 합니다.

Important

이 명령을 실행하면 리포지토리 링크가 삭제됩니다. 확인 대화 상자는 표시되지 않습니다. 새 리포지토리 링크를 생성할 수 있지만 Amazon 리소스 이름(ARN)은 재사용되지 않습니다.

리포지토리 링크를 삭제하려면

- 터미널(Linux, macOS, Unix) 또는 명령 프롬프트(Windows)를 엽니다. AWS CLI 를 사용하여 delete-repository-link 명령을 실행하고 삭제할 저장소 링크의 ID를 지정합니다.

```
aws codestar-connections delete-repository-link --repository-link-id
6053346f-8a33-4edb-9397-10394b695173
```

이 명령은 아무 것도 반환하지 않습니다.

리포지토리 링크 세부 정보 보기

AWS Command Line Interface (AWS CLI) 의 get-repository-link 명령을 사용하여 저장소 링크에 대한 세부 정보를 볼 수 있습니다.

리포지토리 링크 세부 정보를 보려면

- 터미널(Linux, macOS, Unix) 또는 명령 프롬프트(Windows)를 엽니다. AWS CLI 를 사용하여 get-repository-link 명령을 실행하고 저장소 링크 ID를 지정합니다.

```
aws codestar-connections get-repository-link --repository-link-id
6053346f-8a33-4edb-9397-10394b695173
```

- 이 명령은 다음 출력을 반환합니다.

```
{
  "RepositoryLinkInfo": {
    "ConnectionArn": "arn:aws:codestar-connections:us-
east-1:account_id:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
    "OwnerId": "owner_id",
    "ProviderType": "GitHub",
    "RepositoryLinkArn": "arn:aws:codestar-connections:us-
east-1:account_id:repository-link/be8f2017-b016-4a77-87b4-608054f70e77",
    "RepositoryLinkId": "6053346f-8a33-4edb-9397-10394b695173",
    "RepositoryName": "MyRepo",
    "Tags": []
  }
}
```

동기화 구성 작업

동기화 구성은 지정된 리포지토리와 연결 간의 링크를 생성합니다. 동기화 구성을 사용해 Git 리포지토리의 콘텐츠를 동기화하여 지정된 AWS 리소스를 업데이트하세요.

연결에 대한 자세한 내용은 [AWS CodeStar 연결 API](#) 참조를 참조하십시오.

주제

- [동기화 구성 생성](#)
- [동기화 구성 업데이트](#)
- [동기화 구성 나열](#)
- [동기화 구성 삭제](#)
- [동기화 구성 세부 정보 보기](#)

동기화 구성 생성

AWS Command Line Interface (AWS CLI) 의 `create-repository-link` 명령을 사용하여 연결과 동기화할 외부 저장소 사이에 링크를 만들 수 있습니다.

동기화 구성을 생성하려면 먼저 사용자 연결과 서드 파티 리포지토리 간에 리포지토리 링크를 이미 생성해야 합니다.

동기화 구성을 생성하려면

1. 터미널(Linux, macOS, Unix) 또는 명령 프롬프트(Windows)를 엽니다. 를 AWS CLI 사용하여 `create-repository-link` 명령을 실행합니다. 관련 연결의 ARN, 소유자 ID, 리포지토리 이름을 지정합니다. 다음 명령은 AWS CloudFormation의 리소스에 대한 동기화 유형을 사용하여 동기화 구성을 생성합니다. 또한 리포지토리의 리포지토리 브랜치 및 구성 파일을 지정합니다. 이 예제에서 리소스는 **mystack**이라는 이름의 스택입니다.

```
aws codestar-connections create-sync-configuration --branch main --config-file filename --repository-link-id be8f2017-b016-4a77-87b4-608054f70e77 --resource-name mystack --role-arn arn:aws:iam::account_id:role/myrole --sync-type CFN_STACK_SYNC
```

2. 이 명령은 다음 출력을 반환합니다.

```
{
  "SyncConfiguration": {
    "Branch": "main",
```

```

    "ConfigFile": "filename",
    "OwnerId": "account_id",
    "ProviderType": "GitHub",
    "RepositoryLinkId": "be8f2017-b016-4a77-87b4-608054f70e77",
    "RepositoryName": "MyRepo",
    "ResourceName": "mystack",
    "RoleArn": "arn:aws:iam::account_id:role/myrole",
    "SyncType": "CFN_STACK_SYNC"
  }

```

동기화 구성 업데이트

AWS Command Line Interface (AWS CLI)에서 `update-sync-configuration` 명령을 사용하여 지정된 동기화 구성을 업데이트할 수 있습니다.

동기화 구성에 대한 다음 정보를 업데이트할 수 있습니다.

- `--branch`
- `--config-file`
- `--repository-link-id`
- `--resource-name`
- `--role-arn`

동기화 구성을 업데이트하려면

1. 터미널(Linux, macOS, Unix) 또는 명령 프롬프트(Windows)를 엽니다. AWS CLI 를 사용하여 리소스 이름 및 동기화 유형과 함께 업데이트할 값을 지정하여 `update-sync-configuration` 명령을 실행합니다. 예를 들어, 다음 명령은 `--branch` 파라미터를 사용하여 동기화 구성과 관련된 브랜치 이름을 업데이트합니다.

```
aws codestar-connections update-sync-configuration --sync-type CFN_STACK_SYNC --resource-name mystack --branch feature-branch
```

2. 이 명령은 다음 출력을 반환합니다.

```

{
  "SyncConfiguration": {
    "Branch": "feature-branch",
    "ConfigFile": "filename.yaml",

```

```

"OwnerId": "owner_id",
"ProviderType": "GitHub",
"RepositoryLinkId": "6053346f-8a33-4edb-9397-10394b695173",
"RepositoryName": "MyRepo",
"ResourceName": "mystack",
"RoleArn": "arn:aws:iam::account_id:role/myrole",
"SyncType": "CFN_STACK_SYNC"
}

```

동기화 구성 나열

AWS Command Line Interface (AWS CLI)에서 list-sync-configurations 명령을 사용하여 계정의 리포지토리 링크를 나열할 수 있습니다.

리포지토리 링크를 나열하려면

1. 터미널(Linux, macOS, Unix) 또는 명령 프롬프트(Windows)를 엽니다. AWS CLI 를 사용하여 list-sync-configurations 명령을 실행하고 동기화 유형과 리포지토리 링크 ID를 지정합니다.

```

aws codestar-connections list-sync-configurations --repository-link-id
6053346f-8a33-4edb-9397-10394b695173 --sync-type CFN_STACK_SYNC

```

2. 이 명령은 다음 출력을 반환합니다.

```

{
  "SyncConfigurations": [
    {
      "Branch": "main",
      "ConfigFile": "filename.yaml",
      "OwnerId": "owner_id",
      "ProviderType": "GitHub",
      "RepositoryLinkId": "6053346f-8a33-4edb-9397-10394b695173",
      "RepositoryName": "MyRepo",
      "ResourceName": "mystack",
      "RoleArn": "arn:aws:iam::account_id:role/myrole",
      "SyncType": "CFN_STACK_SYNC"
    }
  ]
}

```

동기화 구성 삭제

AWS Command Line Interface (AWS CLI)에서 delete-sync-configuration 명령을 사용하여 지정된 동기화 구성을 삭제할 수 있습니다.

Important

명령을 실행하면 동기화 구성이 삭제됩니다. 확인 대화 상자는 표시되지 않습니다. 새 동기화 구성을 생성할 수 있지만 Amazon 리소스 이름(ARN)은 재사용되지 않습니다.

동기화 구성을 삭제하려면

- 터미널(Linux, macOS, Unix) 또는 명령 프롬프트(Windows)를 엽니다. AWS CLI 를 사용하여 delete-sync-configuration 명령을 실행하고 삭제하려는 동기화 구성의 동기화 유형과 리소스 이름을 지정합니다.

```
aws codestar-connections delete-sync-configuration --sync-type CFN_STACK_SYNC --resource-name mystack
```

이 명령은 아무 것도 반환하지 않습니다.

동기화 구성 세부 정보 보기

AWS Command Line Interface (AWS CLI) 의 get-sync-configuration 명령을 사용하여 동기화 구성의 세부 정보를 볼 수 있습니다.

동기화 구성에 대한 세부 정보를 보려면

- 터미널(Linux, macOS, Unix) 또는 명령 프롬프트(Windows)를 엽니다. AWS CLI 를 사용하여 get-sync-configuration 명령을 실행하고 리포지토리 링크 ID를 지정합니다.

```
aws codestar-connections get-sync-configuration --sync-type CFN_STACK_SYNC --resource-name mystack
```

- 이 명령은 다음 출력을 반환합니다.

```
{
  "SyncConfiguration": {
    "Branch": "main",
```

```
"ConfigFile": "filename",
"OwnerId": "owner_id",
"ProviderType": "GitHub",
"RepositoryLinkId": "be8f2017-b016-4a77-87b4-608054f70e77",
"RepositoryName": "MyRepo",
"ResourceName": "mystack",
"RoleArn": "arn:aws:iam::account_id:role/myrole",
"SyncType": "CFN_STACK_SYNC"
}
}
```

AWS CloudTrail을 사용하여 AWS CodeConnections API 호출 로깅

AWS CodeConnections는 사용자, 역할, 또는 AWS 서비스가 수행한 작업에 대한 레코드를 제공하는 서비스인 AWS CloudTrail과 통합됩니다. CloudTrail은 알림을 제공하기 위해 API 호출을 이벤트로 캡처합니다. 캡처되는 호출에는 개발자 도구 콘솔에서 수행한 호출과 AWS CodeConnections API 작업에 대한 코드 호출이 포함됩니다.

추적을 생성하면 알림을 위한 이벤트를 포함한 CloudTrail 이벤트를 Amazon Simple Storage Service(Amazon S3) 버킷에 지속적으로 전송할 수 있습니다. 추적을 구성하지 않은 경우에도 CloudTrail 콘솔의 이벤트 기록에서 최신 이벤트를 볼 수 있습니다. CloudTrail에서 수집한 정보를 사용하여 AWS CodeConnections에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 기타 세부 정보를 확인할 수 있습니다.

자세한 정보는 [AWS CloudTrail 사용 설명서](#)를 참조하세요.

CloudTrail의 AWS CodeConnections 정보

CloudTrail은 계정 생성 시 AWS 계정에서 사용되도록 설정됩니다. AWS CodeConnections에서 활동이 발생하면 해당 활동이 [이벤트 기록(Event history)]의 다른 AWS 서비스 이벤트와 함께 CloudTrail 이벤트에 기록됩니다. AWS 계정에서 최신 이벤트를 확인, 검색 및 다운로드할 수 있습니다. 자세한 내용은 AWS CloudTrail 사용 설명서에서 [CloudTrail 이벤트 기록을 사용하여 이벤트 보기](#)를 참조하세요.

AWS CodeConnections에 대한 이벤트를 포함하여 AWS 계정에 이벤트를 지속적으로 기록하려면 추적을 생성합니다. CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 콘솔에서 추적을 생성하면 기본적으로 모든 AWS 리전에 추적이 적용됩니다. 추적은 AWS 파티션에 있는 모든 리전의 이벤트를 로깅하고 지정된 Amazon S3 버킷으로 로그 파일을 전송합니다. 또는 CloudTrail 로그에서 수집된 이벤트 데이터를 추가 분석 및 처리하도록 다른 AWS 서비스를 구성할 수 있습니다.

자세한 내용은 AWS CloudTrail 사용 설명서에서 다음 주제를 참조하십시오.

- [추적 생성 개요](#)
- [CloudTrail 지원 서비스 및 통합](#)
- [CloudTrail에 대한 Amazon SNS 알림 구성](#)
- [여러 리전에서 CloudTrail 로그 파일 수신](#)
- [여러 계정에서 CloudTrail 로그 파일 수신](#)

모든 AWS CodeConnections 작업은 CloudTrail에서 로깅되고 [AWS CodeConnections API 참조](#)에 기록됩니다. 예를 들어 CreateConnection, DeleteConnection, GetConnection 작업을 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.

- 루트를 통해서 요청되었는지, 또는 다른 IAM 사용자 보안 인증을 통해 요청되었는지 여부.
- 역할 또는 페더레이션 사용자에게 대한 임시 보안 자격 증명을 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청했는지 여부.

자세한 내용은 [CloudTrail userIdentity 요소](#)를 참조하세요.

로그 파일 항목 이해

추적이란 지정한 Amazon S3 버킷에 이벤트를 로그 파일로 입력할 수 있게 하는 구성입니다.

CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함될 수 있습니다. 이벤트는 모든 소스로부터의 단일 요청을 나타내며 요청 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보가 들어 있습니다. CloudTrail 로그 파일은 퍼블릭 API 호출의 주문 스택 트레이스가 아니므로 특정 순서로 표시되지 않습니다.

다음은 CreateConnection 작업을 보여주는 CloudTrail 로그 항목이 나타낸 예제입니다.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/Mary_Major",
    "accountId": "123456789012",
```



```
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "Mary_Major"
  },
  "eventTime": "2020-04-21T01:09:48Z",
  "eventSource": "codestar-connections.amazonaws.com",
  "eventName": "CreateConnection",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "IP",
  "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/80.0.3987.163 Safari/537.36",
  "requestParameters": {
    "providerType": "Bitbucket",
    "connectionName": "my-connection"
  },
  "responseElements": {
    "connectionArn": "arn:aws:codestar-connections:us-
west-2:123456789012:connection/7EXAMPLE-5da1-4867-960c-4918175ea3ce"
  },
  "requestID": "ac1fbc15-a84f-4568-9f90-f05f1a57749c",
  "eventID": "7548f5b0-7ecf-430f-84bf-72e364644359",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012"
}
```

AWS CodeStar Connections 및 인터페이스 VPC 엔드포인트(AWS PrivateLink)

인터페이스 VPC 엔드포인트를 생성하여 VPC와 AWS CodeStar Connections 간에 프라이빗 연결을 설정할 수 있습니다. 인터페이스 엔드포인트는 인터넷 게이트웨이, NAT 디바이스, VPN 연결 또는 AWS Direct Connect 연결 없이 비공개로 AWS CodeStar Connections API에 액세스할 수 있도록 지원하는 [AWS PrivateLink](#) 기술로 구동됩니다. VPC와 AWS CodeStar Connections 간의 트래픽은 Amazon 네트워크를 벗어나지 않으므로, VPC의 인스턴스가 AWS CodeStar Connections API와 통신하는 데에는 퍼블릭 IP 주소가 필요하지 않습니다.

각 인터페이스 엔드포인트는 서브넷에서 하나 이상의 [탄력적 네트워크 인터페이스](#)로 표현됩니다.

자세한 내용은 Amazon VPC 사용 설명서에서 [인터페이스 VPC 종단점\(AWS PrivateLink\)](#)을 참조하세요.

AWS CodeStar Connections VPC 엔드포인트에 대한 고려 사항

AWS CodeStar Connections에 대한 인터페이스 VPC 엔드포인트를 설정하기 전에 Amazon VPC 사용 설명서에서 [인터페이스 엔드포인트](#)를 검토해야 합니다.

AWS CodeStar Connections는 VPC에서 모든 API 작업에 대한 호출 수행을 지원합니다.

VPC 엔드포인트는 모든 AWS CodeStar Connections 리전에서 지원됩니다.

VPC 엔드포인트 개념

다음은 VPC 엔드포인트의 핵심 개념입니다.

VPC 엔드포인트

서비스에 비공개로 연결할 수 있는 VPC의 진입점입니다. 다음은 다양한 유형의 VPC 엔드포인트입니다. 지원되는 서비스에서 요구하는 유형의 VPC 엔드포인트를 생성합니다.

- [AWS CodeStar Connections 작업을 위한 VPC 엔드포인트](#)
- [AWS CodeStar Connections 웹후크를 위한 VPC 엔드포인트](#)

AWS PrivateLink

VPC와 서비스 간에 프라이빗 연결을 제공하는 기술입니다.

AWS CodeStar Connections 작업을 위한 VPC 엔드포인트

AWS CodeStar Connections 서비스에 대한 VPC 엔드포인트를 관리할 수 있습니다.

AWS CodeStar Connections 작업을 위한 인터페이스 VPC 엔드포인트 생성

Amazon VPC 콘솔 또는 AWS Command Line Interface(AWS CLI)를 사용하여 AWS CodeStar Connections 서비스에 대한 VPC 엔드포인트를 생성할 수 있습니다. 자세한 내용은 Amazon VPC 사용 설명서의 [인터페이스 엔드포인트 생성](#)을 참조하세요.

VPC와의 연결을 사용하여 시작하려면 AWS CodeStar Connections를 위한 인터페이스 VPC 엔드포인트를 생성합니다. AWS CodeStar Connections를 위한 VPC 엔드포인트를 생성할 때 AWS Services(AWS 서비스)를 선택하고 Service Name(서비스 이름)에서 다음을 선택합니다.

- `com.amazonaws.region.codestar-connections.api`: 이 옵션은 AWS CodeStar Connections API 작업을 위한 VPC 엔드포인트를 생성합니다. 예를 들어 사용자가 `CreateConnection`,

ListConnections 및 CreateHost 등의 작업을 위해 AWS CLI, AWS CodeStar Connections API 또는 AWS SDK를 사용하여 AWS CodeStar Connections와 상호 작용하는 경우 이 옵션을 선택합니다.

Enable DNS name(DNS 이름 사용) 옵션에서 엔드포인트에 대해 프라이빗 DNS를 선택하는 경우, 리전에 대한 기본 DNS 이름(예: `codestar-connections.us-east-1.amazonaws.com`)을 사용하여 AWS CodeStar Connections에 대해 API 요청을 수행할 수 있습니다.

Important

프라이빗 DNS는 AWS 서비스 및 AWS Marketplace 파트너 서비스용으로 생성한 엔드포인트에 대해 기본적으로 활성화됩니다.

자세한 내용은 Amazon VPC 사용 설명서의 [인터페이스 엔드포인트를 통해 서비스 액세스](#)를 참조하세요.

AWS CodeStar Connections 작업을 위한 VPC 엔드포인트 정책 생성

AWS CodeStar Connections에 대한 액세스를 제어하는 VPC 엔드포인트에 엔드포인트 정책을 연결할 수 있습니다. 이 정책은 다음 정보를 지정합니다.

- 태스크를 수행할 수 있는 보안 주체.
- 수행할 수 있는 작업입니다.
- 태스크를 수행할 있는 리소스.

자세한 정보는 Amazon VPC 사용 설명서의 [VPC 엔드포인트를 통해 서비스에 대한 액세스 제어](#)를 참조하세요.

Note

`com.amazonaws.region.codestar-connections.webhooks` 엔드포인트는 정책을 지원하지 않습니다.

예: AWS CodeStar Connections 작업을 위한 VPC 엔드포인트 정책

다음은 AWS CodeStar Connections에 대한 엔드포인트 정책의 예입니다. 이 정책은 엔드포인트에 연결될 때 모든 리소스의 모든 보안 주체에 대한 액세스 권한을 나열된 AWS CodeStar Connections 작업에 부여합니다.

```
{
  "Statement": [
    {
      "Sid": "GetConnectionOnly",
      "Principal": "*",
      "Action": [
        "codestar-connections:GetConnection"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

AWS CodeStar Connections 웹후크를 위한 VPC 엔드포인트

AWS CodeStar Connections는 VPC 구성이 있는 호스트를 생성하거나 삭제할 때 웹후크 엔드포인트를 생성합니다. 엔드포인트 이름은 `com.amazonaws.region.codestar-connections.webhooks`입니다.

호스트는 GitHub 웹후크용 VPC 엔드포인트를 사용하여 웹후크를 통해 Amazon 네트워크에서 통합 AWS 서비스로 이벤트 데이터를 전송할 수 있습니다.

Important

GitHub Enterprise Server용으로 호스트를 설정하면 AWS CodeStar Connections가 웹후크 이벤트 데이터를 위한 VPC 엔드포인트를 자동으로 생성합니다. 2020년 11월 24일 이전에 호스트를 생성한 경우 VPC PrivateLink 웹후크 엔드포인트를 사용하려면 먼저 호스트를 [삭제](#)한 다음 새 호스트를 [생성](#)해야 합니다.

AWS CodeStar Connections는 이러한 엔드포인트의 수명 주기를 관리합니다. 엔드포인트를 삭제하려면 해당 호스트 리소스를 삭제해야 합니다.

AWS CodeStar Connections 호스트에 웹후크 엔드포인트를 사용하는 방법

웹후크 엔드포인트는 AWS CodeStar Connections 프로세싱을 위해 서드 파티 리포지토리의 웹후크가 전송되는 곳입니다. 웹후크는 고객 작업을 설명합니다. `git push` 작업을 수행할 때 웹후크 엔드포

인트는 공급자로부터 푸시 작업을 자세히 설명하는 웹후크를 수신합니다. 예를 들어 AWS CodeStar Connections는 파이프라인을 시작하도록 CodePipeline에 알릴 수 있습니다.

VPC를 사용하지 않는 Bitbucket 호스트, GitHub Enterprise Server 호스트 등의 클라우드 공급자의 경우 Amazon 네트워크를 사용하지 않는 AWS CodeStar Connections로 웹후크를 전송하므로 웹후크 VPC 엔드포인트가 적용되지 않습니다.

연결 문제 해결

다음 정보는 AWS CodeBuild, AWS CodeDeploy, AWS CodePipeline에서 리소스 연결과 관련한 일반적인 문제를 해결하는 데 도움이 됩니다.

주제

- [연결을 생성할 수 없음](#)
- [연결을 생성하거나 완료하려고 할 때 권한 오류 발생](#)
- [연결을 사용하려고 할 때 권한 오류 발생](#)
- [연결이 사용 가능한 상태가 아니거나 더 이상 보류 중 상태가 아닙니다.](#)
- [연결에 대한 GitClone 권한 추가](#)
- [호스트가 사용 가능한 상태가 아닙니다.](#)
- [연결 오류가 있는 호스트 문제 해결](#)
- [호스트에 대한 연결을 생성할 수 없습니다.](#)
- [호스트의 VPC 구성 문제 해결](#)
- [GitHub Enterprise Server 연결을 위한 웹후크 VPC 엔드포인트\(PrivateLink\) 문제 해결](#)
- [2020년 11월 24일 이전에 생성된 호스트의 문제 해결](#)
- [GitHub 리포지토리에 대한 연결을 생성할 수 없음](#)
- [GitHub Enterprise Server 연결 앱 권한 편집](#)
- [GitHub에 연결할 때 연결 오류 발생: "A problem occurred, make sure cookies are enabled in your browser" 또는 "An organization owner must install the GitHub app"](#)
- [연결의 한도를 늘리려는 경우](#)

연결을 생성할 수 없음

연결을 생성할 권한이 없을 수 있습니다. 자세한 내용은 [다음에 대한 권한 및 예 AWS CodeConnections](#) 섹션을 참조하세요.

연결을 생성하거나 완료하려고 할 때 권한 오류 발생

CodePipeline 콘솔에서 연결을 생성하거나 보려고 하면 다음 오류 메시지가 반환될 수 있습니다.

사용자: *username*에게 *connection-ARN* 리소스에 대해 *permission* 작업을 수행할 권한이 없음
(User: username is not authorized to perform: permission on resource: connection-ARN)

이 메시지가 나타나면 적절한 권한이 있는지 확인하세요.

AWS Command Line Interface(AWS CLI) 또는 AWS Management Console에서 연결을 생성하고 표시할 권한은 콘솔에서 연결을 생성하고 완료하는 데 필요한 권한의 일부일 뿐입니다. 단순히 연결을 표시하거나 편집하거나 생성한 후 보류 중인 연결을 완료하는 데 필요한 권한은 특정 태스크를 수행해야 하는 사용자로 그 적용 범위를 제한해야 합니다. 자세한 내용은 [다음에 대한 권한 및 예 AWS CodeConnections](#) 섹션을 참조하세요.

연결을 사용하려고 할 때 권한 오류 발생

나열, 가져오기 및 생성 권한이 있더라도, CodePipeline 콘솔에서 연결을 사용하려고 하면 다음 오류 메시지 중 하나 또는 둘 모두 반환될 수 있습니다.

계정을 인증하지 못했습니다.

사용자: *username*에게 *connection-ARN* 리소스에 대해 *codestar-connections:UseConnection* 작업을 수행할 권한이 없음(User: username is not authorized to perform: *codestar-connections:UseConnection* on resource: connection-ARN)

이 문제가 발생할 경우 적절한 권한이 있는지 확인하세요.

공급자 위치에서 사용 가능한 리포지토리를 나열하는 등, 연결을 사용할 권한이 있는지 확인합니다. 자세한 내용은 [다음에 대한 권한 및 예 AWS CodeConnections](#) 섹션을 참조하세요.

연결이 사용 가능한 상태가 아니거나 더 이상 보류 중 상태가 아닙니다.

콘솔에 연결이 사용 가능한 상태가 아니라는 메시지가 표시되면 [연결 완료(Complete connection)]를 선택합니다.

연결을 완료하도록 선택한 경우 연결이 보류 상태가 아니라는 메시지가 나타나면, 연결이 이미 사용 가능한 상태이므로 요청을 취소할 수 있습니다.

연결에 대한 GitClone 권한 추가

소스 작업과 CodeBuild 작업에서 AWS CodeStar 연결을 사용하는 경우 다음 두 가지 방법으로 입력 아티팩트를 빌드에 전달할 수 있습니다.

- 기본 방법: 소스 작업이 CodeBuild가 다운로드하는 코드가 포함된 zip 파일을 생성합니다.
- Git 복제: 빌드 환경에서 소스 코드를 직접 다운로드할 수 있습니다.

Git 복제 모드를 사용하면 작업 Git 리포지토리로 소스 코드와 상호 작용할 수 있습니다. 이 모드를 사용하려면 연결을 사용할 수 있는 권한을 CodeBuild 환경에 부여해야 합니다.

CodeBuild 서비스 역할 정책에 권한을 추가하려면 CodeBuild 서비스 역할에 연결되는 고객 관리형 정책을 만듭니다. 다음 단계에서는 action 필드에 UseConnection 권한이 지정되고 Resource 필드에 연결 Amazon 리소스 이름(ARN)이 지정된 정책을 만듭니다.

콘솔을 사용하여 UseConnection 권한을 추가하려면

1. 파이프라인을 열고 소스 작업에서 (i) 아이콘을 선택하여 파이프라인의 연결 ARN을 찾습니다. [구성(Configuration)] 창이 열리고 연결 ARN이 [ConnectionArn] 옆에 표시됩니다. CodeBuild 서비스 역할 정책에 연결 ARN을 추가합니다.
2. CodeBuild 서비스 역할을 찾으려면 파이프라인에 사용된 빌드 프로젝트를 열고 [빌드 세부 정보(Build details)] 탭으로 이동합니다.
3. [환경(Environment)] 섹션에서 [서비스 역할(Service role)] 링크를 선택합니다. 그러면 연결에 대한 액세스 권한을 부여하는 새 정책을 추가할 수 있는 AWS Identity and Access Management(IAM) 콘솔이 열립니다.
4. IAM 콘솔에서 Attach policies(정책 연결)를 선택하고 Create policy(정책 생성)를 선택합니다.

다음 샘플 정책 템플릿을 사용합니다. 다음 예와 같이 Resource 필드에 연결 ARN을 추가합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codestar-connections:UseConnection",
      "Resource": "insert connection ARN here"
    }
  ]
}
```

JSON 탭에서 정책을 붙여 넣습니다.

5. Review policy(정책 검토)를 선택합니다. 정책 이름(예: **connection-permissions**)을 입력한 후 Create policy(정책 생성)를 선택합니다.

6. 서비스 역할 [권한 연결(Attach Permissions)] 페이지로 돌아가서 정책 목록을 새로 고친 다음 방금 생성한 정책을 선택합니다. 정책 연결(Attach policies)을 선택합니다.

호스트가 사용 가능한 상태가 아닙니다.

호스트가 Available 상태가 아니라는 메시지가 콘솔에 표시될 경우 [호스트 설정(Set up host)]을 선택합니다.

호스트 생성을 위한 첫 단계를 완료하면 Pending 상태로 호스트가 생성됩니다. 호스트를 Available 상태로 전환하려면 콘솔에서 호스트를 설정하도록 선택해야 합니다. 자세한 내용은 [보류 중인 호스트 설정](#) 섹션을 참조하세요.

Note

Pending 호스트를 설정하는 데는 AWS CLI를 사용할 수 없습니다.

연결 오류가 있는 호스트 문제 해결

기본 GitHub 앱이 삭제되거나 수정되면 연결 및 호스트가 오류 상태로 전환될 수 있습니다. 오류 상태의 호스트와 연결은 복구할 수 없으며, 해당 호스트를 다시 생성해야 합니다.

- 앱 pem 키 변경, 앱 이름 변경(초기 생성 후) 등의 작업을 수행할 경우 호스트 및 연결된 모든 연결이 오류 상태로 전환됩니다.

콘솔 또는 CLI에서 호스트 또는 호스트와 관련된 연결이 Error 상태로 반환하는 경우, 다음 단계를 수행해야 할 수 있습니다.

- 호스트 리소스를 삭제하고 다시 생성한 다음 호스트 등록 앱을 다시 설치합니다. 자세한 내용은 [호스트 생성](#) 섹션을 참조하세요.

호스트에 대한 연결을 생성할 수 없습니다.

연결 또는 호스트를 생성하려면 다음 조건이 충족되어야 합니다.

- 호스트가 AVAILABLE 상태입니다. 자세한 내용은 다음을 참조하세요.
- 연결이 호스트와 동일한 리전에 생성되어 있어야 합니다.

호스트의 VPC 구성 문제 해결

호스트 리소스를 생성할 때 GitHub Enterprise Server 인스턴스가 설치된 인프라에 대한 네트워크 연결 또는 VPC 정보를 제공해야 합니다. 호스트의 VPC 또는 서브넷 구성 문제를 해결하려면 여기에 나와 있는 VPC 정보의 예를 참조하세요.

Note

Amazon VPC 내에서 GitHub Enterprise Server 호스트 구성과 관련된 문제를 해결하는 데 이 섹션을 참조하세요. VPC에 웹훅 엔드포인트(PrivateLink)를 사용하도록 구성된 연결과 관련한 문제 해결은 [GitHub Enterprise Server 연결을 위한 웹훅 VPC 엔드포인트\(PrivateLink\) 문제 해결](#) 섹션을 참조하세요.

이 예에서는 다음 프로세스를 사용하여 GitHub Enterprise Server 인스턴스가 설치될 VPC 및 서버를 구성합니다.

1. VPC를 생성합니다. 자세한 내용은 <https://docs.aws.amazon.com/vpc/latest/userguide/working-with-vpcs.html#Create-VPC> 섹션을 참조하세요.
2. VPC에서 서브넷 생성 자세한 내용은 <https://docs.aws.amazon.com/vpc/latest/userguide/working-with-vpcs.html#AddSubnet> 섹션을 참조하세요.
3. VPC로 인스턴스를 시작합니다. 자세한 내용은 https://docs.aws.amazon.com/vpc/latest/userguide/working-with-vpcs.html#VPC_Launch_Instance 섹션을 참조하세요.

Note

각 VPC는 한 번에 하나의 호스트(GitHub 엔터프라이즈 서버 인스턴스)에만 연결 가능합니다.

다음 이미지는 GitHub Enterprise AMI를 사용하여 시작된 EC2 인스턴스를 보여줍니다.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
GitHub Enterprise	i-0b4441c7242dfd867	m5.xlarge	us-east-2b	running	2/2 checked

Instance: **i-0b4441c7242dfd867 (GitHub Enterprise)** Elastic IP: [REDACTED]

Description | Status Checks | Monitoring | Tags

Instance ID	i-0b4441c7242dfd867	Public DNS (IPv4)	ec2-...-us-east-2.compute.amazonaws.com
Instance state	running	IPv4 Public IP	[REDACTED]
Instance type	m5.xlarge	IPv6 IPs	-
Finding	Opt-in to AWS Compute Optimizer for recommendations. Learn more	Elastic IPs	[REDACTED]
Private DNS	ip-[REDACTED]-us-east-2.compute.internal	Availability zone	us-east-2b
Private IPs	[REDACTED]	Security groups	ghe-InstanceSecurityGroup-1IEZ3GYA4DVN6 , view inbound rules , view outbound rules
Secondary private IPs		Scheduled events	No scheduled events
VPC ID	vpc-a04993cb	AMI ID	GitHub Enterprise Server 2.20.9
Subnet ID	subnet-75350e0f	Platform details	Linux/UNIX
Network interfaces	eth0	Usage operation	RunInstances
IAM role	ghe-EC2InstanceRole-1OHLRWYXR1RHR	Source/dest. check	True

GitHub Enterprise Server 연결에 VPC를 사용하는 경우 호스트를 설정할 때 인프라에 대해 다음 정보를 제공해야 합니다.

- VPC ID: GitHub Enterprise Server 인스턴스가 설치된 서버의 VPC ID이거나, VPN 또는 Direct Connect를 통해 설치된 GitHub Enterprise Server 인스턴스에 액세스할 수 있는 VPC입니다.
- 서브넷 ID 또는 ID: GitHub Enterprise Server 인스턴스가 설치된 서버의 서브넷이거나, VPN 또는 Direct Connect를 통해 설치된 GitHub Enterprise Server 인스턴스에 액세스할 수 있는 서브넷입니다.
- 보안 그룹 또는 그룹: GitHub Enterprise Server 인스턴스가 설치된 서버의 보안 그룹이거나, VPN 또는 Direct Connect를 통해 설치된 GitHub Enterprise Server 인스턴스에 액세스할 수 있는 보안 그룹입니다.
- 엔드포인트: 서버 엔드포인트를 준비하고 다음 단계로 계속 진행합니다.

VPC 및 서브넷 작업에 대한 자세한 내용은 Amazon VPC 사용 설명서에서 [IPv4의 경우, VPC 및 서브넷 크기 조정](#)을 참조하세요.

주제

- [보류 중인 상태의 호스트를 가져올 수 없음](#)
- [사용 가능한 상태의 호스트를 가져올 수 없음](#)
- [연결/호스트가 작동하다가 현재 작동이 중지됨](#)
- [네트워크 인터페이스를 삭제할 수 없음](#)

보류 중인 상태의 호스트를 가져올 수 없음

호스트가 VPC_CONFIG_FAILED_INITIALIZATION 상태로 전환되면 이는 호스트에 대해 선택한 VPC, 서브넷 또는 보안 그룹에 문제가 있기 때문일 수 있습니다.

- VPC, 서브넷, 보안 그룹은 모두 호스트를 생성하는 계정에 속해야 합니다.
- 서브넷과 보안 그룹은 선택한 VPC에 속해야 합니다.
- 제공한 각 서브넷은 서로 다른 가용 영역에 있어야 합니다.
- 호스트를 생성하는 사용자에게 IAM 권한이 있어야 합니다.

```
ec2:CreateNetworkInterface
ec2:CreateTags
ec2:DescribeDhcpOptionsec2:DescribeNetworkInterfaces
ec2:DescribeSubnets
ec2>DeleteNetworkInterface
ec2:DescribeVpcs
ec2:CreateVpcEndpoint
ec2>DeleteVpcEndpoints
ec2:DescribeVpcEndpoints
```

사용 가능한 상태의 호스트를 가져올 수 없음

호스트에 대한 AWS CodeStar Connections 앱 설정을 완료할 수 없는 경우, VPC 구성 또는 GitHub Enterprise Server 인스턴스에 문제가 있기 때문일 수 있습니다.

- 퍼블릭 인증 기관을 사용하지 않는 경우 GitHub Enterprise 인스턴스에 사용되는 호스트에 TLS 인증서를 제공해야 합니다. TLS 인증서 값은 인증서의 퍼블릭 키여야 합니다.
- GitHub 앱을 생성하려면 GitHub Enterprise Server 인스턴스의 관리자여야 합니다.

연결/호스트가 작동하다가 현재 작동이 중지됨

연결/호스트가 이전에 작동했지만 현재 작동하지 않는 경우 VPC 구성이 변경되었거나 GitHub 앱이 수정되었기 때문일 수 있습니다. 다음을 확인하세요.

- 연결을 위해 생성한 호스트 리소스에 연결된 보안 그룹이 변경되었거나 더 이상 GitHub Enterprise Server에 액세스할 수 없습니다. AWS CodeStar Connections에는 GitHub Enterprise Server 인스턴스에 연결할 수 있는 보안 그룹이 필요합니다.

- DNS 서버 IP가 최근에 변경되었습니다. 연결을 위해 생성한 호스트 리소스에 지정된 VPC에 연결되어 있는 DHCP 옵션을 통해 이를 확인할 수 있습니다. 최근에 AmazonProvidedDNS에서 사용자 지정 DNS 서버로 이전했거나, 새 사용자 지정 DNS 서버를 사용하기 시작한 경우 호스트/연결의 작동이 중지됩니다. 이 문제를 해결하려면 기존 호스트를 삭제하고 다시 생성하여 데이터베이스에 최신 DNS 설정을 저장합니다.
- 네트워크 ACL 설정이 변경되어 GitHub Enterprise Server 인프라가 위치한 서브넷에 대한 HTTP 연결을 더 이상 허용하지 않습니다.
- GitHub Enterprise Server에서 AWS CodeStar Connections 앱의 모든 구성이 변경되었습니다. URL 또는 앱 보안 암호와 같은 구성을 수정하면 설치된 GitHub Enterprise Server 인스턴스와 AWS CodeStar Connections 간의 연결이 끊어질 수 있습니다.

네트워크 인터페이스를 삭제할 수 없음

네트워크 인터페이스를 감지할 수 없는 경우 다음을 확인하세요.

- AWS CodeStar Connections에 의해 생성된 네트워크 인터페이스는 호스트를 삭제해야만 삭제할 수 있습니다. 사용자가 수동으로 삭제할 수는 없습니다.
- 이 경우 다음 권한이 있어야 합니다.

```
ec2:DescribeNetworkInterfaces
ec2>DeleteNetworkInterface
```

GitHub Enterprise Server 연결을 위한 웹훅 VPC 엔드포인트(PrivateLink) 문제 해결

VPC 구성을 사용하여 호스트를 생성하면 웹훅 VPC 엔드포인트가 생성됩니다.

Note

VPC에 웹훅 엔드포인트(PrivateLink)를 사용하도록 구성된 연결과 관련한 문제 해결은 이 섹션을 참조하세요. Amazon VPC 내의 GitHub Enterprise Server 호스트 구성과 관련한 문제 해결은 [호스트의 VPC 구성 문제 해결](#) 섹션을 참조하세요.

설치된 공급자 유형에 대한 연결을 생성하고 서버가 VPC 내에 구성되도록 지정한 경우, AWS CodeStar Connections가 호스트를 생성하며 웹훅용 VPC 엔드포인트(PrivateLink)가 자동으로 생성됩니다. 따라서 호스트가 웹훅을 통해 Amazon 네트워크에서 통합 AWS 서비스로 이벤트 데이터

를 전송할 수 있습니다. 자세한 내용은 [AWS CodeStar Connections 및 인터페이스 VPC 엔드포인트 \(AWS PrivateLink\)](#) 섹션을 참조하세요.

주제

- [웹훅 VPC 엔드포인트를 삭제할 수 없음](#)

웹훅 VPC 엔드포인트를 삭제할 수 없음

AWS CodeStar Connections는 호스트를 위한 웹훅 VPC 엔드포인트의 수명 주기를 관리합니다. 엔드포인트를 삭제하려면 해당 호스트 리소스를 삭제해야 합니다.

- AWS CodeStar Connections에 의해 생성된 웹훅 VPC 엔드포인트(PrivateLink)는 호스트를 [삭제](#)해야만 삭제할 수 있습니다. 수동으로는 삭제할 수 없습니다.
- 이 경우 다음 권한이 있어야 합니다.

```
ec2:DescribeNetworkInterfaces
ec2>DeleteNetworkInterface
```

2020년 11월 24일 이전에 생성된 호스트의 문제 해결

2020년 11월 24일부터, AWS CodeStar Connections가 호스트를 설정할 때 추가 VPC 엔드포인트 (PrivateLink) 지원이 자동으로 설정됩니다. 이 업데이트가 적용되기 전에 생성한 호스트의 경우 이 문제 해결 섹션을 참조하세요.

자세한 내용은 [AWS CodeStar Connections 및 인터페이스 VPC 엔드포인트\(AWS PrivateLink\)](#) 섹션을 참조하세요.

주제

- [2020년 11월 24일 이전에 생성된 호스트가 있고 웹훅에 VPC 엔드포인트\(PrivateLink\)를 사용하려는 경우](#)
- [사용 가능한 상태의 호스트를 가져올 수 없음\(VPC 오류\)](#)

2020년 11월 24일 이전에 생성된 호스트가 있고 웹훅에 VPC 엔드포인트(PrivateLink)를 사용하려는 경우

GitHub Enterprise Server에 대한 호스트를 설정하면 웹훅 엔드포인트가 자동으로 생성됩니다. 이제 연결에는 VPC PrivateLink 웹훅 엔드포인트가 사용됩니다. 2020년 11월 24일 이전에 호스트를 생성

한 경우 VPC PrivateLink 웹훅 엔드포인트를 사용하려면 먼저 호스트를 [삭제](#)한 다음 새 호스트를 [생성](#)해야 합니다.

사용 가능한 상태의 호스트를 가져올 수 없음(VPC 오류)

호스트가 2020년 11월 24일 이전에 생성된 호스트에 대한 AWS CodeStar Connections 앱 설정을 완료할 수 없는 경우, VPC 구성 또는 GitHub Enterprise Server 인스턴스에 문제가 있기 때문일 수 있습니다.

GitHub Enterprise Server 인스턴스가 GitHub 웹훅에 대한 송신 네트워크 트래픽을 전송하려면 VPC에 NAT 게이트웨이(또는 아웃바운드 인터넷 액세스)가 필요합니다.

GitHub 리포지토리에 대한 연결을 생성할 수 없음

문제:

GitHub 리포지토리에 대한 연결에는 AWS Connector for GitHub가 사용되므로, 연결을 생성하려면 리포지토리에 대한 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-organizations-and-teams/permission-levels-for-an-organization> 섹션을 참조하세요.

GitHub Enterprise Server 연결 앱 권한 편집

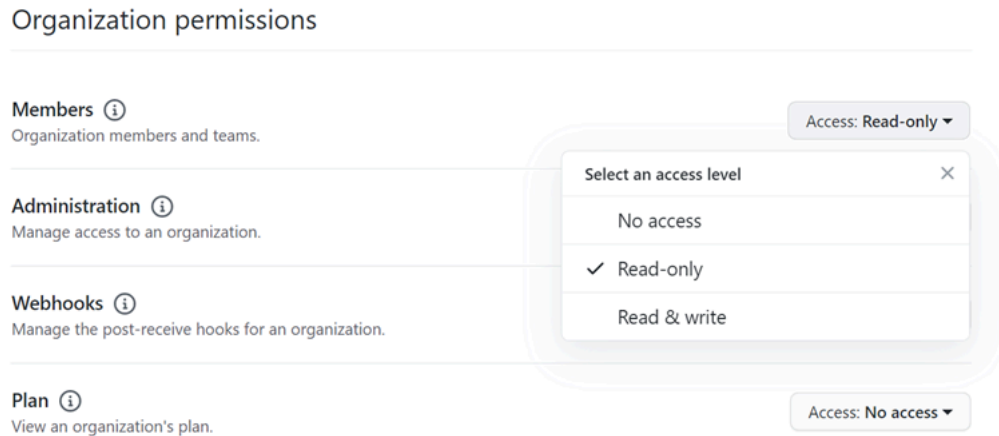
2020년 12월 23일 또는 그 이전에 GitHub Enterprise Server용 앱을 설치한 경우, 조직 구성원에게 읽기 전용 액세스 권한을 부여해야 할 수 있습니다. GitHub 앱 소유자는 다음 단계에 따라, 호스트를 생성할 때 설치된 앱의 권한을 편집할 수 있습니다.

Note

GitHub Enterprise Server 인스턴스에서 다음 단계를 완료해야 하며, GitHub 앱 소유자여야 합니다.

1. GitHub Enterprise Server의 프로필 사진에 있는 드롭다운 옵션에서 [설정(Settings)]을 선택합니다.
2. [개발자 설정(Developer settings)]을 선택한 다음 [GitHub 앱(GitHub Apps)]을 선택합니다.
3. 앱 목록에서 연결에 사용할 앱의 이름을 선택한 다음 설정 화면에서 [권한 및 이벤트(Permissions and events)]를 선택합니다.

- 조직 권한(Organization permissions) 아래의 멤버(Members)에 있는 액세스(Access) 드롭다운에서 읽기 전용(Read-only)을 선택합니다.



- [사용자에게 메모 추가(Add a note to users)]에 업데이트 이유에 대한 설명을 추가합니다. Save changes(변경 사항 저장)를 선택합니다.

GitHub에 연결할 때 연결 오류 발생: "A problem occurred, make sure cookies are enabled in your browser" 또는 "An organization owner must install the GitHub app"

문제:

GitHub 리포지토리에 대한 연결을 생성하려면 GitHub 조직 소유자여야 합니다. 조직 소속이 아닌 리포지토리의 경우 리포지토리 소유자여야 합니다. 조직 소유자가 아닌 다른 사람이 연결을 생성하면 조직 소유자에 대한 요청이 생성되고 다음 오류 중 하나가 표시됩니다.

A problem occurred, make sure cookies are enabled in your browser(문제가 발생했습니다. 브라우저에서 쿠키가 활성화되어 있는지 확인하십시오.)

또는

An organization owner must install the GitHub app(조직 소유자가 GitHub 앱을 설치해야 합니다.)

가능한 수정: GitHub 조직의 리포지토리의 경우 조직 소유자가 GitHub 리포지토리에 대한 연결을 생성해야 합니다. 조직 소속이 아닌 리포지토리의 경우 리포지토리 소유자여야 합니다.

연결의 한도를 늘리려는 경우

AWS CodeStar Connection에서 특정 한도에 대한 한도 증대를 요청할 수 있습니다. 자세한 내용은 [연결에 대한 할당량](#) 섹션을 참조하세요.

연결에 대한 할당량

다음 표에는 개발자 도구 콘솔의 연결에 대한 할당량(한도라고도 함)이 나열되어 있습니다.

이 표의 할당량은 AWS 리전별로 적용되며, 이 할당량을 늘릴 수 있습니다. 할당량 증대를 요청하려면 [지원 센터 콘솔\(Support center console\)](#)을 이용하세요. 변경할 수 있는 할당량 및 AWS 리전 정보는 [AWS 서비스 할당량](#)을 참조하세요.

Note

유럽(밀라노) AWS 리전의 경우 이 리전을 활성화해야 사용할 수 있습니다. 자세한 내용은 [리전 활성화](#)를 참조하세요.

Resource	기본 한도
AWS 계정당 최대 연결 수	250

이 표의 할당량은 고정되어 있으므로 변경할 수 없습니다.

Resource	기본 한도
연결 이름의 최대 문자 수	32자
AWS 계정당 최대 호스트 수	50
최대 리포지토리 링크 수	100
최대 AWS CloudFormation 스택 동기화 구성 수	100
리포지토리 링크당 최대 동기화 구성 수	100
브랜치당 최대 동기화 구성 수	50

허용 목록에 추가할 IP 주소

IP 필터링을 구현하거나 Amazon EC2 인스턴스에서 특정 IP 주소를 허용하는 경우 다음 IP 주소를 허용 목록에 추가하세요. 이렇게 하면 GitHub Bitbucket과 같은 공급자에 연결할 수 있습니다.

다음 표에는 AWS 리전별로 개발자 도구 콘솔의 연결을 위한 IP 주소가 나열되어 있습니다.

Note

유럽(밀라노) 리전의 경우 이 리전을 활성화해야 사용할 수 있습니다. 자세한 내용은 [리전 활성화](#)를 참조하세요.

지역	IP 주소
미국 서부(오레곤)(us-west-2)	35.160.210.199, 54.71.206.108, 54.71.36.205
미국 동부(버지니아 북부)(us-east-1)	3.216.216.90, 3.216.243.220, 3.217.241.85
유럽(아일랜드)(eu-west-1)	34.242.64.82, 52.18.37.201, 54.77.75.62
미국 동부(오하이오)(us-east-2)	18.217.188.190, 18.218.158.91, 18.220.4.80
아시아 태평양(싱가포르)(ap-southeast-1)	18.138.171.151, 18.139.22.70, 3.1.157.176
아시아 태평양(시드니)(ap-southeast-2)	13.236.59.253, 52.64.166.86, 54.206.1.112
아시아 태평양(도쿄)(ap-northeast-1)	52.196.132.231, 54.95.133.227, 18.181.13.91
유럽(프랑크푸르트)(eu-central-1)	18.196.145.164, 3.121.252.59, 52.59.104.195
아시아 태평양(서울)(ap-northeast-2)	13.125.8.239, 13.209.223.177, 3.37.200.23
아시아 태평양(뭄바이)(ap-south-1)	13.234.199.152, 13.235.29.220, 35.154.23 0.124
남아메리카(상파울루)(sa-east-1)	18.229.77.26, 54.233.226.52, 54.233.207.69
캐나다(중부)(ca-central-1)	15.222.219.210, 35.182.166.138, 99.79.111 .198
유럽(런던)(eu-west-2)	3.9.97.205, 35.177.150.185, 35.177.200.225
미국 서부(캘리포니아 북부)(us-west-1)	52.52.16.175, 52.8.63.87
유럽(파리)(eu-west-3)	35.181.127.138, 35.181.145.22, 35.181.20.200

지역	IP 주소
유럽(스톡홀름)(eu-north-1)	13.48.66.148, 13.48.8.79, 13.53.78.182
유럽(밀라노) (eu-south-1)	18.102.28.105, 18.102.35.130, 18.102.8.116

개발자 도구 콘솔의 기능에 대한 보안

클라우드 AWS 보안이 최우선 과제입니다. AWS 고객은 가장 보안에 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처의 혜택을 누릴 수 있습니다.

보안은 기업과 기업 간의 AWS 공동 책임입니다. [공동 책임 모델](#)은 이 사항을 클라우드 내 보안 및 클라우드의 보안으로 설명합니다.

- 클라우드 보안 - AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호하는 역할을 합니다. AWS 또한 안전하게 사용할 수 있는 서비스를 제공합니다. 서드 파티 감사원은 정기적으로 [AWS 규정 준수 프로그램](#)의 일환으로 보안 효과를 테스트하고 검증합니다. AWS CodeStar 알림 및 AWS CodeStar 연결에 적용되는 규정 준수 프로그램에 대해 알아보려면 [규정 준수 프로그램별 범위 내 AWS 서비스를](#) 참조하십시오.
- 클라우드에서의 보안 - 귀하의 책임은 사용하는 AWS 서비스에 따라 결정됩니다. 또한 귀하는 귀사의 데이터의 민감도, 귀사의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 AWS CodeStar 알림 및 AWS CodeStar 연결을 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 항목에서는 보안 및 규정 준수 목표를 충족하도록 AWS CodeStar 알림 및 AWS CodeStar 연결을 구성하는 방법을 보여줍니다. 또한 AWS CodeStar 알림 및 AWS CodeStar 연결 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법도 알아봅니다.

개발자 도구 콘솔의 서비스 보안에 대한 자세한 내용은 다음을 참조하세요.

- [CodeBuild 보안](#)
- [CodeCommit 보안](#)
- [CodeDeploy 보안](#)
- [CodePipeline 보안](#)

알림 콘텐츠 및 보안 이해

알림은 구성된 알림 규칙 대상을 구독한 사용자에게 리소스 관련 정보를 제공합니다. 이 정보에는 리포지토리 콘텐츠, 빌드 상태, 배포 상태 및 파이프라인 실행을 포함한 개발자 도구 리소스 관련 세부 정보가 포함될 수 있습니다.

예를 들어, 커밋 또는 풀 리퀘스트에 대한 설명을 CodeCommit 포함하도록 리포지토리의 알림 규칙을 구성할 수 있습니다. 이러한 경우 해당 규칙에 대한 응답으로 발송된 알림에는 이러한 설명에서 참조된 라인 또는 코드 라인이 포함될 수 있습니다. 마찬가지로 빌드 상태 및 단계의 성공 또는 실패를 CodeBuild 포함하도록 빌드 프로젝트에 대한 알림 규칙을 구성할 수 있습니다. 이 규칙에 대한 응답으로 발송된 알림에는 이러한 정보가 포함될 수 있습니다.

수동 승인에 대한 정보를 CodePipeline 포함하도록 파이프라인의 알림 규칙을 구성할 수 있으며, 이 규칙에 대한 응답으로 전송된 알림에는 해당 승인을 제공하는 사람의 이름이 포함될 수 있습니다. 배포 성공을 CodeDeploy 나타내도록 애플리케이션에 대한 알림 규칙을 구성할 수 있으며, 이 규칙에 대한 응답으로 전송된 알림에는 배포 대상에 대한 정보가 포함될 수 있습니다.

알림에는 빌드 상태, 설명이 있는 코드 줄, 배포 상태 및 파이프라인 승인과 같은 프로젝트별 정보가 포함될 수 있습니다. 프로젝트의 보안을 보장하는 데 도움이 되려면 알림 규칙의 대상과 대상으로 지정된 Amazon SNS 주제의 구독자 목록 모두를 정기적으로 검토해야 합니다. 추가로 이벤트에 대한 응답으로 발송된 알림의 콘텐츠는 기본 서비스에 추가 기능이 추가되면 변경될 수 있습니다. 이러한 변경은 이미 존재하는 규칙 알림에 알리지 않고 발생할 수 있습니다. 알림 메시지의 콘텐츠를 주기적으로 검토하면 어떤 내용이 발송되고 누구에게 발송되는지 이해하는 데 도움이 됩니다.

알림 규칙에 대해 제공되는 이벤트 유형에 대한 자세한 내용은 [알림 관련 개념](#) 단원을 참조하십시오.

알림에 포함된 세부 정보를 이벤트에 포함된 세부 정보로만 제한하도록 선택할 수 있습니다. 이를 기본 세부 정보 유형이라고 합니다. 이러한 이벤트에는 Amazon EventBridge 및 Amazon CloudWatch Events로 전송되는 것과 정확히 동일한 정보가 포함됩니다.

와 같은 CodeCommit 개발자 도구 콘솔 서비스는 알림 메시지에 이벤트에서 제공되는 정보 외에 일부 또는 모든 이벤트 유형에 대한 정보를 추가하도록 선택할 수 있습니다. 이 보충 정보는 언제든지 추가하여 현재 이벤트 유형을 개선하거나 향후 이벤트 유형을 보완할 수 있습니다. 전체 세부 정보 유형을 선택하여 이벤트에 대한 보충 정보를 알림에 포함하도록 선택할 수 있습니다(가능한 경우). 자세한 내용은 [세부 정보 유형](#)(들) 참조하세요.

AWS CodeStar Notifications 및 AWS CodeStar Connections의 데이터 보호

AWS [공동 책임 모델](#)은 AWS CodeStar Notifications 및 AWS CodeStar Connections의 데이터 보호에 적용됩니다. 이 모델에서 설명하는 것처럼 AWS는 모든 AWS 클라우드를 실행하는 글로벌 인프라를 보호할 책임이 있습니다. 이 인프라에서 호스팅되는 콘텐츠에 대한 제어를 유지하는 것은 사용자의 책임입니다. 사용하는 AWS 서비스의 보안 구성과 관리 작업에 대한 책임도 사용자에게 있습니다. 데이

터 프라이버시에 대한 자세한 내용은 [데이터 프라이버시 FAQ](#)를 참조하십시오. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 [AWS 공동 책임 모델 및 GDPR](#) 블로그 게시물을 참조하십시오.

데이터를 보호하려면 AWS 계정 보안 인증 정보를 보호하고 AWS IAM Identity Center 또는 AWS Identity and Access Management(IAM)을 통해 개별 사용자 계정을 설정하는 것이 좋습니다. 이러한 방식에는 각 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 복수 인증(MFA)을 사용합니다.
- SSL/TLS를 사용하여 AWS 리소스와 통신합니다. TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- AWS CloudTrail(으)로 API 및 사용자 활동 로깅을 설정합니다.
- AWS 암호화 솔루션을 AWS 서비스 내의 모든 기본 보안 제어와 함께 사용합니다.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용합니다.
- 명령행 인터페이스 또는 API를 통해 AWS에 액세스할 때 FIPS 140-2 검증된 암호화 모듈이 필요한 경우, FIPS 엔드포인트를 사용합니다. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 [FIPS\(Federal Information Processing Standard\) 140-2](#)를 참조하십시오.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 양식 텍스트 필드에 입력하지 않는 것이 좋습니다. 여기에는 AWS CodeStar Notifications와 AWS CodeStar Connections 또는 기타 AWS 서비스에서 콘솔, API, AWS CLI 또는 AWS SDK를 사용하여 작업하는 경우가 포함됩니다. 이름에 사용되는 태그 또는 자유 형식 텍스트 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 보안 인증 정보를 URL에 포함해서는 안 됩니다.

AWS CodeStar 알림 및 AWS CodeStar 연결에 대한 ID 및 액세스 관리

AWS Identity and Access Management (IAM)은 관리자가 리소스에 대한 액세스를 안전하게 제어할 수 있도록 AWS 서비스 있도록 도와줍니다. IAM 관리자는 AWS CodeStar 알림 및 연결 리소스를 사용할 수 있는 인증(로그인) 및 권한 부여(권한 부여)를 받을 수 있는 사용자를 제어합니다. AWS CodeStar IAM은 추가 AWS 서비스 비용 없이 사용할 수 있습니다.

주제

- [고객](#)
- [자격 증명을 통한 인증](#)
- [정책을 사용한 액세스 관리](#)
- [개발자 도구 콘솔의 기능이 IAM에서 작동하는 방식](#)
- [AWS CodeConnections 권한 참조](#)
- [자격 증명 기반 정책 예시](#)
- [태그를 사용하여 AWS CodeStar Connections 리소스에 대한 액세스를 제어합니다.](#)
- [콘솔에서 알림 및 연결 사용](#)
- [사용자가 자신이 권한을 볼 수 있도록 허용](#)
- [문제 해결: AWS CodeStar 알림 및 AWS CodeStar 연결, ID 및 액세스](#)
- [AWS CodeStar Notifications에 서비스 연결 역할 사용](#)
- [AWS CodeConnections에 서비스 연결 역할 사용](#)
- [AWS CodeConnections의 AWS 관리형 정책](#)

고객

AWS CodeStar 알림 및 연결에서 수행하는 작업에 따라 AWS Identity and Access Management (IAM) 사용 방식이 다릅니다. AWS CodeStar

서비스 사용자 - AWS CodeStar 알림 및 AWS CodeStar 연결 서비스를 사용하여 작업을 수행하는 경우 관리자가 필요한 자격 증명과 권한을 제공합니다. 더 많은 AWS CodeStar 알림 및 AWS CodeStar 연결 기능을 사용하여 작업을 수행함에 따라 추가 권한이 필요할 수 있습니다. 액세스 권한 관리 방식을 이해하면 적절한 권한을 관리자에게 요청할 수 있습니다. AWS CodeStar 알림 및 AWS CodeStar 연결의 기능에 액세스할 수 없는 경우 을 참조하십시오 [문제 해결: AWS CodeStar 알림 및 AWS CodeStar 연결, ID 및 액세스](#).

서비스 관리자 — 회사에서 AWS CodeStar 알림 및 AWS CodeStar 연결 리소스를 담당하는 경우 AWS CodeStar 알림 및 AWS CodeStar 연결에 대한 전체 액세스 권한이 있을 것입니다. 서비스 사용자가 액세스해야 하는 AWS CodeStar 알림 및 AWS CodeStar 연결 기능과 리소스를 결정하는 것은 여러분의 몫입니다. 그런 다음, IAM 관리자에게 요청을 제출하여 서비스 사용자의 권한을 변경해야 합니다. 이 페이지의 정보를 검토하여 IAM의 기본 개념을 이해합니다. 회사에서 IAM을 AWS CodeStar 알림 및 AWS CodeStar 연결과 함께 사용하는 방법에 대한 자세한 내용은 을 참조하십시오 [개발자 도구 콘솔의 기능이 IAM에서 작동하는 방식](#).

IAM 관리자 — IAM 관리자라면 AWS CodeStar 알림 및 연결에 대한 액세스를 관리하는 정책을 작성하는 방법에 대해 자세히 알고 싶을 것입니다. AWS CodeStar IAM에서 사용할 수 있는 AWS CodeStar 알림 및 AWS CodeStar 연결 ID 기반 정책의 예를 보려면 [여기](#)를 참조하십시오. [자격 증명 기반 정책 예시](#)

자격 증명을 통한 인증

인증은 자격 증명 자격 증명을 AWS 사용하여 로그인하는 방법입니다. IAM 사용자로 인증 (로그인 AWS) 하거나 IAM 역할을 맡아 인증 (로그인) 해야 합니다. AWS 계정 루트 사용자

ID 소스를 통해 제공된 자격 증명을 사용하여 페더레이션 ID로 로그인할 수 있습니다. AWS IAM Identity Center (IAM ID 센터) 사용자, 회사의 싱글 사인온 인증, Google 또는 Facebook 자격 증명이 페더레이션 ID의 예입니다. 페더레이션 자격 증명으로 로그인할 때 관리자가 이전에 IAM 역할을 사용하여 ID 페더레이션을 설정했습니다. 페더레이션을 사용하여 액세스하는 경우 AWS 간접적으로 역할을 맡게 됩니다.

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 계정

AWS 프로그래밍 방식으로 액세스하는 경우 자격 증명을 사용하여 요청에 암호화 방식으로 서명할 수 있는 소프트웨어 개발 키트 (SDK)와 명령줄 인터페이스 (CLI)를 AWS 제공합니다. AWS 도구를 사용하지 않는 경우 요청에 직접 서명해야 합니다. 권장 방법을 사용하여 직접 요청에 서명하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 AWS [API 요청 서명을](#) 참조하십시오.

사용하는 인증 방법에 상관없이 추가 보안 정보를 제공해야 할 수도 있습니다. 예를 들어, AWS 계정의 보안을 강화하기 위해 다단계 인증 (MFA)을 사용할 것을 권장합니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서의 [다중 인증](#) 및 IAM 사용 설명서의 [AWS에서 다중 인증\(MFA\) 사용](#)을 참조하세요.

AWS 계정 루트 사용자

계정을 만들 때는 먼저 AWS 계정계정의 모든 AWS 서비스 리소스와 모든 리소스에 완전히 액세스할 수 있는 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 전체 작업 목록은 IAM 사용 설명서의 [루트 사용자 보안 인증이 필요한 태스크](#)를 참조하세요.

IAM 사용자 및 그룹

[IAM 사용자는 단일 사용자](#) 또는 애플리케이션에 대한 특정 권한을 가진 사용자 내의 자격 증명입니다. AWS 계정 가능하면 암호 및 액세스 키와 같은 장기 자격 증명이 있는 IAM 사용자를 생성하는 대신 임

시 자격 증명을 사용하는 것이 좋습니다. 하지만 IAM 사용자의 장기 자격 증명에 필요한 특정 사용 사례가 있는 경우 액세스 키를 교체하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 [장기 보안 인증이 필요한 사용 사례의 경우 정기적으로 액세스 키 교체](#)를 참조하세요.

[IAM 그룹](#)은 IAM 사용자 컬렉션을 지정하는 자격 증명입니다. 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 IAMAdmins라는 그룹이 있고 이 그룹에 IAM 리소스를 관리할 권한을 부여할 수 있습니다.

사용자는 역할과 다릅니다. 사용자는 한 사람 또는 애플리케이션과 고유하게 연결되지만, 역할은 해당 역할이 필요한 사람이라면 누구나 수임할 수 있습니다. 사용자는 영구적인 보안 인증 정보를 가지고 있지만, 역할은 임시 보안 인증 정보만 제공합니다. 자세한 내용은 IAM 사용 설명서의 [IAM 사용자를 만들어야 하는 경우\(역할이 아님\)](#)를 참조하세요.

IAM 역할

[IAM 역할](#)은 특정 권한을 가진 사용자 AWS 계정 내의 자격 증명입니다. IAM 사용자와 유사하지만, 특정 개인과 연결되지 않습니다. 역할을 AWS Management Console [전환하여](#) 에서 일시적으로 IAM 역할을 맡을 수 있습니다. AWS CLI 또는 AWS API 작업을 호출하거나 사용자 지정 URL을 사용하여 역할을 수임할 수 있습니다. 역할 사용 방법에 대한 자세한 내용은 IAM 사용 설명서의 [IAM 역할 사용](#)을 참조하세요.

임시 보안 인증 정보가 있는 IAM 역할은 다음과 같은 상황에서 유용합니다.

- 페더레이션 사용자 액세스 - 페더레이션 자격 증명에 권한을 부여하려면 역할을 생성하고 해당 역할의 권한을 정의합니다. 페더레이션 자격 증명에 인증되면 역할이 연결되고 역할에 정의된 권한이 부여됩니다. 페더레이션 역할에 대한 자세한 내용은 IAM 사용 설명서의 [타사 자격 증명 공급자의 역할 만들기](#)를 참조하세요. IAM Identity Center를 사용하는 경우 권한 집합을 구성합니다. 인증 후 자격 증명이 액세스할 수 있는 항목을 제어하기 위해 IAM Identity Center는 권한 세트를 IAM의 역할과 연관짓습니다. 권한 세트에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 [권한 세트](#)를 참조하세요.
- 임시 IAM 사용자 권한 - IAM 사용자 또는 역할은 IAM 역할을 수임하여 특정 태스크에 대한 다양한 권한을 임시로 받을 수 있습니다.
- 크로스 계정 액세스 - IAM 역할을 사용하여 다른 계정의 사용자(신뢰할 수 있는 보안 주체)가 내 계정의 리소스에 액세스하도록 허용할 수 있습니다. 역할은 계정 간 액세스를 부여하는 기본적인 방법입니다. 그러나 일부 AWS 서비스 경우에는 역할을 프록시로 사용하는 대신 정책을 리소스에 직접 연결할 수 있습니다. 크로스 계정 액세스를 위한 역할과 리소스 기반 정책의 차이점을 알아보려면 IAM 사용 설명서의 [IAM 역할과 리소스 기반 정책의 차이](#)를 참조하십시오.

- 서비스 간 액세스 — 일부는 다른 AWS 서비스서비스의 기능을 AWS 서비스 사용합니다. 예를 들어 서비스에서 직접 호출을 수행하면 일반적으로 해당 서비스는 EC2에서 애플리케이션을 실행하거나 S3에 개체를 저장합니다. 서비스는 직접 호출하는 보안 주체의 권한을 사용하거나, 서비스 역할을 사용하거나, 또는 서비스 연결 역할을 사용하여 이 태스크를 수행할 수 있습니다.
- 순방향 액세스 세션 (FAS) — IAM 사용자 또는 역할을 사용하여 작업을 수행하는 경우 보안 AWS 주체로 간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS는 전화를 거는 주체의 권한을 다운스트림 AWS 서비스서비스에 AWS 서비스 요청하기 위한 요청과 결합하여 사용합니다. FAS 요청은 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. FAS 요청 시 정책 세부 정보는 [전달 액세스 세션](#)을 참조하세요.
- 서비스 역할 - 서비스 역할은 서비스가 사용자를 대신하여 태스크를 수행하기 위해 맡는 [IAM 역할](#)입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [AWS 서비스에 대한 권한을 위임할 역할 생성](#)을 참조하세요.
- 서비스 연결 역할 — 서비스 연결 역할은 에 연결된 서비스 역할의 한 유형입니다. AWS 서비스서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은 사용자에게 AWS 계정 표시되며 해당 서비스가 소유합니다. IAM 관리자는 서비스 링크 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.
- Amazon EC2에서 실행되는 애플리케이션 — IAM 역할을 사용하여 EC2 인스턴스에서 실행되고 API 요청을 AWS CLI 하는 애플리케이션의 임시 자격 증명을 관리할 수 있습니다. AWS 이는 EC2 인스턴스 내에 액세스 키를 저장할 때 권장되는 방법입니다. EC2 인스턴스에 AWS 역할을 할당하고 모든 애플리케이션에서 사용할 수 있게 하려면 인스턴스에 연결된 인스턴스 프로필을 생성합니다. 인스턴스 프로파일에는 역할이 포함되어 있으며 EC2 인스턴스에서 실행되는 프로그램이 임시 보안 인증을 얻을 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여](#)를 참조하세요.

IAM 역할을 사용할지 또는 IAM 사용자를 사용할지를 알아보려면 [IAM 사용 설명서](#)의 IAM 역할(사용자 대신)을 생성하는 경우를 참조하세요.

정책을 사용한 액세스 관리

정책을 생성하고 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 또는 역할 세션) 가 요청할 때 이러한 정책을 평가합니다. 정책에서 권한은 요청이 허용되는지 또는 거부되는지를 결정합니다. 대부분의 정책은 JSON 문서로 AWS 저장됩니다. JSON 정책 문서의 구조와 콘텐츠에 대한 자세한 내용은 IAM 사용 설명서의 [JSON 정책 개요](#)를 참조하세요.

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. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다. 그런 다음 관리자가 IAM 정책을 역할에 추가하고, 사용자가 역할을 수입할 수 있습니다.

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

ID 기반 정책

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 자격 증명에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자와 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 [IAM 정책 생성](#)을 참조하세요.

ID 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 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용 설명서의 [관리형 정책과 인라인 정책의 선택](#)을 참조하세요.

개발자 도구 콘솔의 기능이 IAM에서 작동하는 방식

IAM을 사용하여 개발자 도구 콘솔의 기능에 대한 액세스를 관리하려면 먼저 어떤 IAM 기능을 사용할 수 있는지를 이해해야 합니다. 알림 및 기타 AWS 서비스가 IAM과 어떻게 작동하는지 자세히 알아보려면 IAM 사용 설명서에서 [IAM과 연동되는 AWS 서비스를](#) 참조하십시오.

주제

- [개발자 도구 콘솔의 자격 증명 기반 정책](#)
- [AWS CodeStar 알림 및 AWS CodeStar 연결 리소스 기반 정책](#)
- [태그 기반 인증](#)
- [IAM 역할](#)

개발자 도구 콘솔의 자격 증명 기반 정책

IAM ID 기반 정책을 사용하면 허용되거나 거부되는 태스크와 리소스 뿐만 아니라 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. AWS CodeStar 알림 및 AWS CodeStar 연결은 특정 작업, 리소스 및 조건 키를 지원합니다. JSON 정책에서 사용하는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 [IAM JSON 정책 요소 참조](#)를 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.

JSON 정책의 Action 요소는 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 태스크를 설명합니다. 정책 작업은 일반적으로 관련 AWS API 작업과 이름이 같습니다. 일치하는 API 작업이 없는 권한 전용 작업 같은 몇 가지 예외도 있습니다. 정책에서 여러 작업이 필요한 몇 가지 작업도 있습니다. 이러한 추가 작업을 종속 작업이라고 합니다.

연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함하십시오.

개발자 도구 콘솔의 알림에 대한 정책 작업의 경우 작업 앞에 `codestar-notifications` and `codestar-connections` 접두사를 사용합니다. 예를 들어 누군가에게 계정 내에 있는 모든 알림 규칙을 볼 수 있는 권한을 부여하려면 정책에 `codestar-notifications:ListNotificationRules` 작업을 포함시킵니다. 정책 설명에는 Action OR NotAction 요소가 포함되어야 합니다. AWS CodeStar 알림 및 AWS CodeStar 연결은 이 서비스로 수행할 수 있는 작업을 설명하는 고유한 작업 집합을 정의합니다.

명령문 하나에 AWS CodeStar 알림 작업을 여러 개 지정하려면 다음과 같이 쉼표로 구분하십시오.

```
"Action": [
  "codestar-notifications:action1",
  "codestar-notifications:action2"
```

명령문 하나에 여러 AWS CodeConnections 작업을 지정하려면 다음과 같이 쉼표로 구분합니다.

```
"Action": [
  "codestar-connections:action1",
  "codestar-connections:action2"
```

와일드카드(*)를 사용하여 여러 작업을 지정할 수 있습니다. 예를 들어, List라는 단어로 시작하는 모든 작업을 지정하려면 다음 작업을 포함합니다.

```
"Action": "codestar-notifications:List*"
```

AWS CodeStar 알림 API 작업에는 다음이 포함됩니다.

- CreateNotificationRule
- DeleteNotificationRule
- DeleteTarget
- DescribeNotificationRule
- ListEventTypes
- ListNotificationRules
- ListTagsForResource
- ListTargets
- Subscribe
- TagResource
- Unsubscribe
- UntagResource
- UpdateNotificationRule

AWS CodeConnections API 작업에는 다음이 포함됩니다.

- CreateConnection
- DeleteConnection
- GetConnection
- ListConnections
- ListTagsForResource
- TagResource
- UntagResource

인증 핸드셰이크를 AWS CodeConnections 완료하려면 다음과 같은 권한 전용 조치가 필요합니다.

- GetIndividualAccessToken
- GetInstallationUrl

- ListInstallationTargets
- StartOAuthHandshake
- UpdateConnectionInstallation

연결을 사용하려면 다음과 같은 권한 전용 작업이 필요합니다. AWS CodeConnections

- UseConnection

서비스에 연결을 AWS CodeConnections 전달하려면 다음과 같은 권한 전용 조치가 필요합니다.

- PassConnection

AWS CodeStar 알림 및 AWS CodeStar 연결 작업 목록을 보려면 IAM 사용 설명서의 [AWS CodeStar 알림으로 정의된 작업 및 AWS CodeStar 연결에 의해 정의된 작업을](#) 참조하십시오.

리소스

AWS CodeStar 알림 및 AWS CodeStar 연결은 정책에 리소스 ARN을 지정하는 것을 지원하지 않습니다.

조건 키

AWS CodeStar 알림 및 AWS CodeStar 연결은 고유한 조건 키 세트를 정의하며 일부 글로벌 조건 키 사용도 지원합니다. 모든 AWS 글로벌 조건 키를 보려면 IAM 사용 설명서의 [AWS 글로벌 조건 컨텍스트 키를](#) 참조하십시오.

모든 AWS CodeStar 알림 작업은 `codestar-notifications:NotificationsForResource` 조건 키를 지원합니다. 자세한 정보는 [자격 증명 기반 정책 예시](#)를 참조하세요.

AWS CodeConnections IAM 정책의 Condition 요소에 사용할 수 있는 다음 조건 키를 정의하십시오. 이러한 키를 사용하여 정책 설명이 적용되는 조건을 보다 상세하게 설정할 수 있습니다. 자세한 정보는 [AWS CodeConnections 권한 참조](#)를 참조하세요.

조건 키	설명
<code>codestar-connections:BranchName</code>	서드 파티 리포지토리 브랜치 이름을 기준으로 액세스를 필터링합니다.

조건 키	설명
<code>codestar-connections:FullRepositoryId</code>	요청에서 전달되는 리포지토리를 기준으로 액세스를 필터링합니다. 특정 리포지토리에 액세스하기 위한 <code>UseConnection</code> 요청에만 적용됩니다.
<code>codestar-connections:InstallationId</code>	연결을 업데이트하는 데 사용되는 서드 파티 ID(예: Bitbucket 앱 설치 ID)를 기준으로 액세스를 필터링합니다. 연결을 생성하는 데 사용할 수 있는 서드 파티 앱 설치를 제한할 수 있습니다.
<code>codestar-connections:OwnerId</code>	서드 파티 공급자의 소유자 또는 계정 ID를 기준으로 액세스를 필터링합니다.
<code>codestar-connections:PassedToService</code>	보안 주체가 연결을 전달하도록 허용된 서비스를 기준으로 액세스를 필터링합니다.
<code>codestar-connections:ProviderAction</code>	<code>UseConnection</code> 요청에서 공급자 작업을 기준으로 액세스를 필터링합니다(예: <code>ListRepositories</code>).
<code>codestar-connections:ProviderPermissionsRequired</code>	서드 파티 공급자 권한 유형을 기준으로 액세스를 필터링합니다.
<code>codestar-connections:ProviderType</code>	요청에서 전달되는 타사 공급자 유형을 기준으로 액세스를 필터링합니다.
<code>codestar-connections:ProviderTypeFilter</code>	결과를 필터링하는 데 사용된 타사 공급자 유형을 기준으로 액세스를 필터링합니다.
<code>codestar-connections:RepositoryName</code>	서드 파티 리포지토리 이름을 기준으로 액세스를 필터링합니다.

예제

AWS CodeStar 알림 및 AWS CodeStar 연결 ID 기반 정책의 예를 보려면 [여기](#)를 참조하십시오. [자격 증명 기반 정책 예시](#)

AWS CodeStar 알림 및 AWS CodeStar 연결 리소스 기반 정책

AWS CodeStar 알림 및 AWS CodeStar 연결은 리소스 기반 정책을 지원하지 않습니다.

태그 기반 인증

AWS CodeStar 알림 및 AWS CodeStar 연결 리소스에 태그를 첨부하거나 요청에 태그를 전달할 수 있습니다. 태그를 기반으로 액세스를 제어하려면 `codestar-notifications` and `codestar-connections:ResourceTag/key-name`, `aws:RequestTag/key-name` 또는 `aws:TagKeys` 조건 키를 사용하여 정책의 [조건 요소](#)에 태그 정보를 제공합니다. 태깅 전략에 대한 자세한 내용은 리소스 [AWS 태깅](#)을 참조하십시오. AWS CodeStar 알림 및 AWS CodeStar 연결 리소스에 태그를 지정하는 방법에 대한 자세한 내용은 [이](#)를 참조하십시오. [연결 리소스 태그 지정](#)

리소스의 태그를 기반으로 리소스에 대한 액세스를 제한하는 자격 증명 기반 정책의 예는 [태그를 사용하여 AWS CodeStar Connections 리소스에 대한 액세스를 제어합니다.](#)에서 확인할 수 있습니다.

IAM 역할

[IAM 역할](#)은 AWS 계정 내에서 특정 권한을 가진 엔티티입니다.

임시 자격 증명 사용

임시 자격 증명으로 연동하여 로그인하거나, IAM 역할 또는 교차 계정 역할을 수임할 수 있습니다. [AssumeRole](#) 또는 [GetFederationToken](#)와 같은 AWS STS API 작업을 호출하여 임시 보안 자격 증명을 얻습니다.

AWS CodeStar 알림 및 AWS CodeStar 연결은 임시 자격 증명 사용을 지원합니다.

서비스 연결 역할

[서비스 연결 역할](#)을 사용하면 AWS 서비스가 다른 서비스의 리소스에 액세스하여 사용자를 대신하여 작업을 완료할 수 있습니다. 서비스 연결 역할은 IAM 계정에 나타나고 서비스가 소유합니다. IAM 관리자는 서비스 연결 역할의 권한을 볼 수 있지만 편집할 수 없습니다.

AWS CodeStar 알림은 서비스 연결 역할을 지원합니다. AWS CodeStar 알림 및 AWS CodeStar 연결 서비스 연결 역할을 만들거나 관리하는 방법에 대한 자세한 내용은 [이](#)를 참조하십시오. [AWS CodeStar Notifications에 서비스 연결 역할 사용](#)

AWS CodeStar 연결은 서비스 연결 역할을 지원하지 않습니다.

AWS CodeConnections 권한 참조

다음 표에는 각 AWS CodeConnections API 작업, 권한을 부여할 수 있는 해당 작업, 권한 부여에 사용할 리소스 ARN 형식이 나와 있습니다. AWS CodeConnections API는 해당 API에서 허용하는 작업의 범위에 따라 테이블로 그룹화됩니다. IAM 자격 증명에 연결할 수 있는 쓰기 권한 정책(자격 증명 기반 정책)을 설정할 때 다음 표를 참조하세요.

권한 정책 생성 시 정책의 Action 필드에 작업을 지정합니다. 와일드카드(*) 사용 여부와 상관없이 정책의 Resource 필드에 리소스 값으로 ARN을 지정합니다.

연결 정책에서 조건을 표현하려면 여기에서 설명하고 [조건 키](#)에 나열되어 있는 조건 키를 사용합니다. AWS-wide 조건 키를 사용할 수도 있습니다. AWS-wide 키의 전체 목록은 IAM 사용 설명서의 [사용 가능한 키를](#) 참조하십시오.

작업을 지정하려면 codestar-connections: 접두사 다음에 API 작업 이름을 사용합니다(예: codestar-connections:ListConnections 또는 codestar-connections:CreateConnection).

와일드카드 사용

여러 작업이나 리소스를 지정하려면 ARN에서 와일드카드(*)를 사용합니다. 예를 들어, codestar-connections:* 는 모든 AWS CodeConnections 작업을 지정하고 해당 단어로 시작하는 모든 AWS CodeConnections 작업을 codestar-connections:Get* 지정합니다. Get 다음 예는 이름이 MyConnection으로 시작되는 모든 리소스에 대한 액세스 권한을 부여합니다.

```
arn:aws:codestar-connections:us-west-2:account-ID:connection/*
```

다음 표에 나열된 *connection* 리소스에만 와일드카드를 사용할 수 있습니다. *region* 또는 *account-id* 리소스에는 와일드카드를 사용할 수 없습니다. 와일드카드에 대한 자세한 내용은 IAM 사용 설명서에서 [IAM 식별자](#)를 참조하세요.

주제

- [연결 관리 권한](#)
- [호스트 관리를 위한 권한](#)
- [연결을 완료하기 위한 권한](#)
- [호스트 설정에 대한 권한](#)
- [서비스에 연결 전달](#)

- [연결 사용](#)
- [ProviderAction에 대해 지원되는 액세스 유형](#)
- [연결 리소스 태깅에 대해 지원되는 권한](#)
- [리포지토리 링크에 연결 전달](#)
- [리포지토리 링크에 지원되는 조건 키](#)

연결 관리 권한

또는 SDK를 사용하여 연결을 보거나 만들거나 삭제하도록 지정된 역할 또는 사용자는 다음과 같은 권한으로 제한되어야 합니다. AWS CLI

Note

다음 권한만으로는 콘솔에서 연결을 완료하거나 사용할 수 없습니다. [연결을 완료하기 위한 권한](#)의 권한을 추가해야 합니다.

```
codestar-connections:CreateConnection
codestar-connections>DeleteConnection
codestar-connections:GetConnection
codestar-connections:ListConnections
```

AWS CodeStar 알림 및 AWS CodeStar 연결: 연결 관리 작업에 필요한 권한

CreateConnection

작업: `codestar-connections:CreateConnection`

CLI 또는 콘솔을 사용하여 연결을 생성하는 데 필요합니다.

리소스: `arn:aws:codestar-connections:region:account-id:connection/connection-id`

DeleteConnection

작업: `codestar-connections>DeleteConnection`

CLI 또는 콘솔을 사용하여 연결을 삭제하는 데 필요합니다.

리소스:arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*

GetConnection

작업: codestar-connections:GetConnection

CLI 또는 콘솔을 사용하여 연결에 대한 세부 정보를 보는 데 필요합니다.

리소스:arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*

ListConnections

작업: codestar-connections>ListConnections

CLI 또는 콘솔을 사용하여 계정의 모든 연결을 나열하는 데 필요합니다.

리소스:arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*

다음 작업은 아래의 조건 키를 지원합니다.

작업	조건 키
codestar-connections:CreateConnection	codestar-connections:ProviderType
codestar-connections>DeleteConnection	N/A
codestar-connections:GetConnection	N/A
codestar-connections>ListConnections	codestar-connections:ProviderTypeFilter

호스트 관리를 위한 권한

또는 SDK를 사용하여 호스트를 보거나 만들거나 삭제하도록 지정된 역할 또는 사용자는 다음과 같이 제한된 권한을 가져야 합니다. AWS CLI

Note

다음 권한만으로는 호스트에서 연결을 완료하거나 사용할 수 없습니다. [호스트 설정에 대한 권한](#)의 권한을 추가해야 합니다.

```
codestar-connections:CreateHost
codestar-connections>DeleteHost
codestar-connections:GetHost
codestar-connections:ListHosts
```

AWS CodeStar 알림 및 AWS CodeStar 연결에는 호스트 관리 작업에 필요한 권한이 필요합니다.

CreateHost

작업: `codestar-connections:CreateHost`

CLI 또는 콘솔을 사용하여 호스트를 생성하는 데 필요합니다.

리소스: `arn:aws:codestar-connections:region:account-id:host/host-id`

DeleteHost

작업: `codestar-connections>DeleteHost`

CLI 또는 콘솔을 사용하여 호스트를 삭제하는 데 필요합니다.

리소스: `arn:aws:codestar-connections:region:account-id:host/host-id`

GetHost

작업: `codestar-connections:GetHost`

CLI 또는 콘솔을 사용하여 호스트에 대한 세부 정보를 보는 데 필요합니다.

리소스: `arn:aws:codestar-connections:region:account-id:host/host-id`

ListHosts

작업: `codestar-connections:ListHosts`

CLI 또는 콘솔을 사용하여 계정의 모든 호스트를 나열하는 데 필요합니다.

리소스: `arn:aws:codestar-connections:region:account-id:host/host-id`

다음 작업은 아래의 조건 키를 지원합니다.

작업	조건 키
<code>codestar-connections:CreateHost</code>	<code>codestar-connections:ProviderType</code>
<code>codestar-connections>DeleteHost</code>	N/A
<code>codestar-connections:GetHost</code>	N/A
<code>codestar-connections>ListHosts</code>	<code>codestar-connections:ProviderTypeFilter</code>

연결을 완료하기 위한 권한

콘솔에서 연결을 관리하도록 지정된 역할 또는 사용자에게는 콘솔에서 연결을 완료하고 설치를 만드는 데 필요한 권한이 있어야 합니다. 여기에는 공급자에게 핸드셰이크 권한을 부여하고 사용할 연결에 대한 설치를 만드는 권한이 포함됩니다. 위의 권한 외에 다음 권한을 사용하십시오.

브라우저 기반 핸드셰이크를 수행하는 경우 콘솔에서 다음 IAM 작업이 사용됩니다.

`ListInstallationTargets`, `GetInstallationUrl`, `StartOAuthHandshake`, `UpdateConnectionInstallation`, `GetIndividualAccessToken`은 IAM 정책 권한입니다. API 작업이 아닙니다.

```
codestar-connections:GetIndividualAccessToken
codestar-connections:GetInstallationUrl
codestar-connections>ListInstallationTargets
codestar-connections:StartOAuthHandshake
codestar-connections:UpdateConnectionInstallation
```

이를 기반으로 콘솔에서 연결을 사용, 생성, 업데이트 또는 삭제하려면 다음 권한이 필요합니다.

```
codestar-connections:CreateConnection
codestar-connections>DeleteConnection
codestar-connections:GetConnection
codestar-connections>ListConnections
codestar-connections:UseConnection
codestar-connections>ListInstallationTargets
```

```
codestar-connections:GetInstallationUrl
codestar-connections:StartOAuthHandshake
codestar-connections:UpdateConnectionInstallation
codestar-connections:GetIndividualAccessToken
```

AWS CodeConnections 연결 완료를 위한 작업에 필요한 권한

GetIndividualAccessToken

작업: `codestar-connections:GetIndividualAccessToken`

콘솔을 사용하여 연결을 완료하는 데 필요합니다. IAM 정책 권한일 뿐, API 작업이 아닙니다.

리소스: `arn:aws:codestar-connections:region:account-id:connection/connection-id`

GetInstallationUrl

작업: `codestar-connections:GetInstallationUrl`

콘솔을 사용하여 연결을 완료하는 데 필요합니다. IAM 정책 권한일 뿐, API 작업이 아닙니다.

리소스: `arn:aws:codestar-connections:region:account-id:connection/connection-id`

ListInstallationTargets

작업: `codestar-connections:ListInstallationTargets`

콘솔을 사용하여 연결을 완료하는 데 필요합니다. IAM 정책 권한일 뿐, API 작업이 아닙니다.

리소스: `arn:aws:codestar-connections:region:account-id:connection/connection-id`

StartOAuthHandshake

작업: `codestar-connections:StartOAuthHandshake`

콘솔을 사용하여 연결을 완료하는 데 필요합니다. IAM 정책 권한일 뿐, API 작업이 아닙니다.

리소스: `arn:aws:codestar-connections:region:account-id:connection/connection-id`

UpdateConnectionInstallation

작업: `codestar-connections:UpdateConnectionInstallation`

콘솔을 사용하여 연결을 완료하는 데 필요합니다. IAM 정책 권한일 뿐, API 작업이 아닙니다.

리소스:arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*

다음 작업은 아래의 조건 키를 지원합니다.

작업	조건 키
codestar-connections:GetIndividualAccessToken	codestar-connections:ProviderType
codestar-connections:GetInstallationUrl	codestar-connections:ProviderType
codestar-connections:ListInstallationTargets	N/A
codestar-connections:StartOAuthHandshake	codestar-connections:ProviderType
codestar-connections:UpdateConnectionInstallation	codestar-connections:InstallationId

호스트 설정에 대한 권한

콘솔에서 연결을 관리하도록 지정된 역할 또는 사용자에게는 콘솔에서 호스트를 설정하는 데 필요한 권한이 있어야 합니다. 여기에는 공급자에 대한 핸드셰이크를 승인하고 호스트 앱을 설치하는 권한이 포함됩니다. 위의 호스트에 대한 권한 외에 다음 권한을 사용합니다.

브라우저 기반 호스트 등록을 수행할 때 콘솔에서 다음 IAM 작업이 사용됩니다. RegisterAppCode 및 StartAppRegistrationHandshake는 IAM 정책 권한입니다. API 작업이 아닙니다.

```
codestar-connections:RegisterAppCode
codestar-connections:StartAppRegistrationHandshake
```

이를 기반으로 콘솔에서 호스트(예: 설치된 공급자 유형)를 사용, 생성, 업데이트 또는 삭제하려면 다음 권한이 필요합니다.

```

codestar-connections:CreateConnection
codestar-connections>DeleteConnection
codestar-connections:GetConnection
codestar-connections:ListConnections
codestar-connections:UseConnection
codestar-connections:ListInstallationTargets
codestar-connections:GetInstallationUrl
codestar-connections:StartOAuthHandshake
codestar-connections:UpdateConnectionInstallation
codestar-connections:GetIndividualAccessToken
codestar-connections:RegisterAppCode
codestar-connections:StartAppRegistrationHandshake

```

AWS CodeConnections 호스트 설정 완료를 위한 작업에 필요한 권한

RegisterAppCode

작업: `codestar-connections:RegisterAppCode`

콘솔을 사용하여 호스트 설정을 완료하는 데 필요합니다. IAM 정책 권한일 뿐, API 작업이 아닙니다.

리소스: `arn:aws:codestar-connections:region:account-id:host/host-id`

StartAppRegistrationHandshake

작업: `codestar-connections:StartAppRegistrationHandshake`

콘솔을 사용하여 호스트 설정을 완료하는 데 필요합니다. IAM 정책 권한일 뿐, API 작업이 아닙니다.

리소스: `arn:aws:codestar-connections:region:account-id:host/host-id`

다음 작업은 아래의 조건 키를 지원합니다.

서비스에 연결 전달

서비스에 연결을 전달하는 경우(예: 파이프라인을 만들거나 업데이트하기 위해 파이프라인 정의에 연결 ARN 제공) 사용자에게 `codestar-connections:PassConnection` 권한이 있어야 합니다.

AWS CodeConnections 연결 전달에 필요한 권한

PassConnection

작업: `codestar-connections:PassConnection`

서비스에 연결을 전달하는 데 필요합니다.

리소스: `arn:aws:codestar-connections:region:account-id:connection/connection-id`

또한 이 작업은 다음 조건 키를 지원합니다.

- `codestar-connections:PassedToService`

조건 키에 지원되는 값

키	유효한 작업 공급자
<code>codestar-connections:PassedToService</code>	<ul style="list-style-type: none"> • <code>codeguru-reviewer</code> • <code>codepipeline.amazonaws.com</code> • <code>proton.amazonaws.com</code>

연결 사용

와 같은 서비스가 연결을 CodePipeline 사용하는 경우 서비스 역할에 지정된 연결에 대한 `codestar-connections:UseConnection` 권한이 있어야 합니다.

콘솔에서 연결을 관리하려면 사용자 정책에 `codestar-connections:UseConnection` 권한이 있어야 합니다.

AWS CodeConnections 연결 사용에 필요한 조치

UseConnection

작업: `codestar-connections:UseConnection`

연결을 사용하는 데 필요합니다.

리소스: `arn:aws:codestar-connections:region:account-id:connection/connection-id`

또한 이 작업은 다음 조건 키를 지원합니다.

- `codestar-connections:BranchName`
- `codestar-connections:FullRepositoryId`
- `codestar-connections:OwnerId`
- `codestar-connections:ProviderAction`
- `codestar-connections:ProviderPermissionsRequired`
- `codestar-connections:RepositoryName`

조건 키에 지원되는 값

키	유효한 작업 공급자
<code>codestar-connections:FullRepositoryId</code>	리포지토리의 사용자 이름과 리포지토리 이름 (예: <code>my-owner/my-repository</code>)입니다. 특정 리포지토리에 액세스하는 데 연결을 사용하는 경우에만 지원됩니다.
<code>codestar-connections:ProviderPermissionsRequired</code>	<code>read_only</code> 또는 <code>read_write</code>
<code>codestar-connections:ProviderAction</code>	<code>GetBranch</code> , <code>ListRepositories</code> , <code>ListOwners</code> , <code>ListBranches</code> , <code>StartUploadArchiveToS3</code> , <code>GitPush</code> , <code>GitPull</code> , <code>GetUploadArchiveToS3Status</code> , <code>CreatePullRequestDiffComment</code> , <code>GetPullRequest</code> , <code>ListBranchCommits</code> , <code>ListCommitFiles</code> , <code>ListPullRequestComments</code> , <code>ListPullRequestCommits</code> . 자세한 내용은 다음 섹션을 참조하세요.

일부 기능에 필요한 조건 키는 시간이 지남에 따라 변경될 수 있습니다. 액세스 제어 요구 사항에 따라 다른 권한이 필요하지 않는 한, `codestar-connections:UseConnection`을 사용하여 연결에 대한 액세스를 제어하는 것이 좋습니다.

ProviderAction에 대해 지원되는 액세스 유형

AWS 서비스에서 연결을 사용하면 소스 코드 제공자에게 API 호출이 이루어집니다. 예를 들어 서비스에서 `https://api.bitbucket.org/2.0/repositories/username` API를 호출하여 Bitbucket 연결에 사용할 리포지토리를 나열할 수 있습니다.

ProviderAction 조건 키를 사용하면 공급자에 대한 호출을 수행할 수 있는 API를 제한할 수 있습니다. API 경로는 동적으로 생성될 수 있으며 공급자마다 경로가 다르기 때문에 ProviderAction 값은 API의 URL이 아닌 추상 작업 이름에 매핑됩니다. 이를 통해 연결의 공급자 유형에 관계없이 동일한 효과를 갖는 정책을 작성할 수 있습니다.

지원되는 각 ProviderAction 값에 대해 부여되는 액세스 유형은 다음과 같습니다. IAM 정책 권한은 다음과 같습니다. API 작업이 아닙니다.

AWS CodeConnections 지원되는 액세스 유형 **ProviderAction**

GetBranch

작업: `codestar-connections:GetBranch`

해당 브랜치의 최신 커밋 등 브랜치에 대한 정보에 액세스하는 데 필요합니다.

리소스: `arn:aws:codestar-connections:region:account-id:connection/connection-id`

ListRepositories

작업: `codestar-connections>ListRepositories`

소유자에게 속한 퍼블릭 및 프라이빗 리포지토리 목록(해당 리포지토리에 대한 세부 정보 포함)에 액세스하는 데 필요합니다.

리소스: `arn:aws:codestar-connections:region:account-id:connection/connection-id`

ListOwners

작업: `codestar-connections>ListOwners`

연결에서 액세스할 수 있는 소유자 목록에 액세스하는 데 필요합니다.

리소스: `arn:aws:codestar-connections:region:account-id:connection/connection-id`

ListBranches

작업: `codestar-connections>ListBranches`

해당 리포지토리에 있는 브랜치 목록에 액세스하는 데 필요합니다.

리소스: `arn:aws:codestar-connections:region:account-id:connection/connection-id`

StartUploadArchiveToS3

작업: `codestar-connections:StartUploadArchiveToS3`

소스 코드를 읽고 Amazon S3에 업로드하는 데 필요합니다.

리소스: `arn:aws:codestar-connections:region:account-id:connection/connection-id`

GitPush

작업: `codestar-connections:GitPush`

Git를 사용하여 리포지토리에 쓰는 데 필요합니다.

리소스: `arn:aws:codestar-connections:region:account-id:connection/connection-id`

GitPull

작업: `codestar-connections:GitPull`

Git를 사용하여 리포지토리에서 읽는 데 필요합니다.

리소스: `arn:aws:codestar-connections:region:account-id:connection/connection-id`

GetUploadArchiveToS3 스테이터스

작업: `codestar-connections:GetUploadArchiveToS3Status`

`StartUploadArchiveToS3`에서 시작한 업로드의 상태(관련 오류 메시지 포함)에 액세스하는 데 필요합니다.

리소스: `arn:aws:codestar-connections:region:account-id:connection/connection-id`

CreatePullRequestDiffComment

작업: `codestar-connections:CreatePullRequestDiffComment`

끌어오기 요청에 대한 설명에 액세스하는 데 필요합니다.

리소스: `arn:aws:codestar-connections:region:account-id:connection/connection-id`

GetPullRequest

작업: `codestar-connections:GetPullRequest`

리포지토리에 대한 끌어오기 요청을 보는 데 필요합니다.

리소스: `arn:aws:codestar-connections:region:account-id:connection/connection-id`

ListBranchCommits

작업: `codestar-connections>ListBranchCommits`

리포지토리 브랜치에 대한 커밋 목록을 보는 데 필요합니다.

리소스: `arn:aws:codestar-connections:region:account-id:connection/connection-id`

ListCommitFiles

작업: `codestar-connections>ListCommitFiles`

커밋에 대한 파일 목록을 보는 데 필요합니다.

리소스: `arn:aws:codestar-connections:region:account-id:connection/connection-id`

ListPullRequestComments

작업: `codestar-connections>ListPullRequestComments`

끌어오기 요청에 대한 설명 목록을 보는 데 필요합니다.

리소스: `arn:aws:codestar-connections:region:account-id:connection/connection-id`

ListPullRequestCommits

작업: `codestar-connections:ListPullRequestCommits`

끌어오기 요청에 대한 커밋 목록을 보는 데 필요합니다.

리소스: `arn:aws:codestar-connections:region:account-id:connection/connection-id`

연결 리소스 태깅에 대해 지원되는 권한

연결 리소스에 태깅할 때 다음 IAM 작업을 사용합니다.

```
codestar-connections:ListTagsForResource
codestar-connections:TagResource
codestar-connections:UntagResource
```

AWS CodeConnections 연결 리소스에 태그를 지정하는 데 필요한 조치

ListTagsForResource

작업: `codestar-connections:ListTagsForResource`

연결 리소스와 연결된 태그의 목록을 보는 데 필요합니다.

리소스: `arn:aws:codestar-connections:region:account-id:connection/connection-id`, `arn:aws:codestar-connections:region:account-id:host/host-id`

TagResource

작업: `codestar-connections:TagResource`

연결 리소스를 태깅하는 데 필요합니다.

리소스: `arn:aws:codestar-connections:region:account-id:connection/connection-id`, `arn:aws:codestar-connections:region:account-id:host/host-id`

UntagResource

작업: `codestar-connections:UntagResource`

연결 리소스에서 태그를 제거하는 데 필요합니다.

리소스:arn:aws:codestar-connections:*region*:*account-id*:connection/*connection-id*, arn:aws:codestar-connections:*region*:*account-id*:host/*host-id*

리포지토리 링크에 연결 전달

동기화 구성에서 리포지토리 링크가 제공되는 경우 사용자에게 리포지토리 링크 ARN/리소스에 대한 codestar-connections:PassRepository 권한이 있어야 합니다.

AWS CodeConnections 연결 전달에 필요한 권한

PassRepository

작업: codestar-connections:PassRepository

리포지토리 링크를 동기화 구성에 전달하는 데 필요합니다.

리소스:arn:aws:codestar-connections:*region*:*account-id*:repository-link/*repository-link-id*

또한 이 작업은 다음 조건 키를 지원합니다.

- codestar-connections:PassedToService

조건 키에 지원되는 값

키	유효한 작업 공급자
codestar-connections:PassedToService	<ul style="list-style-type: none"> • cloudformation.sync.codeconnections.amazonaws.com

리포지토리 링크에 지원되는 조건 키

리포지토리 링크 및 동기화 구성 리소스에 대한 작업은 다음 조건 키를 통해 지원됩니다.

- codestar-connections:Branch

요청에서 전달되는 브랜치 이름을 기준으로 액세스를 필터링합니다.

조건 키에 지원되는 작업

키	유효값
codestar-connections:Branch	이 조건 키에는 다음 작업이 지원됩니다. <ul style="list-style-type: none"> CreateSyncConfiguration UpdateSyncConfiguration GetRepositorySyncStatus

자격 증명 기반 정책 예시

기본적으로,, 또는 AWS CodePipeline 적용된 관리형 정책 중 하나를 보유한 IAM 사용자 및 역할은 해당 정책의 의도에 맞는 연결, 알림 및 알림 규칙에 대한 AWS CodeCommit 권한을 가집니다. AWS CodeBuild AWS CodeDeploy 예를 들어 전체 액세스 정책 (AWSCodeCommitFullAccess,, 또는 AWSCodePipeline_FullAccess) 중 하나가 적용된 IAM 사용자 또는 역할에도 해당 서비스의 리소스에 대해 생성된 알림 및 알림 규칙에 대한 전체 액세스 권한이 있습니다. AWSCodeBuildAdminAccessAWSCodeDeployFullAccess

다른 IAM 사용자 및 역할에는 AWS CodeStar 알림 및 AWS CodeStar 연결 리소스를 생성하거나 수정할 권한이 없습니다. 또한 AWS Management Console AWS CLI, 또는 AWS API를 사용하여 작업을 수행할 수 없습니다. IAM 관리자는 지정된 리소스에서 API 작업을 수행할 수 있는 권한을 사용자와 역할에게 부여하는 IAM 정책을 생성해야 합니다. 그런 다음 관리자는 해당 권한이 필요한 IAM 사용자 또는 그룹에 이러한 정책을 연결해야 합니다.

AWS CodeStar 알림에 대한 권한 및 예제

다음 정책 설명 및 예는 AWS CodeStar 알림을 관리하는 데 도움이 될 수 있습니다.

전체 액세스 관리형 정책의 알림과 관련된 권한

AWSCodeCommitFullAccess, AWSCodeBuildAdminAccessAWSCodeDeployFullAccess, 및 AWSCodePipeline_FullAccess 관리형 정책에는 개발자 도구 콘솔에서 알림에 대한 전체 액세스를 허용하는 다음 명령문이 포함되어 있습니다. 이러한 관리형 정책 중 하나가 적용된 사용자는 알림에 대한 Amazon SNS 주제를 생성 및 관리하고, 주제에 대해 사용자를 구독 및 구독 취소하고, 알림 규칙의 대상으로 선택할 주제를 나열할 수도 있습니다.

Note

관리형 정책에서 조건 키 `codestar-notifications:NotificationsForResource`는 서비스의 리소스 유형에 특정한 값을 갖습니다. 예를 들어, 의 전체 액세스 정책에서 값은 `arn:aws:codecommit:*.CodeCommit`

```
{
  "Sid": "CodeStarNotificationsReadWriteAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:CreateNotificationRule",
    "codestar-notifications:DescribeNotificationRule",
    "codestar-notifications:UpdateNotificationRule",
    "codestar-notifications>DeleteNotificationRule",
    "codestar-notifications:Subscribe",
    "codestar-notifications:Unsubscribe"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {"codestar-notifications:NotificationsForResource" :
"arn:aws:<vendor-code>:*"}
  }
},
{
  "Sid": "CodeStarNotificationsListAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:ListNotificationRules",
    "codestar-notifications:ListTargets",
    "codestar-notifications:ListTagsForResource",
    "codestar-notifications:ListEventTypes"
  ],
  "Resource": "*"
},
{
  "Sid": "CodeStarNotificationsSNSTopicCreateAccess",
  "Effect": "Allow",
  "Action": [
    "sns:CreateTopic",
    "sns:SetTopicAttributes"
  ],
}
```



```

    "Resource": "arn:aws:sns:*:*:codestar-notifications*"
  },
  {
    "Sid": "SNSTopicListAccess",
    "Effect": "Allow",
    "Action": [
      "sns:ListTopics"
    ],
    "Resource": "*"
  },
  {
    "Sid": "CodeStarNotificationsChatbotAccess",
    "Effect": "Allow",
    "Action": [
      "chatbot:DescribeSlackChannelConfigurations",
      "chatbot:ListMicrosoftTeamsChannelConfigurations"
    ],
    "Resource": "*"
  }
}

```

읽기 전용 관리형 정책의 알림과 관련된 권한

AWSCodeCommitReadOnlyAccess,

AWSCodeBuildReadOnlyAccessAWSCodeDeployReadOnlyAccess, 및

AWSCodePipeline_ReadOnlyAccess 관리형 정책에는 알림에 대한 읽기 전용 액세스를 허용하는 다음 명령문이 포함됩니다. 예를 들어 개발자 도구 콘솔에서 리소스에 대한 알림을 볼 수 있지만 리소스를 생성, 관리 또는 구독할 수는 없습니다.

Note

관리형 정책에서 조건 키 `codestar-notifications:NotificationsForResource`는 서비스의 리소스 유형에 특정한 값을 갖습니다. 예를 들어, 의 전체 액세스 정책에서 값은 `arn:aws:codecommit:*. CodeCommit`

```

{
  "Sid": "CodeStarNotificationsPowerUserAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:DescribeNotificationRule"
  ],
}

```

```

    "Resource": "*",
    "Condition" : {
        "StringLike" : {"codestar-notifications:NotificationsForResource" :
"arn:aws:<vendor-code>:*"}
    }
},
{
    "Sid": "CodeStarNotificationsListAccess",
    "Effect": "Allow",
    "Action": [
        "codestar-notifications:ListNotificationRules",
        "codestar-notifications:ListEventTypes",
        "codestar-notifications:ListTargets"
    ],
    "Resource": "*"
}

```

다른 관리형 정책의 알림과 관련된 권한

AWSCodeCommitPowerUser, AWSCodeBuildDeveloperAccess, 및

AWSCodeBuildDeveloperAccess 관리형 정책에는 이러한 관리형 정책 중 하나를 적용한 개발자가 알림을 생성, 편집 및 구독할 수 있도록 하는 다음 명령문이 포함됩니다. 알림 규칙을 삭제하거나 리소스에 대한 태그를 관리할 수는 없습니다.

Note

관리형 정책에서 조건 키 `codestar-notifications:NotificationsForResource`는 서비스의 리소스 유형에 특정한 값을 갖습니다. 예를 들어, 의 전체 액세스 정책에서 값은 다음과 같습니다 `arn:aws:codecommit:*`. CodeCommit

```

{
    "Sid": "CodeStarNotificationsReadWriteAccess",
    "Effect": "Allow",
    "Action": [
        "codestar-notifications:CreateNotificationRule",
        "codestar-notifications:DescribeNotificationRule",
        "codestar-notifications:UpdateNotificationRule",
        "codestar-notifications:Subscribe",
        "codestar-notifications:Unsubscribe"
    ],

```

```

    "Resource": "*",
    "Condition" : {
      "StringLike" : {"codestar-notifications:NotificationsForResource" :
"arn:aws:<vendor-code>:*"}
    }
  },
  {
    "Sid": "CodeStarNotificationsListAccess",
    "Effect": "Allow",
    "Action": [
      "codestar-notifications:ListNotificationRules",
      "codestar-notifications:ListTargets",
      "codestar-notifications:ListTagsForResource",
      "codestar-notifications:ListEventTypes"
    ],
    "Resource": "*"
  },
  {
    "Sid": "SNSTopicListAccess",
    "Effect": "Allow",
    "Action": [
      "sns:ListTopics"
    ],
    "Resource": "*"
  },
  {
    "Sid": "CodeStarNotificationsChatbotAccess",
    "Effect": "Allow",
    "Action": [
      "chatbot:DescribeSlackChannelConfigurations",
      "chatbot:ListMicrosoftTeamsChannelConfigurations"
    ],
    "Resource": "*"
  }
}

```

예: 알림 관리를 위한 관리자 수준의 정책 AWS CodeStar

이 예시에서는 AWS 계정의 IAM 사용자에게 알림에 대한 전체 액세스 권한을 부여하여 사용자가 AWS CodeStar 알림 규칙의 세부 정보를 검토하고 알림 규칙, 대상, 이벤트 유형을 나열할 수 있도록 하려고 합니다. 또한 사용자가 알림 규칙을 추가, 업데이트 및 삭제하도록 허용하려고 합니다. 이는 전체 액세스 정책으로, `AWSCodeBuildAdminAccess`, `AWSCodeCommitFullAccess`, `AWSCodeDeployFullAccess`, 및 `AWSCodePipeline_FullAccess` 관리형 정책의 일부로 포함된 알림 권한과 동일합니다. 이러한 관리

형 정책과 마찬가지로 AWS 계정 전체의 알림 및 알림 규칙에 대한 전체 관리 액세스 권한이 필요한 IAM 사용자, 그룹 또는 역할에만 이러한 종류의 정책 설명을 추가해야 합니다.

Note

이 정책에는 CreateNotificationRule 허용이 포함되어 있습니다. IAM 사용자 또는 역할에 이 정책을 적용한 모든 사용자는 해당 리소스에 대한 액세스 권한이 없더라도 AWS 계정의 알림이 지원하는 모든 리소스 유형에 대한 AWS CodeStar 알림 규칙을 생성할 수 있습니다. 예를 들어 이 정책을 사용하는 사용자는 CodeCommit 자신에게 액세스할 권한이 없어도 CodeCommit 저장소에 대한 알림 규칙을 생성할 수 있습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSCodeStarNotificationsFullAccess",
      "Effect": "Allow",
      "Action": [
        "codestar-notifications:CreateNotificationRule",
        "codestar-notifications>DeleteNotificationRule",
        "codestar-notifications:DescribeNotificationRule",
        "codestar-notifications:ListNotificationRules",
        "codestar-notifications:UpdateNotificationRule",
        "codestar-notifications:Subscribe",
        "codestar-notifications:Unsubscribe",
        "codestar-notifications>DeleteTarget",
        "codestar-notifications:ListTargets",
        "codestar-notifications:ListTagsForResource",
        "codestar-notifications:TagResource",
        "codestar-notifications:UntagResource"
      ],
      "Resource": "*"
    }
  ]
}
```

예: 알림 사용에 대한 기여자 수준의 정책 AWS CodeStar

이 예시에서는 알림 생성 및 구독과 같은 AWS CodeStar 알림 day-to-day 사용에 대한 액세스 권한을 부여하고 알림 규칙 또는 대상 삭제와 같은 더 파괴적인 작업에는 액세스하지 않으려고 합니다. 이는

AWSCodeBuildDeveloperAccess, AWSCodeDeployDeveloperAccess, 관리형 정책에서 제공하는 액세스와 AWSCodeCommitPowerUser 동일합니다.

Note

이 정책에는 CreateNotificationRule 허용이 포함되어 있습니다. IAM 사용자 또는 역할에 이 정책을 적용한 모든 사용자는 해당 리소스에 대한 액세스 권한이 없더라도 AWS 계정의 알림이 지원하는 모든 리소스 유형에 대한 AWS CodeStar 알림 규칙을 생성할 수 있습니다. 예를 들어 이 정책을 사용하는 사용자는 CodeCommit 자신에게 액세스할 권한이 없어도 CodeCommit 저장소에 대한 알림 규칙을 생성할 수 있습니다.

```
{
  "Version": "2012-10-17",
  "Sid": "AWSCodeStarNotificationsPowerUserAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:CreateNotificationRule",
    "codestar-notifications:DescribeNotificationRule",
    "codestar-notifications:ListNotificationRules",
    "codestar-notifications:UpdateNotificationRule",
    "codestar-notifications:Subscribe",
    "codestar-notifications:Unsubscribe",
    "codestar-notifications:ListTargets",
    "codestar-notifications:ListTagsForResource"
  ],
  "Resource": "*"
}
```

예: AWS CodeStar 알림 사용 read-only-level 정책

이 예에서는 계정의 IAM 사용자에게 AWS 계정의 알림 규칙, 대상, 이벤트 유형에 대한 읽기 전용 액세스 권한을 부여하려고 합니다. 이 예제에서는 이러한 항목의 보기를 허용하는 정책을 생성할 수 있는 방법을 보여줍니다. 이는 AWSCodeBuildReadOnlyAccess, AWSCodeCommitReadOnly, AWSCodePipeline_ReadOnlyAccess 관리형 정책의 일부로 포함된 권한과 동일합니다.

```
{
  "Version": "2012-10-17",
```

```

    "Id": "CodeNotification__ReadOnly",
    "Statement": [
      {
        "Sid": "Reads_API_Access",
        "Effect": "Allow",
        "Action": [
          "CodeNotification:DescribeNotificationRule",
          "CodeNotification:ListNotificationRules",
          "CodeNotification:ListTargets",
          "CodeNotification:ListEventTypes"
        ],
        "Resource": "*"
      }
    ]
  }
}

```

다음에 대한 권한 및 예 AWS CodeConnections

다음의 정책 설명과 예는 AWS CodeConnections를 관리하는 데 도움이 될 수 있습니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM 자격 증명 기반 정책을 생성하는 방법에 대한 자세한 내용은 IAM 사용 설명서에서 [JSON 탭에서 정책 생성](#)을 참조하세요.

예: AWS CodeConnections CLI로 생성하고 콘솔에서 보기 위한 정책

또는 SDK를 사용하여 연결을 확인, 생성, 태그 지정 AWS CLI 또는 삭제하도록 지정된 역할 또는 사용자는 다음과 같은 권한으로 제한되어야 합니다.

Note

다음 권한만으로는 콘솔에서 연결을 완료할 수 없습니다. 다음 섹션의 권한을 추가해야 합니다.

콘솔을 사용하여 사용 가능한 연결 목록을 보고, 태그를 보고, 연결을 사용하려면 다음 정책을 사용합니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {

```

```

    "Sid": "ConnectionsFullAccess",
    "Effect": "Allow",
    "Action": [
        "codestar-connections:CreateConnection",
        "codestar-connections>DeleteConnection",
        "codestar-connections:UseConnection",
        "codestar-connections:GetConnection",
        "codestar-connections:ListConnections",
        "codestar-connections:TagResource",
        "codestar-connections:ListTagsForResource",
        "codestar-connections:UntagResource"
    ],
    "Resource": "*"
  }
]
}

```

예: 콘솔을 AWS CodeConnections 사용하여 생성하기 위한 정책

콘솔에서 연결을 관리하도록 지정된 역할 또는 사용자에게는 콘솔에서 연결을 완료하고 설치를 만드는 데 필요한 권한이 있어야 합니다. 여기에는 공급자에게 핸드셰이크 권한을 부여하고 사용할 연결에 대한 설치를 만드는 작업이 포함됩니다. 콘솔에서 연결을 사용할 수 있도록 UseConnection도 추가해야 합니다. 콘솔에서 연결을 보거나, 사용하거나, 생성하거나, 태깅하거나, 삭제하려면 다음 정책을 사용합니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codestar-connections:CreateConnection",
        "codestar-connections>DeleteConnection",
        "codestar-connections:GetConnection",
        "codestar-connections:ListConnections",
        "codestar-connections:GetInstallationUrl",
        "codestar-connections:GetIndividualAccessToken",
        "codestar-connections:ListInstallationTargets",
        "codestar-connections:StartOAuthHandshake",
        "codestar-connections:UpdateConnectionInstallation",
        "codestar-connections:UseConnection",
        "codestar-connections:TagResource",
        "codestar-connections:ListTagsForResource",

```

```

        "codestar-connections:UntagResource"
    ],
    "Resource": [
        "*"
    ]
}
]
}

```

예: 관리를 위한 관리자 수준의 정책 AWS CodeConnections

이 예시에서는 AWS 계정의 IAM 사용자에게 모든 액세스 권한을 부여하여 사용자가 연결을 추가, 업데이트 및 삭제할 수 CodeConnections 있도록 하려고 합니다. 이는 AWSCodePipeline_FullAccess관리형 정책과 동일한 전체 액세스 정책입니다. 관리형 정책과 마찬가지로 AWS 계정 전체의 연결에 대한 전체 관리 액세스 권한이 필요한 IAM 사용자, 그룹 또는 역할에만 이러한 종류의 정책 설명을 첨부해야 합니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ConnectionsFullAccess",
      "Effect": "Allow",
      "Action": [
        "codestar-connections:CreateConnection",
        "codestar-connections>DeleteConnection",
        "codestar-connections:UseConnection",
        "codestar-connections:GetConnection",
        "codestar-connections:ListConnections",
        "codestar-connections:ListInstallationTargets",
        "codestar-connections:GetInstallationUrl",
        "codestar-connections:StartOAuthHandshake",
        "codestar-connections:UpdateConnectionInstallation",
        "codestar-connections:GetIndividualAccessToken",
        "codestar-connections:TagResource",
        "codestar-connections:ListTagsForResource",
        "codestar-connections:UntagResource"
      ],
      "Resource": "*"
    }
  ]
}

```


예: 사용을 위한 기여자 수준의 정책 AWS CodeConnections

이 예시에서는 연결 day-to-day 사용 (예: 연결 생성 및 세부 정보 보기) 에는 액세스 권한을 부여하고, 연결 삭제와 같은 더 파괴적인 작업에는 액세스하지 않으려고 합니다. CodeConnections

```
{
  "Version": "2012-10-17",
  "Sid": "AWSCodeStarConnectionsPowerUserAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-connections:CreateConnection",
    "codestar-connections:UseConnection",
    "codestar-connections:GetConnection",
    "codestar-connections:ListConnections",
    "codestar-connections:ListInstallationTargets",
    "codestar-connections:GetInstallationUrl",
    "codestar-connections:GetIndividualAccessToken",
    "codestar-connections:StartOAuthHandshake",
    "codestar-connections:UpdateConnectionInstallation",
    "codestar-connections:ListTagsForResource"
  ],
  "Resource": "*"
}
```

예: 사용에 대한 read-only-level 정책 AWS CodeConnections

이 예시에서는 계정의 IAM 사용자에게 계정 내 연결에 대한 읽기 전용 액세스 권한을 부여하려고 합니다. 이 예제에서는 이러한 항목의 보기를 허용하는 정책을 생성할 수 있는 방법을 보여줍니다.

```
{
  "Version": "2012-10-17",
  "Id": "Connections__ReadOnly",
  "Statement": [
    {
      "Sid": "Reads_API_Access",
      "Effect": "Allow",
      "Action": [
        "codestar-connections:GetConnection",
        "codestar-connections:ListConnections",
        "codestar-connections:ListInstallationTargets",
        "codestar-connections:GetInstallationUrl",

```

```

        "codestar-connections:ListTagsForResource"
      ],
      "Resource": "*"
    }
  ]
}

```

예: 지정된 저장소에서 사용하기 위한 범위 축소 정책 AWS CodeConnections

다음 예제에서 고객은 CodeBuild 서비스 역할을 통해 지정된 Bitbucket 리포지토리에 액세스하기를 원합니다. CodeBuild 서비스 역할에 대한 정책:

```

{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "codestar-connections:UseConnection"
    ],
    "Resource": "arn:aws:codestar-connections:us-west-2:connection:3dee99b9-172f-4ebe-a257-722365a39557",
    "Condition": {"ForAllValues:StringEquals": {"codestar-connections:FullRepositoryId": "myrepoowner/myreponame"}}
  }
}

```

예: 연결을 사용하는 정책 CodePipeline

다음 예제에서 관리자는 사용자가 연결을 사용하기를 원합니다 CodePipeline. 사용자에게 연결된 정책은 다음과 같습니다.

```

{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "codestar-connections:PassConnection"
    ],
    "Resource": "arn:aws:codestar-connections:us-west-2:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
    "Condition": {"ForAllValues:StringEquals": {"codestar-connections:PassedToService": "codepipeline.amazonaws.com"}}
  }
}

```

```

    }
  }
}

```

예: 다음과 같은 Bitbucket 읽기 작업에 CodeBuild 서비스 역할 사용 AWS CodeConnections

다음 예제에서 고객은 리포지토리와 상관없이 CodeBuild 서비스 역할이 Bitbucket에서 읽기 작업을 수행하기를 원합니다. CodeBuild 서비스 역할에 대한 정책:

```

{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "codestar-connections:UseConnection"
    ],
    "Resource": "arn:aws:codestar-connections:us-west-2:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
    "Condition": {"ForAllValues:StringEquals": {"codestar-connections:ProviderPermissionsRequired": "read_only"}}
  }
}

```

예: CodeBuild 서비스 역할이 다음과 같은 작업을 수행하지 못하도록 제한하십시오. AWS CodeConnections

다음 예에서 고객은 CodeBuild 서비스 역할이 다음과 같은 작업을 수행하는 것을 방지하려고 CreateRepository 합니다. CodeBuild 서비스 역할에 대한 정책:

```

{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "codestar-connections:UseConnection"
    ],
    "Resource": "arn:aws:codestar-connections:us-west-2:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
    "Condition": {"ForAllValues:StringNotEquals": {"codestar-connections:ProviderPermissionsRequired": "CreateRepository"}}
  }
}

```

태그를 사용하여 AWS CodeStar Connections 리소스에 대한 액세스를 제어합니다.

리소스에 태그가 연결되거나 태그 지정을 지원하는 서비스에 대한 요청에서 전달될 수 있습니다. CodeConnections에서는 리소스에 태그가 있을 수 있으며 일부 작업에는 태그가 포함될 수 있습니다. IAM 정책을 생성하면 태그 조건 키를 사용하여 다음을 제어할 수 있습니다.

- 파이프라인 리소스에 이미 있는 태그를 기반으로 해당 리소스에 대해 작업을 수행할 수 있는 사용자
- 작업의 요청에서 전달될 수 있는 태그
- 요청에서 특정 키를 사용할 수 있는지 여부를 통제합니다.

다음 예에서는 CodeConnections 사용자에게 정책의 태그 조건을 지정하는 방법을 설명합니다.

Example 1: 요청의 태그 기반 작업 허용

다음 정책은 사용자에게 CodeConnections에서 연결을 생성할 수 있는 권한을 부여합니다.

이와 관련하여 정책은 요청이 ProjectA 값이 포함된 Project 태그를 지정하는 경우 CreateConnection 및 TagResource 작업을 허용합니다. aws:RequestTag 조건 키는 IAM 요청에서 전달할 수 있는 태그를 제어하는 데 사용됩니다. aws:TagKeys 조건은 태그 키의 대/소문자를 구분합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codestar-connections:CreateConnection",
        "codestar-connections:TagResource"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/Project": "ProjectA"
        },
        "ForAllValues:StringEquals": {
          "aws:TagKeys": ["Project"]
        }
      }
    }
  ]
}
```

```

    }
  ]
}

```

Example 2: 리소스 태그 기반 작업 허용

다음 정책은 사용자에게 CodeConnections에서 리소스에 대해 작업을 수행하고 관련 정보를 가져올 수 있는 권한을 부여합니다.

이와 관련하여 정책은 파이프라인에 ProjectA 값이 포함된 Project 태그가 있으면 특정 작업을 허용합니다. `aws:RequestTag` 조건 키는 IAM 요청에서 전달할 수 있는 태그를 제어하는 데 사용됩니다. `aws:TagKeys` 조건은 태그 키의 대/소문자를 구분합니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codestar-connections:CreateConnection",
        "codestar-connections>DeleteConnection",
        "codestar-connections:ListConnections"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Project": "ProjectA"
        },
        "ForAllValues:StringEquals": {
          "aws:TagKeys": ["Project"]
        }
      }
    }
  ]
}

```

콘솔에서 알림 및 연결 사용

알림 환경은 개발자 도구 콘솔의 CodeBuild 설정 탐색 표시줄 자체뿐만 아니라,, 콘솔에도 내장되어 있습니다. CodeCommit CodeDeploy CodePipeline 콘솔의 알림에 액세스하려면 해당 서비스에 대한 관리형 정책 중 하나가 적용되었거나 최소 권한 집합이 있어야 합니다. 이러한 권한을 통해 계정의 AWS CodeStar 알림 및 AWS CodeStar 연결 리소스에 대한 세부 정보를 나열하고 볼 수 AWS 있어야

합니다. 최소 필수 권한보다 더 제한적인 보안 인증 기반 정책을 만들면 콘솔이 해당 정책에 연결된 개체(IAM 사용자 또는 역할)에 대해 의도대로 작동하지 않습니다. AWS CodeBuild, AWS CodeCommit, AWS CodeDeploy, AWS CodePipeline, 및 콘솔에 대한 액세스 권한을 부여하는 방법 (해당 콘솔에 대한 액세스 권한 포함)에 대한 자세한 내용은 다음 항목을 참조하십시오.

- CodeBuild: ID 기반 [정책 사용 CodeBuild](#)
- CodeCommit: [ID 기반 정책 사용 대상 CodeCommit](#)
- AWS CodeDeploy: [ID 및 액세스 관리 대상 AWS CodeDeploy](#)
- CodePipeline: [IAM 정책을 통한 액세스 제어](#)

AWS CodeStar 알림에는 AWS 관리형 정책이 없습니다. 알림 기능에 대한 액세스를 제공하려면 앞에 나열된 서비스 중 하나에 대해 관리형 정책 중 하나를 적용하거나, 사용자 또는 엔터티에 부여하고자 하는 권한 수준이 포함된 정책을 생성한 다음 이 정책을 권한이 필요한 사용자, 그룹 또는 역할에 연결해야 합니다. 자세한 내용과 예는 다음을 참조하십시오.

- [예: 알림 관리를 위한 관리자 수준의 정책 AWS CodeStar](#)
- [예: 알림 사용에 대한 기여자 수준의 정책 AWS CodeStar](#)
- [예: AWS CodeStar 알림 사용 read-only-level 정책.](#)

AWS CodeStar 연결에는 AWS 관리형 정책이 없습니다. [연결을 완료하기 위한 권한](#)에서 설명하는 권한과 같이 개별 권한과 권한의 조합을 액세스에 사용합니다.

자세한 내용은 다음을 참조하십시오.

- [예: 관리를 위한 관리자 수준의 정책 AWS CodeConnections](#)
- [예: 사용을 위한 기여자 수준의 정책 AWS CodeConnections](#)
- [예: 사용에 대한 read-only-level 정책 AWS CodeConnections](#)

AWS CLI 또는 AWS API만 호출하는 사용자에게 콘솔 권한을 허용할 필요는 없습니다. 그 대신, 수행하려는 API 작업과 일치하는 작업에만 액세스할 수 있도록 합니다.

사용자가 자신이 권한을 볼 수 있도록 허용

이 예시는 IAM 사용자가 자신의 사용자 자격 증명에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여줍니다. 이 정책에는 콘솔에서 AWS CLI 또는 AWS API를 사용하여 프로그래밍 방식으로 이 작업을 완료할 수 있는 권한이 포함됩니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupForUser",
        "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": "*"
    }
  ]
}
```

문제 해결: AWS CodeStar 알림 및 AWS CodeStar 연결, ID 및 액세스

다음 정보를 사용하여 알림 및 IAM 작업 시 발생할 수 있는 일반적인 문제를 진단하고 수정할 수 있습니다.

주제

- [관리자이며 다른 사용자의 알림 액세스를 허용하려고 함](#)

- [Amazon SNS 주제를 생성하고 이를 알림 규칙 대상으로 추가했지만 이벤트에 관한 이메일을 받지 못하고 있습니다.](#)
- [내 AWS 계정 외부의 사용자가 내 AWS CodeStar 알림 및 AWS CodeStar 연결 리소스에 액세스할 수 있도록 허용하고 싶습니다.](#)

관리자이며 다른 사용자의 알림 액세스를 허용하려고 함

다른 사람이 AWS CodeStar 알림 및 AWS CodeStar 연결에 액세스할 수 있도록 하려면 액세스가 필요한 개인 또는 애플리케이션에 대한 IAM 엔티티 (사용자 또는 역할) 를 생성해야 합니다. 다른 사용자들은 해당 엔티티에 대한 보안 인증을 사용해 AWS에 액세스합니다. 그런 다음 AWS CodeStar 알림 및 연결에서 올바른 권한을 부여하는 정책을 엔티티에 AWS CodeStar 연결해야 합니다.

바로 시작하려면 IAM 사용 설명서의 [첫 번째 IAM 위임 사용자 및 그룹 생성](#)을 참조하십시오.

AWS CodeStar 알림에 대한 자세한 내용은 [을 참조하십시오](#) [AWS CodeStar 알림에 대한 권한 및 예제](#).

Amazon SNS 주제를 생성하고 이를 알림 규칙 대상으로 추가했지만 이벤트에 관한 이메일을 받지 못하고 있습니다.

이벤트에 관한 알림을 수신하려면 알림 규칙에 대한 대상으로 유효한 Amazon SNS 주제를 구독해야 하고, 이메일 주소가 Amazon SNS 주제에 구독된 상태여야 합니다. Amazon SNS 주제 관련 문제를 해결하려면 다음을 확인합니다.

- Amazon SNS 주제가 알림 규칙과 동일한 AWS 지역에 있는지 확인하십시오.
- 이메일 별칭이 올바른 주제에 구독된 상태이며, 구독을 확인했는지 확인해야 합니다. 자세한 내용은 [엔드포인트를 Amazon SNS 주제에 구독 설정](#)을 참조하세요.
- 알림에서 해당 주제에 AWS CodeStar 알림을 푸시할 수 있도록 주제 정책이 수정되었는지 확인하십시오. 이 주제 정책에 다음과 유사한 문이 포함되어야 합니다.

```
{
  "Sid": "AWSCodeStarNotifications_publish",
  "Effect": "Allow",
  "Principal": {
    "Service": [
      "codestar-notifications.amazonaws.com"
    ]
  },
  "Action": "SNS:Publish",
  "Resource": "arn:aws:sns:us-east-1:123456789012:MyNotificationTopicName",
```



```

"Condition": {
  "StringEquals": {
    "aws:SourceAccount": "123456789012"
  }
}
}

```

자세한 정보는 [설정](#)을 참조하세요.

내 AWS 계정 외부의 사용자가 내 AWS CodeStar 알림 및 AWS CodeStar 연결 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(ACL)을 지원하는 서비스의 경우 이러한 정책을 사용하여 다른 사람에게 리소스에 대한 액세스 권한을 부여할 수 있습니다.

자세히 알아보려면 다음을 참조하세요.

- AWS CodeStar 알림 및 AWS CodeStar 연결이 이러한 기능을 지원하는지 알아보려면 [개발자 도구 콘솔의 기능이 IAM에서 작동하는 방식](#)을 참조하십시오.
- 소유한 리소스에 대한 액세스를 [제공하는 방법을 알아보려면 IAM 사용 설명서의 다른 AWS 계정 IAM 사용자에게 액세스 권한 제공](#)을 참조하십시오. AWS 계정
- [제3자에게 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 타사 AWS 계정 AWS 계정 소유에 대한 액세스 제공](#)을 참조하십시오.
- ID 페더레이션을 통해 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 [외부에서 인증된 사용자에게 액세스 권한 제공\(ID 페더레이션\)](#)을 참조하세요.
- 크로스 계정 액세스를 위한 역할과 리소스 기반 정책 사용의 차이점을 알아보려면 IAM 사용 설명서의 [IAM 역할과 리소스 기반 정책의 차이](#)를 참조하세요.

AWS CodeStar Notifications에 서비스 연결 역할 사용

AWS CodeStar Notifications는 AWS Identity and Access Management(IAM) [서비스 연결 역할](#)을 사용합니다. 서비스 연결 역할은 AWS CodeStar Notifications에 직접 연결된 고유한 유형의 IAM 역할입니다. 서비스 연결 역할은 AWS CodeStar Notifications에서 사전 정의하며 서비스에서 다른 AWS 서비스를 자동으로 호출하기 위해 필요한 모든 권한을 포함합니다. 이 역할은 알림 규칙을 처음 만들 때 생성됩니다. 역할은 생성할 필요가 없습니다.

권한을 수동으로 추가할 필요가 없으므로 서비스 연결 역할을 사용하면 AWS CodeStar Notifications를 더 쉽게 설정할 수 있습니다. AWS CodeStar Notifications가 서비스 연결 역할의 권한을 정의하므로 다르게 정의되지 않은 한, AWS CodeStar Notifications만 해당 역할을 수임할 수 있습니다. 정의된 권한에는 신뢰 정책과 권한 정책이 포함되며, 이 권한 정책은 다른 IAM 엔터티에 연결할 수 없습니다.

서비스 연결 역할을 삭제하려면 먼저 관련 리소스를 삭제해야 합니다. 이렇게 하면 리소스에 대한 액세스 권한을 부주의로 삭제할 수 없기 때문에 AWS CodeStar Notifications 리소스가 보호됩니다.

서비스 연결 역할을 지원하는 다른 서비스에 대한 자세한 내용은 [IAM으로 작업하는 AWS 서비스](#)를 참조하세요.

AWS CodeStar Notifications에 대한 서비스 연결 권한

AWS CodeStar Notifications는 AWSServiceRoleForCodeStarNotifications 서비스 연결 역할을 사용하여 도구 체인에서 발생하는 이벤트에 대한 정보를 검색하고 지정한 대상에 알림을 보냅니다.

AWSServiceRoleForCodeStarNotifications 서비스 연결 역할은 역할을 수임하기 위해 다음 서비스를 신뢰합니다.

- `codestar-notifications.amazonaws.com`

역할 권한 정책은 AWS CodeStar Notifications가 지정된 리소스에서 다음 작업을 완료하도록 허용합니다.

- 작업: CloudWatch Event rules that are named `awscodestar-notifications-*`에 대한 `PutRule`
- 작업: CloudWatch Event rules that are named `awscodestar-notifications-*`에 대한 `DescribeRule`
- 작업: CloudWatch Event rules that are named `awscodestar-notifications-*`에 대한 `PutTargets`
- 작업: create Amazon SNS topics for use with AWS CodeStar Notifications with the prefix `CodeStarNotifications-`하는 `CreateTopic`
- 작업: all comments on all pull requests in all CodeCommit repositories in the AWS account에 대한 `GetCommentsForPullRequests`
- 작업: all comments on all commits in all CodeCommit repositories in the AWS account에 대한 `GetCommentsForComparedCommit`
- 작업: all commits in all CodeCommit repositories in the AWS account에 대한 `GetDifferences`

- 작업: all comments on all commits in all CodeCommit repositories in the AWS account에 대한 `GetCommentsForComparedCommit`
- 작업: all commits in all CodeCommit repositories in the AWS account에 대한 `GetDifferences`
- 작업: all AWS Chatbot clients in the AWS account에 대한 `DescribeSlackChannelConfigurations`
- 작업: all AWS Chatbot clients in the AWS account에 대한 `UpdateSlackChannelConfiguration`
- 작업: all actions in all pipelines in the AWS account에 대한 `ListActionExecutions`
- 작업: all files in all CodeCommit repositories in the AWS account unless otherwise tagged에 대한 `GetFile`

`AWSServiceRoleForCodeStarNotifications` 서비스 연결 역할에 대한 정책 설명에서 다음 작업을 볼 수 있습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "events:PutTargets",
        "events:PutRule",
        "events:DescribeRule"
      ],
      "Resource": "arn:aws:events:*:*:rule/awscodestarnotifications-*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "sns:CreateTopic"
      ],
      "Resource": "arn:aws:sns:*:*:CodeStarNotifications-*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "codecommit:GetCommentsForPullRequest",
        "codecommit:GetCommentsForComparedCommit",
```

```

        "codecommit:GetDifferences",
        "chatbot:DescribeSlackChannelConfigurations",
        "chatbot:UpdateSlackChannelConfiguration",
        "codepipeline:ListActionExecutions"
    ],
    "Resource": "*",
    "Effect": "Allow"
},
{
    "Action": [
        "codecommit:GetFile"
    ],
    "Resource": "*",
    "Condition": {
        "StringNotEquals": {
            "aws:ResourceTag/ExcludeFileContentFromNotifications": "true"
        }
    },
    "Effect": "Allow"
}
]
}

```

IAM 엔터티(사용자, 그룹, 역할 등)가 서비스 연결 역할을 생성하고 편집하거나 삭제할 수 있도록 권한을 구성할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [서비스 연결 역할 권한](#) 단원을 참조하십시오.

AWS CodeStar Notifications에 대한 서비스 연결 역할 생성

서비스 연결 역할은 수동으로 생성할 필요가 없습니다. SDK에서 개발자 도구 콘솔 또는 CreateNotificationRule API를 사용하여 알림 규칙을 생성할 수 있습니다. API를 직접 호출할 수도 있습니다. 어떤 방법을 사용하든 서비스 연결 역할이 생성됩니다.

이 서비스 연결 역할을 삭제한 다음 다시 생성해야 하는 경우 동일한 프로세스를 사용하여 계정에서 역할을 다시 생성할 수 있습니다. SDK에서 개발자 도구 콘솔 또는 CreateNotificationRule API를 사용하여 알림 규칙을 생성할 수 있습니다. API를 직접 호출할 수도 있습니다. 어떤 방법을 사용하든 서비스 연결 역할이 생성됩니다.

AWS CodeStar Notifications에 대한 서비스 연결 역할 편집

서비스 연결 역할을 생성한 후에는 다양한 개체가 해당 역할을 참조할 수 있기 때문에 역할 이름을 변경할 수 없습니다. 그러나 IAM을 사용하여 역할 설명을 편집할 수는 있습니다. 자세한 내용은 IAM 사용 설명서의 [서비스 연결 역할 편집](#)을 참조하십시오.

AWS CodeStar Notifications에 대한 서비스 연결 역할 삭제

서비스 연결 역할이 필요한 기능 또는 서비스가 더 이상 필요 없는 경우에는 해당 역할을 삭제하는 것이 좋습니다. 이렇게 하면 적극적으로 모니터링하거나 유지 관리하지 않는 미사용 개체가 없게 됩니다. 서비스 연결 역할에 대한 리소스를 먼저 정리해야 삭제할 수 있습니다. AWS CodeStar Notifications의 경우 이는 AWS 계정의 서비스 역할을 사용하는 모든 알림 규칙 삭제를 의미합니다.

Note

리소스를 삭제하려 할 때 AWS CodeStar Notifications 서비스가 역할을 사용 중이면 삭제에 실패할 수 있습니다. 이 문제가 발생하면 몇 분 기다렸다가 작업을 다시 시도하십시오.

AWSServiceRoleForCodeStarNotifications에 사용되는 AWS CodeStar Notifications 리소스를 삭제하려면

1. <https://console.aws.amazon.com/codesuite/settings/notifications>에서 AWS 개발자 도구 콘솔을 엽니다.

Note

알림 규칙은 생성된 AWS 리전에 적용됩니다. 둘 이상의 AWS 리전에 알림 규칙이 있는 경우 리전 선택기를 사용하여 AWS 리전을 변경합니다.

2. 목록에 표시되는 모든 알림 규칙을 선택한 다음 삭제를 선택합니다.
3. 알림 규칙을 생성한 모든 AWS 리전에서 이 단계를 반복합니다.

IAM을 사용하여 서비스 연결 역할을 삭제하려면

IAM 콘솔, AWS CLI 또는 AWS Identity and Access Management API를 사용하여 AWSServiceRoleForCodeStarNotifications 서비스 연결 역할을 삭제합니다. 자세한 내용은 [IAM 사용 설명서](#)의 서비스 연결 역할 삭제 단원을 참조하십시오.

AWS CodeStar Notifications 서비스 연결 역할이 지원되는 리전

AWS CodeStar Notifications는 서비스를 사용할 수 있는 모든 AWS 리전에서 서비스 연결 역할 사용을 지원합니다. 자세한 내용은 [AWS 리전 및 엔드포인트](#)와 [AWS CodeStar Notifications](#)를 참조하세요.

AWS CodeConnections에 서비스 연결 역할 사용

AWS CodeConnections는 AWS Identity and Access Management (IAM) [서비스 연결 역할](#)을 사용합니다. 서비스 연결 역할은 AWS CodeConnections에 직접 연결된 고유한 유형의 IAM 역할입니다. 서비스 연결 역할은 AWS CodeConnections에서 사전 정의하며 서비스에서 다른 AWS 서비스를 자동으로 호출하기 위해 필요한 모든 권한을 포함합니다. 이 역할은 연결을 처음 만들 때 생성됩니다. 역할은 생성할 필요가 없습니다.

서비스 연결 역할을 사용하면 권한을 수동으로 추가할 필요가 없기 때문에 간편하게 AWS CodeConnections을 설정할 수 있습니다. AWS CodeConnections에서 서비스 연결 역할 권한을 정의하므로, 달리 정의되지 않은 한 AWS CodeConnections만 해당 역할을 맡을 수 있습니다. 정의된 권한에는 신뢰 정책과 권한 정책이 포함되며, 이 권한 정책은 다른 IAM 엔터티에 연결할 수 없습니다.

서비스 연결 역할을 삭제하려면 먼저 관련 리소스를 삭제해야 합니다. 이렇게 하면 리소스에 대한 액세스 권한을 부주의로 삭제할 수 없기 때문에 AWS CodeConnections 리소스가 보호됩니다.

서비스 연결 역할을 지원하는 다른 서비스에 대한 자세한 내용은 [IAM으로 작업하는 AWS 서비스](#)를 참조하세요.

AWS CodeConnections에 대한 서비스 연결 역할 권한

AWS CodeConnections에서는 AWSServiceRoleForGitSync 서비스 연결 역할을 통해 연결된 Git 기반 리포지토리와 Git 동기화를 사용합니다.

AWSServiceRoleForGitSync 서비스 연결 역할은 역할을 수임하기 위해 다음 서비스를 신뢰합니다.

- `repository.sync.codeconnections.amazonaws.com`

이름이 `AWSGitSyncServiceRolePolicy`인 연결 권한 정책은 AWS CodeConnections이 지정된 리소스에 대해 다음 작업을 수행하도록 허용합니다.

- 작업: 사용자에게 외부 Git 기반 리포지토리와 연결을 생성하고 이러한 리포지토리와 Git 동기화를 사용할 수 있는 권한을 부여합니다.

IAM 엔터티(사용자, 그룹, 역할 등)가 서비스 연결 역할을 생성하고 편집하거나 삭제할 수 있도록 권한을 구성할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [서비스 연결 역할 권한](#)을 참조하십시오.

AWS CodeConnections에 대한 서비스 연결 역할 생성

서비스 연결 역할은 수동으로 생성할 필요가 없습니다. CreateRepositoryLink API를 사용하여 Git과 동기화된 프로젝트를 위한 리소스를 생성할 때 역할을 생성합니다.

이 서비스 연결 역할을 삭제한 다음 다시 생성해야 하는 경우 동일한 프로세스를 사용하여 계정에서 역할을 다시 생성할 수 있습니다.

AWS CodeConnections에 대한 서비스 연결 역할 편집

서비스 연결 역할을 생성한 후에는 다양한 개체가 해당 역할을 참조할 수 있기 때문에 역할 이름을 변경할 수 없습니다. 그러나 IAM을 사용하여 역할 설명을 편집할 수는 있습니다. 자세한 내용은 IAM 사용 설명서의 [서비스 연결 역할 편집](#)을 참조하십시오.

AWS CodeConnections에 대한 서비스 연결 역할 삭제

서비스 연결 역할이 필요한 기능 또는 서비스가 더 이상 필요 없는 경우에는 해당 역할을 삭제하는 것이 좋습니다. 이렇게 하면 적극적으로 모니터링하거나 유지 관리하지 않는 미사용 개체가 없게 됩니다. 서비스 연결 역할에 대한 리소스를 먼저 정리해야 삭제할 수 있습니다. 이는 AWS 계정의 서비스 역할을 사용하는 모든 연결을 삭제함을 의미합니다.

Note

리소스를 삭제하려 할 때 AWS CodeConnections 서비스가 역할을 사용 중이면 삭제에 실패할 수 있습니다. 이 문제가 발생하면 몇 분 기다렸다가 작업을 다시 시도하십시오.

AWSServiceRoleForGitSync에서 사용하는 AWS CodeConnections 리소스를 삭제하려면

1. 개발자 도구 콘솔을 열고 설정을 선택합니다.
2. 목록에 표시되는 모든 연결을 선택한 다음 삭제를 선택합니다.
3. 연결을 생성한 모든 AWS 리전에서 이 단계를 반복합니다.

IAM을 사용하여 서비스 연결 역할을 삭제하려면

IAM 콘솔, AWS CLI 또는 AWS Identity and Access Management API를 사용하여 AWSServiceRoleForGitSync 서비스 연결 역할을 삭제합니다. 자세한 내용은 IAM 사용 설명서의 [서비스 연결 역할 삭제](#)를 참조하십시오.

AWS CodeConnections 서비스 연결 역할이 지원되는 리전

AWS CodeConnections에서는 서비스를 사용할 수 있는 모든 AWS 리전에서 서비스 연결 역할 사용을 지원합니다. 자세한 내용은 [AWS 리전 및 엔드포인트](#) 섹션을 참조하세요.

AWS CodeConnections의 AWS 관리형 정책

AWS 관리형 정책은 AWS에 의해 생성되고 관리되는 독립 실행형 정책입니다. AWS 관리형 정책은 사용자, 그룹 및 역할에 권한 할당을 시작할 수 있도록 많은 일반 사용 사례에 대한 권한을 제공하도록 설계되었습니다.

AWS 관리형 정책은 모든 AWS 고객이 사용할 수 있기 때문에 특정 사용 사례에 대해 최소 권한을 부여하지 않을 수 있습니다. 사용 사례에 고유한 [고객 관리형 정책](#)을 정의하여 권한을 줄이는 것이 좋습니다.

AWS 관리형 정책에서 정의한 권한은 변경할 수 없습니다. AWS에서 AWS 관리형 정책에 정의된 권한을 업데이트할 경우 정책이 연결되어 있는 모든 보안 주체 엔터티(사용자, 그룹 및 역할)에도 업데이트가 적용됩니다. 새로운 AWS 서비스를 시작하거나 새로운 API 작업을 기존 서비스에 이용하는 경우 AWS가 AWS 관리형 정책을 업데이트할 가능성이 높습니다.

자세한 내용은 IAM 사용자 설명서의 [AWS 관리형 정책](#)을 참조하세요.

AWS 관리형 정책: AWSGitSyncServiceRolePolicy

IAM 엔터티에 AWSGitSyncServiceRolePolicy를 연결할 수 없습니다. 이 정책은 AWS CodeConnections이(가) 사용자를 대신하여 작업을 수행할 수 있도록 서비스 연결 역할에 연결됩니다. 자세한 내용은 [AWS CodeConnections에 서비스 연결 역할 사용](#) 섹션을 참조하세요.

이 정책을 통해 고객은 Git 기반 리포지토리를 액세스하여 연결에 사용할 수 있습니다. 고객은 CreateRepositoryLink API를 사용한 후 이러한 리소스에 액세스할 수 있습니다.

권한 세부 정보

이 정책에는 다음 권한이 포함되어 있습니다.

- `codestar-connections` - 사용자에게 외부 Git 기반 리포지토리에 대한 연결을 생성할 수 있는 권한을 부여합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AccessGitRepos",
      "Effect": "Allow",
      "Action": [
        "codestar-connections:UseConnection"
      ],
      "Resource": "arn:aws:codestar-connections:*:*:connection/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
      }
    }
  ]
}
```

AWS CodeConnections가 AWS 관리형 정책으로 업데이트

이 서비스가 이러한 변경 내용을 추적하기 시작한 이후부터 AWS CodeConnections의 AWS 관리형 정책 업데이트에 관한 세부 정보를 봅니다. 이 페이지의 변경 사항에 대한 자동 알림을 받으려면 AWS CodeConnections [문서 기록 페이지](#)에서 RSS 피드를 구독하세요.

변경 사항	설명	날짜
AWSGitSyncServiceRolePolicy - 새 정책	AWS CodeConnections에 정책이 추가되었습니다.	2023년 11월 26일

변경 사항	설명	날짜
	AWS CodeConnections 사용자에게 연결된 Git 기반 리포지토리와 Git 동기화를 사용할 수 있는 권한을 부여합니다.	
AWS CodeConnections에서 변경 사항 추적 시작	AWS CodeConnections가 AWS 관리형 정책에 대한 변경 사항 추적을 시작했습니다.	2023년 11월 26일

AWS CodeStar 알림 및 AWS CodeStar 연결에 대한 규정 준수 검증

AWS CodeStar 알림 및 AWS CodeStar 연결은 AWS 규정 준수 프로그램의 범위에 포함되지 않습니다.

특정 규정 준수 프로그램의 범위 내 AWS 서비스 목록은 규정 준수 [프로그램별 범위 내 AWS 서비스를](#) 참조하십시오. 일반 정보는 [AWS 규정 준수 프로그램](#)을 참조하세요.

를 사용하여 타사 감사 보고서를 다운로드할 수 AWS Artifact 있습니다. 자세한 내용은 [AWS Artifact에서 보고서 다운로드](#)를 참조하십시오.

AWS CodeStar 알림 및 AWS CodeStar 연결 사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표, 관련 법률 및 규정에 따라 결정됩니다. AWS 규정 준수에 도움이 되는 다음 리소스를 제공합니다.

- [보안 및 규정 준수 킷스타트 가이드](#) - 이 배포 가이드에서는 아키텍처 고려 사항을 설명하고 보안 및 규정 준수에 중점을 둔 기본 환경을 배포하기 위한 단계를 제공합니다. AWS
- [AWS 규정 준수 리소스](#) — 이 통합 문서 및 가이드 모음은 해당 산업 및 지역에 적용될 수 있습니다.
- [AWS Config](#)— 이 AWS 서비스는 리소스 구성이 내부 관행, 업계 지침 및 규정을 얼마나 잘 준수하는지 평가합니다.
- [AWS Security Hub](#)— 이 AWS 서비스는 보안 업계 표준 및 모범 사례를 준수하는지 확인하는 데 도움이 되는 보안 상태를 종합적으로 보여줍니다.

AWS CodeStar Notifications 및 AWS CodeStar Connections의 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 내용은 [AWS 글로벌 인프라](#)를 참조하세요.

- 생성된 AWS 리전에 대한 알림 규칙이 생성됩니다. 둘 이상의 AWS 리전에 알림 규칙이 있는 경우 리전 선택기를 사용하여 각 AWS 리전의 알림 규칙을 검토합니다.
- AWS CodeStar Notifications는 Amazon Simple Notification Service(Amazon SNS) 주제를 알림 규칙 대상으로 사용합니다. Amazon SNS 주제 및 알림 규칙 대상에 관한 정보는 알림 규칙을 구성한 리전이 아닌 다른 AWS 리전에 저장될 수 있습니다.

AWS CodeStar Notifications 및 AWS CodeStar Connections의 인프라 보안

관리형 서비스인 AWS CodeStar Notifications 및 AWS CodeStar Connections는 [Amazon Web Services: 보안 프로세스 개요](#) 백서에 설명된 AWS 글로벌 네트워크 보안 절차로 보호됩니다.

AWS가 게시한 API 호출을 사용하여 네트워크를 통해 AWS CodeStar Notifications 및 AWS CodeStar Connections에 액세스합니다. 클라이언트가 전송 계층 보안(TLS) 1.0 이상을 지원해야 합니다. 클라이언트는 Ephemeral Diffie-Hellman(DHE) 또는 Elliptic Curve Ephemeral Diffie-Hellman(ECDHE)과 같은 완전 전송 보안(PFS)이 포함된 암호 제품군도 지원해야 합니다. 대부분의 최신 시스템은 이러한 모드를 지원합니다.

요청은 액세스 키 ID 및 IAM 보안 주체와 관련된 비밀 액세스 키를 사용하여 서명해야 합니다. 또는 [AWS Security Token Service](#)(AWS STS)를 사용하여 임시 보안 자격 증명을 생성하여 요청에 서명할 수 있습니다.

여러 리전에 걸친 AWS CodeConnections 리소스 간 트래픽

연결 기능을 사용하여 리소스 연결을 지원하는 경우, 여러분은 리소스가 생성된 리전 이외의 리전에서 리소스에 대한 연결을 제공하는 것과 관련하여 이를 목적으로만, 기반 서비스를 사용하는 AWS 리전 외부의 AWS 리전에서 해당 연결 리소스와 관련된 정보를 저장하고 처리하는 데 동의합니다.

자세한 내용은 [AWS CodeStar Connections의 글로벌 리소스](#) 섹션을 참조하세요.

Note

연결 기능을 사용하여 먼저 활성화할 필요가 없는 리전 내 리소스에 대한 연결을 활성화하는 경우 이전 주제에서 설명한 대로 정보가 저장 및 처리됩니다.
유럽(밀라노) 리전과 같이 먼저 활성화해야 하는 리전에서 설정된 연결의 경우 해당 리전에서만 해당 연결에 대한 정보가 저장 및 처리됩니다.

문서 이력

다음 표에서는 개발자 도구 콘솔의 이번 릴리스에 대한 문서를 설명합니다.

- AWS CodeStar Notifications API 버전: 2019-10-15
- AWS CodeStar Connections API 버전: 2019-12-01

변경 사항	설명	날짜
GitLab 자체 관리형 지원	AWS 리소스가 GitLab 자체 관리형과 상호 작용하도록 연결 및 호스트를 구성하는 데 대한 지원이 추가되었습니다. 자세한 내용은 호스트 생성 또는 업데이트 워크플로우 및 GitLab 자체 관리형에 대한 연결 생성 을 참조하십시오.	2023년 12월 28일
연결을 위한 새 리포지토리 링크 및 동기화 구성	리포지토리 링크 및 동기화 구성에 대한 정보가 추가되었습니다. 동기화 구성을 사용해 Git 리포지토리의 콘텐츠를 동기화하여 AWS CloudFormation 스택 리소스를 업데이트하세요. 자세한 내용은 리포지토리 링크 작업 및 동기화 구성 작업을 참조하세요.	2023년 11월 27일
연결의 서비스 연결 역할 지원	Git 리포지토리와 Git 동기화를 사용하도록 연결을 구성하는 데 대한 지원이 추가되었습니다. 자세한 내용은 AWS CodeStar 연결에 서비스 연결 역할 사용 및 관리형 정책을 참조하세요.	2023년 11월 26일

<u>GitLab 그룹 지원</u>	AWS 리소스가 GitLab 그룹과 상호 작용하도록 연결을 구성하는 데 대한 지원이 추가되었습니다. 자세한 내용은 <u>연결 생성 및 GitLab에 대한 연결 생성</u> 을 참조하세요.	2023년 9월 15일
<u>새로운 GitLab 제공업체 유형</u>	이제 GitLab에 대한 연결을 만들 수 있습니다. 자세한 내용은 <u>연결 생성 및 GitLab에 대한 연결 생성</u> 을 참조하세요.	2023년 8월 10일
<u>알림 규칙에 대한 새 대상 유형</u>	이제 Microsoft Teams 채널에 대해 구성된 AWS Chatbot 클라이언트를 알림 규칙의 대상으로 선택할 수 있습니다. 자세한 내용은 <u>알림 규칙 생성 및 알림 규칙 대상 작업을 참조</u> 하세요.	2023년 5월 17일
<u>유럽(밀라노) 리전에서 연결 사용 가능</u>	유럽(밀라노) 리전의 연결에 대한 정보를 추가했습니다. 자세한 내용은 <u>여러 리전에 걸친 AWS CodeStar Connections 리소스 간 트래픽</u> 을 참조하세요.	2023년 5월 17일
<u>리포지토리 권한과 관련된 연결 오류의 문제 해결이 추가됨</u>	GitHub 조직의 리포지토리에 대한 연결을 생성하는 경우 GitHub 조직 소유자여야 합니다. 자세한 내용은 <u>Connections error when connecting to GitHub</u> (GitHub에 연결할 때 연결 오류 발생)를 참조하세요.	2022년 8월 29일

[호스트 리소스 태깅에 대한 정보가 추가됨](#)

이제 콘솔과 CLI를 사용하여 호스트에 태깅할 수 있습니다. 자세한 내용은 [AWS CodeStar Connections에서 리소스 태그 지정](#)을 참조하세요.

2021년 4월 19일

[연결에 대한 VPC 엔드포인트 지원](#)

이제 연결에 VPC 엔드포인트를 사용할 수 있습니다. 자세한 내용은 [AWS CodeStar Connections 및 인터페이스 VPC 엔드포인트\(AWS PrivateLink\)](#)를 참조하세요.

2020년 11월 24일

[새로운 GitHub 및 GitHub Enterprise Cloud 공급자 유형](#)

이제 GitHub 및 GitHub Enterprise Cloud에 대한 연결을 생성할 수 있습니다. 자세한 내용은 [연결 생성 및 GitHub에 대한 연결 생성](#)을 참조하세요.

2020년 9월 30일

[GitHub Enterprise Server 공급자 유형 및 호스트 리소스가 추가됨](#)

연결을 위한 호스트 리소스에 대한 정보가 이 가이드에 추가되었습니다. 이제 GitHub Enterprise Server에 대한 연결을 생성할 수 있습니다. 자세한 내용은 [연결 생성 및 호스트 작업을 참조](#)하세요. 이는 개발자 도구 콘솔 사용 설명서에서 설명하는 연결 기능의 정식 출시 릴리스입니다.

2020년 6월 29일

[연결의 사용 및 태깅에 대한 정보가 추가됨](#)

콘솔의 연결 기능에 대한 정보가 이 가이드에 추가되었습니다. 관련 개념, 시작하기 단계, 정책 예를 포함한 권한 참조, 연결을 생성, 표시 및 태깅하는 단계를 참조할 수 있습니다. 자세한 내용은 [연결이란?](#), [연결 개념](#), [연결 시작](#), [연결 생성](#), [AWS CodeStar Connections에서 리소스 태그 지정](#), [보안](#), [연결 할당량](#), [문제 해결](#) 및 [AWS CloudTrail를 사용하는 AWS CodeStar Connections API 호출](#)을 참조하세요. 추가 공급자 작업(권한이 필요한 작업) 목록을 보려면 [ProviderType 작업](#)을 참조하세요.

2020년 6월 28일

[알림 규칙에 대한 새 대상 유형](#)

이제 Slack 채널에 대해 구성된 AWS Chatbot 클라이언트를 알림 규칙의 대상으로 선택할 수 있습니다. 자세한 내용은 [알림 규칙 생성 및 알림 규칙 대상 작업](#)을 참조하세요.

2020년 4월 2일

[추가 AWS CodeCommit 이벤트에 대한 알림이 추가됨](#)

이제 풀 요청 승인과 관련된 이벤트에 대한 알림을 구성할 수 있습니다. 자세한 내용은 [리포지토리의 알림 규칙에 대한 이벤트 및 CodeCommit의 풀 요청 작업](#)을 참조하세요.

2020년 2월 10일

2개의 추가 AWS 리전에서 알림 사용 가능	이제 개발자 도구 콘솔에서 중동(바레인) 및 아시아 태평양(홍콩)의 알림을 지원합니다. 자세한 내용은 AWS 일반 참조에서 AWS CodeStar 알림 을 참조하세요.	2020년 2월 5일
암호화된 Amazon SNS 주제에 대한 지원이 추가됨	암호화된 Amazon SNS 주제를 알림 대상으로 사용하기 위한 지침이 추가되었습니다. 자세한 내용은 알림에 대한 Amazon SNS 주제 구성 을 참조하세요.	2020년 2월 4일
알림에 CodeCommit에 대한 세션 태그 정보가 포함될 수 있음	CodeCommit에 대한 알림에는 이제 세션 태그의 사용을 통해 표시 이름이나 이메일 주소와 같은 사용자 자격 증명 정보가 포함될 수 있습니다. 자세한 내용은 개념 및 태그를 사용하여 CodeCommit에 자격 증명 정보 제공 을 참조하세요.	2019년 12월 19일
최초 릴리스	개발자 도구 콘솔 사용 설명서가 처음으로 릴리스되었습니다.	2019년 11월 5일

AWS 용어집

최신 AWS 용어는 AWS 용어집 참조서의 [AWS 용어집](#)을 참조하십시오.

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