사용 설명서

AWS DevOps 에이전트

AWS DevOps 에이전트: 사용 설명서

Table of Contents

About AWS DevOps 에이전트

AWS DevOps Agent는 인시던트를 해결하고 선제적으로 방지하여 신뢰성과 성능을 지속적으로 개선하는 최전선 에이전트입니다.

AWS DevOps Agent는 인시던트를 조사하고 숙련된 DevOps 엔지니어로서 운영 개선을 식별합니다.

에이전트는 다음을 통해 작동합니다.

- 리소스 및 해당 관계를 학습합니다.
- 관찰성 도구, 기술, 코드 리포지토리 및 CI/CD 파이프라인 작업.
- 원격 측정, 코드 및 배포 데이터를 상호 연관시켜 애플리케이션 리소스 간의 관계를 이해합니다.
- 멀티클라우드 및 하이브리드 환경에서 애플리케이션 지원.

주요 기능

AWS DevOps Agent는 다음 기능을 통해 포괄적인 인시던트 대응 및 예방 기능을 제공합니다.

상시 작동, 자율 인시던트 대응

AWS DevOps Agent는 문제가 발생하는 즉시 문제를 자율적으로 조사합니다.

- 자동화된 인시던트 조사 - 알림 또는 지원 티켓이 도착하면 즉시 조사를 시작합니다.
- AWS DevOps 에이전트 채팅 - DevOps 에이전트 스페이스 웹 앱 전체에서 자연어를 사용하여 인프라를 쿼리하고, 시스템 상태를 분석하고, 조사를 안내합니다. 채팅은 토폴로지의 리소스에 대해 질문하든, 조사를 조정하든, 예방의 권장 사항을 필터링하든 관계없이 보고 있는 페이지를 기반으로 컨텍스트 인식 응답을 제공합니다.
- 세부 완화 계획 - 인시던트를 해결하고, 성공을 검증하고, 필요한 경우 변경 사항을 되돌리기 위한 특정 조치를 제공합니다.
- 자동화된 인시던트 조정 - Slack 및 ServiceNow와 같은 선호하는 통신 채널을 통해 관찰, 조사 결과 및 완화 단계를 라우팅합니다.
- AWS 지원 통합 - AWS 지원 전문가에게 즉시 제공되는 컨텍스트를 사용하여 조사에서 직접 AWS 지원 사례 생성

향후 인시던트 예방

AWS DevOps Agent는 과거 인시던트의 패턴을 분석하여 사후 대응 소방에서 선제적 운영 개선으로 전환하는 데 도움이 됩니다.

- 대상 권장 사항 - 관찰성(모니터링, 알림, 로깅), 인프라 최적화(자동 조정, 용량 조정), 배포 파이프라인 개선(테스트, 검증)이라는 네 가지 주요 영역을 강화하는 구체적이고 실행 가능한 개선 사항을 제공합니다.
- 지속적 학습 - 팀의 피드백을 기반으로 권장 사항을 구체화합니다.

DevOps 도구를 더 많이 활용

AWS DevOps Agent는 워크플로를 변경하지 않고도 기존 도구와 통합됩니다.

- 애플리케이션 리소스 매핑 - 애플리케이션 리소스 및 해당 관계의 토폴로지 그래프를 빌드합니다.
- 기본 제공 통합 - 널리 사용되는 관찰성 도구(Amazon CloudWatch, Dynatrace, Datadog, New Relic 및 Splunk), 코드 리포지토리 및 CI/CD 파이프라인(GitHub Actions 및 리포지토리, GitLab 워크플로 및 리포지토리)과 함께 사용할 수 있습니다.
- 사용자 지정 도구 통합 - 추가 도구를 위해 자체 모델 컨텍스트 프로토콜(MCP) 서버에 연결하여 기능을 확장합니다.
- 대화형 인프라 쿼리 - 자연어를 사용하여 여러 콘솔을 탐색하지 않고 AWS 리소스, 시스템 지표 및 경보 상태를 쿼리합니다. 채팅은 컨텍스트를 이해하고 후속 질문에 대한 대화 기록을 유지합니다.

AWS DevOps 에이전트 작동 방식

AWS DevOps Agent는 이중 콘솔 아키텍처를 통해 작동합니다. 관리자는 AWS 관리 콘솔을 사용하여 에이전트 스페이스를 생성 및 관리하고, 통합을 구성하고, 액세스 제어를 설정합니다. 운영 팀은 AWS DevOps Agent 웹 앱을 사용하여 day-to-day 인시던트 대응 및 조사 활동을 수행합니다. 웹 앱은 운영자가 에이전트 조사와 상호 작용하고, 교차 계정 애플리케이션 토폴로지를 탐색하고, 관찰성, 코드, 파이프라인 및 인프라 아키텍처의 예방 개선 사항에 대해 알아볼 수 있는 곳입니다. 자세한 내용은 the section called “선제적 인시던트 예방”를 참조하세요.

서비스는 에이전트 스페이스를 중심으로 구성됩니다. 에이전트 스페이스는 AWS DevOps가 액세스하고 조사할 수 있는 내용을 정의하는 논리적 컨테이너입니다. 각 에이전트 스페이스에는 AWS 계정 구성, 타사 도구 통합 및 액세스 권한이 포함되어 있습니다. 자세한 내용은 the section called “DevOps 에이전트 스페이스란 무엇입니까?”를 참조하세요.

AWS DevOps Agent는 리소스와 해당 관계를 매핑하는 애플리케이션 토폴로지를 자동으로 빌드합니다. 이 토폴로지는 서비스가 조사 중에 애플리케이션 아키텍처를 이해하는 데 도움이 됩니다. 자세한 내용은 the section called “DevOps 에이전트 토폴로지란 무엇입니까?”를 참조하세요.

이점

- 평균 해결 시간(MTTR) 단축 - 자율 조사가 즉시 시작되어 인시던트 해결이 몇 시간에서 몇 분으로 가속화됩니다.
- 반복되는 인시던트 방지 - 대상 추천은 근본 원인을 해결하고 시스템 복원력을 강화합니다.
- 운영 효율성 개선 - 팀이 반복적인 조사 작업에서 벗어나 혁신에 집중할 수 있습니다.
- 기존 워크플로 내에서 작업 - 중단 없이 기존 도구 및 프로세스와 통합

DevOps 에이전트 웹 앱이란 무엇입니까?

AWS DevOps Agent는 일상적인 운영 활동과 관리 기능을 분리하는 이중 콘솔 아키텍처day-to-day 사용합니다. 이 설계를 통해 관리자는 서비스를 구성하고 운영 팀은 인시던트 대응 및 예방에 집중할 수 있습니다.

콘솔

AWS DevOps Agent는 다음과 같은 두 가지 인터페이스를 제공합니다.

- AWS 관리 콘솔 - 관리자는 AWS 관리 콘솔을 사용하여 AWS DevOps 에이전트를 설정하고 관리합니다. 이 콘솔에서는 AWS 서비스와 타사 도구를 the section called “에이전트 스페이스 생성” 연결하고 조직의 액세스 권한을 관리할 수 있습니다.
- DevOps Agent 웹 앱 - 운영 팀은 일일 인시던트 대응 활동에 DevOps Agent Space 웹 앱을 사용합니다. 이 독립 실행형 애플리케이션은 대기 중인 엔지니어가 조사를 시작하고, 자연어 채팅을 통해 에이전트와 상호 작용하고, 애플리케이션 토폴로지를 보고, 인시던트 방지 권장 사항을 검토할 수 있는 인터페이스를 제공합니다.

웹 앱 기능

DevOps Agent 웹 앱은 다음과 같은 기본 기능을 제공합니다.

- 인시던트 대응 - 이 페이지에서 인시던트 조사를 생성 및 추적하고 인시던트 해결을 위한 완화 계획을 생성합니다.

- 인시던트 예방 - 예방 페이지에서 향후 인시던트를 방지하기 위해 관찰성 태세, 제공 프로세스 및 인프라 아키텍처를 개선하기 위한 권장 사항을 찾을 수 있습니다.
- 토폴로지 - 토폴로지 페이지는 연결된 계정의 모든 리소스에서 계정 리소스 및 해당 관계를 대화형으로 시각적으로 표현합니다. "표시" 드롭다운을 사용하여 시스템, 컨테이너 및 리소스 보기 간에 전환하여 다양한 수준의 세부 정보로 토폴로지를 볼 수 있습니다.
- 기술 - 특수 기능을 갖춘 extend AWS DevOps 에이전트를 확장하는 모듈식 명령 세트입니다. 기술에는 인프라에 맞게 조정된 도메인 지식, 조사 방법론 및 도구 구성이 포함됩니다. 각 기술은 특정 도구를 활성화하고 조사와 관련된 경우에만 지침을 점진적으로 공개합니다.
- 자연어 채팅 인터페이스 - 웹 앱 전체에서 사용할 수 있는 Chat은 인프라를 쿼리하고, 시스템 상태를 분석하고, 자연어를 사용하여 조사를 수행할 수 있는 AI 기반 대화형 어시스턴트입니다. 채팅은 보고 있는 페이지를 기반으로 컨텍스트 인식 응답을 제공합니다.

Authentication

AWS DevOps Agent는 다양한 조직 요구 사항을 수용할 수 있는 유연한 인증 방법을 지원합니다.

- IAM Identity Center 통합(사용자 액세스) - 조직은 AWS Identity Center(IAM Identity Center)를 사용하여 DevOps Agent Space 웹 앱에 대한 사용자 액세스를 중앙에서 관리할 수 있습니다. IAM Identity Center는 Okta, Ping Identity 및 Microsoft Entra ID와 같은 공급자를 포함한 표준 OIDC 및 SAML 프로토콜을 통해 외부 ID 공급자와 페더레이션할 수 있습니다. 이 방법은 자격 증명 공급자의 다중 인증을 지원합니다.
- 외부 ID 제공업체(IdP) 인증 - 조직은 IAM Identity Center 없이 Okta 또는 Microsoft Entra ID와 같은 OIDC 호환 ID 제공업체를 에이전트 스페이스 웹 앱에 직접 연결할 수 있습니다. 사용자는 IdP를 통해 회사 자격 증명으로 로그인합니다. 설정 지침은 단원을 참조하십시오the section called “외부 ID 제공업체(IdP) 인증 설정”.
- IAM 인증 링크(관리자 액세스) - 대체 방법을 사용하면 기존 콘솔 세션을 사용하여 AWS Management Console에서 웹 앱에 직접 액세스할 수 있습니다. 이 옵션은 전체 Identity Center 통합을 구현하기 전에 유용하지만 세션은 10분으로 제한됩니다.

DevOps 에이전트 스페이스란 무엇입니까?

DevOps 에이전트 스페이스는 AWS DevOps 에이전트가 액세스할 수 있는 도구와 인프라를 정의하는 논리적 컨테이너입니다. 각 에이전트 스페이스는 자체 AWS 계정 액세스, 타사 통합 및 사용자 권한으로 독립적으로 작동합니다.

에이전트 공간은 인시던트 대응 중에 AWS DevOps가 액세스하고 조사할 수 있는 항목의 경계를 나타냅니다. 에이전트 스페이스를 생성할 때 에이전트가 액세스할 수 있는 AWS 계정, 연결할 수 있는 외부 도구, 조직의 사용자가 에이전트와 상호 작용할 수 있는 사용자를 정의합니다.

각 에이전트 스페이스는 AWS DevOps 에이전트의 독립적인 배포 역할을 합니다. AWS 관리 콘솔을 통해 에이전트 스페이스를 구성하는 동안 운영 팀은 에이전트 스페이스의 웹 앱을 사용하여 해당 스페이스 내에서 조사를 수행하고 권장 사항을 검토합니다.

에이전트 공간을 격리하는 방법

에이전트 스페이스는 격리를 유지하여 보안을 보장하고 다양한 환경 또는 팀에서 의도하지 않은 액세스를 방지합니다.

- AWS 계정 격리 - 각 에이전트 스페이스는 특정 AWS 계정 및 리소스에만 액세스 권한을 부여하는 전용 IAM 역할을 사용합니다. 에이전트는 에이전트 스페이스에 대해 명시적으로 구성된 리소스 외부의 AWS 리소스에 액세스할 수 없습니다.
- 사용자 액세스 격리 - 각 에이전트 스페이스에 액세스할 수 있는 사용자 또는 그룹을 제어합니다. 이렇게 하면 액세스 권한을 조직 구조에 맞게 조정하여 팀이 지정된 에이전트 스페이스와만 상호 작용할 수 있습니다.
- 데이터 격리 - 조사 데이터, 인시던트 기록 및 권장 사항은 각 에이전트 스페이스 내에서 별도로 유지됩니다. 한 에이전트 스페이스의 정보는 다른 에이전트 스페이스에서 보거나 액세스할 수 없습니다.
- 채팅 데이터 격리 - 채팅 대화 기록도 각 에이전트 스페이스 내에서 격리됩니다. 한 에이전트 스페이스의 대화 및 쿼리는 다른 에이전트 스페이스에서 보거나 액세스할 수 없습니다.

에이전트 스페이스 웹 앱

각 에이전트 스페이스에는 AWS 관리 콘솔 외부에서 액세스할 수 있는 전용 웹 앱이 있습니다. 웹 앱에 대한 자세한 내용은 the section called “DevOps 에이전트 웹 앱이란 무엇입니까?” 섹션을 참조하세요.

여러 에이전트 스페이스를 사용해야 하는 경우

다양한 조직 요구 사항을 지원하기 위해 여러 에이전트 스페이스를 생성하는 것이 좋습니다.

- 팀 분리 - 다양한 애플리케이션 팀 또는 사업부를 위한 전용 에이전트 스페이스를 생성하여 에이전트 스페이스의 명확한 소유권 경계를 유지합니다.
- 환경 격리 - 프로덕션 환경과 비프로덕션 환경을 서로 다른 에이전트 공간으로 분리하여 우발적인 환경 간 액세스를 방지합니다.

- 서비스 경계 - 에이전트 공간을 특정 서비스 또는 애플리케이션 경계와 정렬하여 조사에 집중하고 관련성을 유지합니다.
- 규정 준수 요구 사항 - 규제 요구 사항을 충족하도록 다양한 액세스 제어 또는 데이터 레지던시 설정을 사용하여 별도의 에이전트 스페이스를 구성합니다.

> **Note**
>
> 여러 에이전트 스페이스를 생성할 때 전용 AWS 계정을 에이전트 스페이스의 기본 계정으로 사용하고 개별 애플리케이션 계정을 보조 계정으로 연결할 수 있습니다. 이 접근 방식을 사용하면 자동 역할 생성을 사용하는 경우에도 각 에이전트 스페이스가 의도한 범위와 관련된 리소스에만 액세스할 수 있도록 하면서 세분화된 액세스 제어를 유지할 수 있습니다.

DevOps 에이전트 토폴로지란 무엇입니까?

AWS DevOps Agent는 애플리케이션 내의 리소스와 관계를 자동으로 검색 및 시각화하고 결과 토폴로지를 사용하여 인시던트 조사 중 및 예방 권장 사항 작성 시 인프라를 이해합니다.

토폴로지 그래프 생성 방법

AWS DevOps Agent는 여러 자동화된 프로세스를 통해 토폴로지 그래프를 빌드합니다.

- 리소스 검색 - 에이전트는 자동으로 AWS 계정을 스캔하여 애플리케이션의 일부인 컴퓨팅 인스턴스, 스토리지 서비스, 네트워킹 구성 요소 및 데이터베이스와 같은 리소스를 식별합니다.
- 관계 감지 - 에이전트는 구성 데이터, CloudFormation 스택 및 리소스 태그를 분석하여 리소스가 서로 어떤 관련이 있는지 결정합니다.
- 코드 및 배포 매핑 - CI/CD 파이프라인에 연결되면 에이전트는 인프라 리소스를 배포 프로세스와 변경된 애플리케이션 및 인프라 코드에 다시 연결합니다.
- 관찰성 동작 매핑 - Amazon CloudWatch Application Signals 및 Dynatrace와 같은 관찰성 시스템의 데이터를 사용하여 리소스 간의 관계를 나타내는 관찰된 동작을 식별합니다.

주요 기능

리소스 매핑은 인시던트 조사 및 예방을 개선하는 몇 가지 기능을 제공합니다.

- 대화형 시각화 - 운영자 웹 앱의 대화형 그래프를 통해 애플리케이션 토폴로지를 탐색합니다. 토폴로지를 확대/축소하고 탐색하여 리소스 간의 복잡한 관계를 이해할 수 있습니다. 채팅을 사용하여 '이

DynamoDB 테이블에 연결된 모든 Lambda 함수 표시' 또는 '이 경보의 영향을 받는 리소스는 무엇입니까?'와 같은 자연어를 사용하여 토폴로지 정보를 쿼리할 수도 있습니다.

- 컨텍스트 조사 - 인시던트 조사 중에 AWS DevOps 에이전트는 리소스 토폴로지의 지원을 받아 영향을 받는 구성 요소를 식별하고, 폭발 반경을 이해하고, 시스템을 통해 영향 경로를 추적합니다.
- 근본 원인 분석 - 리소스 관계를 자세히 이해하면 상호 종속성이 많은 복잡한 분산 시스템에서도 문제가 발생하는 위치를 정확히 파악할 수 있습니다.
- 영향 평가 - 인시던트를 분석할 때 에이전트는 토폴로지에서 종속성 체인을 식별하여 영향을 받을 수 있는 다운스트림 서비스를 더 잘 확인할 수 있습니다.
- 예방 권장 사항 - 에이전트는 토폴로지 인사이트를 사용하여 복원력 개선을 위한 대상 권장 사항을 작성하여 시스템 안정성에 가장 큰 영향을 미치는 변경 사항을 제안합니다.

토폴로지 보기

운영자 웹 앱의 토폴로지 페이지에 있는 토폴로지 시각화는 여러 수준의 세부 정보를 제공합니다.

- 학습됨 - 에이전트 공간 이해 스킬에서 생성된 기본 보기입니다. 논리적 서비스 및 요청 경로별로 구성된 인프라의 구조화된 요약을 표시합니다.
- 시스템 - 상위 수준 계정 및 리전 경계를 표시합니다.
- 컨테이너 - 관련 리소스가 포함된 CloudFormation 스택과 같은 배포 스택을 표시합니다.
- 구성 요소 - 컨테이너 내의 개별 구성 요소와 해당 관계를 표시합니다.
- 모든 리소스 - 검색된 모든 리소스 및 해당 관계가 포함된 전체 보기를 표시합니다.

리소스 검색

리소스는 다음 두 가지 방법을 통해 검색됩니다.

- CloudFormation 스택 - 에이전트는 기본 AWS 계정 및 연결된 보조 계정의 모든 CloudFormation 스택과 해당 리소스를 나열합니다. 이는 Cloud AWS Development Kit(AWS CDK)를 포함하여 배포에 CloudFormation을 사용하는 모든 infrastructure-as-code 도구에 지원됩니다.
- Resource Explorer - CloudFormation에서 배포되지 않은 리소스의 경우 태그가 지정된 리소스가 AWS Resource Explorer에서 검색됩니다. 대상 AWS 계정에 Resource Explorer가 활성화되어 있어야 합니다. 이는 AWS 관리 콘솔, AWS 서비스 APIs 또는 기타 infrastructure-as-code 프레임워크를 통해 배포된 리소스의 애플리케이션 경계를 식별하는 데 유용합니다.

토폴로지 이외의 조사 범위

애플리케이션 토폴로지는 조사 중에 중요한 컨텍스트를 제공하지만 AWS , DevOps 에이전트는 토폴로지에 표시된 리소스만 조사하는 것으로 제한되지 않습니다. 에이전트는 AWS 서비스 APIs 또는 연결된 관찰성 도구와 같은 추가 데이터 소스를 사용하여 애플리케이션 토폴로지에 없는 리소스를 조사할 수 있습니다.

에이전트가 액세스할 수 있는 리소스를 제한하려면 에이전트에 할당된 역할이 교차 계정 리소스에 액세스하도록 정책을 제한합니다. 자세한 내용은 the section called “AWS 계정의 에이전트 액세스 제한” 단원을 참조하십시오.

토폴로지 및 에이전트 공간 이해 기술

토폴로지 그래프는 조사 중에 사용할 인프라의 구조화된 요약을 인코딩하는 에이전트 공간 이해 학습 스킬에 제공됩니다. 새 에이전트 공간에 대한 토폴로지 검색이 완료되면 시스템에서 에이전트 공간 이해 스킬을 자동으로 생성합니다. 학습한 기술에 대한 자세한 내용은 섹션을 참조하세요the section called “학습한 기술”.

DevOps 에이전트 기술

AWS DevOps Agent Skills는 인프라 및 운영 워크플로에 맞게 조정된 전문 도메인 지식 및 조사 방법론을 사용하여 에이전트의 기능을 확장하는 모듈식 지침 세트입니다.

Skills란 무엇입니까?

Skills는 AWS DevOps Agent에 특수 기능을 제공하는 마크다운 지침이 포함된 독립형 디렉터리입니다. AWS DevOps Agent는 에이전트 지침 및 리소스를 패키징하기 위한 개방형 표준인 에이전트 기술 사양의 하위 집합을 지원하며 마크다운 지침, PDFs, 이미지 및 데이터 파일과 같은 실행 불가능한 문서만 지원합니다.

모든 스킬에는 AWS DevOps 에이전트에 제공하려는 지침이 포함된 SKILL.md 파일이 필요합니다. 필요한 SKILL.md 파일 외에도 기술에는 다음이 포함될 수 있습니다.

- 특정 시나리오 또는 인프라 유형에 대한 조사 워크플로입니다.
- 아키텍처 패턴 및 운영 절차를 포함한 참조 자료.
- 에이전트 유형 타겟팅 - 특정 에이전트 유형(일반, 온디맨드, 인시던트 분류, 인시던트 RCA, 인시던트 완화, 평가)을 대상으로 하여 컨텍스트 소비를 줄이고 에이전트 포커스를 개선할 수 있습니다.

Skills를 사용하는 이유

Skills transform AWS DevOps 에이전트는 범용 어시스턴트에서 인프라 및 운영 워크플로를 위한 전문가로 전환됩니다. 채팅 메시지에 제공된 일회성 지침과 달리 Skills는 AWS DevOps Agent에서 수행하는 작업과 관련이 있을 때 자동으로 로드되는 재사용 가능한 기능입니다.

주요 이점:

- 에이전트 전문화 - 인프라 및 운영 패턴과 관련된 조사 절차, 모범 사례 및 조직 지식을 갖춘 Tailor AWS DevOps 에이전트입니다.
- 반복 감소 - 및 AWS DevOps Agent가 모든 관련 조사에서 자동으로 조사 워크플로를 사용하면 동일한 지침을 반복적으로 제공할 필요가 없습니다.
- Compose 기능 - 여러 스킬을 결합하여 end-to-end 조사 워크플로를 구축합니다. AWS DevOps Agent는 실행 중에 사용자 지정 CI/CD 파이프라인에서 배포를 검색하는 스킬과 코드 리포지토리를 검색하는 스킬과 같은 여러 스킬을 읽습니다.
- Amplify 사용자 지정 도구 - 사용자 지정 MCP 서버 도구를 효과적으로 사용하여에서 AWS DevOps 에이전트를 안내하는 기술을 생성합니다. 스킬은 특정 도구를 호출하는 시기, 다양한 시나리오에 사용할 파라미터, 결과를 해석하여 인프라별 워크플로를 수행하는 방법을 문서화할 수 있습니다.

기술 작동 방식

AWS DevOps 에이전트는 관련 작업을 발견하면 적절한 기술을 로드하고 지침에 따라 조사를 안내합니다. 예를 들어 "데이터베이스 성능 조사" 스킬에는 RDS 제한 문제를 분석하여 에이전트가 경보 상태를 체계적으로 확인하고, 연결 지표를 분석하고, 느린 쿼리를 식별할 수 있는 step-by-step 절차가 포함될 수 있습니다.

스킬 구조

스킬은 다음을 포함하는 디렉터리로 구성됩니다.

```
my-skill/
### SKILL.md              # Main skill instructions
### references/          # Optional: additional reference documentation
### assets/              # Optional: images, diagrams, data files
```

SKILL.md

는 유일한 필수 파일SKILL.md입니다. 여기에는 마크다운 형식으로 작성된 핵심 지침이 포함되어 있습니다. 이 파일은 다음과 같아야 합니다.

- 스킬을 사용하는 시기와 방법을 설명합니다.
- step-by-step 조사 절차를 제공합니다.
- 다양한 시나리오에 대한 의사 결정 트리를 포함합니다.
- 예상 출력 및 성공 기준을 문서화합니다.

프론트미터

Frontmatter는 SKILL.md 파일 상단의 메타데이터 블록으로, --- 구분 기호 사이에 묶여 있습니다. 여기에는 AWS DevOps 에이전트가 조사 또는 작업 중에 Skill을 활성화할 시기를 결정하는 데 사용하는 name 및 description 필드가 포함되어 있습니다.

```
---
name: rds-performance-investigation
description: Investigation procedures for RDS performance issues including
  connection exhaustion, slow queries, replication lag, and storage capacity.
  Use this skill when investigating database latency, connection errors, or
  read/write performance degradation.
---
```

name - Skill의 고유 식별자입니다. 소문자, 숫자 및 하이픈만 사용합니다(최대 64자). 하이픈으로 시작하거나 끝나지 않아야 합니다.

설명 - AWS DevOps 에이전트가이 Skill. AWS DevOps 에이전트를 사용해야 하는 시기와 이유에 대한 자세한 설명으로,이 필드를 평가하여 Skill이 현재 작업과 관련이 있는지 여부를 결정합니다. 모호하거나 누락된 설명으로 인해 지침이 잘 작성되더라도 에이전트가 스킬을 완전히 건너뛸 수 있습니다.

중요 - 에이전트의 관점에서 설명을 작성합니다. Skill을 트리거해야 하는 특정 시나리오, 서비스, 오류 유형 또는 증상을 포함합니다. 예를 들어 "Amazon RDS 인스턴스의 데이터베이스 지연 시간, 연결 오류 또는 쿼리 제한 시간을 조사할 때이 스킬 사용"은 "RDS 스킬"보다 더 효과적입니다.

UI에서 Skill을 생성하면 사용자가 제공한 이름과 설명에서 시스템이 자동으로 프론트미터를 생성합니다. zip 파일로 업로드된 스킬은 SKILL.md 파일에 프론트미터를 포함해야 합니다.

예: 스킬 완성

다음 예제에서는 RDS 성능 문제를 조사하기 위한 완전하고 체계적인 기술을 보여줍니다. 디렉터리 구조, SKILL.md frontmatter, 실행 가능한 조사 절차 및 보조 참조 파일을 보여줍니다.

디렉터리 구조:

```
rds-performance-investigation/
### SKILL.md
### references/
#   ### rds-metrics-reference.md
### assets/
    ### rds-investigation-flowchart.png
```

SKILL.md:

```
---
name: rds-performance-investigation
description: Investigation procedures for RDS performance issues including
  connection exhaustion, slow queries, replication lag, and storage capacity.
  Use this skill when investigating database latency, connection errors, or
  read/write performance degradation.
---

# RDS Performance Investigation

Use this skill when customers report database latency, connection errors,
query timeouts, or read/write performance degradation.


## Step 1: Check alarm status

Query CloudWatch for active alarms on the affected RDS instance. Look for:
- `DatabaseConnections` exceeding 80% of max_connections
- `ReadLatency` or `WriteLatency` above 20ms
- `FreeStorageSpace` below 20% of total storage
- `ReplicaLag` above 30 seconds (read replicas only)


## Step 2: Analyze connection metrics

Retrieve `DatabaseConnections` over the past hour. If connections are near
the max_connections limit, check for connection pool misconfiguration or
```

```
long-running idle connections.


## Step 3: Identify slow queries

Use Performance Insights (`pi:GetResourceMetrics`) to retrieve the top SQL
statements by average active sessions. Focus on queries with high `db.load`
contribution or frequent I/O waits.


## Step 4: Summarize findings

Provide a summary with:
1. Current performance status (healthy / degraded / critical)
2. Root cause hypothesis with supporting metrics
3. Recommended remediation steps ranked by priority
```

참조/rds-metrics-reference.md:

```
# RDS CloudWatch Metrics Reference


| Metric | Normal Range | Investigation Threshold |
|---|---|---|
| DatabaseConnections | < 70% max_connections | > 80% max_connections |
| ReadLatency | < 5ms | > 20ms |
| WriteLatency | < 5ms | > 20ms |
| FreeStorageSpace | > 30% total storage | < 20% total storage |
| ReplicaLag | < 5 seconds | > 30 seconds |
| CPUUtilization | < 70% | > 85% |
```

기술 생성

스킬을 생성하기 전에 에이전트 스페이스가 있어야 합니다. 자세한 내용은 the section called “에이전트 스페이스 생성” 단원을 참조하십시오.

워크플로 기본 설정과 스킬 복잡성에 따라 두 가지 방법으로 스킬을 생성할 수 있습니다.

UI에서 스킬 생성

AWS DevOps 에이전트 운영자 웹 앱에서 생성된 기술에는 단일 SKILL.md 파일에 이름, 설명 및 지침이 포함되어 있습니다.

UI에서 스킬을 생성하려면:

- 에이전트 스페이스 운영자 웹 앱의 기술 페이지로 이동합니다.
- "기술 추가"를 클릭합니다.
- 모달에서 "기술 생성"을 선택합니다.
- 스킬 양식을 작성합니다.
  - 이름 - 소문자, 숫자 및 하이픈만 가능합니다(최대 64자). 하이픈으로 시작하거나 끝나지 않아야 합니다. 예시: `rds-throttling-investigation`
  - 설명 -이 기술을 사용하는 시기에 대한 간략한 설명입니다(최소 100자 권장, 최대 1,024자). 이렇게 하면 에이전트가 스킬을 활성화할 시기를 결정하는 데 도움이 됩니다.
  - 상태 - 활성(기본값) 또는 비활성으로 설정합니다. 비활성 스킬은 에이전트가 사용하지 않습니다.
  - 에이전트 유형 -이 스킬을 사용할 수 있는 에이전트 유형을 하나 이상 선택합니다. 일반은 기본적으로 선택되며 모든 에이전트 유형에서 스킬을 사용할 수 있습니다. 특정 에이전트를 대상으로 지정하려면 일반를 선택 취소하고 온디맨드, 인시던트 분류, 인시던트 RCA, 인시던트 완화 또는 평가 중에서 선택합니다.
  - 지침 Step-by-step 절차입니다. 구체적이고 실행 가능해야 합니다.
- "생성"을 클릭하여 스킬을 저장합니다.

시스템은 적절한 전면 구조로 SKILL.md 파일을 자동으로 생성합니다.

UI에서 생성된 스킬을 편집하려면:

- 스킬 목록에서 스킬로 이동하여 스킬을 클릭하여 엽니다.
- 편집을 클릭합니다.
- 이름, 설명 또는 지침을 수정합니다.
- 저장을 클릭하여 스킬을 업데이트합니다.

스킬 업로드

zip 파일로 업로드된 기술에는 SKILL.md 파일과 참조 자료 또는 자산과 같은 추가 리소스가 포함됩니다.

스킬 구조:

```
my-skill.zip
```

```
### SKILL.md              # Required: main skill instructions
### references/           # Optional: reference documentation
#   ### architecture.md
#   ### troubleshooting.md
### assets/               # Optional: images, diagrams, data files
    ### topology.png
    ### metrics.csv
```

SKILL.md frontmatter 요구 사항:

zip 파일로 업로드된 스킬에는 name 및 description 필드와 함께 SKILL.md 프런트미터가 포함되어야 합니다. AWS DevOps Agent는 이러한 필드를 사용하여 스킬을 활성화할 시기를 결정합니다. 유효 프론트미터 작성에 대한 자세한 내용은이 주제 앞부분의 프론트미터 섹션을 참조하세요.

```
---
name: rds-performance-analysis
description: Comprehensive RDS performance investigation procedures
  for connection exhaustion, slow queries, and storage capacity issues.
  Use when investigating database latency or read/write degradation.
---


# RDS Performance Analysis


[Your skill instructions here...]
```

zip 업로드를 통해 스킬을 생성하려면:

- 위의 구조에 따라 스킬 파일을 사용하여 디렉터리를 생성합니다.
- SKILL.md 적절한 프론트미터(이름 및 설명)가 포함되어 있는지 확인합니다.
- 디렉터리를 .zip 파일로 압축합니다.
- 에이전트 스페이스 운영자 웹 앱의 기술 페이지로 이동합니다.
- "기술 추가"를 클릭합니다.
- 모달에서 "기술 업로드"를 선택합니다.
- .zip 파일을 끌어서 놓거나 클릭하여 탐색합니다(ZIP 파일만 해당, 최대 6MB).
- 이 기술을 사용할 수 있는 에이전트 유형을 하나 이상 선택합니다(일반적으로 일반이 선택되고 모든 에이전트 유형에 적용됨, 특히 온디맨드, 인시던트 분류, 인시던트 RCA, 인시던트 완화 또는 평가를 대상으로 선택 취소).

- zip 파일 요구 사항 및 검증 결과를 검토합니다.
- "업로드"를 클릭하여 에이전트 스페이스에 스킬을 추가합니다.

zip 파일로 업로드되는 기술에 대한 중요한 제한 사항:

- 스크립트는 현재 지원되지 않음 - `scripts/` 디렉터리에 스크립트가 포함된 스킬은 업로드 중에 거부됩니다. 에이전트가 보안 코딩 환경에 액세스하면 향후 릴리스에서 스크립트 실행이 활성화됩니다.
- 크기 제한 - 총 zip 파일 크기는 6MB(모든 파일 포함)를 초과해서는 안 됩니다.
- SKILL.md 필수 - zip 파일에는 유효한 프론트미터가 있는 SKILL.md 파일이 포함되어야 합니다.

기술 이름 지정 모범 사례:

일반 이름이 아닌 "rds-throttling-investigation"과 같은 명확하고 설명이 포함된 이름을 사용합니다. 좋은 기술 이름은 처리하는 특정 시나리오 또는 서비스를 반영하므로 적절한 기술을 한눈에 쉽게 식별할 수 있습니다.

스킬 관리

AWS DevOps Agent는 운영자 웹 앱을 통해 포괄적인 기술 관리 기능을 제공합니다.

스킬 나열 - 에이전트 스페이스의 모든 스킬을 봅니다. 스킬 페이지에는 스킬 이름, 활성 또는 비활성 상태, 생성 날짜, 마지막 업데이트 날짜 및 사용 가능한 작업이 표시됩니다.

스킬 보기 - 스킬을 클릭하여 세부 정보 보기를 봅니다. UI에서 생성된 기술에는 UI에서 직접 이름, 설명 또는 지침을 수정하고 "저장"을 클릭하여 업데이트할 수 있는 편집 가능한 콘텐츠가 표시됩니다. zip 파일로 업로드된 기술에는 SKILL.md 및 참조/ 및 자산/과 같은 추가 디렉터리를 보여주는 파일 트리가 표시됩니다. 트리에서 파일을 클릭하여 내용을 읽기 전용 모드로 봅니다.

스킬에 대한 에이전트 선택 - 스킬을 생성하거나 편집할 때 각 스킬을 사용할 수 있는 에이전트 유형을 구성합니다. 에이전트 유형 드롭다운에서 일반(기본값 - 모든 에이전트 유형에 적용), 온디맨드(대화형 쿼리), 인시던트 분류(최초 인시던트 평가), 인시던트 RCA(근본 원인 분석), 인시던트 완화(자동 인시던트 대응) 또는 평가(사전 권장 사항) 확인란을 사용하여 하나 이상의 에이전트 유형을 선택합니다. 일반은 기본적으로 선택되며 모든 에이전트 유형에서 스킬을 사용할 수 있습니다. 특정 에이전트를 대상으로 하는 기술은 컨텍스트 소비를 줄이고 에이전트 포커스를 개선합니다.

스킬 활성화 및 비활성화 - 활성/비활성 토글을 사용하여 스킬을 삭제하지 않고 일시적으로 비활성화합니다. 스킬 세부 정보 보기를 열고 스위치를 "비활성"으로 전환하여 에이전트가 모든 콘텐츠와 구성을

유지하면서 새 조사를 위해 로드하지 못하도록 합니다. 진행 중인 조사는 스킬을 계속 사용합니다. 스킬을 즉시 다시 사용할 수 있도록 '활성'으로 다시 전환합니다.

기술 업데이트 - 기존 기술이 생성된 방식에 따라 수정합니다. UI에서 생성된 스킬의 경우 스킬 세부 정보 보기에서 "편집"을 클릭하고 이름, 설명 또는 지침을 수정한 다음 "저장"을 클릭하여 업데이트합니다. zip 파일로 업로드된 스킬의 경우 파일을 로컬에서 수정하고, 새 zip 파일을 생성하고, 새 버전을 업로드합니다.

스킬 삭제 - 에이전트 공간에서 스킬을 영구적으로 제거합니다. 스킬 목록 보기를 열고 추가 옵션 메뉴(")를 클릭한 다음 "삭제"를 선택하고 영구 삭제에 대한 경고를 검토하고 확인할 스킬 이름을 입력한 다음 "스킬 삭제"를 클릭합니다. 삭제는 실행 취소할 수 없습니다. 삭제된 스킬을 로드하려고 하면 진행 중인 조사가 영향을 받을 수 있습니다. zip 파일로 업로드된 스킬의 경우 백업으로 삭제하기 전에 zip 파일을 다운로드합니다. 다시 필요한 경우 스킬을 삭제하는 대신 비활성화하는 것이 좋습니다.

런북에서 마이그레이션

기존 런북은 고객 작업 없이 자동으로 Skills로 마이그레이션됩니다. 에이전트 스페이스가 Skills 모델로 전환되면 모든 런북이 Skills로 변환되고 Skills UI에 표시됩니다. 마이그레이션 후 다음을 수행할 수 있습니다.

- 마이그레이션된 기술 검토 - 자동 마이그레이션이 런북을 올바르게 변환했는지 확인합니다.
- 필요에 따라 업데이트 - UI에서 직접 기술을 편집하여 지침을 구체화하거나, 설명을 업데이트하거나, 에이전트 유형 타겟팅을 구성합니다.
- 참조로 확장 - 추가 참조 자료 또는 아키텍처 다이어그램의 이점을 누릴 수 있는 스킬의 경우 참조/ 또는 자산/ 디렉터리를 사용하여 zip 업로드 스킬로 다시 생성합니다.
- 새 스킬 생성 - 런북에서 이전에 다루지 않은 조사 워크플로를 위한 새 스킬을 추가합니다.

자동으로 마이그레이션된 기술에서 문제가 발생하거나 마이그레이션 후 업데이트에 대한 지원이 필요한 경우 AWS Support에 문의하세요.

학습한 기술

학습된 기술이란 무엇입니까?

학습된 기술은 DevOps 에이전트가 에이전트 스페이스 데이터에서 생성하는 구조화된 지식 파일입니다. 학습된 각 스킬은 AWS DevOps 에이전트가 작업을 수행할 때 사용하는 특정 유형의 지식을 인코

딩합니다. 시작 시 에이전트 공간 이해와 도구 사용 모범 사례라는 두 가지 학습된 기술을 사용할 수 있습니다.

에이전트 공간 이해

에이전트 공간 이해 스킬(`understanding-agent-space`)은 연결된 클라우드 계정, 코드 리포지토리 및 원격 측정 통합을 분석하여 에이전트 공간의 리소스 및 관계 맵을 구축합니다.

스킬은 기본 `SKILL.md` 파일과 참조 파일 세트를 생성합니다. 기본 파일에는 주요 도메인 개념이 포함된 일반 언어 시스템 개요, 배포 환경(AWS 계정 및 리전 페어, Azure 구독 및 리전 등), 논리적 서비스가 연결되는 방법을 보여주는 컨테이너 수준 아키텍처 다이어그램, 애플리케이션이 통과하는 구성 요소와 함께 애플리케이션에 중심이 되는 요청 경로, 컨테이너에 대한 코드 리포지토리 매핑이 포함되어 있습니다.

각 논리적 컨테이너는 ARNs, 테이블 이름 및 대기열 URLs과 같은 리소스 유형 및 물리적 식별자를 사용하여 내부 구성 요소(컴퓨팅, 데이터, 메시징, 네트워크 등)를 설명하는 전용 참조 파일을 받습니다. 또한 참조 파일은 각 구성 요소에 연결된 경보, 대시보드 및 모니터를 포함하여 관찰성 범위를 캡처합니다. 또한 각 구성 요소를 연결된 코드 리포지토리, 패키지 및 infrastructure-as-code 정의에 매핑하여 소스 코드에서 배포된 리소스까지 완전한 추적성 체인을 제공합니다.

각 중요 요청 경로는 진입점부터 각 중간 서비스, 데이터 스토어 및 외부 종속성까지 구성 요소 세부 수준에서 전체 end-to-end 요청 흐름을 설명하는 전용 참조 파일을 수신합니다. 파일에는 각 참가자의 책임과 함께 구성 요소 간의 작업 및 상호 작용 메커니즘 순서를 보여주는 순서가 지정된 흐름도가 포함되어 있습니다. 또한 경로와 관련된 관찰성 신호, 즉 각 홉에 대한 로그 그룹 패턴, 경보 이름 및 차원이 포함된 주요 지표(지연 시간, 오류율, 제한, 토큰 할당량), 서비스 및 계정 간에 상관관계가 있을 수 있는 분산 추적 범위를 분류합니다.

도구 사용 모범 사례

도구 사용 모범 사례 스킬은가 사용하는 과거 조사 도구를 분석하여 효과적인 사용 패턴, 일반적인 장애 모드 및 파라미터 지침을 추출합니다. 이렇게 하면 DevOps 에이전트가 알려진 위험을 방지하고 낭비되는 단계를 줄이면서 조사를 실행할 수 있습니다. 스킬은 기본 파일과 도구별 참조 파일 세트를 생성합니다. 기본 파일은 지원하는 조사 시나리오와 해당 참조 파일에 대한 링크가 포함된 각 도구를 나열하는 라우팅 인덱스 역할을 합니다.

각 도구별 참조 파일에는 최대 3개의 섹션이 포함될 수 있습니다.

- 모범 사례 - CloudWatch Logs Insights 쿼리 템플릿, 환경별 지표 네임스페이스 및 차원, CloudTrail 이벤트 소스 필터와 같이 성공적인 도구 사용에서 추출한 조사 기반 기법입니다. 각 항목은 조사 시나리오를 중심으로 구성되며 과거 조사에서 관찰된 구체적인 파라미터 값과 예제를 포함합니다.

- 일반적인 오류 - 반복 실패 모드 및 수정 사항입니다. 각 항목은 액세스할 수 없는 계정 쿼리 또는 잘못된 형식의 집계 쿼리 구성과 같은 특정 오류 조건을 설명하고 에이전트가 조사 단계를 낭비하지 않고 오류를 방지하거나 복구할 수 있도록 수정 조치를 제공합니다.
- 출력 관리 - 대규모 응답을 반환하는 경향이 있는 도구 호출에 대한 지침입니다. 각 항목은 진단 값을 유지하면서 출력 크기를 줄이는 파라미터 변경 또는 처리 전략을 설명합니다.

라이브 인프라 액세스를 사용할 수 있는 경우 스킬은 패턴을 포함하기 전에 환경에 대해 패턴을 검증합니다. 확인된 패턴은 자신 있게 언급되고, 확인되지 않은 패턴은 신중한 언어를 사용하며, 입증되지 않은 패턴은 제외됩니다. 이렇게 하면 기술이 인프라의 현재 상태에 맞게 조정됩니다.

학습된 기술 관리

업데이트 - DevOps 에이전트는 에이전트 스페이스의 활동을 기반으로 학습된 스킬을 자동으로 생성하고 업데이트합니다. 다음은 각 스킬이 업데이트되는 시기를 설명합니다.

DevOps 에이전트는 30건의 조사마다 업데이트된 도구 사용 모범 사례 스킬을 생성합니다.

에이전트 공간 이해 스킬은 에이전트 공간 기능 또는 통합을 추가, 업데이트 또는 제거할 때마다 실행되는 학습 에이전트에 의해 생성됩니다.

학습한 스킬을 수동으로 재생성하려면 운영자 앱의 토폴로지 페이지에서 재생성 버튼을 선택하거나 에이전트와 채팅하고 학습한 스킬을 업데이트하도록 요청합니다.

비활성화 - 학습한 스킬은 기본적으로 활성화됩니다. 활성화되면 DevOps 에이전트는 각 DevOps 에이전트 작업을 시작할 때 로드합니다. 학습된 스킬이 적용되지 않도록 하려면 연산자 앱의 스킬 뷰어에서 비활성화합니다. 스킬을 비활성화해도 스킬은 삭제되지 않습니다. 스킬은 유지되며 언제든지 다시 활성화할 수 있습니다. 스킬이 비활성화되면 DevOps 에이전트는 해당 스킬의 지식 없이 작동합니다.

토폴로지 보기 - 에이전트 스페이스 웹 앱의 토폴로지 페이지는 에이전트 스페이스 이해 스킬을 사용하여 에이전트 스페이스 환경을 논리적 컨테이너 및 구성 요소로 시각적으로 표시합니다. 컨테이너를 클릭하여 구성 요소, 리소스 식별자 및 원격 측정을 확인합니다.

지원되는 리전:

이 주제에서는 AWS DevOps 에이전트를 사용할 수 있는 AWS 리전에 대해 설명합니다. AWS 리전에 대한 자세한 내용은 계정 AWS 관리 참조 안내서의 계정에서 사용할 수 있는 AWS 리전 지정을 참조하세요.

리전 간 리소스 모니터링

AWS DevOps Agent는 에이전트 스페이스를 생성하는 지원되는 AWS 리전에 관계없이 모든 리전에 위치한 AWS 계정의 리소스를 모니터링하고 조사할 수 있습니다. AWS 계정을 에이전트 스페이스와 연결하면 에이전트는 해당 계정 내의 모든 리전에서 리소스를 검색하고 매핑합니다. 즉, 워크로드가 실행되는 모든 리전에 에이전트 공간이 필요하지 않습니다.

선호하는 데이터 레지던시, 운영 팀과의 근접성 또는 조직 요구 사항에 따라 지원되는 리전을 선택합니다.

지원되는 리전:

AWS DevOps 에이전트는 다음 AWS 리전에서 사용할 수 있습니다.

리전 이름	리전 코드	콘솔 링크
미국 동부(버지니아 북부)	us-east-1	콘솔 열기
미국 서부(오리건)	us-west-2	콘솔 열기
아시아 태평양(시드니)	ap-southeast-2	콘솔 열기
아시아 태평양(도쿄)	ap-northeast-1	콘솔 열기
유럽(프랑크푸르트)	eu-central-1	콘솔 열기
유럽(아일랜드)	eu-west-1	콘솔 열기

Service endpoints

리전 이름	리전 코드	엔드포인트	프로토콜
미국 동부(버지니아 북부)	us-east-1	aidevops.us-east-1.amazonaws.com	HTTPS
미국 서부(오레곤)	us-west-2	aidevops.us-west-2.amazonaws.com	HTTPS

리전 이름	리전 코드	엔드포인트	프로토콜
아시아 태평양(시드니)	ap-southeast-2	aidevops.ap-southeast-2.amazonaws.com	HTTPS
아시아 태평양(도쿄)	ap-northeast-1	aidevops.ap-northeast-1.amazonaws.com	HTTPS
유럽(프랑크푸르트)	eu-central-1	aidevops.eu-central-1.amazonaws.com	HTTPS
유럽(아일랜드)	eu-west-1	aidevops.eu-west-1.amazonaws.com	HTTPS

고려 사항

- 에이전트 스페이스 리전 선택 - 에이전트 스페이스 및 해당 데이터(조사,

토폴로지, 권장 사항)는 생성한 리전에 저장됩니다. 데이터 레지던시 요구 사항을 충족하는 리전을 선택합니다.

- 리전 간 모니터링 - 에이전트와 연결된 AWS 계정의 리소스

스페이스는 리소스가 배포되는 리전에 관계없이 모니터링됩니다. 워크로드가 실행되는 각 리전에서 별도의 에이전트 스페이스를 생성할 필요는 없습니다.

- 타사 통합 - CI/CD 공급자(GitHub, GitLab)에 대한 연결,

관찰성 도구(Dynatrace, Datadog, New Relic, Splunk) 및 MCP 서버는 에이전트 스페이스별로 구성되며 리전에 따라 달라지지 않습니다.

AWS DevOps 에이전트 시작하기

이 시작 안내서에서는 기본 에이전트 공간을 생성하고, 최소 권한을 구성하고, 첫 번째 AI 기반 조사를 수행합니다.

주제:

에이전트 스페이스 생성

에이전트 스페이스는 AWS DevOps 에이전트가 액세스할 수 있는 도구와 인프라를 정의합니다. 이 가이드에서는 에이전트 스페이스 생성, 기본 계정 액세스 구성, DevOps 에이전트 웹 앱 활성화를 안내합니다. 에이전트 공간 개념에 대한 자세한 내용은 “에이전트 공간이란 무엇입니까?”를 참조하세요.

에이전트 스페이스 생성

AWS DevOps 에이전트 콘솔에 액세스

1. AWS Management Console에 로그인
2. AWS DevOps 에이전트 콘솔로 이동

에이전트 공간 이름 지정

1. 에이전트 공간 생성을 클릭합니다.

에이전트 스페이스 세부 정보 섹션에서 다음을 제공합니다.

1. 이름 필드에 에이전트 공간의 이름을 입력합니다.
2. (선택 사항) 설명 필드에 에이전트 스페이스의 용도에 대한 세부 정보를 추가합니다.

3. (선택 사항) 에이전트 응답 언어 드롭다운에서 에이전트가 응답, 조사 결과 및 조사 출력을 생성할 때 사용하는 언어를 선택합니다. 옵션에는 인도네시아어, 중국어(간체/PRC), 중국어(번체/대만), 영어(영국), 프랑스어(프랑스), 독일어(독일), 이탈리아어(이탈리아), 일본어(일본), 한국어(한국), 포르투갈어(브라질), 스페인어(라틴 아메리카), 터키어(터키), 아랍어(사우디아라비아), 태국어(태국), 베트남어(베트남)가 포함됩니다. 언어를 선택하지 않으면 에이전트가 입력 언어로 응답합니다.

기본 계정 액세스 구성

이 에이전트 스페이스 AWS 리소스 액세스 권한 부여 섹션에서 에이전트 스페이스에 기본 AWS 계정에 대한 액세스 권한을 부여하는 IAM 역할을 설정합니다. 기본 계정은 에이전트 스페이스를 생성하는 AWS 계정입니다. AWS DevOps 에이전트는 조사 중에이 계정의 AWS 리소스를 검색하고 액세스하려면 IAM 역할이 필요합니다.

역할 구성 방법을 선택합니다. 다음 옵션 중 하나를 선택합니다.

옵션 1: new AWS DevOps 에이전트 역할 자동 생성(권장)

이 옵션은 AWS DevOps 에이전트가 계정의 리소스를 조사할 수 있는 적절한 권한이 있는 역할을 자동으로 생성합니다.

> **Note**
>
> 이 옵션을 사용하려면 새 역할을 생성하려면 IAM 권한이 있어야 합니다.

1. new AWS DevOps 에이전트 역할 자동 생성을 선택합니다.
2. (선택 사항) 생성할 에이전트 스페이스 역할 이름 업데이트

옵션 2: 기존 역할 할당

다른 관리자가 이전에 AWS DevOps 에이전트 전용 역할을 생성한 경우이 옵션을 사용합니다.

1. 기존 역할 할당을 선택합니다.
2. 드롭다운 메뉴에서 적절한 권한이 있는 기존 역할을 선택합니다.

옵션 3: 정책 템플릿을 사용하여 new AWS DevOps 에이전트 역할 생성

에이전트가 기본 계정에서 액세스할 수 있는 서비스와 리소스를 제한해야 하는 경우이 옵션을 사용합니다.

1. 정책 템플릿을 사용하여 new AWS DevOps 에이전트 역할 생성을 선택합니다.
2. 지침에 따라 새 역할의 신뢰 정책 및 인라인 정책을 생성합니다.

에이전트 스페이스 웹 앱 활성화

웹 앱은 직원이 인시던트 조사 및 권장 사항 검토를 위해 AWS DevOps 에이전트와 상호 작용하는 곳입니다. 자세한 내용은 AWS DevOps 에이전트 콘솔 아키텍처[링크]를 참조하세요. 활성화하면 사용자는 AWS 관리 콘솔의 IAM 인증 링크를 통해 에이전트 스페이스 웹 앱에 액세스할 수 있습니다.

다음 옵션 중 하나를 선택합니다.

옵션 1: new AWS DevOps 에이전트 역할 자동 생성(권장)

이 옵션은 DevOps 에이전트 웹 앱에 액세스할 수 있는 적절한 권한이 있는 역할을 자동으로 생성합니다.

> **Note**
>
> 이 옵션을 사용하려면 새 역할을 생성하려면 IAM 권한이 있어야 합니다.

1. new AWS DevOps 에이전트 역할 자동 생성을 선택합니다.
2. 역할에 부여될 권한 검토

옵션 2: 기존 역할 할당

다른 관리자가 이전에 연산자 역할을 생성한 경우이 옵션을 사용합니다.

1. 기존 역할 할당을 선택합니다.
2. 드롭다운 메뉴에서 적절한 권한이 있는 기존 역할을 선택합니다.

옵션 3: 정책 템플릿을 사용하여 new AWS DevOps 에이전트 역할 생성

웹 앱 액세스 권한을 사용자 지정해야 하는 경우이 옵션을 사용합니다.

1. 정책 템플릿을 사용하여 new AWS DevOps 에이전트 역할 생성을 선택합니다.
2. 지침에 따라 새 역할의 신뢰 정책 및 인라인 정책을 생성합니다.

태그 추가(선택 사항)

생성 중에 에이전트 스페이스에 AWS 태그를 추가할 수 있습니다. 태그는 리소스를 구성하고 식별하는 데 도움이 되는 키-값 페어입니다. 에이전트 공간당 최대 50개의 태그를 추가할 수 있습니다. 태그를 추가하려면 에이전트 공간 생성 페이지에서 태그 섹션을 확장하고 새 태그 추가를 클릭합니다.

에이전트 공간 생성 완료

모든 섹션이 작성되면 생성을 클릭합니다.

에이전트 스페이스 설정 확인

구성되면 에이전트 스페이스 세부 정보 페이지에 운영자 액세스 버튼이 나타납니다. 이를 클릭하면 새 탭에서 웹 앱이 열리고 성공적으로 인증됩니다.

다음 단계

에이전트 스페이스를 설정한 후 다음 단계를 고려하세요.

- 애플리케이션이 여러 계정에 걸쳐 있는 경우 보조 AWS 계정 추가
- 관찰성 도구 또는 티켓팅 시스템과 같은 타사 통합 구성
- 프로덕션 환경에 대한 AWS Identity Center 인증 설정
- 애플리케이션 리소스 매핑을 탐색하여 AWS DevOps 에이전트가 인프라를 이해하는 데 도움이 됩니다.

AWS DevOps Agent CLI 온보딩 가이드

개요

AWS DevOps 에이전트를 사용하면 AWS 인프라를 모니터링하고 관리할 수 있습니다. 이 가이드에서는 AWS 명령줄 인터페이스(AWS CLI)를 사용하여 up AWS DevOps 에이전트를 설정하는 방법을 안내합니다. IAM 역할을 생성하고, 에이전트 공간을 설정하고, AWS 계정을 연결합니다. 또한 운영자 앱을 활성화하고 선택적으로 타사 통합을 연결합니다. 이 안내서를 완료하는 데 약 20분이 걸립니다.

AWS DevOps 에이전트는 미국 동부(버지니아 북부), 미국 서부(오레곤), 아시아 태평양(시드니), 아시아 태평양(도쿄), 유럽(프랑크푸르트) 및 유럽(아일랜드)의 6개 AWS 리전에서 사용할 수 있습니다. 지원되는 리전에 대한 자세한 내용은 섹션을 참조하세요the section called “지원되는 리전:”.

사전 조건

시작하기 전에 다음 사항을 갖췄는지 확인하세요.

- AWS CLI 버전 2 설치 및 구성
- AWS 모니터링 계정에 대한 인증
- AWS Identity and Access Management(IAM) 역할을 생성하고 정책을 연결할 수 있는 권한
- 모니터링 AWS 계정으로 사용할 계정
- AWS CLI 및 JSON 구문에 대한 지식

이 가이드 전체에서 다음 자리 표시자 값을 사용자의 값으로 바꿉니다.

- `<MONITORING_ACCOUNT_ID>` - 모니터링(기본) AWS 계정의 12자리 계정 ID
- `<EXTERNAL_ACCOUNT_ID>` - 모니터링할 보조 AWS 계정의 12자리 계정 ID(4단계에서 사용됨)
- `<REGION>` - 에이전트 공간의 AWS 리전 코드(예: `us-east-1` 또는 `eu-central-1`)
- `<AGENT_SPACE_ID>` - `create-agent-space` 명령에서 반환되는 에이전트 공간 식별자입니다.

IAM 역할 설정

1. DevOps 에이전트 스페이스 역할 생성

다음 명령을 실행하여 IAM 신뢰 정책을 생성합니다.

```
cat > devops-agentspace-trust-policy.json << 'EOF'
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "aidevops.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "<MONITORING_ACCOUNT_ID>"
        },
        "ArnLike": {
```

```
          "aws:SourceArn":
 "arn:aws:aidevops:<REGION>:<MONITORING_ACCOUNT_ID>:agentspace/*"
        }
      }
    }
  ]
}
EOF
```

IAM 역할을 생성합니다.

```
aws iam create-role \
  --region <REGION> \
  --role-name DevOpsAgentRole-AgentSpace \
  --assume-role-policy-document file://devops-agentspace-trust-policy.json
```

다음 명령을 실행하여 역할 ARN을 저장합니다.

```
aws iam get-role --role-name DevOpsAgentRole-AgentSpace --query 'Role.Arn' --output
 text
```

AWS 관리형 정책을 연결합니다.

```
aws iam attach-role-policy \
  --role-name DevOpsAgentRole-AgentSpace \
  --policy-arn arn:aws:iam::aws:policy/AIDevOpsAgentAccessPolicy
```

Resource Explorer 서비스 연결 역할 생성을 허용하는 인라인 정책을 생성하고 연결합니다.

```
cat > devops-agentspace-additional-policy.json << 'EOF'
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCreateServiceLinkedRoles",
      "Effect": "Allow",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": [
        "arn:aws:iam::<MONITORING_ACCOUNT_ID>:role/aws-service-role/resource-
explorer-2.amazonaws.com/AWSServiceRoleForResourceExplorer"
```

```
      ]
    }
  ]
}
EOF

aws iam put-role-policy \
  --role-name DevOpsAgentRole-AgentSpace \
  --policy-name AllowCreateServiceLinkedRoles \
  --policy-document file://devops-agentspace-additional-policy.json
```

2. 운영자 앱 IAM 역할 생성

다음 명령을 실행하여 IAM 신뢰 정책을 생성합니다.

```
cat > devops-operator-trust-policy.json << 'EOF'
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "aidevops.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRole",
        "sts:TagSession"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "<MONITORING_ACCOUNT_ID>"
        },
        "ArnLike": {
          "aws:SourceArn":
 "arn:aws:aidevops:<REGION>:<MONITORING_ACCOUNT_ID>:agentspace/*"
        }
      }
    }
  ]
}
EOF
```

IAM 역할을 생성합니다.

```
aws iam create-role \
  --role-name DevOpsAgentRole-WebappAdmin \
  --assume-role-policy-document file://devops-operator-trust-policy.json \
  --region <REGION>
```

다음 명령을 실행하여 역할 ARN을 저장합니다.

```
aws iam get-role --role-name DevOpsAgentRole-WebappAdmin --query 'Role.Arn' --output
 text
```

AWS 관리형 운영자 앱 정책을 연결합니다.

```
aws iam attach-role-policy \
  --role-name DevOpsAgentRole-WebappAdmin \
  --policy-arn arn:aws:iam::aws:policy/AIDevOpsOperatorAppAccessPolicy
```

이 관리형 정책은 운영자 앱에 에이전트 공간 기능에 액세스할 수 있는 권한을 부여합니다. 이러한 기능에는 조사, 권장 사항, 지식 관리, 채팅 및 AWS 지원 통합이 포함됩니다. 이 정책은 `aws:PrincipalTag/AgentSpaceId` 조건을 사용하여 특정 에이전트 공간에 대한 액세스 범위를 지정합니다. 전체 작업 목록에 대한 자세한 내용은 섹션을 참조하세요the section called “DevOps 에이전트 IAM 권한”.

온보딩 단계

1. 에이전트 공간 생성

다음 명령을 실행하여 에이전트 공간을 생성합니다.

```
aws devops-agent create-agent-space \
  --name "MyAgentSpace" \
  --description "AgentSpace for monitoring my application" \
  --region <REGION>
```

선택적으로 암호화`--kms-key-arn`에 고객 관리형 AWS KMS 키를 사용하도록를 지정합니다. `--tags`를 사용하여 리소스 태그를 추가하고 에이전트 응답 `--locale` 언어를 설정할 수도 있습니다.

응답`agentSpaceId`에서를 저장합니다(에 위치`agentSpace.agentSpaceId`).

나중에 에이전트 공간을 나열하려면 다음 명령을 실행합니다.

```
aws devops-agent list-agent-spaces \
```

```
  --region <REGION>
```

2. AWS 계정 연결

AWS 계정을 연결하여 토폴로지 검색을 켭니다. accountType를 다음 값 중 하나로 설정합니다.

- monitor - 에이전트 공간이 있는 기본 계정입니다. 이 계정은 에이전트를 호스팅하며 토폴로지 검색에 사용됩니다.
- source - 에이전트가 모니터링하는 추가 계정입니다. 4단계에서 외부 계정을 연결할 때이 유형을 사용합니다.

```
aws devops-agent associate-service \
  --agent-space-id <AGENT_SPACE_ID> \
  --service-id aws \
  --configuration '{
    "aws": {
      "assumableRoleArn": "arn:aws:iam::<MONITORING_ACCOUNT_ID>:role/DevOpsAgentRole-
AgentSpace",
      "accountId": "<MONITORING_ACCOUNT_ID>",
      "accountType": "monitor"
    }
  }' \
  --region <REGION>
```

3. 연산자 앱 활성화

인증 흐름은 IAM, IAM Identity Center(IDC) 또는 외부 ID 제공업체(IdP)를 사용할 수 있습니다. 다음 명령을 실행하여 에이전트 공간에 운영자 앱을 활성화합니다.

```
aws devops-agent enable-operator-app \
  --agent-space-id <AGENT_SPACE_ID> \
  --auth-flow iam \
  --operator-app-role-arn "arn:aws:iam::<MONITORING_ACCOUNT_ID>:role/DevOpsAgentRole-
WebappAdmin" \
  --region <REGION>
```

IAM Identity Center 인증의 경우 --auth-flow idc를 사용하고를 제공합니다--idc-instance-arn. 외부 자격 증명 공급자의 경우를 --auth-flow idp 사용하고 --issuer-url, 및 --idp-client-id를 제공합니다--idp-client-secret. 자세한 내용은 the section called “IAM Identity Center 인증 설정” 및 the section called “외부 ID 제공업체(IdP) 인증 설정” 섹션을 참조하세요.

참고: 이전에 계정의 다른 에이전트 공간에 대한 운영자 앱 역할을 생성한 경우 해당 역할 ARN을 재사용할 수 있습니다.

4. (선택 사항) 추가 소스 계정 연결

AWS DevOps Agent를 사용하여 추가 계정을 모니터링하려면 IAM 교차 계정 역할을 생성합니다.

외부 계정에서 교차 계정 역할 생성

외부 계정으로 전환하고 신뢰 정책을 생성합니다. MONITORING_ACCOUNT_ID는 2단계에서 설정한 에이전트 공간을 호스팅하는 기본 계정입니다. 이 구성을 사용하면 AWS DevOps 에이전트 서비스가 모니터링 계정을 대신하여 보조 소스 계정에서 역할을 수임할 수 있습니다.

다음 명령을 실행하여 신뢰 정책을 생성합니다.

```
cat > devops-cross-account-trust-policy.json << 'EOF'
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "aidevops.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "<MONITORING_ACCOUNT_ID>",
          "sts:ExternalId":
 "arn:aws:aidevops:<REGION>:<MONITORING_ACCOUNT_ID>:agentspace/<AGENT_SPACE_ID>"
        }
      }
    }
  ]
}
EOF
```

교차 계정 IAM 역할을 생성합니다.

```
aws iam create-role \
  --role-name DevOpsAgentCrossAccountRole \
  --assume-role-policy-document file://devops-cross-account-trust-policy.json
```

다음 명령을 실행하여 역할 ARN을 저장합니다.

```
aws iam get-role --role-name DevOpsAgentCrossAccountRole --query 'Role.Arn' --output
 text
```

AWS 관리형 정책을 연결합니다.

```
aws iam attach-role-policy \
  --role-name DevOpsAgentCrossAccountRole \
  --policy-arn arn:aws:iam::aws:policy/AIDevOpsAgentAccessPolicy
```

외부 계정에서 Resource Explorer 서비스 연결 역할을 생성할 수 있도록 인라인 정책을 연결합니다.

```
cat > devops-cross-account-additional-policy.json << 'EOF'
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCreateServiceLinkedRoles",
      "Effect": "Allow",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": [
        "arn:aws:iam::<EXTERNAL_ACCOUNT_ID>:role/aws-service-role/resource-
explorer-2.amazonaws.com/AWSServiceRoleForResourceExplorer"
      ]
    }
  ]
}
EOF

aws iam put-role-policy \
  --role-name DevOpsAgentCrossAccountRole \
  --policy-name AllowCreateServiceLinkedRoles \
  --policy-document file://devops-cross-account-additional-policy.json
```

외부 계정 연결

모니터링 계정으로 다시 전환한 다음 다음 명령을 실행하여 외부 계정을 연결합니다.

```
aws devops-agent associate-service \
```

```
  --agent-space-id <AGENT_SPACE_ID> \
  --service-id aws \
  --configuration '{
    "sourceAws": {
      "accountId": "<EXTERNAL_ACCOUNT_ID>",
      "accountType": "source",
      "assumableRoleArn": "arn:aws:iam::<EXTERNAL_ACCOUNT_ID>:role/
DevOpsAgentCrossAccountRole"
    }
  }' \
  --region <REGION>
```

5. (선택 사항) GitHub 연결

> 참고: CLI를 통해 연결하려면 먼저 OAuth 흐름을 사용하여 AWS DevOps 에이전트 콘솔을 통해 GitHub를 등록해야 합니다.

콘솔을 통해 GitHub를 등록하는 방법에 대한 지침은 섹션을 참조하세요the section called “CI/CD 파이프라인에 연결”.

등록된 서비스를 나열합니다.

```
aws devops-agent list-services \
  --region <REGION>
```

serviceType:에 <SERVICE_ID> 대한를 저장합니다github.

콘솔에 GitHub를 등록한 후 다음 명령을 실행하여 GitHub 리포지토리를 연결합니다.

```
aws devops-agent associate-service \
  --agent-space-id <AGENT_SPACE_ID> \
  --service-id <SERVICE_ID> \
  --configuration '{
    "github": {
      "repoName": "<GITHUB_REPO_NAME>",
      "repoId": "<GITHUB_REPO_ID>",
      "owner": "<GITHUB_OWNER>",
      "ownerType": "organization"
    }
  }' \
  --region <REGION>
```

6. (선택 사항) ServiceNow 등록 및 연결

먼저 ServiceNow 서비스를 OAuth 자격 증명으로 등록합니다.

```
aws devops-agent register-service \
  --service servicenow \
  --service-details  '{
    "servicenow": {
      "instanceUrl": "<SERVICENOW_INSTANCE_URL>",
      "authorizationConfig": {
        "oAuthClientCredentials": {
            "clientName": "<SERVICENOW_CLIENT_NAME>",
            "clientId": "<SERVICENOW_CLIENT_ID>",
            "clientSecret": "<SERVICENOW_CLIENT_SECRET>"
        }
      }
    }
  }' \
  --region <REGION>
```

반환된를 저장<SERVICE_ID>한 다음 ServiceNow를 연결합니다.

```
aws devops-agent associate-service \
  --agent-space-id <AGENT_SPACE_ID> \
  --service-id <SERVICE_ID> \
  --configuration '{
    "servicenow": {
      "instanceUrl": "<SERVICENOW_INSTANCE_URL>"
    }
  }' \
  --region <REGION>
```

7. (선택 사항) Dynatrace 등록 및 연결

먼저 Dynatrace 서비스를 OAuth 자격 증명으로 등록합니다.

```
aws devops-agent register-service \
  --service dynatrace \
  --service-details '{
  "dynatrace": {
    "accountUrn": "<DYNATRACE_ACCOUNT_URN>",
    "authorizationConfig": {
```

```
        "oAuthClientCredentials": {
            "clientName": "<DYNATRACE_CLIENT_NAME>",
            "clientId": "<DYNATRACE_CLIENT_ID>",
            "clientSecret": "<DYNATRACE_CLIENT_SECRET>"
        }
      }
    }
  }' \
  --region <REGION>
```

반환된를 저장<SERVICE_ID>한 다음 Dynatrace를 연결합니다. 리소스는 선택 사항입니다. 환경은 연결할 Dynatrace 환경을 지정합니다.

```
aws devops-agent associate-service \
  --agent-space-id <AGENT_SPACE_ID> \
  --service-id <SERVICE_ID> \
  --configuration '{
    "dynatrace": {
      "envId": "<DYNATRACE_ENVIRONMENT_ID>",
      "resources": [
        "<DYNATRACE_RESOURCE_1>",
        "<DYNATRACE_RESOURCE_2>"
      ]
    }
  }' \
  --region <REGION>
```

응답에는 통합을 위한 웹후크 정보가 포함됩니다. 이 웹후크를 사용하여 Dynatrace에서 조사를 트리거할 수 있습니다. 자세한 내용은 the section called “Dynatrace 연결” 단원을 참조하십시오.

8. (선택 사항) Splunk 등록 및 연결

먼저 BearerToken 자격 증명으로 Splunk 서비스를 등록합니다.

엔드포인트는 다음 형식을 사용합니다. `https://<XXX>.api.scs.splunk.com/<XXX>/mcp/v1/`

```
aws devops-agent register-service \
  --service mcpserversplunk \
  --service-details '{
  "mcpserversplunk": {
    "name": "<SPLUNK_NAME>",
    "endpoint": "<SPLUNK_ENDPOINT>",
    "authorizationConfig": {
```

```
        "bearerToken": {
            "tokenName": "<SPLUNK_TOKEN_NAME>",
            "tokenValue": "<SPLUNK_TOKEN_VALUE>"
        }
      }
    }
  }' \
  --region <REGION>
```

반환된를 저장<SERVICE_ID>한 다음 Splunk를 연결합니다.

```
aws devops-agent associate-service \
  --agent-space-id <AGENT_SPACE_ID> \
  --service-id <SERVICE_ID> \
  --configuration '{
    "mcpserversplunk":  {
      "name": "<SPLUNK_NAME>",
      "endpoint": "<SPLUNK_ENDPOINT>"
    }
  }' \
  --region <REGION>
```

응답에는 통합을 위한 웹후크 정보가 포함됩니다. 이 웹후크를 사용하여 Splunk에서 조사를 트리거할 수 있습니다. 자세한 내용은 the section called “Splunk 연결” 단원을 참조하십시오.

9. (선택 사항) New Relic 등록 및 연결

먼저 New Relic 서비스를 API 키 자격 증명으로 등록합니다.

리전: US 또는 EU.

선택적 필드: `applicationIds`, `entityGuids`, `alertPolicyIds`

```
aws devops-agent register-service \
  --service mcpservernewrelic \
  --service-details '{
    "mcpservernewrelic": {
      "authorizationConfig": {
        "apiKey": {
          "apiKey": "<YOUR_NEW_RELIC_API_KEY>",
          "accountId": "<YOUR_ACCOUNT_ID>",
          "region": "US",
          "applicationIds": ["<APP_ID_1>", "<APP_ID_2>"],
```

```
          "entityGuids": ["<ENTITY_GUID_1>"],
          "alertPolicyIds": ["<POLICY_ID_1>"]
        }
      }
    }
  }' \
  --region <REGION>
```

반환된를 저장<SERVICE_ID>한 다음 New Relic을 연결합니다.

```
aws devops-agent associate-service \
  --agent-space-id <AGENT_SPACE_ID> \
  --service-id <SERVICE_ID> \
  --configuration '{
    "mcpservernewrelic":  {
      "accountId": "<YOUR_ACCOUNT_ID>",
      "endpoint": "https://mcp.newrelic.com/mcp/"
    }
  }' \
  --region <REGION>
```

응답에는 통합을 위한 웹후크 정보가 포함됩니다. 이 웹후크를 사용하여 New Relic에서 조사를 트리거할 수 있습니다. 자세한 내용은 the section called “새 복제본 연결” 단원을 참조하십시오.

10. (선택 사항) Datadog 등록 및 연결

CLI를 통해 연결하려면 먼저 OAuth 흐름을 사용하여 AWS DevOps 에이전트 콘솔을 통해 Datadog을 등록해야 합니다. 자세한 내용은 the section called “DataDog 연결” 단원을 참조하십시오.

등록된 서비스를 나열합니다.

```
aws devops-agent list-services \
  --region <REGION>
```

serviceType:에 <SERVICE_ID> 대한를 저장합니다mcpserverdatadog.

그런 다음 Datadog을 연결합니다.

```
aws devops-agent associate-service \
  --agent-space-id <AGENT_SPACE_ID> \
  --service-id <SERVICE_ID> \
  --configuration '{
```

```
    "mcpserverdatadog": {
      "name": "Datadog-MCP-Server",
      "endpoint": "<DATADOG_MCP_ENDPOINT>"
    }
  }' \
  --region <REGION>
```

응답에는 통합을 위한 웹후크 정보가 포함됩니다. 이 웹후크를 사용하여 Datadog에서 조사를 트리거할 수 있습니다. 자세한 내용은 the section called “DataDog 연결” 단원을 참조하십시오.

11. (선택 사항) 에이전트 공간 삭제

에이전트 공간을 삭제하면 해당 에이전트 공간에 대한 모든 연결, 구성 및 조사 데이터가 제거됩니다. 이 작업은 실행 취소할 수 없습니다.

에이전트 공간을 삭제하려면 다음 명령을 실행합니다.

```
aws devops-agent delete-agent-space \
  --agent-space-id <AGENT_SPACE_ID> \
  --region <REGION>
```

Verification(확인)

설정을 확인하려면 다음 명령을 실행합니다.

```
# List your agent spaces
aws devops-agent list-agent-spaces \
  --region <REGION>

# Get details of a specific agent space
aws devops-agent get-agent-space \
  --agent-space-id <AGENT_SPACE_ID> \
  --region <REGION>

# List associations for an agent space
aws devops-agent list-associations \
  --agent-space-id <AGENT_SPACE_ID> \
  --region <REGION>
```

다음 단계

- 추가 통합을 연결하려면 섹션을 참조하세요AWS DevOps Agent에 대한 기능 구성.

- 에이전트 기술 및 기능에 대한 자세한 내용은 섹션을 참조하세요the section called “DevOps 에이전트 기술”.
- 연산자 웹 앱을 이해하려면 섹션을 참조하세요the section called “DevOps 에이전트 웹 앱이란 무엇입니까?”.

참고

- <AGENT_SPACE_ID>, <MONITORING_ACCOUNT_ID>, <EXTERNAL_ACCOUNT_ID><REGION>, 등을 실제 값으로 바꿉니다.
- 지원되는 리전 목록은 the section called “지원되는 리전:” 섹션을 참조하세요.

테스트 환경 생성

이 가이드에서는 샘플 아키텍처를 사용하여 AWS DevOps 에이전트의 인시던트 대응 기능을 검증하기 위한 실습 테스트를 제공합니다. 프로덕션 시스템을 연결하기 전에 DevOps 에이전트를 테스트하려면 이 추가 기능을 사용합니다.

사전 조건

- AWS 관리 액세스 권한이 있는 계정
- AWS DevOps 에이전트 역할 자동 생성 흐름을 사용하여 생성 및 구성된 DevOps 에이전트 공간

비용 및 안전 개요

비용 보호

- EC2 테스트: 2시간 동안 무료(AWS 프리 티어) 또는 ~$0.02
- Lambda 테스트: 무료(1백만 요청/월 프리 티어)
- CloudWatch: 무료(경보 10개, 기본 지표 포함)
- 예상 총 비용: 전체 테스트의 경우 0.00~0.05 USD

이러한 테스트의 안전 기능

- 자동 종료: 기본 제공 자동 종료
- 프리 티어 적격: 가장 작은 인스턴스 유형 사용

- 제한된 범위: 최소한의 격리된 테스트 리소스
- 간편한 정리: 모든 것을 제거하는 간단한 콘솔 단계
- 프로덕션 영향 없음: 완전히 분리된 테스트 환경

테스트용 AWS 계정 설정

> **Important**
>
> 인프라 리소스는 DevOps Agent Space의 기본 클라우드 AWS 계정을 생성한 계정에 배포해야 합니다. 특정 리전은 중요하지 않습니다.

1. AWS 콘솔에 로그인: https://console.aws.amazon.com
2. DevOps 에이전트 스페이스가 위치한 동일한 AWS 계정에서 작업하고 있는지 확인합니다.
3. 테스트 리소스에 모든 리전을 사용할 수 있습니다.

> **Note**
>
> DevOps 에이전트의 기본 계정과 생성 중인 테스트 환경 리소스 간의 1:1 매핑은 테스트 설정을 간소화합니다. DevOps 에이전트 공간을 쉽게 확장하여 보조 계정을 포함하고 교차 계정 조사를 활성화할 수 있습니다.

테스트 선택

테스트를 독립적으로 실행하거나 둘 다 함께 실행할 수 있습니다.

테스트 옵션 A: EC2 CPU 용량 테스트

용도: EC2 성능 문제를 감지하고 조사하는 Validate AWS DevOps 에이전트의 기능

예상 시간: 5분 설정 + 10분 자동 실행

난이도: 완전 자동화(수동 단계가 필요하지 않음)

테스트 옵션 B: Lambda 오류율 테스트

용도: Lambda 함수 오류를 감지하고 조사하는 Validate AWS DevOps 에이전트의 기능

예상 시간: 설정 10분 + 트리거 2분

난이도: 매우 쉬움

테스트 옵션 A: EC2 CPU 용량 테스트

1단계: EC2 테스트를 위한 CloudFormation 스택 배포

CloudFormation을 사용하여 테스트 리소스를 생성하여 AWS DevOps 에이전트가 리소스를 올바르게 추적하고 조사할 수 있도록 합니다.

1. CloudFormation으로 이동합니다.
   a. AWS 콘솔에서 "CloudFormation"을 검색하고 CloudFormation을 클릭합니다.
   b. 스택 생성 > 새 리소스 사용(표준)을 클릭합니다.
2. 템플릿 업로드:
   a. 라는 새 로컬 파일 생성`AWS-DevOpsAgent-ec2-test.yaml`
   b. 이 CloudFormation 템플릿을 복사하여 파일에 붙여 넣습니다.
      i.
```
AWSTemplateFormatVersion: '2010-09-09'
Description: 'AWS DevOps Agent EC2 CPU Test Stack'
Parameters:
  MyIP:
    Type: String
    Description: Your current IP address for SSH access (find at https://
whatismyipaddress.com)
    Default: '0.0.0.0/0'
Resources:
  # Security Group for SSH access
  TestSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupName: AWS-DevOpsAgent-test-sg
      GroupDescription: AWS DevOps Agent beta testing security group
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 22
          ToPort: 22
          CidrIp: !Ref MyIP
          Description: SSH access from your IP
      Tags:
        - Key: Name
          Value: AWS-DevOpsAgent-Test-SG
```

```
        - Key: Purpose
          Value: AWS-DevOpsAgent-Testing
  # Key Pair for SSH access
  TestKeyPair:
    Type: AWS::EC2::KeyPair
    Properties:
      KeyName: AWS-DevOpsAgent-test-key
      KeyType: rsa
      Tags:
        - Key: Name
          Value: AWS-DevOpsAgent-Test-Key
        - Key: Purpose
          Value: AWS-DevOpsAgent-Testing
  # EC2 Instance for CPU testing
  TestInstance:
    Type: AWS::EC2::Instance
    Properties:
      InstanceType: t3.micro
      ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-
kernel-6.1-x86_64}}'
      KeyName: !Ref TestKeyPair
      SecurityGroupIds:
        - !Ref TestSecurityGroup
      UserData:
        Fn::Base64: !Sub |
          #!/bin/bash
          yum update -y
          yum install -y htop

          # Create the CPU stress test script
          cat > /home/ec2-user/cpu-stress-test.sh << 'EOF'
          #!/bin/bash
          echo "Starting AWS DevOpsAgent CPU Stress Test"
          echo "Time: $(date)"
          echo "Instance: $(curl -s http://169.254.169.254/latest/meta-data/
instance-id)"
          echo ""

          # Get number of CPU cores
          CORES=$(nproc)
          echo "CPU Cores: $CORES"
          echo ""

          echo "Starting stress test (5 minutes)..."
```

```
          echo "This will generate >70% CPU usage to trigger CloudWatch alarm"
          echo ""

          # Create CPU load using yes command
          echo "Starting CPU load processes..."
          for i in $(seq 1 $CORES); do
              (yes > /dev/null) &
              CPU_PID=$!
              echo "Started CPU load process $i (PID: $CPU_PID)"
              echo $CPU_PID >> /tmp/cpu_test_pids
          done

          # Auto-cleanup after 5 minutes
          (sleep 300 && echo "Stopping CPU load processes..." && kill $(cat /
tmp/cpu_test_pids 2>/dev/null) 2>/dev/null && rm -f /tmp/cpu_test_pids) &

          echo ""
          echo "CPU load processes started for 5 minutes"
          echo "Check CloudWatch for alarm trigger in 3-5 minutes"
          EOF

          chmod +x /home/ec2-user/cpu-stress-test.sh
          chown ec2-user:ec2-user /home/ec2-user/cpu-stress-test.sh

          # Create auto-shutdown script (safety mechanism)
          cat > /home/ec2-user/auto-shutdown.sh << 'SHUTDOWN_EOF'
          #!/bin/bash
          echo "Auto-shutdown scheduled for 2 hours from now: $(date)"
          sleep 7200
          echo "Auto-shutdown executing at: $(date)"
          sudo shutdown -h now
          SHUTDOWN_EOF

          chmod +x /home/ec2-user/auto-shutdown.sh
          nohup /home/ec2-user/auto-shutdown.sh > /home/ec2-user/auto-
shutdown.log 2>&1 &

          echo "AWS DevOpsAgent test setup completed at $(date)" > /home/ec2-
user/setup-complete.txt
      Tags:
        - Key: Name
          Value: AWS-DevOpsAgent-Test-Instance
        - Key: Purpose
          Value: AWS-DevOpsAgent-Testing
```

```
  # CloudWatch Alarm for CPU utilization
  CPUAlarm:
    Type: AWS::CloudWatch::Alarm
    Properties:
      AlarmName: AWS-DevOpsAgent-EC2-CPU-Test
      AlarmDescription: AWS-DevOpsAgent beta test - EC2 CPU utilization alarm
      MetricName: CPUUtilization
      Namespace: AWS/EC2
      Statistic: Average
      Period: 60
      EvaluationPeriods: 1
      Threshold: 70
      ComparisonOperator: GreaterThanThreshold
      Dimensions:
        - Name: InstanceId
          Value: !Ref TestInstance
      TreatMissingData: notBreaching
Outputs:
  InstanceId:
    Description: EC2 Instance ID for testing
    Value: !Ref TestInstance

  SecurityGroupId:
    Description: Security Group ID
    Value: !Ref TestSecurityGroup

  AlarmName:
    Description: CloudWatch Alarm Name
    Value: !Ref CPUAlarm

  SSHCommand:
    Description: SSH command to connect to instance
    Value: !Sub 'ssh -i "AWS-DevOpsAgent-test-key.pem" ec2-user@
${TestInstance.PublicDnsName}'
```

c. CloudFormation 콘솔에서 템플릿 파일 업로드를 선택합니다.

d. 파일 선택을 클릭합니다.

e. AWS-DevOpsAgent-ec2-test.yaml 파일 선택

f. 다음을 클릭합니다.

3. 스택 구성:

a. 스택 이름:AWS-DevOpsAgent-EC2-Test

b. 파라미터:

i. MyIP: 기본값으로 둡니다`0.0.0.0/0`(필요한 경우 나중에 보호할 수 있음).

c. 다음을 클릭합니다.

4. 스택 옵션 구성:

a. 기본값을 그대로 두고 다음을 클릭합니다.

5. 검토 및 생성

a. AWS CloudFormation에서 IAM 리소스를 생성할 수 있음을 확인합니다.

b. 제출을 클릭합니다.

6. 완료될 때까지 기다립니다.

a. 스택 생성에는 3~5분이 소요됩니다.

b. 상태가에서 `CREATE_IN_PROGRESS`로 변경됩니다.`CREATE_COMPLETE`

c. 중요: 이제 EC2 인스턴스가 AWS DevOpsAgent가 추적할 수 있는 CloudFormation 스택의 일부입니다!

선택 사항: SSH 액세스 보호(인스턴스에 연결하려는 경우에만 해당)

자동 테스트를 실행하려는 경우이 단계를 건너뜁니다.

1. EC2 보안 그룹으로 이동합니다.

a. AWS 콘솔에서 EC2 → 보안 그룹으로 이동합니다.

b. 찾기`AWS-DevOpsAgent-test-sg`

2. SSH 규칙 업데이트:

a. 보안 그룹 선택 → 인바운드 규칙 탭 → 인바운드 규칙 편집

b. SSH 규칙 찾기(포트 22)

c. 소스를에서 IP`0.0.0.0/0`로 변경합니다.`[YOUR_IP]/32`

d. https://whatismyipaddress.com IP 가져오기

e. 규칙 저장을 클릭합니다.

2단계: 자동 테스트 실행 대기

1. 자동 테스트 실행:

- ~~CPU 스트레스 테스트는 인스턴스 시작 후 5분 후에 자동으로 시작됩니다.~~

- 수동 개입 필요 없음 - 대기하면 테스트가 백그라운드에서 완전히 실행됩니다.

2. 테스트를 모니터링합니다.
   - 인스턴스가 테스트를 자동으로 부팅하고 준비합니다.
   - 스크립트는 5분 동안 실행되고 >70% CPU 사용량을 생성합니다.
   - CloudWatch 경보는 총 8~10분(5분 지연 + 경보의 경우 3~5분) 이내에 트리거되어야 합니다.
3. 선택 사항: 수동 재실행(추가 테스트용):
   - 인스턴스에 연결: EC2 콘솔 → `AWS-DevOpsAgent-Test-Instance`→ 연결 → 세션 관리자
   - 스트레스 테스트를 다시 실행합니다`../cpu-stress-test.sh`
   - testing AWS DevOpsAgent의 응답에 여러 번 적합

테스트 옵션 B: Lambda 오류율 테스트

1단계: Lambda 테스트를 위한 CloudFormation 스택 배포

1. CloudFormation으로 이동합니다.
   a. AWS 콘솔에서 CloudFormation으로 이동합니다.
   b. 스택 생성 → 새 리소스 사용(표준)을 클릭합니다.
2. 템플릿 업로드:
   a. 라는 새 로컬 파일 생성`AWS-DevOpsAgent-lambda-test.yaml`
   b. 이 CloudFormation 템플릿을 복사하여 파일에 붙여 넣습니다.
      i.

```
AWSTemplateFormatVersion: '2010-09-09'
Description: 'AWS DevOpsAgent Lambda Error Test Stack'
Resources:
  # IAM Role for Lambda function
  LambdaExecutionRole:
    Type: AWS::IAM::Role
    Properties:
      RoleName: AWS-DevOpsAgentLambdaTestRole
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              Service: lambda.amazonaws.com
            Action: sts:AssumeRole
      ManagedPolicyArns:
```

```
        - arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole
      Tags:
        - Key: Name
          Value: AWS-DevOpsAgent-Lambda-Test-Role
        - Key: Purpose
          Value: AWS-DevOpsAgent-Testing
  # Lambda function that generates errors
  TestLambdaFunction:
    Type: AWS::Lambda::Function
    Properties:
      FunctionName: AWS-DevOpsAgent-test-lambda
      Runtime: python3.12
      Handler: index.lambda_handler
      Role: !GetAtt LambdaExecutionRole.Arn
      Code:
        ZipFile: |
          import json
          import random
          import time
          from datetime import datetime
          def lambda_handler(event, context):
              print(f"AWS DevOpsAgent Test Lambda - {datetime.now()}")
              print(f"Event: {json.dumps(event)}")

              # Intentionally generate errors for testing
              error_scenarios = [
                  "Simulated database connection timeout",
                  "Test API rate limit exceeded",
                  "Intentional validation error for AWS DevOpsAgent testing"
              ]

              # Always throw an error for testing purposes
              error_message = random.choice(error_scenarios)
              print(f"Generating test error: {error_message}")

              # This will create a Lambda error that CloudWatch will detect
              raise Exception(f"AWS DevOpsAgent Test Error: {error_message}")
      Description: AWS DevOpsAgent beta test function - intentionally generates
errors
      Timeout: 30
      Tags:
        - Key: Name
          Value: AWS-DevOpsAgent-Test-Lambda
        - Key: Purpose
```

```
          Value: AWS-DevOpsAgent-Testing
  # CloudWatch Alarm for Lambda errors
  LambdaErrorAlarm:
    Type: AWS::CloudWatch::Alarm
    Properties:
      AlarmName: AWS-DevOpsAgent-Lambda-Error-Test
      AlarmDescription: AWS-DevOpsAgent beta test - Lambda error rate alarm
      MetricName: Errors
      Namespace: AWS/Lambda
      Statistic: Sum
      Period: 60
      EvaluationPeriods: 1
      Threshold: 0
      ComparisonOperator: GreaterThanThreshold
      Dimensions:
        - Name: FunctionName
          Value: !Ref TestLambdaFunction
      TreatMissingData: notBreaching
Outputs:
  LambdaFunctionName:
    Description: Lambda Function Name for testing
    Value: !Ref TestLambdaFunction

  LambdaFunctionArn:
    Description: Lambda Function ARN
    Value: !GetAtt TestLambdaFunction.Arn

  AlarmName:
    Description: CloudWatch Alarm Name
    Value: !Ref LambdaErrorAlarm

  TestCommand:
    Description: AWS CLI command to test the function
    Value: !Sub 'aws lambda invoke --function-name ${TestLambdaFunction} --
payload "{\"test\":\"AWS DevOpsAgent validation\"}" response.json'
```

c. CloudFormation 콘솔에서 템플릿 파일 업로드를 선택합니다.

d. 파일 선택을 클릭합니다.

e. `AWS-DevOpsAgent-lambda-test.yaml` 파일 선택

f. 다음을 클릭합니다.

3. 스택 구성:

a. 스택 이름:`AWS-DevOpsAgent-Lambda-Test`

b. 다음을 클릭합니다.

4. 스택 옵션 구성:

a. 기본값을 그대로 두고 다음을 클릭합니다.

5. 검토 및 생성

a. AWS CloudFormation에서 IAM 리소스를 생성할 수 있음을 확인합니다.

b. 제출을 클릭합니다.

6. 완료될 때까지 기다립니다.

a. 스택 생성에는 2~3분이 소요됩니다.

b. 상태가 로 변경됩니다.`CREATE_COMPLETE`

2단계: Lambda 오류 트리거

1. Lambda 콘솔로 이동합니다.

a. AWS Lambda 콘솔로 이동

b. 함수 찾기`AWS-DevOpsAgent-test-lambda`

2. 함수를 테스트합니다.

a. 테스트 탭을 클릭합니다.

b. 새 이벤트 생성을 클릭합니다.

c. 이벤트 이름:`AWS-DevOpsAgent-test-event`

d. 이 JSON 페이로드를 사용합니다.

i.
```
{
"test": "AWS DevOpsAgent validation",
"timestamp": "2024-01-01T00:00:00Z"
}
```

e. 저장을 클릭합니다.

3. 오류 생성:

a. 테스트 버튼을 3번 클릭합니다(각각 간에 10초 대기).

b. 각 테스트는 의도적인 오류를 생성합니다.

c. CloudWatch 경보는 2~3분 이내에 트리거되어야 합니다.

d. 이제 AWS DevOpsAgent는 다음에 설정할 운영자 앱에서 조사를 통해 경보를 감지할 수 있습니

Validate AWS DevOps 에이전트 감지

1단계: CloudWatch 경보 안전성 검사(선택 사항)

이 단계는 위의 테스트가 이제 경보 상태인지 확인하기 위한 것입니다.

EC2 테스트의 경우:

- CloudWatch 콘솔에서 경보로 이동합니다.
- 스트레스 테스트를 시작한 후 3~5분 정도 기다립니다.
- 경보에 경보 상태가 표시되어야 합니다.
- 여전히 “확인”인 경우: 2~3분 더 기다립니다(CloudWatch 지표가 지연될 수 있음).

Lambda 테스트의 경우:

- `AWS-DevOpsAgent-Lambda-Error-Test`경보 확인
- 테스트 실행 후 2~3분 이내에 경보가 표시되어야 합니다.

2단계: a AWS DevOps 에이전트 조사 시작

1. AWS DevOps Agent AgentSpace 열기
2. 관리자 액세스를 클릭합니다. 그러면 DevOps 에이전트 스페이스 웹 앱이 새 창에서 열립니다.
3. 화면 오른쪽에 있는 조사 시작 버튼을 클릭합니다.
4. 다음 양식을 작성합니다.
   a. 조사 세부 정보: 실행하려는 조사를 설명합니다. 조사 목표, 탐색할 영역 또는 관련 정보에 대해 가능한 모든 세부 정보를 포함합니다.
   b. 조사 시작점: 조사를 시작하려는 정보를 설명합니다. 경보, 지표, 로그 조각 등을 언급하여 DevOps Agent에 작업 시작점을 제공할 수 있습니다. 이 경우 방금 생성한 경보의 요약을 제공합니다.
   c. 인시던트 날짜 및 시간(ISO 8601 선호): YYYY-MM-DDTHH:MMZ
   d. 조사 이름 지정: 예: `Oncall_investigation_1:2025-10-27`
   e. 인시던트의 AWS 계정 ID
   f. 인시던트가 발생한 리전
   g. Priority - AWS DevOpsAgent는 두 개의 동시 조사를 허용합니다. 우선 순위를 사용하면 조사 실행 순서를 정의할 수 있습니다.

5. 조사를 클릭하여 조사를 시작합니다.
6. 대시보드에 나열된 조사를 클릭합니다. DevOps 에이전트가 수행하는 세분화된 단계를 볼 수 있는 조사 세부 정보 화면으로 이동합니다.

예상 결과

EC2 테스트 결과:

- EC2 CPU 경보 감지
- 근본 원인 식별: "CPU 스트레스 테스트 워크로드"
- 타임라인 표시: 스트레스 테스트 → CPU 스파이크 → 경보
- 모니터링 및 조정에 대한 권장 사항을 제공합니다.

Lambda 테스트 결과:

- Lambda 오류율 스파이크 감지
- 근본 원인 식별: "의도적 테스트 예외"
- 타임라인 표시: 함수 호출 → 오류 → 경보
- 오류 처리 및 모니터링에 대한 권장 사항을 제공합니다.

정리 지침

정리 테스트 A(EC2 테스트)

자동 정리

- 인스턴스는 2시간 후에 자동 종료됩니다(CloudFormation 템플릿에 빌드됨).

수동 정리(즉시)

1. CloudFormation 스택 삭제:
   a. CloudFormation 콘솔로 이동
   b. `AWS-DevOpsAgent-EC2-Test`스택 선택
   c. 삭제를 클릭합니다.
   d. 삭제 확인

e. 그러면 EC2 인스턴스, 보안 그룹, 키 페어 및 CloudWatch 경보와 같은 모든 리소스가 자동으로 삭제됩니다.

정리 테스트 B(Lambda 테스트)

1. CloudFormation 스택 삭제:
   a. CloudFormation 콘솔로 이동
   b. `AWS-DevOpsAgent-Lambda-Test`스택 선택
   c. 삭제를 클릭합니다.
   d. 삭제 확인
   e. 그러면 Lambda 함수, IAM 역할 및 CloudWatch 경보와 같은 모든 리소스가 자동으로 삭제됩니다.

문제 해결

일반적인 문제

"EC2 인스턴스에 연결할 수 없음"

- 보안 그룹 확인: SSH(포트 22)가 IP에 열려 있는지 확인합니다.
- 키 권한 확인: 실행`chmod 400 AWS-DevOpsAgent-test-key.pem`
- 퍼블릭 IP 확인: 인스턴스에 퍼블릭 IP가 할당되어 있어야 합니다.
- 인스턴스 대기: 인스턴스가 "실행" 상태인지 확인합니다.

"경보가 트리거되지 않음"

- 지표 대기: CloudWatch 지표가 표시되는 데 2~5분이 걸릴 수 있습니다.
- CPU 로드 확인: 인스턴스에 대한 SSH 및를 실행`top`하여 CPU >70% 확인
- 스트레스 테스트 확인:를 실행`ps aux | grep yes`하여 로드 프로세스가 실행 중인지 확인
- 대기 시간 연장: 첫 번째 경보 트리거에 최대 7~8분이 걸리는 경우도 있음

테스트 검증

Your AWS DevOp 에이전트 테스트는 다음과 같은 경우에 성공합니다.

기술 검증

- 조사 정확도: EC2 테스트 결과는 CPU 로드로 인해 경보가 트리거되었음을 올바르게 나타내야 합니다. Lambda 테스트 결과는 이것이 의도적인 실패였음을 나타내야 합니다.
- 타임라인 정확도: 표시된 이벤트의 올바른 시퀀스
- 권장 품질: 실행 가능한 제안 제공

AWS CDK를 사용하여 AWS DevOps 에이전트 시작하기

개요

이 가이드에서는 AWS 클라우드 개발 키트(AWS CDK)를 사용하여 AWS DevOps 에이전트 리소스를 생성하고 배포하는 방법을 보여줍니다. AWS CDK 애플리케이션은 AWS CloudFormation을 통해 에이전트 공간, AWS 자격 증명 및 액세스 관리(IAM) 역할, 운영자 앱 및 AWS 계정 연결 생성을 자동화합니다.

AWS CDK 접근 방식은 필요한 모든 리소스를 코드형 인프라로 정의하여 CLI 온보딩 가이드에 설명된 수동 단계를 자동화합니다.

AWS DevOps 에이전트는 미국 동부(버지니아 북부), 미국 서부(오레곤), 아시아 태평양(시드니), 아시아 태평양(도쿄), 유럽(프랑크푸르트) 및 유럽(아일랜드)의 6개 AWS 리전에서 사용할 수 있습니다. 지원되는 리전에 대한 자세한 내용은 섹션을 참조하세요the section called “지원되는 리전:”.

사전 조건

시작하기 전에 다음 사항을 갖췄는지 확인하세요.

- AWS 적절한 자격 증명으로 설치 및 구성된 명령줄 인터페이스(AWS CLI)
- Node.js 버전 18 이상
- AWS CDK 명령줄 인터페이스(CLI)는 전역적으로 설치됩니다. AWS CDK CLI를 설치하려면 다음 명령을 실행합니다.

```
npm install -g aws-cdk
```

- 모니터링(기본) AWS 계정에 대한 하나의 계정
- (선택 사항) 교차 AWS 계정 모니터링을 설정하려는 경우 두 번째 계정

이 가이드에서 다루는 내용

이 가이드는 두 부분으로 나뉩니다.

- 1부 - 운영자 앱과 모니터링 계정의 AWS 연결을 사용하여 에이전트 공간을 배포합니다. 이 부분을 완료하면 에이전트가 해당 계정의 문제를 모니터링할 수 있습니다.
- 2부(선택 사항) - 서비스 계정에 대한 소스 AWS 연결을 추가하고 해당 계정에 교차 계정 IAM 역할을 배포합니다. 이 구성을 사용하면 에이전트 공간이 계정 전체에서 리소스를 모니터링할 수 있습니다.

생성할 리소스

1부: DevOpsAgentStack(모니터링 계정)

- IAM 역할(`DevOpsAgentRole-AgentSpace`) - 계정을 모니터링하기 위해 DevOps 에이전트 서비스에서 수임합니다. `AIDevOpsAgentAccessPolicy` 관리형 정책과 Resource Explorer 서비스 연결 역할 생성을 허용하는 인라인 정책을 포함합니다.
- IAM 역할(`DevOpsAgentRole-WebappAdmin`) - 에이전트 작업에 대한 `AIDevOpsOperatorAppAccessPolicy` 관리형 정책이 있는 운영자 앱 역할입니다.
- 에이전트 공간(`MyCDKAgentSpace`) - `AWS::DevOpsAgent::AgentSpace` CloudFormation 리소스를 사용하여 생성된 중앙 에이전트 공간입니다. 운영자 앱 구성을 포함합니다.
- 연결(AWS 모니터) - `AWS::DevOpsAgent::Association` CloudFormation 리소스를 사용하여 모니터링 계정을 에이전트 공간에 연결합니다.
- 연결(AWS 소스) - (선택 사항) 교차 계정 모니터링을 위해 서비스 계정을 에이전트 스페이스에 연결합니다.

2부: ServiceStack(서비스 계정, 선택 사항)

- IAM 역할(`DevOpsAgentRole-SecondaryAccount`) - 이름이 고정된 교차 계정 역할입니다. 모니터링 계정의 에이전트 공간에서 신뢰합니다. `AIDevOpsAgentAccessPolicy` 관리형 정책과 Resource Explorer 서비스 연결 역할 생성을 허용하는 인라인 정책을 포함합니다.
- Lambda 함수(`echo-service`) - 입력 이벤트를 에코백하는 간단한 예제 서비스입니다.

설정

1단계: 샘플 리포지토리 복제

다음 명령을 실행하여 리포지토리를 복제하고 프로젝트 디렉터리로 변경합니다.

```
git clone https://github.com/aws-samples/sample-aws-devops-agent-cdk.git
cd sample-aws-devops-agent-cdk
```

2단계: 종속성 설치

다음 명령을 실행하여 프로젝트 종속성을 설치합니다.

```
npm install
```

1부: 에이전트 공간 배포

이 섹션에서는 모니터링 계정에 에이전트 공간, IAM 역할, 운영자 앱 및 AWS 연결을 생성합니다.

1단계: 모니터링 계정 ID 구성

`lib/constants.ts`를 열고 모니터링 계정 ID를 설정합니다.

다음 예제에서는 업데이트할 상수를 보여줍니다.

```
export const MONITORING_ACCOUNT_ID = "<YOUR_MONITORING_ACCOUNT_ID>";
```

2단계: AWS CDK 환경 부트스트랩

모니터링 계정에서 AWS CDK를 부트스트래핑하지 않은 경우 다음 명령을 실행합니다.

```
cdk bootstrap aws://<MONITORING_ACCOUNT_ID>/<REGION> --profile monitoring
```

3단계: 빌드 및 배포

다음 명령을 실행하여 TypeScript 코드를 빌드하고 스택을 배포합니다.

```
npm run build
```

```
cdk deploy DevOpsAgentStack --profile monitoring
```

4단계: 스택 출력 기록

배포가 완료되면 AWS CDK는 스택 출력을 인쇄합니다. 나중에 사용할 수 있도록 이러한 값을 기록합니다.

다음 예제에서는 예상 출력을 보여줍니다.

```
Outputs:
DevOpsAgentStack.AgentSpaceArn = arn:aws:aidevops:<REGION>:123456789012:agentspace/
abc123
DevOpsAgentStack.AgentSpaceRoleArn = arn:aws:iam::123456789012:role/DevOpsAgentRole-
AgentSpace
DevOpsAgentStack.OperatorRoleArn = arn:aws:iam::123456789012:role/DevOpsAgentRole-
WebappAdmin
DevOpsAgentStack.AssociationId = assoc-xyz
```

2부를 완료하려는 경우 AgentSpaceArn 값을 저장합니다. 서비스 계정 스택을 구성하는 데 필요합니다.

5단계: 배포 확인

에이전트 공간이 성공적으로 생성되었는지 확인하려면 다음 AWS CLI 명령을 실행합니다.

```
aws devopsagent get-agent-space \
  --agent-space-id <AGENT_SPACE_ID> \
  --region <REGION>
```

이 시점에서 에이전트 공간은 운영자 앱이 활성화되고 모니터링 계정이 연결된 상태로 배포됩니다. 에이전트는이 계정의 문제를 모니터링할 수 있습니다.

2부(선택 사항): 교차 계정 모니터링 추가

이 섹션에서는 에이전트 공간이 두 번째 AWS 계정(서비스 계정)의 리소스를 모니터링할 수 있도록 설정을 확장합니다. 여기에는 다음 두 가지 작업이 포함됩니다.

1. 서비스 계정을 가리키는 소스 AWS 연결을 DevOpsAgentStack에 추가합니다.
2. 에이전트 공간을 신뢰하는 IAM 역할을 사용하여 ServiceStack을 서비스 계정에 배포합니다.

> **Important**
>
> 계속하기 전에 1부를 완료해야 합니다. ServiceStack에는 DevOpsAgentStack 배포 출력AgentSpaceArn의이 필요합니다.

1단계: 서비스 계정 ID 구성

서비스 계정 ID를 lib/constants.ts 열고 설정합니다.

다음 예제에서는 업데이트할 상수를 보여줍니다.

```
export const SERVICE_ACCOUNT_ID = "<YOUR_SERVICE_ACCOUNT_ID>";
```

DevOpsAgentStack은이 계정 ID를 사용하여 소스 AWS 연결을 생성합니다. 이 값을 설정하기 전에 DevOpsAgentStack을 배포한 경우 다시 배포하여 연결을 생성합니다.

다음 명령을 실행하여 재배포합니다.

```
npm run build
cdk deploy DevOpsAgentStack --profile monitoring
```

2단계: 에이전트 공간 ARN 설정

DevOpsAgentStack 출력(1부, 4단계)에서 AgentSpaceArn 값을 복사하고에서 설정합니다lib/constants.ts.

다음 예제에서는 업데이트할 상수를 보여줍니다.

```
export const AGENT_SPACE_ARN =
 "arn:aws:aidevops:<REGION>:<MONITORING_ACCOUNT_ID>:agentspace/<SPACE_ID>";
```

ServiceStack은이 값을 사용하여 보조 계정 역할에 대한 신뢰 정책의 범위를 지정합니다. ServiceStack은이 값이 설정된 경우에만 합성됩니다.

3단계: 서비스 계정 부트스트랩

서비스 계정에서 AWS CDK를 부트스트래핑하지 않은 경우 다음 명령을 실행합니다.

```
cdk bootstrap aws://<SERVICE_ACCOUNT_ID>/<REGION> --profile service
```

4단계: ServiceStack 배포

다음 명령을 실행하여 서비스 계정의 자격 증명을 사용하여 ServiceStack을 빌드하고 배포합니다.

```
npm run build
cdk deploy ServiceStack --profile service
```

이렇게 하면 서비스 계정에 다음 리소스가 생성됩니다.

- 모니터링 계정의 에이전트 공간을 신뢰하는 IAM 역할(`DevOpsAgentRole-SecondaryAccount`)
- 예제 서비스로 사용되는 에코 Lambda 함수(`echo-service`)

5단계: 배포 확인

Lambda 함수가 성공적으로 배포되었는지 확인하려면 다음 명령을 실행하여 에코 서비스를 테스트합니다.

```
aws lambda invoke \
  --function-name echo-service \
  --payload '{"test": "hello world"}' \
  --profile service \
  response.json
cat response.json
```

문제 해결

이 섹션에서는 일반적인 문제와 해결 방법을 설명합니다.

CloudFormation 리소스 유형을 찾을 수 없음

- 에 배포하고 있는지 확인합니다the section called “지원되는 리전:”.
- AWS CLI가 적절한 권한으로 구성되어 있는지 확인합니다.

IAM 역할 생성 실패

- 배포 역할에 IAM 역할을 생성할 수 있는 권한이 있는지 확인합니다.
- 신뢰 정책 조건이 계정 ID와 일치하는지 확인합니다.

"대상 계정에서 역할을 수임할 수 없음"과 함께 교차 계정 배포가 실패합니다.

- 각 스택은 대상 계정의 자격 증명으로 배포해야 합니다. `--profile` 플래그를 사용하여 올바른 AWS CLI 프로파일을 지정합니다.
- 대상 계정에서 AWS CDK가 부트스트랩되었는지 확인합니다.

IAM 전파 지연

- IAM 역할 변경 사항이 전파되는 데 몇 분 정도 걸릴 수 있습니다. 역할 생성 직후 에이전트 공간 생성에 실패하면 몇 분 정도 기다렸다가 다시 배포합니다.

정리

모든 리소스를 제거하려면 스택을 역순으로 삭제합니다.

다음 명령을 실행하여 스택을 삭제합니다.

```
# If you deployed the ServiceStack, destroy it first
cdk destroy ServiceStack --profile service
# Then destroy the DevOpsAgentStack
cdk destroy DevOpsAgentStack --profile monitoring
```

경고:이 작업은 에이전트 공간과 모든 관련 데이터를 영구적으로 삭제합니다. 이 작업은 실행 취소할 수 없습니다. 계속하기 전에 중요한 정보를 백업했는지 확인합니다.

보안 고려 사항

- AWS CDK 애플리케이션은 `aidevops.amazonaws.com` 서비스 보안 주체만 해당 역할을 수임하도록 허용하는 신뢰 정책을 사용하여 IAM 역할을 생성합니다.
- 신뢰 정책에는 특정 AWS 계정 및 에이전트 공간 ARN에 대한 액세스를 제한하는 조건이 포함됩니다.
- 모든 정책은 최소 권한 원칙을 따릅니다. 조직의 보안 요구 사항에 따라 IAM 정책을 검토하고 사용자 지정합니다.
- 교차 계정 역할(`DevOpsAgentRole-SecondaryAccount`)은 고정된 이름을 사용하며 특정 에이전트 공간 ARN으로 범위가 지정됩니다.

다음 단계

AWS CDK를 사용하여 AWS DevOps 에이전트를 배포한 후:

1. DevOps 에이전트 AWS 사용 설명서에서 DevOps 에이전트 기능의 전체 범위에 대해 알아봅니다.
2. 자동화된 인프라 관리를 위해 AWS CDK 배포를 CI/CD 파이프라인에 통합하는 것이 좋습니다.

추가 리소스

- AWS DevOps 에이전트 사용 설명서
- GitHub 웹 사이트의 샘플 CDK 리포지토리
- CLI 온보딩 가이드

AWS CloudFormation을 사용하여 AWS DevOps 에이전트 시작하기

개요

이 가이드에서는 AWS CloudFormation 템플릿을 사용하여 AWS DevOps 에이전트 리소스를 생성하고 배포하는 방법을 보여줍니다. 템플릿은 에이전트 공간, AWS 자격 증명 및 액세스 관리(IAM) 역할, 운영자 앱 및 AWS 계정 연결을 코드형 인프라로 자동 생성합니다.

CloudFormation 접근 방식은 선언적 YAML 템플릿에 필요한 모든 리소스를 정의하여 CLI 온보딩 가이드에 설명된 수동 단계를 자동화합니다.

AWS DevOps 에이전트는 미국 동부(버지니아 북부), 미국 서부(오레곤), 아시아 태평양(시드니), 아시아 태평양(도쿄), 유럽(프랑크푸르트) 및 유럽(아일랜드)의 6개 AWS 리전에서 사용할 수 있습니다. 지원되는 리전에 대한 자세한 내용은 섹션을 참조하세요the section called “지원되는 리전:”.

사전 조건

시작하기 전에 다음 사항을 갖췄는지 확인하세요.

- AWS 적절한 자격 증명으로 설치 및 구성된 명령줄 인터페이스(AWS CLI)
- IAM 역할 및 CloudFormation 스택을 생성할 수 있는 권한
- 모니터링(기본) AWS 계정에 대한 하나의 계정
- (선택 사항) 교차 AWS 계정 모니터링을 설정하려는 경우 두 번째 계정

이 가이드에서 다루는 내용

이 가이드는 두 부분으로 나뉩니다.

- 1부 - 운영자 앱과 모니터링 계정의 AWS 연결을 사용하여 에이전트 공간을 배포합니다. 이 부분을 완료하면 에이전트가 해당 계정의 문제를 모니터링할 수 있습니다.
- 2부(선택 사항) - 교차 계정 IAM 역할을 보조 계정에 배포하고 소스 AWS 연결을 추가합니다. 이 구성을 사용하면 에이전트 공간이 계정 전체에서 리소스를 모니터링할 수 있습니다.

1부: 에이전트 공간 배포

이 섹션에서는 에이전트 공간, IAM 역할, 운영자 앱 및 모니터링 계정의 AWS 연결을 프로비저닝하는 CloudFormation 템플릿을 생성합니다.

1단계: CloudFormation 템플릿 생성

다음 템플릿을 로 저장합니다`devops-agent-stack.yaml`.

```
AWSTemplateFormatVersion: '2010-09-09'
Description: AWS DevOps Agent - Agent Space with IAM roles, operator app, and AWS
 association

Parameters:
  AgentSpaceName:
    Type: String
    Default: MyCloudFormationAgentSpace
    Description: Name for the agent space
  AgentSpaceDescription:
    Type: String
    Default: Agent space deployed with CloudFormation
    Description: Description for the agent space

Resources:
  # IAM role assumed by the DevOps Agent service to monitor the account
  DevOpsAgentSpaceRole:
    Type: AWS::IAM::Role
    Properties:
      RoleName: DevOpsAgentRole-AgentSpace
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              Service: aidevops.amazonaws.com
            Action: sts:AssumeRole
            Condition:
```

```
              StringEquals:
                aws:SourceAccount: !Ref AWS::AccountId
              ArnLike:
                aws:SourceArn: !Sub arn:aws:aidevops:${AWS::Region}:
${AWS::AccountId}:agentspace/*
      ManagedPolicyArns:
        - arn:aws:iam::aws:policy/AIDevOpsAgentAccessPolicy
      Policies:
        - PolicyName: AllowCreateServiceLinkedRoles
          PolicyDocument:
            Version: '2012-10-17'
            Statement:
              - Sid: AllowCreateServiceLinkedRoles
                Effect: Allow
                Action:
                  - iam:CreateServiceLinkedRole
                Resource:
                  - !Sub arn:aws:iam::${AWS::AccountId}:role/aws-service-role/resource-
explorer-2.amazonaws.com/AWSServiceRoleForResourceExplorer

  # IAM role for the operator app interface
  DevOpsOperatorRole:
    Type: AWS::IAM::Role
    Properties:
      RoleName: DevOpsAgentRole-WebappAdmin
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              Service: aidevops.amazonaws.com
            Action:
              - sts:AssumeRole
              - sts:TagSession
            Condition:
              StringEquals:
                aws:SourceAccount: !Ref AWS::AccountId
              ArnLike:
                aws:SourceArn: !Sub arn:aws:aidevops:${AWS::Region}:
${AWS::AccountId}:agentspace/*
      ManagedPolicyArns:
        - arn:aws:iam::aws:policy/AIDevOpsOperatorAppAccessPolicy

  # The agent space resource
```

```
  AgentSpace:
    Type: AWS::DevOpsAgent::AgentSpace
    DependsOn:
      - DevOpsAgentSpaceRole
      - DevOpsOperatorRole
    Properties:
      Name: !Ref AgentSpaceName
      Description: !Ref AgentSpaceDescription
      OperatorApp:
        Iam:
          OperatorAppRoleArn: !GetAtt DevOpsOperatorRole.Arn

  # Association linking the monitoring account to the agent space
  MonitorAssociation:
    Type: AWS::DevOpsAgent::Association
    Properties:
      AgentSpaceId: !GetAtt AgentSpace.AgentSpaceId
      ServiceId: aws
      Configuration:
        Aws:
          AssumableRoleArn: !GetAtt DevOpsAgentSpaceRole.Arn
          AccountId: !Ref AWS::AccountId
          AccountType: monitor

Outputs:
  AgentSpaceId:
    Description: The agent space ID
    Value: !GetAtt AgentSpace.AgentSpaceId
  AgentSpaceArn:
    Description: The agent space ARN
    Value: !GetAtt AgentSpace.Arn
  AgentSpaceRoleArn:
    Description: The agent space IAM role ARN
    Value: !GetAtt DevOpsAgentSpaceRole.Arn
  OperatorRoleArn:
    Description: The operator app IAM role ARN
    Value: !GetAtt DevOpsOperatorRole.Arn
```

2단계: 스택 배포

다음 명령을 실행하여 스택을 배포합니다. 를 <REGION>로 바꿉니다the section called “지원되는 리전:”(예: us-east-1).

```
aws cloudformation deploy \
  --template-file devops-agent-stack.yaml \
  --stack-name DevOpsAgentStack \
  --capabilities CAPABILITY_NAMED_IAM \
  --region <REGION>
```

3단계: 스택 출력 기록

배포가 완료되면 다음 명령을 실행하여 스택 출력을 검색합니다. 나중에 사용할 수 있도록 이러한 값을 기록합니다.

```
aws cloudformation describe-stacks \
  --stack-name DevOpsAgentStack \
  --query 'Stacks[0].Outputs' \
  --region <REGION>
```

다음 예제에서는 예상 출력을 보여줍니다.

```
[
  {
    "OutputKey": "AgentSpaceId",
    "OutputValue": "abc123def456"
  },
  {
    "OutputKey": "AgentSpaceArn",
    "OutputValue": "arn:aws:aidevops:<REGION>:<ACCOUNT_ID>:agentspace/abc123def456"
  },
  {
    "OutputKey": "AgentSpaceRoleArn",
    "OutputValue": "arn:aws:iam::<ACCOUNT_ID>:role/DevOpsAgentRole-AgentSpace"
  },
  {
    "OutputKey": "OperatorRoleArn",
    "OutputValue": "arn:aws:iam::<ACCOUNT_ID>:role/DevOpsAgentRole-WebappAdmin"
  }
]
```

2부를 완료하려는 경우 `AgentSpaceArn` 값을 저장합니다. 교차 계정 역할을 구성하는 데 필요합니다.

4단계: 배포 확인

에이전트 공간이 성공적으로 생성되었는지 확인하려면 다음 AWS CLI 명령을 실행합니다.

```
aws devops-agent get-agent-space \
  --agent-space-id <AGENT_SPACE_ID> \
  --region <REGION>
```

이 시점에서 에이전트 공간은 운영자 앱이 활성화되고 모니터링 계정이 연결된 상태로 배포됩니다. 에이전트는이 계정의 문제를 모니터링할 수 있습니다.

2부(선택 사항): 교차 계정 모니터링 추가

이 섹션에서는 에이전트 공간이 두 번째 AWS 계정(서비스 계정)의 리소스를 모니터링할 수 있도록 설정을 확장합니다. 여기에는 다음 두 가지 작업이 포함됩니다.

1. 에이전트 공간을 신뢰하는 서비스 계정에 IAM 역할을 배포합니다.
2. 서비스 계정을 가리키는 소스 AWS 연결을 모니터링 계정에 추가합니다.

> 참고: 계속하기 전에 1부를 완료해야 합니다. 서비스 계정 템플릿에는 파트 1 스택 출력`AgentSpaceArn`의가 필요합니다.

1단계: 서비스 계정 템플릿 생성

다음 템플릿을 로 저장합니다`devops-agent-service-account.yaml`. 이 템플릿은 보조 계정에 교차 계정 IAM 역할을 생성합니다.

```
AWSTemplateFormatVersion: '2010-09-09'
Description: AWS DevOps Agent - Cross-account IAM role for secondary account monitoring

Parameters:
  MonitoringAccountId:
    Type: String
    Description: The 12-digit AWS account ID of the monitoring account
  AgentSpaceArn:
    Type: String
    Description: The ARN of the agent space from the monitoring account

Resources:
  # Cross-account IAM role trusted by the agent space
  DevOpsSecondaryAccountRole:
    Type: AWS::IAM::Role
    Properties:
      RoleName: DevOpsAgentRole-SecondaryAccount
```

```
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              Service: aidevops.amazonaws.com
            Action: sts:AssumeRole
            Condition:
              StringEquals:
                aws:SourceAccount: !Ref MonitoringAccountId
              ArnLike:
                aws:SourceArn: !Ref AgentSpaceArn
      ManagedPolicyArns:
        - arn:aws:iam::aws:policy/AIDevOpsAgentAccessPolicy
      Policies:
        - PolicyName: AllowCreateServiceLinkedRoles
          PolicyDocument:
            Version: '2012-10-17'
            Statement:
              - Sid: AllowCreateServiceLinkedRoles
                Effect: Allow
                Action:
                  - iam:CreateServiceLinkedRole
                Resource:
                  - !Sub arn:aws:iam::${AWS::AccountId}:role/aws-service-role/resource-
explorer-2.amazonaws.com/AWSServiceRoleForResourceExplorer

Outputs:
  SecondaryAccountRoleArn:
    Description: The cross-account IAM role ARN
    Value: !GetAtt DevOpsSecondaryAccountRole.Arn
```

2단계: 서비스 계정 스택 배포

서비스 계정에 대한 자격 증명을 사용하여 다음 명령을 실행합니다.

```
aws cloudformation deploy \
  --template-file devops-agent-service-account.yaml \
  --stack-name DevOpsAgentServiceAccountStack \
  --capabilities CAPABILITY_NAMED_IAM \
  --parameter-overrides \
    MonitoringAccountId=<MONITORING_ACCOUNT_ID> \
    AgentSpaceArn=<AGENT_SPACE_ARN> \
```

```
  --region <REGION>
```

3단계: 소스 AWS 연결 추가

모니터링 계정으로 다시 전환하고 소스 AWS 연결을 생성합니다. 별도의 스택을 생성하거나 원본 템플릿을 업데이트하여이 작업을 수행할 수 있습니다. 다음 예제에서는 독립 실행형 템플릿을 사용합니다.

다음 템플릿을 로 저장합니다`devops-agent-source-association.yaml`.

```
AWSTemplateFormatVersion: '2010-09-09'
Description: AWS DevOps Agent - Source AWS association for cross-account monitoring

Parameters:
  AgentSpaceId:
    Type: String
    Description: The agent space ID from the monitoring account stack
  ServiceAccountId:
    Type: String
    Description: The 12-digit AWS account ID of the service account
  ServiceAccountRoleArn:
    Type: String
    Description: The ARN of the DevOpsAgentRole-SecondaryAccount role in the service
 account

Resources:
  SourceAssociation:
    Type: AWS::DevOpsAgent::Association
    Properties:
      AgentSpaceId: !Ref AgentSpaceId
      ServiceId: aws
      Configuration:
        SourceAws:
          AccountId: !Ref ServiceAccountId
          AccountType: source
          AssumableRoleArn: !Ref ServiceAccountRoleArn

Outputs:
  SourceAssociationId:
    Description: The source association ID
    Value: !Ref SourceAssociation
```

모니터링 계정 자격 증명을 사용하여 연결 스택을 배포합니다.

```
aws cloudformation deploy \
  --template-file devops-agent-source-association.yaml \
  --stack-name DevOpsAgentSourceAssociationStack \
  --parameter-overrides \
    AgentSpaceId=<AGENT_SPACE_ID> \
    ServiceAccountId=<SERVICE_ACCOUNT_ID> \
    ServiceAccountRoleArn=arn:aws:iam::<SERVICE_ACCOUNT_ID>:role/DevOpsAgentRole-
SecondaryAccount \
  --region <REGION>
```

Verification(확인)

다음 AWS CLI 명령을 실행하여 설정을 확인합니다.

```
# List your agent spaces
aws devops-agent list-agent-spaces \
  --region <REGION>

# Get details of a specific agent space
aws devops-agent get-agent-space \
  --agent-space-id <AGENT_SPACE_ID> \
  --region <REGION>

# List associations for an agent space
aws devops-agent list-associations \
  --agent-space-id <AGENT_SPACE_ID> \
  --region <REGION>
```

문제 해결

이 섹션에서는 일반적인 문제와 해결 방법을 설명합니다.

CloudFormation 리소스 유형을 찾을 수 없음

- 에 배포하고 있는지 확인합니다the section called “지원되는 리전:”.
- AWS CLI가 적절한 권한으로 구성되어 있는지 확인합니다.

IAM 역할 생성 실패

- 배포 자격 증명에 사용자 지정 이름()으로 IAM 역할을 생성할 수 있는 권한이 있는지 확인합니다CAPABILITY_NAMED_IAM.

- 신뢰 정책 조건이 계정 ID와 일치하는지 확인합니다.

교차 계정 배포 실패

- 각 스택은 대상 계정의 자격 증명으로 배포해야 합니다. `--profile` 플래그를 사용하여 올바른 AWS CLI 프로파일을 지정합니다.
- `AgentSpaceArn` 파라미터가 파트 1 스택 출력의 정확한 ARN과 일치하는지 확인합니다.

IAM 전파 지연

- IAM 역할 변경 사항이 전파되는 데 몇 분 정도 걸릴 수 있습니다. 역할 생성 직후 에이전트 공간 생성에 실패하면 몇 분 정도 기다렸다가 다시 배포합니다.

정리

모든 리소스를 제거하려면 스택을 역순으로 삭제합니다.

경고:이 작업은 에이전트 공간과 모든 관련 데이터를 영구적으로 삭제합니다. 이 작업은 실행 취소할 수 없습니다. 계속하기 전에 중요한 정보를 백업했는지 확인합니다.

다음 명령을 실행하여 스택을 삭제합니다.

```
# If you deployed the source association stack, delete it first
aws cloudformation delete-stack \
  --stack-name DevOpsAgentSourceAssociationStack \
  --region <REGION>

aws cloudformation wait stack-delete-complete \
  --stack-name DevOpsAgentSourceAssociationStack \
  --region <REGION>

# If you deployed the service account stack, delete it next (using service account
 credentials)
aws cloudformation delete-stack \
  --stack-name DevOpsAgentServiceAccountStack \
  --region <REGION>

aws cloudformation wait stack-delete-complete \
  --stack-name DevOpsAgentServiceAccountStack \
  --region <REGION>
```

```
# Delete the main stack last
aws cloudformation delete-stack \
  --stack-name DevOpsAgentStack \
  --region <REGION>
```

다음 단계

AWS CloudFormation을 사용하여 AWS DevOps 에이전트를 배포한 후:

- 추가 통합을 연결하려면 섹션을 참조하세요AWS DevOps Agent에 대한 기능 구성.
- 에이전트 기술 및 기능에 대한 자세한 내용은 섹션을 참조하세요the section called “DevOps 에이전트 기술”.
- 연산자 웹 앱을 이해하려면 섹션을 참조하세요the section called “DevOps 에이전트 웹 앱이란 무엇입니까?”.

Terraform을 사용하여 AWS DevOps 에이전트 시작하기

개요

이 가이드에서는 Terraform을 사용하여 AWS DevOps 에이전트 리소스를 생성하고 배포하는 방법을 보여줍니다. Terraform 구성은 에이전트 공간, IAM 역할, 운영자 앱 및 AWS 계정 연결 생성을 자동화합니다.

Terraform 접근 방식은 필요한 모든 리소스를 코드형 인프라로 정의하여 CLI 온보딩 가이드에 설명된 수동 단계를 자동화합니다.

AWS DevOps 에이전트는 미국 동부(버지니아 북부), 미국 서부(오레곤), 아시아 태평양(시드니), 아시아 태평양(도쿄), 유럽(프랑크푸르트) 및 유럽(아일랜드)의 6개 AWS 리전에서 사용할 수 있습니다. 지원되는 리전에 대한 자세한 내용은 섹션을 참조하세요the section called “지원되는 리전:”.

사전 조건

시작하기 전에 다음 항목이 준비되었는지 확인합니다.

- Terraform >= 1.0 설치됨
- AWS 적절한 자격 증명으로 CLI 설치 및 구성

- 모니터링(기본) AWS 계정에 대한 하나의 계정
- (선택 사항) 교차 AWS 계정 모니터링을 설정하려는 경우 두 번째 계정

이 가이드에서 다루는 내용

이 가이드는 두 부분으로 나뉩니다.

- 1부 - 운영자 앱과 모니터링 계정의 AWS 연결을 사용하여 에이전트 공간을 배포합니다. 이 부분을 완료한 후 에이전트는 해당 계정의 문제를 모니터링할 수 있습니다.
- 2부(선택 사항) - 서비스 계정에 대한 소스 AWS 연결을 추가하고 교차 계정 IAM 역할과 에코 Lambda를 해당 계정에 배포합니다. 이렇게 하면 에이전트 공간이 계정 전체에서 리소스를 모니터링할 수 있습니다.

생성할 리소스

1부: 계정 모니터링

- IAM 역할(`DevOpsAgentRole-AgentSpace-*`) - 계정을 모니터링하기 위해 DevOps 에이전트 서비스에서 수임합니다. `AIDevOpsAgentAccessPolicy` 관리형 정책과 Resource Explorer 서비스 연결 역할 생성을 허용하는 인라인 정책을 포함합니다.
- IAM 역할(`DevOpsAgentRole-WebappAdmin-*`) - 에이전트 작업에 대한 `AIDevOpsOperatorAppAccessPolicy` 관리형 정책이 있는 운영자 앱 역할입니다.
- 에이전트 공간(구성 가능한 이름) - `awscc_devopsagent_agent_space` 리소스를 사용하여 생성된 중앙 에이전트 공간입니다. 운영자 앱 구성을 포함합니다.
- 연결(AWS 모니터) - `awscc_devopsagent_association` 리소스를 사용하여 모니터링 계정을 에이전트 스페이스에 연결합니다.
- 연결(AWS 소스) - (선택 사항) 교차 계정 모니터링을 위해 서비스 계정을 에이전트 스페이스에 연결합니다.

2부: 서비스 계정(선택 사항)

- IAM 역할(`DevOpsAgentRole-SecondaryAccount-TF`) - 이름이 고정된 교차 계정 역할입니다. 모니터링 계정의 에이전트 공간에서 신뢰합니다. `AIDevOpsAgentAccessPolicy` 관리형 정책과 Resource Explorer 서비스 연결 역할 생성을 허용하는 인라인 정책을 포함합니다.
- Lambda 함수(`echo-service-tf`) - 입력 이벤트를 에코백하는 간단한 예제 서비스입니다.

설정

1단계: 샘플 리포지토리 복제

```
git clone https://github.com/aws-samples/sample-aws-devops-agent-terraform.git
cd sample-aws-devops-agent-terraform
```

2단계: 변수 구성

예제 변수 파일을 복사하여 환경에 맞게 사용자 지정합니다.

```
cp terraform.tfvars.example terraform.tfvars
```

에이전트 스페이스 이름 및 설명`terraform.tfvars`으로를 편집합니다.

```
agent_space_name        = "MyCompanyAgentSpace"
agent_space_description = "DevOps Agent Space for monitoring production workloads"
```

1부: 에이전트 공간 배포

이 섹션에서는 모니터링 계정에 에이전트 공간, IAM 역할, 운영자 앱 및 AWS 연결을 생성합니다.

1단계: 자동화를 사용하여 배포(권장)

간소화된 설정을 위해 제공된 배포 스크립트를 사용합니다.

```
./deploy.sh
```

이 스크립트는 자동으로 다음을 수행합니다.

- 사전 조건 확인(Terraform, AWS CLI, 자격 증명)
- 필요한 경우 예제`terraform.tfvars`에서 생성
- Terraform 초기화, 검증, 계획 및 적용

또는 수동 제어를 선호하는 경우:

```
terraform init
terraform plan
terraform apply
```

배포를 확인하라는 메시지가 yes 표시되면를 입력합니다.

2단계: 출력 기록

배포가 완료되면 Terraform이 출력을 인쇄합니다. 나중에 사용할 수 있도록 다음 값을 기록합니다.

```
Outputs:
agent_space_id              = "abc123"
agent_space_arn             =
 "arn:aws:aidevops:<REGION>:<MONITORING_ACCOUNT_ID>:agentspace/abc123"
agent_space_name            = "MyCompanyAgentSpace"
devops_agentspace_role_arn  = "arn:aws:iam::<MONITORING_ACCOUNT_ID>:role/
DevOpsAgentRole-AgentSpace-a1b2c3d4"
devops_operator_role_arn    = "arn:aws:iam::<MONITORING_ACCOUNT_ID>:role/
DevOpsAgentRole-WebappAdmin-a1b2c3d4"
primary_account_id          = "<MONITORING_ACCOUNT_ID>"
primary_account_association_id = "assoc-xyz"
```

2부를 완료하려는 경우 agent_space_arn 값을 저장합니다. 서비스 계정 리소스를 구성하는 데 필요합니다.

3단계: 배포 확인

배포 후 확인 스크립트를 실행합니다.

```
./post-deploy.sh
```

또는 AWS CLI를 사용하여 에이전트 공간이 성공적으로 생성되었는지 확인합니다.

```
aws devops-agent get-agent-space \
  --agent-space-id <AGENT_SPACE_ID> \
  --region <REGION>
```

이 시점에서 에이전트 공간은 운영자 앱이 활성화되고 모니터링 계정이 연결된 상태로 배포됩니다. 에이전트는이 계정의 문제를 모니터링할 수 있습니다.

2부(선택 사항): 교차 계정 모니터링 추가

이 섹션에서는 에이전트 공간이 두 번째 AWS 계정(서비스 계정)의 리소스를 모니터링할 수 있도록 설정을 확장합니다. 여기에는 다음 두 가지 작업이 포함됩니다.

1. 서비스 계정을 가리키는 소스 AWS 연결 추가.
2. 교차 계정 IAM 역할과 에코 Lambda 함수를 서비스 계정에 배포합니다.

> **Important**
>
> 계속하기 전에 1부를 완료해야 합니다. 서비스 계정 리소스에는 파트 1 배포 출력`agent_space_arn`의가 필요합니다.

1단계: 서비스 계정 ID 구성

에서 서비스 계정 ID를 `terraform.tfvars`설정합니다.

```
service_account_id = "<YOUR_SERVICE_ACCOUNT_ID>"
```

2단계: 에이전트 공간 ARN 설정

1부 출력(2단계)에서 `agent_space_arn` 값을 복사하고에서 설정합니다`terraform.tfvars`.

```
agent_space_arn = "arn:aws:aidevops:<REGION>:<MONITORING_ACCOUNT_ID>:agentspace/
<SPACE_ID>"
```

서비스 계정 리소스는이 값을 사용하여 보조 계정 역할에 대한 신뢰 정책의 범위를 지정합니다. 이러한 리소스는이 값이 설정된 경우에만 생성됩니다.

3단계: `aws.service` 공급자 구성

에서 서비스 계정의 자격 증명을 사용하여 `aws.service` 공급자 별칭을 `main.tf`구성합니다. 명명된 프로필 또는 수임 역할을 사용할 수 있습니다.

프로필 사용:

```
provider "aws" {
  alias   = "service"
  region  = var.aws_region
  profile = "your-service-account-profile"
}
```

또는 수임 역할 사용:

```
provider "aws" {
  alias  = "service"
  region = var.aws_region
  assume_role {
    role_arn = "arn:aws:iam::<SERVICE_ACCOUNT_ID>:role/OrganizationAccountAccessRole"
  }
}
```

4단계: 배포

업데이트된 구성을 적용합니다.

```
terraform apply
```

이렇게 하면 서비스 계정에 다음 리소스가 생성됩니다.

- 모니터링 계정의 에이전트 공간을 신뢰하는 IAM 역할(`DevOpsAgentRole-SecondaryAccount-TF`)
- 예제 서비스로 사용되는 에코 Lambda 함수(`echo-service-tf`)

또한 서비스 계정을 연결하는 소스 AWS 연결을 모니터링 계정에 생성합니다.

5단계: 배포 확인

에코 서비스를 테스트하여 Lambda 함수가 성공적으로 배포되었는지 확인합니다.

```
aws lambda invoke \
  --function-name echo-service-tf \
  --payload '{"test": "hello world"}' \
  --profile <your-service-account-profile> \
  --region <REGION> \
  response.json
cat response.json
```

문제 해결

IAM 전파 지연

- 구성에는 IAM 역할 생성과 에이전트 공간 생성 `time_sleep` 사이에 30초가 포함됩니다. DevOps 에이전트 서비스는 에이전트 스페이스 생성 중에 운영자 역할의 신뢰 정책을 검증하며, IAM이 완전

히 전파되지 않은 경우 실패할 수 있습니다. 그래도 신뢰 정책 오류가 계속 표시되면 잠시 기다렸다가 `terraform apply` 다시 실행합니다. IAM 역할이 이미 존재하고 적용이 중단된 위치를 선택합니다.

권한 오류

- 자격 AWS 증명에 역할 및 정책을 생성하는 데 필요한 IAM 권한이 있는지 확인합니다.
- 신뢰 정책 조건이 계정 ID와 일치하는지 확인합니다.

교차 계정 배포 실패

- 서비스 계정의 자격 증명으로 `aws.service` 공급자를 구성해야 합니다. 명명된 프로필 또는 수임 역할 블록을 사용합니다.
- `agent_space_arn` 값이 파트 1 출력의 ARN과 일치하는지 확인합니다.

Terraform 리소스 유형을 찾을 수 없음

- `awscc` 공급자 버전 `~> 1.0` 이상이 있는지 확인합니다. `awscc_devopsagent_agent_space` 및 `awscc_devopsagent_association` 리소스에는 AWS Cloud Control 공급자가 필요합니다.

정리

모든 리소스를 제거하려면 2부를 배포한 경우 역순으로 삭제합니다.

```
./cleanup.sh
```

또는 수동으로 다음을 수행합니다.

```
terraform destroy
```

경고: 그러면 에이전트 공간과 모든 관련 데이터가 영구적으로 삭제됩니다. 계속하기 전에 중요한 정보를 백업했는지 확인합니다.

보안 고려 사항

- Terraform 구성은 `aidevops.amazonaws.com` 서비스 보안 주체만 해당 역할을 수임하도록 허용하는 신뢰 정책을 사용하여 IAM 역할을 생성합니다.

- 신뢰 정책에는 특정 AWS 계정 및 에이전트 공간 ARN에 대한 액세스를 제한하는 조건이 포함됩니다.
- 모든 정책은 최소 권한 원칙을 따릅니다. 조직의 보안 요구 사항에 따라 IAM 정책을 검토하고 사용자 지정합니다.
- 교차 계정 역할(`DevOpsAgentRole-SecondaryAccount-TF`)은 고정된 이름을 사용하며 특정 에이전트 공간 ARN으로 범위가 지정됩니다.

다음 단계

Terraform을 사용하여 AWS DevOps 에이전트를 배포한 후:

1. DevOps 에이전트 AWS 사용 설명서에서 DevOps 에이전트 기능의 전체 범위에 대해 알아봅니다.
2. 자동화된 인프라 관리를 위해 Terraform 배포를 CI/CD 파이프라인에 통합하는 것이 좋습니다.

추가 리소스

- AWS DevOps 에이전트 사용 설명서
- 샘플 Terraform 리포지토리
- CLI 온보딩 가이드

DevOps 에이전트 작업

DevOps 에이전트 작업

AWS DevOps Agent는 탐지부터 조사, 복구 및 예방에 이르기까지 전체 인시던트 수명 주기에서 운영 팀과 협력합니다. 다음 주제에서는 DevOps 에이전트를 사용하여이 수명 주기의 각 단계를 관리하는 방법을 설명합니다.

자율 인시던트 대응

티켓팅 시스템과의 기본 통합, 모니터링 도구의 웹후크 또는 수동 트리거를 통해 인시던트가 감지되면 DevOps Agent가 자동으로 조사를 시작합니다. 에이전트는 지표, 로그, 추적, 코드 변경 및 배포 기록을 분석하여 근본 원인을 파악하고 완화 계획을 제안합니다. 추가 도움이 필요한 경우 DevOps 에이전트 스페이스 웹 앱에서 AWS Support로 직접 에스컬레이션할 수 있습니다. DevOps 에이전트 스페이스 웹 앱은 지원 엔지니어와 조사 컨텍스트를 자동으로 공유하므로 에이전트가 이미 찾은 내용을 반복할 필요가 없습니다. 자세한 내용은 the section called “자율 인시던트 대응” 단원을 참조하십시오.

온디맨드 DevOps 작업

인시던트 수명 주기 중 언제든지 대화형 채팅 인터페이스를 통해 DevOps 에이전트와 상호 작용할 수 있습니다. 자연어를 사용하여 AWS 리소스, 시스템 상태, 경보 상태 및 배포 기록에 대해 질문합니다. 채팅은 컨텍스트 인식입니다. 특정 조사를 볼 때 에이전트가 특정 가설을 탐색하거나, 특정 로그에 집중하거나, 근본 원인 분석을 업데이트하도록 할 수 있습니다. 또한 콘솔 간에 이동하지 않고도 환경 전체에서 리소스 구성, 오류 추세 및 조사 인사이트를 쿼리할 수 있습니다. 자세한 내용은 the section called “온디맨드 DevOps 작업” 단원을 참조하십시오.

선제적 인시던트 방지

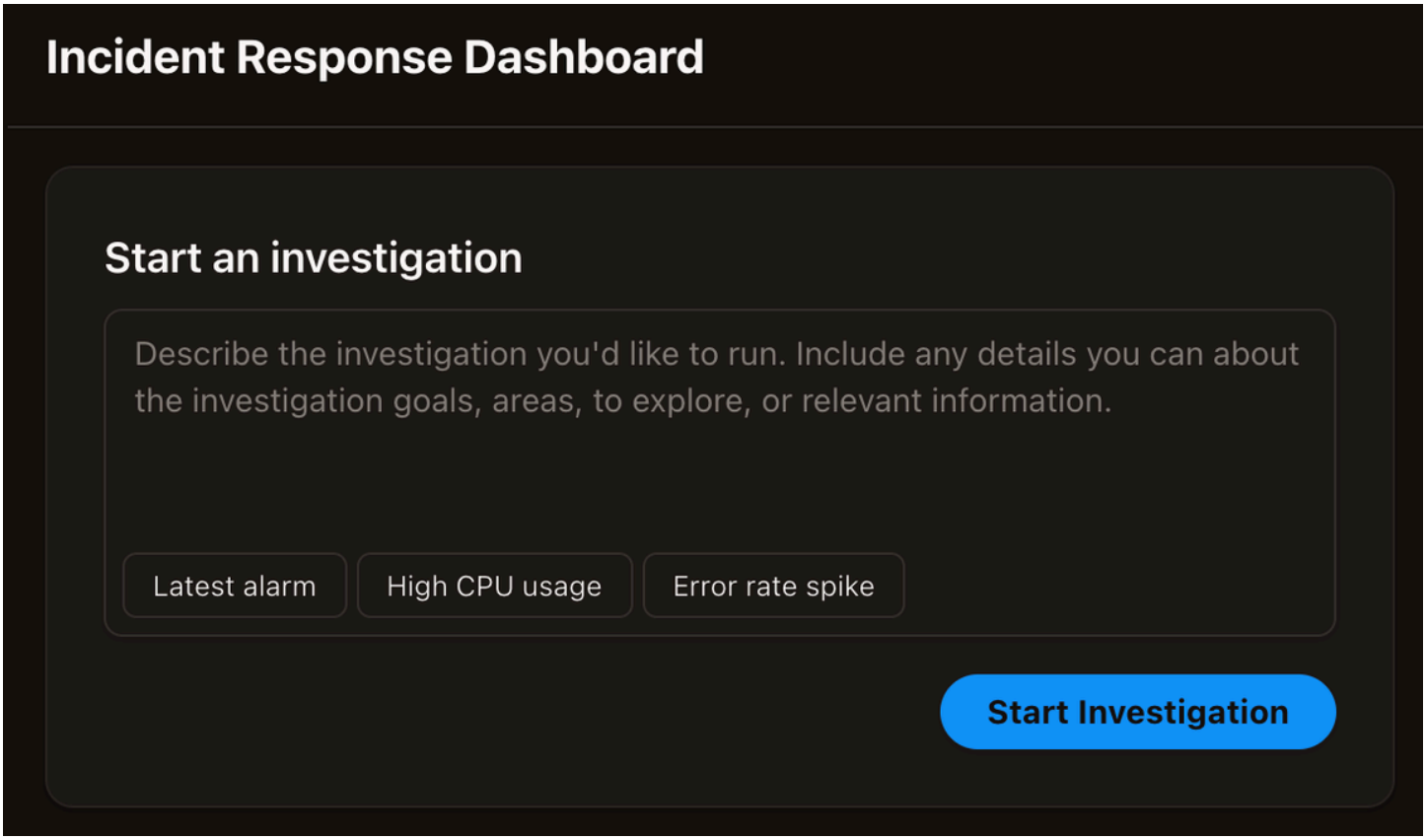
DevOps Agent는 인시던트를 해결한 후 조사 기록 전반의 패턴을 분석하여 향후 인시던트를 방지하고 평균 탐지 시간을 줄이는 권장 사항을 생성합니다. 권장 사항은 관찰성 태세, 테스트 격차, 코드 변경, 인프라 아키텍처라는 네 가지 영역에 걸쳐 있습니다. 에이전트는 매주 평가를 실행하고 새 인시던트가 발생하면 권장 사항을 업데이트합니다. 추천을 수락, 거부 또는 추적할 수 있으며 에이전트는 피드백에서 학습하여 향후 제안을 구체화합니다. 자세한 내용은 the section called “선제적 인시던트 예방” 단원을 참조하십시오.

자율 인시던트 대응

조사 시작

인시던트 대응 조사는 세 가지 방법 중 하나로 시작할 수 있습니다.

- 기본 제공 통합 - 기본 제공 통합을 사용하여 DevOps 에이전트 스페이스를 ServiceNow와 같은 티켓팅 시스템에 연결할 수 있습니다. 연결되면 지원 티켓에서 DevOps 에이전트 인시던트 대응 조사가 자동으로 트리거되고 DevOps 에이전트는 주요 조사 결과, 근본 원인 분석 및 완화 계획에 대한 업데이트를 원래 티켓에 제공합니다.
- Webhooks - Webhooks를 사용하여 이벤트를 AWS DevOps Agent로 보낼 수 있습니다. 예를 들어 웹후크를 사용하여 PagerDuty 티켓 또는 Grafana 경보에서 인시던트 대응 조사를 트리거할 수 있습니다.
- 수동 - DevOps 에이전트 스페이스 웹 앱의 인시던트 대응 탭에서 인시던트 대응 조사를 수동으로 시작할 수 있습니다. DevOps 에이전트가 조사할 인시던트를 설명하는 자유 형식 텍스트를 입력할 수 있습니다. 그러면 DevOps 에이전트가 조사 계획을 생성하고, 조사 결과를 수집하고, 근본 원인을 파악하고, 완화 계획을 생성하도록 제안합니다. 사전 구성된 여러 시작점 중에서 선택하여 조사를 빠르게 시작할 수도 있습니다. 가장 최근에 트리거된 경보를 조사하고 기본 지표 및 로그를 분석하여 근본 원인을 확인하는 최신 경보 높은 CPU 사용량을 통해 컴퓨팅 리소스에서 높은 CPU 사용률 지표를 조사하고 과도한 리소스를 소비하는 프로세스 또는 서비스를 식별합니다. 또는 지표를 분석하여 애플리케이션 오류율의 최근 증가를 조사하기 위한 오류율 스파이크, 애플리케이션 로그, 및 장애의 원인을 식별합니다.



"조사 시작"을 클릭하면 에이전트가 작업에 집중할 수 있도록 몇 가지 추가 세부 정보를 제공하라는 메시지가 표시됩니다. 조사 대화 상자에는 다음 필드가 포함됩니다.

- 조사 세부 정보 - 설명으로 미리 채워집니다. 이를 편집하여 조사 범위를 세분화할 수 있습니다.
- 조사 시작점 - 선택적으로 에이전트에 대한 특정 경보, 지표, 로그 조각 또는 기타 시작점을 설명합니다.
- 인시던트 날짜 및 시간 - UTC 형식으로 현재 시간으로 자동 채워집니다. 인시던트가 이전에 발생했는지 조정합니다.
- 조사 이름 지정 - 타임스탬프로 자동 생성됩니다. 이를 사용자 지정할 수 있습니다(최대 400자).
- 우선순위 - 드롭다운에서 조사 우선순위를 선택합니다(중간이 기본값임).

필요에 따라 이러한 필드를 검토하고 조정한 다음 "조사 시작..."을 클릭하여 시작합니다. 그러면 DevOps 에이전트가 실행 중인 것을 볼 수 있는 조사 세부 정보 페이지로 이동합니다.

인시던트 분류

분류 단계는 AWS DevOps 에이전트 인시던트 대응 시스템의 첫 번째 단계입니다. Datadog의 경보, ServiceNow의 인시던트 티켓 또는 Dynatrace의 문제와 같은 외부 이벤트가 트리거되면 AWS DevOps

에이전트는 몇 초 내에 이를 자동으로 처리하여 독립적으로 조사해야 하는지 아니면 기존 조사와 연결해야 하는지 결정합니다.

분류 단계의 주요 기능은 인시던트 상관관계입니다. 즉, 관련 인시던트를 식별하고 이를 단일 조사로 통합하여 중복 작업 및 리소스 낭비를 방지하는 것입니다. 새 인시던트가 도착하면 AWS DevOps Agent는 룩백 기간(일반적으로 20분) 내에 활성 조사와 함께 이를 분석합니다. AI 기반 분석을 사용하여 구성 요소 유사성, 지리적 리전 및 타이밍 패턴과 같은 요소를 검사하여 인시던트 간의 관계를 결정합니다.

AWS DevOps Agent는 다음 두 가지 결정 중 하나를 수행합니다.

- 연결됨 - 인시던트를 기존 조사와 연관시키고 새 인시던트에 대한 컨텍스트와 함께 해당 조사에 조향 메시지를 보냅니다.
- 진행 - 인시던트에 대한 새로운 독립 조사를 예약합니다.

분류 결정 보기

인시던트가 연결되면 기본 조사는 연결된 인시던트의 세부 정보 및 상관관계 추론이 포함된 조향 메시지를 수신합니다. AWS DevOps 에이전트 스페이스 웹 앱에서 인시던트가 연결된 이유를 설명하는 상관관계 추론과 함께 LINKED 상태가 표시됩니다. 기본 조사에는 연결된 모든 인시던트 목록이 표시되므로 함께 조사 중인 관련 문제의 전체 범위를 볼 수 있습니다. 외부 티켓 시스템(ServiceNow, PagerDuty 등) 및 통신 채널(Slack)은 인시던트가 상관관계 추론과 함께 연결되었다는 알림을 받게 됩니다.

인시던트 및 사용자 지정 상관관계 규칙 연결 해제

만약 AWS DevOps 에이전트가 인시던트의 상관관계를 잘못 파악한 경우 AWS , DevOps 에이전트 스페이스 웹 앱을 통해 수동으로 연결을 해제할 수 있습니다. 이렇게 하면 연결되지 않은 인시던트가 독립적인 조사로 다시 예약됩니다. 상관관계 로직이 포함된 AWS DevOps 에이전트 스킬을 생성하고 이를 분류 단계와 연결하여 Guide AWS DevOps 에이전트에 사용자 지정 상관관계 규칙을 제공할 수도 있습니다.

인적 지원 요청

AWS DevOps Agent는 AWS Support와 직접 연결하여 인시던트 대응 프로세스를 간소화할 수 있습니다. AWS Support의 추가 도움이 필요한 경우 DevOps Agent Space 웹 앱에서 지원 엔지니어와 조사 컨텍스트를 자동으로 공유하는 AWS 지원 사례를 생성하여 문제를 설명하는 데 필요한 시간을 줄일 수 있습니다.

작동 방식

인시던트를 조사할 때 AWS DevOps Agent는 다음을 포함하여 포괄적인 분석 로그를 작성합니다.

- 근본 원인 조사 결과
- 분석된 지표, 로그 및 추적
- 코드 변경 및 배포 기록 검토
- 권장 문제 해결 작업
- 이벤트 및 시스템 동작의 타임라인

AWS DevOps 에이전트 스페이스 웹 앱에서 직접 AWS Support로 조사를 에스컬레이션할 수 있습니다. 이렇게 하면 AWS DevOps 에이전트는 자동으로 조사 로그를 AWS Support에 전달하여 세부 정보를 수동으로 수집하고 설명할 필요 없이 지원 엔지니어에게 조사에 대한 전체 컨텍스트를 제공합니다.

AWS Support와 채팅

지원 사례를 생성하면 AWS DevOps 에이전트 스페이스 웹 앱 내의 별도의 채팅 창에서 AWS Support와 통신할 수 있습니다. 다음 작업을 수행할 수 있습니다.

- AWS Support 엔지니어와 함께 AWS DevOps 에이전트의 조사 타임라인과 함께 문제에 대해 논의합니다.
- 동일한 인터페이스에서 AWS DevOps 에이전트의 자동 분석과 AWS Support의 전문가 지침 모두 보기
- 필요에 따라 추가 정보 또는 설명을 원활하게 공유

채팅 경험을 통해 AWS DevOps 에이전트 조사 및 AWS 지원 대화에 쉽게 액세스할 수 있으므로 협업 및 해결 속도가 빨라집니다.

지원 계획 요구 사항

AWS DevOps 에이전트를 통해 지원 사례를 생성하고 상호 작용하는 기능은 AWS 지원 계획에 따라 다릅니다. 권한에 대한 자세한 내용은 지원 플랜 사용 설명서를 참조하세요.

참고 기본 지원 고객은 기술 지원 사례를 생성할 수 없으므로 AWS 지원 개발자 지원에 대한 AWS DevOps 에이전트 조사를 에스컬레이션할 수 없습니다. 고객은 AWS DevOps 에이전트를 통해 사례를 생성할 수 있지만, AWS 개발자 지원에는 채팅 기반 지원이 포함되지 않으므로 지원 엔지니어와 대응하기 위해 지원 센터를 방문해야 합니다. 다른 모든 플랜은 AWS DevOps 에이전트 내에서 통합 채팅

경험을 사용할 수 있습니다. 응답 시간 및 사용 가능한 사례 심각도를 포함하여 지원 플랜 권한에 대한 전체 세부 정보는 AWS 지원 플랜 사용 설명서를 참조하세요.

AWS Support와 공유되는 정보

AWS DevOps Agent Space 웹 앱에서 지원 사례를 생성하면 다음 정보가 AWS Support와 자동으로 공유됩니다.

- 조사 타임라인: AWS DevOps 에이전트 분석의 연대 기록
- 리소스 정보: 영향을 받는 AWS 리소스
- 관찰성 데이터: 통합 모니터링 도구의 관련 지표, 로그 및 추적
- 최근 변경 사항: 코드 배포, 인프라 변경 및 구성 업데이트
- 해결 시도: Actions AWS DevOps 에이전트 권장
- 영향 평가: 인시던트의 범위 및 심각도

AWS Support와 공유되는 모든 데이터는 기존 AWS 데이터 레지던시 및 보안 구성을 따릅니다. AWS DevOps 에이전트는 특정 조사와 관련된 정보만 공유하고 조직의 데이터 거버넌스 정책을 준수합니다.

시작하기

AWS DevOps 에이전트의 AWS 지원 통합을 사용하려면:

1. 활성 AWS 지원 플랜이 있는지 확인합니다.
2. your AWS DevOps 에이전트의 IAM 권한에 지원 사례 생성(support:CreateCase, support:DescribeCases)이 포함되어 있는지 확인합니다.
3. AWS DevOps 에이전트가 문제를 조사 중이고 AWS 지원 지원이 필요한 경우 DevOps 에이전트 스페이스 웹 앱에서 인간 지원 요청을 선택합니다.
4. AWS Support와 공유될 조사 요약을 검토합니다.
5. 지원 플랜 권한에 따라 적절한 사례 심각도를 선택합니다.
6. 사례 제출 - AWS DevOps 에이전트에 조사 로그가 자동으로 포함됩니다.

채팅 창이 자동으로 열리므로 AWS Support와 즉시 협업을 시작할 수 있습니다.

선제적 인시던트 예방

AWS DevOps Agent는 인시던트 조사 전반의 패턴을 분석하여 운영 태세를 지속적으로 개선하고 향후 인시던트를 방지하는 대상 추천을 제공합니다. 운영자 웹 앱의 Ops 백로그 페이지를 통해 선제적 인시던트 예방에 액세스합니다.

선제적 인시던트 예방 작동 방식

AWS DevOps Agent는 최근 인시던트 조사를 평가하여 지속적인 개선 사항을 식별하여 향후 인시던트를 방지하고 평균 탐지 시간(MTTD)을 단축합니다. 에이전트는 여러 인시던트를 분석하여 향후 전체 인시던트 클래스를 방지할 수 있는 권장 사항을 식별하며, 가장 영향력 있는 권장 사항에 집중하여 조치를 취할 수 있도록 합니다.

기본적으로 에이전트는 매주 평가를 자동으로 실행합니다. 온디맨드 방식으로만 평가를 실행하려면 일정을 일시 중지할 수 있습니다. 수동 평가는 항상 사용할 수 있으며, 이는 최근 조사를 통해 권장 개선 사항을 신속하게 해결해야 할 때 유용합니다.

에이전트는 Ops 백로그 페이지의 권장 범주화 차트에 표시된 네 가지 범주의 개선 사항을 식별합니다.

- 관찰성 - 모니터링, 알림, 로깅 및 시스템 가시성을 향상하여 문제를 더 빠르고 정확하게 감지하는 것이 좋습니다.
- 인프라 - 리소스 구성, 용량 튜닝 및 아키텍처 복원력을 최적화하기 위한 권장 사항입니다.
- 거버넌스 - 배포 프로세스, 파이프라인 개선, 테스트 관행 및 운영 제어를 강화하기 위한 권장 사항입니다.
- 코드 최적화 - 애플리케이션 코드 품질, 오류 처리 및 코드 복원력을 개선하기 위한 권장 사항입니다.

이 분류를 통해 운영 개선이 가장 필요한 부분을 파악하고 팀의 중점 영역에 따라 권장 사항의 우선순위를 지정할 수 있습니다.

이점

- 반복되는 인시던트 방지 - 동일한 유형의 문제에 반복적으로 대응하지 않고 근본 원인을 체계적으로 해결합니다.
- 운영 위험 감소 - 팀이 반복적인 소방에서 벗어나 혁신과 전략적 개선에 집중할 수 있습니다.
- 시스템 복원력 개선 - 실제 인시던트 데이터를 기반으로 인프라, 관찰성 및 배포 프로세스 강화
- 과거 패턴에서 알아보기 - 과거 인시던트의 인사이트를 활용하여 가장 큰 영향을 미치는 목표 개선

에이전트 요약

웹 앱의 운영 백로그 페이지에 있는 에이전트 요약은 최근 인시던트의 마지막 평가 결과에 대한 설명을 제공합니다. 요약에서는 분석된 인시던트 조사 수, 과거 인시던트와 유사한 인시던트, 새로운 정보로 생성되거나 업데이트된 권장 사항을 설명합니다.

요약은 에이전트가 가장 최근 평가 중에 발견한 내용을 빠르게 이해하는 데 도움이 되며 운영 태세에 가장 큰 영향을 미칠 수 있는 가장 주목할 만한 권장 사항을 강조합니다.

평가 제어

AWS DevOps 에이전트가 인시던트를 평가하고 권장 사항을 생성하는 시기를 제어할 수 있습니다.

- 수동으로 평가 실행 - Ops 백로그 페이지에서 지금 실행 버튼을 클릭하여 평가를 즉시 시작합니다. 이는 최근 조사에서 권장 개선 사항을 신속하게 처리해야 하는 경우에 유용합니다.
- 활성 평가 중지 - 운영 백로그 페이지에서 평가 중지 버튼을 클릭하여 현재 진행 중인 평가를 중지합니다.

권장 사항 관리

AWS DevOps Agent는 Ops 백로그 페이지에서 권장 사항을 검토하고 관리할 수 있는 권장 사항을 제공합니다.

- 권장 사항 세부 정보 보기 - 권장 사항을 클릭하여 권장 사항 세부 정보 페이지를 엽니다. 권장 사항에 영향을 준 인시던트, 예상되는 영향 및 다음 단계를 포함하여 제안된 개선 사항에 대한 자세한 정보를 볼 수 있습니다. 코드 변경에 대한 권장 사항의 경우 구현을 위해 코딩 에이전트에 전달할 수 있는 에이전트 지원 사양을 볼 수도 있습니다.
- 유지 - 추적을 위해 백로그에 권장 사항을 유지하려면 '보관'을 클릭합니다. 이를 통해 구현하려는 개선 사항을 모니터링하고 진행 상황을 추적할 수 있습니다.
- 폐기 - 'Discard'를 클릭하여 백로그에서 권장 사항을 제거합니다. 권장 사항을 삭제하면 요구 사항을 충족하지 않는 이유에 대한 자연어 설명을 제공할 수 있습니다. 에이전트는이 피드백을 통해 학습하고 이를 사용하여 향후 권장 사항을 알려 시간이 지남에 따라 운영 우선순위 및 요구 사항에 더 부합하도록 합니다.
- 구현됨 - '구현됨'을 클릭하여 권장 사항을 완료됨으로 표시합니다. 이를 통해 적용된 개선 사항을 추적하고 에이전트가 시간 경과에 따른 권장 사항의 효과를 측정할 수 있습니다.

- 자동 제거 - 유지 또는 구현으로 표시되지 않은 권장 사항은 권장 사항을 구현하여 새 인시던트가 방지되지 않은 경우 약 6주 후에 제거할 수 있습니다. 이렇게 하면 운영 백로그 페이지가 운영 문제에 가장 관련성이 높은 개선 사항에 초점을 맞출 수 있습니다.
- 권장 사항 업데이트 - 기존 권장 사항은 권장 사항으로 인해 방지되었을 새 인시던트가 발견되면 업데이트됩니다. 업데이트는 권장 사항의 우선 순위를 변경하거나 새로운 인사이트를 기반으로 권장 사항을 구체화할 수 있습니다.

에이전트 지원 사양

코드 또는 구성 변경과 관련된 권장 사항의 경우 AWS DevOps 에이전트는 에이전트 지원 사양을 생성할 수 있습니다. 이 사양은 구현을 위해 코딩 에이전트에 직접 전달할 수 있는 구조화된 문서를 제공합니다.

사양에는 다음이 포함됩니다.

- 문제 설명 - 문제 및 문제의 근본 원인 요약
- 솔루션 요약 - 권장 접근 방식에 대한 개략적인 설명
- 대상 리포지토리 - 변경해야 하는 특정 리포지토리
- 코드 변경 - 특정 파일 경로 및 구현 고려 사항과 함께 변경해야 하는 사항 및 이유에 대한 자세한 설명
- 테스트 요구 사항 - 테스트해야 하는 시나리오
- 구현 계획 - 변경 사항을 구현하기 위한 단계별 접근 방식

에이전트 지원 사양은 코딩 에이전트가 엔지니어와 광범위한 back-and-forth을 가속화합니다.

권장 사항 구현

선제적 인시던트 예방 권장 사항의 가치를 극대화하려면 다음과 같은 조치를 취하는 방법을 고려하세요.

- 에이전트 지원 사양 사용 - 코드 변경이 있는 권장 사항의 경우 생성된 사양을 사용하여 코딩 에이전트에 전달하거나 수동 구현을 위한 자세한 가이드로 사용하여 구현을 가속화합니다.
- 티켓 백로그에 권장 사항 추가 - 팀의 티켓팅 시스템 또는 프로젝트 관리 도구에 권장 사항을 복사하여 다른 엔지니어링 작업과 함께 우선 순위를 지정합니다.
- 영향을 기반으로 권장 사항 우선 순위 지정 - 가장 빈번하거나 심각한 인시던트 유형 또는 중요한 시스템에 영향을 미치는 권장 사항에 먼저 집중합니다.

- 구현 진행 상황 추적 - 구현된 권장 사항을 모니터링하고 시간이 지남에 따라 유사한 인시던트가 감소하는지 관찰하여 효과를 측정합니다.
- 개발 팀과 조정 - 영향을 받는 시스템을 소유한 적절한 팀과 권장 사항을 공유하여 개선 사항을 구현하는 데 필요한 컨텍스트와 리소스를 확보합니다.

온디맨드 DevOps 작업

AWS DevOps Agent On Demand Tasks는 운영 팀이 애플리케이션 아키텍처를 쿼리하고, 시스템 상태를 분석하고, 자연어를 사용하여 조사 인사이트에 액세스할 수 있는 생성형 인공 지능(AI) 기반 대화형 어시스턴트입니다. AWS 리소스, 시스템 지표, 경보 상태, 배포 기록 및 인시던트 패턴에 대해 질문할 수 있습니다. Chat은 실제 인프라 및 운영 데이터에 기반한 즉각적인 답변을 제공하므로 여러 AWS 콘솔 또는 모니터링 도구를 탐색할 필요가 없습니다.

채팅은 DevOps Agent Space 웹 앱 전체에 통합되며 보고 있는 페이지를 기반으로 컨텍스트 인식 응답을 제공합니다. 인터페이스는 대화 기록을 유지 관리하므로 이전 논의를 계속하고 이전 쿼리를 기반으로 구축할 수 있습니다.

작업 기능

AWS DevOps Agent On Demand Tasks는 인프라를 관리하고 이해하는 데 도움이 되는 포괄적인 기능을 제공합니다.

리소스 쿼리 - Lambda 함수, DynamoDB 테이블, EKS 배포, 인증서 및 인프라 구성을 포함하여 에이전트 스페이스의 AWS 리소스에 대해 질문합니다. 채팅은 런타임 버전, 용량 설정 또는 배포 상태와 같은 속성을 기반으로 리소스를 필터링하고 분석할 수 있습니다. 예를 들어 "Python 3.8을 사용하는 Lambda는 몇 개입니까?"라고 질문합니다. 또는 "만료될 인증서가 있습니까?"

시스템 상태 분석 - 경보 상태, 오류율, CPU 사용률 및 서비스 가용성을 포함하여 현재 및 과거 시스템 상태 지표를 쿼리합니다. 채팅은 특정 기간을 다루는 상태 요약을 생성하고 시스템 동작의 추세를 식별할 수 있습니다. “지난 24시간 동안 어떤 경보가 발생했나요?”와 같은 질문을 합니다. 또는 "지난 1시간 동안 5xx 오류가 있습니까?"

조사 인사이트 - 근본 원인 분석, 탐색된 가설, 검토된 로그, 해결 패턴을 포함하여 완료 및 진행 중인 조사의 정보에 액세스합니다. 채팅은 일반적인 인시던트 원인을 식별하고 기록 데이터를 기반으로 권장 사항을 제공할 수 있습니다. "지난 달 인시던트의 가장 일반적인 원인은 무엇입니까?" 쿼리 또는 "완료된 조사의 평균 해결 시간은 얼마입니까?"

조사 조향 - 조사 세부 정보 페이지를 볼 때 에이전트에게 특정 로그에 집중하거나, 특정 가설을 탐색하거나, 근본 원인 분석을 업데이트하도록 지시하여 조사를 안내합니다. "결제 서비스에 대한 로그에 집

중하고 RCA 업데이트" 또는 "DynamoDB 제한으로 인해 문제가 발생했다는 가설 탐색"과 같은 조향 입력을 제공합니다.

채팅 아티팩트 - 운영 상태 요약, 오류 보고서 및 인시던트 분석과 같은 구조화된 보고서 및 문서를 생성합니다. 아티팩트는 전용 패널에 나타나며 대화 내에서 버전이 지정된 편집을 지원합니다.

권장 사항 필터링 - 특정 서비스 또는 운영 문제와 관련된 권장 사항과 같은 특정 기준으로 인시던트 방지 권장 사항을 쿼리합니다. Chat은 각 권장 사항에 대한 영향 및 구현 고려 사항을 설명합니다. 예를 들어 "DynamoDB와 관련된 인시던트를 방지하는 권장 사항 표시" 또는 "요청 지연 시간 문제를 더 빠르게 감지하는 데 도움이 되는 권장 사항은 무엇입니까?"

채팅 액세스

채팅은 DevOps 에이전트 스페이스 웹 앱의 왼쪽에 영구 패널로 제공됩니다. 왼쪽 사이드바에는 + 새 채팅 버튼, 인시던트, 운영 백로그 및 토폴로지로 이동하기 위한 페이지 섹션, 최근 대화를 표시하는 채팅 섹션이 포함되어 있습니다. 전체 대화 기록을 보려면 모두 보기를 선택합니다.

Chat은 액세스하는 위치에 따라 컨텍스트 인식 응답을 제공합니다.

토폴로지 - 에이전트 스페이스 리소스, 아키텍처 및 운영 상태에 대한 일반적인 질문을 합니다. 채팅은 연결된 모든 계정과 서비스를 완벽하게 볼 수 있습니다. 이 컨텍스트에서 리소스 구성, 배포 기록, 토폴로지 정보 및 관찰성 도구 통합을 쿼리할 수 있습니다.

인시던트 대응 - 인시던트 대응 페이지를 볼 때 에이전트 스페이스 전반의 조사 추세, 해결 시간 및 인시던트 패턴에 대해 질문합니다. 채팅은 과거 조사 데이터를 분석하여 일반적인 원인과 개선 기회를 식별할 수 있습니다.

조사 세부 정보 - 특정 조사를 보는 동안 Chat은 해당 조사에 대한 컨텍스트 인식 응답을 제공합니다. 검토된 로그, 탐색된 가설, 근본 원인 결론 및 완화 계획에 대해 질문합니다. 또한 조향 입력을 제공하여 조사 포커스를 안내할 수 있습니다.

예방 - 예방 페이지에서 필터로 권장 사항을 쿼리하고, 권장 사항이 작성된 이유를 이해하고, 구현 접근 방식을 살펴봅니다. 채팅은 인시던트 방지 권장 사항의 영향을 우선시하고 이해하는 데 도움이 됩니다.

페이지 간에 전환할 때 채팅 인터페이스는 계속 사용할 수 있지만 현재 보기와 관련된 정보를 제공하도록 컨텍스트가 변경됩니다. 새 대화를 시작하면 이전 컨텍스트 없이 시작됩니다. 기존 대화를 계속하면 Chat은 후속 질문에 대한 전체 대화 기록을 유지합니다.

컨텍스트 인식 응답

Chat은 DevOps 에이전트 스페이스 웹 앱에서 보고 있는 페이지를 기반으로 응답을 조정합니다. 이러한 컨텍스트 인식을 통해 요청하려는 조사 또는 리소스 범위를 지정할 필요 없이 관련 정보를 받을 수 있습니다.

조사 세부 정보 페이지를 볼 때 Chat은 사용자가 특정 조사에 대해 질문하고 있음을 자동으로 이해합니다. “어떤 로그를 보셨나요?”와 같은 질문 또는 "어떤 가설을 탐색했나요?" 는 현재 표시된 조사를 참조합니다. 조향 입력을 제공하면 Chat은 이를 활성 조사에 적용하고 적절한 경우 새 근본 원인 버전을 생성합니다.

예방 페이지에서 Chat은 인시던트 예방 권장 사항에 관심이 있음을 이해합니다. 쿼리는 에이전트 스페이스 컨텍스트 내에서 추천을 자동으로 필터링하고 분석합니다. 시스템은 일반 권장 사항 또는 특정 권장 사항 세부 정보에 대해 질문하는지 여부를 인식합니다.

토폴로지 페이지에서 채팅에 액세스할 때 Chat은 에이전트 스페이스의 모든 리소스, 지표 및 기록 데이터에 대한 광범위한 가시성을 제공합니다. 조사 또는 권장 사항 컨텍스트를 지정하지 않고도 리소스, 서비스 또는 운영 문제에 대해 질문할 수 있습니다.

이러한 컨텍스트 인식을 사용하면 참조하는 조사, 권장 사항 또는 리소스 범위를 반복적으로 지정할 필요가 없으므로 보다 자연스러운 대화 흐름이 가능합니다.

대화 관리

채팅은 이전 토론을 계속하고 이전 쿼리를 참조할 수 있도록 대화 기록을 유지합니다.

새 대화 생성 - 채팅 패널에서 "새 세션" 버튼을 클릭하여 이전 컨텍스트 없이 새 대화를 시작합니다. 새 대화는 이전 채팅의 정보를 전달하지 않으므로 혼동 없이 관련 없는 질문을 할 수 있습니다.

대화 기록 액세스 - 에이전트 스페이스 내의 모든 이전 대화를 보려면 "기록"을 클릭합니다. 대화는 타임스탬프와 미리 보기 텍스트로 시간순으로 구성됩니다. 대화 기록은 90일 동안 유지되며 에이전트 스페이스 내의 사용자 계정에 비공개입니다.

대화 계속 - 기록에서 대화를 선택하여 중단한 부분에서 다시 시작합니다. Chat은 이전 메시지의 전체 컨텍스트를 유지하므로 대화의 이전 부분을 참조하는 후속 질문을 할 수 있습니다. 대화를 보는 동안 페이지를 전환하면 대화 컨텍스트는 그대로 유지되지만 현재 위치에 따라 페이지별 컨텍스트가 업데이트됩니다.

대화 기록은 각 에이전트 스페이스 내에서 격리됩니다. 한 에이전트 스페이스의 대화는 다른 에이전트 스페이스에서 보거나 액세스할 수 없습니다. 이 격리를 통해 민감한 정보를 조직의 경계에 따라 구분된 상태로 유지할 수 있습니다.

아티팩트 생성

AWS DevOps Agent는 대화 중에 에이전트가 생성한 구조화되고 버전이 지정된 문서인 채팅 아티팩트를 지원합니다. 아티팩트는 운영 보고서, 오류 요약 및 상태 평가와 같은 AI 생성 콘텐츠를 검토하고 편집하기 위한 전용 대화형 패널을 채팅 UI에 제공합니다.

DevOps 에이전트 스페이스 웹 앱의 모든 페이지에서 아티팩트를 요청할 수 있습니다. Chat은 현재 페이지 컨텍스트를 사용하여 아티팩트 콘텐츠의 범위를 지정합니다.

아티팩트 작동 방식

Chat에 콘텐츠를 생성하거나 업데이트하도록 요청하면 Chat은 일반적으로 형식이 지정된 문서인 아티팩트를 생성하여 대화와 함께 아티팩트 패널에 표시합니다.

생성 - 자연어 요청을 보내 보고서 또는 문서를 생성합니다. 예를 들어 '에이전트 스페이스에 대한 주간 운영 상태 보고서 생성' 또는 '지난 주의 4xx 오류에 대한 보고서 표시'를 요청합니다.

검토 - 아티팩트가 대화와 함께 전용 패널에 나타납니다. 채팅과 계속 상호 작용하면서 전체 콘텐츠를 검토할 수 있습니다.

편집 - 채팅을 통해 아티팩트에 대한 변경 사항을 요청합니다. 예를 들어 "Lambda 콜드 스타트에 대한 섹션 추가" 또는 "지난 달의 데이터를 포함하도록 보고서 업데이트"를 요청합니다. Chat은 요청된 변경 사항을 사용하여 아티팩트의 새 버전을 생성합니다.

샘플 쿼리

다음 예제에서는 채팅에 질문할 수 있는 질문 유형을 보여줍니다. 이 예제는 사용 사례 및 컨텍스트별로 구성되어 있습니다.

아티팩트 생성 쿼리

DevOps 에이전트 스페이스 웹 앱의 모든 페이지에서:

- 내 에이전트 스페이스에 대한 주간 운영 상태 요약 생성
- 지난 주의 모든 4xx 오류에 대한 보고서 생성
- 지난 30일 동안의 인시던트 요약 보고서 작성
- 이번 주에 결제 서비스에 대한 경보 활동 요약 생성
- 지난 7일 동안의 배포 기록 보고서 생성

- 열려 있는 모든 권장 사항을 보고서로 요약

리소스 정보 쿼리

DevOps 에이전트 스페이스 웹 앱의 모든 페이지에서:

- Python 3.8을 사용하는 Lambda 함수는 몇 개입니까?
- 만료될 인증서가 있나요?
- 온디맨드 결제로 모든 DynamoDB 테이블 나열
- 프로덕션 환경에서 EKS 클러스터 표시
- 지난 90일 동안 배포되지 않은 Lambda 함수는 무엇입니까?
- 버전 관리가 활성화되지 않은 S3 버킷 나열
- 데이터베이스 버전 X를 실행하는 RDS 인스턴스는 무엇입니까?

시스템 상태 쿼리

토폴로지 또는 인시던트 대응 페이지에서:

- 지난 24시간 동안 발생한 경보는 무엇입니까?
- 지난 한 시간 동안 5xx 오류가 있었나요?
- 결제 서비스에 대한 Lambda 오류 추세 표시
- ECS 클러스터의 CPU 사용률은 얼마입니까?
- 로드 밸런서에 비정상 대상이 있나요?
- 어제의 API Gateway 제한 이벤트 표시
- 지난 주에 오류율이 가장 높은 서비스는 무엇입니까?
- 지난 24시간 동안의 전체 상태 보고서 제공

관찰성 도구 쿼리

토폴로지에서:

- Splunk 로그 그룹 나열
- Prometheus 지표 및 경보 임계값 표시

- 이 서비스에 대해 구성된 Datadog 모니터는 무엇입니까?
- New Relic 알림 정책 나열
- Dynatrace 대시보드 구성 표시

조사 인사이트 쿼리

인시던트 대응 페이지에서:

- 지난 달 인시던트의 가장 일반적인 원인은 무엇입니까?
- 완료된 조사의 평균 해결 시간은 얼마입니까?
- 지난 주의 조사 및 해당 RCA 요약
- DynamoDB 제한으로 인해 발생한 인시던트 수는 몇 개입니까?
- 지난 분기의 조사 추세 표시
- 인시던트가 가장 자주 발생하는 서비스는 무엇입니까?

조사 세부 정보 쿼리

조사 세부 정보 페이지에서:

- 어떤 로그를 보셨나요?
- 어떤 가설을 탐색했나요?
- 제안하는 완화 조치는 얼마나 위험합니까?
- 이 인시던트의 이벤트 타임라인은 어떻게 되었나요?
- 이것이 근본 원인이라는 결론을 내린 이유는 무엇입니까?
- 근본 원인 분석을 지원하는 증거는 무엇입니까?
- 조사 중에 누가 조향을 제공했습니까?
- 이 인시던트 조사에 대한 요약 제공

조사 조향 쿼리

조사 세부 정보 페이지에서:

- 14:00~15:00 UTC 사이에 결제 서비스에 대한 로그에 집중하고 RCA를 업데이트합니다.
- DynamoDB 제한으로 인해 문제가 발생했다는 가설 탐색

- ECS 클러스터 구성을 확인하여 이로 인해 경보가 발생했는지 확인합니다.
- 하루가 아닌 지난 2시간 동안의 로그만 확인
- 오후 3시에 오류 급증 조사
- Lambda 로그 대신 API Gateway 로그 보기

예방 권장 사항 쿼리

예방 페이지에서:

- 상위 3가지 인시던트 예방 권장 사항은 무엇인가요?
- DynamoDB와 관련된 인시던트를 방지하는 권장 사항 표시
- 요청 지연 시간 문제를 더 빠르게 감지하는 데 도움이 되는 권장 사항은 무엇입니까?
- 유사한 인시던트를 방지할 수 있는 관찰성 개선 사항 나열
- 결제 서비스에 대한 인프라 권장 사항 표시
- 시스템 복원력에 가장 큰 영향을 미치는 권장 사항은 무엇입니까?

에이전트 스페이스에서 채팅 활성화

채팅은 모든 DevOps 에이전트 스페이스 웹 앱에서 사용할 수 있습니다. 설정 프로세스는 새 에이전트 스페이스가 있는지 기존 에이전트 스페이스가 있는지에 따라 달라집니다.

새 에이전트 공간

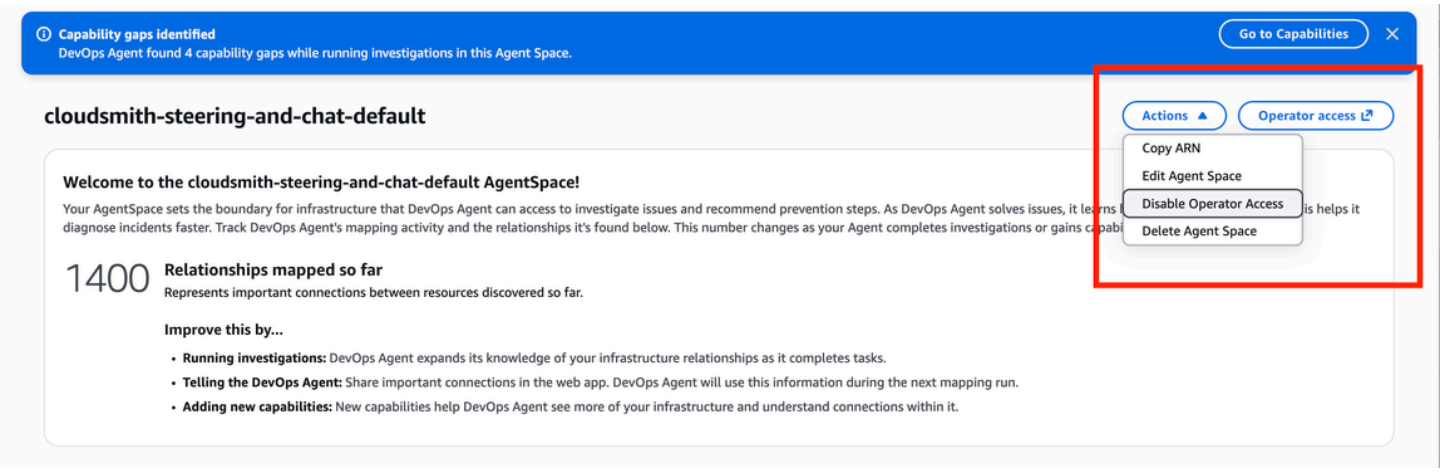
채팅은 새 에이전트 스페이스를 생성할 때 자동으로 활성화됩니다. 추가 구성 또는 IAM 권한 설정은 필요하지 않습니다. DevOps 에이전트 스페이스 웹 앱을 구성한 후에는 페이지 왼쪽에 영구 패널로 채팅을 즉시 사용할 수 있습니다.

기존 에이전트 공간

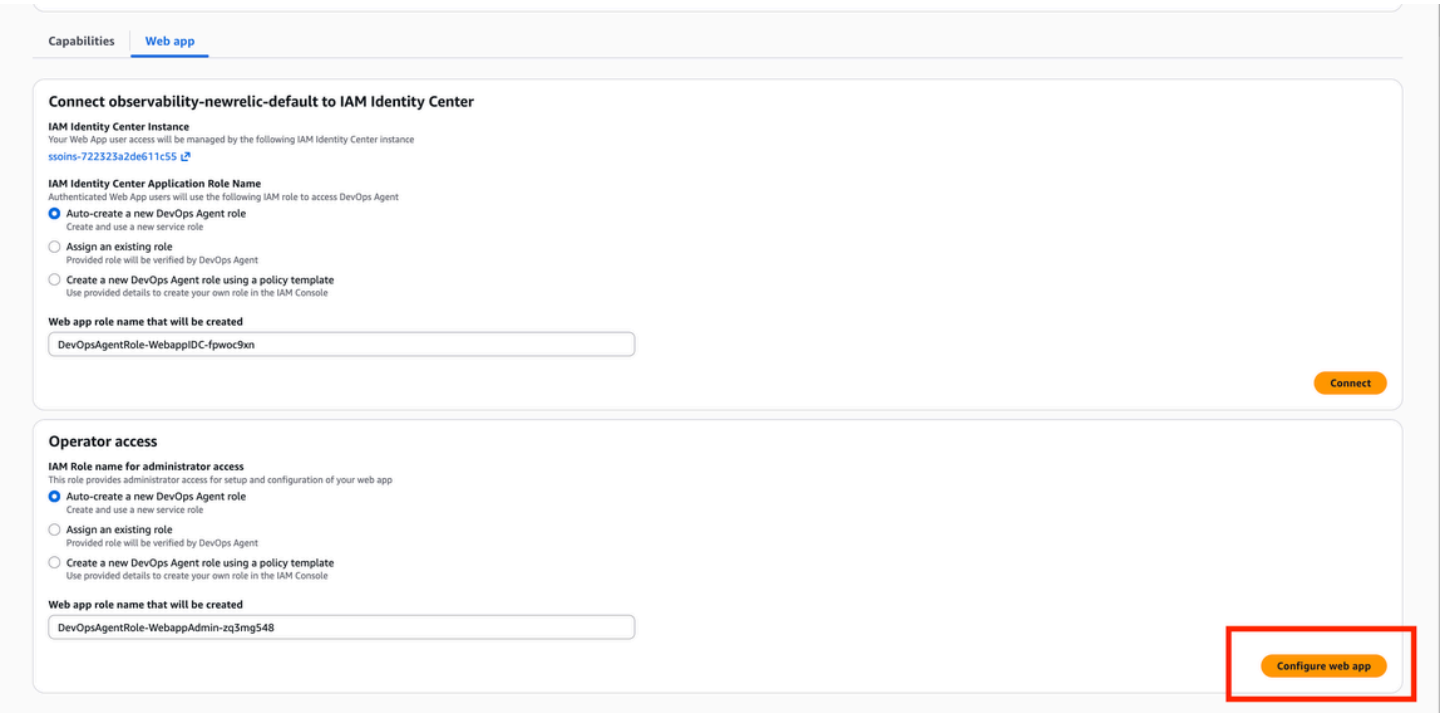
채팅이 릴리스되기 전에 에이전트 스페이스를 생성한 경우 필요한 IAM 권한을 활성화해야 합니다. 여기에는 두 가지 옵션이 있습니다.

옵션 1: 운영자 앱 액세스 취소 및 다시 활성화

AWS DevOps 에이전트 관리 콘솔로 이동하여 오른쪽 상단 모서리에서 작업 드롭다운을 찾아 현재 운영자 액세스 구성을 비활성화합니다.



그런 다음 운영자 액세스를 위한 자동 생성 옵션을 활성화합니다.



이렇게 하면 채팅에 필요한 IAM 권한이 다른 모든 현재 운영자 권한과 함께 자동으로 적용됩니다.

옵션 2: 수동으로 IAM 권한 추가

기존 운영자 액세스 역할에 다음 IAM 권한을 추가합니다.

- `aidevops:ListChats` - 채팅 대화 기록 보기
- `aidevops:CreateChat` - 새 채팅 대화 생성
- `aidevops:SendMessage` - 메시지 전송 및 응답 수신

AWS IAM 콘솔로 이동하여 DevOps 에이전트 운영자 역할을 찾고 이러한 권한을 역할 정책에 추가합니다. 권한이 추가된 직후 채팅을 사용할 수 있습니다.

두 옵션 중 하나를 완료한 후 DevOps Agent Space 웹 앱을 새로 고치면 페이지 왼쪽에 채팅 패널이 나타납니다.

AWS DevOps Agent에 대한 기능 구성

AWS DevOps 에이전트 기능은 에이전트를 기존 도구 및 인프라에 연결하여 에이전트의 기능을 확장합니다. 이러한 기능을 구성하여 포괄적인 인시던트 조사, 자동화된 대응 워크플로, DevOps 에코시스템과의 원활한 통합을 지원합니다.

다음 기능은 DevOps 에이전트의 효과를 극대화하는 데 도움이 됩니다.

- AWS EKS 액세스 설정 - 퍼블릭 및 프라이빗 EKS 환경 모두에 대해 Kubernetes 클러스터, 포드 로그 및 클러스터 이벤트의 내부 검사 활성화
- Azure 통합 - Azure 구독 및 Azure DevOps 조직을 연결하여 Azure 리소스를 조사하고 Azure DevOps 배포와 인시던트의 상관관계를 파악합니다.
- CI/CD 파이프라인 통합 - GitHub 및 GitLab 파이프라인을 연결하여 배포를 인시던트와 연관시키고 조사 중에 코드 변경을 추적합니다.
- MCP 서버 연결 - 모델 컨텍스트 프로토콜을 통해 외부 관찰성 도구와 사용자 지정 모니터링 시스템을 연결하여 조사 기능 확장
- 다중 계정 AWS 액세스 - 인시던트 대응 중에 전체 조직의 리소스를 조사하도록 보조 AWS 계정을 구성합니다.
- 원격 측정 소스 통합 - 포괄적인 관찰성 데이터 액세스를 위해 Datadog, Dynatrace, Grafana, New Relic 및 Splunk와 같은 모니터링 플랫폼을 연결합니다.
- 티켓팅 및 채팅 통합 - ServiceNow, PagerDuty 및 Slack을 연결하여 인시던트 대응 워크플로를 자동화하고 팀 협업을 활성화합니다.
- Webhook 구성 - 외부 시스템이 HTTP 요청을 통해 DevOps 에이전트 조사를 자동으로 트리거하도록 허용
- Amazon EventBridge 통합 - 조사 및 완화 수명 주기 이벤트를 Amazon EventBridge 대상으로 라우팅하여 이벤트 기반 애플리케이션에 AWS DevOps 에이전트 통합

팀의 특정 요구 사항과 기존 도구 스택에 따라 각 기능을 독립적으로 구성할 수 있습니다. 인시던트 대응 워크플로에 가장 중요한 통합으로 시작한 다음 필요에 따라 추가 기능으로 확장합니다.

공개 미리 보기에서 정식 출시로 마이그레이션

공개 미리 보기 중에 AWS DevOps 에이전트를 사용한 경우 GA 릴리스 전에 IAM 역할을 업데이트해야 합니다. 이 가이드에서는 계정의 모니터링 역할 및 운영자 역할을 업데이트하는 방법을 안내합니다.

변경 사항

1. 미리 보기 중 온디맨드 채팅 기록에 더 이상 액세스할 수 없음
2. 새로운 관리형 정책은 미리 보기 중에 사용 가능한 정책을 대체합니다.
3. 에이전트 스페이스에는 오래된 IAM Identity Center 애플리케이션 액세스 범위가 있을 수 있습니다.

공개 미리 보기의 온디맨드 채팅 기록

GA 릴리스에는 채팅 기록에 대한 액세스 제어를 강화하기 위한 추가 보안 조치가 도입되었습니다. 이러한 변경으로 인해 공개 미리 보기 기간(2026년 3월 30일 이전)의 온디맨드 채팅 기록에 더 이상 액세스할 수 없습니다. 공개 미리 보기 중에 생성된 조사 저널 및 조사 결과는 영향을 받지 않습니다. 이 변경 사항은 온디맨드 채팅 대화에만 적용됩니다.

새로운 관리형 정책

GA의 경우 미리 보기 기간 정책을 대체하는 새 관리형 정책을 AWS 제공합니다.

역할 유형	제거	더하기
모니터링	`AIOpsAssistantPolicy` 관리형 정책	`AIDevOpsAgentAccessPolicy` 관리형 정책
연산자(IAM 및 IDC)	인라인 정책	`AIDevOpsOperatorAppAccessPolicy` 관리형 정책

또한 운영자 역할에는 업데이트된 신뢰 정책이 필요하고 IDC 운영자 역할에는 새 인라인 정책이 필요합니다.

사전 조건

- DevOps 에이전트 역할이 구성된 AWS 계정에 대한 액세스(기본 및 모든 보조 계정)
- 역할, 정책 및 신뢰 관계를 수정할 수 있는 IAM 권한
- 에이전트 스페이스 ID, AWS 계정 ID 및 리전(DevOps 에이전트 콘솔에서 볼 수 있음)

1단계: 모니터링 역할 업데이트

기본 계정 및 각 보조 계정에서 모니터링 역할을 업데이트합니다. 다음은 에이전트 공간의 기능 탭 아래에 구성된 기본/보조 소스 역할입니다(예: 기본/보조 역할: DevOpsAgentRole-AgentSpace-3xj2396z).

1. DevOps 에이전트 콘솔에서 에이전트 공간으로 이동하여 기능 탭을 선택합니다.
2. 기본/보조 소스의 모니터링 역할(예: DevOpsAgentRole-AgentSpace-3xj2396z)을 찾아 편집을 선택합니다.
3. 권한 정책에서 AIOpsAssistantPolicy AWS 관리형 정책을 제거합니다.
4. 권한 추가, 정책 연결을 선택하고 AIDevOpsAgentAccessPolicy 관리형 정책을 연결합니다.
5. 인라인 정책을 편집하고 해당 내용을 다음으로 대체하여 계정 ID를 대체합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateServiceLinkedRoles",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": [
                "arn:aws:iam::<account-id>:role/aws-service-role/resource-
explorer-2.amazonaws.com/AWSServiceRoleForResourceExplorer"
            ]
        }
    ]
}
```

1. 모니터링 역할에 대한 신뢰 정책은 변경할 필요가 없습니다. 다음과 일치하는지 확인합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
```

```
                "Service": "aidevops.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "<account-id>"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:aidevops:<region>:<account-
id>:agentspace/*"
                }
            }
        }
    ]
}
```

- 각 보조 계정의 모니터링 역할에 대해 2~6단계를 반복합니다.

2단계: 연산자 역할 업데이트(IAM)

1. DevOps 에이전트 콘솔에서 액세스 탭을 선택하고 운영자 역할을 찾습니다.
2. IAM 콘솔에서 연산자 역할에서 기존 인라인 정책을 제거합니다.
3. 권한 추가, 정책 연결을 선택하고 `AIDevOpsOperatorAppAccessPolicy` 관리형 정책을 연결합니다.
4. 신뢰 관계 탭을 선택하고 신뢰 정책 편집을 선택합니다. 계정 ID, 리전 및 에이전트 스페이스 ID를 대체하여 신뢰 정책을 다음으로 바꿉니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "aidevops.amazonaws.com"
            },
            "Action": ["sts:AssumeRole", "sts:TagSession"],
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "<account-id>"
```

```
                },
                "ArnEquals": {
                    "aws:SourceArn": "arn:aws:aidevops:<region>:<account-
id>:agentspace/<agentspace-id>"
                }
            }
        }
    ]
}
```

3단계: 운영자 역할 업데이트(IDC)

DevOps 에이전트와 함께 IAM Identity Center를 사용하는 경우 각 IDC 운영자 역할을 업데이트합니다.

1. IAM 콘솔에서 역할로 이동하여를 검색WebappIDC하여 DevOps 에이전트 IDC 역할(예: DevOpsAgentRole-WebappIDC-<id>)을 찾습니다.
2. 각 IDC 역할에 대해:

a. 기존 인라인 정책을 제거합니다.

b. 권한 추가, 정책 연결을 선택하고 AIDevOpsOperatorAppAccessPolicy 관리형 정책을 연결합니다.

c. 신뢰 관계 탭을 선택하고 신뢰 정책 편집을 선택합니다. 계정 ID, 리전 및 에이전트 스페이스 ID를 대체하여 신뢰 정책을 다음으로 바꿉니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "aidevops.amazonaws.com"
            },
            "Action": ["sts:AssumeRole", "sts:TagSession"],
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "<account-id>"
                },
                "ArnEquals": {
                    "aws:SourceArn": "arn:aws:aidevops:<region>:<account-
id>:agentspace/<agentspace-id>"
```

```
                }
            }
        },
        {
            "Sid": "TrustedIdentityPropagation",
            "Effect": "Allow",
            "Principal": {
                "Service": "aidevops.amazonaws.com"
            },
            "Action": "sts:SetContext",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "<account-id>"
                },
                "ArnEquals": {
                    "aws:SourceArn": "arn:aws:aidevops:<region>:<account-
id>:agentspace/<agentspace-id>"
                },
                "ForAllValues:ArnEquals": {
                    "sts:RequestContextProviders": [
                        "arn:aws:iam::aws:contextProvider/IdentityCenter"
                    ]
                },
                "Null": {
                    "sts:RequestContextProviders": "false"
                }
            }
        }
    ]
}
```

d. 계정 ID를 대체하여 다음 권한으로 새 인라인 정책을 생성합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowDevOpsAgentSSOAccess",
            "Effect": "Allow",
            "Action": [
                "sso:ListInstances",
                "sso:DescribeInstance"
            ],
```

```
            "Resource": "*"
        },
        {
            "Sid": "AllowDevOpsAgentIDCUserAccess",
            "Effect": "Allow",
            "Action": "identitystore:DescribeUser",
            "Resource": [
                "arn:aws:identitystore::<account-id>:identitystore/*",
                "arn:aws:identitystore:::user/*"
            ]
        }
    ]
}
```

IAM Identity Center 다시 연결(해당하는 경우)

공개 미리 보기 중에 생성된 에이전트 공간에는 오래된 액세스 범위로 구성된 IAM Identity Center 애플리케이션이 있을 수 있습니다. GA의 경우 올바른 범위는 입니다**aidevops:read_write**. IAM Identity Center 애플리케이션에 이전 범위(**awsaidevops:read_write**)가 있는 경우 IAM Identity Center의 연결을 끊었다가 다시 연결해야 합니다.

IAM Identity Center 애플리케이션 범위를 확인하는 방법

다음 AWS CLI 명령을 실행하여 IAM Identity Center 애플리케이션의 범위를 확인합니다. 애플리케이션 아래의 IAM Identity Center 콘솔에서 애플리케이션 ARN을 찾을 수 있습니다.

```
aws sso-admin list-application-access-scopes \
  --application-arn arn:aws:sso::<account-id>:application/<instance-id>/<application-
id>
```

출력에 올바른 범위가 표시되어야 합니다**aidevops:read_write**.

```
{
    "Scopes": [
        {
            "Scope": "aidevops:read_write"
        }
    ]
}
```

범위에가 표시되면 오래된 **awsaidevops:read_write**것입니다. 아래 단계에 따라 업데이트합니다.

IAM Identity Center를 다시 연결하는 방법

AWS 관리형 IAM Identity Center 애플리케이션의 액세스 범위는 직접 업데이트할 수 없습니다. 연결을 끊었다가 다시 연결해야 합니다.

1. AWS DevOps 에이전트 콘솔에서 에이전트 공간으로 이동하여 액세스 탭을 선택합니다.
2. IAM Identity Center 구성 옆에 있는 연결 해제를 선택합니다.
3. 연결 해제를 확인합니다.
4. 연결을 선택하여 IAM Identity Center를 다시 설정합니다. 서비스는 올바른 범위를 가진 새 IAM Identity Center 애플리케이션을 생성합니다.
5. IAM Identity Center 콘솔에서 새 애플리케이션에 사용자 및 그룹을 재할당합니다.

> **Important**
>
> 연결을 해제하면 IAM Identity Center 사용자 계정과 연결된 개별 사용자 채팅 및 아티팩트 기록이 제거됩니다. 사용자는 다시 연결한 후 다시 로그인해야 합니다.

Verification(확인)

모든 단계를 완료한 후:

1. DevOps 에이전트 콘솔로 돌아가 에이전트 스페이스 액세스 탭에 권한 오류가 나타나지 않는지 확인합니다.
2. 연산자 웹 앱을 테스트하여 올바르게 로드되고 작동하는지 확인합니다.
3. IDC를 사용하는 경우 사용자가 운영자 환경을 인증하고 액세스할 수 있는지 확인합니다.

문제 해결

마이그레이션 후 권한 거부 오류

- 이 제거`AIOpsAssistantPolicy`되었고 모니터링 역할에 `AIDevOpsAgentAccessPolicy` 연결되어 있는지 확인합니다.
- 이전 인라인 정책`AIDevOpsOperatorAppAccessPolicy`이 제거되었고 운영자 역할에 연결되어 있는지 확인합니다.

- 연산자 신뢰 정책에가 포함되어 있는지 확인합니다`sts:TagSession`.
- 모든 자리 표시자 값(`<account-id>`, `<region>`, `<agentspace-id>`)을 실제 값으로 바꾸었는지 확인합니다.

보조 계정이 작동하지 않음

- 각 보조 계정의 모니터링 역할은 독립적으로 업데이트해야 합니다. 각 계정에 로그인하고 1단계를 반복합니다.

IDC 인증 실패

- IDC 신뢰 정책에 `sts:AssumeRole`/`sts:TagSession` 문과 `TrustedIdentityPropagation` 문이 모두 포함되어 있는지 확인합니다.
- `sso:ListInstances`, `sso:DescribeInstance`및를 사용하여 인라인 정책이 생성`identitystore:DescribeUser`되었는지 확인합니다.

마이그레이션 후 온디맨드 채팅 기록 누락

- GA 릴리스 후에는 공개 미리 보기 기간의 온디맨드 채팅 기록에 액세스할 수 없습니다. 이는 GA에 도입된 향상된 보안 조치로 인해 예상되는 동작입니다. 조사 저널 및 공개 미리 보기 결과는 영향을 받지 않습니다.

AWS EKS 액세스 설정

퍼블릭 클러스터와 프라이빗 클러스터 모두에 대해 읽기 전용 `kubectl` 명령을 실행하여 Amazon EKS 클러스터의 문제를 조사 AWS DevOps 할 수 있습니다. 원하는 수의 EKS 클러스터를 동일한 에이전트 스페이스에 연결할 수 있습니다.

연결되면 에이전트는 리소스 설명, 포드 로그 검색, 클러스터 이벤트 검사, 노드 상태 확인 등 클러스터의 운영 문제를 진단하는 데 도움이 될 수 있습니다. 에이전트는 클러스터에서 리소스를 생성, 수정 또는 삭제할 수 없습니다.

사전 조건

EKS 액세스를 설정하기 전에 EKS 클러스터의 인증 모드에 EKS API가 포함되어 있는지 확인합니다. Amazon EKS 콘솔의 액세스 탭에서 이를 확인할 수 있습니다. 모드에 EKS API가 포함되지 않은 경우 진행하기 전에에서 수행하는 모드를 선택합니다.

설정

이러한 단계는 액세스 항목을 생성하려는 각 클러스터의 Amazon EKS 콘솔에서 완료해야 합니다. IAM 역할 ARN은 에이전트 스페이스(참조the section called “에이전트 스페이스 생성”)의 기능 > 클라우드 > 기본 소스 > 편집에서 찾을 수 있습니다.

1. 액세스 탭으로 이동합니다. 인증 모드에 이미 EKS API가 표시된 경우 액세스 항목을 추가할 수 있습니다. 그렇지 않으면 EKS API가 포함된 모드를 선택합니다.
2. 액세스 탭에서 새 IAM 액세스 항목을 생성합니다. 기본 클라우드 소스 IAM 역할 ARN을 복사하여 액세스 항목의 IAM 보안 주체로 입력합니다. 다음을 클릭합니다.
3. AWS 관리형 AmazonAIOpsAssistantPolicy 액세스 정책을 선택하고 액세스 범위에 대해 클러스터를 선택합니다. (또는 에이전트가 특정 네임스페이스에만 액세스하도록 하려면 원하는 Kubernetes 네임스페이스를 선택합니다.) 정책 추가를 클릭한 후 다음을 클릭합니다.
4. 변경 사항을 검토하고 올바른 액세스 항목 정책 및 IAM 역할이 선택되었는지 확인하고 "생성"을 클릭하여 액세스 항목을 생성합니다.

EKS 액세스가 올바르게 구성되었는지 확인하려면 운영자 앱으로 이동하여 에이전트에게 "기본 네임스페이스의 모든 포드 나열" 또는 "클러스터의 최근 이벤트 표시"와 같은 클러스터에 대한 질문을 하여 새 조사를 시작합니다.

문제 해결

에이전트가 클러스터에 연결할 수 없는 경우 액세스 항목이 설정 대화 상자에 표시된 올바른 IAM 역할 ARN을 사용하고 있고 AmazonAIOpsAssistantPolicy 액세스 정책이 연결되어 있는지 확인합니다.

Azure 연결

Azure 통합을 통해 AWS DevOps Agent는 Azure 환경의 리소스를 조사하고 Azure DevOps 파이프라인 배포와 운영 인시던트의 상관관계를 파악할 수 있습니다. 에이전트는 Azure를 연결하여 Azure 인프라에 대한 가시성을 확보하고 AWS 및 Azure 리소스 모두에서 근본 원인 분석을 수행할 수 있습니다.

Azure 통합은 두 가지 독립적인 기능으로 구성됩니다.

- Azure 리소스 - 에이전트가 가상 머신, Azure Kubernetes Service(AKS) 클러스터, 데이터베이스 및 네트워킹 구성 요소와 같은 Azure 클라우드 리소스를 검색하고 조사할 수 있습니다. 에이전트는 Azure 리소스 그래프를 사용하여 인시던트 조사 중에 리소스를 쿼리합니다.

- Azure DevOps - 에이전트가 Azure DevOps 리포지토리 및 파이프라인 실행 내역에 액세스할 수 있습니다. 에이전트는 코드 변경 및 배포를 인시던트와 연관시켜 잠재적 근본 원인을 식별할 수 있습니다.

각 기능은 AWS 계정 수준에서 등록된 다음 개별 에이전트 스페이스와 연결할 수 있습니다.

등록 방법

AWS DevOps Agent는 Azure에 연결하는 두 가지 방법을 지원합니다.

- 관리자 동의 - Azure 테넌트에서 AWS DevOps 에이전트 Entra 애플리케이션을 승인하는 간소화된 동의 기반 흐름입니다. 콘솔에서 관리자 동의 옵션으로 표시됩니다. 이 방법을 사용하려면 Microsoft Entra ID에서 관리자 동의를 수행할 권한이 있는 계정으로 로그인해야 합니다.
- 앱 등록 - 아웃바운드 자격 증명 연동을 사용하여 페더레이션 자격 증명으로 자체 Entra 애플리케이션을 생성하는 자체 관리형 접근 방식입니다. 콘솔에서 앱 등록 옵션으로 표시됩니다. 이 방법은 애플리케이션 구성을 더 잘 제어해야 하거나 관리자 동의 권한을 사용할 수 없는 경우에 적합합니다.

두 방법 모두 동일한 기능을 제공합니다. 동일한 AWS 계정 내에서 하나 또는 두 가지 방법을 모두 사용할 수 있습니다.

알려진 제한 사항

- 관리자 동의: Azure 테넌트당 AWS 계정 1개 - 각 Azure 테넌트는 한 번에 하나의 AWS 계정과 연결된 AWS DevOps 에이전트 Entra 앱만 가질 수 있습니다. 동일한 테넌트를 다른 AWS 계정과 연결하려면 먼저 기존 등록을 등록 취소해야 합니다.
- 앱 등록: 등록당 고유한 애플리케이션 - 각 앱 등록은 다른 애플리케이션(클라이언트 ID)을 사용해야 합니다. 동일한 클라이언트 ID로 여러 구성을 등록할 수 없습니다.
- Azure DevOps: 소스 코드 액세스 - Azure DevOps 통합은 소스 코드가 호스팅되는 위치에 관계없이 파이프라인 실행 기록에 대한 액세스를 제공합니다. 그러나 실제 소스 코드에 액세스하려면 지원되는 소스 공급자(예:)를 통해 리포지토리를 별도로 연결해야 합니다the section called “GitHub 연결”. Bitbucket에서 호스팅되는 소스 코드는 Azure DevOps 통합을 통해 직접 액세스할 수 없습니다.

주제

- the section called “Azure 리소스 연결”
- the section called “Azure DevOps 연결”

Azure 리소스 연결

Azure 리소스 통합을 통해 AWS DevOps Agent는 인시던트 조사 중에 Azure 구독의 리소스를 검색하고 조사할 수 있습니다. 에이전트는 리소스 검색에 Azure 리소스 그래프를 사용하며 Azure 환경 전반의 지표, 로그 및 구성 데이터에 액세스할 수 있습니다.

이 통합은 AWS 계정 수준에서 Azure를 등록한 다음 특정 Azure 구독을 개별 에이전트 스페이스와 연결하는 2단계 프로세스를 따릅니다.

사전 조건

Azure 리소스를 연결하기 전에 다음이 있는지 확인합니다.

- AWS DevOps 에이전트 콘솔에 대한 액세스
- 대상 구독에 액세스할 수 있는 Azure 계정
- 관리자 동의 방법의 경우: Microsoft Entra ID에서 관리자 동의를 수행할 권한이 있는 계정
- 앱 등록 방법의 경우: 페더레이션 자격 증명을 구성할 수 있는 권한이 있는 Entra 애플리케이션 및 AWS 계정에서 활성화된 아웃바운드 자격 증명 페더레이션

> 참고: 에이전트 스페이스 내에서 등록을 시작할 수도 있습니다. 보조 소스로 이동하여 추가를 클릭하고 Azure를 선택합니다. Azure Cloud가 아직 등록되지 않은 경우 콘솔이 먼저 등록을 안내합니다.

관리자 동의를 통한 Azure 리소스 등록

관리자 동의 메서드는 AWS DevOps 에이전트 관리형 애플리케이션과 함께 동의 기반 흐름을 사용합니다.

1단계: 등록 시작

1. AWS Management Console에 로그인하고 AWS DevOps 에이전트 콘솔로 이동합니다.
2. 기능 공급자 페이지로 이동
3. Azure 클라우드 섹션을 찾아 등록을 클릭합니다.
4. 관리자 동의 등록 방법 선택

2단계: 관리자 동의 완료

1. 요청 중인 권한 검토

2. 계속하려면 클릭 - Microsoft Entra 관리자 동의 페이지로 리디렉션됩니다.
3. 관리자 동의를 수행할 권한이 있는 사용자 보안 주체 계정으로 로그인
4. AWS DevOps 에이전트 애플리케이션 검토 및 동의 부여

3단계: 사용자 권한 부여 완료

1. 관리자 동의 후 권한 있는 테넌트의 멤버 자격 증명을 확인하기 위한 사용자 권한 부여 메시지가 표시됩니다.
2. 동일한 Azure 테넌트에 속한 계정으로 로그인
3. 권한 부여 후 성공 상태의 AWS DevOps 에이전트 콘솔로 다시 리디렉션됩니다.

4단계: 역할 할당

아래 Azure 역할 할당을 참조하세요. 멤버를 선택할 때 AWS DevOps 에이전트를 검색합니다.

앱 등록을 통해 Azure 리소스 등록

앱 등록 메서드는 페더레이션 자격 증명이 있는 자체 Entra 애플리케이션을 사용합니다.

1단계: 등록 시작

1. AWS DevOps 에이전트 콘솔에서 기능 공급자 페이지로 이동합니다.
2. Azure 클라우드 섹션을 찾아 등록을 클릭합니다.
3. 앱 등록 방법 선택

2단계: Entra 애플리케이션 생성 및 구성

콘솔에 표시된 지침에 따라 다음을 수행합니다.

1. AWS 계정에서 아웃바운드 자격 증명 연동 활성화(IAM 콘솔에서 계정 설정 → 아웃바운드 자격 증명 연동으로 이동)
2. Microsoft Entra ID에서 Entra 애플리케이션을 생성하거나 기존 애플리케이션을 사용합니다.
3. 애플리케이션에서 페더레이션 자격 증명 구성

3단계: 등록 세부 정보 제공

등록 양식에 다음을 입력합니다.

- 테넌트 ID - Azure 테넌트 식별자
- 테넌트 이름 - 테넌트의 표시 이름입니다.
- 클라이언트 ID - 생성한 Entra 애플리케이션의 애플리케이션(클라이언트) ID입니다.
- 대상 - 페더레이션 자격 증명의 대상 식별자입니다.

4단계: IAM 역할 생성

콘솔을 통해 등록을 제출하면 IAM 역할이 자동으로 생성됩니다. 이를 통해 AWS DevOps 에이전트는 자격 증명을 수임하고를 호출할 수 있습니다`sts:GetWebIdentityToken`.

5단계: 역할 할당

아래 Azure 역할 할당을 참조하세요. 멤버를 선택할 때 생성한 Entra 애플리케이션을 검색합니다.

6단계: 등록 완료

1. AWS DevOps 에이전트 콘솔에서 구성 확인
2. 제출을 클릭하여 등록을 완료합니다.

Azure 역할 할당

등록 후 애플리케이션에 Azure 구독에 대한 읽기 액세스 권한을 부여합니다. 이 단계는 관리자 동의 및 앱 등록 방법 모두에서 동일합니다.

1. Azure 포털에서 대상 구독으로 이동합니다.
2. 액세스 제어(IAM)로 이동
3. 추가 > 역할 할당 추가를 클릭합니다.
4. 독자 역할을 선택하고 다음을 클릭합니다.
5. 멤버 선택을 클릭하고 애플리케이션(관리자 동의용 AWS DevOps 에이전트 또는 앱 등록용 자체 Entra 애플리케이션)을 검색합니다.
6. 애플리케이션을 선택하고 검토 + 할당을 클릭합니다.
7. (선택 사항) 에이전트가 Azure Kubernetes Service(AKS) 클러스터에 액세스할 수 있도록 하려면 다음 AKS 액세스 설정을 완료합니다.

> 보안 요구 사항: 서비스 보안 주체에는 리더 역할(및 선택적으로 아래 나열된 AKS 읽기 전용 역할)만 할당해야 합니다. 리더 역할은 에이전트를 읽기 전용 작업으로 제한하고 간

접 프롬프트 주입 공격의 영향을 제한하는 보안 경계 역할을 합니다. 쓰기 또는 작업 권한이 있는 역할을 할당하면 프롬프트 주입의 폭발 반경이 크게 증가하여 Azure 리소스가 손상될 수 있습니다. AWS DevOps Agent는 읽기 작업만 수행합니다. 에이전트는 Azure 리소스를 수정, 생성 또는 삭제하지 않습니다.

AKS 액세스 설정(선택 사항)

1단계: Azure Resource Manager(ARM) 수준 액세스

Azure Kubernetes 서비스 클러스터 사용자 역할을 애플리케이션에 할당합니다.

Azure 포털에서 구독 → 구독 선택 → 액세스 제어(IAM) → 역할 할당 추가 → Azure Kubernetes 서비스 클러스터 사용자 역할 선택 → 애플리케이션에 할당(관리자 동의용 AWS DevOps 에이전트 또는 앱 등록용 자체 Entra 애플리케이션)으로 이동합니다.

여기에는 구독의 모든 AKS 클러스터가 포함됩니다. 특정 클러스터로 범위를 지정하려면 대신 리소스 그룹 또는 개별 클러스터 수준에서를 할당합니다.

2단계: Kubernetes API 액세스

클러스터의 인증 구성을 기반으로 한 가지 옵션을 선택합니다.

옵션 A: Kubernetes용 Azure 역할 기반 액세스 제어(RBAC)(권장)

1. 아직 활성화되지 않은 경우 클러스터에서 Azure RBAC 활성화: Azure 포털 → AKS 클러스터 → 설정 → 보안 구성 → 인증 및 권한 부여 → Azure RBAC 선택
2. 읽기 전용 역할 할당: Azure 포털 → 구독 → 구독 선택 → 액세스 제어(IAM) → 역할 할당 추가 → Azure Kubernetes Service RBAC Reader 선택 → 애플리케이션에 할당

여기에는 구독의 모든 AKS 클러스터가 포함됩니다.

옵션 B: Azure Active Directory(Azure AD) + Kubernetes RBAC

클러스터가 이미 기본 Azure AD 인증 구성을 사용하고 Azure RBAC를 활성화하지 않으려는 경우이 옵션을 사용합니다. 이를 위해서는 클러스터당 `kubectl` 설정이 필요합니다.

1. 다음 매니페스트를 로 저장합니다`devops-agent-reader.yaml`.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
```

```
metadata:
  name: devops-agent-reader
rules:
  - apiGroups: [""]
    resources: ["namespaces", "pods", "pods/log", "services", "events", "nodes"]
    verbs: ["get", "list"]
  - apiGroups: ["apps"]
    resources: ["deployments", "replicasets", "statefulsets", "daemonsets"]
    verbs: ["get", "list"]
  - apiGroups: ["metrics.k8s.io"]
    resources: ["pods", "nodes"]
    verbs: ["get", "list"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: devops-agent-reader-binding
subjects:
  - kind: User
    name: "<SERVICE_PRINCIPAL_OBJECT_ID>"
    apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: devops-agent-reader
  apiGroup: rbac.authorization.k8s.io
```

1. 를 서비스 보안 주체의 객체 ID<SERVICE_PRINCIPAL_OBJECT_ID>로 바꿉니다. 이를 찾으려면 Azure 포털 → Entra ID → 엔터프라이즈 애플리케이션 → 애플리케이션 이름(관리자 동의용 AWS DevOps 에이전트 또는 앱 등록용 자체 Entra 애플리케이션)을 검색합니다.
2. 각 클러스터에 적용:

```
az aks get-credentials --resource-group <rg> --name <cluster-name>
kubectl apply -f devops-agent-reader.yaml
```

> 참고: 로컬 계정만 사용하는 클러스터(Azure AD 제외)는 지원되지 않습니다. 이 기능을 사용하려면 클러스터에서 Azure AD 통합을 활성화하는 것이 좋습니다.

최소 권한 사용자 지정 역할(선택 사항)

더 엄격한 액세스 제어를 위해 광범위한 리더 역할 대신 AWS DevOps 에이전트가 사용하는 리소스 공급자로만 범위가 지정된 사용자 지정 Azure 역할을 생성할 수 있습니다.

```
{
  "Name": "AWS DevOps Agent - Azure Reader",
  "Description": "Least-privilege read-only access for AWS DevOps Agent incident
 investigations.",
  "Actions": [
    "Microsoft.AlertsManagement/*/read",
    "Microsoft.Compute/*/read",
    "Microsoft.ContainerRegistry/*/read",
    "Microsoft.ContainerService/*/read",
    "Microsoft.ContainerService/managedClusters/commandResults/read",
    "Microsoft.DocumentDB/*/read",
    "Microsoft.Insights/*/read",
    "Microsoft.KeyVault/vaults/read",
    "Microsoft.ManagedIdentity/*/read",
    "Microsoft.Monitor/*/read",
    "Microsoft.Network/*/read",
    "Microsoft.OperationalInsights/*/read",
    "Microsoft.ResourceGraph/resources/read",
    "Microsoft.ResourceHealth/*/read",
    "Microsoft.Resources/*/read",
    "Microsoft.Sql/*/read",
    "Microsoft.Storage/*/read",
    "Microsoft.Web/*/read"
  ],
  "NotActions": [],
  "DataActions": [],
  "NotDataActions": [],
  "AssignableScopes": [
    "/subscriptions/{your-subscription-id}"
  ]
}
```

에이전트 스페이스와 구독 연결

계정 수준에서 Azure를 등록한 후 특정 구독을 에이전트 스페이스와 연결합니다.

1. AWS DevOps 에이전트 콘솔에서 에이전트 스페이스를 선택합니다.
2. 기능 탭으로 이동
3. 보조 소스 섹션에서 추가를 클릭합니다.
4. Azure 선택
5. 연결할 Azure 구독의 구독 ID를 제공합니다.

6. 추가를 클릭하여 연결을 완료합니다.

여러 구독을 동일한 에이전트 스페이스와 연결하여 Azure 환경 전체에서 에이전트에게 가시성을 제공할 수 있습니다.

Azure 리소스 연결 관리

- 연결된 구독 보기 - 기능 탭의 보조 소스 섹션에는 연결된 모든 Azure 구독이 나열됩니다.
- 구독 제거 - 에이전트 스페이스에서 구독을 연결 해제하려면 보조 소스 목록에서 구독을 선택하고 제거를 클릭합니다. 이는 계정 수준 등록에는 영향을 주지 않습니다.
- 등록 제거 - Azure 클라우드 등록을 완전히 제거하려면 기능 공급자 페이지로 이동하여 등록을 삭제합니다. 먼저 모든 에이전트 스페이스 연결을 제거해야 합니다.

Azure DevOps 연결

Azure DevOps 통합을 통해 AWS DevOps 에이전트는 Azure DevOps 조직의 리포지토리 및 파이프라인 실행 기록에 액세스할 수 있습니다. 에이전트는 코드 변경 및 배포를 운영 인시던트와 연관시켜 잠재적 근본 원인을 식별할 수 있습니다.

> 참고: Azure DevOps 파이프라인은 Azure Repos, GitHub 또는 Bitbucket의 소스 코드를 사용할 수 있습니다. Azure DevOps 통합은 소스 공급자에 관계없이 파이프라인 실행 기록에 대한 액세스를 제공합니다. 그러나 조사 중에 실제 소스 코드에 액세스하려면 와 같이 지원되는 통합을 통해 리포지토리를 별도로 연결해야 합니다the section called “GitHub 연결”. Bitbucket의 소스 코드는이 통합을 통해 직접 액세스할 수 없습니다.

이 통합은 AWS 계정 수준에서 Azure DevOps를 등록한 다음 특정 프로젝트를 개별 에이전트 스페이스와 연결하는 2단계 프로세스를 따릅니다.

사전 조건

Azure DevOps를 연결하기 전에 다음이 있는지 확인합니다.

- AWS DevOps 에이전트 콘솔에 대한 액세스
- 리포지토리 및 파이프라인 기록이 포함된 프로젝트가 하나 이상 있는 Azure DevOps 조직
- Azure DevOps 조직에 사용자를 추가할 수 있는 권한
- 관리자 동의 방법의 경우: Microsoft Entra ID에서 관리자 동의를 수행할 권한이 있는 계정

- 앱 등록 방법의 경우: 페더레이션 ID 자격 증명을 구성할 수 있는 권한이 있는 Entra 애플리케이션 및 AWS 계정에서 활성화된 아웃바운드 자격 증명 페더레이션

> 참고: 에이전트 스페이스 내에서 등록을 시작할 수도 있습니다. 파이프라인 섹션으로 이동하여 추가를 클릭하고 Azure DevOps를 선택합니다. Azure DevOps가 아직 등록되지 않은 경우 콘솔이 먼저 등록을 안내합니다.

관리자 동의를 통한 Azure DevOps 등록

관리자 동의 메서드는 AWS DevOps 에이전트 관리형 애플리케이션과 함께 동의 기반 흐름을 사용합니다.

1단계: 등록 시작

1. AWS Management Console에 로그인하고 AWS DevOps 에이전트 콘솔로 이동합니다.
2. 기능 공급자 페이지로 이동
3. Azure DevOps 섹션을 찾아 등록을 클릭합니다.
4. 메시지가 표시되면 Azure DevOps 조직 이름을 입력합니다.

2단계: 관리자 동의 완료

1. 계속하려면 클릭 - Microsoft Entra 관리자 동의 페이지로 리디렉션됩니다.
2. 관리자 동의를 수행할 권한이 있는 사용자 보안 주체 계정으로 로그인
3. AWS DevOps 에이전트 애플리케이션 검토 및 동의 부여

3단계: 사용자 권한 부여 완료

1. 관리자 동의 후 권한 있는 테넌트의 멤버 자격 증명을 확인하기 위한 사용자 권한 부여 메시지가 표시됩니다.
2. 동일한 Azure 테넌트에 속한 계정으로 로그인
3. 권한 부여 후 성공 상태의 AWS DevOps 에이전트 콘솔로 다시 리디렉션됩니다.

4단계: Azure DevOps에서 액세스 권한 부여

아래 Azure DevOps에서 액세스 권한 부여를 참조하세요. 사용자를 추가할 때 AWS DevOps 에이전트를 검색합니다.

앱 등록을 통해 Azure DevOps 등록

앱 등록은 Azure 리소스와 Azure DevOps 간에 공유됩니다. Azure 리소스에 대한 앱 등록을 이미 완료한 경우 Azure DevOps에서 액세스 권한 부여로 건너뛸 수 있습니다.

1단계: ADO 앱 등록 시작

1. AWS DevOps 에이전트 콘솔에서 기능 공급자 페이지로 이동합니다.
2. Azure 클라우드 섹션을 찾아 등록을 클릭합니다.
3. 앱 등록 방법 선택

2단계: Entra 애플리케이션 생성 및 구성

콘솔에 표시된 지침에 따라 다음을 수행합니다.

1. AWS 계정에서 아웃바운드 자격 증명 연동 활성화(IAM 콘솔에서 계정 설정 → 아웃바운드 자격 증명 연동으로 이동)
2. Microsoft Entra ID에서 Entra 애플리케이션을 생성하거나 기존 애플리케이션을 사용합니다.
3. 애플리케이션에서 페더레이션 ID 자격 증명 구성

3단계: 등록 세부 정보 제공

등록 양식에 다음을 입력합니다.

- 테넌트 ID - Azure 테넌트 식별자
- 테넌트 이름 - 테넌트의 표시 이름입니다.
- 클라이언트 ID - Entra 애플리케이션의 애플리케이션(클라이언트) ID입니다.
- 대상 - 페더레이션 자격 증명의 대상 식별자입니다.

4단계: IAM 역할 생성

콘솔을 통해 등록을 제출하면 IAM 역할이 자동으로 생성됩니다. 이를 통해 AWS DevOps 에이전트는 자격 증명을 수임하고를 호출할 수 있습니다`sts:GetWebIdentityToken`.

5단계: 등록 완료

1. AWS DevOps 에이전트 콘솔에서 구성 확인

2. 제출을 클릭하여 등록을 완료합니다.

6단계: Azure DevOps에서 액세스 권한 부여

아래 Azure DevOps에서 액세스 권한 부여를 참조하세요. 사용자를 추가할 때 앱 등록 중에 생성한 Entra 애플리케이션을 검색합니다.

Azure DevOps에서 액세스 권한 부여

등록 후 Azure DevOps 조직에 애플리케이션 액세스 권한을 부여합니다. 이 단계는 관리자 동의 및 앱 등록 방법 모두에서 동일합니다.

1. Azure DevOps에서 조직 설정 > 사용자 > 사용자 추가로 이동합니다.
2. 애플리케이션 검색(관리자 동의용 AWS DevOps 에이전트 또는 앱 등록용 자체 Entra 애플리케이션)
3. 액세스 수준을 기본으로 설정
4. 프로젝트에 추가에서 에이전트가 액세스할 프로젝트를 선택합니다.
5. Azure DevOps 그룹에서 프로젝트 리더를 선택합니다.
6. 추가를 클릭하여 완료

> 보안 요구 사항: Project Readers 그룹만 할당합니다. 읽기 전용 액세스는 에이전트를 읽기 전용 작업으로 제한하고 간접 프롬프트 주입 공격의 영향을 제한하는 보안 경계 역할을 합니다. 쓰기 또는 작업 권한이 있는 그룹을 할당하면 프롬프트 주입의 폭발 반경이 크게 증가하여 Azure DevOps 리소스가 손상될 수 있습니다.

프로젝트를 에이전트 스페이스와 연결

계정 수준에서 Azure DevOps를 등록한 후 특정 프로젝트를 에이전트 스페이스와 연결합니다.

1. AWS DevOps 에이전트 콘솔에서 에이전트 스페이스를 선택합니다.
2. 기능 탭으로 이동
3. 파이프라인 섹션에서 추가를 클릭합니다.
4. 사용 가능한 공급자 목록에서 Azure DevOps를 선택합니다.
5. 사용 가능한 프로젝트 드롭다운에서 프로젝트를 선택합니다.
6. 추가를 클릭하여 연결을 완료합니다.

Azure DevOps 연결 관리

- 연결된 프로젝트 보기 - 기능 탭의 파이프라인 섹션에는 연결된 모든 Azure DevOps 프로젝트가 나열됩니다.
- 프로젝트 제거 - 에이전트 공간에서 프로젝트를 연결 해제하려면 파이프라인 섹션에서 프로젝트를 선택하고 제거를 클릭합니다.
- 등록 제거 - Azure DevOps 등록을 완전히 제거하려면 기능 공급자 페이지로 이동하여 등록을 삭제합니다. 먼저 모든 에이전트 스페이스 연결을 제거해야 합니다.

CI/CD 파이프라인에 연결

CI/CD 파이프라인 통합을 통해 AWS DevOps Agent는 배포를 모니터링하고 조사 중에 코드 변경 사항을 운영 인시던트와 연관시킬 수 있습니다. 에이전트는 CI/CD 공급자를 연결하여 배포 이벤트를 추적하고 AWS 리소스와 연결하여 인시던트 대응 중에 잠재적 근본 원인을 식별할 수 있습니다.

AWS DevOps Agent는 2단계 프로세스를 통해 인기 있는 CI/CD 플랫폼과의 통합을 지원합니다.

1. 계정 수준 등록 - AWS 계정 수준에서 CI/CD 공급자를 한 번 등록합니다.
2. 에이전트 스페이스 연결 - 조직의 요구 사항에 따라 특정 프로젝트 또는 리포지토리를 개별 에이전트 스페이스에 연결

이 접근 방식을 사용하면 여러 에이전트 스페이스에서 CI/CD 공급자 등록을 공유하는 동시에 각 스페이스에서 모니터링하는 프로젝트를 세부적으로 제어할 수 있습니다.

지원되는 CI/CD 공급자

AWS DevOps Agent는 다음 CI/CD 플랫폼을 지원합니다.

- GitHub - AWS DevOps 에이전트 GitHub 앱을 사용하여 GitHub.com에서 리포지토리를 연결합니다. GitHub
- GitLab - GitLab.com, 관리형 GitLab 인스턴스 또는 공개적으로 액세스할 수 있는 자체 호스팅 GitLab 배포에서 프로젝트를 연결합니다.

주제

- the section called “GitHub 연결”
- the section called “GitLab 연결”

GitHub 연결

GitHub 통합을 통해 AWS DevOps Agent는 코드 리포지토리에 액세스하고 인시던트 조사 중에 배포 이벤트를 수신할 수 있습니다. 이 통합은 GitHub의 계정 수준 등록 후 특정 리포지토리를 개별 에이전트 스페이스에 연결하는 2단계 프로세스를 따릅니다.

AWS DevOps Agent는 GitHub.com(SaaS) 및 GitHub Enterprise Server(자체 호스팅) 인스턴스를 모두 지원합니다.

사전 조건

GitHub를 연결하기 전에 다음이 있는지 확인합니다.

- AWS DevOps 에이전트 관리자 콘솔에 대한 액세스
- 관리자 권한이 있는 GitHub 사용자 계정 또는 조직
- 계정 또는 조직에 GitHub 앱을 설치하기 위한 권한 부여

GitHub Enterprise Server의 경우 다음 사항도 필요합니다.

- HTTPS를 통해 액세스할 수 있는 GitHub Enterprise Server 인스턴스(버전 3.x 이상)
- GitHub Enterprise Server 인스턴스의 HTTPS URL(예: `https://github.example.com`)
- (선택 사항) GitHub Enterprise Server 인스턴스에 공개적으로 액세스할 수 없는 경우의 프라이빗 연결

GitHub 등록(계정 수준)

GitHub는 AWS 계정 수준에서 등록되고 해당 계정의 모든 에이전트 스페이스 간에 공유됩니다. GitHub는 AWS 계정당 한 번만 등록하면 됩니다.

1단계: 파이프라인 공급자로 이동

1. AWS Management Console에 로그인
2. AWS DevOps 에이전트 콘솔로 이동
3. 기능 탭으로 이동
4. 파이프라인 섹션에서 추가를 클릭합니다.
5. 사용 가능한 공급자 목록에서 GitHub를 선택합니다.

GitHub가 아직 등록되지 않은 경우 먼저 등록하라는 메시지가 표시됩니다.

2단계: 연결 유형 선택

"GitHub 계정/조직 등록" 화면에서 사용자 또는 조직으로 연결할지 여부를 선택합니다.

- 사용자 - 사용자 이름과 프로필이 있는 개인 GitHub 계정
- 조직 - 여러 사람이 여러 프로젝트에서 한 번에 협업할 수 있는 공유 GitHub 계정

GitHub Enterprise Server 인스턴스에 연결하는 경우 GitHub Enterprise Server 사용 확인란을 선택하고 인스턴스의 HTTPS URL(예: `https://github.example.com`)을 입력합니다.

GitHub Enterprise Server 인스턴스에 공개적으로 액세스할 수 없는 경우 선택적으로 프라이빗 연결을 구성하여 AWS DevOps 에이전트가 인스턴스에 안전하게 연결할 수 있도록 할 수 있습니다. 자세한 내용은 the section called “프라이빗 호스팅 도구에 연결” 단원을 참조하십시오.

> **Note**
>
> URL에 `/api/v3` 또는 후행 경로를 포함하지 마십시오. 기본 URL만 입력합니다.

3단계: GitHub 앱 설정

제출을 클릭하여 앱 설정 프로세스를 시작합니다. 다음 단계는 GitHub.com에 연결하는지 아니면 GitHub Enterprise Server에 연결하는지에 따라 다릅니다.

GitHub.com의 경우

1. AWS DevOps 에이전트 GitHub 앱을 설치하기 위해 GitHub로 리디렉션됩니다.
2. 앱을 설치할 계정 또는 조직을 선택합니다.
3. 앱은 AWS DevOps 에이전트가 연결된 리포지토리에서 배포 이벤트를 포함한 이벤트를 수신할 수 있도록 허용합니다.

GitHub Enterprise Server의 경우

GitHub Enterprise Server는 인스턴스에 새 GitHub 앱을 자동으로 설정하는 GitHub 앱 매니페스트 흐름을 사용합니다. 여기에는 GitHub Enterprise Server 인스턴스에 대한 두 개의 리디렉션이 포함됩니다.

1. 브라우저가 GitHub Enterprise Server 인스턴스의 "GitHub 앱 생성" 페이지로 리디렉션됩니다.

2. 앱 이름이 미리 채워져 표시됩니다. 필요에 따라 이름을 자유롭게 변경할 수 있습니다. GitHub 앱 생성을 클릭합니다.
3. 매니페스트 코드를 앱 자격 증명으로 교환하는 AWS DevOps 에이전트로 다시 리디렉션됩니다.

4단계: 리포지토리 선택 및 설치 완료

1. GitHub 앱의 설치 및 권한 부여 페이지가 표시됩니다.
2. 앱이 액세스할 수 있도록 허용할 리포지토리를 선택합니다.
   - 모든 리포지토리 - 모든 현재 및 향후 리포지토리에 대한 액세스 권한 부여
   - 리포지토리만 선택 - 계정 또는 조직에서 특정 리포지토리를 선택합니다.
3. 설치 및 권한 부여를 클릭합니다.
4. AWS DevOps 에이전트 콘솔로 다시 리디렉션됩니다. 여기서 GitHub는 계정 수준에서 등록된 것으로 표시됩니다.

에이전트 스페이스에 리포지토리 연결

계정 수준에서 GitHub를 등록한 후 특정 리포지토리를 개별 에이전트 스페이스에 연결할 수 있습니다.

1. AWS DevOps 에이전트 콘솔에서 에이전트 스페이스를 선택합니다.
2. 기능 탭으로 이동
3. 파이프라인 섹션에서 추가를 클릭합니다.
4. 사용 가능한 공급자 목록에서 GitHub를 선택합니다.
5. 이 에이전트 스페이스와 관련된 리포지토리의 하위 집합 선택
6. 추가를 클릭하여 연결을 완료합니다.

조직의 필요에 따라 서로 다른 리포지토리 세트를 서로 다른 에이전트 스페이스에 연결할 수 있습니다.

GitHub 앱 이해

AWS DevOps 에이전트 GitHub 앱:

- 리포지토리에 대한 읽기 전용 액세스 요청
- 배포 이벤트 및 기타 리포지토리 이벤트를 수신합니다.
- 코드 변경 사항을 운영 인시던트와 연관시킬 수 있는 Allow AWS DevOps Agent
- GitHub 설정을 통해 언제든지 제거할 수 있습니다.

GitHub Enterprise Server의 경우 GitHub 앱은 등록 중에 인스턴스에 자동으로 생성됩니다. 설정 > 애플리케이션 > 설치된 GitHub 앱을 통해 앱의 리포지토리 액세스를 관리하거나 제거할 수 있습니다. 앱 정의를 완전히 삭제하려면 설정 > 개발자 설정 > GitHub 앱으로 이동합니다.

GitHub 연결 관리

- 리포지토리 액세스 업데이트 - GitHub 앱이 액세스할 수 있는 리포지토리를 변경하려면 GitHub 계정 또는 조직 설정(또는 GitHub Enterprise Server 인스턴스 설정)으로 이동하여 설치된 GitHub 앱으로 이동하여 AWS DevOps Agent 앱 구성을 수정합니다.
- 연결된 리포지토리 보기 - AWS DevOps 에이전트 콘솔에서 에이전트 스페이스를 선택하고 기능 탭으로 이동하여 파이프라인 섹션에서 연결된 리포지토리를 봅니다.
- GitHub 연결 제거 - 에이전트 공간에서 GitHub를 연결 해제하려면 파이프라인 섹션에서 연결을 선택하고 제거를 클릭합니다. GitHub 앱을 완전히 제거하려면 GitHub 계정 또는 조직 설정에서 제거합니다. GitHub Enterprise Server의 경우 GitHub 앱은 등록 중에 인스턴스에서 직접 생성되므로 선택적으로 다음 두 가지를 모두 수행하여 앱을 완전히 정리할 수 있습니다.
  - 앱 제거 - 설정 > 애플리케이션 > 설치된 GitHub 앱으로 이동하여 앱에서 구성을 클릭한 다음 제거합니다.
  - 앱 삭제 - 설정 > 개발자 설정 > GitHub 앱으로 이동하여 앱을 선택하고 고급 탭으로 이동한 다음 GitHub 앱 삭제를 선택합니다. 경고: GitHub 앱 삭제는 영구적이며 실행 취소할 수 없습니다. 삭제하는 경우 AWS DevOps 에이전트 콘솔의 처음부터 GitHub Enterprise Server를 다시 등록하여 새 앱을 생성해야 합니다.

GitLab 연결

GitLab 통합을 통해 AWS DevOps Agent는 GitLab Pipelines의 배포를 모니터링하여 인시던트 대응 중에 인과 조사를 알릴 수 있습니다. 이 통합은 GitLab의 계정 수준 등록 후 특정 프로젝트를 개별 에이전트 스페이스에 연결하는 2단계 프로세스를 따릅니다.

GitLab 등록(계정 수준)

GitLab은 AWS 계정 수준에서 등록되고 해당 계정의 모든 에이전트 스페이스 간에 공유됩니다. 그러면 개별 에이전트 스페이스는 에이전트 스페이스에 적용할 특정 프로젝트를 선택할 수 있습니다.

1단계: 파이프라인 공급자로 이동

1. AWS Management Console에 로그인
2. AWS DevOps 에이전트 콘솔로 이동

3. 기능 공급자 페이지로 이동(측면 탐색에서 액세스 가능)
4. 파이프라인 아래의 사용 가능한 공급자 섹션에서 GitLab을 찾고 등록을 클릭합니다.

2단계: GitLab 연결 구성

GitLab 등록 페이지에서 다음을 구성합니다.

연결 유형 - 개인 또는 그룹으로 연결할지 여부를 선택합니다.

- 개인(기본값) - 사용자 이름과 프로필이 있는 개별 GitLab 사용자 계정
- 그룹 - GitLab에서는 그룹을 사용하여 하나 이상의 관련 프로젝트를 동시에 관리합니다.

GitLab 인스턴스 유형 - 연결할 GitLab 인스턴스 유형을 선택합니다.

- GitLab.com(기본값) - 퍼블릭 GitLab 서비스
- 공개적으로 액세스할 수 있는 자체 호스팅 GitLab - GitLab 자체 호스팅 엔드포인트 사용 확인란을 선택하고 GitLab 인스턴스에 URL을 제공합니다.

> **Note**
>
> 현재 공개적으로 액세스할 수 있는 GitLab 인스턴스만 지원됩니다.

액세스 토큰 - GitLab 개인 액세스 토큰을 제공합니다.

1. 별도의 브라우저 탭에서 GitLab 계정에 로그인합니다.
2. 사용자 설정으로 이동하여 액세스 토큰을 선택합니다.
3. 다음 권한을 사용하여 새 개인 액세스 토큰을 생성합니다.
   - `read_repository` - 리포지토리 콘텐츠에 액세스하는 데 필요합니다.
   - `read_virtual_registry` - 가상 레지스트리 정보에 액세스하는 데 필요합니다.
   - `read_registry` - 레지스트리 정보에 액세스하는 데 필요합니다.
   - `api` - 읽기 및 쓰기 API 액세스에 필요합니다.
   - `self_rotate` - 토큰 교체에 필요합니다. 이 기능은 현재 AWS DevOps Agent에서 지원되지 않지만 나중에 지원됩니다. 이제를 추가하면 향후 새 토큰을 생성할 필요가 없습니다.
4. 토큰 만료를 현재 날짜로부터 최대 365일로 설정합니다.

5. 생성된 토큰 복사
6. AWS DevOps 에이전트 콘솔로 돌아가기
7. 토큰을 “토큰 액세스” 필드에 붙여넣습니다.

3단계: 등록 완료

(선택 사항) 태그 - 조직용으로 GitLab 등록에 AWS 태그를 추가합니다.

다음을 클릭하여 구성을 검토한 다음 제출을 클릭하여 GitLab 등록 프로세스를 완료합니다. 시스템에서 액세스 토큰을 검증하고 연결을 설정합니다.

에이전트 스페이스에 프로젝트 연결

계정 수준에서 GitLab을 등록한 후 특정 프로젝트를 개별 에이전트 스페이스에 연결할 수 있습니다.

1. AWS DevOps 에이전트 콘솔에서 에이전트 스페이스를 선택합니다.
2. 기능 탭으로 이동
3. 파이프라인 섹션에서 추가를 클릭합니다.
4. 사용 가능한 공급자 목록에서 GitLab을 선택합니다.
5. 에이전트 스페이스와 관련된 GitLab 프로젝트를 선택합니다.
6. 저장을 클릭합니다.

AWS DevOps Agent는 이러한 프로젝트의 GitLab Pipelines 배포를 모니터링하여 인과 조사를 알립니다.

GitLab 연결 관리

- 액세스 토큰 업데이트 - 액세스 토큰이 만료되거나 업데이트해야 하는 경우 계정 수준에서 GitLab 등록을 수정하여 AWS DevOps 에이전트 콘솔에서 업데이트할 수 있습니다.
- 연결된 프로젝트 보기 - AWS DevOps 에이전트 콘솔에서 에이전트 스페이스를 선택하고 기능 탭으로 이동하여 파이프라인 섹션에서 연결된 프로젝트를 봅니다.
- GitLab 연결 제거 - 에이전트 공간에서 GitLab 프로젝트를 연결 해제하려면 파이프라인 섹션에서 연결을 선택하고 제거를 클릭합니다. GitLab 등록을 완전히 제거하려면 먼저 모든 에이전트 스페이스에서 제거한 다음 계정 수준에서 등록을 삭제합니다.

MCP 서버 연결

모델 컨텍스트 프로토콜(MCP) 서버는 외부 관찰성 도구, 사용자 지정 모니터링 시스템 및 운영 데이터 소스의 데이터에 대한 액세스를 제공하여 AWS DevOps 에이전트의 조사 기능을 확장합니다. 이 가이드에서는 MCP 서버를 AWS DevOps Agent에 연결하는 방법을 설명합니다.

요구 사항

MCP 서버를 연결하기 전에 서버가 다음 요구 사항을 충족하는지 확인합니다.

- 스트리밍 가능한 HTTP 전송 프로토콜 - 스트리밍 가능한 HTTP 전송 프로토콜을 구현하는 MCP 서버만 지원됩니다.
- 인증 지원 - MCP 서버는 OAuth 2.0 인증 흐름 또는 API 키/토큰 기반 인증을 지원해야 합니다.

보안 고려 사항

MCP 서버를 AWS DevOps Agent에 연결할 때 다음 보안 측면을 고려하세요.

- 도구 허용 목록 - MCP 서버의 모든 도구를 노출하는 대신 에이전트 스페이스에 필요한 특정 도구만 허용 목록으로 지정해야 합니다. 에이전트 공간당 목록 도구를 허용하는 방법은 에이전트 공간에서 MCP 도구 구성을 참조하세요.

MCP 도구의 최대 도구 길이는 64입니다.

- 프롬프트 주입 위험 - 사용자 지정 MCP 서버는 프롬프트 주입 공격의 추가 위험을 초래할 수 있습니다. 자세한 내용은 프롬프트 주입 방지: AWS DevOps 에이전트 보안을 참조하세요.
- 읽기 전용 도구 및 액세스 - 읽기 전용 MCP 도구만 허용하고 인증 자격 증명이 읽기 전용 액세스만 허용되도록 합니다.

프롬프트 주입 및 공동 책임 모델에 대한 AWS DevOps 에이전트 보안 자세한 내용은 섹션을 참조하세요.

> **Note**
>
> MCP 서버가 프라이빗 네트워크에 있는 경우 섹션을 참조하세요. the section called “프라이빗 호스팅 도구에 연결”

MCP 서버 등록(계정 수준)

MCP 서버는 AWS 계정 수준에서 등록되고 해당 계정의 모든 에이전트 스페이스 간에 공유됩니다. 그러면 개별 에이전트 스페이스가 각 MCP 서버에 필요한 특정 도구를 선택할 수 있습니다.

1단계: MCP 서버 세부 정보

1. AWS Management Console에 로그인
2. AWS DevOps 에이전트 콘솔로 이동
3. 기능 공급자 페이지로 이동(측면 탐색에서 액세스 가능)
4. 사용 가능한 공급자 섹션에서 MCP 서버를 찾아 등록을 클릭합니다.
5. MCP 서버 세부 정보 페이지에서 다음 정보를 입력합니다.
   - 이름 - MCP 서버의 설명 이름을 입력합니다.
   - 엔드포인트 URL - MCP 서버 엔드포인트의 전체 HTTPS URL을 입력합니다.
   - 설명(선택 사항) - 서버의 목적을 식별하는 데 도움이 되는 설명을 추가합니다.
   - 동적 클라이언트 등록 활성화 - MCP 서버의 권한 부여 서버에 AWS DevOps 에이전트가 자동으로 등록하도록 허용하려면이 확인란을 선택합니다.
6. 다음을 클릭합니다.

> **Note**
>
> MCP 서버 엔드포인트 URL은 계정의 AWS CloudTrail 로그에 표시됩니다.

2단계: 권한 부여 흐름

MCP 서버의 인증 방법을 선택합니다.

OAuth 클라이언트 자격 증명 - MCP 서버가 OAuth 클라이언트 자격 증명 흐름을 사용하는 경우:

1. OAuth 클라이언트 자격 증명 선택
2. 다음을 클릭합니다.

OAuth 3LO(3레깅 OAuth) - MCP 서버가 인증에 OAuth 3LO를 사용하는 경우:

1. OAuth 3LO 선택

2. 다음을 클릭합니다.

API 키 - MCP 서버가 API 키 인증을 사용하는 경우:

1. API 키 선택
2. 다음을 클릭합니다.

3단계: 권한 부여 구성

선택한 인증 방법을 기반으로 추가 권한 부여 파라미터를 구성합니다.

OAuth 클라이언트 자격 증명의 경우:

1. 클라이언트 ID - OAuth 클라이언트의 클라이언트 ID를 입력합니다.
2. 클라이언트 보안 암호 - OAuth 클라이언트의 클라이언트 보안 암호를 입력합니다.
3. Exchange URL - OAuth 토큰 교환 엔드포인트 URL을 입력합니다.
4. Exchange 파라미터 - 서비스 인증을 위한 OAuth 토큰 교환 파라미터를 입력합니다.
5. 범위 추가 - 인증을 위한 OAuth 범위 추가
6. 다음을 클릭합니다.

OAuth 3LO의 경우:

1. 클라이언트 ID - OAuth 클라이언트의 클라이언트 ID를 입력합니다.
2. 클라이언트 보안 암호 - OAuth 클라이언트에 필요한 경우 OAuth 클라이언트의 클라이언트 보안 암호를 입력합니다.
3. Exchange URL - OAuth 토큰 교환 엔드포인트 URL을 입력합니다.
4. 권한 부여 URL - OAuth 권한 부여 엔드포인트 URL을 입력합니다.
5. 코드 챌린지 지원 - OAuth 클라이언트가 코드 챌린지를 지원하는 경우이 확인란을 선택합니다.
6. 범위 추가 - 인증을 위한 OAuth 범위 추가
7. 다음을 클릭합니다.

API 키의 경우:

1. API 키 이름 입력

2. 요청에 API 키를 포함할 헤더의 이름을 입력합니다.
3. API 키 값 입력
4. 다음을 클릭합니다.

4단계: 검토 및 제출

1. 모든 MCP 서버 구성 세부 정보 검토
2. 제출을 클릭하여 등록을 완료합니다.
3. AWS DevOps 에이전트가 MCP 서버에 대한 연결을 검증합니다.
4. 검증에 성공하면 MCP 서버가 계정 수준에서 등록됩니다.

에이전트 스페이스에서 MCP 도구 구성

계정 수준에서 MCP 서버를 등록한 후 해당 서버에서 특정 에이전트 스페이스에 사용할 수 있는 도구를 구성할 수 있습니다.

1. AWS DevOps 에이전트 콘솔에서 에이전트 스페이스를 선택합니다.
2. 기능 탭으로 이동
3. MCP 서버 섹션에서 추가를 클릭합니다.
4. 이 에이전트 스페이스에 연결할 등록된 MCP 서버를 선택합니다.
5. 이 MCP 서버의 도구를 에이전트 스페이스에서 사용할 수 있도록 구성합니다.
   - 모든 도구 허용 - MCP 서버의 모든 도구를 사용할 수 있도록 설정합니다.
   - 특정 도구 선택 - 허용 목록에 추가할 도구를 선택할 수 있습니다.
6. 추가를 클릭하여 MCP 서버를 에이전트 스페이스에 연결합니다.

이제AWS DevOps 에이전트는이 에이전트 스페이스에서 조사하는 동안 MCP 서버의 허용 목록에 있는 도구를 사용할 수 있습니다.

MCP 서버 연결 관리

인증 자격 증명 업데이트 - 인증 자격 증명을 업데이트해야 하는 경우 MCP 서버를 다시 등록해야 합니다. AWS DevOps 에이전트 콘솔의 기능 공급자 페이지로 이동하여 MCP 서버를 찾고 활성 연결을 제거한 다음 등록 취소를 클릭합니다. 그런 다음 새 인증 자격 증명으로 MCP 서버를 등록하고 에이전트 스페이스와 필요한 연결을 다시 생성합니다.

연결된 MCP 서버 보기 - 에이전트 스페이스에 연결된 모든 MCP 서버를 보려면 에이전트 스페이스를 선택하고 기능 탭으로 이동하여 MCP 서버 섹션을 확인합니다. 여기에서 선택한 도구를 업데이트할 수도 있습니다.

MCP 서버 연결 제거 - 에이전트 공간에서 MCP 서버를 연결 해제하려면 MCP 서버 섹션에서 서버를 선택하고 제거를 클릭합니다. MCP 서버 등록을 완전히 삭제하려면 먼저 모든 에이전트 스페이스에서 제거한 다음 계정 수준 등록을 삭제합니다.

관련 주제

- in AWS DevOps 에이전트의 보안
- 에이전트 스페이스 설정
- 프롬프트 주입 보호

여러 AWS 계정 연결

보조 AWS 계정을 사용하면 AWS DevOps Agent가 조직의 여러 AWS 계정에서 리소스를 조사할 수 있습니다. 애플리케이션이 여러 계정에 걸쳐 있는 경우 보조 계정을 추가하면 인시던트 조사 중에 에이전트가 모든 관련 리소스를 볼 수 있습니다. 애플리케이션을 구성하는 계정 및 리소스에 대한 액세스 권한이 높을수록 조사 정확도가 높아집니다.

사전 조건

보조 AWS 계정을 추가하기 전에 다음이 있는지 확인합니다.

- 기본 계정의 AWS DevOps 에이전트 콘솔에 대한 액세스
- 보조 AWS 계정에 대한 관리 액세스
- 보조 계정에서 역할을 생성할 수 있는 IAM 권한

보조 AWS 계정 추가

아래 단계 외에도를 사용하여 보조 계정을 the section called “AWS DevOps Agent CLI 온보딩 가이드” 프로그래밍 방식으로 추가할 수 있습니다.

1단계: 보조 계정 구성 시작

1. AWS Management Console에 로그인하고 AWS DevOps 에이전트 콘솔로 이동합니다.

2. 에이전트 스페이스 선택
3. 기능 탭으로 이동
4. 클라우드 섹션에서 보조 소스 하위 섹션을 찾습니다.
5. 추가를 클릭합니다.

2단계: 역할 이름 지정

1. 역할 이름 지정 필드에 보조 계정에서 생성할 역할의 이름을 입력합니다.
2. 이 이름 참고 - 보조 계정에서 역할을 생성할 때 다시 사용합니다.
3. 콘솔에 제공된 신뢰 정책을 복사하여 스크래치 공간에 저장합니다.

3단계: 보조 계정에서 역할 생성

1. 새 브라우저 탭을 열고 보조 AWS 계정의 IAM 콘솔에 로그인합니다.
2. IAM > 역할 > 역할 생성으로 이동합니다.
3. 사용자 지정 신뢰 정책 선택
4. 2단계에서 복사한 신뢰 정책 붙여넣기
5. 다음을 클릭합니다.

4단계: AWS 관리형 정책 연결

1. 권한 정책 섹션에서 AIOpsAssistantPolicy를 검색합니다.
2. AIOpsAssistantPolicy 관리형 정책 옆의 확인란을 선택합니다.
3. 다음을 클릭합니다.

5단계: 역할 이름 지정 및 생성

1. 역할 이름 필드에 2단계에서 제공한 것과 동일한 역할 이름을 입력합니다.
2. (선택 사항) 역할의 목적을 식별하는 데 도움이 되는 설명을 추가합니다.
3. 신뢰 정책 및 연결된 권한 검토
4. 역할 생성을 클릭합니다.

6단계: 인라인 정책 연결

1. IAM 콘솔에서 방금 생성한 역할을 찾아 선택합니다.
2. 권한 탭으로 이동
3. 권한 추가 > 인라인 정책 생성을 클릭합니다.
4. JSON 탭으로 전환
5. 2단계에서 저장한 정책 붙여넣기
6. IAM 콘솔의 JSON 편집기에 정책 붙여넣기
7. 다음을 클릭합니다.
8. 인라인 정책의 이름을 입력합니다(예: "DevOpsAgentInlinePolicy").
9. 정책 생성을 클릭합니다.

7단계: 구성 완료

1. 기본 계정의 AWS DevOps 에이전트 콘솔로 돌아가기
2. 다음을 클릭하여 보조 계정 구성을 완료합니다.
3. 연결 상태가 활성으로 표시되는지 확인

필수 정책 이해

AWS DevOps Agent는 보조 계정의 리소스에 액세스하려면 세 가지 정책 구성 요소가 필요합니다.

- 신뢰 정책 - 기본 계정의 AWS DevOps 보조 계정의 역할을 수임하도록 허용합니다. 이렇게 하면 계정 간의 신뢰 관계가 설정됩니다.
- AIOpsAssistantPolicy(AWS 관리형 정책) - 보조 계정의 리소스를 조사하는 데 필요한 핵심 읽기 전용 권한 AWS DevOps Agent를 제공합니다. 이 정책은에서 유지 AWS 관리하며 새 기능이 추가되면 업데이트됩니다.
- 인라인 정책 - 에이전트 공간 구성과 관련된 추가 권한을 제공합니다. 이 정책은 에이전트 공간 설정을 기반으로 생성되며 특정 통합 또는 기능에 대한 권한을 포함할 수 있습니다.

기본 계정에서 AWS DevOps 에이전트 IAM 역할은 보조 계정에서 생성된 역할을 수임할 수 있어야 합니다.

보조 계정 관리

- 연결된 계정 보기 - 기능 탭의 보조 소스 하위 섹션에는 연결된 모든 보조 계정이 연결 상태로 나열됩니다.
- IAM 역할 업데이트 - 권한을 수정해야 하는 경우 보조 계정의 역할에 연결된 인라인 정책을 업데이트합니다. 변경 사항은 즉시 적용됩니다.
- 보조 계정 제거 - 보조 계정의 연결을 해제하려면 보조 소스 목록에서 해당 계정을 선택하고 제거를 클릭합니다. 이렇게 해도 보조 계정의 IAM 역할은 삭제되지 않습니다.

원격 측정 소스 연결

AWS DevOps Agent는 원격 측정 소스에 연결하는 세 가지 방법을 제공합니다.

기본 제공 양방향 통합

현재 AWS DevOps 에이전트는 다음을 지원하는 기본 제공 양방향 통합을 통해 Dynatrace 사용자를 지원합니다.

- 토폴로지 리소스 매핑 - AWS DevOps 에이전트는 DevOps 에이전트가 호스팅하는 Dynatrace MCP 서버를 통해 사용할 수 있는 엔터티 및 관계로 AWS DevOps 에이전트 스페이스 토폴로지를 강화합니다.
- 자동 조사 트리거 - Dynatrace 문제에서 인시던트 해결 조사를 트리거하도록 Dynatrace 워크플로를 구성할 수 있습니다.
- 원격 측정 내부 검사 - AWS DevOps 에이전트는 AWS DevOps 에이전트 호스팅 Dynatrace MCP 서버를 통해 문제를 조사할 때 Dynatrace 원격 측정을 내부 검사할 수 있습니다.
- 상태 업데이트 - AWS DevOps Agent는 주요 조사 결과, 근본 원인 분석 및 생성된 완화 계획을 Dynatrace 사용자 인터페이스에 게시합니다.

양방향 통합에 대한 자세한 내용은 섹션을 참조하세요.

- the section called “Dynatrace 연결”

기본 제공 단방향 통합

현재 AWS DevOps 에이전트는 내장된 단방향 통합을 통해 AWS CloudWatch, Datadog, Grafana, New Relic 및 Splunk 사용자를 지원합니다.

보안 모범 사례: 기본 제공 단방향 통합을 위한 자격 증명을 구성할 때는 API 키와 토큰을 읽기 전용 액세스로 조정하는 것이 좋습니다. AWS DevOps Agent는 원격 측정 내부 검사에만 이러한 자격 증명을 사용하며 원격 측정 공급자에 대한 쓰기 액세스가 필요하지 않습니다.

AWS CloudWatch 기본 제공 단방향 통합은 추가 설정이 필요하지 않으며 다음을 활성화합니다.

- 토폴로지 리소스 매핑 - AWS DevOps 에이전트는 구성된 기본 및 보조 AWS 클라우드 계정을 통해 사용할 수 있는 엔터티 및 관계로 DevOps 에이전트 스페이스 토폴로지를 강화합니다.
- 원격 측정 내부 검사 - AWS DevOps Agent는 기본 및 보조 AWS 클라우드 계정 구성 중에 제공된 IAM 역할(들)을 통해 문제를 조사할 때 AWS CloudWatch 원격 측정을 내부 검사할 수 있습니다.

Datadog, Grafana, New Relic 및 Splunk 기본 제공 단방향 통합에는 다음을 설정하고 활성화해야 합니다.

- 자동 조사 트리거 - Datadog, Grafana, New Relic 및 Splunk 이벤트를 AWS DevOps 에이전트 웹후크를 통해 AWS DevOps 에이전트 인시던트 해결 조사를 트리거하도록 구성할 수 있습니다.
- 원격 측정 내부 검사 - AWS DevOps Agent는 각 공급자의 원격 MCP 서버를 통해 문제를 조사할 때 Datadog, Grafana, New Relic 및 Splunk 원격 측정을 내부 검사할 수 있습니다.

단방향 통합에 대한 자세한 내용은 다음을 참조하세요.

- the section called “DataDog 연결”
- the section called “Grafana 연결”
- the section called “새 복제본 연결”
- the section called “Splunk 연결”

Bring-your-own

Prometheus 지표를 포함한 다른 원격 측정 소스의 경우 웹후크와 MCP 서버 통합 모두에 대한 AWS DevOps 에이전트의 지원을 활용할 수 있습니다.

bring-your-own 통합에 대한 자세한 내용은 다음을 참조하세요.

- the section called “Webhook를 통해 DevOps 에이전트 호출”
- the section called “MCP 서버 연결”

Dynatrace 연결

기본 제공 양방향 통합

현재 AWS DevOps 에이전트는 다음을 지원하는 기본 제공 양방향 통합을 통해 Dynatrace 사용자를 지원합니다.

- 토폴로지 리소스 매핑 - AWS DevOps 에이전트는 Dynatrace 환경에서 사용할 수 있는 엔터티 및 관계로 DevOps 에이전트 스페이스 토폴로지를 강화합니다.
- 자동 조사 트리거 - Dynatrace 문제에서 인시던트 해결 조사를 트리거하도록 Dynatrace 워크플로를 구성할 수 있습니다.
- 원격 측정 내부 검사 - AWS DevOps 에이전트는 AWS DevOps 에이전트 호스팅 Dynatrace MCP 서버를 통해 문제를 조사할 때 Dynatrace 원격 측정을 내부 검사할 수 있습니다.
- 상태 업데이트 - AWS DevOps Agent는 주요 조사 결과, 근본 원인 분석 및 생성된 완화 계획을 Dynatrace 사용자 인터페이스에 게시합니다.

온보딩

온보딩 프로세스

Dynatrace 관찰성 시스템 온보딩에는 세 단계가 포함됩니다.

1. 연결 - 필요한 모든 환경에서 계정 액세스 자격 증명을 구성하여 Dynatrace에 대한 연결을 설정합니다.
2. 활성화 - 특정 Dynatrace 환경의 특정 에이전트 공간에서 Dynatrace 활성화
3. Dynatrace 환경 구성 - 워크플로 및 대시보드를 다운로드하고 Dynatrace로 가져와 지정된 에이전트 공간에서 조사를 트리거하기 위한 웹후크 세부 정보를 기록해 둡니다.

1단계: 연결

Dynatrace 환경에 대한 연결 설정

구성

1. 기능 공급자 페이지로 이동(측면 탐색에서 액세스 가능)
2. 텔레메트리 아래의 사용 가능한 공급자 섹션에서 Dynatrace를 찾고 등록을 클릭합니다.
3. 자세한 권한을 사용하여 Dynatrace에서 OAuth 클라이언트를 생성합니다.

a. Dynatrace 설명서 참조
b. 준비가 되면 다음을 누릅니다.
c. 여러 Dynatrace 환경을 연결하고 나중에 보유한 각 DevOps 에이전트 스페이스의 특정 환경에 범위를 지정할 수 있습니다.

4. OAuth 클라이언트 설정에서 Dynatrace 세부 정보를 입력합니다.
   - 클라이언트 이름
   - 클라이언트 ID
   - 클라이언트 보안 암호
   - 계정 URN
5. 다음을 클릭합니다.
6. 검토 및 추가

2단계: 활성화

특정 에이전트 공간에서 Dynatrace를 활성화하고 적절한 범위 지정을 구성합니다.

구성

1. 에이전트 공간 페이지에서 에이전트 공간을 선택하고 세부 정보 보기를 누릅니다.
2. 기능 탭을 선택합니다.
3. 텔레메트리 섹션을 찾아 추가를 누릅니다.
4. '등록됨' 상태의 Dynatrace가 표시됩니다. 추가를 클릭하여 에이전트 스페이스에 추가합니다.
5. Dynatrace 환경 ID -이 DevOps 에이전트 공간과 연결할 Dynatrace 환경 ID를 제공합니다.
6. 하나 이상의 Dynatrace 엔터티 IDs 입력합니다. 이를 통해 DevOps 에이전트가 가장 중요한 리소스를 검색할 수 있습니다. 예를 들면 서비스 또는 애플리케이션일 수 있습니다. 확실하지 않은 경우 제거를 누릅니다.
7. 검토 후 저장을 누릅니다.
8. Webhook URL 및 Webhook 보안 암호를 복사합니다. 이러한 자격 증명을 Dynatrace에 추가하려면 Dynatrace 설명서를 참조하세요.

3단계: Dynatrace 환경 구성

Dynatrace 설정을 완료하려면 Dynatrace 환경에서 특정 설정 단계를 수행해야 합니다. Dynatrace 설명서의 지침을 따릅니다.

지원되는 이벤트 스키마

AWS DevOps 에이전트는 웹후크를 사용하여 Dynatrace의 두 가지 유형의 이벤트를 지원합니다. 지원되는 이벤트 스키마는 아래에 설명되어 있습니다.

인시던트 이벤트

인시던트 이벤트는 조사를 트리거하는 데 사용됩니다. 이벤트 스키마는 다음과 같습니다.

```
{
    "event.id": string;
    "event.status": "ACTIVE" | "CLOSED";
    "event.status_transition": string;
    "event.description": string;
    "event.name": string;
    "event.category": "AVAILABILITY" | "ERROR" | "SLOWDOWN" | "RESOURCE_CONTENTION" |
 "CUSTOM_ALERT" | "MONITORING_UNAVAILABLE" | "INFO";
    "event.start"?: string;
    "affected_entity_ids"?: string[];
}
```

완화 이벤트

완화 이벤트는 다음 단계에 대한 조사를 위한 완화 보고서 생성을 트리거하는 데 사용됩니다. 이벤트 스키마는 다음과 같습니다.

```
{
    "task_id": string;
    "task_version": number;
    "event.type": "mitigation_request";
}
```

제거

원격 측정 소스는 에이전트 공간 수준과 계정 수준에서 두 가지 수준으로 연결됩니다. 완전히 제거하려면 먼저 에이전트 공간이 사용되는 모든 에이전트 공간에서 제거한 다음 등록을 취소할 수 있습니다.

1단계: 에이전트 공간에서 제거

1. 에이전트 공간 페이지에서 에이전트 공간을 선택하고 세부 정보 보기를 누릅니다.
2. 기능 탭을 선택합니다.

3. 아래로 스크롤하여 원격 측정 섹션으로 이동합니다.
4. Dynatrace 선택
5. 제거를 누릅니다.

2단계: 계정에서 등록 취소

1. 기능 공급자 페이지로 이동(측면 탐색에서 액세스 가능)
2. 현재 등록된 섹션으로 스크롤합니다.
3. 에이전트 공간 수가 0인지 확인합니다(다른 에이전트 공간에서 위의 1단계를 반복하지 않는 경우).
4. Dynatrace 옆의 등록 취소를 누릅니다.

DataDog 연결

기본 제공, 단방향 통합

현재 AWS DevOps 에이전트는 기본 제공 단방향 통합을 통해 Datadog 사용자를 지원하므로 다음을 사용할 수 있습니다.

- 자동 조사 트리거링 - AWS DevOps 에이전트 웹후크를 통해 AWS DevOps 에이전트 인시던트 해결 조사를 트리거하도록 Datadog 이벤트를 구성할 수 있습니다.
- 원격 측정 내부 검사 - AWS DevOps Agent는 각 공급자의 원격 MCP 서버를 통해 문제를 조사할 때 Datadog 원격 측정을 내부 검사할 수 있습니다.

온보딩

1단계: 연결

계정 액세스 자격 증명을 사용하여 Datadog 원격 MCP 엔드포인트에 대한 연결 설정

구성

1. 기능 공급자 페이지로 이동(측면 탐색에서 액세스 가능)
2. 텔레메트리 아래의 사용 가능한 공급자 섹션에서 Datadog을 찾고 등록을 클릭합니다.
3. Datadog MCP 서버 세부 정보를 입력합니다.
   - 서버 이름 - 고유 식별자(예: my-datadog-server)

- 엔드포인트 URL - Datadog MCP 서버 엔드포인트입니다. 엔드포인트 URL은 Datadog 사이트에 따라 다릅니다. 아래 Datadog 사이트 엔드포인트 표를 참조하세요.
- 설명 - 선택적 서버 설명

4. 다음을 클릭합니다.
5. 검토 및 제출

Datadog 사이트 엔드포인트

MCP 엔드포인트 URL은 Datadog 사이트에 따라 다릅니다. 사이트를 식별하려면 Datadog에 로그인할 때 브라우저의 URL을 확인하거나 Datadog 사이트 액세스를 참조하세요.

Datadog 사이트	사이트 도메인	MCP 엔드포인트 URL
US1(기본값)	`datadoghq.com`	`https://mcp.datadoghq.com/api/unstable/mcp-server/mcp`
US3	`us3.datadoghq.com`	`https://mcp.us3.datadoghq.com/api/unstable/mcp-server/mcp`
US5	`us5.datadoghq.com`	`https://mcp.us5.datadoghq.com/api/unstable/mcp-server/mcp`
EU1	`datadoghq.eu`	`https://mcp.datadoghq.eu/api/unstable/mcp-server/mcp`
AP1	`ap1.datadoghq.com`	`https://mcp.ap1.datadoghq.com/api/unstable/mcp-server/mcp`

Datadog 사이트	사이트 도메인	MCP 엔드포인트 URL
AP2	`ap2.datadoghq.com`	`https://mcp.ap2.datadoghq.com/api/unstable/mcp-server/mcp`

권한 부여

다음을 통해 OAuth 권한 부여 완료:

- Datadog OAuth 페이지에서 사용자 권한 부여
- 로그인하지 않은 경우 허용, 로그인을 클릭한 다음 권한 부여를 클릭합니다.

구성되면 모든 에이전트 스페이스에서 Datadog을 사용할 수 있게 됩니다.

2단계: 활성화

특정 에이전트 공간에서 DataDog를 활성화하고 적절한 범위 지정을 구성합니다.

구성

1. 에이전트 스페이스 페이지에서 에이전트 스페이스를 선택하고 세부 정보 보기를 누릅니다(에이전트 스페이스를 아직 생성하지 않은 경우 참조the section called “에이전트 스페이스 생성”).
2. 기능 탭을 선택합니다.
3. 아래로 스크롤하여 원격 측정 섹션으로 이동합니다.
4. 추가를 누릅니다.
5. Datadog 선택
6. 다음
7. 검토 후 저장을 누릅니다.
8. Webhook URL 및 API 키 복사

3단계: 웹후크 구성

Webhook URL 및 API 키를 사용하여 예를 들어 경보에서 조사를 트리거하는 이벤트를 보내도록 Datadog을 구성할 수 있습니다.

DevOps 에이전트가 전송된 이벤트를 사용할 수 있도록 하려면 웹후크로 전송된 데이터가 아래에 지정된 데이터 스키마와 일치하는지 확인합니다. 이 스키마와 일치하지 않는 이벤트는 DevOps Agent에서 무시할 수 있습니다.

메서드 및 헤더 설정

```
    method: "POST",
    headers: {
      "Content-Type": "application/json",
      "Authorization": "Bearer <Token>",
    },
```

본문을 JSON 문자열로 전송합니다.

```
{
    eventType: 'incident';
    incidentId: string;
    action: 'created' | 'updated' | 'closed' | 'resolved';
    priority: "CRITICAL" | "HIGH" | "MEDIUM" | "LOW" | "MINIMAL";
    title: string;
    description?: string;
    timestamp?: string;
    service?: string;
    // The original event generated by service is attached here.
    data?: object;
}
```

Datadog https://docs.datadoghq.com/integrations/webhooks/ 웹후크를 전송합니다(권한 부여 없음을 선택하고 대신 사용자 지정 헤더 옵션을 사용).

자세히 알아보기: Datadog 원격 MCP 서버

제거

원격 측정 소스는 에이전트 공간 수준과 계정 수준에서 두 가지 수준으로 연결됩니다. 완전히 제거하려면 먼저 에이전트 공간이 사용되는 모든 에이전트 공간에서 제거한 다음 등록을 취소할 수 있습니다.

1단계: 에이전트 공간에서 제거

1. 에이전트 공간 페이지에서 에이전트 공간을 선택하고 세부 정보 보기를 누릅니다.
2. 기능 탭을 선택합니다.
3. 아래로 스크롤하여 원격 측정 섹션으로 이동합니다.

4. Datadog 선택
5. 제거를 누릅니다.

2단계: 계정에서 등록 취소

1. 기능 공급자 페이지로 이동(측면 탐색에서 액세스 가능)
2. 현재 등록된 섹션으로 스크롤합니다.
3. 에이전트 공간 수가 0인지 확인합니다(다른 에이전트 공간에서 위의 1단계를 반복하지 않는 경우).
4. Datadog 옆의 등록 취소를 누릅니다.

Grafana 연결

Grafana 통합을 통해 AWS DevOps Agent는 인시던트 조사 중에 Grafana 인스턴스에서 지표, 대시보드 및 알림 데이터를 쿼리할 수 있습니다. 이 통합은 Grafana의 계정 수준 등록 후 개별 에이전트 스페이스에 연결하는 2단계 프로세스를 따릅니다.

보안을 개선하기 위해 Grafana 통합은 읽기 전용 도구만 활성화합니다. 쓰기 도구는 비활성화되어 있으며 활성화할 수 없습니다. 즉, 에이전트는 Grafana 인스턴스에서 데이터를 쿼리하고 읽을 수 있지만 대시보드, 알림 또는 주석과 같은 Grafana 리소스를 생성, 수정 또는 삭제할 수는 없습니다. 자세한 내용은 Security in AWS DevOps Agent를 참조하세요.

Grafana 요구 사항

Grafana를 연결하기 전에 다음을 확인하세요.

- Grafana 버전 9.0 이상. 일부 기능, 특히 데이터 소스 관련 작업은 API 엔드포인트 누락으로 인해 이전 버전에서 제대로 작동하지 않을 수 있습니다.
- HTTPS를 통해 액세스할 수 있는 Grafana 인스턴스입니다. 퍼블릭 및 프라이빗 네트워크 엔드포인트가 모두 지원됩니다. 프라이빗 네트워크 연결을 사용하면 퍼블릭 인터넷 액세스 없이 VPC 내에서 Grafana 인스턴스를 호스팅할 수 있습니다. 자세한 내용은 the section called “프라이빗 호스팅 도구에 연결”을 참조하세요.
- 적절한 읽기 권한이 있는 액세스 토큰이 있는 Grafana 서비스 계정

Grafana 등록(계정 수준)

Grafana는 AWS 계정 수준에서 등록되며 해당 계정의 모든 에이전트 스페이스 간에 공유됩니다.

1단계: Grafana 구성

1. AWS Management Console에 로그인
2. AWS DevOps 에이전트 콘솔로 이동
3. 기능 공급자 페이지로 이동(측면 탐색에서 액세스 가능)
4. 텔레메트리 아래의 사용 가능한 공급자 섹션에서 Grafana를 찾고 등록을 클릭합니다.
5. Grafana 구성 페이지에서 다음 정보를 입력합니다.
   - 서비스 이름(필수) - 영숫자, 하이픈 및 밑줄만 사용하여 Grafana 서버의 설명 이름을 입력합니다. 예를 들어 `my-grafana-server`입니다.
   - Grafana URL(필수) - Grafana 인스턴스의 전체 HTTPS URL을 입력합니다. 예를 들어 `https://myinstance.grafana.net`입니다.
   - 서비스 계정 액세스 토큰(필수) - Grafana 서비스 계정 액세스 토큰을 입력합니다. 토큰은 일반적으로 로 시작합니다`glsa_`. 서비스 계정 토큰을 생성하려면 Grafana 인스턴스로 이동하여 관리 > 서비스 계정으로 이동하여 최종 사용자 역할이 있는 서비스 계정을 생성하고 토큰을 생성합니다.
   - 설명(선택 사항) - 서버의 목적을 식별하는 데 도움이 되는 설명을 추가합니다. 예를 들어 `Production Grafana server for monitoring`입니다.
6. (선택 사항) 조직의 목적으로 등록에 AWS 태그를 추가합니다.
7. 다음을 클릭합니다.

2단계: Grafana 등록 검토 및 제출

1. 모든 Grafana 구성 세부 정보 검토
2. 제출을 클릭하여 등록을 완료합니다.
3. 등록에 성공하면 Grafana가 기능 공급자 페이지의 현재 등록된 섹션에 나타납니다.

에이전트 스페이스에 Grafana 추가

계정 수준에서 Grafana를 등록한 후 개별 에이전트 스페이스에 연결할 수 있습니다.

1. AWS DevOps 에이전트 콘솔에서 에이전트 스페이스를 선택합니다.
2. 기능 탭으로 이동
3. 원격 측정 섹션에서 추가를 클릭합니다.
4. 사용 가능한 공급자 목록에서 Grafana를 선택합니다.

5. 저장을 클릭합니다.

Grafana 알림 웹후크 구성

Grafana 연락 지점을 통해 웹후크를 전송하여 알림이 실행될 때 자동으로 AWS DevOps 에이전트 조사를 트리거하도록 Grafana를 구성할 수 있습니다. 웹후크 인증 방법 및 자격 증명 관리에 대한 자세한 내용은 섹션을 참조하세요the section called “Webhook를 통해 DevOps 에이전트 호출”.

1단계: 사용자 지정 알림 템플릿 생성

Grafana 인스턴스에서 알림 > 연락 지점 > 알림 템플릿으로 이동하여 다음 콘텐츠가 포함된 새 템플릿을 생성합니다.

```
{{ define "devops-agent-payload" }}
{
  "eventType": "incident",
  "incidentId": "{{ (index .Alerts 0).Labels.alertname }}-{{ (index .Alerts
 0).Fingerprint }}",
  "action": "{{ if eq .Status "resolved" }}resolved{{ else }}created{{ end }}",
  "priority": "{{ if eq .Status "resolved" }}MEDIUM{{ else }}HIGH{{ end }}",
  "title": "{{ (index .Alerts 0).Labels.alertname }}",
  "description": "{{ (index .Alerts 0).Annotations.summary }}",
  "service": "{{ if (index .Alerts 0).Labels.job }}{{ (index .Alerts 0).Labels.job }}
{{ else }}grafana{{ end }}",
  "timestamp": "{{ (index .Alerts 0).StartsAt }}",
  "data": {
    "metadata": {
      {{ range $k, $v := (index .Alerts 0).Labels }}
      "{{ $k }}": "{{ $v }}",
      {{ end }}
      "_source": "grafana"
    }
  }
}
{{ end }}
```

이 템플릿은 Grafana 알림을 AWS DevOps Agent에서 예상하는 웹후크 페이로드 구조로 포맷합니다. 알림 레이블, 주석 및 상태를 적절한 필드에 매핑하고 모든 알림 레이블을 메타데이터로 포함합니다.

> 참고:이 템플릿은 그룹의 첫 번째 알림만 처리합니다. Grafana는 기본적으로 여러 실행 알림을 단일 알림으로 그룹화합니다. 각 알림이 개별적으로 전송되도록 하려면 별로 그

룹화하도록 알림 정책을 구성합니다`alertname`. 또한이 템플릿은 레이블 값 또는 주석에서 특수 JSON 문자를 이스케이프 처리하지 않습니다. 알림 레이블과 `summary` 주석에 큰따옴표 또는 줄 바꿈과 같은 문자가 포함되어 있어서 잘못된 JSON이 생성되지 않는지 확인합니다.

2단계: Webhook 연락 지점 생성

1. Grafana에서 알림 > 연락 지점으로 이동하여 연락 지점 추가를 클릭합니다.
2. Webhook를 통합 유형으로 선택
3. URL을 your AWS DevOps Agent 웹후크 엔드포인트로 설정
4. 선택적 웹후크 설정에서 웹후크 유형에 따라 인증 헤더를 구성합니다. 자세한 내용은 Webhook 인증 방법을 참조하세요.
5. 사용자 지정 템플릿을 사용하도록 메시지 필드를 설정합니다. `{{ template "devops-agent-payload" . }}`
6. 연락 지점 저장을 클릭합니다.

3단계: 알림 정책에 연락 지점 할당

1. 알림 > 알림 정책으로 이동
2. 기존 정책 편집 또는 새 정책 생성
3. 연락 지점을 생성한 웹후크 연락 지점으로 설정합니다.
4. 정책 저장을 클릭합니다.

일치하는 알림이 실행되면 Grafana는 형식이 지정된 페이로드를 AWS DevOps 에이전트로 전송하여 조사를 자동으로 시작합니다.

제한 사항

- ClickHouse 데이터 소스 도구 - ClickHouse 데이터 소스 도구는 현재 지원되지 않습니다.
- 선제적 인시던트 예방 - the section called “선제적 인시던트 예방”는 현재 Grafana 도구를 사용하지 않습니다. 향후 릴리스에 대한 지원이 계획되어 있습니다.

Amazon Managed Grafana 고려 사항

Amazon Managed Grafana(AMG)를 사용하는 경우 다음 제한 사항에 유의하세요.

- Webhook 연락 지점은 지원되지 않음 - AMG는 현재 알림 구성에서 Webhook 연락 지점을 지원하지 않습니다. AMG를 사용하여 알림 웹후크를 AWS DevOps Agent로 직접 보낼 수 없습니다. 자세한 내용은 Amazon Managed Grafana의 연락 지점 알림을 참조하세요.
- 서비스 계정 토큰 만료 - AMG 서비스 계정 토큰의 최대 만료 기간은 30일입니다. 토큰이 만료되기 AWS DevOps 전에 토큰을 교체하고 Grafana 등록을 업데이트해야 합니다. 자격 증명을 업데이트하는 방법은 Grafana 연결 관리를 참조하세요. AMG 토큰 제한에 대한 자세한 내용은 Amazon Managed Grafana의 서비스 계정을 참조하세요.

Grafana 연결 관리

- 자격 증명 업데이트 - 서비스 계정 토큰이 만료되거나 업데이트해야 하는 경우 기능 공급자 페이지에서 Grafana 등록을 취소하고 새 토큰으로 다시 등록합니다.
- 연결된 인스턴스 보기 - AWS DevOps 에이전트 콘솔에서 에이전트 스페이스를 선택하고 기능 탭으로 이동하여 연결된 원격 측정 소스를 봅니다.
- Grafana 제거 - 에이전트 공간에서 Grafana를 연결 해제하려면 원격 측정 섹션에서 Grafana를 선택하고 제거를 클릭합니다. 등록을 완전히 제거하려면 먼저 모든 에이전트 스페이스에서 제거한 다음 기능 공급자 페이지에서 등록을 취소합니다.

새 복제본 연결

기본 제공, 단방향 통합

현재 AWS DevOps Agent는 기본 제공 단방향 통합을 통해 New Relic 사용자를 지원하므로 다음을 사용할 수 있습니다.

- 자동 조사 트리거링 - AWS DevOps 에이전트 웹후크를 통해 AWS DevOps 에이전트 인시던트 해결 조사를 트리거하도록 새 Relic 이벤트를 구성할 수 있습니다.
- 원격 측정 내부 검사 - AWS DevOps Agent는 각 공급자의 원격 MCP 서버를 통해 문제를 조사할 때 New Relic 원격 측정을 내부 검사할 수 있습니다.

온보딩

1단계: 연결

계정 액세스 자격 증명을 사용하여 New Relic 원격 MCP 엔드포인트에 대한 연결 설정

New Relic MCP 도구를 활성화하려면 New relic의 전체 플랫폼 사용자(기본/코어 아님)를 사용하세요.

구성

1. 기능 공급자 페이지로 이동(측면 탐색에서 액세스 가능)
2. 텔레메트리 아래의 사용 가능한 공급자 섹션에서 새 복제본을 찾고 등록을 클릭합니다.
3. 지침에 따라 New Relic API 키를 가져옵니다.
4. New Relic MCP 서버 API 키 세부 정보를 입력합니다.
   - 계정 ID: 위에서 얻은 New Relic 계정 ID를 입력합니다.
   - API 키: 위에서 얻은 API 키를 입력합니다.
   - New Relic 계정의 위치에 따라 미국 또는 EU 리전을 선택합니다.
5. 추가를 클릭합니다.

2단계: 활성화

특정 에이전트 공간에서 New Relic을 활성화하고 적절한 범위 지정을 구성합니다.

구성

1. 에이전트 공간 페이지에서 에이전트 공간을 선택하고 세부 정보 보기를 누릅니다(에이전트 공간을 아직 생성하지 않은 경우 참조the section called “에이전트 스페이스 생성”).
2. 기능 탭 선택
3. 아래로 스크롤하여 원격 측정 섹션으로 이동합니다.
4. 추가를 누릅니다.
5. 새 복제본 선택
6. 다음
7. 검토 후 저장을 누릅니다.
8. Webhook URL 및 API 키 복사

3단계: 웹후크 구성

Webhook URL 및 API 키를 사용하여 경보에서와 같이 조사를 트리거하는 이벤트를 보내도록 New Relic을 구성할 수 있습니다. 웹후크 설정에 대한 자세한 내용은 변경 추적 웹후크를 참조하세요.

DevOps 에이전트가 전송된 이벤트를 사용할 수 있도록 하려면 웹후크로 전송된 데이터가 아래에 지정된 데이터 스키마와 일치하는지 확인합니다. 이 스키마와 일치하지 않는 이벤트는 DevOps Agent에서 무시할 수 있습니다.

메서드 및 헤더 설정

```
    method: "POST",
    headers: {
      "Content-Type": "application/json",
      "Authorization": "Bearer <Token>",
    },
```

본문을 JSON 문자열로 전송합니다.

```
{
    eventType: 'incident';
    incidentId: string;
    action: 'created' | 'updated' | 'closed' | 'resolved';
    priority: "CRITICAL" | "HIGH" | "MEDIUM" | "LOW" | "MINIMAL";
    title: string;
    description?: string;
    timestamp?: string;
    service?: string;
    // The original event generated by service is attached here.
    data?: object;
}
```

New Relic https://newrelic.com/instant-observability/webhook-notifications 웹후크를 전송합니다. 권한 부여 유형에 대해 보유자 토큰을 선택하거나 권한 부여 없음을 선택하고 대신를 사용자 지정 헤더`Authorization: Bearer <Token>`로 추가할 수 있습니다.

자세히 알아보기: https://docs.newrelic.com/docs/agentic-ai/mcp/overview/

제거

원격 측정 소스는 에이전트 스페이스 수준과 계정 수준에서 두 가지 수준으로 연결됩니다. 완전히 제거하려면 먼저 에이전트 공간이 사용되는 모든 에이전트 공간에서 제거한 다음 등록을 취소할 수 있습니다.

1단계: 에이전트 공간에서 제거

1. 에이전트 스페이스 페이지에서 에이전트 스페이스를 선택하고 세부 정보 보기를 누릅니다.
2. 기능 탭 선택
3. 아래로 스크롤하여 원격 측정 섹션으로 이동합니다.

4. 새 복제본 선택
5. 제거를 누릅니다.

2단계: 계정에서 등록 취소

1. 기능 공급자 페이지로 이동(측면 탐색에서 액세스 가능)
2. 현재 등록된 섹션으로 스크롤합니다.
3. 에이전트 공간 수가 0인지 확인합니다(다른 에이전트 공간에서 위의 1단계를 반복하지 않는 경우).
4. New Relic 옆의 등록 취소를 누릅니다.

Splunk 연결

기본 제공, 단방향 통합

현재 AWS DevOps Agent는 기본 제공 단방향 통합을 통해 Splunk 사용자를 지원하므로 다음을 사용할 수 있습니다.

- 자동 조사 트리거 - Splunk 이벤트는 AWS DevOps 에이전트 웹후크를 통해 AWS DevOps 에이전트 인시던트 해결 조사를 트리거하도록 구성할 수 있습니다.
- 원격 측정 내부 검사 - AWS DevOps Agent는 각 공급자의 원격 MCP 서버를 통해 문제를 조사할 때 Splunk 원격 측정을 내부 검사할 수 있습니다.

사전 조건

Splunk API 토큰 가져오기

Splunk를 연결하려면 MCP URL과 토큰이 필요합니다.

Splunk 관리자 단계

Splunk 관리자는 다음 단계를 수행해야 합니다.

- REST API 액세스 활성화
- 배포에서 토큰 인증을 활성화합니다.
- 새 역할 'mcp_user'를 생성하면 새 역할에 기능이 없어도 됩니다.

- MCP 서버를 사용할 권한이 있는 배포의 모든 사용자에게 'mcp_user' 역할을 할당합니다.
- 대상을 'mcp'로 하는 권한 있는 사용자에 대한 토큰을 생성하고, 사용자에게 토큰을 직접 생성할 권한이 없는 경우 적절한 만료를 설정합니다.

Splunk 사용자 단계

Splunk 사용자는 다음 단계를 수행해야 합니다.

- Splunk 관리자로부터 적절한 토큰을 가져오거나 권한이 있는 경우 직접 토큰을 생성합니다. 토큰의 대상은 'mcp'여야 합니다.

온보딩

1단계: 연결

계정 액세스 자격 증명을 사용하여 Splunk 원격 MCP 엔드포인트에 대한 연결 설정

구성

1. 기능 공급자 페이지로 이동(측면 탐색에서 액세스 가능)
2. 사용 가능한 공급자 섹션의 원격 측정에서 Splunk를 찾고 등록을 클릭합니다.
3. Splunk MCP 서버 세부 정보를 입력합니다.
   - 서버 이름 - 고유 식별자(예: my-splunk-server)
   - 엔드포인트 URL - Splunk MCP 서버 엔드포인트:

```
https://<YOUR_SPLUNK_DEPLOYMENT_NAME>.api.scs.splunk.com/
<YOUR_SPLUNK_DEPLOYMENT_NAME>/mcp/v1/
```

- 설명 - 선택적 서버 설명
- 토큰 이름 - 인증을 위한 보유자 토큰의 이름: `my-splunk-token`
- 토큰 값 인증을 위한 보유자 토큰 값

2단계: 활성화

특정 에이전트 공간에서 Splunk를 활성화하고 적절한 범위 지정을 구성합니다.

구성

1. 에이전트 스페이스 페이지에서 에이전트 스페이스를 선택하고 세부 정보 보기를 누릅니다(에이전트 스페이스를 아직 생성하지 않은 경우 참조the section called “에이전트 스페이스 생성”).
2. 기능 탭을 선택합니다.
3. 아래로 스크롤하여 원격 측정 섹션으로 이동합니다.
4. 추가를 누릅니다.
5. Splunk 선택
6. 다음
7. 검토 후 저장을 누릅니다.
8. Webhook URL 및 API 키 복사

3단계: 웹후크 구성

Webhook URL 및 API 키를 사용하여 경보에서와 같이 조사를 트리거하는 이벤트를 보내도록 Splunk를 구성할 수 있습니다.

DevOps 에이전트가 전송된 이벤트를 사용할 수 있도록 하려면 웹후크로 전송된 데이터가 아래에 지정된 데이터 스키마와 일치하는지 확인합니다. 이 스키마와 일치하지 않는 이벤트는 DevOps Agent에서 무시할 수 있습니다.

메서드 및 헤더 설정

```
    method: "POST",
    headers: {
      "Content-Type": "application/json",
      "Authorization": "Bearer <Token>",
    },
```

본문을 JSON 문자열로 전송합니다.

```
{
    eventType: 'incident';
    incidentId: string;
    action: 'created' | 'updated' | 'closed' | 'resolved';
    priority: "CRITICAL" | "HIGH" | "MEDIUM" | "LOW" | "MINIMAL";
```

```
    title: string;
    description?: string;
    timestamp?: string;
    service?: string;
    // The original event generated by service is attached here.
    data?: object;
}
```

Splunk https://help.splunk.com/en/splunk-enterprise/alert-and-respond/alerting-manual/9.4/configure-alert-actions/use-a-webhook-alert-action 사용하여 웹후크 전송(권한 부여 없음 선택 대신 사용자 지정 헤더 옵션 사용)

자세히 알아보기:

- Splunk의 MCP 서버 설명서: https://help.splunk.com/en/splunk-cloud-platform/mcp-server-for-splunk-platform/about-mcp-server-for-splunk-platform
- Splunk Cloud Platform REST API의 액세스 요구 사항 및 제한 사항: https://docs.splunk.com/Documentation/SplunkCloud/latest/RESTTUT/RESTandCloud
- Splunk Cloud Platform에서 인증 토큰 관리: https://help.splunk.com/en/splunk-cloud-platform/administer/manage-users-and-security/9.3.2411/authenticate-into-the-splunk-platform-with-tokens/manage-or-delete-authentication-tokens
- Splunk Web을 사용하여 역할 생성 및 관리: https://docs.splunk.com/Documentation/SplunkCloud/latest/Security/Addandeditroles

제거

원격 측정 소스는 에이전트 공간 수준과 계정 수준에서 두 가지 수준으로 연결됩니다. 완전히 제거하려면 먼저 에이전트 공간이 사용되는 모든 에이전트 공간에서 제거한 다음 등록을 취소할 수 있습니다.

1단계: 에이전트 공간에서 제거

1. 에이전트 공간 페이지에서 에이전트 공간을 선택하고 세부 정보 보기를 누릅니다.
2. 기능 탭을 선택합니다.
3. 아래로 스크롤하여 원격 측정 섹션으로 이동합니다.
4. Splunk 선택
5. 제거를 누릅니다.

2단계: 계정에서 등록 취소

1. 기능 공급자 페이지로 이동(측면 탐색에서 액세스 가능)
2. 현재 등록된 섹션으로 스크롤합니다.
3. 에이전트 공간 수가 0인지 확인합니다(다른 에이전트 공간에서 위의 1단계를 반복하지 않는 경우).
4. Splunk 옆의 등록 취소를 누릅니다.

티켓팅 및 채팅에 연결

AWS DevOps 에이전트는 팀의 기존 커뮤니케이션 채널에 참여하여 팀의 구성원 역할을 하도록 설계되었습니다. DevOps 에이전트를 ServiceNow 및 PagerDuty와 같은 티켓팅 및 경보 시스템에 연결하여 인시던트 티켓에서 조사를 자동으로 시작하여 기존 워크플로 내에서 인시던트 대응을 가속화하여 의도한 복구 시간(MTTR)을 줄일 수 있습니다. 또한 DevOps 에이전트를 Slack과 같은 팀 협업 시스템에 연결하여 채팅 채널에서 DevOps 에이전트로부터 활동 요약을 받을 수 있습니다.

티켓팅 및 채팅 통합 연결에 대한 자세한 내용은 다음을 참조하세요.

- the section called “PagerDuty 연결”
- the section called “ServiceNow 연결”
- the section called “Slack 연결”

PagerDuty 연결

PagerDuty 통합을 통해 AWS DevOps Agent는 인시던트 조사 및 자동 대응 중에 PagerDuty 계정에서 인시던트 데이터, 대기 일정 및 서비스 정보에 액세스하고 업데이트할 수 있습니다. 이 통합은 보안 인증을 위해 OAuth 2.0을 사용합니다.

> **Important**
>
> AWS DevOps Agent는 최신 PagerDuty OAuth 2.0(범위 지정 OAuth)만 지원합니다. 리디렉션 uri가 있는 레거시 PagerDuty OAuth는 지원되지 않습니다.

PagerDuty 요구 사항

PagerDuty를 연결하기 전에 다음을 확인해야 합니다.

- OAuth 클라이언트 ID 및 클라이언트 보안 암호가 있는 PagerDuty 계정
- PagerDuty 계정 하위 도메인(예: PagerDuty URL이 `https://your-company.pagerduty.com`인 경우 하위 도메인은 `your-company`)

PagerDuty 등록

PagerDuty는 AWS 계정 수준에서 등록되고 해당 계정의 모든 에이전트 스페이스 간에 공유됩니다.

1단계: PagerDuty에서 액세스 구성

1. AWS Management Console에 로그인
2. AWS DevOps 에이전트 콘솔로 이동
3. 기능 공급자 페이지로 이동(측면 탐색에서 액세스 가능)
4. 통신 아래의 사용 가능한 공급자 섹션에서 PagerDuty를 찾고 등록을 클릭합니다.
5. PagerDuty의 액세스 구성 페이지에서 안내 설정을 따릅니다.

서비스 리전 및 하위 도메인을 확인합니다.

- 계정 범위 - PagerDuty 리전(미국 또는 EU)을 선택하고 PagerDuty 하위 도메인을 입력합니다. 예를 들어 PagerDuty URL이 인 경우 `https://your-company.pagerduty.com`를 입력합니다`your-company`.

PagerDuty에서 새 앱을 생성합니다.

- 별도의 브라우저 탭에서 PagerDuty에 로그인하고 통합 > 앱 등록으로 이동합니다.
- OAuth 2.0 범위 OAuth를 사용하여 새 앱 생성
- 권한에서 , `incidents.read` `incidents.write`및 최소 필수 범위를 부여합니다. `services.read`
- Events Integration을 활성화하여 AWS DevOps 에이전트와 PagerDuty 간의 양방향 통신 허용

OAuth 자격 증명 구성:

- 권한 범위 - 최소 필수 범위는 `incidents.read`, `incidents.write`, 입니다. `services.read`
- 클라이언트 이름 - OAuth 클라이언트에 대한 설명이 포함된 이름을 입력합니다.
- 클라이언트 ID - PagerDuty 앱 등록의 OAuth 클라이언트 ID를 입력합니다.

- 클라이언트 보안 암호 - PagerDuty 앱 등록의 OAuth 클라이언트 보안 암호를 입력합니다.

2단계: PagerDuty 등록 검토 및 제출

1. 모든 PagerDuty 구성 세부 정보 검토
2. 제출을 클릭하여 등록을 완료합니다.
3. 등록에 성공하면 PagerDuty가 기능 공급자 페이지의 현재 등록된 섹션에 나타납니다.

에이전트 스페이스에 PagerDuty 추가

계정 수준에서 PagerDuty를 등록한 후 개별 에이전트 스페이스에 연결할 수 있습니다.

1. AWS DevOps 에이전트 콘솔에서 에이전트 스페이스를 선택합니다.
2. 기능 탭으로 이동
3. 커뮤니케이션 섹션에서 추가를 클릭합니다.
4. 사용 가능한 공급자 목록에서 PagerDuty를 선택합니다.
5. 저장을 클릭합니다.

PagerDuty 연결 관리

- 자격 증명 업데이트 - OAuth 자격 증명을 업데이트해야 하는 경우 기능 공급자 페이지에서 PagerDuty 등록을 취소하고 새 자격 증명으로 다시 등록합니다.
- 연결 보기 - AWS DevOps 에이전트 콘솔에서 에이전트 스페이스를 선택하고 기능 탭으로 이동하여 연결된 통신 통합을 확인합니다.
- PagerDuty 제거 - 에이전트 공간에서 PagerDuty를 연결 해제하려면 통신 섹션에서 선택한 다음 제거를 클릭합니다. 등록을 완전히 제거하려면 먼저 모든 에이전트 스페이스에서 제거한 다음 기능 공급자 페이지에서 등록을 취소합니다.

Webhook 지원

AWS DevOps Agent는 PagerDuty V3 웹후크만 지원합니다. 이전 웹후크 버전은 지원되지 않습니다.

PagerDuty V3 웹후크 구독에 대한 자세한 내용은 PagerDuty 개발자 설명서의 Webhooks 개요를 참조하세요.

ServiceNow 연결

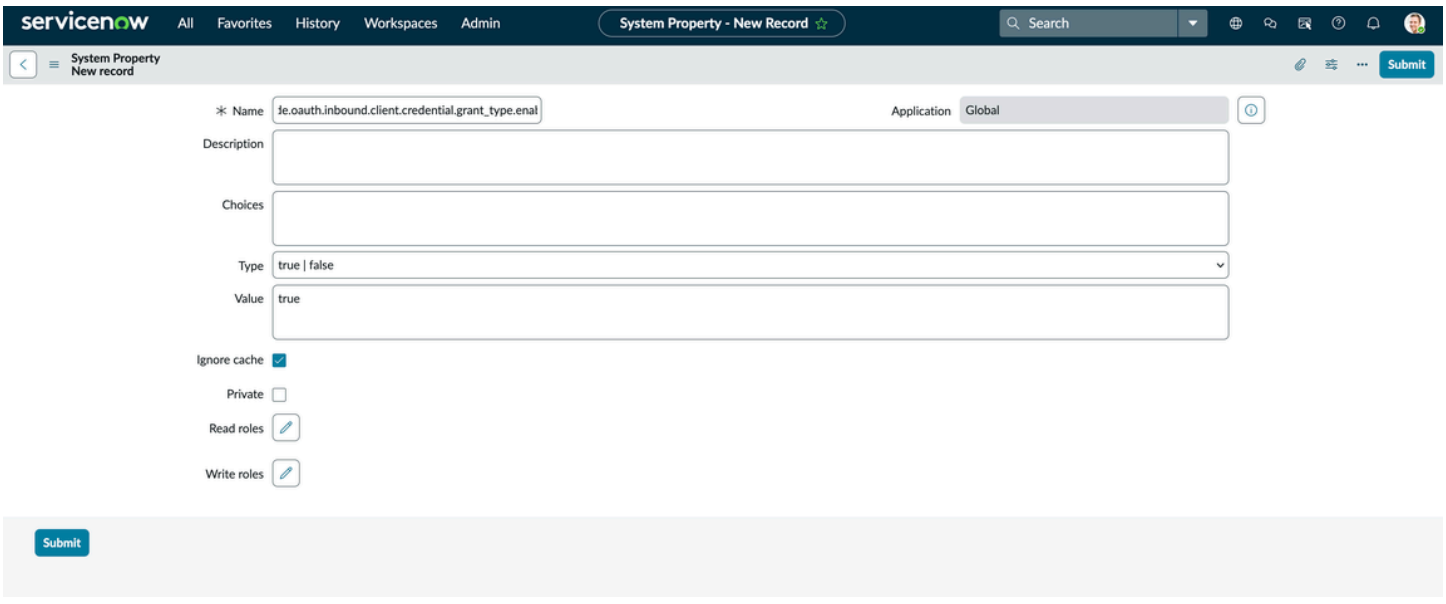
이 자습서에서는 ServiceNow 인스턴스를 AWS DevOps Agent에 연결하여 티켓이 생성될 때 인시던트 대응 조사를 자동으로 시작하고 주요 조사 결과를 발신 티켓에 게시할 수 있도록 안내합니다. 또한 특정 티켓만 DevOps 에이전트 스페이스로 전송하도록 ServiceNow 인스턴스를 구성하는 방법과 여러 DevOps 에이전트 스페이스에서 티켓 라우팅을 오케스트레이션하는 방법에 대한 예제도 포함되어 있습니다.

초기 설정

첫 번째 단계는 ServiceNow에서 ServiceNow 인스턴스에 액세스하는 데 AWS DevOps가 사용할 수 있는 OAuth 애플리케이션 클라이언트를 생성하는 것입니다.

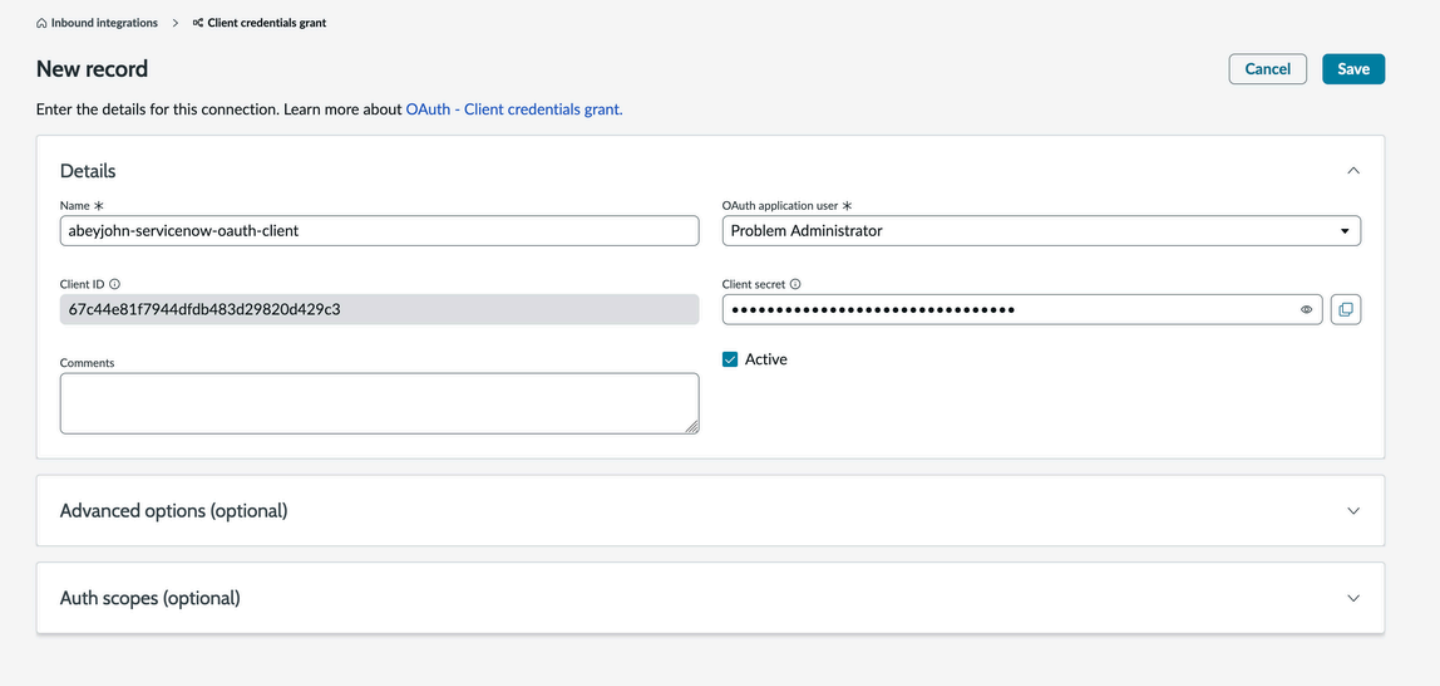
ServiceNow OAuth 애플리케이션 클라이언트 생성

1. 인스턴스의 클라이언트 자격 증명 시스템 속성 활성화
   a. 필터 검색 상자에서 검색한 다음 Entersys_properties.list를 누릅니다(옵션은 표시되지 않지만 Enter를 누르면 작동함).
   b. 새로 만들기를 선택합니다.
   c. 이름을 로, 값을 true에 추가`glide.oauth.inbound.client.credential.grant_type.enabled`하고 유형은 true | false



1. 필터 검색 상자에서 시스템 OAuth > 애플리케이션 레지스트리로 이동합니다.

2. “New” > “New Inbound Integration Experience” > “New Integration” > “OAuth - Client Credentials Grant”를 선택합니다.
3. 이름을 선택하고 OAuth 애플리케이션 사용자를 “문제 관리자”로 설정하고 “저장”을 클릭합니다.

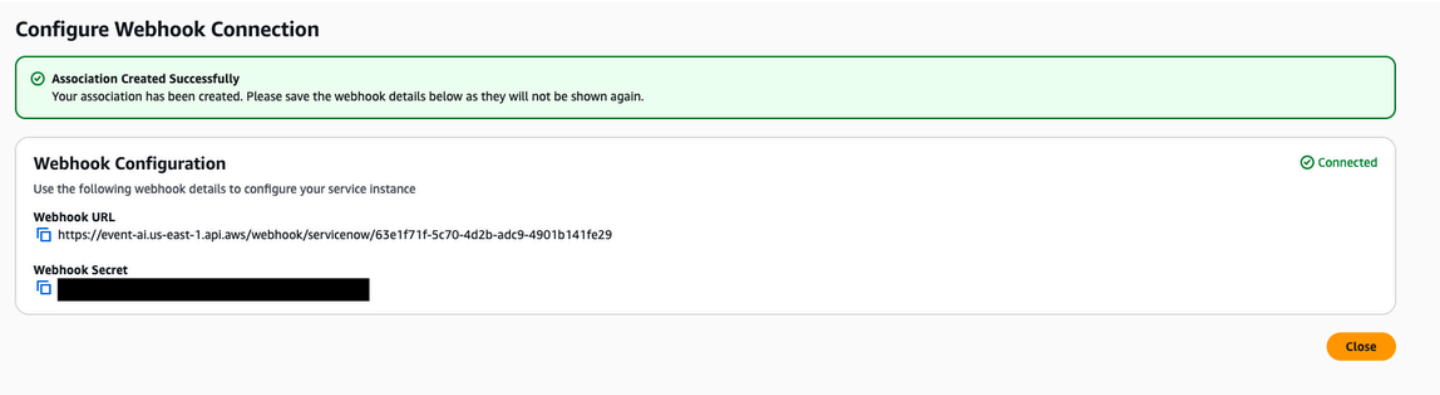


ServiceNow OAuth 클라이언트를 AWS DevOps 에이전트에 연결

1. 이 프로세스는 두 곳에서 시작할 수 있습니다. 먼저 기능 공급자 페이지로 이동하여 통신에서 ServiceNow를 찾은 다음 등록을 클릭합니다. 또는 생성한 DevOps 에이전트 스페이스를 선택하고 기능 → 통신 → 추가 → ServiceNow로 이동하여 등록을 클릭할 수 있습니다.
2. 그런 다음 방금 생성한 OAuth 애플리케이션 클라이언트를 사용하여 DevOps Agent가 ServiceNow 인스턴스에 액세스할 수 있도록 권한을 부여합니다.

- 다음 단계에 따라 Webhook에 대한 결과 정보를 저장합니다.

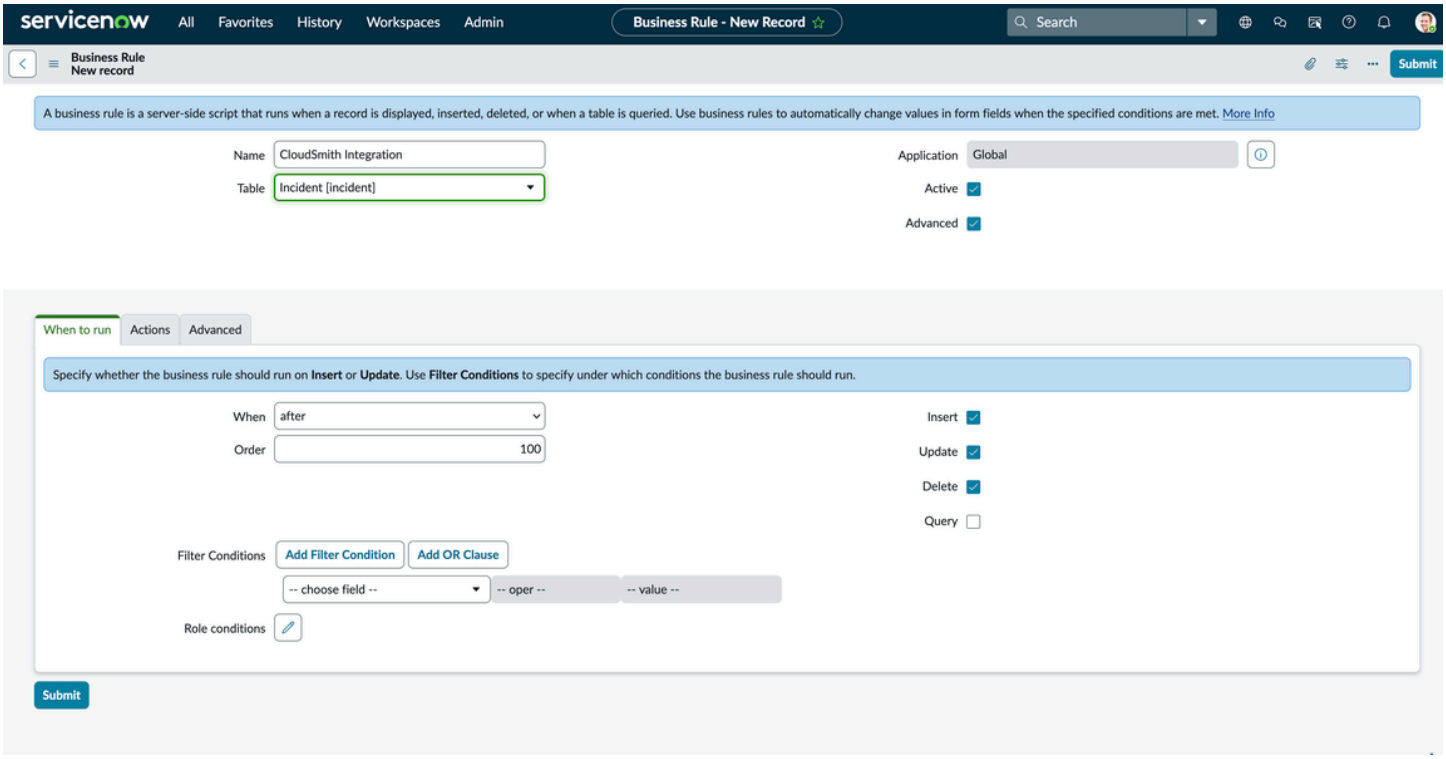
> **Important**
>
> 이 정보는 다시 표시되지 않습니다.



ServiceNow 비즈니스 규칙 구성

연결을 설정한 후에는 ServiceNow에서 비즈니스 규칙을 구성하여 DevOps 에이전트 스페이스(들)에 티켓을 보내야 합니다.

1. 활동 구독 → 관리 → 비즈니스 규칙으로 이동하여 새로 만들기를 클릭합니다.
2. “Table” 필드를 “Incident [incident]”로 설정하고 “Advanced” 상자를 선택한 다음 삽입, 업데이트 및 삭제 후 실행되도록 규칙을 설정합니다.



1. “고급” 탭으로 이동하여 다음 웹후크 스크립트를 추가하고 표시된 위치에 웹후크 보안 암호와 URL을 삽입한 다음 제출을 클릭합니다.

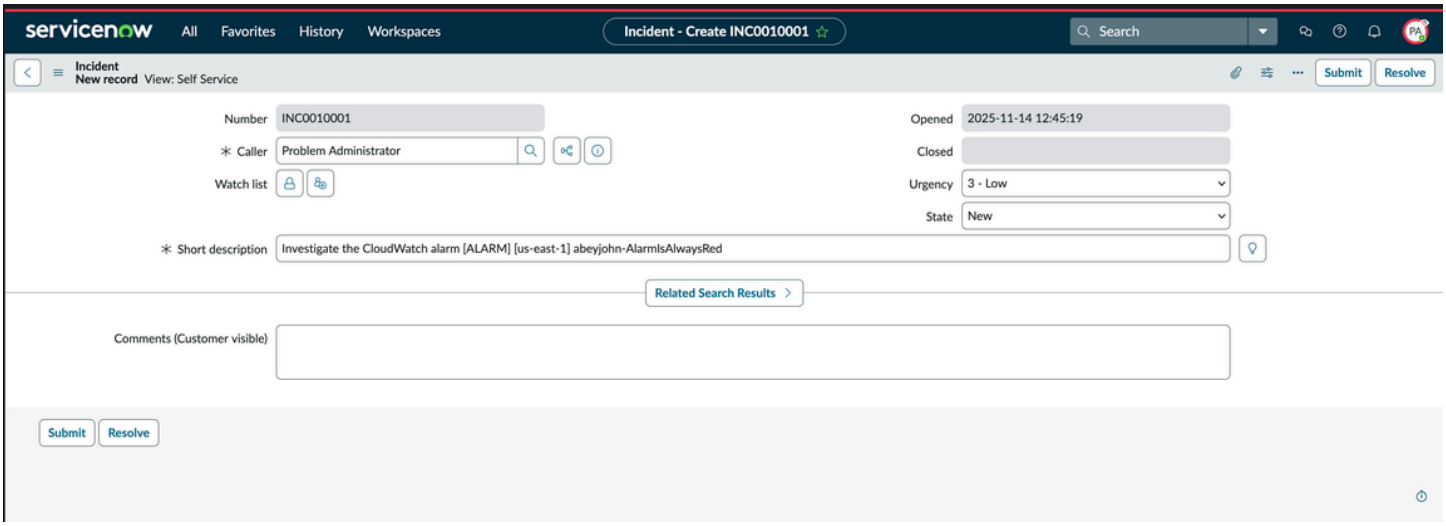
```
(function executeRule(current, previous /*null when async*/ ) {

    var WEBHOOK_CONFIG = {
        webhookSecret: GlideStringUtil.base64Encode('<<< INSERT WEBHOOK SECRET HERE
 >>>'),
        webhookUrl: '<<< INSERT WEBHOOK URL HERE >>>'
    };

    function generateHMACSignature(payloadString, secret) {
        try {
            var mac = new GlideCertificateEncryption();
            var signature = mac.generateMac(secret, "HmacSHA256", payloadString);
            return signature;
        } catch (e) {
            gs.error('HMAC generation failed: ' + e);
            return null;
        }
    }
```

```
    function callWebhook(payload, config) {
        try {
            var timestamp = new Date().toISOString();
            var payloadString = JSON.stringify(payload);
            var payloadWithTimestamp =`${timestamp}:${payloadString}`;

            var signature = generateHMACSignature(payloadWithTimestamp,
config.webhookSecret);

            if (!signature) {
                gs.error('Failed to generate signature');
                return false;
            }

            gs.info('Generated signature: ' + signature);

            var request = new sn_ws.RESTMessageV2();
            request.setEndpoint(config.webhookUrl);
            request.setHttpMethod('POST');

            request.setRequestHeader('Content-Type', 'application/json');
            request.setRequestHeader('x-amzn-event-signature', signature);
            request.setRequestHeader('x-amzn-event-timestamp', timestamp);

            request.setRequestBody(payloadString);

            var response = request.execute();
            var httpStatus = response.getStatusCode();
            var responseBody = response.getBody();

            if (httpStatus >= 200 && httpStatus < 300) {
                gs.info('Webhook sent successfully. Status: ' + httpStatus);
                return true;
            } else {
                gs.error('Webhook failed. Status: ' + httpStatus + ', Response: ' +
responseBody);
                return false;
            }

        } catch (ex) {
            gs.error('Error sending webhook: ' + ex.getMessage());
            return false;
        }
    }
```

```
    function createReference(field) {
        if (!field || field.nil()) {
            return null;
        }

        return {
            link: field.getLink(true),
            value: field.toString()
        };
    }

    function getStringValue(field) {
        if (!field || field.nil()) {
            return null;
        }
        return field.toString();
    }

    function getIntValue(field) {
        if (!field || field.nil()) {
            return null;
        }
        var val = parseInt(field.toString());
        return isNaN(val) ? null : val;
    }

    var eventType = (current.operation() == 'insert') ? "create" : "update";

    var incidentEvent = {
        eventType: eventType.toString(),
        sysId: current.sys_id.toString(),
        priority: getStringValue(current.priority),
        impact: getStringValue(current.impact),
        active: getStringValue(current.active),
        urgency: getStringValue(current.urgency),
        description: getStringValue(current.description),
        shortDescription: getStringValue(current.short_description),
        parent: getStringValue(current.parent),
        incidentState: getStringValue(current.incident_state),
        severity: getStringValue(current.severity),
        problem: createReference(current.problem),
        additionalContext: {}
    };
```

```
    incidentEvent.additionalContext = {
        number: current.number.toString(),
        opened_at: getStringValue(current.opened_at),
        opened_by: current.opened_by.nil() ? null :
 current.opened_by.getDisplayValue(),
        assigned_to: current.assigned_to.nil() ? null :
 current.assigned_to.getDisplayValue(),
        category: getStringValue(current.category),
        subcategory: getStringValue(current.subcategory),
        knowledge: getStringValue(current.knowledge),
        made_sla: getStringValue(current.made_sla),
        major_incident: getStringValue(current.major_incident)
    };

    for (var key in incidentEvent.additionalContext) {
        if (incidentEvent.additionalContext[key] === null) {
            delete incidentEvent.additionalContext[key];
        }
    }

    gs.info(JSON.stringify(incidentEvent, null, 2)); // Pretty print for logging only

    if (WEBHOOK_CONFIG.webhookUrl && WEBHOOK_CONFIG.webhookSecret) {
        callWebhook(incidentEvent, WEBHOOK_CONFIG);
    } else {
        gs.info('Webhook not configured.');
    }

})(current, previous);
```

기능 공급자 페이지에서 ServiceNow 연결을 등록하기로 선택한 경우 이제 ServiceNow 인시던트 티켓을 조사하려는 DevOps 에이전트 공간으로 이동하여 기능 → 통신을 선택한 다음 기능 공급자 페이지에 등록한 ServiceNow 인스턴스를 등록해야 합니다. 이제 모든 것을 설정해야 하며 호출자가 “문제 관리자”(AWS DevOps OAuth 클라이언트에 부여한 권한을 모방하기 위해)로 설정된 모든 인시던트는 구성된 DevOps 에이전트 스페이스에서 인시던트 대응 조사를 트리거합니다. ServiceNow에서 새 인시던트를 생성하고 인시던트의 발신자 필드를 “문제 관리자”로 설정하여 이를 테스트할 수 있습니다.



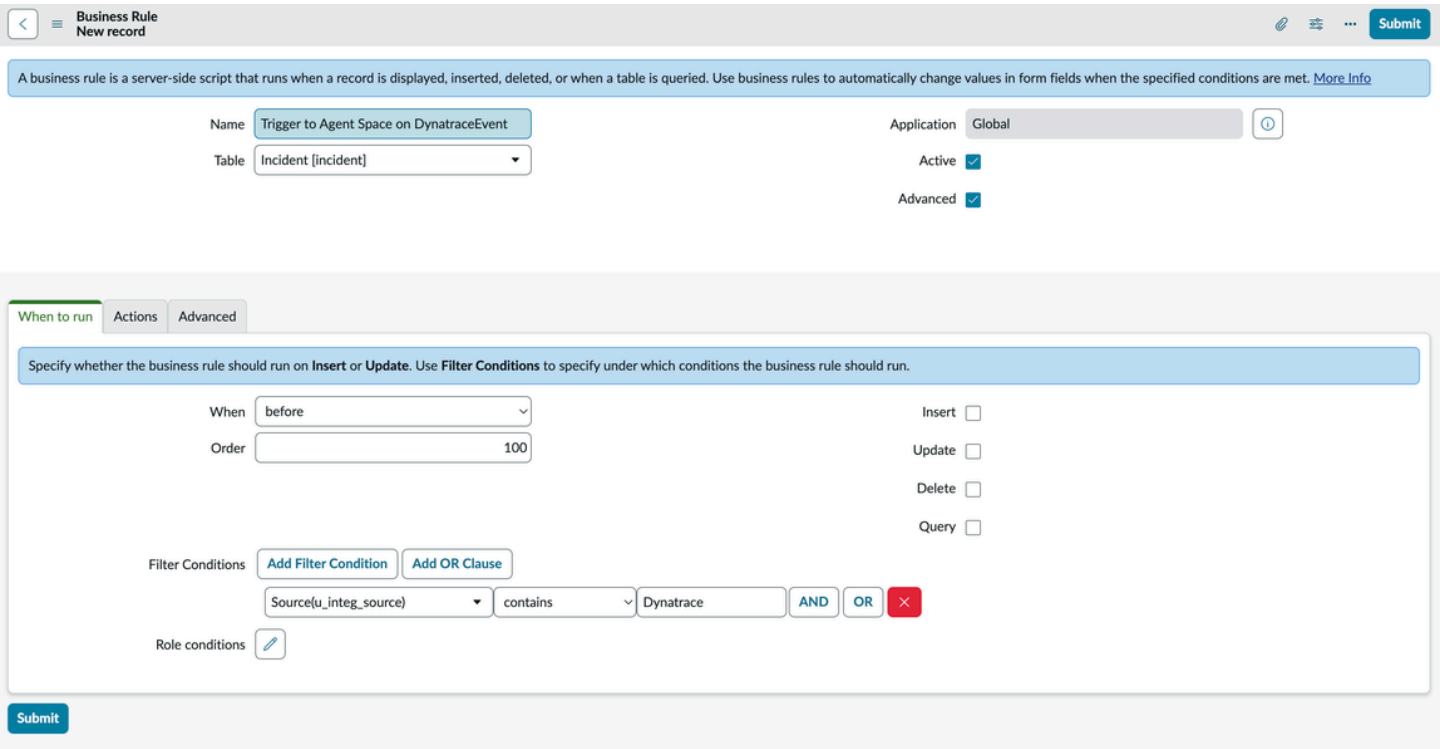
ServiceNow 티켓 업데이트

트리거된 모든 인시던트 대응 조사 중에 DevOps 에이전트는 주요 조사 결과, 근본 원인 분석 및 완화 계획에 대한 업데이트를 원래 티켓에 제공합니다. 에이전트 조사 결과는 인시던트의 설명에 게시되며, 현재는 유형 , `finding`, `causeinvestigation_summarymitigation_summary`, 및 조사 상태 업데이트(예: `AWS DevOps Agent started/finished its investigation`)의 에이전트 레코드만 게시합니다.

티켓 라우팅 및 오케스트레이션 예제

시나리오: DevOps 에이전트 스페이스로 전송되는 인시던트 필터링

이는 간단한 시나리오이지만 인시던트 소스를 추적하기 위해 ServiceNow에서 필드를 생성하려면 ServiceNow에서 일부 구성이 필요합니다. 이 예제에서는 SNOW 양식 빌더를 사용하여 새 소스(u_source) 필드를 생성합니다. 이렇게 하면 인시던트 소스를 추적하고 이를 사용하여 특정 소스에서 DevOps 에이전트 스페이스로 요청을 라우팅할 수 있습니다. 라우팅은 Service Now Business 규칙을 생성하고 실행 시기 탭에서 “시기” 트리거 및 “조건 필터링”을 설정하여 수행됩니다. 이 예제에서 필터 조건은 다음과 같이 설정됩니다.

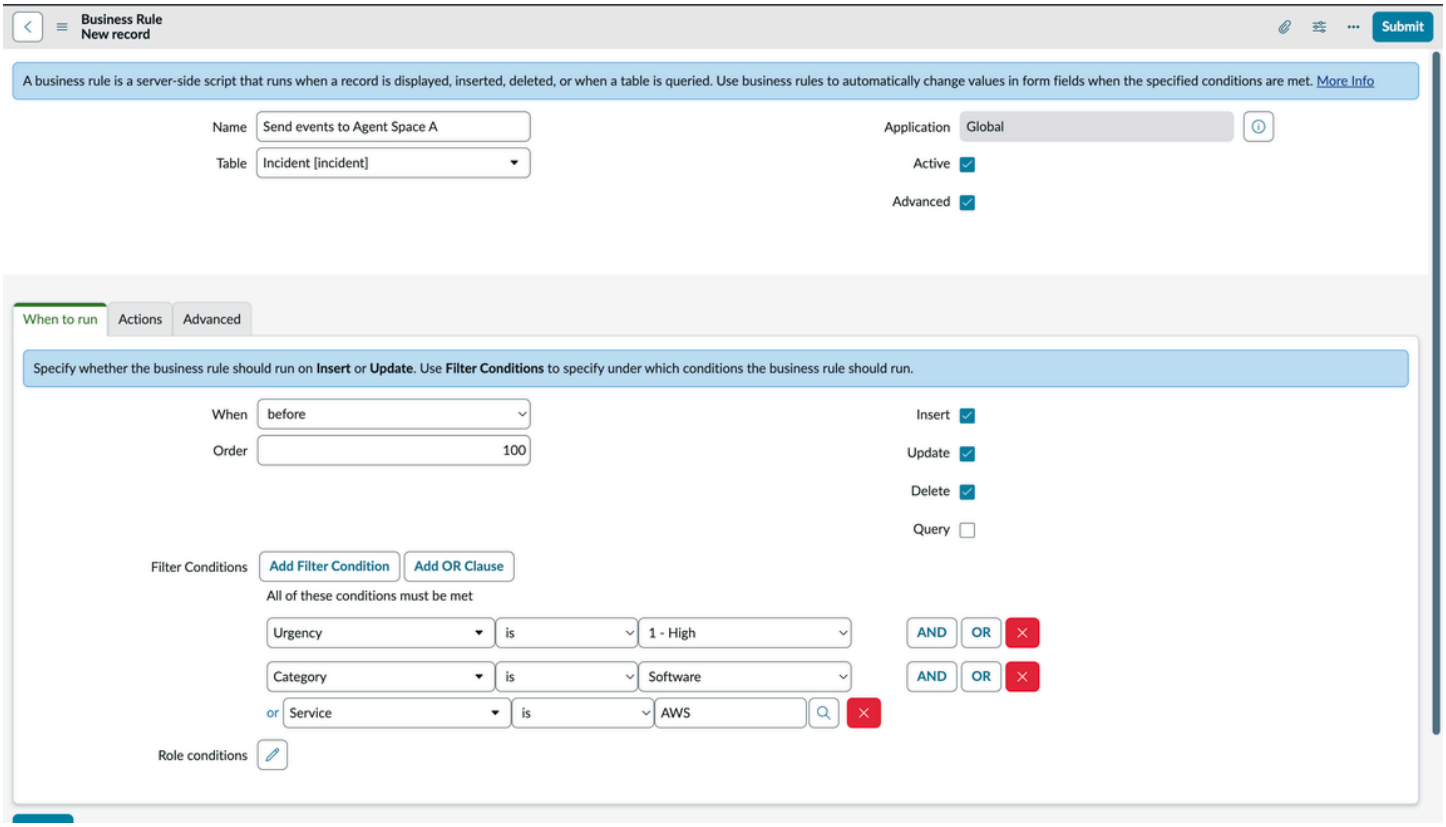


시나리오: 여러 DevOps 에이전트 스페이스에서 인시던트 라우팅

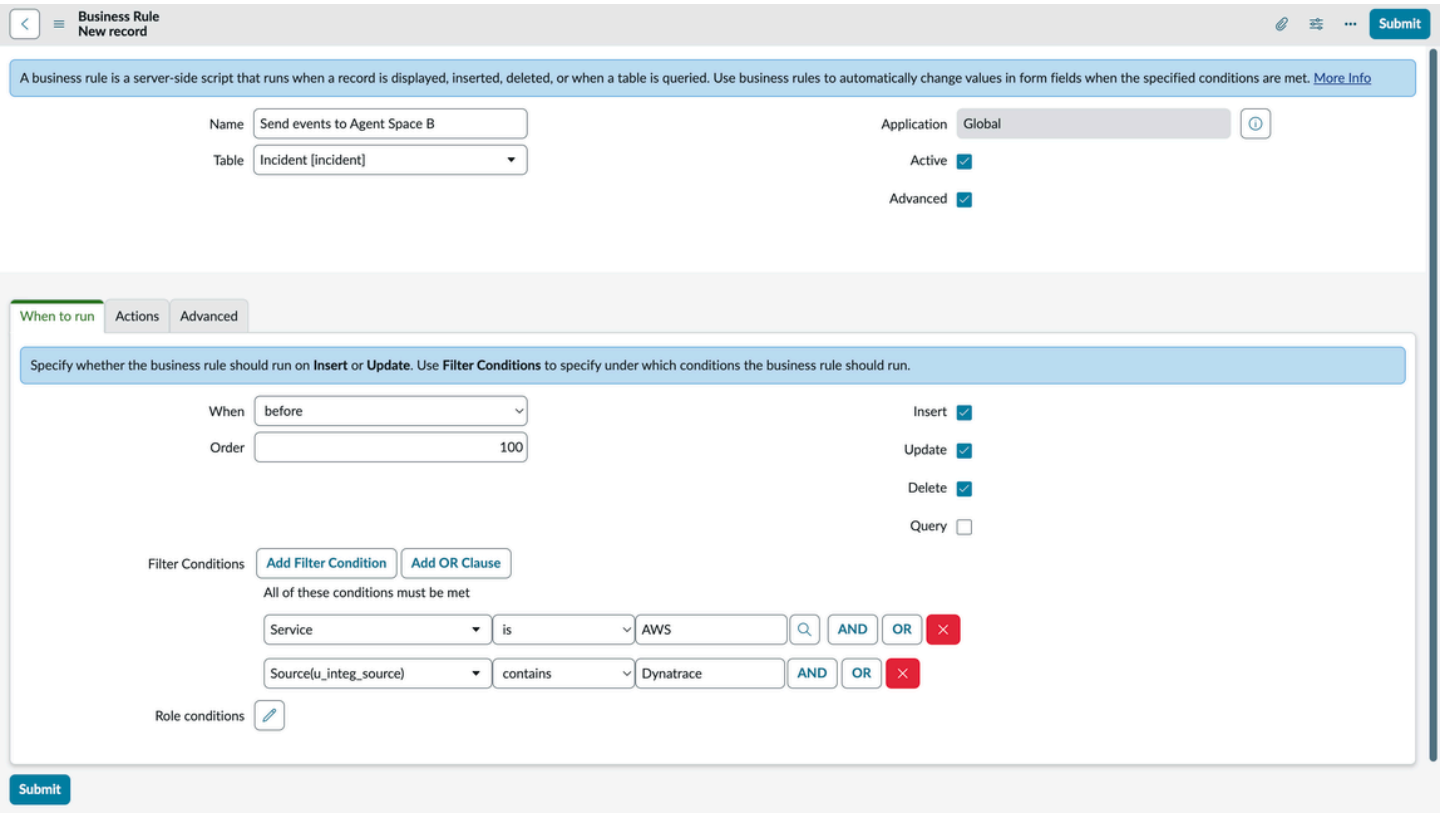
이 예제는 긴급성이 , 범주가 1, Software 서비스가 인 경우 DevOps 에이전트 스페이스 B에서 조사를 트리거하고 서비스가 AWS, AWS소스가 인 경우 DevOps 에이전트 스페이스 A에서 조사를 트리거하는 방법을 보여줍니다Dynatrace.

이 시나리오는 두 가지 방법으로 수행할 수 있습니다. 이 비즈니스 로직을 포함하도록 웹후크 스크립트 자체를 업데이트할 수 있습니다. 이 시나리오에서는 투명성을 높이고 디버깅을 간소화하기 위해 ServiceNow 비즈니스 규칙을 사용하여 이를 수행하는 방법을 보여줍니다. 라우팅은 Service Now Business 규칙 2개를 생성하여 수행됩니다.

- ServiceNow에서 DevOps 에이전트 공간 A에 대한 비즈니스 규칙을 생성하고 조건 빌더를 사용하여 지정된 조건에 따라 이벤트만 전송하는 조건을 생성합니다.



- 그런 다음 ServiceNow for AgentSpace B에서 서비스가 AWS 이고 소스가 Dynatrace인 경우에만 비즈니스 규칙이 트리거되는 다른 비즈니스 규칙을 생성합니다.



이제 지정된 조건과 일치하는 새 인시던트를 생성하면 DevOps 에이전트 스페이스 A 또는 DevOps 에이전트 스페이스 B에 대한 조사를 트리거하여 인시던트 라우팅을 세밀하게 제어할 수 있습니다.

Slack 연결

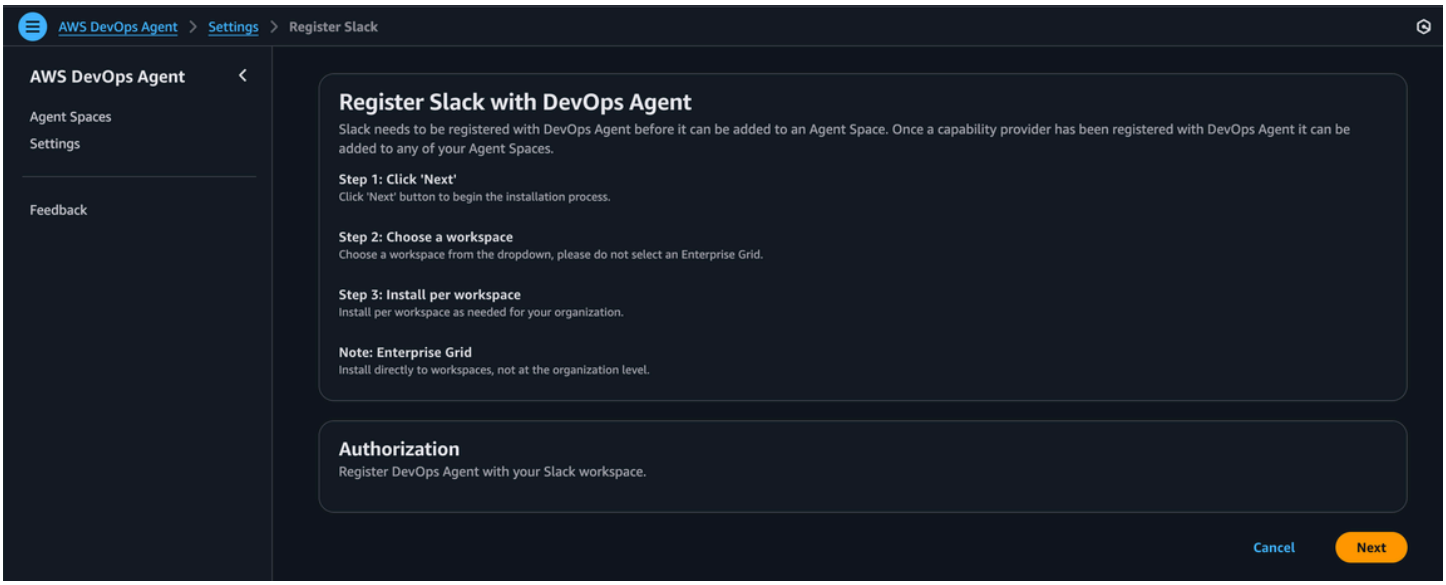
선택한 Slack 채널을 인시던트 대응 조사 주요 조사 결과, 근본 원인 분석 및 생성된 완화 계획으로 업데이트하도록 AWS DevOps 에이전트를 구성할 수 있습니다.

시작하기 전 준비 사항

Slack을 에이전트 스페이스에 추가하려면 먼저 DevOps 에이전트에 등록해야 합니다. Slack과 AWS DevOps 에이전트를 통합하려면 다음 요구 사항을 충족해야 합니다.

- 타사 애플리케이션을 설치하고 승인할 수 있는 기능을 갖춘 Slack 워크스페이스에 액세스할 수 있습니다.
- AWS DevOps 에이전트가 알림을 보낼 Slack 채널을 식별했습니다.

Slack과 AWS DevOps 에이전트 통합 등록



1. AWS DevOps 에이전트 콘솔의 기능 공급자 페이지에서 통신 아래의 사용 가능한 공급자 섹션에서 Slack을 찾아 등록을 클릭합니다.
2. 등록 버튼을 선택합니다.
3. 워크스페이스에 대한 AWS DevOps 에이전트 애플리케이션을 승인하기 위해 Slack으로 리디렉션 됩니다.
4. Slack 권한 부여 페이지에서 조직 수준이 아닌 워크스페이스에 직접를 설치합니다.
5. 드롭다운에서 워크스페이스를 선택합니다. 엔터프라이즈 그리드를 선택하지 마십시오.
6. 조직에 필요한 워크스페이스별로를 설치합니다.
7. 요청된 범위를 검토하고 허용을 클릭하여 통합을 승인합니다.
8. 권한 부여 후 AWS DevOps 에이전트 콘솔로 돌아갑니다.

Slack을 DevOps 에이전트 스페이스(들)와 연결

DevOps 에이전트 스페이스에 Slack을 등록한 후 DevOps 에이전트 스페이스(들)와 연결할 수 있습니다.

1. 구성된 AgentSpace 내의 기능 탭에서 통신 > Slack으로 이동합니다.
2. Slack 추가를 선택합니다.
3. 채널 ID를 입력합니다.
4. 생성을 선택하여 Slack 구성을 완료합니다.

> **Note**
>
> 에이전트의 봇 사용자는 메시지를 게시하기 전에 프라이빗 채널에 추가해야 합니다.

> **Important**
>
> Slack 앱을 제거하면 Slack 앱을 다시 설치할 수 없게 될 수 있습니다. Slack 앱을 제거하지 마십시오.

Webhook를 통해 DevOps 에이전트 호출

Webhook를 사용하면 외부 시스템이 자동으로 AWS DevOps 에이전트 조사를 트리거할 수 있습니다. 이를 통해 티켓팅 시스템, 모니터링 도구 및 인시던트 발생 시 HTTP 요청을 보낼 수 있는 기타 플랫폼과 통합할 수 있습니다.

사전 조건

웹후크 액세스를 구성하기 전에 다음을 갖추어야 합니다.

- in AWS DevOps 에이전트에 구성된 에이전트 공간
- AWS DevOps 에이전트 콘솔에 대한 액세스
- Webhook 요청을 전송할 외부 시스템

Webhook 유형

AWS DevOps Agent는 다음과 같은 유형의 웹후크를 지원합니다.

- 통합별 웹후크 - Dynatrace, Splunk, Datadog, New Relic, ServiceNow 또는 Slack과 같은 타사 통합을 구성할 때 자동으로 생성됩니다. 이러한 웹후크는 특정 통합과 연결되며 통합 유형에 따라 결정되는 인증 방법을 사용합니다.
- 일반 웹후크 - 특정 통합에서 다루지 않는 소스에서 조사를 트리거하기 위해 수동으로 생성할 수 있습니다. 일반 웹후크는 현재 HMAC 인증을 사용합니다(베어러 토큰은 현재 사용할 수 없음).
- Grafana 알림 웹후크 - Grafana는 웹후크 연락 지점을 통해 AWS DevOps 에이전트에 직접 알림 알림을 보낼 수 있습니다. 사용자 지정 알림 템플릿을 포함한 설정 지침은 Grafana 연결을 참조하세요.

Webhook 인증 방법

웹후크의 인증 방법은 연결된 통합에 따라 다릅니다.

HMAC 인증 - 다음에서 사용:

- Dynatrace 통합 웹후크
- 일반 웹후크(특정 타사 통합에 연결되지 않음)

베어러 토큰 인증 - 다음에서 사용:

- Splunk 통합 웹후크
- Datadog 통합 웹후크
- New Relic 통합 웹후크
- ServiceNow 통합 웹후크
- Slack 통합 웹후크

웹후크 액세스 구성

1단계: Webhook 구성으로 이동

1. AWS Management Console에 로그인하고 AWS DevOps 에이전트 콘솔로 이동합니다.
2. 에이전트 스페이스 선택
3. 기능 탭으로 이동
4. Webhook 섹션에서 구성을 클릭합니다.

2단계: Webhook 자격 증명 생성

통합별 웹후크의 경우:

타사 통합 구성을 완료하면 Webhook가 자동으로 생성됩니다. Webhook 엔드포인트 URL 및 자격 증명은 통합 설정 프로세스가 끝날 때 제공됩니다.

일반 웹후크의 경우:

1. 웹후크 생성을 클릭합니다.

2. 시스템에서 HMAC 키 페어를 생성합니다.
3. 생성된 키와 보안 암호를 안전하게 저장합니다. 다시 검색할 수 없습니다.
4. 제공된 웹후크 엔드포인트 URL 복사

3단계: 외부 시스템 구성

Webhook 엔드포인트 URL 및 자격 증명을 사용하여 외부 시스템이 AWS DevOps Agent에 요청을 보내도록 구성합니다. 특정 구성 단계는 외부 시스템에 따라 다릅니다.

Webhook 자격 증명 관리

자격 증명 제거 - 웹후크 자격 증명을 삭제하려면 웹후크 구성 섹션으로 이동하여 제거를 클릭합니다. 자격 증명을 제거한 후에는 새 자격 증명을 생성할 때까지 웹후크 엔드포인트가 더 이상 요청을 수락하지 않습니다.

자격 증명 재생성 - 새 자격 증명을 생성하려면 먼저 기존 자격 증명을 제거한 다음 새 키 페어 또는 토큰을 생성합니다.

Webhook 사용

Webhook 요청 형식

조사를 트리거하려면 외부 시스템이 HTTP POST 요청을 웹후크 엔드포인트 URL로 보내야 합니다.

버전 1(HMAC 인증)의 경우:

헤더:

- `Content-Type: application/json`
- `x-amzn-event-signature: <HMAC signature>`
- `x-amzn-event-timestamp: <+%Y-%m-%dT%H:%M:%S.000Z>`

HMAC 서명은 SHA-256을 사용하여 보안 키로 요청 본문에 서명하여 생성됩니다.

버전 2(베어러 토큰 인증)의 경우:

헤더:

- `Content-Type: application/json`

- `Authorization: Bearer <your-token>`

요청 본문:

요청 본문에는 인시던트에 대한 정보가 포함되어야 합니다.

```
json

{
  "title": "Incident title",
  "severity": "high",
  "affectedResources": ["resource-id-1", "resource-id-2"],
  "timestamp": "2025-11-23T18:00:00Z",
  "description": "Detailed incident description",
  "data": {
    "metadata": {
        "region": "us-east-1",
        "environment": "production"
    }
  }
}
```

예제 코드

버전 1(HMAC 인증) - JavaScript:

```
const crypto = require('crypto');

// Webhook configuration
const webhookUrl = 'https://your-webhook-endpoint.amazonaws.com/invoke';
const webhookSecret = 'your-webhook-secret-key';

// Incident data
const incidentData = {
    eventType: 'incident',
    incidentId: 'incident-123',
    action: 'created',
    priority: "HIGH",
    title: 'High CPU usage on production server',
    description: 'High CPU usage on production server host ABC in AWS account 1234
 region us-east-1',
    timestamp: new Date().toISOString(),
```

```
    service: 'MyTestService',
    data: {
      metadata: {
        region: 'us-east-1',
        environment: 'production'
      }
    }
};

// Convert data to JSON string
const payload = JSON.stringify(incidentData);
const timestamp = new Date().toISOString();
const hmac = crypto.createHmac("sha256", webhookSecret);
hmac.update(`${timestamp}:${payload}`, "utf8");
const signature = hmac.digest("base64");

// Send the request
fetch(webhookUrl, {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
    'x-amzn-event-timestamp': timestamp,
    'x-amzn-event-signature': signature
  },
  body: payload
})
.then(res => {
  console.log(`Status Code: ${res.status}`);
  return res.text();
})
.then(data => {
  console.log('Response:', data);
})
.catch(error => {
  console.error('Error:', error);
});
```

버전 1(HMAC 인증) - cURL:

```
#!/bin/bash

# Configuration
WEBHOOK_URL="https://event-ai.us-east-1.api.aws/webhook/generic/YOUR_WEBHOOK_ID"
```

```
SECRET="YOUR_WEBHOOK_SECRET"

# Create payload
TIMESTAMP=$(date -u +%Y-%m-%dT%H:%M:%S.000Z)
INCIDENT_ID="test-alert-$(date +%s)"

PAYLOAD=$(cat <<EOF
{
"eventType": "incident",
"incidentId": "$INCIDENT_ID",
"action": "created",
"priority": "HIGH",
"title": "Test Alert",
"description": "Test alert description",
"service": "TestService",
"timestamp": "$TIMESTAMP"
}
EOF
)

# Generate HMAC signature
SIGNATURE=$(echo -n "${TIMESTAMP}:${PAYLOAD}" | openssl dgst -sha256 -hmac "$SECRET" -
binary | base64)

# Send webhook
curl -X POST "$WEBHOOK_URL" \
-H "Content-Type: application/json" \
-H "x-amzn-event-timestamp: $TIMESTAMP" \
-H "x-amzn-event-signature: $SIGNATURE" \
-d "$PAYLOAD"
```

버전 2(베어러 토큰 인증) - JavaScript:

```
function sendEventToWebhook(webhookUrl, secret) {
  const timestamp = new Date().toISOString();

  const payload = {
    eventType: 'incident',
    incidentId: 'incident-123',
    action: 'created',
    priority: "HIGH",
    title: 'Test Alert',
    description: 'Test description',
```

```
    timestamp: timestamp,
    service: 'TestService',
    data: {}
  };

  fetch(webhookUrl, {
    method: "POST",
    headers: {
      "Content-Type": "application/json",
      "x-amzn-event-timestamp": timestamp,
      "Authorization": `Bearer ${secret}`,  // Fixed: template literal
    },
    body: JSON.stringify(payload),
  });
}
```

버전 2(베어러 토큰 인증) - cURL:

```
#!/bin/bash

# Configuration
WEBHOOK_URL="https://event-ai.us-east-1.api.aws/webhook/generic/YOUR_WEBHOOK_ID"
SECRET="YOUR_WEBHOOK_SECRET"

# Create payload
TIMESTAMP=$(date -u +%Y-%m-%dT%H:%M:%S.000Z)
INCIDENT_ID="test-alert-$(date +%s)"

PAYLOAD=$(cat <<EOF
{
"eventType": "incident",
"incidentId": "$INCIDENT_ID",
"action": "created",
"priority": "HIGH",
"title": "Test Alert",
"description": "Test alert description",
"service": "TestService",
"timestamp": "$TIMESTAMP"
}
EOF
)

# Send webhook
```

```
curl -X POST "$WEBHOOK_URL" \
-H "Content-Type: application/json" \
-H "x-amzn-event-timestamp: $TIMESTAMP" \
-H "Authorization: Bearer $SECRET" \
-d "$PAYLOAD"
```

웹후크 문제 해결

200을 받지 못한 경우

200과 웹후크와 같은 메시지가 수신되면 인증이 통과되었고 시스템에서 확인 및 처리할 수 있도록 메시지가 대기열에 있음을 나타냅니다. 200은 얻지 못하지만 4xx는 인증 또는 헤더에 문제가 있을 가능성이 높습니다. 인증을 디버깅하는 데 도움이 되도록 curl 옵션을 사용하여 수동으로 전송해 보세요.

200을 수신했지만 조사가 시작되지 않는 경우

잘못된 페이로드가 원인일 수 있습니다.

1. 타임스탬프와 인시던트 ID가 모두 업데이트되고 고유한지 확인합니다. 중복 메시지는 중복 제거됩니다.
2. 메시지가 유효한 JSON인지 확인합니다.
3. 형식이 올바른지 확인합니다.

200을 수신하고 조사가 즉시 취소되는 경우

해당 월의 한도에 도달했을 가능성이 높습니다. 해당하는 경우 AWS 연락처에 문의하여 속도 제한 변경을 요청하십시오.

관련 주제

- the section called “에이전트 스페이스 생성”
- the section called “DevOps 에이전트 웹 앱이란 무엇입니까?”
- the section called “DevOps 에이전트 IAM 권한”

Amazon EventBridge와 AWS DevOps 에이전트 통합

조사 및 완화 수명 주기 동안 발생하는 이벤트를 사용하여 이벤트 기반 애플리케이션과 AWS DevOps 에이전트를 통합할 수 있습니다. AWS DevOps 에이전트는 조사 또는 완화 상태가 변경될 때 Amazon

EventBridge로 이벤트를 전송합니다. 그런 다음 이러한 이벤트를 기반으로 조치를 취하는 EventBridge 규칙을 생성할 수 있습니다.

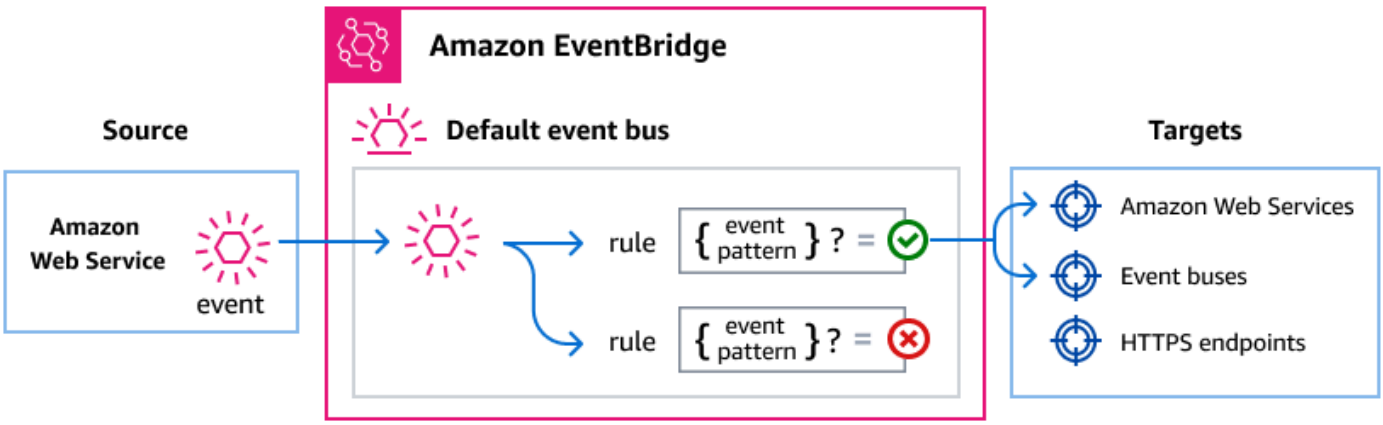
예를 들어 다음 작업을 수행하는 규칙을 생성할 수 있습니다.

- 조사가 완료되면 AWS Lambda 함수를 호출하여 조사 결과를 처리합니다.
- 조사가 실패하거나 시간이 초과되면 Amazon SNS 알림을 보냅니다.
- 새 조사가 생성되면 티켓팅 시스템을 업데이트합니다.
- 완화 작업이 완료되면 AWS Step Functions 워크플로를 시작합니다.

EventBridge가 AWS DevOps 에이전트 이벤트를 라우팅하는 방법

AWS DevOps Agent는 EventBridge 기본 이벤트 버스로 이벤트를 전송합니다. 그런 다음 EventBridge는 생성한 규칙을 기준으로 이벤트를 평가합니다. 이벤트가 규칙의 이벤트 패턴과 일치하면 EventBridge는 이벤트를 지정된 대상으로 보냅니다.

다음 다이어그램은 EventBridge가 AWS DevOps 에이전트 이벤트를 라우팅하는 방법을 보여줍니다.



1. AWS DevOps Agent는 조사 또는 완화 수명 주기 상태가 변경될 때 EventBridge 기본 이벤트 버스로 이벤트를 전송합니다.
2. EventBridge는 생성한 규칙을 기준으로 이벤트를 평가합니다.
3. 이벤트가 규칙의 이벤트 패턴과 일치하면 EventBridge는 규칙에 지정된 대상으로 이벤트를 보냅니다.

AWS DevOps 에이전트 이벤트

AWS DevOps Agent는 다음 이벤트를 EventBridge로 전송합니다. 모든 이벤트는 소스를 사용합니다`aws.aidevops`.

지원되는 조사 이벤트

detail-type	설명
`Investigation Created`	에이전트 공간에 조사가 생성되었습니다.
`Investigation Priority Updated`	조사의 우선 순위가 변경되었습니다.
`Investigation In Progress`	조사가 활성 분석을 시작했습니다.
`Investigation Completed`	조사 결과가 성공적으로 완료되었습니다.
`Investigation Failed`	조사에서 오류가 발생하여 완료할 수 없습니다.
`Investigation Timed Out`	조사가 최대 허용 기간을 초과했습니다.
`Investigation Cancelled`	완료 전에 조사가 취소되었습니다.
`Investigation Pending Triage`	활성 분석이 시작되기 전에 조사가 분류를 기다리고 있습니다.
`Investigation Linked`	조사가 관련 인시던트 또는 티켓과 연결되었습니다.

지원되는 완화 이벤트

detail-type	설명
`Mitigation In Progress`	완화 작업이 시작되었습니다.
`Mitigation Completed`	완화 작업이 성공적으로 완료되었습니다.
`Mitigation Failed`	완화 작업에서 오류가 발생하여 완료할 수 없습니다.

detail-type	설명
`Mitigation Timed Out`	완화 작업이 최대 허용 기간을 초과했습니다.
`Mitigation Cancelled`	완료 전에 완화 작업이 취소되었습니다.

자세한 필드 설명 및 예제 이벤트는 섹션을 참조하세요the section called “AWS DevOps 에이전트 이벤트 세부 정보 참조”.

AWS DevOps 에이전트 이벤트와 일치하는 이벤트 패턴 생성

EventBridge 규칙은 이벤트 패턴을 사용하여 이벤트를 선택하고 대상으로 라우팅합니다. 이벤트 패턴은 처리하는 이벤트의 구조와 일치합니다. 이벤트 필드를 기반으로 이벤트 패턴을 생성하여 filter AWS DevOps 에이전트 이벤트를 필터링합니다.

다음 예제에서는 일반적인 사용 사례에 대한 이벤트 패턴을 보여줍니다.

all AWS DevOps 에이전트 이벤트 일치

다음 이벤트 패턴은 AWS DevOps Agent의 모든 이벤트와 일치합니다.

```
{
  "source": ["aws.aidevops"]
}
```

조사 이벤트만 일치

다음 이벤트 패턴은 접두사 일치를 사용하여 조사 수명 주기 이벤트만 선택합니다.

```
{
  "source": ["aws.aidevops"],
  "detail-type": [{"prefix": "Investigation"}]
}
```

완료 및 실패 이벤트만 일치

다음 이벤트 패턴은 완료되거나 실패한 조사 및 완화에 대한 이벤트와 일치합니다.

```
{
  "source": ["aws.aidevops"],
  "detail-type": [
```

```
    "Investigation Completed",
    "Investigation Failed",
    "Mitigation Completed",
    "Mitigation Failed"
  ]
}
```

특정 에이전트 스페이스에 대한 이벤트 일치

다음 이벤트 패턴은 특정 에이전트 공간의 이벤트와 일치합니다.

```
{
  "source": ["aws.aidevops"],
  "detail": {
    "metadata": {
      "agent_space_id": ["your-agent-space-id"]
    }
  }
}
```

이벤트 패턴에 대한 자세한 내용은 Amazon EventBridge 사용 설명서에서 Amazon EventBridge 이벤트 패턴을 참조하세요.

Amazon EventBridge 권한

AWS DevOps Agent는 EventBridge에 이벤트를 전송하는 데 추가 권한이 필요하지 않습니다. 이벤트는 기본 이벤트 버스로 자동으로 전송됩니다.

EventBridge 규칙에 대해 구성하는 대상에 따라 특정 권한을 추가해야 할 수 있습니다. 대상에 필요한 권한에 대한 자세한 내용은 Amazon EventBridge 사용 설명서의 Amazon EventBridge에 대한 리소스 기반 정책 사용을 참조하세요. EventBridge

추가 EventBridge 리소스

EventBridge 개념 및 구성에 대한 자세한 내용은 Amazon EventBridge 사용 설명서의 다음 주제를 참조하세요.

- EventBridge 이벤트 버스
- EventBridge 이벤트
- EventBridge 이벤트 패턴
- EventBridge 규칙

- EventBridge 대상

AWS DevOps 에이전트 이벤트 세부 정보 참조

AWS 서비스의 이벤트에는 , , source, detail-typeaccountregion, 등 공통 메타데이터 필드가 있습니다time. 이러한 이벤트에는 서비스와 관련된 데이터가 있는 detail 필드도 포함됩니다. For AWS DevOps 에이전트 이벤트의 경우 source는 항상 aws.aidevops 이고는 특정 이벤트를 detail-type 식별합니다.

조사 이벤트

다음 detail-type 값은 조사 이벤트를 식별합니다.

- Investigation Created
- Investigation Priority Updated
- Investigation In Progress
- Investigation Completed
- Investigation Failed
- Investigation Timed Out
- Investigation Cancelled
- Investigation Pending Triage
- Investigation Linked

source 및 detail-type 필드는 AWS DevOps 에이전트 이벤트에 대한 특정 값을 포함하므로 아래에 포함되어 있습니다. 모든 이벤트에 포함된 다른 메타데이터 필드의 정의는 Amazon EventBridge 이벤트 참조의 이벤트 구조를 참조하세요.

다음은 조사 이벤트에 대한 JSON 구조입니다.

```
{
  . . .,
  "detail-type" : "string",
  "source" : "aws.aidevops",
  . . .,
  "detail" : {
    "version" : "string",
    "metadata" : {
```

```
      "agent_space_id" : "string",
      "task_id" : "string",
      "execution_id" : "string"
    },
    "data" : {
      "task_type" : "string",
      "priority" : "string",
      "status" : "string",
      "created_at" : "string",
      "updated_at" : "string",
      "summary_record_id" : "string"
    }
  }
}
```

detail-type 이벤트 유형을 식별합니다. 조사 이벤트의 경우 이는 이전에 나열된 이벤트 이름 중 하나입니다.

source 이벤트를 생성한 서비스를 식별합니다. For AWS DevOps 에이전트 이벤트의 경우이 값은 입니다`aws.aidevops`.

detail 이벤트별 데이터가 포함된 JSON 객체입니다. `detail` 객체에는 다음 필드가 포함됩니다.

- `version` (문자열) - 이벤트 세부 정보의 스키마 버전입니다. 현재 입니다`1.0.0`.
- `metadata.agent_space_id` (문자열) - 이벤트가 시작된 에이전트 공간의 고유 식별자입니다.
- `metadata.task_id` (문자열) - 작업의 고유 식별자입니다.
- `metadata.execution_id` (문자열) - 실행 실행의 고유 식별자입니다. 실행이 조사에 할당되었을 때 표시됩니다.
- `data.task_type` (문자열) - 작업 유형입니다. 값: `INVESTIGATION`.
- `data.priority` (문자열) - 우선 순위 수준입니다. 값: `CRITICAL`, `HIGH`, `MEDIUM`, `LOW`, `MINIMAL`.
- `data.status` (문자열) - 현재 상태입니다. 값: `PENDING_START`, `IN_PROGRESS`, `COMPLETED`, `FAILED`, `TIMED_OUT`, `CANCELLED`, `PENDING_TRIAGE`, `LINKED`.
- `data.created_at` (문자열) - 작업이 생성되었을 때 ISO 8601 타임스탬프입니다.
- `data.updated_at` (문자열) - 작업이 마지막으로 업데이트된 시점의 ISO 8601 타임스탬프입니다.
- `data.summary_record_id` (문자열) - 조사 결과가 포함된 요약 레코드의 식별자입니다. 완료된 조사에 대한 요약이 생성될 때 포함됩니다. 이 식별자를 사용하여 레코드 유형이 인 저널 레코드를 조회하여 AWS DevOps 에이전트 API를 통해 요약 콘텐츠를 검색할 수 있습니다`investigation_summary_md`.

예: 조사 완료 이벤트

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789015",
  "detail-type": "Investigation Completed",
  "source": "aws.aidevops",
  "account": "123456789012",
  "time": "2026-03-12T18:10:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:aidevops:us-
east-1:123456789012:agentspace/8f6187a7-0388-4926-8217-3a0fe32f757c"
  ],
  "detail": {
    "version": "1.0.0",
    "metadata": {
      "agent_space_id": "8f6187a7-0388-4926-8217-3a0fe32f757c",
      "task_id": "a1b2c3d4-5678-90ab-cdef-example11111",
      "execution_id": "b2c3d4e5-6789-01ab-cdef-example22222"
    },
    "data": {
      "task_type": "INVESTIGATION",
      "priority": "CRITICAL",
      "status": "COMPLETED",
      "created_at": "2026-03-12T18:00:00Z",
      "updated_at": "2026-03-12T18:10:00Z",
      "summary_record_id": "d4e5f6g7-6789-01ab-cdef-example44444"
    }
  }
}
```

예: 조사 실패 이벤트

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789016",
  "detail-type": "Investigation Failed",
  "source": "aws.aidevops",
  "account": "123456789012",
  "time": "2026-03-12T18:10:00Z",
  "region": "us-east-1",
  "resources": [
```

```
    "arn:aws:aidevops:us-
east-1:123456789012:agentspace/8f6187a7-0388-4926-8217-3a0fe32f757c"
  ],
  "detail": {
    "version": "1.0.0",
    "metadata": {
      "agent_space_id": "8f6187a7-0388-4926-8217-3a0fe32f757c",
      "task_id": "a1b2c3d4-5678-90ab-cdef-example11111",
      "execution_id": "b2c3d4e5-6789-01ab-cdef-example22222"
    },
    "data": {
      "task_type": "INVESTIGATION",
      "priority": "CRITICAL",
      "status": "FAILED",
      "created_at": "2026-03-12T18:00:00Z",
      "updated_at": "2026-03-12T18:10:00Z"
    }
  }
}
```

완화 이벤트

다음 `detail-type` 값은 완화 이벤트를 식별합니다.

- `Mitigation In Progress`
- `Mitigation Completed`
- `Mitigation Failed`
- `Mitigation Timed Out`
- `Mitigation Cancelled`

`source` 및 `detail-type` 필드는 AWS DevOps 에이전트 이벤트에 대한 특정 값을 포함하므로 아래에 포함되어 있습니다. 모든 이벤트에 포함된 다른 메타데이터 필드의 정의는 Amazon EventBridge 이벤트 참조의 이벤트 구조를 참조하세요.

다음은 완화 이벤트에 대한 JSON 구조입니다.

```
{
  . . .,
  "detail-type" : "string",
  "source" : "aws.aidevops",
```

```
  . . .,
  "detail" : {
    "version" : "string",
    "metadata" : {
      "agent_space_id" : "string",
      "task_id" : "string",
      "execution_id" : "string"
    },
    "data" : {
      "task_type" : "string",
      "priority" : "string",
      "status" : "string",
      "created_at" : "string",
      "updated_at" : "string",
      "summary_record_id" : "string"
    }
  }
}
```

detail-type 이벤트 유형을 식별합니다. 완화 이벤트의 경우 이는 이전에 나열된 이벤트 이름 중 하나입니다.

source 이벤트를 생성한 서비스를 식별합니다. For AWS DevOps 에이전트 이벤트의 경우이 값은 입니다`aws.aidevops`.

detail 이벤트별 데이터가 포함된 JSON 객체입니다. `detail` 객체에는 다음 필드가 포함됩니다.

- `version` (문자열) - 이벤트 세부 정보의 스키마 버전입니다. 현재 입니다`1.0.0`.
- `metadata.agent_space_id` (문자열) - 이벤트가 시작된 에이전트 공간의 고유 식별자입니다.
- `metadata.task_id` (문자열) - 작업의 고유 식별자입니다.
- `metadata.execution_id` (문자열) - 실행 실행의 고유 식별자입니다. 실행이 완화에 할당되었을 때 표시됩니다.
- `data.task_type` (문자열) - 작업 유형입니다. 값: `INVESTIGATION`.
- `data.priority` (문자열) - 우선 순위 수준입니다. 값: `CRITICAL`, `HIGH`, `MEDIUM`, `LOW`, `MINIMAL`.
- `data.status` (문자열) - 현재 상태입니다. 값: `IN_PROGRESS`, `COMPLETED`, `FAILED`, `TIMED_OUT`, `CANCELLED`.
- `data.created_at` (문자열) - 작업이 생성되었을 때 ISO 8601 타임스탬프입니다.
- `data.updated_at` (문자열) - 작업이 마지막으로 업데이트된 시점의 ISO 8601 타임스탬프입니다.

- `data.summary_record_id` (문자열) - 완화 조사 결과를 포함하는 요약 레코드의 식별자입니다. 완료된 완화를 위한 요약이 생성될 때 포함됩니다. 이 식별자를 사용하여 레코드 유형이 인저널 레코드를 조회하여 AWS DevOps 에이전트 API를 통해 요약 콘텐츠를 검색할 수 있습니다`mitigation_summary_md`.

예: 완화 완료 이벤트

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-12345678901c",
  "detail-type": "Mitigation Completed",
  "source": "aws.aidevops",
  "account": "123456789012",
  "time": "2026-03-12T18:20:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:aidevops:us-
east-1:123456789012:agentspace/8f6187a7-0388-4926-8217-3a0fe32f757c"
  ],
  "detail": {
    "version": "1.0.0",
    "metadata": {
      "agent_space_id": "8f6187a7-0388-4926-8217-3a0fe32f757c",
      "task_id": "a1b2c3d4-5678-90ab-cdef-example11111",
      "execution_id": "c3d4e5f6-7890-12ab-cdef-example33333"
    },
    "data": {
      "task_type": "INVESTIGATION",
      "priority": "CRITICAL",
      "status": "COMPLETED",
      "created_at": "2026-03-12T18:00:00Z",
      "updated_at": "2026-03-12T18:20:00Z",
      "summary_record_id": "e5f6g7h8-7890-12ab-cdef-example55555"
    }
  }
}
```

예: 완화 실패 이벤트

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-12345678901d",
```

```
  "detail-type": "Mitigation Failed",
  "source": "aws.aidevops",
  "account": "123456789012",
  "time": "2026-03-12T18:20:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:aidevops:us-
east-1:123456789012:agentspace/8f6187a7-0388-4926-8217-3a0fe32f757c"
  ],
  "detail": {
    "version": "1.0.0",
    "metadata": {
      "agent_space_id": "8f6187a7-0388-4926-8217-3a0fe32f757c",
      "task_id": "a1b2c3d4-5678-90ab-cdef-example11111",
      "execution_id": "c3d4e5f6-7890-12ab-cdef-example33333"
    },
    "data": {
      "task_type": "INVESTIGATION",
      "priority": "CRITICAL",
      "status": "FAILED",
      "created_at": "2026-03-12T18:00:00Z",
      "updated_at": "2026-03-12T18:20:00Z"
    }
  }
}
```

판매 로그 및 지표

판매된 Amazon CloudWatch 지표 및 로그를 사용하여 에이전트 공간 및 서비스 작업을 모니터링할 수 있습니다. 이 주제에서는 AWS DevOps 에이전트가 계정에 자동으로 게시하는 CloudWatch 지표와 원하는 대상으로 전송하도록 구성할 수 있는 판매 로그에 대해 설명합니다.

판매된 CloudWatch 지표

AWS DevOps Agent는 계정의 Amazon CloudWatch에 지표를 자동으로 게시합니다. 이러한 지표는 구성 없이 사용할 수 있습니다. 이를 사용하여 사용량을 모니터링하고, 운영 활동을 추적하고, 경보를 생성할 수 있습니다.

서비스 연결 역할

Amazon CloudWatch 지표가이 서비스에 대한 계정에 게시되도록 하기 위해 AWS DevOps 에이전트는 서비스 연결 역할 AWSServiceRoleForAIDevOps 서비스 연결 역할을 자동으로 생성합니다. API를

호출하는 IAM 역할에 적절한 권한이 없는 경우 InvalidParameterException과 함께 리소스 생성이 실패합니다.

> **Important**
>
> 2026년 3월 13일 이전에 AgentSpace를 생성한 고객은 AWSServiceRoleForAIDevOps 서비스 연결 역할을 수동으로 생성하여 계정에 AWS DevOps 에이전트에 대한 CloudWatch 지표를 게시해야 합니다.

서비스 연결 역할 수동 생성(기존 고객의 경우)

다음 중 하나를 수행하세요.

- IAM 콘솔에서 AWS DevOps 에이전트 서비스 아래에 AWSServiceRoleForAIDevOps 역할을 생성합니다.
- AWS CLI에서 다음 명령을 실행합니다.

```
aws iam create-service-linked-role --aws-service-name aidevops.amazonaws.com
```

네임스페이스

모든 지표는 `AWS/AIDevOps` 네임스페이스 아래에 게시됩니다.

측정 기준

모든 지표에는 다음 차원이 포함됩니다.

차원	설명
`AgentSpaceUUID`	에이전트 공간의 고유 식별자입니다. 계정의 모든 에이전트 공간에서 지표를 집계하려면 CloudWatch 수학 표현식을 사용하거나 차원 필터를 생략합니다.

지표 참조

지표 이름	설명	단위	게시 빈도	유용한 통계
ConsumedChatRequests	에이전트 공간이 사용한 채팅 요청 수입니다. 계정의 총 개수를 가져오려면 모든 SUM 차원에서 통계를 사용합니다AgentSpaceUUID .	개수	5분마다	합계, 평균
ConsumedInvestigationTime	에이전트 공간에서 조사를 실행하는 데 소요된 시간입니다.	초	5분마다	합계, 평균, 최대
ConsumedEvaluationTime	에이전트 공간에서 평가를 실행하는 데 소요된 시간입니다.	초	5분마다	합계, 평균, 최대
TopologyCompletionCount	토폴로지 처리 완료 횟수입니다. AWS DevOps Agent는 온보딩 중 초기 생성부터 수동 업데이트 또는 예약된 일일 새로 고침까지 토폴로지 처리가 완료되면이 지표를 내보냅니다.	개수	이벤트 기반(완료할 때마다 발생)	합계, SampleCount

CloudWatch 콘솔에서 지표 보기

1. CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 지표를 선택한 다음 모든 지표를 선택합니다.
3. AWS/AIDevOps 네임스페이스를 선택합니다.
4. 에이전트 공간에 대한 지표를 보려면 AgentSpace별을 선택합니다.

> **Note**
>
> 이러한 지표에 CloudWatch 경보를 생성하여 사용량이 임계값을 초과할 때 알림을 받을 수 있습니다. 예를 들어에서 경보를 생성`ConsumedChatRequests`하여 채팅 요청 소비를 모니터링합니다.

사전 조건

로그 전송을 구성하기 전에 다음이 있는지 확인합니다.

- AWS DevOps 에이전트 콘솔에 액세스할 수 있는 활성 AWS 계정
- CloudWatch Logs 전송 APIs에 대한 권한이 있는 IAM 보안 주체
- (선택 사항) 로그 대상으로 사용할 계획인 경우 Amazon S3 버킷 또는 Amazon Data Firehose 전송 스트림

벤딩 로그

AWS DevOps Agent는 에이전트 스페이스 및 서비스 등록이 처리하는 이벤트에 대한 가시성을 제공하는 판매 로그를 지원합니다. 판매 로그는 Amazon CloudWatch Logs 인프라를 사용하여 원하는 대상으로 로그를 전송합니다.

판매 로그를 사용하려면 전송 대상을 구성해야 합니다. 지원되는 대상은 다음과 같습니다.

- Amazon CloudWatch Logs – 계정의 로그 그룹
- Amazon S3 - 계정의 S3 버킷
- Amazon Data Firehose – 계정의 Firehose 전송 스트림

지원되는 로그 유형

단일 로그 유형이 지원됩니다`APPLICATION_LOGS`. 이 로그 유형은 서비스가 내보내는 모든 운영 이벤트를 포함합니다.

로그 이벤트 유형

다음 표에는 AWS DevOps 에이전트가 로깅하는 이벤트가 요약되어 있습니다.

이벤트	설명	로그 수준
에이전트 인바운드 이벤트 수신	에이전트는 통합 소스에 의해 트리거되고 인바운드 이벤트(예: PagerDuty 인시던트 이벤트)를 수신합니다.	INFO
에이전트 인바운드 이벤트 삭제됨	인바운드 이벤트는 에이전트가 처리하기 전에 삭제되었습니다. 로그에는 이유(예: 잘못된 형식의 데이터)가 포함됩니다.	미정
에이전트 아웃바운드 통신 실패	타사 통합에 대한 아웃바운드 통신이 실패했습니다. 로그에는 작업 ID와 대상 식별자(예: 인증 오류)가 포함됩니다.	미정
대기 중인 토폴로지 생성	토폴로지 생성 작업이 처리를 위해 대기열에 추가되었습니다.	INFO
토폴로지 생성 시작됨	토폴로지 생성 작업이 처리를 시작했습니다.	INFO
토폴로지 생성 완료	토폴로지 생성 작업 처리가 완료되었습니다. 이 이벤트는 초기 생성, 업데이트 및 일일 새로 고침에 적용됩니다.	INFO

이벤트	설명	로그 수준
리소스 검색 실패	토폴로지 생성 중 리소스 검색에 오류가 발생했습니다.	오류
서비스 등록 실패	서비스 등록에서 복구할 수 없는 실패 발생	오류
Webhook 검증 실패	Devops 에이전트가 수신한 웹후크가 예상 스키마와 일치하지 않는 경우	오류
연결 검증 상태 업데이트	에이전트 스페이스 연결(일반적인 기본/보조 계정)이 유효에서 유효하지 않음으로 바뀌고 그 반대의 경우도 마찬가지입니다(예: 잘못된 역할로 인해 발생하며 서비스에서 수임할 수 없음).	오류/정보

권한

AWS DevOps Agent는 CloudWatch 벤딩 로그(V2 권한)를 사용하여 로그를 전송합니다. 로그 전송을 설정하려면 전송을 구성하는 IAM 역할에 다음 권한이 있어야 합니다.

- `aidevops:AllowVendedLogDeliveryForResource` - 에이전트 스페이스 리소스에 대한 로그 전송을 허용하는 데 필요합니다.
- CloudWatch Logs 전송 APIs(`logs:PutDeliverySource`, `logs:PutDeliveryDestinationlogs:CreateDelivery`, 및 관련 작업)에 대한 권한입니다.
- 선택한 전송 대상과 관련된 권한입니다.

각 대상 유형에 필요한 전체 IAM 정책은 Amazon CloudWatch Logs 사용 설명서의 다음 주제를 참조하세요.

- CloudWatch Logs로 전송된 로그
- Amazon S3로 보낸 로그

- Firehose로 전송된 로그

로그 전송 구성(콘솔)

AWS DevOps Agent는 AWS 관리 콘솔에서 로그 전송을 구성하기 위한 두 위치를 제공합니다.

- 서비스 등록 설정 페이지 - 서비스 수준 이벤트에 대한 로그 전송을 구성합니다. 이러한 로그는 서비스 ARN(`arn:aws:aidevops:<region>:<account-id>:service/<account-id>`)을 리소스로 사용합니다.
- 에이전트 스페이스 페이지 - 개별 에이전트 스페이스와 관련된 이벤트에 대한 로그 전송을 구성합니다. 이러한 로그는 에이전트 공간 ARN(`arn:aws:aidevops:<region>:<account-id>:agentspace/<agent-space-id>`)을 리소스로 사용합니다.

서비스 등록에 대한 로그 전송을 구성하려면

1. AWS 관리 콘솔에서 AWS DevOps 에이전트 콘솔을 엽니다.
2. 탐색 창에서 설정을 선택합니다.
3. 기능 공급자 > 로그 탭에서 구성을 선택합니다.
4. 대상 유형에서 다음 중 하나를 선택합니다.
5. CloudWatch Logs - 로그 그룹을 선택하거나 생성합니다.
6. Amazon S3 - S3 버킷 ARN을 입력합니다.
7. Amazon Data Firehose - Firehose 전송 스트림을 선택하거나 생성합니다.
8. 추가 설정 - 선택 사항의 경우 다음 옵션을 지정할 수 있습니다.
   a. 필드 선택에서 대상으로 전송할 로그 필드 이름을 선택합니다. 액세스 로그 필드 및 실시간 액세스 로그 필드의 하위 집합을 선택할 수 있습니다.
   b. (Amazon S3만 해당) 파티셔닝에서 로그 파일 데이터를 파티셔닝할 경로를 지정합니다.
   c. (Amazon S3만 해당) Hive 호환 파일 형식에서 확인란을 선택하여 Hive 호환 S3 경로를 사용할 수 있습니다. 이렇게 하면 Hive 호환 도구에 새 데이터를 쉽게 로드할 수 있습니다.
   d. 출력 형식에서 원하는 형식을 지정합니다.
   e. 필드 구분 기호에서 로그 필드를 구분하는 방법을 지정합니다.
9. 저장을 선택합니다.
10. 전송 상태가 활성으로 표시되는지 확인합니다.

에이전트 공간에 대한 로그 전송을 구성하려면

1. AWS 관리 콘솔에서 AWS DevOps 에이전트 콘솔을 엽니다.
2. 구성할 에이전트 공간을 선택합니다.
3. 구성 탭에서 구성을 선택합니다.
4. 대상 유형에서 다음 중 하나를 선택합니다.
5. CloudWatch Logs - 로그 그룹을 선택하거나 생성합니다.
6. Amazon S3 - S3 버킷 ARN을 입력합니다.
7. Amazon Data Firehose - Firehose 전송 스트림을 선택하거나 생성합니다.
8. 추가 설정 - *선택 사항 *의 경우 다음 옵션을 지정할 수 있습니다.
   a. 필드 선택에서 대상으로 전송할 로그 필드 이름을 선택합니다. 액세스 로그 필드 및 실시간 액세스 로그 필드의 하위 집합을 선택할 수 있습니다.
   b. (Amazon S3만 해당) 파티셔닝에서 로그 파일 데이터를 파티셔닝할 경로를 지정합니다.
   c. (Amazon S3만 해당) Hive 호환 파일 형식에서 확인란을 선택하여 Hive 호환 S3 경로를 사용할 수 있습니다. 이렇게 하면 Hive 호환 도구에 새 데이터를 쉽게 로드할 수 있습니다.
   d. 출력 형식에서 원하는 형식을 지정합니다.
   e. 필드 구분 기호에서 로그 필드를 구분하는 방법을 지정합니다.
9. 저장을 선택합니다.
10. 전송 상태가 활성으로 표시되는지 확인합니다.

로그 전송 구성(CloudWatch API)

CloudWatch Logs API를 사용하여 프로그래밍 방식으로 로그 전송을 구성할 수도 있습니다. 작동하는 로그 전송은 다음 세 가지 요소로 구성됩니다.

- DeliverySource - 로그를 생성하는 AWS DevOps 에이전트 스페이스 리소스를 나타냅니다.
- DeliveryDestination - 로그가 기록되는 대상을 나타냅니다.
- 전송 - 전송 소스를 전송 대상에 연결합니다.

1단계: 전송 소스 생성

PutDeliverySource 작업을 사용하여 전송 소스를 생성합니다. AWS DevOps 에이전트 스페이스 리소스의 ARN을 전달하고를 로그 유형APPLICATION_LOGS으로 지정합니다.

다음 예시에서는 에이전트 공간에 대한 전송 소스를 생성합니다.

```
{
    "name": "my-agent-space-delivery-source",
    "resourceArn": "arn:aws:aidevops:us-east-1:123456789012:agentspace/my-agent-space-
id",
    "logType": "APPLICATION_LOGS"
}
```

다음 예시에서는 서비스에 대한 전송 소스를 생성합니다.

```
{
    "name": "my-service-delivery-source",
    "resourceArn": "arn:aws:aidevops:us-east-1:123456789012:service",
    "logType": "APPLICATION_LOGS"
}
```

2단계: 전송 대상 생성

PutDeliveryDestination 작업을 사용하여 로그가 저장되는 위치를 구성합니다. Amazon CloudWatch Logs, Amazon S3 또는 Amazon Data Firehose를 선택할 수 있습니다.

다음 예시에서는 CloudWatch Logs 대상을 생성합니다.

```
{
    "name": "my-cwl-destination",
    "deliveryDestinationConfiguration": {
        "destinationResourceArn": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/
aidevops/my-agent-space"
    },
    "outputFormat": "json"
}
```

다음 예시에서는 Amazon S3 대상을 생성합니다.

```
{
    "name": "my-s3-destination",
    "deliveryDestinationConfiguration": {
        "destinationResourceArn": "arn:aws:s3:::my-aidevops-logs-bucket"
    },
    "outputFormat": "json"
}
```

다음 예시에서는 Amazon Data Firehose 대상을 생성합니다.

```
{
    "name": "my-firehose-destination",
    "deliveryDestinationConfiguration": {
        "destinationResourceArn": "arn:aws:firehose:us-
east-1:123456789012:deliverystream/my-aidevops-log-stream"
    },
    "outputFormat": "json"
}
```

> **Note**
>
> 교차 계정을 통해 로그를 전송하는 경우 대상 계정에서 PutDeliveryDestinationPolicy를 사용하여 전송을 승인해야 합니다.

CloudFormation을 사용하려면 다음을 사용할 수 있습니다.

- Delivery
- DeliveryDestination
- DeliverySource

`ResourceArn`은 `AgentSpaceArn`이며 `LogType`은 지원되는 로그 유형의 `APPLICATION_LOGS`여야 합니다.

3단계: 전송 생성

CreateDelivery 작업을 사용하여 전송 소스를 전송 대상에 연결합니다.

```
{
    "deliverySourceName": "my-agent-space-delivery-source",
    "deliveryDestinationArn": "arn:aws:logs:us-east-1:123456789012:delivery-
destination:my-cwl-destination"
}
```

AWS CloudFormation

다음 리소스와 함께 AWS CloudFormation을 사용하여 로그 전송을 구성할 수도 있습니다.

- AWS::Logs::DeliverySource
- AWS::Logs::DeliveryDestination
- AWS::Logs::Delivery

`ResourceArn`를 AWS DevOps 에이전트 공간 또는 서비스 ARN으로 설정하고를 `LogType`로 설정합니다`APPLICATION_LOGS`.

로그 스키마 참조

AWS DevOps Agent는 모든 이벤트 유형에서 공유 로그 스키마를 사용합니다. 모든 로그 이벤트가 모든 필드를 사용하는 것은 아닙니다.

다음 표에서는 로그 스키마의 필드를 설명합니다.

Field	Type	설명
event_timestamp	Long	이벤트 발생 시점의 Unix 타임스탬프
resource_arn	문자열	이벤트를 생성한 리소스의 ARN
optional_account_id	문자열	AWS 로그와 연결된 계정 ID입니다.
optional_level	문자열	로그 수준: `INFO`, `WARN`, `ERROR`
optional_agent_space_id	문자열	에이전트 공간의 식별자입니다.
optional_association_id	문자열	로그의 연결 식별자입니다.
optional_status	문자열	토폴로지 작업의 상태입니다.
optional_webhook_id	문자열	Webhook 식별자입니다.
optional_mcp_endpoint_url	문자열	MCP 서버 엔드포인트 URL

Field	Type	설명
optional_service_type	문자열	서비스 유형: `DYNATRACE` , `DATADOG`, `GITHUB`, `SLACK`, `SERVICENOW` .
optional_service_endpoint_url	문자열	타사 통합을 위한 엔드포인트 URL입니다.
optional_service_id	문자열	소스의 식별자입니다.
request_id	문자열	AWS CloudTrail 또는 지원 티켓과의 상관관계를 확인하기 위한 요청 식별자입니다.
optional_operation	문자열	수행된 작업의 이름입니다.
optional_task_type	문자열	에이전트 백로그 작업 유형: `INVESTIGATION` 또는 `EVALUATION`
optional_task_id	문자열	에이전트 백로그 작업 IDAgent 백로그 작업 식별자입니다.
optional_reference	문자열	에이전트 작업의 참조(예: Jira 티켓).
optional_error_type	문자열	오류 유형
optional_error_message	문자열	작업이 실패할 때 오류 설명입니다.
optional_details	문자열(JSON)	작업 파라미터 및 결과가 포함된 서비스별 이벤트 페이로드입니다.

로그 전송 관리 및 비활성화

AWS 관리 콘솔의 AWS DevOps 에이전트 콘솔에서 또는 CloudWatch Logs API를 사용하여 언제든지 로그 전송을 수정하거나 제거할 수 있습니다.

로그 전송 관리(콘솔)

1. AWS 관리 콘솔에서 AWS DevOps 에이전트 콘솔을 엽니다.
2. 설정 페이지(서비스 수준 로그의 경우) 또는 특정 에이전트 스페이스 페이지(에이전트 스페이스 수준 로그의 경우)로 이동합니다.
3. 구성 탭(에이전트 스페이스 수준 로그의 경우) 또는 기능 공급자 > 로그 탭(서비스 수준 로그의 경우)에서 수정할 전송을 선택합니다.
4. 필요에 따라 구성을 업데이트하고 저장을 선택합니다.

> 참고: 기존 전송의 대상 유형은 변경할 수 없습니다. 대상 유형을 변경하려면 현재 전송을 삭제하고 새 전송을 생성합니다.

로그 전송 비활성화(콘솔)

1. AWS 관리 콘솔에서 AWS DevOps 에이전트 콘솔을 엽니다.
2. 설정 페이지(서비스 수준 로그의 경우) 또는 특정 에이전트 스페이스 페이지(에이전트 스페이스 수준 로그의 경우)로 이동합니다.
3. 구성 탭(에이전트 스페이스 수준 로그의 경우) 또는 기능 공급자 > 로그 탭(서비스 수준 로그의 경우)에서 제거할 전송을 선택합니다.
4. 삭제를 선택하고 확인합니다.

로그 전송 비활성화(API)

API를 사용하여 로그 전송을 제거하려면 다음 순서로 리소스를 삭제합니다.

1. DeleteDelivery를 사용하여 전송을 삭제합니다.
2. DeleteDeliverySource를 사용하여 전송 소스를 삭제합니다.
3. (선택 사항) 전송 대상이 더 이상 필요하지 않은 경우 DeleteDeliveryDestination을 사용하여 삭제합니다.

> **Important**
>
> 로그를 생성하는 에이전트 공간 리소스를 삭제한 후(예: 에이전트 공간을 삭제한 후) 로그 전송 리소스를 제거하는 것은 사용자의 책임입니다. 이러한 리소스를 제거하지 않으면 분리된 전송 구성이 남아 있을 수 있습니다.

가격 책정

AWS DevOps 에이전트는 벤딩 로그 활성화에 대해 요금을 부과하지 않습니다. 그러나 선택한 로그 전송 대상에 따라 전송, 수집, 스토리지, 액세스에 대한 요금이 발생할 수 있습니다. 요금 세부 정보는 Amazon CloudWatch 요금의 로그 탭에서 판매 로그를 참조하세요.

대상별 요금은 다음을 참조하세요.

- Amazon CloudWatch Logs 요금
- Amazon S3 요금
- Amazon Data Firehose 요금

프라이빗 호스팅 도구에 연결

프라이빗 연결 개요

AWS DevOps 에이전트는 사용자 지정 모델 컨텍스트 프로토콜(MCP) 도구 및 에이전트에게 프라이빗 패키지 레지스트리, 자체 호스팅 관찰성 플랫폼, 내부 설명서 APIs 및 소스 제어 인스턴스와 같은 내부 시스템에 대한 액세스 권한을 부여하는 기타 통합을 통해 확장할 수 있습니다(참조AWS DevOps Agent에 대한 기능 구성). 이러한 서비스는 퍼블릭 인터넷 액세스가 제한되거나 없는 Amazon Virtual Private Cloud(Amazon VPC) 내에서 실행되는 경우가 많습니다. 즉, AWS DevOps Agent는 기본적으로 해당 서비스에 연결할 수 없습니다.

프라이빗 연결 AWS DevOps 에이전트를 사용하면 에이전트 스페이스를 퍼블릭 인터넷에 노출하지 않고 VPC에서 실행되는 서비스에 안전하게 연결할 수 있습니다. 프라이빗 연결은 MCP 서버, 자체 호스팅 Grafana 또는 Splunk 인스턴스, GitHub Enterprise Server 및 GitLab 자체 관리형과 같은 소스 제어 시스템을 포함하여 프라이빗 엔드포인트에 도달해야 하는 모든 통합과 함께 작동합니다.

> **Note**
>
> 프라이빗 호스팅 도구가 VPC 내에서 AWS DevOps 에이전트에 아웃바운드 요청을 하는 경우 네트워크 내에 유지되도록 VPC 엔드포인트를 사용하여이 트래픽을 보호할 수도 있습니다 AWS . 예를 들어 웹후크 이벤트를 통해 DevOps 에이전트를 트리거하는 도구와 함께 사용할 수 있습니다(참조the section called “Webhook를 통해 DevOps 에이전트 호출”). 자세한 내용은 the section called “VPC 엔드포인트(AWS PrivateLink)” 단원을 참조하십시오.

프라이빗 연결 작동 방식

프라이빗 연결은 AWS DevOps 에이전트와 VPC의 대상 리소스 간에 보안 네트워크 경로를 생성합니다. 후드에서 AWS DevOps 에이전트는 Amazon VPC Lattice를 사용하여이 보안 프라이빗 연결 경로를 설정합니다. VPC Lattice는 기본 네트워크 인프라를 관리하지 않고도 VPCs, 계정 및 컴퓨팅 유형 전반의 애플리케이션 간 통신을 연결, 보호 및 모니터링할 수 있는 애플리케이션 네트워킹 서비스입니다.

프라이빗 연결을 생성하면 다음이 발생합니다.

- 대상 서비스에 네트워크로 연결된 VPC, 서브넷 및 (선택 사항) 보안 그룹을 제공합니다.
- AWS DevOps Agent는 서비스 관리형 리소스 게이트웨이를 생성하고 지정한 서브넷에 탄력적 네트워크 인터페이스(ENIs)를 프로비저닝합니다.
- 에이전트는 리소스 게이트웨이를 사용하여 프라이빗 네트워크 경로를 통해 트래픽을 대상 서비스의 IP 주소 또는 DNS 이름으로 라우팅합니다.

리소스 게이트웨이는 AWS DevOps Agent에서 완벽하게 관리되며 계정에 읽기 전용 리소스(명명)로 표시됩니다`aidevops-{your-private-connection-name}`. 구성하거나 유지 관리할 필요가 없습니다. VPC에서 생성된 유일한 리소스는 지정한 서브넷의 ENIs입니다. 이러한 ENIs 프라이빗 트래픽의 진입점 역할을 하며 전적으로 서비스에 의해 관리됩니다. 인터넷으로부터의 인바운드 연결을 허용하지 않으며 자체 보안 그룹을 통해 트래픽을 완전히 제어할 수 있습니다.

보안

프라이빗 연결은 여러 보안 계층으로 설계되었습니다.

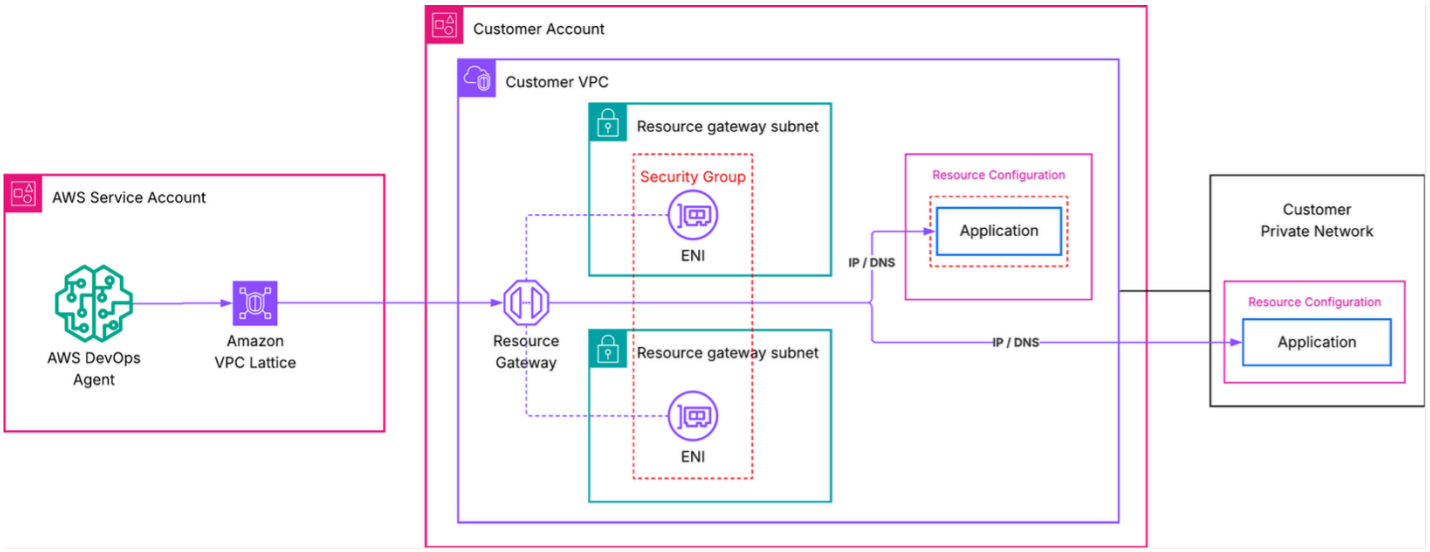
- 퍼블릭 인터넷 노출 없음 - AWS DevOps 에이전트와 대상 서비스 간의 모든 트래픽은 AWS 네트워크에 남아 있습니다. 서비스에 퍼블릭 IP 주소나 인터넷 게이트웨이가 필요하지 않습니다.

- 서비스 제어 리소스 게이트웨이 - 서비스 관리형 리소스 게이트웨이는 계정에서 읽기 전용입니다. AWS DevOps 에이전트만 사용할 수 있으며 다른 서비스 또는 보안 주체는 트래픽을 라우팅할 수 없습니다. 모든 VPC Lattice API 호출을 기록하는 AWS CloudTrail 로그에서 이를 확인할 수 있습니다.
- 보안 그룹 및 규칙 - 소유하고 관리하는 보안 그룹을 통해 ENIs에 대한 인바운드 및 아웃바운드 트래픽을 제어합니다. 보안 그룹을 지정하지 않으면 AWS DevOps Agent는 정의한 포트로 범위가 지정된 기본 보안 그룹을 생성합니다.
- 최소 권한이 있는 서비스 연결 역할 - AWS DevOps Agent는 서비스 연결 역할을 사용하여 필요한 VPC Lattice 및 Amazon EC2 리소스만 생성합니다. 이 역할은 태그가 지정된 리소스로 범위가 지정`AWSAIDevOpsManaged`되며 계정의 다른 리소스에 액세스할 수 없습니다.

> **Note**
>
> 조직에 VPC Lattice API 작업을 제한하는 서비스 제어 정책(SCPs)이 있는 경우 서비스 연결 역할을 통해 서비스 관리형 리소스 게이트웨이가 생성됩니다. SCPs 서비스 연결 역할에 필요한 작업을 허용하는지 확인합니다.

아키텍처

다음 다이어그램은 프라이빗 연결의 네트워크 경로를 보여줍니다.



이 아키텍처에서,

- AWS DevOps Agent는 대상 서비스에 대한 요청을 시작합니다.

- Amazon VPC Lattice는 VPC의 서비스 관리형 리소스 게이트웨이를 통해 요청을 라우팅합니다. 자체 VPC Lattice 리소스를 사용한 고급 설정은 기존 VPC Lattice 리소스를 사용한 고급 설정을 참조하세요.
- VPC의 ENI는 트래픽을 수신하여 대상 서비스의 IP 주소 또는 DNS 이름으로 전달합니다.
- 보안 그룹은 ENIs.
- 대상 서비스의 관점에서 요청은 VPC 내 ENIs의 프라이빗 IP 주소에서 시작됩니다.

프라이빗 연결 생성

AWS Management Console 또는 AWS CLI를 사용하여 프라이빗 연결을 생성할 수 있습니다.

> **Note**
>
> VPC Lattice에서 지원하지 않는 가용 영역은 `use1-az3`, , `usw1-az2`, `apne1-az3`, `apne2-az2`, `euc1-az2euw1-az4`, , `cac1-az3`입니다`ilc1-az2`.

사전 조건

프라이빗 연결을 생성하기 전에 다음이 있는지 확인합니다.

- 활성 에이전트 스페이스 - 계정에 기존 에이전트 스페이스가 필요합니다. 없으면 AWS DevOps 에이전트 시작하기 섹션을 참조하세요.
- 비공개로 연결할 수 있는 대상 서비스 - 리소스 게이트웨이가 배포된 VPC의 알려진 프라이빗 IP 주소 또는 DNS 이름에서 MCP 서버, 관찰성 플랫폼 또는 기타 서비스에 연결할 수 있어야 합니다. 리소스 게이트웨이 서브넷에서 라우팅할 수 있는 한 서비스는 동일한 VPC, 피어링된 VPC 또는 온프레미스에서 실행할 수 있습니다. 서비스는 연결을 생성할 때 지정한 포트에서 최소 TLS 버전이 1.2인 HTTPS 트래픽을 제공해야 합니다.
- VPC의 서브넷 - ENIs될 서브넷 1~20개를 식별합니다. 고가용성을 위해 여러 가용 영역에서 서브넷을 선택하는 것이 좋습니다. 이러한 서브넷에는 대상 서비스에 대한 네트워크 연결이 있어야 합니다. VPC Lattice는 가용 영역당 하나의 서브넷을 사용할 수 있습니다.
- (선택 사항) 보안 그룹 - 특정 규칙으로 트래픽을 제어하려면 ENIs에 연결할 보안 그룹 IDs를 최대 5개까지 준비합니다. 보안 그룹을 생략하면 AWS DevOps 에이전트가 기본 보안 그룹을 생성합니다.

프라이빗 연결은 계정 수준 리소스입니다. 프라이빗 연결을 생성한 후 동일한 호스트에 연결해야 하는 여러 통합 및 에이전트 스페이스에서 프라이빗 연결을 재사용할 수 있습니다.

콘솔을 사용하여 프라이빗 연결 생성

1. AWS DevOps 에이전트 콘솔을 엽니다.
2. 탐색 창에서 기능 공급자를 선택한 다음 프라이빗 연결을 선택합니다.
3. 새 연결 생성을 선택합니다.
4. 이름에와 같이 연결을 설명하는 이름을 입력합니다`my-mcp-tool-connection`.
5. VPC에서 리소스 게이트웨이 ENIs 배포할 VPC를 선택합니다.
6. 서브넷에서 하나 이상의 서브넷(최대 20개)을 선택합니다. 두 개 이상의 가용 영역에서 서브넷을 선택하는 것이 좋습니다.
7. IP 주소 유형에서 대상 서비스의 IP 주소 유형(`IPv4`, `IPv6`또는 `DualStack`)을 선택합니다.
8. (선택 사항) IPv4 주소 수에 IP 주소 유형으로 IPv4 또는 듀얼 스택을 선택한 경우 리소스 게이트웨이의 ENI당 IPv4 주소 수를 입력할 수 있습니다. 기본값은 ENI당 16개의 IPv4 주소입니다.
9. (선택 사항) 보안 그룹에서 기존 보안 그룹(최대 5개)을 선택하여 대상 서비스에 도달할 수 있는 트래픽을 제한합니다. 선택하지 않으면 기본 보안 그룹이 생성됩니다.
10. (선택 사항) 포트 범위에서 대상 애플리케이션이 수신하는 TCP 포트(예: `443` 또는)를 지정합니다`8080-8090`. 최대 11개의 포트 범위를 지정할 수 있습니다.
11. 호스트 주소에 대상 서비스의 IP 주소 또는 DNS 이름(예: `mcp.internal.example.com` 또는 `10.0.1.50`)을 입력합니다. 선택한 VPC에서 서비스에 연결할 수 있어야 합니다. DNS 이름을 선택하는 경우 선택한 VPC에서 확인할 수 있어야 합니다.
12. (선택 사항) 인증서 퍼블릭 키의 경우 지정한 호스트 주소가 프라이빗 인증 기관에서 발급한 TLS 인증서를 사용하는 경우 인증서의 PEM 인코딩 퍼블릭 키를 입력합니다. 이렇게 하면 AWS DevOps 에이전트가 대상 서비스에 대한 TLS 연결을 신뢰할 수 있습니다.
13. 연결 생성을 선택합니다.

연결 상태가 생성 진행 중으로 변경됩니다. 이 프로세스는 최대 10분이 걸릴 수 있습니다. 상태가 활성으로 변경되면 네트워크 경로가 준비된 것입니다.

상태가 생성 실패로 변경되면 다음을 확인합니다.

- 지정한 서브넷에 사용 가능한 IP 주소가 있습니다.
- 계정이 VPC Lattice 서비스 할당량에 도달하지 않았습니다.
- 제한적인 IAM 정책으로 인해 서비스 연결 역할이 리소스를 생성할 수 없습니다.

> **Note**
>
> 이러한 단계는 기능 공급자를 등록하는 `Create a new private connection` 동안를 선택하여 수행할 수도 있습니다. 자세한 내용은 기능 공급자와의 프라이빗 연결 사용을 참조하세요.

AWS CLI를 사용하여 프라이빗 연결 생성

다음 명령을 실행하여 프라이빗 연결을 생성합니다. 자리 표시자 값을 자신의 값으로 바꿉니다.

```
aws devops-agent create-private-connection \
    --name my-mcp-tool-connection \
    --mode '{
        "serviceManaged": {
            "hostAddress": "mcp.internal.example.com",
            "vpcId": "vpc-0123456789abcdef0",
            "subnetIds": [
                "subnet-0123456789abcdef0",
                "subnet-0123456789abcdef1"
            ],
            "securityGroupIds": [
                "sg-0123456789abcdef0"
            ],
            "portRanges": ["443"]
        }
    }'
```

응답에는 연결 이름과 상태가 포함됩니다`CREATE_IN_PROGRESS`.

```
{
    "name": "my-mcp-tool-connection",
    "status": "CREATE_IN_PROGRESS",
    "resourceGatewayId": "rgw-0123456789abcdef0",
    "hostAddress": "mcp.internal.example.com",
    "vpcId": "vpc-0123456789abcdef0"
}
```

연결 상태를 확인하려면 `describe-private-connection` 명령을 사용합니다.

```
aws devops-agent describe-private-connection \
```

```
    --name my-mcp-tool-connection
```

상태가 이면 프라이빗 연결을 사용할 준비가 된 ACTIVE것입니다.

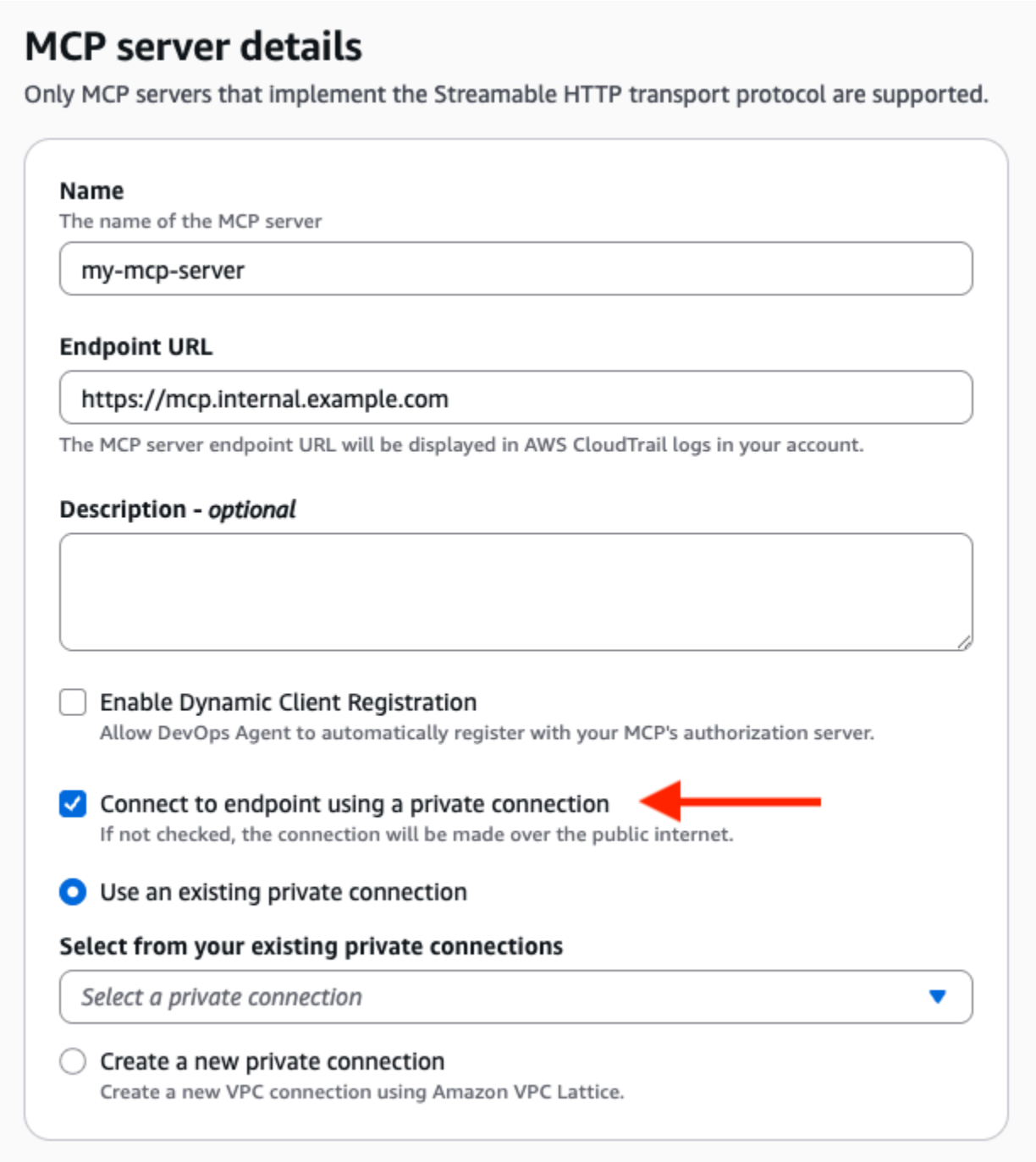
기능 공급자와의 프라이빗 연결 사용

프라이빗 연결을 사용하려면 기능 공급자를 등록하는 동안 프라이빗 연결에 연결할 수 있습니다. 프라이빗 연결에 사용할 수 있는 지원되는 기능은 GitHub, GitLab, MCP Server및 입니다Grafana. AWS Management Console 또는 AWS CLI를 사용하여이 단계를 수행할 수 있습니다.

> **Note**
>
> 기능 공급자를 등록할 때 AWS DevOps Agent는 엔드포인트에 연결할 수 있고 응답하는지 확인합니다. 등록을 완료하기 전에 대상 서비스가 실행 중이고 연결을 수락하고 있는지 확인합니다.

콘솔을 사용하여 기능 공급자와의 프라이빗 연결 사용

AWS DevOps 에이전트 콘솔에서 "프라이빗 연결을 사용하여 엔드포인트에 연결" 옵션을 선택하여 등록 중에 프라이빗 연결을 기능에 연결할 수 있습니다.



1. AWS DevOps 에이전트 콘솔을 열고 에이전트 공간으로 이동합니다.
2. 기능 공급자 섹션에서 등록을 선택합니다.
3. 프라이빗 연결에 사용할 기능 유형에 대해 등록을 선택합니다.
4. 등록 세부 정보 보기에서 프라이빗 연결을 사용하여 연결하려는 엔드포인트 URL(예: `https://mcp.internal.example.com`)을 입력합니다.
5. 프라이빗 연결을 사용하여 엔드포인트에 연결을 선택합니다.

6. 연결하려는 엔드포인트 URL에 해당하는 기존 프라이빗 연결을 선택하거나 새 프라이빗 연결 생성을 선택하여 생성합니다.
7. 기능 공급자의 등록 프로세스를 완료합니다.

AWS CLI를 사용하여 기능 공급자와의 프라이빗 연결 사용

`private-connection-name` 인수를 포함하여 프라이빗 연결에 기능을 등록할 수 있습니다. 다음은 `my-mcp-tool-connection` 프라이빗 연결을 사용하여 API 키 권한 부여로 MCP 서버를 등록하는 예입니다. 자리 표시자 값을 자신의 값으로 바꿉니다.

```
aws devops-agent register-service \
    --service mcpserver \
    --private-connection-name my-mcp-tool-connection \
    --service-details '{
        "mcpserver": {
            "name": "my-mcp-tool",
            "endpoint": "https://mcp.internal.example.com",
            "authorizationConfig": {
                "apiKey": {
                    "apiKeyName": "api-key",
                    "apiKeyValue": "secret-value",
                    "apiKeyHeader": "x-api-key"
                }
            }
        }
    }' \
    --region us-east-1
```

프라이빗 연결 확인

프라이빗 연결이 활성 상태에 도달하고 기능 공급자가 이를 활용한 후 AWS DevOps 에이전트가 대상 서비스에 도달할 수 있는지 확인합니다.

1. AWS DevOps 에이전트 콘솔을 열고 에이전트 공간으로 이동합니다.
2. 새 채팅 세션을 시작합니다.
3. 프라이빗 연결에서 지원하는 통합을 사용하는 명령을 호출합니다. 예를 들어 MCP 도구가 내부 지식 기반에 대한 액세스를 제공하는 경우 에이전트에게 해당 지식 기반이 필요한 질문을 합니다.
4. 에이전트가 프라이빗 서비스의 결과를 반환하는지 확인합니다.

연결이 실패하면 다음을 확인합니다.

- VPC Lattice 제한 - 리소스 게이트웨이 또는 기타 VPC Lattice 할당량 제한에 도달하지 않았는지 확인합니다.
- 보안 그룹 규칙 - ENIs에 연결된 보안 그룹이 서비스가 수신하는 포트에서 아웃바운드 트래픽을 허용하는지 확인합니다. 또한 서비스의 보안 그룹이 대상 포트에서 인바운드 트래픽을 허용하는지 확인합니다. 트래픽은 VPC CIDR 범위 내의 VPC Lattice 데이터 영역 IPs에서 도착합니다. 보안 그룹 참조(ENI 보안 그룹을 소스로 허용)를 사용하거나 VPC CIDR에서 인바운드를 허용할 수 있습니다.
- 서브넷 연결 - 선택한 서브넷이 서비스로 트래픽을 라우팅할 수 있는지 확인합니다. 서비스가 다른 서브넷에서 실행되는 경우 라우팅 테이블이 둘 사이의 트래픽을 허용하는지 확인합니다.
- 서비스 가용성 - 서비스가 실행 중이고 예상 포트에서 연결을 수락하고 있는지 확인합니다.
- 지원되지 않는 가용 영역 - 서브넷이 지원되는 가용 영역에 있는지 확인합니다. 를 실행`aws ec2 describe-subnets --subnet-ids <your-subnet-ids> --query 'Subnets[*].[SubnetId,AvailabilityZoneId]'`하고 위에 나열된 지원되지 않는 가용 영역을 확인합니다.

프라이빗 연결 삭제

AWS Management Console 또는 AWS CLI를 사용하여 미사용 프라이빗 연결을 삭제할 수 있습니다.

콘솔을 사용하여 프라이빗 연결 삭제

1. AWS DevOps 에이전트 콘솔을 엽니다.
2. 탐색 창에서 기능 공급자를 선택한 다음 프라이빗 연결을 선택합니다.
3. 삭제할 프라이빗 연결의 작업 메뉴를 선택하고 제거를 선택합니다.

프라이빗 연결은 "연결 제거" 상태로 표시되는 반면 AWS , DevOps Agent는 VPC에서 관리형 리소스 게이트웨이와 ENIs 제거합니다. 삭제가 완료되면 프라이빗 연결 목록에 연결이 더 이상 표시되지 않습니다.

AWS CLI를 사용하여 프라이빗 연결 삭제

```
aws devops-agent delete-private-connection \
    --name my-mcp-tool-connection
```

응답은 상태를 반환합니다DELETE_IN_PROGRESS. AWS DevOps 에이전트는 VPC에서 관리형 리소스 게이트웨이 및 ENIs 제거합니다. 삭제가 완료되면 프라이빗 연결 목록에 연결이 더 이상 표시되지 않습니다.

기존 VPC Lattice 리소스를 사용한 고급 설정

조직에서 이미 Amazon VPC Lattice를 사용하고 자체 리소스 구성을 관리하는 경우 자체 관리형 모드에서 프라이빗 연결을 생성할 수 있습니다. AWS DevOps 에이전트가 리소스 게이트웨이를 생성하도록 하는 대신 대상 서비스를 가리키는 기존 리소스 구성의 Amazon 리소스 이름(ARN)을 제공합니다.

이 접근 방식은 다음과 같은 경우에 유용합니다.

- 리소스 게이트웨이 및 리소스 구성 수명 주기를 완전히 제어하고자 합니다.
- 여러 AWS 계정 또는 서비스에서 리소스 구성을 공유해야 합니다.
- 자세한 트래픽 모니터링을 위해 VPC Lattice 액세스 로그가 필요합니다.
- hub-and-spoke 네트워크 아키텍처를 실행합니다.

AWS CLI를 사용하여 자체 관리형 프라이빗 연결을 생성하려면:

```
aws devops-agent create-private-connection \
    --name my-advanced-connection \
    --mode '{
        "selfManaged": {
            "resourceConfigurationId": "arn:aws:vpc-lattice:us-
east-1:123456789012:resourceconfiguration/rcfg-0123456789abcdef0"
        }
    }'
```

VPC Lattice 리소스 게이트웨이 및 리소스 구성 설정에 대한 자세한 내용은 Amazon VPC Lattice 사용 설명서를 참조하세요.

관련 주제

- the section called “VPC 엔드포인트(AWS PrivateLink)”
- the section called “MCP 서버 연결”
- AWS DevOps Agent에 대한 기능 구성
- AWS DevOps 에이전트 보안
- the section called “DevOps 에이전트 IAM 권한”

AWS DevOps 에이전트 보안

이 문서에서는 AWS DevOps Agent의 보안 고려 사항, 데이터 보호, 액세스 제어 및 규정 준수 기능에 대한 정보를 제공합니다. 이 정보를 사용하여 AWS DevOps 에이전트가 보안 및 규정 준수 요구 사항을 충족하도록 설계된 방법을 이해합니다.

다중 계층 보안

AWS DevOps Agent는 여러 계층에서 보안을 구현합니다. 에이전트의 IAM 역할에 더 광범위한 권한이 부여되더라도 에이전트는 자체 내부 액세스 제어를 적용하여 작업 범위를 제한합니다. 예를 들어 고객이 에이전트의 IAM 역할에 전체 Amazon S3 액세스 IAM 정책을 추가하는 경우 AWS DevOps 에이전트는 문제 해결을 위해 `AWSLogs` 접두사 이후의 로그만 읽도록 합니다.

AWS DevOps Agent에 대한 IAM 권한을 구성하고 여러 계층에서 보안을 구현할 때는 최소 권한 원칙을 따르는 것이 좋습니다. 심층 방어는 잘못된 단일 구성이 환경의 보안을 손상시키지 않도록 합니다.

에이전트 공간

에이전트 스페이스는 AWS DevOps 에이전트의 기본 보안 경계 역할을 합니다. 각 에이전트 공간:

- 자체 구성 및 권한으로 독립적으로 작동
- 에이전트가 액세스할 수 있는 AWS 계정과 리소스를 정의합니다.
- 타사 플랫폼에 대한 연결 설정

에이전트 스페이스는 보안을 보장하고 다양한 환경 또는 팀에서 의도하지 않은 액세스를 방지하기 위해 엄격한 격리를 유지합니다.

리전별 처리 및 데이터 흐름

AWS DevOps Agent는 리전별 처리 기능을 통해 전 세계적으로 운영됩니다. 에이전트는 구성된 에이전트 스페이스 내에서 액세스 권한이 부여된 모든 AWS 계정의 AWS 리전에서 운영 데이터를 검색합니다. 이 다중 리전 교차 계정 데이터 수집은 추론 처리를 위한 지리적 경계를 준수하면서 포괄적인 인시던트 분석을 보장합니다.

Amazon Bedrock 사용 및 리전 간 추론

AWS DevOps Agent는 지리 내에서 추론 요청을 처리할 최적의 리전을 자동으로 선택합니다. 이렇게 하면 사용 가능한 컴퓨팅 리소스와 모델 가용성이 극대화되고 최상의 고객 경험이 제공됩니다. 데이터는 에이전트 스페이스가 생성된 리전에만 저장되지만 다음 목록에 설명된 대로 입력 프롬프트 및 출력 결과가 해당 리전 외부에서 처리될 수 있습니다. 모든 데이터는 Amazon의 보안 네트워크를 통해 암호화되어 전송됩니다.

AWS DevOps Agent는 다음과 같이 요청이 시작된 지리적 영역 내의 사용 가능한 컴퓨팅 리소스로 추론 요청을 안전하게 라우팅합니다.

- 유럽 연합에서 시작된 추론 요청은 유럽 연합 내에서 처리됩니다.
- 미국에서 시작된 추론 요청은 미국 내에서 처리됩니다.
- 호주에서 시작된 추론 요청은 호주 내에서 처리됩니다.
- 일본 내에서 시작되는 추론 요청은 일본 내에서 처리됩니다.
- 추론 요청이 목록에 없는 영역에서 시작되는 경우 기본적으로 미국 내에서 처리됩니다.
- DevOps Agent 및 Bedrock은 고객 콘텐츠를 특정 리전으로 제한하는 서비스 제어 정책(SCPs) 또는 Control Tower의 고객 정책의 영향을 받지 않습니다.
- Bedrock은 지리 내 발신 리전 이외의 리전을 사용하여 상태 비저장 추론을 수행하여 성능과 가용성을 최적화할 수 있습니다.

ID 및 액세스 관리

인증 방법

AWS DevOps Agent는 AWS DevOps Agent Space 웹 앱에 로그인하는 두 가지 인증 방법을 제공합니다.

- AWS Identity Center 통합 - 기본 인증 방법은 HTTP 전용 쿠키를 사용한 세션 기반 인증과 함께 OAuth 2.0을 사용합니다. AWS Identity Center는 Okta, Ping Identity 및 Microsoft Entra ID와 같은 공급자를 포함한 표준 OIDC 및 SAML 프로토콜을 통해 외부 ID 공급자와 페더레이션할 수 있습니다. 이 방법은 ID 제공업체를 통한 멀티 팩터 인증을 지원합니다. AWS ID 센터는 기본적으로 최대 12시간의 세션 기간으로 설정되며 원하는 기간으로 구성할 수 있습니다.
- IAM 인증 링크 - 대체 방법을 사용하면 기존 AWS Management Console 세션에서 파생된 JWT 기반 토큰을 사용하여 AWS Management Console에서 웹 앱에 직접 액세스할 수 있습니다. 이 옵션은 전체 Identity Center 통합을 구현하기 전에 AWS DevOps 에이전트를 평가하고 Identity Center 기반 인

증을 통해 AWS DevOps 에이전트 웹 앱에 액세스할 수 없는 경우 관리 액세스 권한을 얻는 데 유용합니다. 세션은 10분으로 제한됩니다.

IAM 역할

AWS DevOps Agent는 IAM 역할을 사용하여 액세스 권한을 정의합니다.

- 기본 계정 역할 - 에이전트 스페이스를 생성하는 AWS 계정의 리소스에 대한 액세스 권한과 보조 계정 역할에 대한 액세스 권한을 에이전트에게 부여합니다.
- 보조 계정 역할 - 에이전트 스페이스에 연결된 추가 AWS 계정의 리소스에 대한 액세스 권한을 에이전트에게 부여합니다.
- 웹 앱 역할 - 웹 앱의 AWS DevOps 에이전트 조사 데이터 및 결과에 대한 액세스 권한을 사용자에게 부여합니다.

이러한 역할은 최소 권한 원칙에 따라 구성되어야 하며, 조사에 필요한 읽기 전용 권한만 부여해야 합니다.

데이터 보호

데이터 암호화

AWS DevOps Agent는 모든 고객 데이터를 암호화합니다.

- 유휴 시 암호화 AWS- 모든 데이터는 관리형 키로 암호화됩니다.
- 전송 중 암호화 - 검색된 모든 로그, 지표, 지식 항목, 티켓 메타데이터 및 기타 데이터는 에이전트의 프라이빗 네트워크 내부 및 외부 네트워크로 전송 중 암호화됩니다.

데이터 스토리지 및 보존

데이터는 에이전트 스페이스가 생성된 리전에 저장되지만, 추론 처리는 위의 Amazon Bedrock 사용 섹션에 설명된 대로 리전 내에서 발생할 수 있습니다.

개인 식별 정보(PII)

AWS DevOps Agent는 조사, 권장 사항 평가 또는 채팅 응답 중에 수집된 데이터를 요약할 때 PII 정보를 필터링하지 않습니다. 관찰성 로그에 저장하기 전에 PII 데이터를 수정하는 것이 좋습니다.

에이전트 저널 및 감사 로깅

에이전트 저널

인시던트 조사 및 예방 기능 모두 다음과 같은 세부 저널을 유지합니다.

- 모든 추론 단계 및 취해진 조치 로깅
- 에이전트 의사 결정 프로세스에 대한 완전한 투명성 생성
- 에이전트가 녹음한 후에는 수정할 수 없으므로 중요한 작업을 숨기지 않도록 프롬프트 주입과 같은 공격을 최소화할 수 있습니다.
- 조사 페이지의 모든 채팅 메시지 포함

AWS CloudTrail 통합

All AWS DevOps 에이전트 API 호출은 호스팅 AWS 계정 내에서 AWS CloudTrail에 의해 자동으로 캡처됩니다. CloudTrail에서 수집한 정보를 사용하여 다음을 확인할 수 있습니다.

- 에이전트에게 수행된 요청
- 요청을 보낸 IP 주소
- 요청한 사람
- 요청한 시기

프롬프트 주입 보호

즉각적인 주입 공격은 공격자가 악의적인 지침을 외부 데이터에 포함할 때 발생합니다. 웹 페이지 또는 문서, 생성형 AI 시스템이 나중에 처리할 것입니다. AWS DevOps Agent는 기본적으로 정상 작업의 일부로 많은 데이터 소스를 사용합니다. 로그, 리소스 태그, 및 기타 운영 데이터. AWS DevOps Agent는 아래 보호 조치를 통해 프롬프트 인젝션 공격으로부터 보호합니다. 그러나 연결된 모든 데이터 소스와 해당 데이터 소스에 대한 사용자 액세스를 신뢰할 수 있도록 하는 것이 중요합니다. 자세한 내용은 공동 책임 모델 섹션을 참조하세요.

프롬프트 주입 보호:

- 제한된 쓰기 기능 - 티켓 열기 및 지원 사례를 제외하고 에이전트가 사용할 수 있는 도구는 리소스를 변경할 수 없습니다. 이렇게 하면 악의적인 지침이 인프라 또는 애플리케이션을 수정하는 것을 방지할 수 있습니다.

- 계정 경계 적용 - AWS DevOps 에이전트는 기본 및 연결된 보조 AWS 계정에서 에이전트에 할당된 역할이 허용하는 경계 내에서만 작동합니다. 에이전트는 구성된 범위를 벗어나는 리소스에 액세스하거나 수정할 수 없습니다.
- AI 안전 보호 - AWS DevOps 에이전트는 AI 안전 레벨 3(ASL-3) 보호 기능이 있는 모델을 사용합니다. 이러한 보호에는 에이전트 동작에 영향을 미치기 전에 프롬프트 주입 공격을 감지하고 방지하는 분류자가 포함됩니다.
- 변경 불가능한 감사 추적 - 에이전트 저널은 모든 추론 단계와 취해진 조치를 기록합니다. 저널 항목은 에이전트가 기록한 후에는 수정할 수 없으므로 프롬프트 인젝션 공격이 악의적인 작업을 숨기는 것을 방지할 수 있습니다.

AWS DevOps Agent는 프롬프트 주입 공격에 대해 여러 계층의 보호를 제공하지만 특정 구성은 위험을 증가시킬 수 있습니다.

- 사용자 지정 MCP 서버 도구 bring-your-own MCP 기능을 사용하면 에이전트에 사용자 지정 도구를 도입할 수 있으므로 즉시 삽입할 수 있는 추가 기회를 제공할 수 있습니다. 사용자 지정 도구에는 native AWS DevOps 에이전트 도구와 동일한 보안 제어가 없을 수 있으며, 악의적인 지침은 의도하지 않은 방식으로 이러한 도구를 활용할 수 있습니다. 자세한 내용은 공동 책임 모델 섹션을 참조하세요.
- 승인된 사용자 공격 - AWS 계정 경계 또는 연결된 도구 내에서 작업할 권한이 있는 사용자는 에이전트에 대한 공격을 시도할 가능성이 더 높습니다. 이러한 사용자는 로그 또는 리소스 태그와 같이 에이전트가 사용하는 데이터 소스를 수정하여 에이전트가 처리할 악성 지침을 더 쉽게 포함할 수 있습니다.

이러한 위험을 완화하려면:

1. 에이전트 스페이스에 배포하기 전에 사용자 지정 MCP 서버를 주의 깊게 검토하고 테스트합니다.
   a. 읽기 전용 작업만 수행할 수 있는지 확인합니다.
   b. MCP 서버가 액세스하는 외부 도구의 사용자가 신뢰할 수 있는 엔터티인지 확인합니다. MCP와 인터페이스하는 AWS DevOps 에이전트는 이러한 도구 사용자와 AWS DevOps 에이전트 간에 설정된 암시적 신뢰 관계에 의존하기 때문입니다.
2. 사용자에게 에이전트에 데이터를 제공하는 시스템에 대한 액세스 권한을 부여할 때 최소 권한 원칙을 적용합니다.
3. 에이전트 스페이스에 연결된 MCP 서버를 정기적으로 감사
4. 허용 목록에 있는 URLs에서 검색된 모든 콘텐츠는 에이전트의 동작을 조작하려고 할 수 있으므로 허용 목록에 신뢰할 수 있는 소스만 포함합니다.

통합 보안

AWS DevOps Agent는 여러 통합 유형을 지원하며, 각 통합 유형에는 자체 보안 모델이 있습니다.

- 기본 양방향 통합 - 에이전트에 데이터를 전송하고 에이전트로부터 업데이트를 수신할 수 있는 기본 제공 통합입니다. 공급업체의 인증 방법을 사용합니다.
- MCP 서버 - OAuth 2.0 인증 흐름 및 API 키를 활용하여 외부 시스템과 안전하게 통신하는 원격 모델 컨텍스트 프로토콜 서버입니다.
- Webhook 트리거 - 티켓 또는 관찰성 시스템과 같은 원격 서비스의 조사 트리거입니다. Webhook는 보안을 위해 해시 기반 메시지 인증 코드(HMAC)를 사용합니다.
- 아웃바운드 통신 - Slack 및 티켓팅 시스템과 같은 통합은 에이전트로부터 업데이트를 수신하지만 양방향 통신을 아직 지원하지 않습니다.

등록 공급자

일부 외부 도구는 계정 수준에서 인증되고 계정의 모든 에이전트 스페이스 간에 공유됩니다. 이러한 도구를 등록할 때 계정 수준에서 한 번 인증하면 각 에이전트 스페이스가 등록된 연결 내의 특정 리소스에 연결할 수 있습니다.

다음 도구는 계정 수준 등록을 사용합니다.

- GitHub - 인증에 OAuth 흐름을 사용합니다. 계정 수준에서 GitHub를 등록한 후 각 에이전트 스페이스는 GitHub 조직 내의 특정 리포지토리에 연결할 수 있습니다.
- Dynatrace - OAuth 토큰 인증을 사용합니다. 계정 수준에서 Dynatrace를 등록한 후 각 에이전트 스페이스는 특정 Dynatrace 환경 또는 모니터링 구성에 연결할 수 있습니다.
- Slack - OAuth 토큰 인증을 사용합니다. 계정 수준에서 Slack을 등록한 후 각 에이전트 스페이스는 특정 Slack 채널 채널에 연결할 수 있습니다.
- Datadog - 인증에 OAuth 흐름과 함께 MCP를 사용합니다. 계정 수준에서 Datadog을 등록한 후 각 에이전트 스페이스는 특정 Datadog 모니터링 리소스에 연결할 수 있습니다.
- New Relic - API 키 인증을 사용합니다. 계정 수준에서 New Relic을 등록한 후 각 에이전트 스페이스는 특정 New Relic 모니터링 구성에 연결할 수 있습니다.
- Splunk - 보유자 토큰 인증을 사용합니다. 계정 수준에서 Splunk를 등록한 후 각 에이전트 스페이스는 특정 Splunk 데이터 소스에 연결할 수 있습니다.
- GitLab - 액세스 토큰 인증을 사용합니다. 계정 수준에서 GitLab을 등록한 후 각 에이전트 스페이스는 특정 GitLab 리포지토리에 연결할 수 있습니다.

- ServiceNow - OAuth 클라이언트 키/토큰 인증을 사용합니다. 계정 수준에서 ServiceNow를 등록한 후 각 에이전트 스페이스는 특정 ServiceNow 인스턴스 또는 티켓 대기열에 연결할 수 있습니다.
- 일반 퍼블릭 액세스 가능 원격 MCP 서버 - 인증에 OAuth 흐름을 사용합니다. 계정 수준에서 원격 MCP 서버를 등록한 후 각 에이전트 스페이스는 해당 서버에서 노출되는 특정 리소스에 연결할 수 있습니다.

네트워크 연결

AWS DevOps Agent는 타사 시스템 및 원격 MCP 서버에 연결하여 조사 및 기타 작업을 수행합니다.

AWS DevOps 에이전트에서 시스템으로의 인바운드 트래픽

AWS DevOps Agent는 인프라에 대한 인바운드 트래픽으로 도착하는 타사 시스템 및 원격 MCP 서버에 대한 아웃바운드 연결을 시작합니다. 이 트래픽을 보호하는 방법은 도구가 호스팅되는 방식에 따라 달라집니다.

- 프라이빗 호스팅 도구 - AWS VPC 내에서 도구에 연결할 수 있는 경우 AWS DevOps 에이전트 프라이빗 연결을 사용하여 트래픽을 AWS 네트워크로 격리하고 퍼블릭 인터넷에서 차단할 수 있습니다. 자세한 내용은 the section called “프라이빗 호스팅 도구에 연결” 단원을 참조하십시오.
- 퍼블릭 호스팅 도구 - 퍼블릭 인터넷을 통해 도구에 연결할 수 있고 IP 허용 목록 또는 방화벽 규칙을 사용하는 경우 다음 AWS DevOps 에이전트 소스 IP 주소의 인바운드 트래픽을 허용해야 합니다.
  - 아시아 태평양(시드니)(ap-southeast-2)
    - `13.237.95.197`
    - `13.238.84.102`
  - 아시아 태평양(도쿄)(ap-northeast-1)
    - `13.192.12.233`
    - `35.74.181.230`
    - `57.183.50.158`
  - 유럽(프랑크푸르트)(eu-central-1)
    - `18.158.110.140`
    - `52.57.96.160`
    - `52.59.55.56`
  - 유럽(아일랜드)(eu-west-1)
    - `34.251.85.24`

  - 52.30.157.157
  - 52.51.192.222
- 미국 동부(버지니아 북부)(us-east-1)
  - 34.228.181.128
  - 44.219.176.187
  - 54.226.244.221
- 미국 서부(오레곤)(us-west-2)
  - 34.212.16.133
  - 52.89.67.212
  - 54.187.135.61

VPC에서 AWS DevOps 에이전트로의 아웃바운드 트래픽

AWS VPC에서 AWS DevOps 에이전트로의 아웃바운드 트래픽(예: 사용the section called “Webhook를 통해 DevOps 에이전트 호출”)의 경우 VPC 엔드포인트를 사용하여이 네트워크 트래픽을 AWS 네트워크에 격리된 상태로 유지할 수 있습니다. 자세한 내용은 the section called “VPC 엔드포인트(AWS PrivateLink)” 단원을 참조하십시오.

공동 책임 모델

AWS 책임

AWS 는 다음에 대한 책임이 있습니다.

- 에이전트가 검색한 데이터의 보안 유지
- 에이전트가 사용할 수 있는 기본 도구 보호
- AWS DevOps 에이전트를 실행하는 인프라 보호

고객 책임

고객은 다음에 대한 책임이 있습니다.

- 에이전트 공간에 대한 사용자 액세스 관리

- 악의적인 프롬프트 주입을 시도하는 데 사용할 수 있는 로그, CloudTrail 이벤트, 티켓 등을 생성하는 서비스 및 리소스와 같이 에이전트에 입력을 제공하는 외부 시스템의 신뢰할 수 있는 사용자로 액세스를 제한합니다.
- 연결된 모든 데이터 소스에 프롬프트 주입 공격을 시도하는 데 사용할 가능성이 낮은 신뢰할 수 있는 데이터가 있는지 확인합니다.
- bring-your-own MCP 서버 통합이 안전하게 작동하도록 보장
- 에이전트에 할당된 IAM 역할의 범위가 적절하게 지정되었는지 확인
- 관찰성 로그 및 기타 에이전트 데이터 소스에 저장하기 전에 PII 데이터 수정
- bring-your-own MCP 서버를 포함하여 연결된 데이터 소스에 읽기 전용 권한만 부여하는 권장 관행을 따릅니다.

데이터 사용량

AWS 는 모델을 훈련하거나 제품을 개선하기 위해 에이전트 데이터, 채팅 메시지 또는 통합 데이터 소스의 데이터를 사용하지 않습니다. AWS DevOps 에이전트 스페이스는 고객 제품 내 피드백을 사용하여 에이전트의 응답 및 조사를 개선하지만 서비스 자체를 개선하는 데는 사용하지 AWS 않습니다.

규정 준수

미리 보기에서 AWS DevOps 에이전트는 SOC 2, PCI-DSS, ISO 27001 또는 FedRAMP를 포함한 표준을 준수하지 않습니다. AWS 는 나중에 사용할 수 있는 규정 준수 인증을 발표할 예정입니다.

DevOps 에이전트 IAM 권한

AWS DevOps Agent는 서비스별 AWS Identity and Access Management(IAM) 작업을 사용하여 기능 및 기능에 대한 액세스를 제어합니다. 이러한 작업은 사용자가 AWS DevOps 에이전트 콘솔 및 운영자 웹 앱에서 수행할 수 있는 작업을 결정합니다. 이는 에이전트 자체가 리소스를 조사하는 데 사용하는 AWS 서비스 API 권한과는 별개입니다.

에이전트 액세스 제한에 대한 자세한 내용은 AWS 계정의 에이전트 액세스 제한을 참조하세요.

에이전트 스페이스 관리 작업

이러한 작업은 에이전트 공간 구성 및 관리에 대한 액세스를 제어합니다.

- aidevops:GetAgentSpace - 사용자가 구성, 상태 및 관련 계정을 포함하여 에이전트 스페이스에 대한 세부 정보를 볼 수 있도록 허용합니다. AWS Management Console에서 에이전트 스페이스에 액세스하려면 사용자에게이 권한이 필요합니다.
- aidevops:GetAssociation - 사용자가 IAM 역할 구성 및 연결 상태를 포함하여 특정 계정 연결에 대한 세부 정보를 볼 수 있습니다.
- aidevops:ListAssociations - 사용자가 기본 및 보조 AWS 계정을 포함하여 에이전트 스페이스에 대해 구성된 모든 계정 연결을 나열할 수 있습니다.

조사 및 실행 작업

이러한 작업은 인시던트 조사 기능에 대한 액세스를 제어합니다.

- aidevops:ListExecutions - 사용자가 작업과 관련된 조사, 완화, 평가 및 채팅 대화에 대한 ID, 상태 등을 포함한 실행 메타데이터를 볼 수 있습니다.
- aidevops:ListJournalRecords - 사용자가 에이전트의 추론 단계, 수행한 작업 및 조사, 완화, 평가 및 채팅 대화 중에 상담한 데이터 소스를 보여주는 세부 로그에 액세스할 수 있도록 허용합니다. 이는 에이전트가 어떻게 결론에 도달했는지 이해하는 데 유용합니다.

채팅 관리 작업

채팅이 작동하려면 다음과 같은 IAM 권한이 필요합니다.

- aidevops:ListChats – 사용자가 채팅 대화 기록을 나열하고 액세스할 수 있습니다.
- aidevops:CreateChat – 사용자가 새 채팅 대화를 생성할 수 있습니다.
- aidevops:SendMessage – 사용자가 쿼리를 제출하고 스트리밍 응답을 수신할 수 있습니다.

토폴로지 및 검색 작업

이러한 작업은 애플리케이션 리소스 매핑 기능에 대한 액세스를 제어합니다.

- aidevops:DiscoverTopology – 사용자가 에이전트 스페이스에 대한 토폴로지 검색 및 매핑을 트리거할 수 있습니다. 이 작업은 AWS 계정을 스캔하고 애플리케이션 리소스 토폴로지를 구축하는 프로세스를 시작합니다.

예방 및 권장 조치

이러한 작업은 방지 기능에 대한 액세스를 제어합니다.

- aidevops:ListGoals - 사용자가 에이전트가 최근 인시던트 패턴을 기반으로 작업하는 예방 목표와 목표를 볼 수 있도록 허용합니다.
- aidevops:ListRecommendations - 사용자가 우선 순위 및 범주를 포함하여 방지 기능에서 생성된 모든 권장 사항을 볼 수 있도록 허용합니다.
- aidevops:GetRecommendation - 사용자가 방지했을 인시던트 및 구현 지침을 포함하여 특정 권장 사항에 대한 자세한 정보를 볼 수 있습니다.

백로그 작업 관리 작업

이러한 작업은 권장 사항을 백로그 작업으로 관리하는 기능을 제어합니다.

- aidevops:CreateBacklogTask – 사용자가 인시던트 조사 또는 예방 평가 작업을 생성할 수 있도록 허용합니다.
- aidevops:UpdateBacklogTask - 사용자가 완화 계획을 승인하거나 활성 조사 또는 평가를 취소할 수 있도록 허용합니다.
- aidevops:GetBacklogTask - 사용자가 특정 작업에 대한 세부 정보를 검색할 수 있습니다.
- aidevops:ListBacklogTasks - 사용자가 에이전트 스페이스에 대한 작업을 나열하고 작업 유형, 상태, 우선 순위 또는 생성 시간을 기준으로 필터링할 수 있습니다.

지식 관리 작업

이러한 작업은 에이전트가 조사 중에 사용할 수 있는 사용자 지정 지식을 추가하고 관리하는 기능을 제어합니다.

- aidevops:CreateKnowledgeItem - 사용자가 에이전트가 참조해야 하는 기술, 문제 해결 가이드 또는 애플리케이션별 정보와 같은 사용자 지정 지식 항목을 추가할 수 있도록 허용합니다.
- aidevops:ListKnowledgeItems - 사용자가 에이전트 스페이스에 대해 구성된 모든 지식 항목을 볼 수 있도록 허용합니다.
- aidevops:GetKnowledgeItem - 사용자가 특정 지식 항목의 세부 정보를 검색할 수 있습니다.
- aidevops:UpdateKnowledgeItem - 사용자가 기존 지식 항목을 수정하여 정보를 최신 상태로 유지할 수 있습니다.

- aidevops:DeleteKnowledgeItem - 사용자가 더 이상 관련이 없는 지식 항목을 제거할 수 있습니다.

AWS 통합 작업 지원

이러한 작업은 AWS 지원 사례와의 통합을 제어합니다.

- aidevops:InitiateChatForCase - 사용자가 조사에서 직접 AWS Support와의 채팅 세션을 시작할 수 있도록 허용하여 인시던트에 대한 컨텍스트를 자동으로 제공합니다.
- aidevops:EndChatForCase - 사용자가 활성 AWS 지원 사례 채팅 세션을 종료할 수 있도록 허용합니다.
- aidevops:DescribeSupportLevel – 사용자가 계정의 AWS 지원 플랜 수준을 확인하여 사용 가능한 지원 옵션을 결정할 수 있습니다.

사용 및 모니터링 작업

이러한 작업은 사용 정보에 대한 액세스를 제어합니다.

- aidevops:GetAccountUsage - 사용자가 조사 시간, 예방 평가 시간, 채팅 요청 및 이번 달 사용량에 대한 AWS DevOps 에이전트 월별 할당량을 볼 수 있도록 허용합니다.

일반적인 IAM 정책 예제

관리자 정책

이 정책은 모든 AWS DevOps 에이전트 기능에 대한 전체 액세스 권한을 부여합니다.

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

연산자 정책

이 정책은 관리 기능 없이 조사 및 예방 기능에 대한 액세스 권한을 부여합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aidevops:GetAgentSpace",
        "aidevops:InvokeAgent",
        "aidevops:ListExecutions",
        "aidevops:ListJournalRecords",
        "aidevops:ListAssociations",
        "aidevops:GetAssociation",
        "aidevops:DiscoverTopology",
        "aidevops:ListRecommendations",
        "aidevops:GetRecommendation",
        "aidevops:CreateBacklogTask",
        "aidevops:UpdateBacklogTask",
        "aidevops:GetBacklogTask",
        "aidevops:ListBacklogTasks",
        "aidevops:ListKnowledgeItems",
        "aidevops:GetKnowledgeItem",
        "aidevops:InitiateChatForCase",
        "aidevops:EndChatForCase",
        "aidevops:ListChats",
        "aidevops:CreateChat",
        "aidevops:SendMessage",
        "aidevops:ListGoals",
        "aidevops:CreateKnowledgeItem",
        "aidevops:UpdateKnowledgeItem",
        "aidevops:DescribeSupportLevel",
        "aidevops:ListPendingMessages"
      ],
      "Resource": "*"
    }
  ]
}
```

읽기 전용 정책

이 정책은 조사 및 권장 사항에 대한 보기 전용 액세스 권한을 부여합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aidevops:GetAgentSpace",
        "aidevops:ListAssociations",
        "aidevops:GetAssociation",
        "aidevops:ListExecutions",
        "aidevops:ListJournalRecords",
        "aidevops:ListRecommendations",
        "aidevops:GetRecommendation",
        "aidevops:ListBacklogTasks",
        "aidevops:GetBacklogTask",
        "aidevops:ListKnowledgeItems",
        "aidevops:GetKnowledgeItem",
        "aidevops:GetAccountUsage"
      ],
      "Resource": "*"
    }
  ]
}
```

AWS DevOps 에이전트에 서비스 연결 역할 사용

AWS DevOps Agent는 AWS Identity and Access Management(IAM) 서비스 연결 역할을 사용합니다. 서비스 연결 역할은 AWS DevOps 에이전트에 직접 연결된 고유한 유형의 IAM 역할입니다. 서비스 연결 역할은 AWS DevOps Agent에서 사전 정의하며 서비스가 사용자를 대신하여 다른 AWS 서비스를 호출하는 데 필요한 모든 권한을 포함합니다.

서비스 연결 역할 권한

`AWSServiceRoleForAIDevOps` 서비스 연결 역할은 역할을 수임하기 위해 `aidevops.amazonaws.com` 서비스 보안 주체를 신뢰합니다.

역할은 다음 권한이 `AWSServiceRoleForAIDevOpsPolicy` 있는 관리형 정책을 사용합니다.

- `cloudwatch:PutMetricData` - AWS/AIDevOps CloudWatch 네임스페이스에 사용량 지표를 게시합니다. AWS/AIDevOps 네임스페이스만 허용하도록 `cloudwatch:namespace` 조건에 의해 범위가 지정됩니다.
- `vpc-lattice:CreateResourceGateway` - 프라이빗 연결을 위한 VPC Lattice 리소스 게이트웨이를 생성합니다. 서비스가 AWSAIDevOpsManaged 태그를 전달하는 리소스 게이트웨이만 생성할 수 있도록 `aws:RequestTag/AWSAIDevOpsManaged` 조건에 따라 범위가 지정됩니다.
- `vpc-lattice:TagResource` - VPC Lattice 리소스 게이트웨이에 태그를 지정합니다. `aws:RequestTag/AWSAIDevOpsManaged` 조건에 따라 범위가 지정됩니다.
- `vpc-lattice:DeleteResourceGateway` - VPC Lattice 리소스 게이트웨이를 삭제합니다. 서비스가 생성한 리소스 게이트웨이만 삭제할 수 있도록 `aws:ResourceTag/AWSAIDevOpsManaged` 조건에 따라 범위가 지정됩니다.
- `vpc-lattice:GetResourceGateway` - VPC Lattice 리소스 게이트웨이에 대한 정보를 검색합니다. 서비스가 생성한 리소스 게이트웨이만 읽을 수 있도록 `aws:ResourceTag/AWSAIDevOpsManaged` 조건에 따라 범위가 지정됩니다.
- `ec2:DescribeVpcs`, `ec2:DescribeSubnets`, `ec2:DescribeSecurityGroups`- 리소스 게이트웨이를 구성하는 데 필요한 VPC 네트워킹 리소스에 대한 정보를 검색합니다. EC2 API는 호출 설명에 대한 리소스 수준 권한을 지원하지 않으므로 이러한 읽기 전용 작업은 모든 VPC 리소스에 적용됩니다.
- `iam:CreateServiceLinkedRole` - 리소스 게이트웨이 작업에 필요한 VPC Lattice 서비스 연결 역할을 생성합니다. 이 권한은 `vpc-lattice.amazonaws.com` 서비스 보안 주체로만 범위가 지정되며 다른 서비스에 대한 서비스 연결 역할을 생성하는 데 사용할 수 없습니다.

서비스 연결 역할 생성

AWSServiceRoleForAIDevOps 서비스 연결 역할을 수동으로 생성할 필요가 없습니다. AWS DevOps 에이전트 사용을 시작하면 서비스에서 서비스 연결 역할을 생성합니다.

서비스가 사용자를 대신하여 역할을 생성하도록 허용하려면 `iam:CreateServiceLinkedRole` 권한이 있어야 합니다. 최소 권한 원칙을 따르`aidevops.amazonaws.com`려면의 `iam:AWSServiceName` 조건으로이 권한의 범위를 조정하는 것이 좋습니다. 자세한 내용은 서비스 연결 역할 권한을 참조하세요.

서비스 연결 역할 편집

AWSServiceRoleForAIDevOps 서비스 연결 역할은 편집할 수 없습니다. 역할이 생성된 후에는 다양한 엔터티가 이름으로 역할을 참조할 수 있으므로 역할 이름을 변경할 수 없습니다. 하지만 IAM을 사용하여 역할의 설명을 편집할 수 있습니다. 자세한 내용은 서비스 연결 역할 편집을 참조하세요.

서비스 연결 역할 삭제

더 이상 AWS DevOps 에이전트를 사용할 필요가 없는 경우 AWSServiceRoleForAIDevOps 서비스 연결 역할을 삭제하는 것이 좋습니다. 역할을 삭제하려면 먼저 에이전트 스페이스에 구성된 프라이빗 연결을 제거해야 합니다. 서비스 연결 역할을 삭제해도 서비스에서 이전에 생성한 로 태그가 지정된 VPC Lattice 리소스 게이트웨이AWSAIDevOpsManaged는 자동으로 제거되지 않습니다. 이러한 리소스 게이트웨이가 더 이상 필요하지 않은 경우 수동으로 삭제해야 합니다. 자세한 내용은 서비스 연결 역할 삭제를 참조하세요.

AWS AWS DevOps 에이전트에 대한 관리형 정책

AWS 는에서 생성하고 관리하는 독립 실행형 IAM 정책을 제공하여 많은 일반적인 사용 사례를 처리합니다 AWS. 이러한 AWS 관리형 정책은 일반적인 사용 사례에 필요한 권한을 부여하므로 필요한 권한을 조사할 필요가 없습니다. 자세한 내용은 _IAM 사용 설명서_의 AWS 관리형 정책을 참조하세요.

계정의 사용자에게 연결할 수 있는 다음 AWS 관리형 정책은 AWS DevOps 에이전트에 고유합니다.

AIDevOpsAgentReadOnlyAccess

AWS 관리 콘솔을 통해 Amazon DevOps 에이전트에 대한 읽기 전용 액세스 권한을 제공합니다.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "AIDevOpsAgentReadOnlyAccess",
   "Effect": "Allow",
   "Action": [
    "aidevops:Get*",
    "aidevops:List*",
    "aidevops:SearchServiceAccessibleResource"
   ],
   "Resource": "*"
  }
 ]
```

```
}
```

AIDevOpsAgentFullAccess

AWS 관리 콘솔을 통해 Amazon DevOps 에이전트에 대한 전체 액세스 권한을 제공합니다.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "AIDevOpsAgentSpaceAccess",
   "Effect": "Allow",
   "Action": [
    "aidevops:CreateAgentSpace",
    "aidevops:DeleteAgentSpace",
    "aidevops:GetAgentSpace",
    "aidevops:ListAgentSpaces",
    "aidevops:UpdateAgentSpace"
   ],
   "Resource": "*"
  },
  {
   "Sid": "AIDevOpsServiceAccess",
   "Effect": "Allow",
   "Action": [
    "aidevops:DeregisterService",
    "aidevops:GetService",
    "aidevops:ListServices",
    "aidevops:RegisterService",
    "aidevops:SearchServiceAccessibleResource"
   ],
   "Resource": "*"
  },
  {
   "Sid": "AIDevOpsAssociationAccess",
   "Effect": "Allow",
   "Action": [
    "aidevops:AssociateService",
    "aidevops:DisassociateService",
    "aidevops:GetAssociation",
    "aidevops:ListAssociations",
    "aidevops:UpdateAssociation",
    "aidevops:ValidateAwsAssociations"
```

```
    ],
    "Resource": "*"
   },
   {
    "Sid": "AIDevOpsWebhookAccess",
    "Effect": "Allow",
    "Action": [
     "aidevops:ListWebhooks"
    ],
    "Resource": "*"
   },
   {
    "Sid": "AIDevOpsOperatorAppAccess",
    "Effect": "Allow",
    "Action": [
     "aidevops:DisableOperatorApp",
     "aidevops:EnableOperatorApp",
     "aidevops:GetOperatorApp",
     "aidevops:UpdateOperatorAppIdpConfig"
    ],
    "Resource": "*"
   },
   {
    "Sid": "AIDevOpsKnowledgeAccess",
    "Effect": "Allow",
    "Action": [
     "aidevops:CreateKnowledgeItem",
     "aidevops:DeleteKnowledgeItem",
     "aidevops:GetKnowledgeItem",
     "aidevops:ListKnowledgeItems",
     "aidevops:ListKnowledgeItemVersions",
     "aidevops:UpdateKnowledgeItem"
    ],
    "Resource": "*"
   },
   {
    "Sid": "AIDevOpsBacklogAccess",
    "Effect": "Allow",
    "Action": [
     "aidevops:CreateBacklogTask",
     "aidevops:GetBacklogTask",
     "aidevops:ListBacklogTasks",
     "aidevops:ListGoals",
     "aidevops:UpdateBacklogTask",
```

```
      "aidevops:UpdateGoal"
     ],
     "Resource": "*"
    },
    {
     "Sid": "AIDevOpsRecommendationAccess",
     "Effect": "Allow",
     "Action": [
      "aidevops:GetRecommendation",
      "aidevops:ListRecommendations",
      "aidevops:UpdateRecommendation"
     ],
     "Resource": "*"
    },
    {
     "Sid": "AIDevOpsAgentChatAccess",
     "Effect": "Allow",
     "Action": [
      "aidevops:CreateChat",
      "aidevops:ListChats",
      "aidevops:ListPendingMessages",
      "aidevops:SendMessage"
     ],
     "Resource": "*"
    },
    {
     "Sid": "AIDevOpsJournalAccess",
     "Effect": "Allow",
     "Action": [
      "aidevops:ListExecutions",
      "aidevops:ListJournalRecords"
     ],
     "Resource": "*"
    },
    {
     "Sid": "AIDevOpsTopologyAccess",
     "Effect": "Allow",
     "Action": [
      "aidevops:DiscoverTopology"
     ],
     "Resource": "*"
    },
    {
     "Sid": "AIDevOpsSupportAccess",
```

```
    "Effect": "Allow",
    "Action": [
     "aidevops:DescribeSupportLevel",
     "aidevops:EndChatForCase",
     "aidevops:InitiateChatForCase"
    ],
    "Resource": "*"
   },
   {
    "Sid": "AIDevOpsUsageAccess",
    "Effect": "Allow",
    "Action": [
     "aidevops:GetAccountUsage"
    ],
    "Resource": "*"
   },
   {
    "Sid": "AIDevOpsTaggingAccess",
    "Effect": "Allow",
    "Action": [
     "aidevops:ListTagsForResource",
     "aidevops:TagResource",
     "aidevops:UntagResource"
    ],
    "Resource": "*"
   },
   {
    "Sid": "AIDevOpsVendedLogs",
    "Effect": "Allow",
    "Action": [
     "aidevops:AllowVendedLogDeliveryForResource"
    ],
    "Resource": "*"
   }
  ]
}
```

AIDevOpsOperatorAppAccessPolicy

에이전트 스페이스에 AWS DevOps 운영자 웹 앱을 사용할 수 있는 액세스 권한을 제공합니다.

```
{
 "Version": "2012-10-17",
```

```
"Statement": [
 {
  "Sid": "AllowOperatorAgentSpaceActions",
  "Effect": "Allow",
  "Action": [
   "aidevops:GetAgentSpace",
   "aidevops:GetAssociation",
   "aidevops:ListAssociations",
   "aidevops:CreateBacklogTask",
   "aidevops:GetBacklogTask",
   "aidevops:UpdateBacklogTask",
   "aidevops:ListBacklogTasks",
   "aidevops:ListJournalRecords",
   "aidevops:DiscoverTopology",
   "aidevops:ListGoals",
   "aidevops:ListRecommendations",
   "aidevops:ListExecutions",
   "aidevops:GetRecommendation",
   "aidevops:UpdateRecommendation",
   "aidevops:CreateKnowledgeItem",
   "aidevops:ListKnowledgeItems",
   "aidevops:ListKnowledgeItemVersions",
   "aidevops:GetKnowledgeItem",
   "aidevops:UpdateKnowledgeItem",
   "aidevops:DeleteKnowledgeItem",
   "aidevops:ListPendingMessages",
   "aidevops:InitiateChatForCase",
   "aidevops:EndChatForCase",
   "aidevops:DescribeSupportLevel",
   "aidevops:ListChats",
   "aidevops:CreateChat",
   "aidevops:SendMessage"
  ],
  "Resource": "arn:aws:aidevops:*:*:agentspace/${aws:PrincipalTag/AgentSpaceId}",
  "Condition": {
   "StringEquals": {
    "aws:ResourceAccount": "${aws:PrincipalAccount}"
   }
  }
 },
 {
  "Sid": "AllowOperatorAccountActions",
  "Effect": "Allow",
  "Action": [
```

```
    "aidevops:GetAccountUsage"
   ],
   "Resource": "*",
   "Condition": {
    "StringEquals": {
     "aws:ResourceAccount": "${aws:PrincipalAccount}"
    }
   }
  },
  {
   "Sid": "AllowSupportOperatorActions",
   "Effect": "Allow",
   "Action": [
    "support:DescribeCases",
    "support:InitiateChatForCase",
    "support:DescribeSupportLevel"
   ],
   "Resource": "*",
   "Condition": {
    "StringEquals": {
     "aws:ResourceAccount": "${aws:PrincipalAccount}"
    }
   }
  }
 ]
}
```

AIDevOpsAgentAccessPolicy

AWS DevOps 에이전트가 고객 AWS 리소스에 대한 조사를 수행하고 분석을 수행하는 데 필요한 권한을 제공합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AIOPSServiceAccess",
            "Effect": "Allow",
            "Action": [
                "access-analyzer:GetAnalyzer",
                "access-analyzer:List*",
                "acm-pca:Describe*",
                "acm-pca:GetCertificate",
```

```
                "acm-pca:GetCertificateAuthorityCertificate",
                "acm-pca:GetCertificateAuthorityCsr",
                "acm-pca:List*",
                "acm:DescribeCertificate",
                "acm:GetAccountConfiguration",
                "aidevops:GetKnowledgeItem",
                "aidevops:ListKnowledgeItems",
                "airflow:List*",
                "amplify:GetApp",
                "amplify:GetBranch",
                "amplify:GetDomainAssociation",
                "amplify:List*",
                "aoss:BatchGetCollection",
                "aoss:BatchGetLifecyclePolicy",
                "aoss:BatchGetVpcEndpoint",
                "aoss:GetAccessPolicy",
                "aoss:GetSecurityConfig",
                "aoss:GetSecurityPolicy",
                "aoss:List*",
                "appconfig:GetApplication",
                "appconfig:GetConfigurationProfile",
                "appconfig:GetEnvironment",
                "appconfig:GetHostedConfigurationVersion",
                "appconfig:List*",
                "appflow:Describe*",
                "appflow:List*",
                "application-autoscaling:Describe*",
                "application-signals:BatchGetServiceLevelObjectiveBudgetReport",
                "application-signals:GetService",
                "application-signals:GetServiceLevelObjective",
                "application-signals:List*",
                "applicationinsights:Describe*",
                "applicationinsights:List*",
                "apprunner:Describe*",
                "apprunner:List*",
                "appstream:Describe*",
                "appstream:List*",
                "appsync:GetApiAssociation",
                "appsync:GetDataSource",
                "appsync:GetDomainName",
                "appsync:GetFunction",
                "appsync:GetGraphqlApi",
                "appsync:GetGraphqlApiEnvironmentVariables",
                "appsync:GetIntrospectionSchema",
```

```
                "appsync:GetResolver",
                "appsync:GetSourceApiAssociation",
                "appsync:List*",
                "aps:Describe*",
                "aps:List*",
                "arc-zonal-shift:GetManagedResource",
                "arc-zonal-shift:List*",
                "athena:GetCapacityAssignmentConfiguration",
                "athena:GetCapacityReservation",
                "athena:GetDataCatalog",
                "athena:GetNamedQuery",
                "athena:GetPreparedStatement",
                "athena:GetWorkGroup",
                "athena:List*",
                "auditmanager:GetAssessment",
                "auditmanager:List*",
                "autoscaling:Describe*",
                "backup-gateway:GetHypervisor",
                "backup-gateway:List*",
                "backup:Describe*",
                "backup:GetBackupPlan",
                "backup:GetBackupSelection",
                "backup:GetBackupVaultAccessPolicy",
                "backup:GetBackupVaultNotifications",
                "backup:GetRestoreTestingPlan",
                "backup:GetRestoreTestingSelection",
                "backup:List*",
                "batch:DescribeComputeEnvironments",
                "batch:DescribeJobQueues",
                "batch:DescribeSchedulingPolicies",
                "batch:List*",
                "bedrock:GetAgent",
                "bedrock:GetAgentActionGroup",
                "bedrock:GetAgentAlias",
                "bedrock:GetAgentKnowledgeBase",
                "bedrock:GetDataSource",
                "bedrock:GetGuardrail",
                "bedrock:GetKnowledgeBase",
                "bedrock:List*",
                "budgets:Describe*",
                "budgets:List*",
                "ce:Describe*",
                "ce:GetAnomalyMonitors",
                "ce:GetAnomalySubscriptions",
```

```
                "ce:List*",
                "chatbot:Describe*",
                "chatbot:GetMicrosoftTeamsChannelConfiguration",
                "chatbot:List*",
                "cleanrooms-ml:GetTrainingDataset",
                "cleanrooms-ml:List*",
                "cleanrooms:GetAnalysisTemplate",
                "cleanrooms:GetCollaboration",
                "cleanrooms:GetConfiguredTable",
                "cleanrooms:GetConfiguredTableAnalysisRule",
                "cleanrooms:GetConfiguredTableAssociation",
                "cleanrooms:GetMembership",
                "cleanrooms:List*",
                "cloudformation:Describe*",
                "cloudformation:GetResource",
                "cloudformation:GetStackPolicy",
                "cloudformation:GetTemplate",
                "cloudformation:List*",
                "cloudfront:Describe*",
                "cloudfront:GetCachePolicy",
                "cloudfront:GetCloudFrontOriginAccessIdentity",
                "cloudfront:GetContinuousDeploymentPolicy",
                "cloudfront:GetDistribution",
                "cloudfront:GetDistributionConfig",
                "cloudfront:GetFunction",
                "cloudfront:GetKeyGroup",
                "cloudfront:GetMonitoringSubscription",
                "cloudfront:GetOriginAccessControl",
                "cloudfront:GetOriginRequestPolicy",
                "cloudfront:GetPublicKey",
                "cloudfront:GetRealtimeLogConfig",
                "cloudfront:GetResponseHeadersPolicy",
                "cloudfront:List*",
                "cloudtrail:Describe*",
                "cloudtrail:GetChannel",
                "cloudtrail:GetEventConfiguration",
                "cloudtrail:GetEventDataStore",
                "cloudtrail:GetEventSelectors",
                "cloudtrail:GetInsightSelectors",
                "cloudtrail:GetQueryResults",
                "cloudtrail:GetResourcePolicy",
                "cloudtrail:GetTrail",
                "cloudtrail:GetTrailStatus",
                "cloudtrail:List*",
```

```
                "cloudtrail:LookupEvents",
                "cloudtrail:StartQuery",
                "cloudwatch:Describe*",
                "cloudwatch:GenerateQuery",
                "cloudwatch:GetDashboard",
                "cloudwatch:GetInsightRuleReport",
                "cloudwatch:GetMetricData",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:GetMetricStream",
                "cloudwatch:GetService",
                "cloudwatch:GetServiceLevelObjective",
                "cloudwatch:List*",
                "codeartifact:Describe*",
                "codeartifact:GetDomainPermissionsPolicy",
                "codeartifact:GetRepositoryPermissionsPolicy",
                "codeartifact:List*",
                "codebuild:BatchGetFleets",
                "codebuild:List*",
                "codecommit:GetRepository",
                "codecommit:GetRepositoryTriggers",
                "codedeploy:BatchGetDeployments",
                "codedeploy:BatchGetDeploymentTargets",
                "codedeploy:GetApplication",
                "codedeploy:GetDeploymentConfig",
                "codedeploy:GetDeploymentTarget",
                "codedeploy:List*",
                "codeguru-profiler:Describe*",
                "codeguru-profiler:GetNotificationConfiguration",
                "codeguru-profiler:GetPolicy",
                "codeguru-profiler:List*",
                "codeguru-reviewer:Describe*",
                "codeguru-reviewer:List*",
                "codepipeline:GetPipeline",
                "codepipeline:GetPipelineState",
                "codepipeline:List*",
                "codestar-connections:GetConnection",
                "codestar-connections:GetRepositoryLink",
                "codestar-connections:GetSyncConfiguration",
                "codestar-connections:List*",
                "codestar-notifications:Describe*",
                "codestar-notifications:List*",
                "cognito-identity:DescribeIdentityPool",
                "cognito-identity:GetIdentityPoolRoles",
                "cognito-identity:ListIdentityPools",
```

```
                "cognito-identity:ListTagsForResource",
                "cognito-idp:AdminListGroupsForUser",
                "cognito-idp:DescribeIdentityProvider",
                "cognito-idp:DescribeResourceServer",
                "cognito-idp:DescribeRiskConfiguration",
                "cognito-idp:DescribeUserImportJob",
                "cognito-idp:DescribeUserPool",
                "cognito-idp:DescribeUserPoolDomain",
                "cognito-idp:GetGroup",
                "cognito-idp:GetLogDeliveryConfiguration",
                "cognito-idp:GetUICustomization",
                "cognito-idp:GetUserPoolMfaConfig",
                "cognito-idp:GetWebACLForResource",
                "cognito-idp:ListGroups",
                "cognito-idp:ListIdentityProviders",
                "cognito-idp:ListResourceServers",
                "cognito-idp:ListUserPoolClients",
                "cognito-idp:ListUserPools",
                "cognito-idp:ListTagsForResource",
                "comprehend:Describe*",
                "comprehend:List*",
                "config:Describe*",
                "config:GetStoredQuery",
                "config:List*",
                "connect:Describe*",
                "connect:GetTaskTemplate",
                "connect:List*",
                "databrew:Describe*",
                "databrew:List*",
                "datapipeline:Describe*",
                "datapipeline:GetPipelineDefinition",
                "datapipeline:List*",
                "datasync:Describe*",
                "datasync:List*",
                "deadline:GetFarm",
                "deadline:GetFleet",
                "deadline:GetLicenseEndpoint",
                "deadline:GetMonitor",
                "deadline:GetQueue",
                "deadline:GetQueueEnvironment",
                "deadline:GetQueueFleetAssociation",
                "deadline:GetStorageProfile",
                "deadline:List*",
                "detective:GetMembers",
```

```
                "detective:List*",
                "devicefarm:GetDevicePool",
                "devicefarm:GetInstanceProfile",
                "devicefarm:GetNetworkProfile",
                "devicefarm:GetProject",
                "devicefarm:GetTestGridProject",
                "devicefarm:GetVPCEConfiguration",
                "devicefarm:List*",
                "devops-guru:Describe*",
                "devops-guru:GetResourceCollection",
                "devops-guru:List*",
                "dms:Describe*",
                "dms:List*",
                "ds:Describe*",
                "dynamodb:Describe*",
                "dynamodb:GetResourcePolicy",
                "dynamodb:List*",
                "ec2:Describe*",
                "ec2:GetAssociatedEnclaveCertificateIamRoles",
                "ec2:GetIpamPoolAllocations",
                "ec2:GetIpamPoolCidrs",
                "ec2:GetManagedPrefixListEntries",
                "ec2:GetNetworkInsightsAccessScopeContent",
                "ec2:GetSnapshotBlockPublicAccessState",
                "ec2:GetTransitGatewayMulticastDomainAssociations",
                "ec2:GetTransitGatewayRouteTableAssociations",
                "ec2:GetTransitGatewayRouteTablePropagations",
                "ec2:GetVerifiedAccessEndpointPolicy",
                "ec2:GetVerifiedAccessGroupPolicy",
                "ec2:GetVerifiedAccessInstanceWebAcl",
                "ec2:SearchLocalGatewayRoutes",
                "ec2:SearchTransitGatewayRoutes",
                "ecr:Describe*",
                "ecr:GetLifecyclePolicy",
                "ecr:GetRegistryPolicy",
                "ecr:GetRepositoryPolicy",
                "ecr:List*",
                "ecs:Describe*",
                "ecs:List*",
                "eks:AccessKubernetesApi",
                "eks:Describe*",
                "eks:List*",
                "elasticache:Describe*",
                "elasticache:List*",
```

```
                "elasticbeanstalk:Describe*",
                "elasticbeanstalk:List*",
                "elasticfilesystem:Describe*",
                "elasticloadbalancing:GetResourcePolicy",
                "elasticloadbalancing:GetTrustStoreCaCertificatesBundle",
                "elasticloadbalancing:GetTrustStoreRevocationContent",
                "elasticloadbalancing:Describe*",
                "elasticmapreduce:Describe*",
                "elasticmapreduce:List*",
                "emr-containers:Describe*",
                "emr-containers:List*",
                "emr-serverless:GetApplication",
                "emr-serverless:List*",
                "es:Describe*",
                "es:List*",
                "events:Describe*",
                "events:List*",
                "evidently:GetExperiment",
                "evidently:GetFeature",
                "evidently:GetLaunch",
                "evidently:GetProject",
                "evidently:GetSegment",
                "evidently:List*",
                "firehose:Describe*",
                "firehose:List*",
                "fis:GetExperimentTemplate",
                "fis:GetTargetAccountConfiguration",
                "fis:List*",
                "fms:GetNotificationChannel",
                "fms:GetPolicy",
                "fms:List*",
                "forecast:Describe*",
                "forecast:List*",
                "frauddetector:BatchGetVariable",
                "frauddetector:Describe*",
                "frauddetector:GetDetectors",
                "frauddetector:GetDetectorVersion",
                "frauddetector:GetEntityTypes",
                "frauddetector:GetEventTypes",
                "frauddetector:GetExternalModels",
                "frauddetector:GetLabels",
                "frauddetector:GetListElements",
                "frauddetector:GetListsMetadata",
                "frauddetector:GetModelVersion",
```

```
                "frauddetector:GetOutcomes",
                "frauddetector:GetRules",
                "frauddetector:GetVariables",
                "frauddetector:List*",
                "fsx:Describe*",
                "gamelift:Describe*",
                "gamelift:List*",
                "globalaccelerator:Describe*",
                "globalaccelerator:List*",
                "glue:GetDatabase",
                "glue:GetDatabases",
                "glue:GetJob",
                "glue:GetRegistry",
                "glue:GetSchema",
                "glue:GetSchemaVersion",
                "glue:GetTable",
                "glue:GetTags",
                "glue:GetTrigger",
                "glue:List*",
                "glue:querySchemaVersionMetadata",
                "grafana:Describe*",
                "grafana:List*",
                "greengrass:Describe*",
                "greengrass:GetDeployment",
                "greengrass:List*",
                "groundstation:GetConfig",
                "groundstation:GetDataflowEndpointGroup",
                "groundstation:GetMissionProfile",
                "groundstation:List*",
                "guardduty:GetDetector",
                "guardduty:GetFilter",
                "guardduty:GetIPSet",
                "guardduty:GetMalwareProtectionPlan",
                "guardduty:GetMasterAccount",
                "guardduty:GetMembers",
                "guardduty:GetThreatIntelSet",
                "guardduty:List*",
                "health:DescribeEvents",
                "health:DescribeEventDetails",
                "healthlake:Describe*",
                "healthlake:List*",
                "iam:GetGroup",
                "iam:GetGroupPolicy",
                "iam:GetInstanceProfile",
```

```
                "iam:GetLoginProfile",
                "iam:GetOpenIDConnectProvider",
                "iam:GetPolicy",
                "iam:GetPolicyVersion",
                "iam:GetRole",
                "iam:GetRolePolicy",
                "iam:GetSAMLProvider",
                "iam:GetServerCertificate",
                "iam:GetServiceLinkedRoleDeletionStatus",
                "iam:GetUser",
                "iam:GetUserPolicy",
                "iam:ListAttachedRolePolicies",
                "iam:ListOpenIDConnectProviders",
                "iam:ListRolePolicies",
                "iam:ListRoles",
                "iam:ListServerCertificates",
                "iam:ListVirtualMFADevices",
                "identitystore:DescribeGroup",
                "identitystore:DescribeGroupMembership",
                "identitystore:ListGroupMemberships",
                "identitystore:ListGroups",
                "imagebuilder:GetComponent",
                "imagebuilder:GetContainerRecipe",
                "imagebuilder:GetDistributionConfiguration",
                "imagebuilder:GetImage",
                "imagebuilder:GetImagePipeline",
                "imagebuilder:GetImageRecipe",
                "imagebuilder:GetInfrastructureConfiguration",
                "imagebuilder:GetLifecyclePolicy",
                "imagebuilder:GetWorkflow",
                "imagebuilder:List*",
                "inspector2:List*",
                "inspector:Describe*",
                "inspector:List*",
                "internetmonitor:GetMonitor",
                "internetmonitor:List*",
                "iot:Describe*",
                "iot:GetPackage",
                "iot:GetPackageVersion",
                "iot:GetPolicy",
                "iot:GetThingShadow",
                "iot:GetTopicRule",
                "iot:GetTopicRuleDestination",
                "iot:GetV2LoggingOptions",
```

```
                "iot:List*",
                "iotanalytics:Describe*",
                "iotanalytics:List*",
                "iotevents:Describe*",
                "iotevents:List*",
                "iotsitewise:Describe*",
                "iotsitewise:List*",
                "iotwireless:GetDestination",
                "iotwireless:GetDeviceProfile",
                "iotwireless:GetFuotaTask",
                "iotwireless:GetMulticastGroup",
                "iotwireless:GetNetworkAnalyzerConfiguration",
                "iotwireless:GetServiceProfile",
                "iotwireless:GetWirelessDevice",
                "iotwireless:GetWirelessGateway",
                "iotwireless:GetWirelessGatewayTaskDefinition",
                "iotwireless:List*",
                "ivs:GetChannel",
                "ivs:GetEncoderConfiguration",
                "ivs:GetPlaybackRestrictionPolicy",
                "ivs:GetRecordingConfiguration",
                "ivs:GetStage",
                "ivs:List*",
                "ivschat:GetLoggingConfiguration",
                "ivschat:GetRoom",
                "ivschat:List*",
                "kafka:Describe*",
                "kafka:GetClusterPolicy",
                "kafka:List*",
                "kafkaconnect:Describe*",
                "kafkaconnect:List*",
                "kendra:Describe*",
                "kendra:List*",
                "kinesis:Describe*",
                "kinesis:GetResourcePolicy",
                "kinesis:List*",
                "kinesisanalytics:Describe*",
                "kinesisanalytics:List*",
                "kinesisvideo:Describe*",
                "kms:DescribeKey",
                "kms:ListResourceTags",
                "kms:ListKeys",
                "kms:GetKeyPolicy",
                "kms:GetKeyRotationStatus",
```

```
                "kms:ListAliases",
                "kms:ListKeyRotations",
                "lakeformation:Describe*",
                "lakeformation:GetLFTag",
                "lakeformation:GetResourceLFTags",
                "lakeformation:List*",
                "lambda:GetAlias",
                "lambda:GetCodeSigningConfig",
                "lambda:GetEventSourceMapping",
                "lambda:GetFunctionCodeSigningConfig",
                "lambda:GetFunctionConfiguration",
                "lambda:GetFunctionEventInvokeConfig",
                "lambda:GetFunctionRecursionConfig",
                "lambda:GetFunctionUrlConfig",
                "lambda:GetLayerVersion",
                "lambda:GetLayerVersionPolicy",
                "lambda:GetPolicy",
                "lambda:GetProvisionedConcurrencyConfig",
                "lambda:GetRuntimeManagementConfig",
                "lambda:List*",
                "launchwizard:GetDeployment",
                "launchwizard:List*",
                "license-manager:GetLicense",
                "license-manager:List*",
                "lightsail:GetAlarms",
                "lightsail:GetBuckets",
                "lightsail:GetCertificates",
                "lightsail:GetContainerServices",
                "lightsail:GetDisk",
                "lightsail:GetDisks",
                "lightsail:GetInstance",
                "lightsail:GetInstances",
                "lightsail:GetLoadBalancer",
                "lightsail:GetLoadBalancers",
                "lightsail:GetLoadBalancerTlsCertificates",
                "lightsail:GetStaticIp",
                "lightsail:GetStaticIps",
                "logs:Describe*",
                "logs:FilterLogEvents",
                "logs:GetDataProtectionPolicy",
                "logs:GetDelivery",
                "logs:GetDeliveryDestination",
                "logs:GetDeliveryDestinationPolicy",
                "logs:GetDeliverySource",
```

```
                "logs:GetLogAnomalyDetector",
                "logs:GetLogDelivery",
                "logs:GetLogGroupFields",
                "logs:GetQueryResults",
                "logs:List*",
                "logs:StartQuery",
                "logs:StopLiveTail",
                "logs:StopQuery",
                "logs:TestMetricFilter",
                "m2:GetApplication",
                "m2:GetEnvironment",
                "m2:List*",
                "macie2:GetAllowList",
                "macie2:GetCustomDataIdentifier",
                "macie2:GetFindingsFilter",
                "macie2:GetMacieSession",
                "macie2:List*",
                "mediaconnect:Describe*",
                "mediaconnect:List*",
                "medialive:Describe*",
                "medialive:GetCloudWatchAlarmTemplate",
                "medialive:GetCloudWatchAlarmTemplateGroup",
                "medialive:GetEventBridgeRuleTemplate",
                "medialive:GetEventBridgeRuleTemplateGroup",
                "medialive:GetSignalMap",
                "medialive:List*",
                "mediapackage-vod:Describe*",
                "mediapackage-vod:List*",
                "mediapackage:Describe*",
                "mediapackage:List*",
                "mediapackagev2:GetChannel",
                "mediapackagev2:GetChannelGroup",
                "mediapackagev2:GetChannelPolicy",
                "mediapackagev2:GetOriginEndpoint",
                "mediapackagev2:GetOriginEndpointPolicy",
                "mediapackagev2:List*",
                "memorydb:Describe*",
                "memorydb:List*",
                "mobiletargeting:GetInAppTemplate",
                "mobiletargeting:List*",
                "mq:Describe*",
                "mq:List*",
                "network-firewall:Describe*",
                "network-firewall:List*",
```

```
                "networkmanager:Describe*",
                "networkmanager:GetConnectAttachment",
                "networkmanager:GetConnectPeer",
                "networkmanager:GetCoreNetwork",
                "networkmanager:GetCoreNetworkPolicy",
                "networkmanager:GetCustomerGatewayAssociations",
                "networkmanager:GetDevices",
                "networkmanager:GetLinkAssociations",
                "networkmanager:GetLinks",
                "networkmanager:GetSites",
                "networkmanager:GetSiteToSiteVpnAttachment",
                "networkmanager:GetTransitGatewayPeering",
                "networkmanager:GetTransitGatewayRegistrations",
                "networkmanager:GetTransitGatewayRouteTableAttachment",
                "networkmanager:GetVpcAttachment",
                "networkmanager:List*",
                "oam:GetLink",
                "oam:GetSink",
                "oam:GetSinkPolicy",
                "oam:List*",
                "omics:GetAnnotationStore",
                "omics:GetReferenceStore",
                "omics:GetRunGroup",
                "omics:GetSequenceStore",
                "omics:GetVariantStore",
                "omics:GetWorkflow",
                "omics:List*",
                "organizations:Describe*",
                "organizations:List*",
                "osis:GetPipeline",
                "osis:List*",
                "payment-cryptography:GetAlias",
                "payment-cryptography:GetKey",
                "payment-cryptography:List*",
                "pca-connector-ad:GetConnector",
                "pca-connector-ad:GetDirectoryRegistration",
                "pca-connector-ad:GetServicePrincipalName",
                "pca-connector-ad:GetTemplate",
                "pca-connector-ad:GetTemplateGroupAccessControlEntry",
                "pca-connector-ad:List*",
                "pca-connector-scep:GetChallengeMetadata",
                "pca-connector-scep:GetConnector",
                "pca-connector-scep:List*",
                "personalize:Describe*",
```

```
                "personalize:List*",
                "pi:DescribeDimensionKeys",
                "pi:GetResourceMetadata",
                "pi:GetResourceMetrics",
                "pi:ListAvailableResourceDimensions",
                "pi:ListAvailableResourceMetrics",
                "pipes:Describe*",
                "pipes:List*",
                "proton:GetEnvironmentTemplate",
                "proton:GetServiceTemplate",
                "proton:List*",
                "qbusiness:GetApplication",
                "qbusiness:GetDataSource",
                "qbusiness:GetIndex",
                "qbusiness:GetPlugin",
                "qbusiness:GetRetriever",
                "qbusiness:GetWebExperience",
                "qbusiness:List*",
                "ram:GetPermission",
                "ram:GetResourceShares",
                "ram:List*",
                "rds:Describe*",
                "rds:List*",
                "redshift-serverless:GetNamespace",
                "redshift-serverless:GetWorkgroup",
                "redshift-serverless:List*",
                "redshift:Describe*",
                "refactor-spaces:GetApplication",
                "refactor-spaces:GetEnvironment",
                "refactor-spaces:GetRoute",
                "refactor-spaces:List*",
                "rekognition:Describe*",
                "rekognition:List*",
                "resiliencehub:Describe*",
                "resiliencehub:List*",
                "resource-explorer-2:GetDefaultView",
                "resource-explorer-2:GetIndex",
                "resource-explorer-2:GetView",
                "resource-explorer-2:List*",
                "resource-explorer-2:Search",
                "resource-groups:GetGroup",
                "resource-groups:GetGroupConfiguration",
                "resource-groups:GetGroupQuery",
                "resource-groups:GetTags",
```

```
                "resource-groups:List*",
                "route53-recovery-control-config:Describe*",
                "route53-recovery-control-config:List*",
                "route53-recovery-readiness:GetCell",
                "route53-recovery-readiness:GetReadinessCheck",
                "route53-recovery-readiness:GetRecoveryGroup",
                "route53-recovery-readiness:GetResourceSet",
                "route53-recovery-readiness:List*",
                "route53:GetDNSSEC",
                "route53:GetHealthCheck",
                "route53:GetHealthCheckStatus",
                "route53:GetHostedZone",
                "route53:List*",
                "route53profiles:GetProfile",
                "route53profiles:GetProfileAssociation",
                "route53profiles:GetProfileResourceAssociation",
                "route53profiles:List*",
                "route53resolver:GetFirewallDomainList",
                "route53resolver:GetFirewallRuleGroup",
                "route53resolver:GetFirewallRuleGroupAssociation",
                "route53resolver:GetOutpostResolver",
                "route53resolver:GetResolverConfig",
                "route53resolver:GetResolverQueryLogConfig",
                "route53resolver:GetResolverQueryLogConfigAssociation",
                "route53resolver:GetResolverRule",
                "route53resolver:GetResolverRuleAssociation",
                "route53resolver:List*",
                "rum:GetAppMonitor",
                "rum:List*",
                "s3-outposts:ListEndpoints",
                "s3-outposts:ListOutpostsWithS3",
                "s3:GetAccessGrant",
                "s3:GetAccessGrantsInstance",
                "s3:GetAccessGrantsLocation",
                "s3:GetAccessPoint",
                "s3:GetAccessPointConfigurationForObjectLambda",
                "s3:GetAccessPointForObjectLambda",
                "s3:GetAccessPointPolicy",
                "s3:GetAccessPointPolicyForObjectLambda",
                "s3:GetAccessPointPolicyStatusForObjectLambda",
                "s3:GetBucketAbac",
                "s3:GetBucketAcl",
                "s3:GetBucketCORS",
                "s3:GetBucketLocation",
```

```
                "s3:GetBucketLogging",
                "s3:GetBucketMetadataTableConfiguration",
                "s3:GetBucketNotification",
                "s3:GetBucketObjectLockConfiguration",
                "s3:GetBucketOwnershipControls",
                "s3:GetBucketPolicy",
                "s3:GetBucketPublicAccessBlock",
                "s3:GetBucketTagging",
                "s3:GetBucketVersioning",
                "s3:GetEncryptionConfiguration",
                "s3:GetLifecycleConfiguration",
                "s3:GetMultiRegionAccessPoint",
                "s3:GetMultiRegionAccessPointPolicy",
                "s3:GetMultiRegionAccessPointPolicyStatus",
                "s3:GetReplicationConfiguration",
                "s3:GetStorageLensConfiguration",
                "s3:GetStorageLensConfigurationTagging",
                "s3:GetStorageLensGroup",
                "s3:ListAllMyBuckets",
                "sagemaker:Describe*",
                "sagemaker:List*",
                "scheduler:GetSchedule",
                "scheduler:GetScheduleGroup",
                "scheduler:List*",
                "schemas:Describe*",
                "schemas:GetResourcePolicy",
                "schemas:List*",
                "secretsmanager:Describe*",
                "secretsmanager:GetResourcePolicy",
                "secretsmanager:List*",
                "securityhub:BatchGetAutomationRules",
                "securityhub:BatchGetSecurityControls",
                "securityhub:Describe*",
                "securityhub:GetConfigurationPolicy",
                "securityhub:GetConfigurationPolicyAssociation",
                "securityhub:GetEnabledStandards",
                "securityhub:GetFindingAggregator",
                "securityhub:GetInsights",
                "securityhub:List*",
                "securitylake:GetSubscriber",
                "securitylake:List*",
                "servicecatalog:Describe*",
                "servicecatalog:GetApplication",
                "servicecatalog:GetAttributeGroup",
```

```
                "servicecatalog:List*",
                "servicequotas:GetServiceQuota",
                "ses:Describe*",
                "ses:GetAccount",
                "ses:GetAddonInstance",
                "ses:GetAddonSubscription",
                "ses:GetArchive",
                "ses:GetConfigurationSet",
                "ses:GetConfigurationSetEventDestinations",
                "ses:GetContactList",
                "ses:GetDedicatedIpPool",
                "ses:GetDedicatedIps",
                "ses:GetEmailIdentity",
                "ses:GetEmailTemplate",
                "ses:GetIngressPoint",
                "ses:GetRelay",
                "ses:GetRuleSet",
                "ses:GetTemplate",
                "ses:GetTrafficPolicy",
                "ses:List*",
                "shield:Describe*",
                "shield:List*",
                "signer:GetSigningProfile",
                "signer:List*",
                "sns:GetDataProtectionPolicy",
                "sns:GetSubscriptionAttributes",
                "sns:GetTopicAttributes",
                "sns:List*",
                "sqs:GetQueueAttributes",
                "sqs:GetQueueUrl",
                "sqs:List*",
                "ssm-contacts:GetContact",
                "ssm-contacts:GetContactChannel",
                "ssm-contacts:List*",
                "ssm-incidents:GetReplicationSet",
                "ssm-incidents:GetResponsePlan",
                "ssm-incidents:List*",
                "ssm-sap:GetApplication",
                "ssm-sap:List*",
                "ssm:Describe*",
                "ssm:GetDefaultPatchBaseline",
                "ssm:GetDocument",
                "ssm:GetParameters",
                "ssm:GetPatchBaseline",
```

```
                "ssm:GetResourcePolicies",
                "ssm:List*",
                "sso:GetInlinePolicyForPermissionSet",
                "sso:GetManagedApplicationInstance",
                "sso:GetPermissionsBoundaryForPermissionSet",
                "sso:GetSharedSsoConfiguration",
                "sso:ListAccountAssignments",
                "sso:ListApplicationAssignments",
                "sso:ListApplications",
                "sso:ListCustomerManagedPolicyReferencesInPermissionSet",
                "sso:ListInstances",
                "sso:ListManagedPoliciesInPermissionSet",
                "sso:ListTagsForResource",
                "states:GetExecutionHistory",
                "states:Describe*",
                "states:List*",
                "support:CreateCase",
                "support:DescribeCases",
                "synthetics:Describe*",
                "synthetics:GetCanary",
                "synthetics:GetCanaryRuns",
                "synthetics:GetGroup",
                "synthetics:List*",
                "tag:GetResources",
                "timestream:Describe*",
                "timestream:List*",
                "transfer:Describe*",
                "transfer:List*",
                "verifiedpermissions:GetIdentitySource",
                "verifiedpermissions:GetPolicy",
                "verifiedpermissions:GetPolicyStore",
                "verifiedpermissions:GetPolicyTemplate",
                "verifiedpermissions:GetSchema",
                "verifiedpermissions:List*",
                "vpc-lattice:GetAccessLogSubscription",
                "vpc-lattice:GetAuthPolicy",
                "vpc-lattice:GetListener",
                "vpc-lattice:GetResourcePolicy",
                "vpc-lattice:GetRule",
                "vpc-lattice:GetService",
                "vpc-lattice:GetServiceNetwork",
                "vpc-lattice:GetServiceNetworkServiceAssociation",
                "vpc-lattice:GetServiceNetworkVpcAssociation",
                "vpc-lattice:GetTargetGroup",
```

```
                "vpc-lattice:List*",
                "wafv2:GetIPSet",
                "wafv2:GetLoggingConfiguration",
                "wafv2:GetRegexPatternSet",
                "wafv2:GetRuleGroup",
                "wafv2:GetWebACL",
                "wafv2:GetWebACLForResource",
                "wafv2:List*",
                "workspaces-web:GetBrowserSettings",
                "workspaces-web:GetIdentityProvider",
                "workspaces-web:GetNetworkSettings",
                "workspaces-web:GetPortal",
                "workspaces-web:GetPortalServiceProviderMetadata",
                "workspaces-web:GetTrustStore",
                "workspaces-web:GetUserAccessLoggingSettings",
                "workspaces-web:GetUserSettings",
                "workspaces-web:List*",
                "workspaces:Describe*",
                "xray:BatchGetTraces",
                "xray:GetGroup",
                "xray:GetGroups",
                "xray:GetSamplingRules",
                "xray:GetServiceGraph",
                "xray:GetTraceSummaries",
                "xray:List*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AIOPSAPIGatewayAccess",
            "Effect": "Allow",
            "Action": [
                "apigateway:GET"
            ],
            "Resource": [
                "arn:aws:apigateway:*::/restapis",
                "arn:aws:apigateway:*::/restapis/*",
                "arn:aws:apigateway:*::/restapis/*/deployments",
                "arn:aws:apigateway:*::/restapis/*/deployments/*",
                "arn:aws:apigateway:*::/restapis/*/resources/*/methods/*/integrations",
                "arn:aws:apigateway:*::/restapis/*/resources/*/methods/*/integrations/
*",
                "arn:aws:apigateway:*::/restapis/*/stages",
                "arn:aws:apigateway:*::/restapis/*/stages/*",
```

```
                "arn:aws:apigateway:*::/apis",
                "arn:aws:apigateway:*::/apis/*",
                "arn:aws:apigateway:*::/apis/*/deployments",
                "arn:aws:apigateway:*::/apis/*/deployments/*",
                "arn:aws:apigateway:*::/apis/*/integrations",
                "arn:aws:apigateway:*::/apis/*/integrations/*",
                "arn:aws:apigateway:*::/apis/*/stages",
                "arn:aws:apigateway:*::/apis/*/stages/*",
                "arn:aws:apigateway:*::/domainnames/*"
            ]
        }
    ]
}
```

AWS 계정의 에이전트 액세스 제한

AWS DevOps Agent는 IAM 역할을 사용하여 인시던트 조사 및 예방 평가 중에 AWS 리소스를 검색하고 설명합니다. 이러한 역할에 연결된 IAM 정책을 구성하여 에이전트의 액세스 수준을 제어할 수 있습니다. 애플리케이션 토폴로지에는 에이전트가 액세스할 수 있는 모든 것이 표시되지 않습니다. IAM 정책은 에이전트가 액세스할 수 있는 AWS 서비스 APIs 및 리소스를 진정으로 제한하는 유일한 방법입니다.

AWS DevOps 에이전트의 IAM 역할 이해

AWS DevOps Agent는 IAM 역할을 사용하여 두 가지 유형의 계정의 리소스에 액세스합니다.

- 기본 계정 역할 - 에이전트 스페이스를 생성하는 AWS 계정의 리소스에 대한 액세스 권한을 에이전트에게 부여합니다.
- 보조 계정 역할 - 에이전트 스페이스에 연결하는 추가 AWS 계정의 리소스에 대한 액세스 권한을 에이전트에게 부여합니다.

두 계정 유형 중 하나에 대해 에이전트가 액세스할 수 있는 AWS 서비스를 제한하고, 해당 서비스 내의 특정 리소스에 대한 액세스를 제한하고, 에이전트가 작동할 수 있는 리전을 제어할 수 있습니다.

리소스 경계 선택

리소스 액세스를 제한할 때는 에이전트가 애플리케이션 인시던트를 성공적으로 조사할 수 있는 충분한 권한을 포함해야 합니다. 여기에는 다음이 포함됩니다.

- 에이전트가 모니터링하고 조사해야 하는 범위 내 애플리케이션의 모든 리소스

- 해당 애플리케이션이 의존하는 모든 지원 인프라

지원 인프라에는 다음이 포함될 수 있습니다.

- 네트워킹 구성 요소(VPCs, 서브넷, 로드 밸런서, API 게이트웨이)
- 데이터 스토어(데이터베이스, 캐시, 객체 스토리지)
- 컴퓨팅 리소스(EC2 인스턴스, Lambda 함수, 컨테이너)
- 서비스 모니터링 및 로깅(CloudWatch, CloudTrail)
- 권한을 이해하는 데 필요한 자격 증명 및 액세스 관리 리소스

액세스를 너무 좁게 제한하면 에이전트가 정의된 경계 외부의 지원 인프라에서 시작된 근본 원인을 식별하지 못할 수 있습니다.

서비스 액세스 제한

에이전트의 역할에 연결된 IAM 정책을 수정하여 에이전트가 액세스할 수 있는 AWS 서비스를 제한할 수 있습니다. 사용자 지정 정책을 생성할 때 다음 모범 사례를 따르세요.

- 읽기 전용 권한만 부여 - 에이전트는 조사 중에 리소스 구성, 지표 및 로그를 읽어야 합니다. 에이전트가 리소스를 수정하거나 삭제할 수 있는 권한을 부여하지 마세요.
- 필요한 서비스로 제한 - 애플리케이션과 관련된 리소스가 포함된 AWS 서비스만 포함합니다. 예를 들어 애플리케이션이 Amazon RDS를 사용하지 않는 경우 정책에 RDS 권한을 포함하지 마십시오.
- 와일드카드 대신 특정 작업 사용 - `service:*` 권한을 부여하는 대신 `cloudwatch:GetMetricData` 또는와 같은 개별 작업을 지정합니다`ec2:DescribeInstances`.

특정 서비스로 제한하는 정책 예제:

```
json

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:GetMetricData",
        "cloudwatch:GetMetricStatistics",
```

```
        "cloudwatch:DescribeAlarms",
        "logs:GetLogEvents",
        "logs:FilterLogEvents",
        "ec2:DescribeInstances",
        "lambda:GetFunction",
        "lambda:GetFunctionConfiguration"
      ],
      "Resource": "*"
    }
  ]
}
```

리소스 액세스 제한

에이전트를 서비스 내의 특정 리소스로 제한하려면 IAM 정책에서 리소스 수준 권한을 사용합니다. 이를 통해 특정 패턴과 일치하는 리소스에만 액세스 권한을 부여할 수 있습니다.

리소스 ARN 패턴 사용:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lambda:GetFunction",
        "lambda:GetFunctionConfiguration"
      ],
      "Resource": "arn:aws:lambda:*:*:function:production-*"
    }
  ]
}
```

이 예제에서는 에이전트가 이름이 "production-"로 시작하는 Lambda 함수에만 액세스하도록 제한합니다.

태그 기반 제한 사용:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeInstanceStatus"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Environment": "production"
        }
      }
    }
  ]
}
```

이 예제에서는 에이전트가 태그가 지정된 EC2 인스턴스에만 액세스하도록 제한합니다`Environment=production`.

리전별 액세스 제한

에이전트가 액세스할 수 있는 AWS 리전을 제한하려면 IAM 정책에서 `aws:RequestedRegion` 조건 키를 사용합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:Describe*",
        "lambda:Get*",
        "cloudwatch:Get*"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestedRegion": [
            "us-east-1",
            "us-west-2"
          ]
        }
      }
    }
```

```
  ]
}
```

이 예제에서는 에이전트가 us-east-1 및 us-west-2 리전의 리소스에만 액세스하도록 제한합니다.

사용자 지정 IAM 정책 생성

에이전트 스페이스를 생성하거나 보조 계정을 추가할 때 정책 템플릿을 사용하여 사용자 지정 IAM 역할을 생성할 수 있습니다. 이를 통해 최소 권한 원칙을 구현할 수 있습니다.

에이전트 스페이스를 생성할 때

AWS 관리 콘솔의 DevOps 에이전트 콘솔에서...

- 정책 문서를 사용하여 새 DevOps 에이전트 역할 생성을 선택하고 지침을 따릅니다.

에이전트 스페이스를 편집할 때

AWS 관리 콘솔의 DevOps 에이전트 콘솔에서...

- 기능 탭 선택
- 클라우드 섹션에서 편집하려는 보조 계정을 선택하고 편집을 클릭합니다.
- 템플릿을 사용하여 새 DevOps 에이전트 정책 생성을 선택하고 지침을 따릅니다.

사용자 지정 정책 모범 사례

- 읽기 전용 권한만 부여 - 리소스 수정 또는 삭제를 허용하는 권한 피하기
- 가능한 경우 리소스 수준 권한 사용 - ARN 패턴 또는 태그를 사용하여 특정 리소스에 대한 액세스 제한
- 정기적으로 권한 검토 및 감사 - 에이전트의 IAM 정책을 정기적으로 검토하여 보안 요구 사항에 부합하는지 확인합니다.

IAM Identity Center 인증 설정

IAM Identity Center 인증은 AWS DevOps 에이전트 스페이스 웹 애플리케이션에 대한 사용자 액세스를 관리하는 중앙 집중식 방법을 제공합니다. 이 가이드에서는 IAM Identity Center 인증을 구성하고 사용자를 관리하는 방법을 설명합니다.

사전 조건

IAM Identity Center 인증을 설정하기 전에 다음을 확인해야 합니다.

- 조직 또는 계정에서 활성화된 IAM Identity Center
- in AWS DevOps 에이전트의 관리자 권한
- 구성되었거나 생성할 준비가 된 에이전트 공간

인증 옵션

AWS DevOps Agent는 에이전트 스페이스 웹 앱에 액세스하기 위한 두 가지 인증 방법을 제공합니다.

IAM Identity Center 인증 - 프로덕션 환경에 권장됩니다. 중앙 집중식 사용자 관리, 외부 ID 제공업체와의 통합 및 최대 12시간의 세션을 제공합니다.

관리자 액세스(IAM 인증) - 초기 설정 및 구성 중에 관리자에게 빠른 액세스를 제공합니다. 세션은 30분으로 제한됩니다.

에이전트 스페이스 생성 중 IAM Identity Center 구성

에이전트 스페이스를 생성할 때 액세스 탭에서 IAM Identity Center 인증을 구성할 수 있습니다.

1단계: 웹 앱 구성으로 이동

1. 에이전트 스페이스 세부 정보 및 AWS 계정 액세스를 구성한 후 액세스 탭으로 이동합니다.
2. 'IAM Identity Center 연결'과 '관리자 액세스'의 두 섹션이 표시됩니다.

2단계: IAM Identity Center 통합 구성

[에이전트 스페이스]를 IAM Identity Center에 연결 섹션에서:

1. IAM Identity Center 인스턴스 확인 - 콘솔에 웹 앱 사용자 액세스를 관리할 Identity Center 인스턴스가 표시됩니다(예: `ssoins-7223a9580931edbe`). 가장 가까운 IAM Identity Center 인스턴스가 자동으로 미리 채워집니다.
2. IAM Identity Center 애플리케이션 역할 이름 옵션 선택 - 다음 세 가지 옵션 중 하나를 선택합니다.

새 DevOps 에이전트 역할 자동 생성(권장):

- 시스템은 적절한 권한을 가진 새 서비스 역할을 자동으로 생성합니다.
- 가장 간단한 옵션이며 대부분의 사용 사례에서 작동합니다.

기존 역할 할당:

- 이미 생성한 기존 IAM 역할 사용
- 시스템은 역할에 필요한 권한이 있는지 확인합니다.
- 조직에 AWS DevOps Agent에 대한 역할이 미리 생성된 경우이 옵션을 선택합니다.

정책 템플릿을 사용하여 새 DevOps 에이전트 역할을 생성합니다.

- 제공된 정책 세부 정보를 사용하여 IAM 콘솔에서 사용자 지정 역할을 생성합니다.
- 역할 권한을 사용자 지정해야 하는 경우이 옵션을 선택합니다.

연결을 클릭하면 시스템이 자동으로 다음을 수행합니다.

- 지정된 IAM 역할을 생성하거나 구성합니다.
- 에이전트 스페이스에 대한 IAM Identity Center 애플리케이션 설정
- IAM Identity Center와 에이전트 스페이스 웹 앱 간의 신뢰 관계를 설정합니다.
- 보안 사용자 액세스를 위한 OAuth 2.0 인증 흐름 구성

대안: 관리자 액세스 사용

IAM Identity Center를 설정하지 않고 에이전트 스페이스 웹 앱에 즉시 액세스하려는 경우:

1. 관리자 액세스 섹션에서 관리자 액세스를 제공하는 IAM 역할 ARN을 기록해 둡니다 (예: `arn:aws:iam::440491339484:role/service-role/DevOpsAgentRole-WebappAdmin-15ppoc42`).
2. 파란색 관리자 액세스 버튼을 클릭하여 IAM 인증으로 Agent Space 웹 앱을 시작합니다.
3. 이 방법을 사용하는 세션은 30분으로 제한됩니다.

> **Note**
>
> 관리자 액세스는 초기 설정 및 구성을 위한 것입니다. 프로덕션 사용 및 지속적 작업을 위해 IAM Identity Center 인증을 구성합니다.

사용자 및 그룹 추가

IAM Identity Center 인증을 구성한 후 특정 사용자 및 그룹에 Agent Space 웹 앱에 대한 액세스 권한을 부여해야 합니다.

1단계: 사용자 관리 액세스

1. AWS DevOps 에이전트 콘솔에서 에이전트 스페이스를 선택합니다.
2. 액세스 탭으로 이동
3. 사용자 액세스에서 사용자 및 그룹 관리를 클릭합니다.

2단계: 사용자 또는 그룹 추가

1. 사용자 또는 그룹 추가를 선택합니다.
2. IAM Identity Center 디렉터리에서 사용자 또는 그룹 검색
3. 추가할 사용자 또는 그룹 옆의 확인란을 선택합니다.
4. 추가를 클릭하여 액세스 권한을 부여합니다.

이제 선택한 사용자가 IAM Identity Center 자격 증명을 사용하여 Agent Space 웹 앱에 액세스할 수 있습니다.

외부 자격 증명 공급자 작업

IAM Identity Center에서 외부 ID 제공업체(예: Okta, Microsoft Entra ID 또는 Ping Identity)를 사용하는 경우:

- 사용자 및 그룹은 외부 ID 제공업체에서 IAM Identity Center로 동기화됩니다.
- Agent Space 웹 앱에 사용자 및 그룹을 추가할 때 동기화된 디렉터리에서를 선택합니다.
- 사용자 속성 및 그룹 멤버십은 외부 자격 증명 공급자가 유지 관리합니다.
- ID 제공업체의 변경 사항은 동기화 후 IAM Identity Center에 자동으로 반영됩니다.

사용자가 Agent Space 웹 앱에 액세스하는 방법

에이전트 스페이스에 사용자를 추가한 후:

1. 에이전트 스페이스 웹 앱 URL을 승인된 사용자와 공유
2. 사용자가 URL로 이동하면 IAM Identity Center 로그인 페이지로 리디렉션됩니다.
3. 자격 증명을 입력한 후(구성된 경우 MFA 완료) 에이전트 스페이스 웹 앱으로 다시 리디렉션됩니다.
4. 세션은 기본적으로 8시간 동안 유효합니다(Identity Center 관리자가 구성 가능).

사용자 액세스 관리

언제든지 사용자 액세스를 업데이트할 수 있습니다.

사용자 또는 그룹 추가:

- 위에서 설명한 것과 동일한 단계에 따라 사용자 또는 그룹을 추가합니다.

액세스 제거:

1. 사용자 액세스 섹션에서 제거할 사용자 또는 그룹을 찾습니다.
2. 이름 옆에 있는 제거 버튼을 클릭합니다.
3. 제거 확인

제거된 사용자는 즉시 액세스 권한을 잃게 되지만 활성 세션은 만료될 때까지 계속될 수 있습니다.

세션 관리

에이전트 스페이스 웹 앱에 대한 IAM Identity Center 세션의 특성은 다음과 같습니다.

- 기본 세션 기간 - 8시간
- 세션 보안 - 향상된 보호를 위한 HTTP 전용 쿠키
- 다중 인증 - IAM Identity Center에 구성된 경우 지원됩니다.
- API 자격 증명 - API 호출에 대해 단기(15분) SigV4 자격 증명이 발급되고 자동으로 갱신됩니다.

세션 기간을 구성하려면:

1. IAM Identity Center 콘솔로 이동합니다.
2. 설정 > 인증으로 이동합니다.
3. 세션 기간에서 원하는 기간(1시간에서 12시간)을 구성합니다.
4. 변경 사항 저장을 선택합니다

Identity Center 연결 해제

1. 에이전트 스페이스의 콘솔에서 오른쪽 상단의 작업을 클릭하고 IAM Identity Center에서 연결 해제를 선택합니다.
2. 확인 대화 상자에서 확인

외부 ID 제공업체(IdP) 인증 설정

외부 ID 제공업체(IdP) 인증을 사용하면 조직에서 Okta 또는 Microsoft Entra ID와 같은 기존 OIDC 호환 ID 제공업체를 사용하여 AWS DevOps Agent Space 웹 애플리케이션에 대한 사용자 액세스를 관리할 수 있습니다. 사용자는 AWS IAM Identity Center 없이 IdP를 통해 직접 회사 자격 증명으로 로그인합니다.

사전 조건

외부 IdP 인증을 설정하기 전에 다음을 확인해야 합니다.

- OIDC 호환 자격 증명 공급자(Okta 또는 Microsoft Entra ID)
- ID 제공업체에 대한 관리자 액세스
- Access AWS DevOps Agent 콘솔에 대한 관리자 권한
- 구성되었거나 생성할 준비가 된 에이전트 공간

작동 방식

외부 IdP 인증을 구성하는 경우:

- 사용자는 에이전트 스페이스 웹 앱 URL로 이동합니다.
- ID 제공업체의 로그인 페이지로 리디렉션됩니다.
- 회사 자격 증명으로 인증한 후 웹 앱으로 다시 리디렉션됩니다.

- 웹 앱은 에이전트 스페이스로 범위가 지정된 수명이 짧은 AWS 자격 증명으로 인증 토큰을 교환합니다.

세션은 최대 8시간 동안 유효합니다. 자격 증명은 사용자가 다시 인증할 필요 없이 OIDC 새로 고침 토큰을 사용하여 자동으로 새로 고쳐집니다.

외부 IdP 인증 구성

1단계: 자격 증명 공급자에 애플리케이션 등록

자격 증명 공급자를 선택하고 해당 설정 지침을 따릅니다.

옵션 A: Okta

1. Okta 관리 콘솔에서 애플리케이션 > 애플리케이션으로 이동하여 앱 통합 생성을 선택합니다.
2. OIDC - OpenID Connect를 로그인 방법으로 선택하고 웹 애플리케이션을 애플리케이션 유형으로 선택합니다. 다음을 선택합니다.
3. 애플리케이션에 대한 설명 이름을 설정합니다(예: `AWS DevOps Agent`).
4. 권한 부여 유형에서 다음을 확인합니다.
   - 권한 부여 코드(기본값)
   - 토큰 새로 고침 - 세션 새로 고침에 필요합니다. 활성화하지 않으면 사용자가 세션을 유지할 수 없습니다.

> **Note**
>
> Okta는 기본적으로 새로 고침 토큰 권한 부여 유형을 활성화하지 않습니다. 명시적으로 활성화해야 합니다.

1. 로그인 리디렉션 URIs 기본값으로 그대로 둡니다. 에이전트 스페이스를 구성한 후 업데이트합니다.
2. 할당에서 액세스 권한이 있어야 하는 사용자 또는 그룹을 할당합니다.
3. 저장을 선택합니다.
4. 애플리케이션의 일반 탭에서 다음 값을 기록해 둡니다.
   - 클라이언트 ID
   - 클라이언트 보안 암호 - 복사를 선택하여이 값을 안전하게 저장

5. Okta 도메인 기록 - 발급자 URL입니다(예: `https://dev-12345678.okta.com`).

> **Note**
>
> 로그인 탭에서 발급자가 Okta URL(동적 아님)로 설정되어 있는지 확인합니다. 이렇게 하면 안정적인 발급자 URL이 보장됩니다.

> **Note**
>
> 권한 부여 서버의 클레임 탭에서 ID 토큰에 그룹 클레임을 추가하지 마십시오. AWS DevOps Agent는 IdP의 그룹 멤버십을 사용하지 않습니다.

옵션 B: Microsoft Entra ID

1. Azure 포털에서 Microsoft Entra ID > 앱 등록 > 새 등록으로 이동합니다.
2. 설명 이름 설정(예: `AWS DevOps Agent`)
3. 지원되는 계정 유형에서 조직에 적합한 옵션을 선택합니다(일반적으로 이 조직 디렉터리의 계정만 해당).
4. 지금은 리디렉션 URI를 비워 둡니다. 등록을 선택합니다.
5. 애플리케이션 개요 페이지에서 다음 값을 기록해 둡니다.
   - 애플리케이션(클라이언트) ID - 에이전트 공간을 구성할 때 클라이언트 ID로 사용됩니다.
   - 디렉터리(테넌트) ID - 발급자 URL을 구성하는 데 사용됩니다.
6. 인증서 및 보안 암호 > 새 클라이언트 보안 암호로 이동합니다.
   - 설명 및 만료 기간 설정
   - 보안 암호 값 추가 및 즉시 복사를 선택합니다. 다시 표시되지 않습니다.
7. Entra ID의 발급자 URL은이 형식을 따릅니다. 를 5단계의 디렉터리(테넌트) ID`{tenant-id}`로 바꿉니다.
   - `https://login.microsoftonline.com/{tenant-id}/v2.0`

> **Note**
>
> 토큰 구성 에서 그룹 선택적 클레임을 활성화하지 마십시오. AWS DevOps Agent는 IdP의 그룹 멤버십을 사용하지 않습니다.

2단계: IdP 인증으로 운영자 앱 활성화

1. AWS DevOps 에이전트 콘솔에서 에이전트 스페이스를 선택합니다.
2. 액세스 탭으로 이동
3. 사용자 액세스에서 외부 자격 증명 공급자를 선택합니다.
4. 구성 양식에서 다음을 구성합니다.
   - 자격 증명 공급자 - 자격 증명 공급자(Okta 또는 Microsoft Entra ID)를 선택합니다.
   - 발급자 URL - 자격 증명 공급자의 OIDC 발급자 URL
   - 클라이언트 ID - 생성한 OIDC 애플리케이션의 클라이언트 ID입니다.
   - 클라이언트 보안 암호 - OIDC 애플리케이션의 클라이언트 보안 암호
5. 자격 증명 공급자 애플리케이션 역할 이름에서 다음 세 가지 옵션 중 하나를 선택합니다.
   - 새 DevOps 에이전트 역할 자동 생성(권장) - 적절한 권한을 가진 새 서비스 역할을 생성합니다.
   - 기존 역할 할당 - 이미 생성한 기존 IAM 역할 사용
   - 정책 템플릿을 사용하여 새 DevOps 에이전트 역할 생성 - 제공된 세부 정보를 사용하여 IAM 콘솔에서 자체 역할을 생성합니다.
6. 양식 하단에 표시된 콜백 URL 경고 알림을 검토합니다. 이 URL 복사 - 사용자가 로그인하려면 먼저 자격 증명 공급자의 허용된 리디렉션 URIs에 추가해야 합니다.
7. 연결을 선택합니다.

연결을 선택하면 콘솔에 다음 세부 정보와 함께 외부 자격 증명 공급자 구성이 표시됩니다.

- 공급자 - 선택한 자격 증명 공급자입니다.
- 발급자 URL - 구성된 OIDC 발급자 URL
- 클라이언트 ID - 구성된 클라이언트 ID
- IAM 역할 ARN - 사용자 액세스에 사용되는 IAM 역할
- 콜백 URL - 자격 증명 공급자에서이 URL을 허용된 리디렉션 URI로 구성합니다.
- 로그인 URL -이 URL을 사용하여 자격 증명 공급자를 통해 웹 앱에 액세스합니다.

3단계: 자격 증명 공급자에 콜백 URL 추가

Okta

1. Okta 관리 콘솔에서 애플리케이션의 일반 탭으로 이동합니다.
2. 로그인에서 편집을 선택합니다.
3. 콜백 URL을 로그인 리디렉션 URI로 추가합니다.
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/callback`
4. (선택 사항) Okta 대시보드에서 IdP 시작 로그인을 활성화하려면 로그인 URI 시작을 설정합니다.
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/login`
5. (권장) 로그아웃 후 사용자를 웹 앱으로 다시 리디렉션하는 로그아웃 리디렉션 URI를 추가합니다. 이렇게 하지 않으면 로그아웃 시 사용자에게 오류 페이지가 표시될 수 있습니다.
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome`
6. 저장을 선택합니다.

Microsoft Entra ID

1. Azure 포털에서 애플리케이션의 인증 페이지로 이동합니다.
2. 플랫폼 구성에서 플랫폼 추가 > 웹을 선택합니다.
3. 콜백 URL을 리디렉션 URI로 입력합니다.
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/callback`
4. (선택 사항) 로그아웃 후 사용자를 웹 앱으로 다시 리디렉션하는 로그아웃 리디렉션 URI를 추가합니다.
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome`
5. 구성을 선택합니다.

4단계: 구성 확인

1. 콘솔에 표시된 로그인 URL로 이동합니다.
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/login`
2. ID 제공업체의 로그인 페이지로 리디렉션되어야 합니다.
3. 회사 자격 증명으로 로그인
4. 인증에 성공하면 에이전트 스페이스 웹 앱으로 다시 리디렉션됩니다.

IdP 구성 업데이트

연결을 해제하지 않고도 클라이언트 보안 암호를 교체할 수 있습니다.

1. AWS DevOps 에이전트 콘솔에서 에이전트 스페이스를 선택합니다.
2. 액세스 탭으로 이동
3. 외부 자격 증명 공급자 구성에서 클라이언트 보안 암호 교체를 선택합니다.
4. 새 클라이언트 보안 암호 입력
5. 저장을 선택합니다.

다른 IdP 구성 필드(예: 발급자 URL, 클라이언트 ID 또는 자격 증명 공급자)를 변경하려면 기존 IdP의 연결을 해제하고 새 IdP를 구성해야 합니다.

사용자가 Agent Space 웹 앱에 액세스하는 방법

외부 IdP 인증을 구성한 후:

- 에이전트 스페이스 웹 앱 URL을 승인된 사용자와 공유
- 사용자가 URL로 이동하면 ID 제공업체의 로그인 페이지로 리디렉션됩니다.
- 자격 증명을 입력한 후(IdP에서 구성한 경우 MFA 완료) 에이전트 스페이스 웹 앱으로 다시 리디렉션됩니다.
- 세션 자동 새로 고침 - 자세한 내용은 세션 관리를 참조하세요.

세션 관리

에이전트 스페이스 웹 앱에 대한 외부 IdP 세션의 특성은 다음과 같습니다.

- 세션 기간 - 브라우저 세션은 최대 8시간 동안 지속됩니다. 이는 in AWS DevOps 에이전트에서 구성할 수 없습니다. IdP의 세션 수명이 8시간을 초과하는 경우 사용자는 자격 증명을 입력하지 않고 다음 방문 시 자동으로 재인증될 수 있습니다. 조직의 보안 요구 사항에 따라 IdP의 세션 및 토큰 수명을 구성합니다.
- 자격 증명 새로 고침 - 세션은 사용자가 다시 인증할 필요 없이 OIDC 새로 고침 토큰을 사용하여 자동으로 새로 고쳐집니다.
- 다중 인증 - 자격 증명 공급자에 구성된 경우 지원됩니다. IdP는 로그인 중에 MFA를 처리하므로 AWS DevOps 에이전트에서 추가 구성이 필요하지 않습니다.

로그아웃 동작

사용자가 웹 앱에서 로그아웃을 클릭하는 경우:

1. 모든 세션 쿠키가 즉시 지워집니다.
2. 사용자가 자격 증명 공급자의 OIDC 로그아웃 엔드포인트로 리디렉션되어 SSO 세션을 종료합니다.
3. 로그아웃 리디렉션 URI가 구성된 경우 사용자는 웹 앱 시작 페이지로 다시 리디렉션됩니다.

사용자 액세스 취소

사용자의 액세스를 즉시 취소하려면 자격 증명 공급자의 관리자 포털에서 직접 세션을 취소할 수 있습니다.

- Okta - Okta 관리 콘솔에서 디렉터리 > 사람으로 이동하여 사용자를 선택하고 추가 작업 > 사용자 세션 지우기를 선택합니다.
- Microsoft Entra ID - Azure 포털에서 사용자로 이동하여 사용자를 선택한 다음 세션 취소를 선택합니다.

보안 고려 사항

클라이언트 보안 암호 스토리지 - 설정 중에 제공하는 클라이언트 보안 암호는 에이전트 스페이스를 생성할 때 제공한 경우 고객 관리형 KMS 키를 사용하여 암호화되고, 그렇지 않으면 서비스 소유 키를 사용하여 암호화됩니다. 초기 구성 후에는 API 응답으로 반환되거나 콘솔에 표시되지 않습니다.

클라이언트 보안 암호 교체 - Entra 클라이언트 보안 암호에는 구성 가능한 만료가 있습니다. AWS DevOps 에이전트 콘솔의 클라이언트 보안 암호 교체 옵션을 사용하여 보안 암호가 만료되기 전에 교체하라는 알림을 설정합니다. 보안 암호가 만료되면 교체될 때까지 사용자가 로그인할 수 없습니다.

토큰 수명 관리 - ID 제공업체가 발급한 토큰(액세스 토큰, 새로 고침 토큰)의 수명은 IdP의 구성에 의해 제어됩니다. IdP에서 적절한 토큰 수명을 구성하는 것이 좋습니다.

- Okta - 보안 > API > 권한 부여 서버 > 액세스 정책에서 토큰 수명 구성
- Microsoft Entra ID - 토큰 수명 정책을 사용하여 토큰 수명 구성

그룹 클레임 - ID 제공업체의 토큰 구성에서 그룹 클레임을 활성화하지 마십시오. AWS DevOps Agent는 현재 IdP의 그룹 멤버십을 사용하지 않습니다.

사용자 식별자 - AWS DevOps Agent는 공급자별 클레임을 사용하여 사용자를 고유하게 식별합니다.

- Okta - ID 토큰의 `sub` 클레임을 사용합니다.
- Microsoft Entra ID - ID 토큰의 `oid` (객체 식별자) 클레임을 사용합니다.

이러한 식별자는 변경할 수 없으며 감사 목적으로 CloudTrail 로그에 표시됩니다.

외부 IdP 연결 해제

1. AWS DevOps 에이전트 콘솔에서 에이전트 스페이스를 선택합니다.
2. 액세스 탭으로 이동
3. 사용자 액세스에서 연결 해제를 선택합니다.
4. 확인 대화 상자에 나열된 영향을 검토하고 확인합니다.

연결 해제는 다음을 수행합니다.

- 에이전트 공간에서 IdP 구성 제거
- 사용자가 외부 자격 증명 공급자를 통해 로그인하지 못하도록 방지
- IdP 사용자 계정과 연결된 개별 채팅 및 아티팩트 기록 제거

활성 사용자 세션은 만료되거나 다음 자격 증명 새로 고침이 실패할 때까지 계속됩니다.

문제 해결

- IdP로 리디렉션 실패 - 발급자 URL이 IdP의 OIDC 검색 엔드포인트와 일치하는지 확인합니다. Okta의 경우 로그인 탭에서 발급자가 Okta URL(동적 아님)로 설정되어 있는지 확인합니다. Entra의 경우 형식을 사용합니다`https://login.microsoftonline.com/{tenant-id}/v2.0`.
- 액세스 거부 또는 정책 오류(Okta) - 할당에서 사용자 또는 해당 그룹이 애플리케이션에 할당되었는지 확인합니다. 로그인 > 로그인 정책 규칙을 선택합니다.
- 로그인 후 IdP 구성 오류 - ID 제공업체가 새로 고침 토큰을 반환하지 않았습니다. `offline_access` 범위 및 새로 고침 토큰 권한 부여 유형이 활성화되어 있는지 확인합니다.
  - Okta - 애플리케이션의 일반 탭으로 이동하여 권한 부여 유형에서 토큰 새로 고침 확인란을 활성화합니다.
  - Entra - API 권한으로 이동하여 `offline_access`가 위임된 권한 아래에 나열되어 있는지 확인합니다.

- 인증에 성공했지만 웹 앱에 오류가 표시됨 - IdP의 리디렉션 URI가 AWS DevOps 에이전트 콘솔에 표시된 콜백 URL과 정확히 일치하는지 확인합니다.
- 인증 실패 - IdP에서 그룹 선택적 클레임이 활성화된 경우 비활성화합니다. AWS DevOps Agent는 그룹 클레임을 사용하지 않습니다.
- IdP 인증 후 로그인 실패 - Entra의 경우 애플리케이션 매니페스트`null`에서 `requestedAccessTokenVersion`가 로 설정되지 않았는지 확인합니다. Okta의 경우 발급자 URL이 올바른지 확인합니다.
- 로그아웃(Okta) 클릭 후 오류 페이지 - 로그아웃 후 `post_logout_redirect_uri` 오류가 표시되면 Okta 애플리케이션의 일반 탭에 를 로그아웃 리디렉션 URI`https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome`로 추가합니다.
- 사용자는 로그아웃 후 ID 제공업체 페이지에 남아 있음(Entra) - 로그아웃 후 사용자를 웹 앱으로 다시 리디렉션하려면 Entra 애플리케이션의 인증 페이지에 를 리디렉션 URI`https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome`로 추가합니다.

AWS DevOps 에이전트의 저장 데이터 암호화

AWS DevOps Agent는 저장된 모든 고객 데이터를 암호화합니다. 기본적으로 AWS DevOps Agent는 AWS 소유 키를 사용하여 추가 비용 없이 데이터를 자동으로 암호화합니다. AWS 소유 키의 사용을 보거나 관리하거나 감사할 수 없습니다. 그러나 이러한 키를 보호하기 위해 조치를 취할 필요는 없습니다. 데이터는 자동으로 보호됩니다.

AWS Key Management Service(AWS KMS)에서 생성, 소유 및 관리하는 대칭 고객 관리형 키를 사용하여 데이터를 암호화하도록 선택할 수 있습니다. 이 암호화 계층을 완전히 제어할 수 있으므로 다음과 같은 작업을 수행할 수 있습니다.

- 키 정책 수립 및 유지
- 키 정책 활성화 및 비활성화
- 키 암호화 자료 교체
- 태그 추가
- 키 별칭 만들기
- 삭제를 위한 스케줄 키

자세한 내용은 Key Management Service 개발자 안내서의 고객 관리형 키를 참조하세요. AWS

> **Note**
>
> AWS DevOps Agent는 AWS 소유 키를 사용하여 저장 데이터 암호화를 자동으로 활성화하여 고객 데이터를 무료로 보호합니다. 고객 관리형 키를 사용하는 경우 표준 AWS KMS 요금이 적용됩니다. 요금에 대한 자세한 내용은 AWS Key Management Service 요금을 참조하세요.

고객 관리형 키

고객 관리형 키는 사용자가 생성, 소유 및 관리하는 AWS 계정의 KMS 키입니다. 키 정책 설정 및 유지 관리를 포함하여 이러한 KMS 키를 완전히 제어할 수 있습니다.

고객 관리형 키를 구성할 때 AWS DevOps Agent는 이를 사용하여 민감한 리소스 데이터를 보호합니다. AWS DevOps Agent는 AWS Encryption SDK 계층적 키링을 사용한 봉투 암호화를 사용합니다. KMS 키는 브랜치 키를 생성하는 데 사용되며, 브랜치 키는 데이터를 보호합니다.

다음 리소스를 생성할 때 고객 관리형 키를 지정할 수 있습니다.

- 에이전트 스페이스 - 조사, 기술 및 채팅과 관련하여 DevOps 에이전트 웹 앱에서 생성된 에이전트 스페이스 세부 정보 및 콘텐츠를 암호화합니다.
- 서비스 - 저장된 타사 서비스 자격 증명을 암호화합니다.

AWS DevOps Agent에서 고객 관리형 키를 구성하려면 다음 단계를 따릅니다.

1단계: 고객 관리형 키 만들기

KMS 콘솔 또는 AWS AWS KMS API를 사용하여 대칭 고객 관리형 키를 생성할 수 있습니다. 키는 다음 요구 사항을 충족해야 합니다.

속성	요구 사항
키 유형	대칭
키 사양	`SYMMETRIC_DEFAULT`
키 사용	`ENCRYPT_DECRYPT`

> **Note**
>
> AWS DevOps Agent는 키 사양 및 ENCRYPT_DECRYPT 키 사용량이 있는 대칭 암호화 KMS SYMMETRIC_DEFAULT 키만 지원합니다. 다중 리전 키와 비대칭 키는 현재 지원되지 않습니다.

자세한 내용은 AWS Key Management Service 개발자 안내서의 대칭 고객 관리형 키 생성을 참조하세요.

2단계: 키 정책 설정

키 정책에서는 고객 관리형 키에 대한 액세스를 제어합니다. 모든 고객 관리형 키에는 키를 사용할 수 있는 사람과 키를 사용하는 방법을 결정하는 문장이 포함된 정확히 하나의 키 정책이 있어야 합니다.

키 정책은 호출 보안 주체(IAM 자격 증명)와 AWS DevOps 에이전트 서비스 모두에 권한을 부여해야 합니다. AWS DevOps 에이전트는 두 가지 자격 증명 세트를 사용하여 키에 액세스합니다.

1. 호출자 자격 증명 - 키 검증, 리소스 생성 시 암호화, 호출자에게 직접 응답을 반환하는 API 호출 등 모든 동기 작업에 사용됩니다.
2. AWS DevOps Agent 서비스 보안 주체 - 운영 조사, 인시던트 분석, 이벤트 상관관계 및 근본 원인 분석 생성과 같이 백그라운드에서 실행되는 비동기 작업에 사용됩니다.

다음 표에는 필요한 KMS 작업이 나열되어 있습니다.

KMS 작업	설명
kms:DescribeKey	리소스 생성 시 키 구성 검증
kms:GenerateDataKey	봉투 암호화를 위한 데이터 암호화 키 생성
kms:Decrypt	데이터 해독
kms:Encrypt	Encrypt data
kms:ReEncrypt	동일하거나 다른 키로 데이터 재암호화

AWS DevOps Agent는 구성 시 모의 실행 작업을 사용하여 이러한 모든 권한을 검증합니다. 권한이 누락된 경우 예외와 함께 요청이 실패합니다.

다음은 예제 키 정책입니다. 자리 표시자 값을 자신의 값으로 바꿉니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCallerAccessViaService",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/DevOpsAgentUserRole"
      },
      "Action": [
        "kms:DescribeKey",
        "kms:GenerateDataKey*",
        "kms:Decrypt",
        "kms:Encrypt",
        "kms:ReEncrypt*"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "aidevops.us-east-1.amazonaws.com"
        }
      }
    },
    {
      "Sid": "AllowDevOpsAgentServiceDescribeKeyAccess",
      "Effect": "Allow",
      "Principal": {
        "Service": "aidevops.amazonaws.com"
      },
      "Action": [
        "kms:DescribeKey"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowDevOpsAgentAccessForAgentSpace",
      "Effect": "Allow",
      "Principal": {
        "Service": "aidevops.amazonaws.com"
```

```
      },
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt",
        "kms:Encrypt",
        "kms:ReEncrypt*"
      ],
      "Resource": "*",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:aidevops:us-east-1:111122223333:agentspace/*"
        },
        "StringLike": {
          "kms:EncryptionContext:aws-crypto-ec:aws:aidevops:arn": "arn:aws:aidevops:us-
east-1:111122223333:agentspace/*"
        }
      }
    },
    {
      "Sid": "AllowDevOpsAgentAccessForService",
      "Effect": "Allow",
      "Principal": {
        "Service": "aidevops.amazonaws.com"
      },
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt",
        "kms:Encrypt",
        "kms:ReEncrypt*"
      ],
      "Resource": "*",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:aidevops:us-east-1:111122223333:service/*"
        },
        "StringLike": {
          "kms:EncryptionContext:aws-crypto-ec:aws:aidevops:arn": "arn:aws:aidevops:us-
east-1:111122223333:service/*"
        }
      }
    }
  ]
}
```

정책에는 다음 문이 포함되어 있습니다.

- AllowKeyAdministration - 계정 루트에 키에 대한 전체 관리 액세스 권한을 부여합니다. 를 AWS 계정 ID111122223333로 바꿉니다.
- AllowCallerAccessViaService - IAM 보안 주체에게 모든 synchronous AWS DevOps 에이전트 작업에 필요한 KMS 권한을 부여합니다. 여기에는 리소스 생성 시점의 키 검증과 호출자에게 직접 응답을 반환하는 모든 API 호출에 대한 암호화 및 복호화 작업이 포함됩니다. `kms:ViaService` 조건은 AWS DevOps 에이전트 서비스를 통해서만 키를 사용할 수 있도록 합니다. `111122223333`를 AWS 계정 ID로 바꾸고를 AWS 리전`us-east-1`으로 바꿉니다.
- AllowDevOpsAgentServiceAccessForAgentSpace / AllowDevOpsAgentServiceAccessForService - `aidevops.amazonaws.com` 서비스 보안 주체에게 비동기 작업에 필요한 KMS 권한을 부여합니다. AWS DevOps Agent는 운영 조사, 인시던트 분석, 서비스 간 이벤트 상호 연결, 근본 원인 분석 생성과 같은 백그라운드 작업을 수행할 때이 서비스 보안 주체를 사용하여 데이터를 암호화하고 해독합니다. 이 액세스 권한이 없으면 AWS DevOps Agent는 사용자를 대신하여 조사를 수행하는 데 필요한 암호화된 데이터를 읽을 수 없습니다. `aws:SourceArn` 조건은 your AWS DevOps 에이전트 리소스에서 시작된 요청에 대한 액세스를 제한하며, `kms:EncryptionContext` 조건은 암호화 컨텍스트가 리소스 ARNs과 일치하도록 합니다. `111122223333`를 AWS 계정 ID로 바꾸고를 AWS 리전`us-east-1`으로 바꿉니다.

키 정책에 대한 자세한 내용은 Key Management Service 개발자 안내서의 AWS KMS의 키 정책을 참조하세요. AWS

3단계: 리소스를 생성할 때 키 지정

키를 생성하고 키 정책을 구성한 후 AWS DevOps 에이전트 리소스를 생성할 때 키를 지정할 수 있습니다.

콘솔

콘솔에서 에이전트 스페이스를 생성할 때 고객 관리형 키를 구성하려면:

1. AWS DevOps 에이전트 콘솔을 엽니다.
2. 에이전트 스페이스 생성 또는 서비스 등록을 선택합니다.
3. 에이전트 공간 세부 정보(이름, 설명 및 IAM 역할)를 입력합니다.
4. 고급 구성 섹션을 확장합니다.
5. 암호화 키 유형에서 고객 관리형 키를 선택합니다.

6. 드롭다운 목록에서 KMS 키를 선택하거나 KMS 키 ARN을 입력합니다.
7. 키 정책 확장 가능 섹션에 표시된 키 정책을 검토합니다. 이 정책을 KMS 키에 연결했는지 확인합니다. 복사 버튼을 사용하여 정책을 복사할 수 있습니다.
8. 나머지 구성을 완료하고 생성을 선택합니다.

> **Note**
>
> 드롭다운 목록에 KMS 키가 표시되지 않으면 키가 1단계의 요구 사항을 충족하고 및 `kms:ListKeys` `kms:DescribeKey` 권한이 있는지 확인합니다.

API

고객 관리형 키를 사용하여 에이전트 공간 생성

에이전트 공간을 생성할 때 `kmsKeyArn` 파라미터를 지정합니다. 값은 전체 KMS 키 ARN이어야 합니다.

```
{
  "name": "my-agent-space",
  "description": "An encrypted agent space",
  "kmsKeyArn": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

고객 관리형 키를 사용하여 서비스 등록

서비스를 등록할 때 `kmsKeyArn` 파라미터를 지정합니다. 값은 전체 KMS 키 ARN이어야 합니다. 이 파라미터는 Dynatrace, ServiceNow, PagerDuty, GitLab, GitHub 및 MCP 서버를 포함한 모든 서비스 유형에서 지원됩니다.

```
{
  "service": "dynatrace",
  "kmsKeyArn": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "serviceDetails": { ... }
}
```

> **Note**
>
> 리소스 생성 시 고객 관리형 키를 지정해야 합니다. 기존 리소스에 대한 고객 관리형 키는 추가하거나 변경할 수 없습니다.

AWS DevOps 에이전트 암호화 컨텍스트

암호화 컨텍스트는 데이터에 대한 추가 컨텍스트 정보가 포함된 비밀이 아닌 키-값 페어 세트입니다. AWS KMS는 암호화 컨텍스트를 추가 인증된 데이터로 사용하여 인증된 암호화를 지원합니다. 데이터 암호화 요청에 암호화 컨텍스트를 포함하면 AWS KMS는 암호화 컨텍스트를 암호화된 데이터에 바인딩합니다. 데이터를 복호화하려면 요청에 동일한 암호화 컨텍스트를 포함해야 합니다.

AWS DevOps Agent는 모든 암호화 작업에 다음 암호화 컨텍스트를 사용합니다.

```
{
  "aws-crypto-ec:aws:aidevops:arn": "arn:aws:aidevops:{region}:{accountId}:
{resourceType}/{resourceId}"
}
```

암호화 컨텍스트 값은 암호화되는 AWS DevOps 에이전트 리소스의 ARN입니다. 키 정책 조건 및 in AWS CloudTrail 로그에서이 암호화 컨텍스트를 사용하여 키가 사용되는 방식을 감사할 수 있습니다.

키 관리

KMS 키 삭제를 비활성화하거나 예약하면 AWS DevOps Agent가 데이터를 해독할 수 없습니다. 이로 인해 암호화된 데이터를 읽는 작업에 `AccessDeniedException` 오류가 발생합니다.

> **Important**
>
> 고객 관리형 키를 사용하기로 선택한 경우 키와 해당 권한을 관리할 책임은 사용자에게 있습니다. 키가 비활성화되거나 삭제되거나 AWS DevOps Agent가 키 사용 권한을 상실하면 암호화된 데이터에 대한 액세스 권한이 상실됩니다.

다음 표에서는 일반적인 장애 시나리오를 설명합니다.

작업	영향
키 정책 권한이 취소됨	`AccessDeniedException` 암호화 및 복호화 작업
KMS 키가 비활성화됨	`DisabledException` 암호화 및 복호화 작업
KMS 키 삭제 예정	`KMSInvalidStateException` 암호화 및 복호화 작업
KMS 키가 삭제됨	영구 데이터 손실 - 암호화된 데이터는 복구할 수 없습니다.

키를 비활성화하거나 삭제하기 전에:

1. 키에 의존하는 active AWS DevOps 에이전트 리소스가 없는지 확인합니다.
2. 삭제를 예약하기 전에 먼저 키를 비활성화하여 영향을 테스트하는 것이 좋습니다.
3. AWS KMS는 키 삭제 전에 최소 대기 기간을 적용하므로 필요한 경우 취소할 수 있습니다.

> 참고:: AWS DevOps Agent는 새 키로 데이터를 자동으로 다시 암호화하지 않습니다. 새 고객 관리형 키로 교체해야 하는 경우 새 키를 사용하여 새 리소스를 생성해야 합니다.

암호화 키 모니터링

AWS DevOps 에이전트와 함께 고객 관리형 키를 사용하는 경우 AWS CloudTrail을 사용하여 AWS DevOps 에이전트가 AWS KMS로 보내는 요청을 추적할 수 있습니다.

다음을 기준으로 CloudTrail 이벤트를 필터링할 수 있습니다.

- 이벤트 소스 - `kms.amazonaws.com`
- 암호화 컨텍스트 키 - `aws-crypto-ec:aws:aidevops:arn`
- 키 ARN - 요청 파라미터의 고객 관리형 키 ARN

자세한 내용은 AWS Key Management Service 개발자 안내서의 AWS CloudTrail을 사용하여 AWS KMS API 호출 로깅을 참조하세요.

VPC 엔드포인트(AWS PrivateLink)

AWS PrivateLink를 사용하여 VPC와 AWS DevOps 에이전트 간에 프라이빗 연결을 생성할 수 있습니다. 인터넷 게이트웨이, NAT 디바이스, VPN 연결 또는 Direct Connect 연결을 사용하지 않고 VPC에 있는 것처럼 AWS DevOps 에이전트에 액세스할 수 있습니다. VPC의 인스턴스는 AWS DevOps 에이전트에 액세스하는 데 퍼블릭 IP 주소가 필요하지 않습니다.

이 프라이빗 연결은 AWS PrivateLink로 구동되는 인터페이스 엔드포인트를 생성하여 설정합니다. 인터페이스 엔드포인트에 대해 사용 설정하는 각 서브넷에서 엔드포인트 네트워크 인터페이스를 생성합니다. 이는 AWS DevOps 에이전트로 향하는 트래픽의 진입점 역할을 하는 요청자 관리형 네트워크 인터페이스입니다.

자세한 내용은 PrivateLink 가이드의 Access AWS services AWS PrivateLink throughPrivateLink를 참조하세요. AWS PrivateLink

AWS DevOps Agent VPC 엔드포인트에 대한 고려 사항

AWS DevOps Agent에 대한 인터페이스 엔드포인트를 설정하기 전에 AWS PrivateLink 가이드의 고려 사항을 검토하세요.

AWS DevOps Agent는 다음 VPC 엔드포인트를 통한 API 호출을 지원합니다.

카테고리	엔드포인트 접미사
AWS DevOps 에이전트 제어 플레인 API 작업	`aidevops`
AWS DevOps 에이전트 런타임 작업	`aidevops-dataplane`
AWS DevOps 에이전트 웹후크 이벤트	`event-ai`

AWS DevOps Agent용 인터페이스 엔드포인트 생성

Amazon VPC 콘솔 또는 AWS 명령줄 인터페이스(AWS CLI)를 사용하여 AWS DevOps 에이전트용 인터페이스 엔드포인트를 생성할 수 있습니다. 자세한 내용은 AWS PrivateLink 가이드의 인터페이스 엔드포인트 생성을 참조하세요.

다음 서비스 이름을 사용하여 AWS DevOps Agent용 인터페이스 엔드포인트를 생성합니다.

- com.amazonaws.{region}.aidevops

- com.amazonaws.{region}.aidevops-dataplane
- com.amazonaws.{region}.event-ai

엔드포인트를 만든 다음 프라이빗 DNS 호스트 이름을 활성화할 수 있습니다. VPC 엔드포인트를 생성할 때 VPC 콘솔에서 프라이빗 DNS 이름 활성화(Enable Private DNS Name)를 선택하여 이 설정을 활성화합니다.

인터페이스 엔드포인트에 대해 프라이빗 DNS를 활성화하면 기본 리전 DNS 이름을 사용하여 AWS DevOps 에이전트에 API 요청을 할 수 있습니다. 다음 예제에서는 기본 리전 DNS 이름의 형식을 보여 줍니다.

- aidevops.{region}.api.aws
- aidevops-dataplane.{region}.amazonaws.com
- event-ai.{region}.api.aws

엔드포인트의 엔드포인트 정책 생성

엔드포인트 정책은 인터페이스 엔드포인트에 연결할 수 있는 IAM 리소스입니다. 기본 엔드포인트 정책은 인터페이스 엔드포인트를 통해 AWS DevOps Agent에 대한 전체 액세스를 허용합니다. VPC에서 AWS DevOps Agent에 허용되는 액세스를 제어하려면 인터페이스 엔드포인트에 사용자 지정 엔드포인트 정책을 연결합니다.

엔드포인트 정책은 다음 정보를 지정합니다.

- 작업을 수행할 수 있는 보안 주체(AWS 계정, IAM 사용자 및 IAM 역할).
- 수행할 수 있는 작업
- 작업을 수행할 수 있는 리소스.

자세한 내용은 AWS PrivateLink 가이드의 엔드포인트 정책을 사용하여 서비스에 대한 액세스 제어를 참조하세요.

할당량

AWS DevOps 에이전트 할당량에는 에이전트 공간 수, 동시 조사 등이 포함됩니다. 일부 할당량의 경우 증가를 요청할 수 있지만, 모든 할당량을 늘릴 수 있지는 않습니다. 이러한 인상은 즉시 부여되지 않으므로 인상이 적용되려면 몇 시간에서 며칠이 걸릴 수 있습니다. 다르게 표시되지 않는 한, 리전별로 각 할당량이 적용됩니다.

다음 표에서는 AWS DevOps 에이전트의 할당량을 설명합니다.

이름	기본값	조정 가능	설명
리전별 계정당 에이전트 공간	10	예	각 AWS 리전에서 계정당 생성할 수 있는 최대 에이전트 공간 수입니다.
에이전트 공간당 동시 조사	3	예	단일 에이전트 공간에서 동시에 실행할 수 있는 최대 인시던트 해결 조사 수입니다.
에이전트 공간당 동시 평가	1	아니요	단일 에이전트 공간에서 동시에 실행할 수 있는 최대 인시던트 방지 평가 수입니다.
에이전트 공간당 동시 온디맨드 호출	10	예	단일 에이전트 공간에서 동시에 실행할 수 있는 온디맨드 DevOps 호출의 최대 수입니다.

할당량 증가 요청

다음 옵션 중 하나를 사용하여 할당량 증가를 요청할 수 있습니다.

- AWS 관리 콘솔에서 - Service Quotas 콘솔을 엽니다. 탐색 창에서 AWS 서비스를 선택합니다. DevOps 에이전트를 선택하고 할당량을 선택한 다음 지침에 따라 할당량 증가를 요청합니다. 자세한 내용은 Service Quotas 사용 설명서의 할당량 증가 요청을 참조하세요.
- AWS CLI에서 - request-service-quota-increase AWS CLI 명령을 사용합니다. 자세한 내용은 Service Quotas 사용 설명서의 할당량 증가 요청을 참조하십시오.

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