Benutzer-Leitfaden

AWS DevOps Agentin

AWS DevOps Agentin: Benutzer-Leitfaden

Table of Contents

Informationen AWS DevOps Agentin

AWS DevOps Der Agent ist ein Grenzagent, der Vorfälle löst und proaktiv verhindert und so die Zuverlässigkeit und Leistung kontinuierlich verbessert.

AWS DevOps Als erfahrener Techniker untersucht der Kundendienstmitarbeiter Vorfälle und identifiziert betriebliche Verbesserungen. DevOps

Der Agent arbeitet wie folgt:

- Lernen Sie Ihre Ressourcen und ihre Beziehungen kennen.
- Arbeiten Sie mit Ihren Observability-Tools, Fähigkeiten, Code-Repositorys und CI/CD Pipelines.
- Korrelieren von Telemetrie-, Code- und Bereitstellungsdaten, um die Beziehungen zwischen Ihren Anwendungsressourcen zu verstehen.
- Unterstützung von Anwendungen in Multicloud- und Hybridumgebungen.

Schlüssel-Features

AWS DevOps Agent bietet umfassende Funktionen zur Reaktion auf Vorfälle und zur Vorbeugung von Vorfällen mithilfe der folgenden Funktionen:

Always-on, autonome Reaktion auf Vorfälle

AWS DevOps Der Agent untersucht selbstständig Probleme, sobald sie auftreten:

- Automatisierte Untersuchung von Vorfällen — Beginnt sofort mit der Untersuchung, wenn eine Warnung oder ein Support-Ticket eingeht
- AWS DevOps Agenten-Chat — Fragen Sie Ihre Infrastruktur ab, analysieren Sie den Systemstatus und leiten Sie Untersuchungen in natürlicher Sprache in der gesamten DevOps Agent Space-Web-App. Der Chat bietet kontextsensitive Antworten, die auf der aufgerufenen Seite basieren. Dabei spielt es keine Rolle, ob Sie nach Ressourcen in der Topologie fragen, eine Untersuchung steuern oder Empfehlungen im Bereich Prävention filtern.
- Detaillierte Pläne zur Schadensbegrenzung — Bietet spezifische Maßnahmen zur Behebung von Vorfällen, zur Überprüfung des Erfolgs und zur Rücknahme von Änderungen, falls erforderlich

- Automatisierte Koordination von Vorfällen — Leitet Beobachtungen, Ergebnisse und Maßnahmen zur Schadensbegrenzung über deine bevorzugten Kommunikationskanäle wie Slack und ServiceNow
- AWS Support-Integration — Erstellen Sie AWS Support-Fälle direkt aus einer Untersuchung heraus, wobei den AWS Support-Experten sofort der Kontext zur Verfügung gestellt wird

future Vorfälle verhindern

AWS DevOps Der Agent analysiert Muster bei allen Vorfällen in der Vergangenheit, um Sie dabei zu unterstützen, von der reaktiven Brandbekämpfung zur proaktiven Betriebsverbesserung überzugehen:

- Gezielte Empfehlungen — Bietet spezifische, umsetzbare Verbesserungen, die vier Schlüsselbereiche stärken: Beobachtbarkeit (Überwachung, Warnung, Protokollierung), Infrastrukturoptimierung (automatische Skalierung, Kapazitätsoptimierung) und Verbesserung der Bereitstellungspipeline (Testen, Validierung).
- Kontinuierliches Lernen — Verfeinert die Empfehlungen auf der Grundlage des Feedbacks Ihres Teams

Holen Sie mehr aus Ihren Tools heraus DevOps

AWS DevOps Agent lässt sich in Ihre vorhandenen Tools integrieren, ohne Ihre Workflows zu ändern:

- Zuordnung von Anwendungsressourcen — Erstellt ein Topologiediagramm Ihrer Anwendungsressourcen und ihrer Beziehungen
- Built-in Integrationen — Funktioniert mit gängigen Observability-Tools (Amazon CloudWatch, Dynatrace, Datadog, New Relic und Splunk), Code-Repositorys und Pipelines (Aktionen und Repositorys, Workflows und CI/CD Repositorys) GitHub GitLab
- Integration benutzerdefinierter Tools — Erweitern Sie die Funktionen, indem Sie eine Verbindung zu Ihren eigenen Model Context Protocol (MCP) -Servern herstellen, um zusätzliche Tools zu erhalten
- Abfragen zur Konversationsinfrastruktur — Verwenden Sie natürliche Sprache, um AWS Ressourcen, Systemmetriken und den Alarmstatus abzufragen, ohne durch mehrere Konsolen navigieren zu müssen. Chat versteht den Kontext und speichert den Konversationsverlauf für Folgefragen.

Wie AWS DevOps Der Agent funktioniert

AWS DevOps Der Agent arbeitet mit einer Architektur mit zwei Konsolen. Administratoren verwenden die AWS Managementkonsole, um Agent Spaces zu erstellen und zu verwalten, Integrationen zu konfigurieren und Zugriffskontrollen einzurichten. Betriebsteams verwenden die AWS DevOps Agent-Web-App für die tägliche Reaktion auf Vorfälle und die Untersuchung von Vorfällen. In der Web-App können Bediener mit den Ermittlungen der Agenten interagieren, die kontenübergreifende Anwendungstopologie durchsuchen und sich über präventive Verbesserungen der Beobachtbarkeit, des Codes, der Pipelines und der Infrastrukturarchitekturen informieren. Weitere Informationen hierzu finden Sie unter the section called “Proaktive Prävention von Zwischenfällen”.

Der Service ist in Agent Spaces organisiert. Dabei handelt es sich um logische Container, die definieren, worauf der Agent zugreifen und was AWS DevOps er untersuchen kann. Jeder Agent Space enthält Ihre AWS Kontokonfigurationen, Tool-Integrationen von Drittanbietern und Zugriffsberechtigungen. Weitere Informationen hierzu finden Sie unter the section called “Was sind DevOps Agent Spaces?”.

AWS DevOps Der Agent erstellt automatisch eine Anwendungstopologie, die Ihre Ressourcen und deren Beziehungen abbildet. Diese Topologie hilft dem Service dabei, Ihre Anwendungsarchitektur bei Untersuchungen zu verstehen. Weitere Informationen hierzu finden Sie unter the section called “Was ist eine DevOps Agententopologie?”.

Vorteile

- Reduzieren Sie die mittlere Zeit bis zur Problemlösung (MTTR) — Die autonome Untersuchung beginnt sofort, wodurch die Problembehebung von Stunden auf Minuten beschleunigt wird
- Vermeiden Sie wiederkehrende Vorfälle — Gezielte Empfehlungen befassen sich mit den Grundursachen und stärken die Widerstandsfähigkeit des Systems
- Verbessern Sie die betriebliche Effizienz — Befreien Sie Ihr Team von sich wiederholenden Ermittlungsaufgaben, um sich auf Innovationen zu konzentrieren
- Arbeiten Sie innerhalb vorhandener Workflows — Lässt sich unterbrechungsfrei in Ihre vorhandenen Tools und Prozesse integrieren

Was ist eine DevOps Agent-Web-App?

AWS DevOps Der Agent verwendet eine Architektur mit zwei Konsolen, die Verwaltungsfunktionen von day-to-day betrieblichen Aktivitäten trennt. Dieses Design ermöglicht es Administratoren, den Service zu konfigurieren, während sich die Betriebsteams auf die Reaktion und Prävention von Vorfällen konzentrieren.

Konsolen

AWS DevOps Der Agent bietet zwei unterschiedliche Benutzeroberflächen:

- AWS Management-Konsole — Administratoren verwenden die AWS Management-Konsole, um den AWS DevOps Agenten einzurichten und zu verwalten. In dieser Konsole können Sie AWS Dienste und Tools von Drittanbietern the section called “Einen Agentenbereich erstellen” verbinden und die Zugriffsberechtigungen für Ihr Unternehmen verwalten.
- DevOps Agenten-Web-App — Betriebsteams verwenden DevOps Agent Space-Web-Apps für die tägliche Reaktion auf Vorfälle. Diese eigenständige Anwendung bietet eine Oberfläche, über die Techniker auf Abruf Untersuchungen einleiten, über einen Chat in natürlicher Sprache mit dem Agenten interagieren, Anwendungstopologien einsehen und Empfehlungen zur Vorfallprävention überprüfen können.

Funktionen der Web-App

Die DevOps Agent-Web-App bietet die folgenden Hauptfunktionen:

- Reaktion auf Vorfälle — Auf dieser Seite können Sie Untersuchungen zu Vorfällen erstellen und nachverfolgen sowie Pläne zur Schadensbegrenzung erstellen, um Vorfälle zu lösen.
- Prävention von Vorfällen — Auf der Seite Prävention finden Sie Empfehlungen zur Verbesserung Ihrer Beobachtbarkeit, Ihrer Bereitstellungsprozesse und Ihrer Infrastrukturarchitektur, um future Vorfälle zu verhindern.
- Topologie — Die Topologie-Seite bietet eine interaktive visuelle Darstellung der Kontoressourcen und ihrer Beziehungen zwischen allen Ressourcen in den verbundenen Konten. Sie können die Topologie mit unterschiedlichen Detaillierungsgraden anzeigen, indem Sie das Dropdownmenü „Anzeigen“ verwenden, um zwischen der System-, Container- und Ressourcenansicht zu wechseln.
- Fähigkeiten — Modulare Befehlssätze, die AWS DevOps Agent um spezielle Funktionen erweitern. Die Fähigkeiten umfassen Fachwissen, Untersuchungsmethoden und auf Ihre Infrastruktur zugeschnittene Toolkonfigurationen. Jede Fähigkeit ermöglicht spezifische Tools und ermöglicht

die schrittweise Offenlegung von Anweisungen nur dann, wenn sie für die Untersuchung relevant sind.

- Chat-Oberfläche in natürlicher Sprache — Chat ist in der gesamten Web-App verfügbar und ist ein KI-gestützter Konversationsassistent, mit dem Sie Ihre Infrastruktur abfragen, den Systemzustand analysieren und Untersuchungen in natürlicher Sprache bearbeiten können. Chat bietet kontextsensitive Antworten, die auf der von Ihnen aufgerufenen Seite basieren.

Authentifizierung

AWS DevOps Der Agent unterstützt flexible Authentifizierungsmethoden, um unterschiedlichen Unternehmensanforderungen gerecht zu werden:

- IAM Identity Center-Integration (Benutzerzugriff) — Organizations können AWS Identity Center (IAM Identity Center) verwenden, um den Benutzerzugriff auf die DevOps Agent Space-Web-Apps zentral zu verwalten. IAM Identity Center kann sich über Standard-OIDC- und SAML-Protokolle mit externen Identitätsanbietern verbinden, darunter Anbieter wie Okta, Ping Identity und Microsoft Entra ID. Diese Methode unterstützt die Multi-Faktor-Authentifizierung durch Ihren Identitätsanbieter.
- Authentifizierung durch externen Identitätsanbieter (IdP) — Organizations können einen OIDC-kompatiblen Identitätsanbieter wie Okta oder Microsoft Entra ID direkt mit der Agent Space-Web-App verbinden, ohne IAM Identity Center zu benötigen. Benutzer melden sich mit ihren Unternehmensanmeldedaten über den IdP an. Anweisungen zur Einrichtung finden Sie unterthe section called “Authentifizierung durch externen Identitätsanbieter (IdP) einrichten”.
- IAM-Authentifizierungslink (Administratorzugriff) — Eine alternative Methode ermöglicht den direkten Zugriff auf die Web-App von der AWS Management Console aus über Ihre bestehende Konsolensitzung. Diese Option ist vor der Implementierung der vollständigen Identity Center-Integration nützlich, Sitzungen sind jedoch auf 10 Minuten begrenzt.

Was sind DevOps Agent Spaces?

Ein DevOps Agent Space ist ein logischer Container, der die Tools und die Infrastruktur definiert, auf die der AWS DevOps Agent Zugriff hat. Jeder Agent Space arbeitet unabhängig und verfügt über einen eigenen AWS Kontozugriff, Integrationen von Drittanbietern und Benutzerberechtigungen.

Ein Agentenbereich stellt die Grenze dar, auf die der AWS DevOps Agent während der Reaktion auf einen Vorfall zugreifen und ihn untersuchen kann. Wenn Sie einen Agent Space erstellen, definieren

Sie, auf welche AWS Konten der Agent zugreifen kann, mit welchen externen Tools er sich verbinden kann und welche Benutzer in Ihrem Unternehmen mit dem Agenten interagieren können.

Jeder Agent Space fungiert als unabhängige AWS DevOps Agent-Bereitstellung. Sie konfigurieren den Agent Space über die AWS Management Console, während Ihre Betriebsteams die Agent Space-Web-App verwenden, um in diesem Bereich Untersuchungen durchzuführen und Empfehlungen zu überprüfen.

Wie werden Agent Spaces isoliert

Agent Spaces bleiben isoliert, um die Sicherheit zu gewährleisten und unbeabsichtigten Zugriff über verschiedene Umgebungen oder Teams hinweg zu verhindern:

- AWS Kontoisolierung — Jeder Agent Space verwendet spezielle IAM-Rollen, die nur Zugriff auf bestimmte AWS Konten und Ressourcen gewähren. Der Agent kann nicht auf AWS Ressourcen zugreifen, die nicht explizit für den Agent Space konfiguriert wurden.
- Isolierung des Benutzerzugriffs — Sie steuern, welche Benutzer oder Gruppen auf jeden Agent Space zugreifen können. Auf diese Weise können Sie die Zugriffsberechtigungen an Ihre Organisationsstruktur anpassen und sicherstellen, dass Teams nur mit den ihnen zugewiesenen Agent Spaces interagieren.
- Datenisolierung — Ermittlungsdaten, Vorfallverlauf und Empfehlungen werden in jedem Agentenbereich separat verwaltet. Informationen aus einem Agent Space sind von einem anderen Agent Space aus nicht sichtbar oder zugänglich.
- Isolierung von Chat-Daten — Der Verlauf der Chat-Konversationen ist ebenfalls innerhalb der einzelnen Agentenbereiche isoliert. Konversationen und Abfragen in einem Agentenbereich sind von einem anderen Agentenbereich aus nicht sichtbar oder zugänglich.

Agent Space-Webanwendung

Jeder Agent Space verfügt über eine eigene Web-App, auf die außerhalb der AWS Management Console zugegriffen werden kann. Weitere Informationen the section called “Was ist eine DevOps Agent-Web-App?” zur Web-App finden Sie unter.

Wann sollten mehrere Agent Spaces verwendet werden

Erwägen Sie die Einrichtung mehrerer Agent Spaces, um unterschiedliche organisatorische Anforderungen zu erfüllen:

- Trennung von Teams — Richten Sie spezielle Agent Spaces für verschiedene Anwendungsteams oder Geschäftsbereiche ein, um klare Eigentumsgrenzen im Agent Space aufrechtzuerhalten.
- Isolierung der Umgebung — Trennen Sie Produktions- und Nichtproduktionsumgebungen in verschiedene Agent Spaces, um einen versehentlichen Zugriff zwischen Umgebungen zu verhindern.
- Dienstgrenzen — Richten Sie Agent Spaces an die Grenzen bestimmter Dienste oder Anwendungen aus, damit die Untersuchungen zielgerichtet und relevant bleiben.
- Compliance-Anforderungen — Konfigurieren Sie separate Agent Spaces mit unterschiedlichen Zugriffskontrollen oder Einstellungen für die Datenresidenz, um gesetzliche Anforderungen zu erfüllen.

> **Note**
>
> Wenn Sie mehrere Agent Spaces erstellen, können Sie ein spezielles AWS Konto als primäres Konto für einen Agent Space verwenden und verschiedene Anwendungskonten als sekundäre Konten verbinden. Mit diesem Ansatz können Sie detaillierte Zugriffskontrollen aufrechterhalten und gleichzeitig sicherstellen, dass jeder Agent Space nur auf die Ressourcen zugreifen kann, die für seinen vorgesehenen Bereich spezifisch sind, auch wenn Sie die automatische Rollenerstellung verwenden.

Was ist eine DevOps Agententopologie?

AWS DevOps Der Agent erkennt und visualisiert automatisch die Ressourcen und Beziehungen innerhalb Ihrer Anwendungen und verwendet die daraus resultierende Topologie, um Ihre Infrastruktur bei der Untersuchung von Vorfällen und bei der Erteilung präventiver Empfehlungen besser zu verstehen.

Wie werden Topologiediagramme erstellt

AWS DevOps Der Agent erstellt Topologiediagramme mithilfe mehrerer automatisierter Prozesse:

- Ressourcenerkennung — Der Agent scannt Ihre AWS Konten automatisch, um Ressourcen wie Recheninstanzen, Speicherdienste, Netzwerkkomponenten und Datenbanken zu identifizieren, die Teil Ihrer Anwendungen sind.

- Erkennung von Beziehungen — Der Agent analysiert Konfigurationsdaten, CloudFormation Stacks und Ressourcen-Tags, um festzustellen, wie Ressourcen zueinander in Beziehung stehen.
- Code- und Bereitstellungszuordnung — Wenn der Agent mit CI/CD Pipelines verbunden ist, verknüpft er Infrastrukturressourcen wieder mit ihren Bereitstellungsprozessen und dem geänderten Anwendungs- und Infrastrukturcode.
- Abbildung des Beobachtbarkeitsverhaltens — Daten aus Observability-Systemen wie Amazon CloudWatch Application Signals und Dynatrace werden verwendet, um beobachtete Verhaltensweisen zu identifizieren, die auf Beziehungen zwischen Ressourcen hinweisen.

Die wichtigsten Funktionen

Resource Mapping bietet mehrere Funktionen, die die Untersuchung und Prävention von Vorfällen verbessern:

- Interaktive Visualisierung — Erkunden Sie Ihre Anwendungstopologie anhand eines interaktiven Diagramms in der Operator Web App. Sie können in der Topologie zoomen und darin navigieren, um komplexe Beziehungen zwischen Ressourcen zu verstehen. Sie können Chat auch verwenden, um Topologieinformationen in natürlicher Sprache abzufragen, z. B. „Zeige mir alle Lambda-Funktionen, die mit dieser DynamoDB-Tabelle verbunden sind“ oder „Welche Ressourcen sind von diesem Alarm betroffen?“.
- Kontextuelle Untersuchung — Bei der Untersuchung von Vorfällen wird der AWS DevOps Agent von der Ressourcentopologie unterstützt, um die betroffenen Komponenten zu identifizieren, den Explosionsradius zu verstehen und den Einschlagspfad durch Ihre Systeme zu verfolgen.
- Ursachenanalyse — Das detaillierte Verständnis der Ressourcenbeziehungen hilft dabei, genau zu bestimmen, wo Probleme ihren Ursprung haben, selbst in komplexen verteilten Systemen mit vielen wechselseitigen Abhängigkeiten.
- Folgenabschätzung — Bei der Analyse von Vorfällen kann der Mitarbeiter anhand von Abhängigkeitsketten in der Topologie besser bestimmen, welche nachgelagerten Dienste betroffen sein könnten.
- Präventive Empfehlungen — Der Agent verwendet topologische Erkenntnisse, um gezielte Empfehlungen zur Verbesserung der Ausfallsicherheit abzugeben und Änderungen vorzuschlagen, die sich am stärksten auf die Systemstabilität auswirken werden.

Topologie-Ansichten

Die Topologievisualisierung auf der Topologie-Seite in der Operator Web App bietet mehrere Detaillierungsebenen:

- Gelernt — Die Standardansicht, die anhand des Skills Agent Space Understanding generiert wurde. Zeigt eine strukturierte Zusammenfassung Ihrer Infrastruktur an, die nach logischen Diensten und Anforderungspfaden geordnet ist.
- System — Zeigt Konten- und Regionsgrenzen auf hoher Ebene an.
- Container — Zeigt Deployment-Stacks wie CloudFormation Stacks an, die zugehörige Ressourcen enthalten.
- Komponenten — Zeigt einzelne Komponenten innerhalb von Containern und ihre Beziehungen.
- Alle Ressourcen — Zeigt die vollständige Ansicht mit allen erkannten Ressourcen und ihren Beziehungen an.

Entdeckung von Ressourcen

Ressourcen werden auf zwei Arten entdeckt:

- CloudFormation Stapel — Der Agent listet alle CloudFormation Stacks und ihre Ressourcen im primären AWS Konto und allen verbundenen sekundären Konten auf. Dies wird für alle infrastructure-as-code Tools unterstützt, die CloudFormation für die Bereitstellung verwendet werden, einschließlich AWS Cloud Development Kit (AWS CDK).
- Resource Explorer — Für Ressourcen, die nicht von bereitgestellt wurden CloudFormation, werden markierte Ressourcen im AWS Resource Explorer erkannt. Für das AWS Zielkonto muss Resource Explorer aktiviert sein. Dies ist nützlich, um Anwendungsgrenzen für Ressourcen zu identifizieren, die über die AWS Management Console, den AWS Dienst APIs oder andere infrastructure-as-code Frameworks bereitgestellt werden.

Der Umfang der Untersuchung geht über die Topologie hinaus

Die Anwendungstopologie bietet zwar wichtigen Kontext bei Untersuchungen, aber AWS DevOps Agent ist nicht darauf beschränkt, nur die in der Topologie angegebenen Ressourcen zu untersuchen. Der Agent kann zusätzliche Datenquellen wie AWS Dienste APIs oder verbundene Observability-Tools verwenden, um Ressourcen zu untersuchen, die sich nicht in der Anwendungstopologie befinden.

Um die Ressourcen einzuschränken, auf die der Agent Zugriff hat, beschränken Sie die Richtlinie für die Rolle, die dem Agenten zugewiesen ist, auf kontenübergreifende Ressourcen. Weitere Informationen finden Sie unter the section called “Beschränken des Agentenzugriffs in einem AWS Account”.

Die Fähigkeit „Topologie“ und „Agent Space Understanding“

Das Topologiediagramm fließt in die erlernte Fähigkeit „Agent Space Understanding“ ein, die eine strukturierte Zusammenfassung Ihrer Infrastruktur zur Verwendung bei Untersuchungen kodiert. Wenn die Topologieermittlung für einen neuen Agentenbereich abgeschlossen ist, generiert das System automatisch den Skill Agent Space Understanding. Weitere Informationen zu den erlernten Fähigkeiten finden Sie unterthe section called “Erlernte Fähigkeiten”.

DevOps Fähigkeiten der Agenten

AWS DevOps Agent Skills sind modulare Befehlssätze, die die Fähigkeiten des Agenten um spezialisiertes Fachwissen und Untersuchungsmethoden erweitern, die auf Ihre Infrastruktur und Ihre betrieblichen Arbeitsabläufe zugeschnitten sind.

Was sind Fähigkeiten

Skills sind eigenständige Verzeichnisse, die Markdown-Anweisungen enthalten, die dem Agenten spezielle Funktionen zur AWS DevOps Verfügung stellen. AWS DevOps Der Agent unterstützt einen Teil der Agent Skills-Spezifikation — einen offenen Standard für die Paketierung von Anweisungen und Ressourcen für Agenten — und unterstützt nur Dokumente, die nicht ausführbar sind: Markdown-Anweisungen, PDFs, Bilder und Datendateien.

Für jeden Skill ist eine SKILL.md Datei mit Anweisungen erforderlich, die Sie Ihrem Agenten zur Verfügung stellen möchten. AWS DevOps Zusätzlich zu der erforderlichen SKILL.md Datei können Skills Folgendes beinhalten:

- Untersuchungsabläufe für bestimmte Szenarien oder Infrastrukturtypen.
- Referenzmaterialien, einschließlich Architekturmustern und Betriebsverfahren.
- Ausrichtung auf Agententypen — Die Fähigkeiten können auf bestimmte Agententypen (generisch, Incident Triage On-demand, Incident RCA, Incident Mitigation, Evaluation) zugeschnitten werden, um den Kontextverbrauch zu reduzieren und den Fokus der Mitarbeiter zu verbessern.

Warum Skills einsetzen

Skills machen AWS DevOps Agent von einem Allzweck-Assistenten zu einem Spezialisten für Ihre Infrastruktur und Ihre betrieblichen Arbeitsabläufe. Im Gegensatz zu einmaligen Anweisungen in einer Chat-Nachricht sind Skills wiederverwendbare Funktionen, die automatisch geladen werden, wenn sie für die vom AWS DevOps Agenten ausgeführten Aufgaben relevant sind.

Wichtigste Vorteile:

- Spezialisieren Sie Ihren Agenten — Tailor AWS DevOps Agent verfügt über Ermittlungsverfahren, bewährte Verfahren und organisatorisches Wissen, das speziell auf Ihre Infrastruktur und Betriebsmuster zugeschnitten ist.
- Vermeiden Sie Wiederholungen — Erstellen Sie Ermittlungsworkflows einmal und der AWS DevOps Agent verwendet sie automatisch für alle relevanten Untersuchungen, sodass Sie nicht immer wieder dieselben Anleitungen geben müssen.
- Funktionen zusammenstellen — Kombinieren Sie mehrere Fähigkeiten, um durchgängige Ermittlungsabläufe zu erstellen. AWS DevOps Der Agent liest während der Ausführung mehrere Fähigkeiten, z. B. einen Skill zum Abrufen von Deployments aus Ihrer benutzerdefinierten CI/CD Pipeline und einen Skill zum Durchsuchen Ihrer Code-Repositorys.
- Amplify Tools — Entwickeln Sie Fähigkeiten, die Ihren AWS DevOps Agenten helfen, Ihre benutzerdefinierten MCP-Servertools effektiv zu nutzen. Fähigkeiten können dokumentieren, wann bestimmte Tools aufgerufen werden müssen, welche Parameter für verschiedene Szenarien verwendet werden müssen und wie die Ergebnisse interpretiert werden müssen, um Workflows zu erreichen, die für Ihre Infrastruktur spezifisch sind.

Wie Fähigkeiten funktionieren

Wenn der AWS DevOps Agent auf eine relevante Aufgabe stößt, lädt er die entsprechenden Fähigkeiten ein und folgt den Anweisungen, die ihm bei der Untersuchung zur Verfügung stehen. Ein Skill „Untersuchung der Datenbankleistung“ könnte beispielsweise schrittweise Verfahren zur Analyse von Problemen mit der RDS-Drosselung beinhalten, sodass der Agent den Alarmstatus systematisch überprüfen, Verbindungsmetriken analysieren und langsame Abfragen identifizieren kann.

Struktur der Fähigkeiten

Ein Skill ist als Verzeichnis organisiert, das Folgendes enthält:

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

SKILL.md

Das `SKILL.md` ist die einzige obligatorische Datei. Sie enthält die Kernanweisungen, die im Markdown-Format geschrieben sind. Diese Datei sollte:

- Beschreiben Sie, wann und wie der Skill eingesetzt werden soll.
- Stellen Sie schrittweise Ermittlungsverfahren bereit.
- Fügen Sie Entscheidungsbäume für verschiedene Szenarien hinzu.
- Dokumentieren Sie die erwarteten Ergebnisse und Erfolgskriterien.

Titelsache

Frontmatter ist der Metadatenblock am Anfang einer `SKILL.md` Datei, der zwischen `---` Trennzeichen eingeschlossen ist. Es enthält die `description` Felder `name` und, anhand derer der AWS DevOps Agent bestimmt, wann der Skill während einer Untersuchung oder Aufgabe aktiviert werden soll.

```
---
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 — Eine eindeutige Kennung für den Skill. Verwenden Sie nur Kleinbuchstaben, Zahlen und Bindestriche (maximal 64 Zeichen). Darf nicht mit einem Bindestrich beginnen oder enden.

Beschreibung — Eine ausführliche Erklärung, wann und warum der AWS DevOps Agent diesen Skill einsetzen sollte. AWS DevOps Der Agent bewertet dieses Feld, um zu entscheiden, ob der Skill für die aktuelle Aufgabe relevant ist. Eine vage oder fehlende Beschreibung kann dazu führen, dass der Agent den Skill komplett überspringt, selbst wenn die Anweisungen gut geschrieben sind.

Wichtig — Schreiben Sie die Beschreibung aus der Sicht des Agenten. Geben Sie die spezifischen Szenarien, Dienste, Fehlertypen oder Symptome an, die den Skill auslösen sollten. Beispielsweise ist „Verwenden Sie diesen Skill bei der Untersuchung von Datenbanklatenz, Verbindungsfehlern oder Abfrage-Timeouts für Amazon RDS-Instances“ effektiver als „RDS-Skill“.

Wenn Sie einen Skill in der Benutzeroberfläche erstellen, generiert das System anhand des von Ihnen angegebenen Namens und der Beschreibung automatisch eine Titelseite. Skills, die als ZIP-Dateien hochgeladen werden, müssen Frontmaterial in der `SKILL.md` Datei enthalten.

Beispiel: Fertigkeit abschließen

Das folgende Beispiel zeigt eine umfassende, wohlgeformte Fähigkeit zur Untersuchung von RDS-Leistungsproblemen. Es zeigt die Verzeichnisstruktur, den SKILL.md Hauptteil, umsetzbare Ermittlungsverfahren und eine ergänzende Referenzdatei.

Verzeichnisstruktur:

```
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
```

references/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% |
```

Beispiel: Fähigkeit zur Filterung von Vorfällen

Fähigkeiten, die auf den Agententyp Incident Triage zugeschnitten sind, können Kriterien für das automatische Überspringen von Vorfällen definieren. Verwenden Sie diese Option, um Vorfälle zu filtern, die keiner Untersuchung bedürfen. Wenn ein neuer Vorfall den Übersprungskriterien entspricht, markiert der AWS DevOps Agent ihn als Übersprungen. Das System gibt einen Grund an, warum es gefiltert wurde.

Das folgende Beispiel zeigt eine Fähigkeit, mit der Vorfälle mit niedriger Priorität bei geplanten Wartungsarbeiten übersprungen werden können:

SKILL.md:

```
---
name: skip-scheduled-maintenance
description: Skip low-priority incidents during a scheduled maintenance window.
  Use this skill to automatically filter MEDIUM and LOW severity alarms that
  fire during planned maintenance, avoiding unnecessary investigations for
  expected disruptions.
---

# Skip Scheduled Maintenance

Skip all incidents that meet BOTH of the following criteria:

1. The incident arrived between **2025-03-15 02:00 UTC** and **2025-03-15 06:00 UTC**
2. Severity is MEDIUM or LOW

Do NOT skip HIGH or CRITICAL severity incidents, even during the maintenance window.
```

Wenn Sie diesen Skill erstellen, wählen Sie Incident Triage als Agententyp aus. Dadurch wird sichergestellt, dass der Skill nur während der Triage-Phase bewertet wird.

Fähigkeiten schaffen

Bevor Sie Skills erstellen können, müssen Sie über einen Agent Space verfügen. Weitere Informationen finden Sie unter the section called “Einen Agentenbereich erstellen”.

Je nach Ihren Workflow-Einstellungen und der Komplexität Ihrer Fähigkeiten können Sie Skills auf zwei Arten erstellen:

Einen Skill in der Benutzeroberfläche erstellen

In der AWS DevOps Agent Operator Web App erstellte Fähigkeiten enthalten einen Namen, eine Beschreibung und Anweisungen in einer einzigen SKILL.md Datei.

So erstellen Sie einen Skill in der Benutzeroberfläche:

- Navigieren Sie in Ihrer Agent Space Operator-Web-App zur Seite Skills.
- Klicken Sie auf „Skill hinzufügen“.
- Wählen Sie im Modal „Skill erstellen“ aus.
- Füllen Sie das Skill-Formular aus:
  - Name — Nur Kleinbuchstaben, Zahlen und Bindestriche (maximal 64 Zeichen). Darf nicht mit einem Bindestrich beginnen oder enden. Beispiel: `rds-throttling-investigation`
  - Beschreibung — Kurze Erklärung, wann dieser Skill eingesetzt werden sollte (mindestens 100 Zeichen empfohlen, maximal 1.024 Zeichen). Auf diese Weise kann der Agent bestimmen, wann der Skill aktiviert werden muss.
  - Status — Auf Aktiv (Standard) oder Inaktiv gesetzt. Inaktive Skills werden vom Agenten nicht verwendet.
  - Agententyp — Wählen Sie einen oder mehrere Agententypen aus, die diesen Skill verwenden können. Generisch ist standardmäßig ausgewählt und macht den Skill für alle Agententypen verfügbar. Um bestimmte Agenten gezielt anzusprechen, deaktivieren Sie Generisch und wählen Sie zwischen: Incident Triage On-demand, Incident RCA, Incident Mitigation oder Evaluation.
  - Anweisungen — Step-by-step Verfahren im Markdown-Format. Seien Sie spezifisch und umsetzbar.
- Klicken Sie auf „Erstellen“, um den Skill zu speichern.

Das System generiert automatisch eine SKILL.md Datei mit der richtigen Titelstruktur.

Um einen in der Benutzeroberfläche erstellten Skill zu bearbeiten:

- Navigieren Sie in der Liste der Fähigkeiten zu dem Skill und klicken Sie auf den Skill, um ihn zu öffnen.
- Klicken Sie auf Bearbeiten.
- Ändern Sie den Namen, die Beschreibung oder die Anweisungen.
- Klicken Sie auf Speichern, um den Skill zu aktualisieren.

Einen Skill hochladen

Fähigkeiten, die als ZIP-Dateien hochgeladen wurden, enthalten eine SKILL.md Datei sowie zusätzliche Ressourcen wie Referenzmaterial oder Ressourcen.

Struktur der Fähigkeiten:

```
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 Anforderungen an die erste Stelle:

Fähigkeiten, die als ZIP-Dateien hochgeladen werden, müssen Frontmatter in den Feldern SKILL.md mit `name` und `description` enthalten. AWS DevOps Der Agent bestimmt anhand dieser Felder, wann der Skill aktiviert werden soll. Einzelheiten zum Verfassen effektiver Titelseiten finden Sie weiter oben in diesem Thema im Abschnitt Frontmatter.

```
---
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...]
```

So erstellen Sie einen Skill per ZIP-Upload:

- Erstellen Sie ein Verzeichnis mit Ihren Skill-Dateien gemäß der obigen Struktur.
- Stellen Sie sicher, dass das richtige Titelbild (Name und Beschreibung) SKILL.md enthalten ist.

- Komprimieren Sie das Verzeichnis in eine ZIP-Datei.
- Navigieren Sie in Ihrer Agent Space Operator-Web-App zur Seite Skills.
- Klicken Sie auf „Skill hinzufügen“.
- Wähle im Modal „Skill hochladen“ aus.
- Ziehen Sie Ihre ZIP-Datei per Drag & Drop oder klicken Sie, um sie zu durchsuchen (nur ZIP-Dateien, maximal 6 MB).
- Wählen Sie einen oder mehrere Agententypen aus, die diesen Skill verwenden können (Generisch ist standardmäßig ausgewählt und gilt für alle Agententypen; deaktivieren Sie die Optionen „Target“, „Incident Triage“ On-demand, „Incident RCA“, „Incident Mitigation“ oder „Evaluation“).
- Überprüfen Sie die Anforderungen für die ZIP-Datei und die Validierungsergebnisse.
- Klicken Sie auf „Hochladen“, um den Skill zu Ihrem Agent Space hinzuzufügen.

Wichtige Einschränkungen für Skills, die als ZIP-Dateien hochgeladen wurden:

- Skripte werden derzeit nicht unterstützt — Skills, die Skripte im `scripts/` Verzeichnis enthalten, werden beim Upload abgelehnt. Die Skriptausführung wird in einer future Version aktiviert, sobald Agenten Zugriff auf eine sichere Codierungsumgebung haben.
- Größenbeschränkung — Die Gesamtgröße der Zip-Datei darf 6 MB (einschließlich aller Dateien) nicht überschreiten.
- SKILL.md erforderlich — Die ZIP-Datei muss eine SKILL.md Datei mit gültigem Titelbild enthalten.

Bewährte Methoden für Benennungskompetenzen:

Verwenden Sie klare, aussagekräftige Namen wie „rds-throttling-investigation“ anstelle von generischen Namen. Ein guter Skillname spiegelt das spezifische Szenario oder den Service wider, für den er sich eignet, sodass es einfacher ist, den richtigen Skill auf einen Blick zu identifizieren.

Verwalten von Qualifikationen

AWS DevOps Agent bietet über die Operator Web App umfassende Funktionen zur Verwaltung von Fähigkeiten:

Fähigkeiten auflisten — Sehen Sie sich alle Fähigkeiten in Ihrem Agentenbereich an. Auf der Seite Skills werden der Skillname, der Status Aktiv oder Inaktiv, das Erstellungsdatum, das Datum der letzten Aktualisierung und die verfügbaren Aktionen angezeigt.

Skills anzeigen — Klicken Sie auf einen Skill, um dessen Detailansicht zu öffnen. In der Benutzeroberfläche erstellte Fähigkeiten zeigen bearbeitbare Inhalte an, in denen Sie den Namen, die Beschreibung oder Anweisungen direkt in der Benutzeroberfläche ändern und zur Aktualisierung auf „Speichern“ klicken können. Bei Fähigkeiten, die als ZIP-Dateien hochgeladen wurden, wird ein Dateibaum mit allen zusätzlichen Verzeichnissen wie References/ SKILL.md und Assets/ angezeigt. Klicken Sie in der Baumstruktur auf Dateien, um deren Inhalt im schreibgeschützten Modus anzuzeigen.

Agenten für einen Skill auswählen — Konfigurieren Sie, welche Agententypen die einzelnen Skills bei der Erstellung oder Bearbeitung verwenden können. Wählen Sie in der Dropdownliste Agententyp mithilfe der folgenden Kontrollkästchen einen oder mehrere Agententypen aus: Generisch (Standard — gilt für alle Agententypen), On-demand(Konversationsanfragen), Incident Triage (erste Bewertung des Vorfalls), Incident RCA (Ursachenanalyse), Incident Mitigation (automatisierte Reaktion auf Vorfälle) oder Bewertung (proaktive Empfehlungen). Generisch ist standardmäßig ausgewählt und macht den Skill für alle Agententypen verfügbar. Fähigkeiten, die auf bestimmte Agenten ausgerichtet sind, reduzieren den Kontextverbrauch und verbessern den Fokus der Agenten.

Skills aktivieren und deaktivieren — Skills vorübergehend deaktivieren, ohne sie mit dem Active/ Inactive Schalter zu löschen. Öffne die Skill-Detailansicht und stelle den Schalter auf „Inaktiv“, um zu verhindern, dass der Agent sie für neue Untersuchungen lädt, während alle Inhalte und Konfigurationen erhalten bleiben. In-progress Bei Untersuchungen wird der Skill weiterhin verwendet. Wechsle zurück zu „Aktiv“, um den Skill sofort wieder verfügbar zu machen.

Skills aktualisieren — Ändern Sie bestehende Skills auf Grundlage der Art und Weise, wie sie erstellt wurden. Klicken Sie bei Fähigkeiten, die in der Benutzeroberfläche erstellt wurden, in der Skill-Detailansicht auf „Bearbeiten“, ändern Sie den Namen, die Beschreibung oder die Anweisungen und klicken Sie zur Aktualisierung auf „Speichern“. Bei Skills, die als ZIP-Dateien hochgeladen wurden, ändern Sie die Dateien lokal, erstellen Sie eine neue ZIP-Datei und laden Sie eine neue Version hoch.

Skills löschen — Fertigkeiten dauerhaft aus Ihrem Agentenbereich entfernen. Öffnen Sie die Skill-Listenansicht, klicken Sie auf das Menü mit weiteren Optionen () und wählen Sie „Löschen“ aus. Lesen Sie die Warnung vor dem dauerhaften Löschen, geben Sie zur Bestätigung den Namen des Skills ein und klicken Sie auf „Skill löschen“. Das Löschen kann nicht rückgängig gemacht werden. In-progress Untersuchungen können beeinträchtigt werden, wenn versucht wird, den gelöschten Skill zu laden. Bei Skills, die als ZIP-Dateien hochgeladen wurden, laden Sie die ZIP-Datei herunter, bevor Sie sie als Backup löschen. Erwägen Sie, den Skill zu deaktivieren, anstatt ihn zu löschen, falls Sie ihn erneut benötigen.

Migration von Runbooks

Bestehende Runbooks werden automatisch zu Skills migriert, ohne dass ein Eingreifen des Kunden erforderlich ist. Wenn Ihr Agent Space auf das Skills-Modell umgestellt wird, werden alle Runbooks in Skills umgewandelt und erscheinen in Ihrer Skills-Benutzeroberfläche. Nach der Migration können Sie:

- Überprüfen Sie die migrierten Skills — Vergewissern Sie sich, dass Ihre Runbooks bei der automatischen Migration korrekt konvertiert wurden.
- Nach Bedarf aktualisieren — Bearbeiten Sie Skills direkt in der Benutzeroberfläche, um Anweisungen zu verfeinern, Beschreibungen zu aktualisieren oder die Ausrichtung auf Agententypen zu konfigurieren.
- Mit Referenzen erweitern — Fertigkeiten, die von zusätzlichen Referenzmaterialien oder Architekturdiagrammen profitieren würden, können Sie sie als Zip-Upload-Skills mit einem Verzeichnis references/ oder assets/neu erstellen.
- Neue Fähigkeiten erstellen — Fügen Sie neue Fähigkeiten für Ermittlungsabläufe hinzu, die bisher nicht in Runbooks behandelt wurden.

Wenden Sie sich an den AWS Support, wenn Sie Probleme mit automatisch migrierten Skills haben oder Unterstützung bei Updates nach der Migration benötigen.

Erlernte Fähigkeiten

Was sind erlernte Fähigkeiten?

Erlernte Fähigkeiten sind strukturierte Wissensdateien, die der DevOps Agent aus Ihren Agent Space-Daten generiert. Jede erlernte Fähigkeit kodiert eine bestimmte Art von Wissen, das der AWS DevOps Agent bei der Ausführung von Aufgaben verwendet. Bei der Markteinführung stehen zwei erlernte Fähigkeiten zur Verfügung: Verständnis des Agentenbereichs und Bewährte Methoden zur Verwendung von Tools.

Überblick über den Agentenraum

Der Skill Agent Space Understanding (`understanding-agent-space`) analysiert Ihre verbundenen Cloud-Konten, Code-Repositorys und Telemetrie-Integrationen, um eine Übersicht der Ressourcen und Beziehungen in einem Agent Space zu erstellen.

Der Skill erstellt eine `SKILL.md` Hauptdatei und eine Reihe von Referenzdateien. Die Hauptdatei enthält eine einfache Systemübersicht mit den wichtigsten Domänenkonzepten, den Bereitstellungsumgebungen (AWS Konto- und Regionspaare, Azure-Abonnements und -Regionen usw.), einem Architekturdiagramm auf Containerebene, das zeigt, wie logische Dienste miteinander verbunden sind, die Anforderungspfade, die für Ihre Anwendung von zentraler Bedeutung sind, mit den Komponenten, die sie durchlaufen, und eine Zuordnung von Code-Repositorys zu Containern.

Jeder logische Container erhält eine spezielle Referenzdatei, die seine internen Komponenten (Rechenleistung, Daten, Messaging, Netzwerk und andere) mit Ressourcentypen und physischen Kennungen wie ARNs Tabellennamen und Warteschlangen beschreibt URLs. In der Referenzdatei wird auch der Umfang der Beobachtbarkeit erfasst, einschließlich der Alarme, Dashboards und Monitore, die mit den einzelnen Komponenten verknüpft sind. Außerdem ordnet sie jede Komponente den zugehörigen Code-Repositorys, Paketen und infrastructure-as-code Definitionen zu und bietet so eine vollständige Rückverfolgbarkeitskette vom Quellcode bis zu den bereitgestellten Ressourcen.

Jeder kritische Anforderungspfad erhält eine spezielle Referenzdatei, in der der gesamte end-to-end Anforderungsfluss mit der Granularität der Komponenten beschrieben wird, vom Einstiegspunkt über jeden Zwischendienst, jeden Datenspeicher und jede externe Abhängigkeit. Die Datei enthält ein sequenziertes Flussdiagramm, das die Reihenfolge der Operationen und die Interaktionsmechanismen zwischen den Komponenten sowie die Verantwortung der einzelnen Teilnehmer zeigt. Außerdem werden die für den Pfad relevanten Beobachtbarkeitssignale katalogisiert: Protokollgruppenmuster für jeden Hop, wichtige Kennzahlen (Latenz, Fehlerraten, Drosselung, Token-Kontingente) mit ihren Alarmnamen und Dimensionen sowie verteilte Trace-Spans, die zwischen Diensten und Konten korreliert werden können.

Bewährte Methoden zur Verwendung von Tools

Die Fähigkeit „Bewährte Methoden zur Verwendung von Tools“ analysiert frühere Verwendungen des Tools, um effektive Nutzungsmuster, häufige Fehlerquellen und Hinweise zu Parametern zu ermitteln. Auf diese Weise kann der DevOps Agent bekannte Fallstricke vermeiden und Untersuchungen mit weniger unnötigen Schritten durchführen. Der Skill erstellt eine Hauptdatei und eine Reihe von Referenzdateien für jedes Tool. Die Hauptdatei dient als Routing-Index, der jedes Tool mit den von ihm unterstützten Untersuchungsszenarien auflistet und auf die entsprechende Referenzdatei verweist.

Jede Referenzdatei für jedes Tool kann bis zu drei Abschnitte enthalten:

- Bewährte Methoden — Ermittlungsorientierte Techniken, die aus der erfolgreichen Verwendung von Tools gewonnen wurden, wie z. B. CloudWatch Logs Insights-Abfragevorlagen,

umgebungsspezifische Metrik-Namespaces und Dimensionen sowie Filter für Ereignisquellen. CloudTrail Jeder Eintrag ist um ein Untersuchungsszenario herum organisiert und enthält konkrete Parameterwerte und Beispiele, die in früheren Untersuchungen beobachtet wurden.

- Häufige Fehler — Wiederkehrende Fehlermodi und deren Behebung. Jeder Eintrag beschreibt eine bestimmte Fehlerbedingung, z. B. die Abfrage eines Kontos, auf das nicht zugegriffen werden kann, oder die Erstellung einer falsch formatierten Aggregationsabfrage, und bietet eine Abhilfemaßnahme, damit der Agent den Fehler vermeiden oder beheben kann, ohne weitere Untersuchungsschritte zu verschwenden.
- Output Management — Hilfestellung für Tool-Aufrufe, die in der Regel zu großen Rückmeldungen führen. Jeder Eintrag beschreibt eine Parameteränderung oder eine Verarbeitungsstrategie, mit der die Ausgabegröße reduziert und gleichzeitig der diagnostische Wert erhalten bleibt.

Wenn Live-Zugriff auf die Infrastruktur verfügbar ist, validiert der Skill Muster anhand Ihrer Umgebung, bevor er sie einbezieht. Bestätigte Muster werden mit Sicherheit angegeben, unbestätigte Muster werden vorsichtig formuliert und widerlegte Muster werden ausgeschlossen. Dadurch wird sichergestellt, dass Ihre Fähigkeiten stets auf den aktuellen Stand Ihrer Infrastruktur abgestimmt sind.

Verwaltung der erlernten Fähigkeiten

Aktualisierungen — Der DevOps Agent generiert und aktualisiert automatisch erlernte Fähigkeiten auf der Grundlage der Aktivitäten in Ihrem Agentenbereich. Im Folgenden wird beschrieben, wann die einzelnen Fähigkeiten aktualisiert werden.

Der DevOps Agent generiert alle 30 Untersuchungen eine aktualisierte Fähigkeit zur Verwendung von Tools mit bewährten Methoden.

Der Skill Agent Space Understanding wird vom Lernagenten generiert, der immer dann ausgeführt wird, wenn Sie eine Agent Space-Funktion oder -Integration hinzufügen, aktualisieren oder entfernen.

Um erlernte Fähigkeiten manuell zu regenerieren, wählen Sie auf der Seite „Topologie“ in der Operator-App die Schaltfläche „Regenerieren“ oder chatten Sie mit dem Agenten und bitten Sie ihn, die erlernten Fähigkeiten zu aktualisieren.

Deaktivierung — Erlernte Fähigkeiten sind standardmäßig aktiv. Wenn sie aktiv sind, lädt der DevOps Agent sie zu Beginn jeder DevOps Agentenaufgabe. Um zu verhindern, dass eine erlernte Fähigkeit angewendet wird, deaktivieren Sie sie in der Skill-Anzeige in der Operator-App. Wenn Sie einen Skill deaktivieren, wird er nicht gelöscht. Der Skill bleibt erhalten und kann jederzeit wieder aktiviert

werden. Wenn eine Fähigkeit deaktiviert ist, arbeitet der DevOps Agent ohne das Wissen dieser Fähigkeit.

Topologieansicht — Die Topologieseite in der Web-App Ihres Agent Space verwendet den Agent Space Understanding Skill, um Ihre Agent Space-Umgebung visuell als logische Container und Komponenten darzustellen. Klicken Sie auf einen beliebigen Container, um seine Komponenten, Ressourcen-IDs und Telemetrie zu sehen.

Anweisungen für Agenten

Verwenden Sie die Anweisungen des Agenten, um stets aktuelle Anweisungen zu geben, die der AWS DevOps Agent bei jeder Sitzung anwendet. Eine Sitzung ist eine einzelne Konversation oder Untersuchung mit einem Agenten. Auf der Agenten-Seite in Ihrer Agent Space Operator-Web-App können Sie globale Anweisungen festlegen, die für alle Agenten gelten, oder Anweisungen für einen bestimmten verwalteten Agenten festlegen, z. B. Chat oder Incident-Triage. Diese Anweisungen werden als AGENTS.md Datei gespeichert. Im Gegensatz zu Programmenthe section called “DevOps Fähigkeiten der Agenten”, die bei Bedarf geladen werden, wenn der Agent der aktuellen Aufgabe eine Skillbeschreibung zuordnet, sind die Anweisungen des Agenten immer von Beginn jeder Sitzung an vorhanden, unabhängig davon, woran der Agent gerade arbeitet.

Was sind Anweisungen für Agenten

Anweisungen für Agenten bieten eine bedingungslose, stets verfügbare Anleitung für Ihre Agenten. Zu Beginn jeder Sitzung ruft der Agentendienst die für Ihren Agent Space konfigurierten Anweisungen ab und fügt deren Inhalt direkt in die Eingabeaufforderung des Agentensystems ein. Der Agent entscheidet nicht, ob sie geladen werden sollen; sie sind immer vorhanden.

Jede Agentensitzung erhält Anweisungen sowohl aus den globalen Anweisungen als auch aus den entsprechenden agentenspezifischen Anweisungen, z. B. Chat.

Agentenanweisungen werden als AGENTS.md Dateien gespeichert und unterscheiden sich the section called “DevOps Fähigkeiten der Agenten” in mehreren wichtigen Punkten von:

Aspekt	Fähigkeit	Anweisungen für Agenten (AGENTS.md)
Name und Beschreibung	Erforderlich	Nicht zutreffend

Aspekt	Fähigkeit	Anweisungen für Agenten (AGENTS.md)
Format des Inhalts	Markdown- oder ZIP-Paket	Nur Markdown
Ressourcendateien	Unterstützt	Nicht unterstützt
Kontext-Injektion	Auf Anfrage (der Agent entscheidet anhand des Abgleichs der Beschreibung der Fähigkeiten)	Immer (bedingungslos, jede Sitzung)
Eindeutigkeit	Mehrere pro Agentenbereich	Eine pro Agent (eine für globale Anweisungen, eine pro verwaltetem Agenten)

Anweisungen für Agenten haben kein Namens- oder Beschreibungsfeld. Die zugrunde liegende AGENTS.md Datei enthält nur Markdown ohne Frontmatter, ohne Unterstützung für ZIP-Pakete und ohne Ressourcendateien.

Warum sollten Sie Anweisungen für Agenten verwenden

Mit Anweisungen für Agenten können Sie auf zuverlässige Weise sicherstellen, dass bestimmte Anleitungen immer im richtigen Kontext stehen, ohne von den Entscheidungen des Agenten abhängig zu sein, die Fähigkeiten zu erweitern.

Die wichtigsten Vorteile:

- Vorhersehbarkeit: Anweisungen sind immer vorhanden, unabhängig davon, an welcher Aufgabe der Agent gerade arbeitet. Es ist kein Abgleich der Beschreibung erforderlich, und der Agent kann den Inhalt nicht überspringen.
- Garantierter Schutz: Im Gegensatz zu Fähigkeiten, die der Agent je nach Relevanz der Aufgabe laden kann oder auch nicht, werden die Anweisungen des Agenten immer zu Beginn jeder Sitzung eingegeben.
- Ständige Richtlinien: Verwenden Sie die Anweisungen der Agenten für dauerhafte Betriebsrichtlinien, Sicherheitsrichtlinien, Codierungsstandards oder andere Richtlinien, die ausnahmslos für jede Sitzung gelten müssen.

- Gezielter Geltungsbereich: Sie können mithilfe globaler Anweisungen Anweisungen auf alle Agententypen gleichzeitig anwenden oder Anweisungen auf einen bestimmten Agententyp beschränken, wenn die Anweisungen nur für die Arbeit dieses Agenten relevant sind.

So funktionieren Anweisungen für Agenten

Wenn eine Sitzung gestartet wird, ruft der Agentendienst die für Ihren Agent Space konfigurierten Anweisungen ab und fügt deren Inhalt vor Beginn der Sitzung in die Eingabeaufforderung des Agentensystems ein. Dies geschieht automatisch für jede Sitzung. Der Agent bewertet nicht, ob sie geladen werden sollen; er fügt den Inhalt immer ein.

Bei jeder neuen Sitzung werden die Anweisungen beim Start neu geladen. Wenn Sie Ihre Anweisungen aktualisieren, wird die Änderung sofort für Sitzungen wirksam, die nach dem Speichern beginnen. Sitzungen, die bereits laufen, verwenden weiterhin den Inhalt, der beim Start geladen wurde.

Der Geltungsbereich bestimmt, welche Anweisungen eine Sitzung erhält. Globale Anweisungen gelten für alle Agententypen in Ihrem Agentenbereich, sodass sie in jeder Sitzung empfangen werden. Agent-specific Anweisungen gelten nur für Sitzungen dieses bestimmten Agententyps. Eine Sitzung erhält Anweisungen sowohl aus den globalen Anweisungen als auch aus den entsprechenden agentenspezifischen Anweisungen.

Geltungsbereich des Agententyps

Das Scoping steuert, welche Agentensitzungen einen bestimmten Befehlssatz erhalten. Es gibt zwei Optionen für die Festlegung des Geltungsbereichs:

- Allgemeine Anweisungen: Gilt für alle Agententypen in Ihrem Agentenbereich. Jede Agentensitzung erhält diesen Inhalt.
- Agent-specific: Gilt nur für Sitzungen des ausgewählten Agententyps.

Folgende verwaltete Agenten stehen für agentenspezifische Anweisungen zur Verfügung:

- Chat — Ad-hoc Fragen und Anfragen während Chat-Sitzungen.
- Triage von Vorfällen — Filterung von Alarmen, Klassifizierung des Schweregrads und anfängliche Festlegung des Umfangs.
- Incident RCA — Ursachenanalyse mit Erfassung und Validierung von Nachweisen.

- Minderung von Zwischenfällen — Empfehlungen Short-term zur Behebung und langfristigen Behebung.
- Bewertung — Bewertung der Agentenleistung und Überprüfung der Einhaltung von Richtlinien.

Hinweise zur Größe von Inhalten

Jedes Mal, wenn Sie ein Gespräch oder eine Untersuchung beginnen, liest der Agent Ihre gesamten Anweisungen, bevor er etwas anderes tut. Der Agent verfügt über eine feste Menge an Arbeitsspeicher pro Sitzung, und Ihre Anweisungen verwenden einen Teil davon. Je größer die Datei, desto weniger Platz bleibt für Ihre Fragen, Untersuchungen, die vom Agenten gelesenen Protokolle und seine eigenen Überlegungen. Kürzere, zielgerichtete Anweisungen geben dem Agenten mehr Möglichkeiten, Ihr Problem zu lösen.

- Festes Limit: 25 KB
- Empfohlene Größe: 120 Zeilen (für die meisten Konfigurationen empfohlen)

Konzentrieren Sie sich bei Ihren Anweisungen auf Anleitungen, die in jeder Sitzung enthalten sein müssen. Für spezielle Ermittlungsverfahren, die nur für bestimmte Aufgaben gelten, sollten Sie the section called “DevOps Fähigkeiten der Agenten” stattdessen die Verwendung von.

Beispiel

Das folgende Beispiel zeigt gut formulierte Anweisungen für Agenten mit Hinweisen zur Untersuchung, Formatierungsstandards für Antworten und Sicherheitsanforderungen, die für jede Agentensitzung gelten.

```
# Agent Instructions

## Investigation approach
- Always check CloudWatch alarms and recent deployments before proposing a root cause.

## Response format
- Lead with a one-sentence summary of findings before listing details.
- Include the AWS region and resource identifier for any resource you reference.
- Use bullet points for lists of findings or recommendations.

## Security
- Never log, display, or suggest storing credentials or secrets in plaintext.
```

```
- When recommending IAM changes, follow least-privilege principles.
```

Anweisungen für den Agenten einrichten

Bevor Sie Agentenanweisungen festlegen können, benötigen Sie einen Agent Space. Weitere Informationen finden Sie unter the section called “Einen Agentenbereich erstellen”.

Jeder Agent hat genau einen Befehlssatz. Wenn Sie neuen Inhalt speichern, überschreibt er den vorhandenen Inhalt für diesen Agenten.

So legen Sie globale Anweisungen fest (gilt für alle Agenten):

1. Navigieren Sie in Ihrer Agent Space Operator-Web-App zur Seite „Agenten“.
2. Wählen Sie neben „Allgemeine Anweisungen“ die Option „Anzeigen“.
3. Geben Sie Ihre Markdown-Anweisungen im Editor ein.
4. Wählen Sie Speichern.

So legen Sie Anweisungen für einen bestimmten Agenten fest:

1. Navigieren Sie in Ihrer Agent Space Operator-Web-App zur Seite Agenten.
2. Wählen Sie unter Verwaltete Agenten neben dem Agenten, den Sie konfigurieren möchten, die Option Ansicht aus: Chat, Incident Triage, Incident RCA, Incident Mitigation oder Evaluation.
3. Geben Sie Ihre Markdown-Anweisungen im Editor ein.
4. Wählen Sie Speichern.

Anweisungen für Agenten verwalten

AWS DevOps Agent bietet Verwaltungsfunktionen für Agentenanweisungen über die Operator Web App.

Anweisungen anzeigen: Navigieren Sie zur Seite „Agenten“ und wählen Sie neben „Allgemeine Anweisungen“ oder dem jeweiligen verwalteten Agenten die Option „Anzeigen“. Der Editor zeigt den aktuellen Inhalt an. Verwenden Sie die Registerkarte Vorschau, um den gerenderten Markdown zu sehen, oder die Registerkarte Code, um den unformatierten Markdown zu sehen.

Anweisungen zur Bearbeitung: Öffnen Sie den Agenten wie oben beschrieben, ändern Sie den Inhalt im Editor und wählen Sie Speichern.

Anweisungen aus einer Datei hochladen: Öffnen Sie den Agenten und wählen Sie dann im Editor die Schaltfläche Hochladen, um eine Markdown-Datei von Ihrem Computer hochzuladen.

Anweisungen zum Herunterladen: Öffnen Sie den Agenten und wählen Sie dann im Editor die Schaltfläche Herunterladen, um den aktuellen Inhalt als Datei herunterzuladen.

Anweisungen zum Löschen: Öffnen Sie den Agenten, wählen Sie im Editor die Schaltfläche Löschen und bestätigen Sie den Löschvorgang. Diese Aktion kann nicht rückgängig gemacht werden. Erwägen Sie, den Inhalt zuerst herunterzuladen, falls Sie ihn erneut benötigen.

Unterstützte Regionen

In diesem Thema werden die AWS Regionen beschrieben, in denen Sie AWS DevOps Agent verwenden können. Weitere Informationen zu AWS Regionen finden Sie im Referenzhandbuch zur Kontoverwaltung unter „Geben Sie an, welche AWS Regionen Ihr AWS Konto verwenden kann“.

Regionsübergreifende Ressourcenüberwachung

AWS DevOps Der Agent kann Ressourcen in AWS Konten in jeder AWS Region überwachen und untersuchen, unabhängig davon, in welcher unterstützten Region Sie Ihren Agent-Bereich einrichten. Wenn Sie ein AWS Konto mit einem Agent Space verknüpfen, erkennt der Agent Ressourcen in allen Regionen innerhalb dieses Kontos und ordnet sie zu. Das bedeutet, dass Sie nicht in jeder Region, in der Ihre Workloads ausgeführt werden, einen Agentenbereich benötigen.

Wählen Sie eine unterstützte Region basierend auf Ihrem bevorzugten Datenstandort, der Nähe zu Ihrem Betriebsteam oder den organisatorischen Anforderungen.

Unterstützte Regionen

AWS DevOps Der Agent ist in den folgenden AWS Regionen verfügbar.

Name der Region	Regionscode	Link zur Konsole
USA Ost (Nord-Virginia)	us-east-1	Konsole öffnen
USA West (Oregon)	us-west-2	Konsole öffnen
Asien-Pazifik (Sydney)	ap-southeast-2	Konsole öffnen
Asien-Pazifik (Tokio)	ap-northeast-1	Konsole öffnen

Name der Region	Regionscode	Link zur Konsole
Europa (Frankfurt)	eu-central-1	Konsole öffnen
Europa (Irland)	eu-west-1	Konsole öffnen

Service-Endpunkte

Name der Region	Regionscode	Endpoint	Protocol (Protokoll)
USA Ost (Nord-Virginia)	us-east-1	aidevops.us-east-1.amazonaws.com	HTTPS
USA West (Oregon)	us-west-2	aidevops.us-west-2.amazonaws.com	HTTPS
Asien-Pazifik (Sydney)	ap-southeast-2	aidevops.ap-southeast-2.amazonaws.com	HTTPS
Asien-Pazifik (Tokio)	ap-northeast-1	aidevops.ap-northeast-1.amazonaws.com	HTTPS
Europa (Frankfurt)	eu-central-1	aidevops.eu-central-1.amazonaws.com	HTTPS
Europa (Irland)	eu-west-1	aidevops.eu-west-1.amazonaws.com	HTTPS

Überlegungen

- Auswahl der Agent Space-Region — Ein Agent Space und seine Daten (Untersuchungen,

Topologie, Empfehlungen) werden in der Region gespeichert, in der Sie ihn erstellen. Wählen Sie eine Region aus, die Ihre Anforderungen an die Datenresidenz erfüllt.

- Regionsübergreifende Überwachung — Ressourcen in AWS Konten, die einem Agenten zugeordnet sind

Der Speicherplatz wird unabhängig davon überwacht, in welcher Region diese Ressourcen eingesetzt werden. Sie müssen nicht in jeder Region, in der Ihre Workloads ausgeführt werden, separate Agentenbereiche einrichten.

- Integrationen von Drittanbietern — Verbindungen zu CI/CD Anbietern (GitHub,), GitLab

Observability-Tools (Dynatrace, Datadog, New Relic, Splunk) und MCP-Server werden pro Agent Space konfiguriert und sind nicht regionsabhängig.

Erste Schritte mit AWS DevOps Agent

In diesem Handbuch für die ersten Schritte erstellen Sie einen grundlegenden Agent-Bereich, konfigurieren Mindestberechtigungen und führen Ihre erste KI-gestützte Untersuchung durch.

Themen:

Einen Agentenbereich erstellen

Ein Agent Space definiert die Tools und die Infrastruktur, auf die der AWS DevOps Agent Zugriff hat. Diese Anleitung führt Sie durch die Einrichtung eines Agent Space, die Konfiguration des Zugriffs auf das primäre Konto und die Aktivierung der DevOps Agent Web App. Weitere Informationen zum Agent Space-Konzept finden Sie unter „Was ist ein Agent Space“.

Einen Agent Space erstellen

Greifen Sie auf die AWS DevOps Agentenkonsole zu

1. Melden Sie sich bei der AWS Managementkonsole an
2. Navigieren Sie zur AWS DevOps Agent-Konsole

Geben Sie dem Agent-Bereich einen Namen

1. Klicken Sie auf Agentenbereich erstellen

Geben Sie im Abschnitt „Agent Space-Details“ Folgendes ein:

1. Geben Sie im Feld Name einen Namen für Ihren Agent Space ein

2. (Optional) Fügen Sie im Feld Beschreibung Details zum Zweck des Agent Space hinzu
3. (Optional) Wählen Sie in der Dropdownliste „Sprache der Agentenantwort“ die Sprache aus, die der Agent bei der Generierung von Antworten, Ergebnissen und Ergebnissen der Untersuchung verwendet. Zu den Optionen gehören: Bahasa Indonesisch, Chinesisch (Simplified/PRC), Chinese (Traditional/Taiwan), Englisch (Großbritannien), Französisch (Frankreich), Deutsch (Deutschland), Italienisch (Italien), Japanisch (Japan), Koreanisch (Korea), Portugiesisch (Brasilien), Spanisch (Lateinamerika), Türkisch (Türkei), Arabisch (Saudi-Arabien), Thailändisch (Thailand) und Vietnamesisch (Vietnam). Wenn keine Sprache ausgewählt ist, antwortet der Agent in der Sprache der Eingabe. Diese Einstellung wird auch verwendet, um die Sprache für AWS Supportanfragen festzulegen, die mit der Funktion „Kundensupport fragen“ erstellt wurden.

Konfiguration des Zugriffs auf das primäre Konto

Im Abschnitt Diesem Agent AWS Space-Ressourcenzugriff gewähren richten Sie eine IAM-Rolle ein, um dem Agent Space Zugriff auf das AWS Hauptkonto zu gewähren. Das Hauptkonto ist das AWS Konto, in dem Sie Ihren Agent Space erstellen. AWS DevOps Der Agent benötigt eine IAM-Rolle, um bei Ermittlungen AWS Ressourcen in diesem Konto zu finden und darauf zuzugreifen.

Wählen Sie eine Methode zur Rollenkonfiguration. Wählen Sie eine der folgenden Optionen aus:

Option 1: Automatische Erstellung einer neuen AWS DevOps Agentenrolle (empfohlen)

Mit dieser Option wird automatisch eine Rolle mit den entsprechenden Berechtigungen für den AWS DevOps Agenten erstellt, um Ressourcen in Ihrem Konto zu untersuchen.

> **Note**
>
> Sie müssen über IAM-Berechtigungen verfügen, um neue Rollen zu erstellen, um diese Option verwenden zu können.

1. Wählen Sie Neue AWS DevOps Agentenrolle automatisch erstellen
2. (Optional) Aktualisieren Sie den Namen der Agent Space-Rolle, die erstellt werden soll

Option 2: Weisen Sie eine bestehende Rolle zu

Verwenden Sie diese Option, wenn ein anderer Administrator zuvor eine Rolle speziell für den AWS DevOps Agenten erstellt hat.

1. Wählen Sie Eine bestehende Rolle zuweisen aus
2. Wählen Sie im Dropdownmenü eine bestehende Rolle aus, die über die entsprechenden Berechtigungen verfügt

Option 3: Erstellen Sie mithilfe einer Richtlinienvorlage eine neue AWS DevOps Agentenrolle

Verwenden Sie diese Option, wenn Sie die Dienste und Ressourcen einschränken müssen, auf die der Agent im Hauptkonto zugreifen kann.

1. Wählen Sie Neue AWS DevOps Agentenrolle mithilfe einer Richtlinienvorlage erstellen aus
2. Folgen Sie den Anweisungen, um die Vertrauensrichtlinie und die Inline-Richtlinie für die neue Rolle zu erstellen.

Die Agent Space Web App aktivieren

In der Web-App interagieren Mitarbeiter mit dem AWS DevOps Agenten, um Vorfälle zu untersuchen und Empfehlungen zu überprüfen. Weitere Informationen finden Sie unter Architektur der AWS DevOps Agentenkonsole [Link]. Wenn diese Option aktiviert ist, können Benutzer über einen IAM-Authentifizierungslink von der AWS Managementkonsole aus auf die Agent Space Web App zugreifen.

Wählen Sie eine der folgenden Optionen aus:

Option 1: Automatische Erstellung einer neuen AWS DevOps Agentenrolle (empfohlen)

Diese Option erstellt automatisch eine Rolle mit den entsprechenden Berechtigungen für den Zugriff auf die DevOps Agent Web App.

> **Note**
>
> Sie müssen über IAM-Berechtigungen verfügen, um neue Rollen zu erstellen, um diese Option verwenden zu können.

1. Wählen Sie Neue AWS DevOps Agentenrolle automatisch erstellen
2. Überprüfen Sie die Berechtigungen, die der Rolle gewährt werden

Option 2: Weisen Sie eine bestehende Rolle zu

Verwenden Sie diese Option, wenn ein anderer Administrator zuvor eine Operatorrolle erstellt hat.

1. Wählen Sie Eine bestehende Rolle zuweisen aus
2. Wählen Sie im Dropdownmenü eine bestehende Rolle aus, die über die entsprechenden Berechtigungen verfügt

Option 3: Erstellen Sie mithilfe einer Richtlinienvorlage eine neue AWS DevOps Agentenrolle

Verwenden Sie diese Option, wenn Sie die Berechtigungen für den Zugriff auf Webanwendungen anpassen müssen.

1. Wählen Sie Neue AWS DevOps Agentenrolle mithilfe einer Richtlinienvorlage erstellen aus
2. Folgen Sie den Anweisungen, um die Vertrauensrichtlinie und die Inline-Richtlinie für die neue Rolle zu erstellen.

Hinzufügen von Stichwörtern (optional)

Sie können Ihrem Agent Space während der Erstellung AWS Tags hinzufügen. Tags sind Schlüssel-Wert-Paare, die Ihnen helfen, Ihre Ressourcen zu organisieren und zu identifizieren. Sie können bis zu 50 Tags pro Agent Space hinzufügen. Um Tags hinzuzufügen, erweitern Sie auf der Seite Create Agent Space den Abschnitt Tags und klicken Sie auf Neues Tag hinzufügen.

Schließen Sie die Erstellung des Agentenbereichs ab

Sobald alle Abschnitte ausgefüllt sind, klicken Sie auf Erstellen

Überprüfen Sie Ihre Agent Space-Einrichtung

Nach der Konfiguration erscheint die Schaltfläche für den Operator-Zugriff auf der Agent-Space-Detailseite. Wenn Sie darauf klicken, wird die Web-App auf einer neuen Registerkarte geöffnet und die Authentifizierung wurde erfolgreich durchgeführt.

Nächste Schritte

Nachdem Sie Ihren Agent Space eingerichtet haben, sollten Sie die folgenden Schritte in Betracht ziehen:

- Fügen Sie sekundäre Konten hinzu, wenn sich Ihre Anwendungen auf mehrere AWS Konten erstrecken
- Konfigurieren Sie Integrationen von Drittanbietern wie Observability-Tools oder Ticketsysteme
- Richten Sie die AWS Identity Center-Authentifizierung für Produktionsumgebungen ein
- Machen Sie sich mit der Zuordnung Ihrer Anwendungsressourcen vertraut, damit AWS DevOps Agent Ihre Infrastruktur besser versteht

AWS DevOps Leitfaden für das CLI Onboarding von Agenten

-Übersicht

Mit AWS DevOps Agent können Sie Ihre AWS Infrastruktur überwachen und verwalten.
Diese Anleitung führt Sie durch die Einrichtung des AWS DevOps Agenten mithilfe der AWS Befehlszeilenschnittstelle (AWS CLI). Sie erstellen IAM-Rollen, richten einen Agentenbereich ein und verknüpfen Ihr AWS Konto. Sie aktivieren auch die Operator-App und verbinden optional Integrationen von Drittanbietern. Das Ausfüllen dieses Handbuchs dauert ungefähr 20 Minuten.

AWS DevOps Der Agent ist in sechs AWS Regionen verfügbar: USA Ost (Nord-Virginia), USA West (Oregon), Asien-Pazifik (Sydney), Asien-Pazifik (Tokio), Europa (Frankfurt) und Europa (Irland). Weitere Informationen zu den unterstützten Regionen finden Sie unterthe section called “Unterstützte Regionen”.

Voraussetzungen

Stellen Sie vor dem Beginn sicher, dass Sie über Folgendes verfügen:

- AWS CLI Version 2 installiert und konfiguriert
- Authentifizierung für Ihr AWS Monitoring-Konto
- Berechtigungen zum Erstellen von AWS Identitäts- und Zugriffsmanagement-Rollen (IAM) und zum Anhängen von Richtlinien
- Ein AWS Konto, das als Überwachungskonto verwendet werden soll
- Vertrautheit mit der AWS CLI- und JSON-Syntax

Ersetzen Sie in diesem Handbuch die folgenden Platzhalterwerte durch Ihre eigenen:

- <MONITORING_ACCOUNT_ID>— Ihre 12-stellige AWS Konto-ID für das (primäre) Überwachungskonto
- <EXTERNAL_ACCOUNT_ID>— Die 12-stellige AWS Konto-ID des zu überwachenden Sekundärkontos (wird in Schritt 4 verwendet)
- <REGION>— Der AWS Regionalcode für Ihren Agentenbereich (z. B. `us-east-1` oder`eu-central-1`)
- <AGENT_SPACE_ID>— Die Agenten-Space-ID, die vom `create-agent-space` Befehl zurückgegeben wird

Einrichtung von IAM-Rollen

1. Erstellen Sie die DevOps Agent-Space-Rolle

Erstellen Sie die IAM-Vertrauensrichtlinie, indem Sie den folgenden Befehl ausführen:

```
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
```

Erstellen Sie die IAM-Rolle:

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

Speichern Sie den Rollen-ARN, indem Sie den folgenden Befehl ausführen:

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

Hängen Sie die AWS verwaltete Richtlinie an:

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

Erstellen Sie eine Inline-Richtlinie und fügen Sie sie an, um die Erstellung der serviceverknüpften Resource Explorer-Rolle zu ermöglichen:

```
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. Erstellen Sie die IAM-Rolle der Operator-App

Erstellen Sie die IAM-Vertrauensrichtlinie, indem Sie den folgenden Befehl ausführen:

```
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
```

Erstellen Sie die IAM-Rolle:

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

Speichern Sie den Rollen-ARN, indem Sie den folgenden Befehl ausführen:

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

Hängen Sie die Richtlinie für AWS verwaltete Operator-Apps an:

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

Diese verwaltete Richtlinie gewährt der Operator-App Berechtigungen für den Zugriff auf Funktionen im Agentenbereich. Zu diesen Funktionen gehören Untersuchungen, Empfehlungen, Wissensmanagement, Chat und AWS Support-Integration. Die Richtlinie beschränkt den Zugriff auf den jeweiligen Agentenbereich anhand der `aws:PrincipalTag/AgentSpaceId` Bedingung. Weitere Informationen zur vollständigen Liste der Aktionen finden Sie unterthe section called “DevOps IAM-Berechtigungen für Agenten”.

Schritte zum Onboarding

1. Erstellen Sie einen Agentenbereich

Führen Sie den folgenden Befehl aus, um einen Agentenbereich zu erstellen:

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

Geben Sie optional `--kms-key-arn` an, dass ein vom Kunden verwalteter AWS KMS-Schlüssel für die Verschlüsselung verwendet werden soll. Sie können ihn auch verwenden`--tags`, um Ressourcen-Tags `--locale` hinzuzufügen und die Sprache für Agentenantworten festzulegen.

Speichern Sie das `agentSpaceId` aus der Antwort (befindet sich unter`agentSpace.agentSpaceId`).

Führen Sie den folgenden Befehl aus, um Ihre Agentenbereiche später aufzulisten:

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

2. Ordnen Sie Ihr AWS Konto zu

Ordnen Sie Ihr AWS Konto zu, um die Topologieerkennung zu aktivieren. Stellen Sie `accountType` für einen der folgenden Werte ein:

- `monitor`— Das primäre Konto, in dem der Agentenbereich vorhanden ist. Dieses Konto hostet den Agenten und wird für die Topologieermittlung verwendet.
- `source`— Ein zusätzliches Konto, das der Agent überwacht. Verwenden Sie diesen Typ, wenn Sie in Schritt 4 externe Konten zuordnen.

```
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. Aktivieren Sie die Operator-App

Authentifizierungsabläufe können IAM, IAM Identity Center (IDC) oder einen externen Identitätsanbieter (IdP) verwenden. Führen Sie den folgenden Befehl aus, um die Operator-App für Ihren Agentenbereich zu aktivieren:

```
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>
```

Verwenden `--auth-flow idc` und stellen Sie für die IAM Identity Center-Authentifizierung bereit`--idc-instance-arn`. Verwenden `--auth-flow idp` und stellen Sie für einen externen Identitätsanbieter `--issuer-url``--idp-client-id`, und `--idp-client-secret` bereit.

Weitere Informationen erhalten Sie unter the section called “Einrichtung der IAM Identity Center-Authentifizierung” und the section called “Authentifizierung durch externen Identitätsanbieter (IdP) einrichten”.

> Hinweis: Wenn Sie zuvor eine Operator-App-Rolle für einen anderen Agentenbereich in Ihrem Konto erstellt haben, können Sie diesen Rollen-ARN wiederverwenden.

4. (Optional) Ordnen Sie zusätzliche Quellkonten zu

Um weitere Konten mit AWS DevOps Agent zu überwachen, erstellen Sie eine kontenübergreifende IAM-Rolle.

Erstellen Sie die kontoübergreifende Rolle im externen Konto

Wechseln Sie zum externen Konto und erstellen Sie die Vertrauensrichtlinie. Das `MONITORING_ACCOUNT_ID` ist das Hauptkonto, das den Agent-Bereich hostet, den Sie in Schritt 2 eingerichtet haben. Diese Konfiguration ermöglicht es dem AWS DevOps Agent-Dienst, im Namen des Überwachungskontos eine Rolle in den sekundären Quellkonten zu übernehmen.

Führen Sie den folgenden Befehl aus, um die Vertrauensrichtlinie zu erstellen:

```
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
```

Erstellen Sie die kontoübergreifende IAM-Rolle:

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

Speichern Sie den Rollen-ARN, indem Sie den folgenden Befehl ausführen:

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

Hängen Sie die AWS verwaltete Richtlinie an:

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

Hängen Sie die Inline-Richtlinie an, um die Erstellung der mit dem Resource Explorer-Dienst verknüpften Rolle im externen Konto zu ermöglichen:

```
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
```

Ordnen Sie das externe Konto zu

Wechseln Sie zurück zu Ihrem Überwachungskonto und führen Sie dann den folgenden Befehl aus, um das externe Konto zuzuordnen:

```
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. (Optional) Zuordnen GitHub

> Hinweis: Sie müssen sich zuerst GitHub über die AWS DevOps Agentenkonsole registrieren, indem Sie den OAuth Flow verwenden, bevor Sie ihn über die CLI verknüpfen können.

Anweisungen zur Registrierung GitHub über die Konsole finden Sie unterthe section called “Anschluss an CI/CD Rohrleitungen”.

Listet die registrierten Dienste auf:

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

Speichern Sie den <SERVICE_ID> für ServiceType:. `github`

Nachdem Sie sich GitHub in der Konsole registriert haben, ordnen Sie GitHub Repositorys zu, indem Sie den folgenden Befehl ausführen:

```
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. (Optional) Registrieren und Zuordnen ServiceNow

Registrieren Sie zunächst den ServiceNow Dienst mit den OAuth Anmeldeinformationen:

```
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>
```

Speichern Sie die zurückgegebenen <SERVICE_ID> Daten und verknüpfen Sie dann ServiceNow:

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

7. (Optional) Registrieren Sie Dynatrace und verknüpfen Sie es

Registrieren Sie zunächst den Dynatrace-Dienst mit Anmeldeinformationen: OAuth

```
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>
```

Speichern Sie das zurückgesendete Dynatrace <SERVICE_ID> und ordnen Sie es dann zu. Ressourcen sind optional. Die Umgebung gibt an, welcher Dynatrace-Umgebung zugeordnet werden soll.

```
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>
```

Die Antwort enthält Webhook-Informationen für die Integration. Sie können diesen Webhook verwenden, um eine Untersuchung von Dynatrace auszulösen. Weitere Informationen finden Sie unter the section called “Dynatrace verbinden”.

8. (Optional) Registrieren Sie Splunk und verknüpfen Sie es

Registrieren Sie zunächst den Splunk-Service mit Zugangsdaten. BearerToken

Der Endpunkt verwendet das folgende Format: `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>
```

Speichern Sie den zurückgegebenen Text `<SERVICE_ID>` und verknüpfen Sie ihn anschließend mit Splunk:

```
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>
```

Die Antwort enthält Webhook-Informationen für die Integration. Sie können diesen Webhook verwenden, um eine Untersuchung von Splunk auszulösen. Weitere Informationen finden Sie unter the section called “Splunk verbinden”.

9. (Optional) Registrieren Sie New Relic und verknüpfen Sie es

Registrieren Sie zunächst den New Relic-Service mit den API-Schlüsselanmeldedaten.

Region: Entweder US oderEU.

Optionale Felder:`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>
```

Speichern Sie das zurückgegebene <SERVICE_ID> und verknüpfen Sie es dann mit New Relic:

```
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>
```

Die Antwort enthält Webhook-Informationen für die Integration. Sie können diesen Webhook verwenden, um eine Untersuchung von New Relic auszulösen. Weitere Informationen finden Sie unter the section called “New Relic verbinden”.

10. (Optional) Registrieren Sie Datadog und verknüpfen Sie es

Sie müssen Datadog zuerst über die AWS DevOps Agentenkonsole mithilfe des OAuth Flows registrieren, bevor Sie es über die CLI verknüpfen können. Weitere Informationen finden Sie unter the section called “Verbindung herstellen DataDog”.

Listet die registrierten Dienste auf:

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

Speichern Sie den `<SERVICE_ID>` für ServiceType:. `mcpserverdatadog`

Dann ordne Datadog zu:

```
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>
```

Die Antwort enthält Webhook-Informationen für die Integration. Sie können diesen Webhook verwenden, um eine Untersuchung von Datadog auszulösen. Weitere Informationen finden Sie unter the section called “Verbindung herstellen DataDog”.

11. (Optional) Löschen Sie einen Agentenbereich

Durch das Löschen eines Agentenbereichs werden alle Zuordnungen, Konfigurationen und Ermittlungsdaten für diesen Agentenbereich entfernt. Diese Aktion kann nicht rückgängig gemacht werden.

Führen Sie den folgenden Befehl aus, um einen Agentenbereich zu löschen:

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

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

Verifizierung

Führen Sie die folgenden Befehle aus, um Ihr Setup zu überprüfen:

```
# 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>
```

Nächste Schritte

- Informationen zum Verbinden zusätzlicher Integrationen finden Sie unterKonfiguration von Funktionen für AWS DevOps Agent.
- Weitere Informationen zu den Fähigkeiten und Fertigkeiten von Agenten finden Sie unterthe section called “DevOps Fähigkeiten der Agenten”.
- Weitere Informationen zur Web-App für Operatoren finden Sie unterthe section called “Was ist eine DevOps Agent-Web-App?”.

Hinweise

- Ersetzen Sie `<AGENT_SPACE_ID>` `<MONITORING_ACCOUNT_ID><EXTERNAL_ACCOUNT_ID>`,`<REGION>`,, usw. durch Ihre tatsächlichen Werte.
- Eine Liste der unterstützten -Regionen finden Sie unter the section called “Unterstützte Regionen”.

Eine Testumgebung erstellen

Dieses Handbuch enthält praktische Tests zur Validierung der Incident-Response-Funktionalität des AWS DevOps Agenten anhand einer Beispielarchitektur. Verwenden Sie diese Ergänzung, wenn Sie DevOps Agent testen möchten, bevor Sie Ihre Produktionssysteme anschließen.

Voraussetzungen

- AWS Konto mit Administratorzugriff
- AWS DevOps Agent Space, der mit dem Rollenablauf „ DevOps Agent automatisch erstellen“ erstellt und konfiguriert wurde
- Für den EC2-Test: eine bestehende VPC mit mindestens einem Subnetz in der Region, in der Sie die Bereitstellung durchführen werden.

Überblick über Kosten und Sicherheit

Kostenschutz

- EC2-Test: KOSTENLOS (AWS kostenloses Kontingent) oder ~0,02 $ für 2 Stunden
- Lambda-Test: KOSTENLOS (1 Mio. € requests/month kostenloses Kontingent)
- CloudWatch: KOSTENLOS (10 Alarme, grundlegende Messwerte enthalten)
- Voraussichtliche Gesamtkosten: 0,00$ bis 0,05$ für vollständige Tests

Sicherheitsmerkmale bei diesen Tests

- Auto-termination: Built-in automatische Abschaltung
- Kostenloses Kontingent berechtigt: Verwendet die kleinsten Instance-Typen
- Eingeschränkter Umfang: Minimale, isolierte Testressourcen
- Einfache Bereinigung: Einfache Konsolenschritte, um alles zu entfernen
- Keine Auswirkungen auf die Produktion: Vollständig separate Testumgebung

Richten Sie Ihre ein AWS Konto zum Testen

> **Important**
>
> Infrastrukturressourcen müssen in dem AWS Konto bereitgestellt werden, in dem Sie das primäre Cloud-Konto Ihres DevOps Agent Space erstellt haben. Die spezifische Region spielt keine Rolle.

1. Loggen Sie sich in die AWS Konsole ein: https://console.aws.amazon.com
2. Stellen Sie sicher, dass Sie mit demselben AWS Konto arbeiten, in dem sich Ihr DevOps Agent Space befindet
3. Sie können jede Region für Ihre Testressourcen verwenden

> **Note**
>
> Die 1:1 -Zuordnung zwischen dem Hauptkonto Ihres DevOps Agenten und den Ressourcen der Testumgebung, die Sie erstellen, vereinfacht den Testaufbau. Sie können Ihren DevOps Agent-Bereich ganz einfach um sekundäre Konten erweitern und kontoübergreifende Untersuchungen ermöglichen.

Wählen Sie Ihren Test

Sie können einen der Tests unabhängig voneinander oder beide zusammen ausführen:

Testoption A: EC2-CPU-Kapazitätstest

Zweck: Überprüfen Sie, ob der AWS DevOps Agent in der Lage ist, EC2-Leistungsprobleme zu erkennen und zu untersuchen

Geschätzte Zeit: 5 Minuten Einrichtung + 10 Minuten automatische Ausführung

Schwierigkeit: Vollständig automatisiert (keine manuellen Schritte erforderlich)

Testoption B: Lambda-Fehlerratentest

Zweck: Überprüfung der Fähigkeit des AWS DevOps Agenten, Lambda-Funktionsfehler zu erkennen und zu untersuchen

Geschätzte Zeit: 10 Minuten für die Einrichtung +2 Minuten bis zum Auslösen

Schwierigkeitsgrad: Sehr einfach

Testoption A: EC2-CPU-Kapazitätstest

Schritt 1: CloudFormation Stack für den EC2-Test bereitstellen

Wir verwenden CloudFormation , um unsere Testressourcen zu erstellen, die es dem AWS DevOps Agenten ermöglichen, sie ordnungsgemäß zu verfolgen und zu untersuchen.

1. Navigieren Sie zu CloudFormation:
   a. Suchen Sie in der AWS Konsole nach "CloudFormation" und klicken Sie auf CloudFormation
   b. Klicken Sie auf Stack erstellen > Mit neuen Ressourcen (Standard)
2. Vorlage hochladen:
   a. Erstellen Sie eine neue lokale Datei mit dem Namen `AWS-DevOpsAgent-ec2-test.yaml`
   b. Kopieren Sie diese CloudFormation Vorlage und fügen Sie sie in die Datei ein:
      i.

```
AWSTemplateFormatVersion: '2010-09-09'
Description: 'AWS DevOps Agent EC2 CPU Test Stack'
Parameters:
  VpcId:
    Type: AWS::EC2::VPC::Id
    Description: ID of an existing VPC where the test instance will be launched.
  SubnetId:
    Type: AWS::EC2::Subnet::Id
    Description: ID of an existing subnet within the selected VPC. Choose a
 subnet that routes to an internet gateway if you plan to connect via SSH.
  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:
      GroupDescription: AWS DevOps Agent beta testing security group
      VpcId: !Ref VpcId
      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
# IAM Role for Session Manager access
SSMInstanceRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            Service: ec2.amazonaws.com
          Action: sts:AssumeRole
    ManagedPolicyArns:
      - arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
    Tags:
      - Key: Name
        Value: AWS-DevOpsAgent-Test-SSMRole
      - Key: Purpose
        Value: AWS-DevOpsAgent-Testing
# Instance profile wrapping the SSM role
SSMInstanceProfile:
  Type: AWS::IAM::InstanceProfile
  Properties:
    Roles:
      - !Ref SSMInstanceRole
```

```
  # 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
      SubnetId: !Ref SubnetId
      SecurityGroupIds:
        - !GetAtt TestSecurityGroup.GroupId
      IamInstanceProfile: !Ref SSMInstanceProfile
      InstanceInitiatedShutdownBehavior: terminate
      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: !GetAtt TestSecurityGroup.GroupId

  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. Wählen Sie in der CloudFormation Konsole die Option Vorlagendatei hochladen

d. Klicken Sie auf Datei auswählen

e. Wählen Sie die `AWS-DevOpsAgent-ec2-test.yaml` Datei

f. Klicken Sie auf Weiter

3. Stack konfigurieren:

   a. Name des Stapels: `AWS-DevOpsAgent-EC2-Test`

   b. Parameter:

      i. VpcId: Wählen Sie eine vorhandene VPC aus der Drop-down-Liste aus.

      ii. SubnetId: Wählen Sie ein Subnetz innerhalb der ausgewählten VPC aus. Für den SSH-Zugriff muss das Subnetz zu einem Internet-Gateway weiterleiten, und der Instance muss eine öffentliche IPv4-Adresse zugeordnet sein. Andernfalls ist die `SSHCommand` Ausgabe leer und SSH-Verbindungen werden nicht erfolgreich sein.

      iii. MyIP: Als Standard belassen `0.0.0.0/0` (Sie können dies bei Bedarf später sichern)

   c. Klicken Sie auf Weiter

4. Stack-Optionen konfigurieren:
   a. Behalten Sie die Standardeinstellungen bei und klicken Sie auf Weiter
5. Überprüfen und erstellen
   a. Markieren Sie Ich bestätige, dass dadurch AWS CloudFormation möglicherweise IAM-Ressourcen erstellt werden
   b. Klicken Sie auf Absenden
6. Warten Sie auf den Abschluss:
   a. Die Erstellung des Stapels dauert 3—5 Minuten
   b. Der Status ändert sich von `CREATE_IN_PROGRESS` zu `CREATE_COMPLETE`
   c. Wichtig: Ihre EC2-Instance ist jetzt Teil eines CloudFormation Stacks, der nachverfolgt AWS DevOpsAgent werden kann!

Optional: Sicherer SSH-Zugriff (nur, wenn Sie eine Verbindung zur Instance herstellen möchten)

Überspringen Sie diesen Schritt, wenn Sie nur den automatisierten Test ausführen möchten

1. Suchen Sie die Sicherheitsgruppe:
   a. Gehen Sie in der AWS Konsole zu dem `AWS-DevOpsAgent-EC2-Test` Stack CloudFormationund wählen Sie ihn aus
   b. Öffnen Sie den Tab Outputs und kopieren Sie den Wert von `SecurityGroupId` (beginnt mit`sg-`)
   c. Gehen Sie zu EC2 → Sicherheitsgruppen und fügen Sie die ID in die Suchleiste ein, um die Sicherheitsgruppe zu öffnen
2. SSH-Regel aktualisieren:
   a. Wählen Sie die Sicherheitsgruppe → Registerkarte Regeln für eingehenden Datenverkehr → Regeln für eingehenden Datenverkehr bearbeiten
   b. Suchen Sie die SSH-Regel (Port 22)
   c. Ändern Sie die Quelle von `0.0.0.0/0` zu Ihrer IP: `[YOUR_IP]/32`
   d. Hol dir deine IP von https://whatismyipaddress.com
   e. Klicken Sie auf Regeln speichern

Schritt 2: Warten Sie auf die automatische Testausführung

1. Automatische Testausführung:
   - Der CPU-Stresstest wird automatisch 5 Minuten nach dem Start der Instance gestartet
   - Es ist kein manuelles Eingreifen erforderlich — warten Sie einfach, der Test läuft komplett im Hintergrund
2. Überwachen Sie den Test:
   - Die Instanz startet und bereitet den Test automatisch vor
   - Das Skript wird 5 Minuten lang ausgeführt und generiert eine CPU-Auslastung von **>** 70%
   - CloudWatch Der Alarm sollte innerhalb von insgesamt 8 bis 10 Minuten ausgelöst werden (5 Minuten Verzögerung + 3 bis 5 Minuten für den Alarm)
3. Optional: Manueller Wiederholungslauf (für zusätzliche Tests):
   - Connect zu Ihrer Instance her: EC2-Konsole → → Connect **`AWS-DevOpsAgent-Test-Instance`** → Session Manager
   - Führen Sie den Stresstest erneut durch: `./cpu-stress-test.sh`
   - Perfekt, um AWS DevOpsAgent die Reaktion mehrmals zu testen

Testoption B: Lambda-Fehlerratentest

Schritt 1: CloudFormation Stack für Lambda-Test bereitstellen

1. Navigieren Sie zu CloudFormation:
   a. Gehen Sie in der AWS Konsole zu CloudFormation
   b. Klicken Sie auf Stack erstellen → Mit neuen Ressourcen (Standard)
2. Vorlage hochladen:
   a. Erstellen Sie eine neue lokale Datei mit dem Namen `AWS-DevOpsAgent-lambda-test.yaml`
   b. Kopieren Sie diese CloudFormation Vorlage und fügen Sie sie in die Datei ein:
      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. Wählen Sie in der CloudFormation Konsole die Option Vorlagendatei hochladen
d. Klicken Sie auf Datei auswählen
e. Wählen Sie die `AWS-DevOpsAgent-lambda-test.yaml` Datei
f. Klicken Sie auf Weiter

3. Stack konfigurieren:
   a. Name des Stapels: `AWS-DevOpsAgent-Lambda-Test`
   b. Klicken Sie auf Weiter
4. Stack-Optionen konfigurieren:
   a. Behalten Sie die Standardeinstellungen bei und klicken Sie auf Weiter
5. Überprüfen und erstellen
   a. Markieren Sie Ich bestätige, dass dadurch AWS CloudFormation möglicherweise IAM-Ressourcen erstellt werden
   b. Klicken Sie auf Absenden
6. Warten Sie auf den Abschluss:
   a. Die Erstellung des Stapels dauert 2-3 Minuten
   b. Der Status wird geändert zu `CREATE_COMPLETE`

Schritt 2: Lambda-Fehler auslösen

1. Navigieren Sie zur Lambda-Konsole:
   a. Gehen Sie zur AWS Lambda-Konsole
   b. Finden Sie Ihre Funktion `AWS-DevOpsAgent-test-lambda`
2. Testen Sie die Funktion:
   a. Klicken Sie auf die Registerkarte Test
   b. Klicken Sie auf Neues Ereignis erstellen
   c. Name des Ereignisses: `AWS-DevOpsAgent-test-event`
   d. Verwenden Sie diese JSON-Nutzlast:
      i.
```
{
"test": "AWS DevOpsAgent validation",
"timestamp": "2024-01-01T00:00:00Z"
```

```
}
```

e. Klicken Sie auf Speichern

3. Fehler generieren:
   a. Klicken Sie dreimal auf die Testtaste (warten Sie jeweils 10 Sekunden)
   b. Bei jedem Test wird ein absichtlicher Fehler generiert
   c. CloudWatch Der Alarm sollte innerhalb von 2-3 Minuten ausgelöst werden
   d. AWS DevOpsAgentsollte nun in der Lage sein, den Alarm mit einer Untersuchung in der Operator-App zu erkennen, die Sie als Nächstes einrichten werden.

Bestätigen AWS DevOps Erkennung von Agenten

Schritt 1: CloudWatch Alarme zur Überprüfung der Integrität (optional)

Mit diesem Schritt wird sichergestellt, dass sich die oben genannten Tests jetzt im Alarmzustand befinden.

Für den EC2-Test:

- Gehen Sie in der CloudWatch Konsole zu Alarme
- Warten Sie nach dem Start des Stresstests 3-5 Minuten
- Ihr Alarm sollte Im Alarmzustand angezeigt werden
- Falls immer noch „OK“: Warten Sie weitere 2-3 Minuten (die CloudWatch Messwerte können sich verzögern)

Für den Lambda-Test:

- Alarmanlage überprüfen `AWS-DevOpsAgent-Lambda-Error-Test`
- Sollte innerhalb von 2-3 Minuten nach dem Ausführen der Tests als Alarm angezeigt werden

Schritt 2: Starten Sie eine AWS DevOps Untersuchung durch Agenten

1. Öffnen Sie Ihren AWS DevOps Agenten AgentSpace
2. Klicken Sie auf Admin-Zugriff. Dadurch wird die DevOps Agent Space-Web-App in einem neuen Fenster geöffnet
3. Klicken Sie auf der rechten Seite des Bildschirms auf die Schaltfläche Untersuchung starten

4. Füllen Sie das folgende Formular aus:
   a. Einzelheiten der Untersuchung: Beschreiben Sie die Untersuchung, die Sie durchführen möchten. Geben Sie alle möglichen Details zu den Ermittlungszielen, zu erforschenden Bereichen oder zu relevanten Informationen an.
   b. Ausgangspunkt der Untersuchung: Beschreiben Sie die Informationen, mit denen Sie die Untersuchung beginnen möchten. Sie können einen Alarm, eine Metrik, einen Protokollausschnitt oder etwas anderes angeben, um dem DevOps Agenten einen Ausgangspunkt zu geben, von dem aus er arbeiten kann. Geben Sie in diesem Fall eine Zusammenfassung der Alarme an, die Sie gerade erstellt haben.
   c. Datum und Uhrzeit des Vorfalls (bevorzugt ISO 8601): YYYY-MM-DDTHH:MMZ
   d. Benennen Sie Ihre Untersuchung: Beispiel: `Oncall_investigation_1:2025-10-27`
   e. AWS Konto-ID für den Vorfall
   f. Region, in der sich der Vorfall ereignet hat
   g. Priorität — AWS DevOpsAgent ermöglicht zwei gleichzeitige Untersuchungen. Mit der Priorität können Sie die Reihenfolge festlegen, in der Ihre Untersuchungen ausgeführt werden.
5. Klicken Sie auf Investigate, um die Untersuchung zu starten.
6. Klicken Sie auf Ihre Untersuchung, die im Dashboard aufgeführt ist. Sie werden zum Bildschirm mit den Ermittlungsdetails weitergeleitet, auf dem Sie die detaillierten Schritte sehen können, die der DevOps Agent unternimmt.

Erwartete Ergebnisse

EC2-Testergebnisse:

- Erkennt den EC2-CPU-Alarm
- Identifiziert die Hauptursache: „CPU-Stresstest-Workload“
- Zeigt den Zeitplan an: Stresstest → CPU-Spitze → Alarm
- Bietet Empfehlungen für die Überwachung und Skalierung

Lambda-Testergebnisse:

- Erkennt einen Anstieg der Lambda-Fehlerrate
- Identifiziert die Hauptursache: „Absichtliche Testausnahmen“
- Zeigt die Zeitleiste an: Funktionsaufrufe → Fehler → Alarm

- Bietet Empfehlungen zur Fehlerbehandlung und -überwachung

Anweisungen zur Bereinigung

Bereinigungstest A (EC2-Test)

Automatische Säuberung

- Die Instanz wird nach 2 Stunden automatisch beendet (in die CloudFormation Vorlage integriert)

Manuelle Bereinigung (sofort)

1. CloudFormation Stapel löschen:
   a. Gehe zur CloudFormation Konsole
   b. Wählen Sie `AWS-DevOpsAgent-EC2-Test` Stapel
   c. Klicken Sie auf Löschen
   d. Bestätigen Sie das Löschen
   e. Dadurch werden automatisch alle Ressourcen gelöscht: EC2-Instance, Sicherheitsgruppe, key pair und Alarm CloudWatch

Bereinigungstest B (Lambda-Test)

1. Stapel löschen CloudFormation :
   a. Gehe zur CloudFormation Konsole
   b. Wählen Sie `AWS-DevOpsAgent-Lambda-Test` Stapel
   c. Klicken Sie auf Löschen
   d. Bestätigen Sie das Löschen
   e. Dadurch werden automatisch alle Ressourcen gelöscht: Lambda-Funktion, IAM-Rolle und Alarm CloudWatch

Fehlerbehebung

Häufige Probleme

„Es kann keine Verbindung zur EC2-Instance hergestellt werden“

- Überprüfen Sie die Sicherheitsgruppe: Stellen Sie sicher, dass SSH (Port 22) für Ihre IP geöffnet ist
- Überprüfen Sie die wichtigsten Berechtigungen: Ausführen `chmod 400 AWS-DevOpsAgent-test-key.pem`
- Öffentliche IP überprüfen: Der Instanz muss eine öffentliche IP zugewiesen sein
- Auf Instanz warten: Stellen Sie sicher, dass sich die Instanz im Status „Running“ befindet

„Alarm wird nicht ausgelöst“

- Warten Sie auf CloudWatch Metriken: Es kann 2-5 Minuten dauern, bis Metriken angezeigt werden
- Überprüfen Sie die CPU-Auslastung: Stellen Sie eine SSH-Verbindung zur Instanz her und führen Sie die Ausführung aus`top`, um zu überprüfen, ob die CPU-
- Überprüfen Sie den Stresstest: Führen Sie `ps aux | grep yes` ihn aus, um festzustellen, ob Ladeprozesse ausgeführt werden
- Verlängerte Wartezeit: Manchmal dauert es bis zu 7-8 Minuten, bis der erste Alarm ausgelöst wird

Testvalidierung

Ihr AWS DevOp Agententest ist erfolgreich, wenn:

Technische Validierung

- Genauigkeit der Untersuchung: Die Ergebnisse des EC2-Tests sollten korrekt darauf hinweisen, dass der Alarm aufgrund einer CPU-Last ausgelöst wurde. Das Ergebnis des Lambda-Tests sollte darauf hindeuten, dass es sich um einen vorsätzlichen Fehler handelte.
- Genauigkeit des Zeitplans: Die korrekte Reihenfolge der Ereignisse wird angezeigt
- Qualität der Empfehlungen: Es wurden umsetzbare Vorschläge gemacht

Erste Schritte mit AWS DevOps Agent using AWS CDK

-Übersicht

Diese Anleitung zeigt Ihnen, wie Sie das AWS Cloud Development Kit (AWS CDK) verwenden, um AWS DevOps Agentenressourcen zu erstellen und bereitzustellen. Die AWS CDK-Anwendung automatisiert die Erstellung eines Agentenbereichs, AWS Identity and Access Management (IAM) -Rollen, einer Operator-App und AWS Kontozuordnungen durch. AWS CloudFormation

Der AWS CDK-Ansatz automatisiert die im CLI Onboarding-Leitfaden beschriebenen manuellen Schritte, indem alle erforderlichen Ressourcen als Infrastruktur als Code definiert werden.

AWS DevOps Der Agent ist in den folgenden 6 AWS Regionen verfügbar: USA Ost (Nord-Virginia), USA West (Oregon), Asien-Pazifik (Sydney), Asien-Pazifik (Tokio), Europa (Frankfurt) und Europa (Irland). Weitere Informationen zu den unterstützten Regionen finden Sie unterthe section called “Unterstützte Regionen”.

Voraussetzungen

Stellen Sie vor dem Beginn sicher, dass Sie über Folgendes verfügen:

- AWS Die Befehlszeilenschnittstelle (AWS CLI) wurde mit den entsprechenden Anmeldeinformationen installiert und konfiguriert
- Node.js Version 18 oder höher
- AWS CDK-Befehlszeilenschnittstelle (CLI) ist global installiert. Führen Sie den folgenden Befehl aus, um die AWS CDK-CLI zu installieren:

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

- Ein AWS Konto für das (primäre) Überwachungskonto
- (Optional) Ein zweites AWS Konto, wenn Sie eine kontoübergreifende Überwachung einrichten möchten

Was dieser Leitfaden behandelt

Dieser Leitfaden ist in zwei Teile gegliedert:

- Teil 1 — Stellen Sie einen Agentenbereich mit einer Operator-App und einer AWS Verknüpfung in Ihrem Monitoring-Konto bereit. Nachdem Sie diesen Teil abgeschlossen haben, kann der Agent Probleme in diesem Konto überwachen.
- Teil 2 (optional) — Fügen Sie eine AWS Quellenzuordnung für ein Dienstkonto hinzu und stellen Sie eine kontoübergreifende IAM-Rolle für dieses Konto bereit. Diese Konfiguration ermöglicht es dem Agentenbereich, Ressourcen kontenübergreifend zu überwachen.

Ressourcen wurden erstellt

Teil 1: DevOpsAgentStack (Überwachungskonto)

- IAM-Rolle (`DevOpsAgentRole-AgentSpace`) — Wird vom DevOps Agent-Dienst zur Überwachung des Kontos übernommen. Beinhaltet die `AIDevOpsAgentAccessPolicy` verwaltete Richtlinie und eine Inline-Richtlinie, die die Erstellung der serviceverknüpften Resource Explorer-Rolle ermöglicht.
- IAM-Rolle (`DevOpsAgentRole-WebappAdmin`) — Operator-App-Rolle mit der `AIDevOpsOperatorAppAccessPolicy` verwalteten Richtlinie für Agentenoperationen.
- Agentenbereich (`MyCDKAgentSpace`) — Der zentrale Agentenbereich, der mithilfe der `AWS::DevOpsAgent::AgentSpace` CloudFormation Ressource erstellt wurde. Beinhaltet die Konfiguration der Bediener-App.
- Zuordnung (AWS Monitor) — Verknüpft das Überwachungskonto mithilfe der `AWS::DevOpsAgent::Association` CloudFormation Ressource mit dem Agentenbereich.
- Zuordnung (AWS Quelle) — (Optional) Verknüpft das Dienstkonto mit dem Agentenbereich für die kontenübergreifende Überwachung.

Teil 2: ServiceStack (Dienstkonto, optional)

- IAM-Rolle (`DevOpsAgentRole-SecondaryAccount`) — Kontoübergreifende Rolle mit festem Namen. Wird vom Agent-Bereich im Monitoring-Konto als vertrauenswürdig eingestuft. Beinhaltet die `AIDevOpsAgentAccessPolicy` verwaltete Richtlinie und eine Inline-Richtlinie, die die Erstellung der mit dem Resource Explorer-Dienst verknüpften Rolle ermöglicht.
- Lambda-Funktion (`echo-service`) — Ein einfacher Beispieldienst, der Eingabeereignisse zurückgibt.

Einrichtung

Schritt 1: Klonen Sie das Beispiel-Repository

Führen Sie die folgenden Befehle aus, um das Repository zu klonen und zum Projektverzeichnis zu wechseln:

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

Schritt 2: Abhängigkeiten installieren

Führen Sie den folgenden Befehl aus, um die Projektabhängigkeiten zu installieren:

```
npm install
```

Teil 1: Stellen Sie den Agentenbereich bereit

In diesem Abschnitt erstellen Sie den Agentenbereich, die IAM-Rollen, die Operator-App und eine AWS Zuordnung in Ihrem Monitoring-Konto.

Schritt 1: Konfigurieren Sie die ID des Überwachungskontos

Öffnen Sie Ihre Überwachungskonto-ID `lib/constants.ts` und legen Sie sie fest:

Das folgende Beispiel zeigt die zu aktualisierende Konstante:

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

Schritt 2: Bootstrap für die AWS CDK-Umgebung

Wenn Sie das AWS CDK in Ihrem Monitoring-Konto nicht gebootet haben, führen Sie den folgenden Befehl aus:

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

Schritt 3: Erstellen und Bereitstellen

Führen Sie die folgenden Befehle aus, um den TypeScript Code zu erstellen und den Stack bereitzustellen:

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

Schritt 4: Notieren Sie die Stack-Ausgaben

Nach Abschluss der Bereitstellung druckt das AWS CDK die Stack-Ausgaben. Notieren Sie sich diese Werte für die spätere Verwendung.

Das folgende Beispiel zeigt die erwartete Ausgabe:

```
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
```

Wenn Sie planen, Teil 2 abzuschließen, speichern Sie den `AgentSpaceArn` Wert. Sie benötigen es, um den Dienstkontenstapel zu konfigurieren.

Schritt 5: Überprüfen Sie die Bereitstellung

Führen Sie den folgenden AWS CLI-Befehl aus, um zu überprüfen, ob der Agent-Bereich erfolgreich erstellt wurde:

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

Zu diesem Zeitpunkt wird Ihr Agentenbereich bereitgestellt, wobei die Operator-App aktiviert und Ihr Monitoring-Konto verknüpft ist. Der Agent kann Probleme in diesem Konto überwachen.

Teil 2 (optional): Kontenübergreifende Überwachung hinzufügen

In diesem Abschnitt erweitern Sie das Setup, sodass Ihr Agentenbereich Ressourcen in einem zweiten AWS Konto (dem Dienstkonto) überwachen kann. Dies beinhaltet zwei Aktionen:

1. Hinzufügen einer AWS Quellzuordnung in der DevOpsAgentStack , die auf das Dienstkonto verweist.

2. Bereitstellung des im ServiceStack Dienstkonto mit einer IAM-Rolle, die dem Agentenbereich vertraut.

> **Important**
>
> Sie müssen Teil 1 abschließen, bevor Sie fortfahren können. Das ServiceStack erfordert das `AgentSpaceArn` aus der DevOpsAgentStack Bereitstellungsausgabe.

Schritt 1: Konfigurieren Sie die Dienstkonto-ID

Öffnen Sie Ihre Dienstkonto-ID `lib/constants.ts` und legen Sie sie fest:

Das folgende Beispiel zeigt die zu aktualisierende Konstante:

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

Der DevOpsAgentStack erstellt mithilfe dieser Konto-ID eine AWS Quellzuordnung. Wenn Sie das bereitgestellt haben, DevOpsAgentStack bevor Sie diesen Wert festgelegt haben, führen Sie die Bereitstellung erneut durch, um die Zuordnung zu erstellen:

Führen Sie die folgenden Befehle aus, um die Bereitstellung erneut durchzuführen:

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

Schritt 2: Den ARN für den Agentenbereich einrichten

Kopieren Sie den `AgentSpaceArn` Wert aus der DevOpsAgentStack Ausgabe (Teil 1, Schritt 4) und geben Sie ihn ein`lib/constants.ts`:

Das folgende Beispiel zeigt die zu aktualisierende Konstante:

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

Der ServiceStack verwendet diesen Wert, um den Geltungsbereich der Vertrauensrichtlinie für die sekundäre Kontorolle festzulegen. Der ServiceStack wird nur synthetisiert, wenn dieser Wert festgelegt ist.

Schritt 3: Bootstrap für das Dienstkonto

Wenn Sie das AWS CDK in Ihrem Dienstkonto nicht gebootet haben, führen Sie den folgenden Befehl aus:

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

Schritt 4: Stellen Sie das bereit ServiceStack

Führen Sie die folgenden Befehle aus, um das ServiceStack mithilfe der Anmeldeinformationen für das Dienstkonto zu erstellen und bereitzustellen:

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

Dadurch werden die folgenden Ressourcen im Dienstkonto erstellt:

- Eine IAM-Rolle (`DevOpsAgentRole-SecondaryAccount`), die dem Agentenbereich im Überwachungskonto vertraut
- Eine Echo-Lambda-Funktion (`echo-service`) als Beispieldienst

Schritt 5: Überprüfen Sie die Bereitstellung

Um zu überprüfen, ob die Lambda-Funktion erfolgreich bereitgestellt wurde, führen Sie die folgenden Befehle aus, um den Echo-Service zu testen:

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

Fehlerbehebung

In diesem Abschnitt werden häufig auftretende Probleme und deren Behebung beschrieben.

CloudFormation Ressourcentyp wurde nicht gefunden

- Stellen Sie sicher, dass Sie die Bereitstellung in einem durchführenthe section called “Unterstützte Regionen”.
- Vergewissern Sie sich, dass Ihre AWS CLI mit den entsprechenden Berechtigungen konfiguriert ist.

Die Erstellung der IAM-Rolle ist fehlgeschlagen

- Stellen Sie sicher, dass Ihre Bereitstellungsrolle über Berechtigungen zum Erstellen von IAM-Rollen verfügt.
- Vergewissern Sie sich, dass die Bedingungen der Vertrauensrichtlinie mit Ihrer Konto-ID übereinstimmen.

Die kontoübergreifende Bereitstellung schlägt mit der Meldung „Die Rolle im Zielkonto konnte nicht übernommen werden“ fehl

- Jeder Stack muss mit Anmeldeinformationen für das Zielkonto bereitgestellt werden. Verwenden Sie das `--profile` Flag, um das richtige AWS CLI-Profil anzugeben.
- Stellen Sie sicher, dass das AWS CDK im Zielkonto gebootet wurde.

Verzögerungen bei der IAM-Übertragung

- Die Übertragung von IAM-Rollenänderungen kann einige Minuten dauern. Wenn die Erstellung des Agentenbereichs unmittelbar nach der Rollenerstellung fehlschlägt, warten Sie einige Minuten und führen Sie die Bereitstellung erneut durch.

Bereinigen

Um alle Ressourcen zu entfernen, zerstören Sie die Stapel in umgekehrter Reihenfolge.

Führen Sie die folgenden Befehle aus, um die Stapel zu zerstören:

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

Warnung: Durch diese Aktion werden Ihr Agentenbereich und alle zugehörigen Daten dauerhaft gelöscht. Diese Aktion kann nicht rückgängig gemacht werden. Stellen Sie sicher, dass Sie alle wichtigen Informationen gesichert haben, bevor Sie fortfahren.

Sicherheitsüberlegungen

- Die AWS CDK-Anwendung erstellt IAM-Rollen mit Vertrauensrichtlinien, die es nur dem `aidevops.amazonaws.com` Dienstprinzipal ermöglichen, diese Rollen zu übernehmen.
- Zu den Vertrauensrichtlinien gehören Bedingungen, die den Zugriff auf Ihr bestimmtes AWS Konto und den ARN Ihres Agentenbereichs einschränken.
- Alle Richtlinien folgen dem Prinzip der geringsten Rechte. Überprüfen Sie die IAM-Richtlinien und passen Sie sie an die Sicherheitsanforderungen Ihres Unternehmens an.
- Die kontoübergreifende Rolle (`DevOpsAgentRole-SecondaryAccount`) verwendet einen festen Namen und ist auf einen bestimmten Agent-Space-ARN ARN.

Nächste Schritte

Nachdem Sie Ihren AWS DevOps Agenten mithilfe des CDK bereitgestellt haben: AWS

1. Erfahren Sie im DevOps Agent-Benutzerhandbuch mehr über den gesamten Funktionsumfang des AWS DevOps Agenten.
2. Erwägen Sie, die AWS CDK-Implementierung in Ihre CI/CD Pipelines für ein automatisiertes Infrastrukturmanagement zu integrieren.

Weitere Ressourcen

- AWS DevOps Benutzerhandbuch für Agenten
- Beispiel für ein CDK-Repository auf der Website GitHub
- CLI Onboarding-Leitfaden

Erste Schritte mit AWS DevOps Agent using AWS CloudFormation

-Übersicht

In diesem Handbuch erfahren Sie, wie Sie mithilfe von AWS CloudFormation Vorlagen AWS DevOps Agentenressourcen erstellen und bereitstellen. Die Vorlagen automatisieren die Erstellung eines Agentenbereichs, AWS Identitäts- und Zugriffsmanagement-Rollen (IAM), einer Operator-App und AWS Kontozuordnungen als Infrastruktur als Code.

Der CloudFormation Ansatz automatisiert die im CLI-Onboarding-Leitfaden beschriebenen manuellen Schritte, indem alle erforderlichen Ressourcen in deklarativen YAML-Vorlagen definiert werden.

AWS DevOps Der Agent ist in den folgenden 6 AWS Regionen verfügbar: USA Ost (Nord-Virginia), USA West (Oregon), Asien-Pazifik (Sydney), Asien-Pazifik (Tokio), Europa (Frankfurt) und Europa (Irland). Weitere Informationen zu den unterstützten Regionen finden Sie unterthe section called “Unterstützte Regionen”.

Voraussetzungen

Stellen Sie vor dem Beginn sicher, dass Sie über Folgendes verfügen:

- AWS Die Befehlszeilenschnittstelle (AWS CLI) wurde mit den entsprechenden Anmeldeinformationen installiert und konfiguriert
- Berechtigungen zum Erstellen von IAM-Rollen und CloudFormation -Stacks
- Ein AWS Konto für das (primäre) Überwachungskonto
- (Optional) Ein zweites AWS Konto, wenn Sie eine kontoübergreifende Überwachung einrichten möchten

Worum geht es in diesem Leitfaden

Dieser Leitfaden ist in zwei Teile gegliedert:

- Teil 1 — Stellen Sie einen Agentenbereich mit einer Operator-App und einer AWS Verknüpfung in Ihrem Monitoring-Konto bereit. Nachdem Sie diesen Teil abgeschlossen haben, kann der Agent Probleme in diesem Konto überwachen.
- Teil 2 (optional) — Stellen Sie eine kontoübergreifende IAM-Rolle in einem sekundären Konto bereit und fügen Sie eine AWS Quellenzuordnung hinzu. Diese Konfiguration ermöglicht es dem Agentenbereich, Ressourcen kontenübergreifend zu überwachen.

Teil 1: Stellen Sie den Agentenbereich bereit

In diesem Abschnitt erstellen Sie eine CloudFormation Vorlage, die den Agentenbereich, die IAM-Rollen, die Operator-App und eine AWS Zuordnung in Ihrem Monitoring-Konto bereitstellt.

Schritt 1: Erstellen Sie die Vorlage CloudFormation

Speichern Sie die folgende Vorlage unter`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
```

Schritt 2: Stellen Sie den Stack bereit

Führen Sie den folgenden Befehl aus, um den Stack bereitzustellen. `<REGION>`Ersetzen Sie durch ein the section called “Unterstützte Regionen” (z. B.`us-east-1`).

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

Schritt 3: Notieren Sie die Stack-Ausgaben

Führen Sie nach Abschluss der Bereitstellung den folgenden Befehl aus, um die Stack-Ausgaben abzurufen. Notieren Sie sich diese Werte für die spätere Verwendung.

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

Das folgende Beispiel zeigt die erwartete Ausgabe:

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

Wenn Sie planen, Teil 2 abzuschließen, speichern Sie den `AgentSpaceArn` Wert. Sie benötigen es, um die kontoübergreifende Rolle zu konfigurieren.

Schritt 4: Überprüfen Sie die Bereitstellung

Führen Sie den folgenden AWS CLI-Befehl aus, um zu überprüfen, ob der Agent-Bereich erfolgreich erstellt wurde:

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

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

Zu diesem Zeitpunkt wird Ihr Agentenbereich bereitgestellt, wobei die Operator-App aktiviert und Ihr Monitoring-Konto verknüpft ist. Der Agent kann Probleme in diesem Konto überwachen.

Teil 2 (optional): Kontenübergreifende Überwachung hinzufügen

In diesem Abschnitt erweitern Sie das Setup, sodass Ihr Agentenbereich Ressourcen in einem zweiten AWS Konto (dem Dienstkonto) überwachen kann. Dies beinhaltet zwei Aktionen:

1. Bereitstellung einer IAM-Rolle in dem Dienstkonto, das dem Agentenbereich vertraut.
2. Hinzufügen einer AWS Quellzuordnung zum Überwachungskonto, die auf das Dienstkonto verweist.

> Hinweis: Sie müssen Teil 1 abschließen, bevor Sie fortfahren können. Für die Vorlage für das Dienstkonto sind die Stack-Ausgaben `AgentSpaceArn` aus Teil 1 erforderlich.

Schritt 1: Erstellen Sie die Vorlage für das Dienstkonto

Speichern Sie die folgende Vorlage unter`devops-agent-service-account.yaml`. Diese Vorlage erstellt eine kontoübergreifende IAM-Rolle im sekundären Konto.

```
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
```

Schritt 2: Stellen Sie den Dienstkontenstapel bereit

Führen Sie mit den Anmeldeinformationen für das Dienstkonto den folgenden Befehl aus:

```
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>
```

Schritt 3: Fügen Sie die AWS Quellzuordnung hinzu

Wechseln Sie zurück zum Überwachungskonto und erstellen Sie eine AWS Quellenzuordnung. Sie können dies tun, indem Sie einen separaten Stapel erstellen oder die ursprüngliche Vorlage aktualisieren. Im folgenden Beispiel wird eine eigenständige Vorlage verwendet.

Speichern Sie die folgende Vorlage als`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
```

Stellen Sie den Zuordnungsstapel mithilfe der Anmeldeinformationen für das Überwachungskonto bereit:

```
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>
```

Verifizierung

Überprüfen Sie Ihr Setup, indem Sie die folgenden AWS CLI-Befehle ausführen:

```
# 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>
```

Fehlerbehebung

In diesem Abschnitt werden häufig auftretende Probleme und deren Behebung beschrieben.

CloudFormation Ressourcentyp wurde nicht gefunden

- Stellen Sie sicher, dass Sie die Bereitstellung in einem durchführenthe section called “Unterstützte Regionen”.
- Vergewissern Sie sich, dass Ihre AWS CLI mit den entsprechenden Berechtigungen konfiguriert ist.

Die Erstellung der IAM-Rolle ist fehlgeschlagen

- Stellen Sie sicher, dass Ihre Bereitstellungsanmeldedaten berechtigt sind, IAM-Rollen mit benutzerdefinierten Namen () `CAPABILITY_NAMED_IAM` zu erstellen.
- Vergewissern Sie sich, dass die Bedingungen der Vertrauensrichtlinie mit Ihrer Konto-ID übereinstimmen.

Die kontoübergreifende Bereitstellung schlägt fehl

- Jeder Stapel muss mit Anmeldeinformationen für das Zielkonto bereitgestellt werden. Verwenden Sie das `--profile` Flag, um das richtige AWS CLI-Profil anzugeben.
- Stellen Sie sicher, dass der `AgentSpaceArn` Parameter genau mit dem ARN aus den Stackausgaben von Teil 1 übereinstimmt.

Verzögerungen bei der IAM-Übertragung

- Die Übertragung von IAM-Rollenänderungen kann einige Minuten dauern. Wenn die Erstellung des Agentenbereichs unmittelbar nach der Rollenerstellung fehlschlägt, warten Sie einige Minuten und führen Sie die Bereitstellung erneut durch.

Bereinigen

Um alle Ressourcen zu entfernen, löschen Sie die Stapel in umgekehrter Reihenfolge.

Warnung: Durch diese Aktion werden Ihr Agentenbereich und alle zugehörigen Daten dauerhaft gelöscht. Diese Aktion kann nicht rückgängig gemacht werden. Stellen Sie sicher, dass Sie alle wichtigen Informationen gesichert haben, bevor Sie fortfahren.

Führen Sie die folgenden Befehle aus, um die Stapel zu löschen:

```
# 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>
```

Nächste Schritte

Nachdem Sie Ihren AWS DevOps Agenten bereitgestellt haben, indem AWS CloudFormation Sie:

- Informationen zum Herstellen zusätzlicher Integrationen finden Sie unterKonfiguration von Funktionen für AWS DevOps Agent.
- Weitere Informationen zu den Fähigkeiten und Fertigkeiten von Agenten finden Sie unterthe section called “DevOps Fähigkeiten der Agenten”.
- Weitere Informationen zur Web-App für Operatoren finden Sie unterthe section called “Was ist eine DevOps Agent-Web-App?”.

Erste Schritte mit AWS DevOps Agent using Terraform

-Übersicht

In diesem Handbuch erfahren Sie, wie Sie mit Terraform Agentenressourcen erstellen und bereitstellen. AWS DevOps Die Terraform-Konfiguration automatisiert die Erstellung eines Agentenbereichs, von IAM-Rollen, einer Operator-App und Kontozuordnungen. AWS

Der Terraform-Ansatz automatisiert die im CLI-Onboarding-Leitfaden beschriebenen manuellen Schritte, indem alle erforderlichen Ressourcen als Infrastruktur als Code definiert werden.

AWS DevOps Der Agent ist in den folgenden 6 AWS Regionen verfügbar: USA Ost (Nord-Virginia), USA West (Oregon), Asien-Pazifik (Sydney), Asien-Pazifik (Tokio), Europa (Frankfurt) und Europa (Irland). Weitere Informationen zu den unterstützten Regionen finden Sie unterthe section called “Unterstützte Regionen”.

Voraussetzungen

Stellen Sie vor dem Beginn sicher, dass Sie über das Folgende verfügen:

- Terraform >= 1.0 ist installiert
- AWS CLI wurde mit den entsprechenden Anmeldeinformationen installiert und konfiguriert
- Ein AWS Konto für das (primäre) Überwachungskonto
- (Optional) Ein zweites AWS Konto, wenn Sie eine kontoübergreifende Überwachung einrichten möchten

Was dieser Leitfaden behandelt

Dieser Leitfaden ist in zwei Teile gegliedert:

- Teil 1 — Stellen Sie einen Agentenbereich mit einer Operator-App und einer AWS Verknüpfung in Ihrem Monitoring-Konto bereit. Nach Abschluss dieses Teils kann der Agent Probleme in diesem Konto überwachen.
- Teil 2 (optional) — Fügen Sie eine AWS Quellzuordnung für ein Dienstkonto hinzu und stellen Sie eine kontoübergreifende IAM-Rolle sowie ein Echolambda für dieses Konto bereit. Auf diese Weise kann der Agentenbereich Ressourcen kontenübergreifend überwachen.

Ressourcen erstellt

Teil 1: Überwachungskonto

- IAM-Rolle (`DevOpsAgentRole-AgentSpace-*`) — Wird vom DevOps Agent-Dienst zur Überwachung des Kontos übernommen. Beinhaltet die `AIDevOpsAgentAccessPolicy` verwaltete Richtlinie und eine Inline-Richtlinie, die die Erstellung der serviceverknüpften Resource Explorer-Rolle ermöglicht.
- IAM-Rolle (`DevOpsAgentRole-WebappAdmin-*`) — Operator-App-Rolle mit der `AIDevOpsOperatorAppAccessPolicy` verwalteten Richtlinie für Agentenoperationen.
- Agentenbereich (konfigurierbarer Name) — Der zentrale Agentenbereich, der mithilfe der `awscc_devopsagent_agent_space` Ressource erstellt wurde. Beinhaltet die Konfiguration der Bediener-App.
- Zuordnung (AWS Monitor) — Verknüpft das Überwachungskonto mithilfe der `awscc_devopsagent_association` Ressource mit dem Agentenbereich.

- Zuordnung (AWS Quelle) — (Optional) Verknüpft das Dienstkonto mit dem Agentenbereich für die kontenübergreifende Überwachung.

Teil 2: Dienstkonto (optional)

- IAM-Rolle (`DevOpsAgentRole-SecondaryAccount-TF`) — Kontoübergreifende Rolle mit festem Namen. Wird vom Agent-Bereich im Monitoring-Konto als vertrauenswürdig eingestuft. Beinhaltet die `AIDevOpsAgentAccessPolicy` verwaltete Richtlinie und eine Inline-Richtlinie, die die Erstellung der mit dem Resource Explorer-Dienst verknüpften Rolle ermöglicht.
- Lambda-Funktion (`echo-service-tf`) — Ein einfacher Beispieldienst, der Eingabeereignisse zurückgibt.

Einrichtung

Schritt 1: Klonen Sie das Beispiel-Repository

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

Schritt 2: Variablen konfigurieren

Kopieren Sie die Beispielvariablendatei und passen Sie sie an Ihre Umgebung an:

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

Bearbeiten Sie `terraform.tfvars` mit dem Namen und der Beschreibung Ihres Agentenbereichs:

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

Teil 1: Stellen Sie den Agentenbereich bereit

In diesem Abschnitt erstellen Sie den Agentenbereich, die IAM-Rollen, die Operator-App und eine AWS Zuordnung in Ihrem Monitoring-Konto.

Schritt 1: Automatisierte Bereitstellung (empfohlen)

Verwenden Sie das bereitgestellte Bereitstellungsskript für eine optimierte Einrichtung:

```
./deploy.sh
```

Dieses Skript führt automatisch Folgendes durch:

- Prüft die Voraussetzungen (Terraform, AWS CLI, Anmeldeinformationen)
- Erstellt bei Bedarf `terraform.tfvars` anhand eines Beispiels
- Initialisiert, validiert, plant und wendet Terraform an

Alternativ, wenn Sie eine manuelle Steuerung bevorzugen:

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

Geben Sie ein`yes`, wenn Sie aufgefordert werden, die Bereitstellung zu bestätigen.

Schritt 2: Notieren Sie die Ausgaben

Nach Abschluss der Bereitstellung druckt Terraform die Ausgaben. Notieren Sie sich diese Werte für die spätere Verwendung:

```
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"
```

Wenn Sie Teil 2 abschließen möchten, speichern Sie den `agent_space_arn` Wert. Sie benötigen es, um die Ressourcen des Dienstkontos zu konfigurieren.

Schritt 3: Überprüfen Sie die Bereitstellung

Führen Sie das Überprüfungsskript nach der Bereitstellung aus:

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

Oder verwenden Sie die AWS CLI, um zu überprüfen, ob der Agentenbereich erfolgreich erstellt wurde:

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

Zu diesem Zeitpunkt wird Ihr Agentenbereich bereitgestellt, wobei die Operator-App aktiviert und Ihr Monitoring-Konto verknüpft ist. Der Agent kann Probleme in diesem Konto überwachen.

Teil 2 (optional): Kontenübergreifende Überwachung hinzufügen

In diesem Abschnitt erweitern Sie das Setup, sodass der Agentenbereich Ressourcen in einem zweiten AWS Konto (dem Dienstkonto) überwachen kann. Dies beinhaltet zwei Aktionen:

1. Hinzufügen einer AWS Quellzuordnung, die auf das Dienstkonto verweist.
2. Bereitstellung einer kontoübergreifenden IAM-Rolle und einer Echo-Lambda-Funktion im Dienstkonto.

> **Important**
>
> Sie müssen Teil 1 abschließen, bevor Sie fortfahren können. Für die Ressourcen des Dienstkontos ist die Ausgabe `agent_space_arn` aus der Bereitstellung von Teil 1 erforderlich.

Schritt 1: Konfigurieren Sie die Dienstkonto-ID

Geben Sie `terraform.tfvars` unter Ihre Dienstkonto-ID ein:

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

Schritt 2: Den ARN für den Agentenbereich einrichten

Kopieren Sie den `agent_space_arn` Wert aus der Ausgabe von Teil 1 (Schritt 2) und geben Sie ihn ein`terraform.tfvars`:

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

Die Ressourcen des Dienstkontos verwenden diesen Wert, um den Geltungsbereich der Vertrauensrichtlinie für die sekundäre Kontorolle festzulegen. Diese Ressourcen werden nur erstellt, wenn dieser Wert festgelegt ist.

Schritt 3: Konfigurieren Sie den Anbieter `aws.service`

Konfigurieren Sie unter den `aws.service` Anbieteralias mit den Anmeldeinformationen für das Dienstkonto. `main.tf` Sie können entweder ein benanntes Profil oder eine Rolle übernehmen verwenden:

Ein Profil verwenden:

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

Oder mit „Rolle übernehmen“:

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

Schritt 4: Bereitstellen

Wenden Sie die aktualisierte Konfiguration an:

```
terraform apply
```

Dadurch werden die folgenden Ressourcen im Dienstkonto erstellt:

- Eine IAM-Rolle (`DevOpsAgentRole-SecondaryAccount-TF`), die dem Agentenbereich im Überwachungskonto vertraut

- Eine Echo-Lambda-Funktion (`echo-service-tf`) als Beispieldienst

Sie erstellt auch eine AWS Quellzuordnung im Überwachungskonto, die das Dienstkonto verknüpft.

Schritt 5: Überprüfen Sie die Bereitstellung

Testen Sie den Echo-Service, um sicherzustellen, dass die Lambda-Funktion erfolgreich bereitgestellt wurde:

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

Fehlerbehebung

Verzögerungen bei der IAM-Übertragung

- Die Konfiguration umfasst einen Zeitraum von 30 Sekunden `time_sleep` zwischen der Erstellung der IAM-Rolle und der Erstellung des Agent Space. Der DevOps Agent-Dienst validiert die Vertrauensrichtlinie der Operatorrolle bei der Erstellung des Agent Space. Dies kann fehlschlagen, wenn IAM nicht vollständig weitergegeben wurde. Wenn weiterhin Fehler in der Vertrauensrichtlinie angezeigt werden, warten Sie eine Minute und führen Sie den Vorgang `terraform apply` erneut aus. Die IAM-Rollen sind bereits vorhanden und die Anwendung setzt dort fort, wo sie aufgehört hat.

Fehler bei der Genehmigung

- Stellen Sie sicher, dass Ihre AWS Anmeldeinformationen über die erforderlichen IAM-Berechtigungen zum Erstellen von Rollen und Richtlinien verfügen.
- Vergewissern Sie sich, dass die Bedingungen der Vertrauensrichtlinie mit Ihrer Konto-ID übereinstimmen.

Die kontoübergreifende Bereitstellung schlägt fehl

- Der `aws.service` Anbieter muss mit Anmeldeinformationen für das Dienstkonto konfiguriert sein. Verwenden Sie ein benanntes Profil oder einen Block „Rolle übernehmen“.
- Stellen Sie sicher, dass der `agent_space_arn` Wert mit dem ARN aus der Ausgabe von Teil 1 übereinstimmt.

Der Terraform-Ressourcentyp wurde nicht gefunden

- Stellen Sie sicher, dass Sie die `awscc` Anbieterversion `~> 1.0` oder höher haben. Für die `awscc_devopsagent_association` Ressourcen `awscc_devopsagent_agent_space` und ist der AWS Cloud Control-Anbieter erforderlich.

Bereinigen

Um alle Ressourcen zu entfernen, löschen Sie sie in umgekehrter Reihenfolge, falls Sie Teil 2 bereitgestellt haben:

```
./cleanup.sh
```

Oder manuell:

```
terraform destroy
```

Warnung: Dadurch werden Ihr Agentenbereich und alle zugehörigen Daten dauerhaft gelöscht. Stellen Sie sicher, dass Sie alle wichtigen Informationen gesichert haben, bevor Sie fortfahren.

Sicherheitsüberlegungen

- Die Terraform-Konfiguration erstellt IAM-Rollen mit Vertrauensrichtlinien, die es nur dem `aidevops.amazonaws.com` Dienstprinzipal ermöglichen, sie zu übernehmen.
- Zu den Vertrauensrichtlinien gehören Bedingungen, die den Zugriff auf Ihr bestimmtes AWS Konto und den ARN Ihres Agentenbereichs einschränken.
- Alle Richtlinien folgen dem Prinzip der geringsten Rechte. Überprüfen Sie die IAM-Richtlinien und passen Sie sie an die Sicherheitsanforderungen Ihres Unternehmens an.
- Die kontoübergreifende Rolle (`DevOpsAgentRole-SecondaryAccount-TF`) verwendet einen festen Namen und ist auf einen bestimmten Agent-Space-ARN ARN.

Nächste Schritte

Nachdem Sie Ihren AWS DevOps Agenten mit Terraform bereitgestellt haben:

1. Erfahren Sie im DevOps Agent-Benutzerhandbuch mehr über den gesamten Funktionsumfang des AWS DevOps Agenten.
2. Erwägen Sie, die Terraform-Bereitstellung in Ihre CI/CD Pipelines zu integrieren, um das Infrastrukturmanagement zu automatisieren.

Weitere Ressourcen

- AWS DevOps Benutzerhandbuch für Agenten
- Beispiel für ein Terraform-Repository
- CLI Onboarding-Leitfaden

Mit dem DevOps Agenten arbeiten

Mit dem DevOps Agenten arbeiten

AWS DevOps Der Agent steht Ihrem Betriebsteam während des gesamten Lebenszyklus eines Vorfalls zur Seite — von der Erkennung über die Untersuchung bis hin zur Behebung und Vorbeugung. In den folgenden Themen wird beschrieben, wie Sie DevOps Agent verwenden, um die einzelnen Phasen dieses Lebenszyklus zu verwalten.

Autonome Reaktion auf Vorfälle

Wenn ein Vorfall erkannt wird — sei es durch eine integrierte Integration in Ihr Ticketsystem, einen Webhook aus Ihren Überwachungstools oder einen manuellen Auslöser — leitet der DevOps Agent automatisch eine Untersuchung ein. Der Agent analysiert Metriken, Protokolle, Ablaufverfolgungen, Codeänderungen und den Bereitstellungsverlauf, um die Ursache zu ermitteln und einen Plan zur Schadensbegrenzung vorzuschlagen. Wenn Sie zusätzliche Hilfe benötigen, können Sie über die DevOps Agent Space-Web-App direkt an den AWS Support weiterleiten. Diese teilt den Untersuchungskontext automatisch mit den Support-Technikern, sodass Sie nicht wiederholen müssen, was der Agent bereits gefunden hat. Weitere Informationen finden Sie unter the section called “Autonome Reaktion auf Vorfälle”.

Aufgaben auf Abruf DevOps

Sie können zu jedem Zeitpunkt des Incident-Lebenszyklus über eine Chat-Oberfläche mit dem DevOps Agenten interagieren. Stellen Sie in natürlicher Sprache Fragen zu Ihren AWS Ressourcen, dem Systemzustand, dem Alarmstatus und dem Bereitstellungsverlauf. Der Chat ist kontextsensitiv — wenn Sie sich eine bestimmte Untersuchung ansehen, können Sie den Agenten anweisen, bestimmte Hypothesen zu untersuchen, sich auf bestimmte Protokolle zu konzentrieren oder seine Ursachenanalyse zu aktualisieren. Sie können auch Ressourcenkonfigurationen, Fehlertrends und Erkenntnisse zu Untersuchungen in Ihrer gesamten Umgebung abfragen, ohne zwischen den Konsolen wechseln zu müssen. Weitere Informationen finden Sie unter the section called “DevOps Aufgaben auf Abruf”.

Proaktive Vermeidung von Zwischenfällen

Nach der Behebung von Vorfällen analysiert der DevOps Agent Muster in Ihrem gesamten Ermittlungsverlauf, um Empfehlungen zu erstellen, mit denen future Vorfälle verhindert und die durchschnittliche Erkennungszeit reduziert werden können. Die Empfehlungen beziehen sich auf vier Bereiche: Zustand der Beobachtbarkeit, Testlücken, Codeänderungen und Infrastrukturarchitektur. Der Agent führt wöchentlich Evaluierungen durch und aktualisiert die Empfehlungen, sobald neue Vorfälle auftreten. Sie können Empfehlungen annehmen, ablehnen oder nachverfolgen, und der Agent lernt aus Ihrem Feedback, um future Vorschläge zu verfeinern. Weitere Informationen finden Sie unter the section called “Proaktive Prävention von Zwischenfällen”.

Verbindung mit dem Agenten herstellen DevOps

AWS DevOps Der Agent unterstützt mehrere Zugriffsmethoden, darunter die Web-App-Konsole, die MCP-Integration für IDEs, das Agent Client Protocol (ACP), Webhooks für ereignisgesteuerte Automatisierung und direkten API-Zugriff. Weitere Informationen finden Sie unter the section called “Schnittstelle zum Agenten DevOps”.

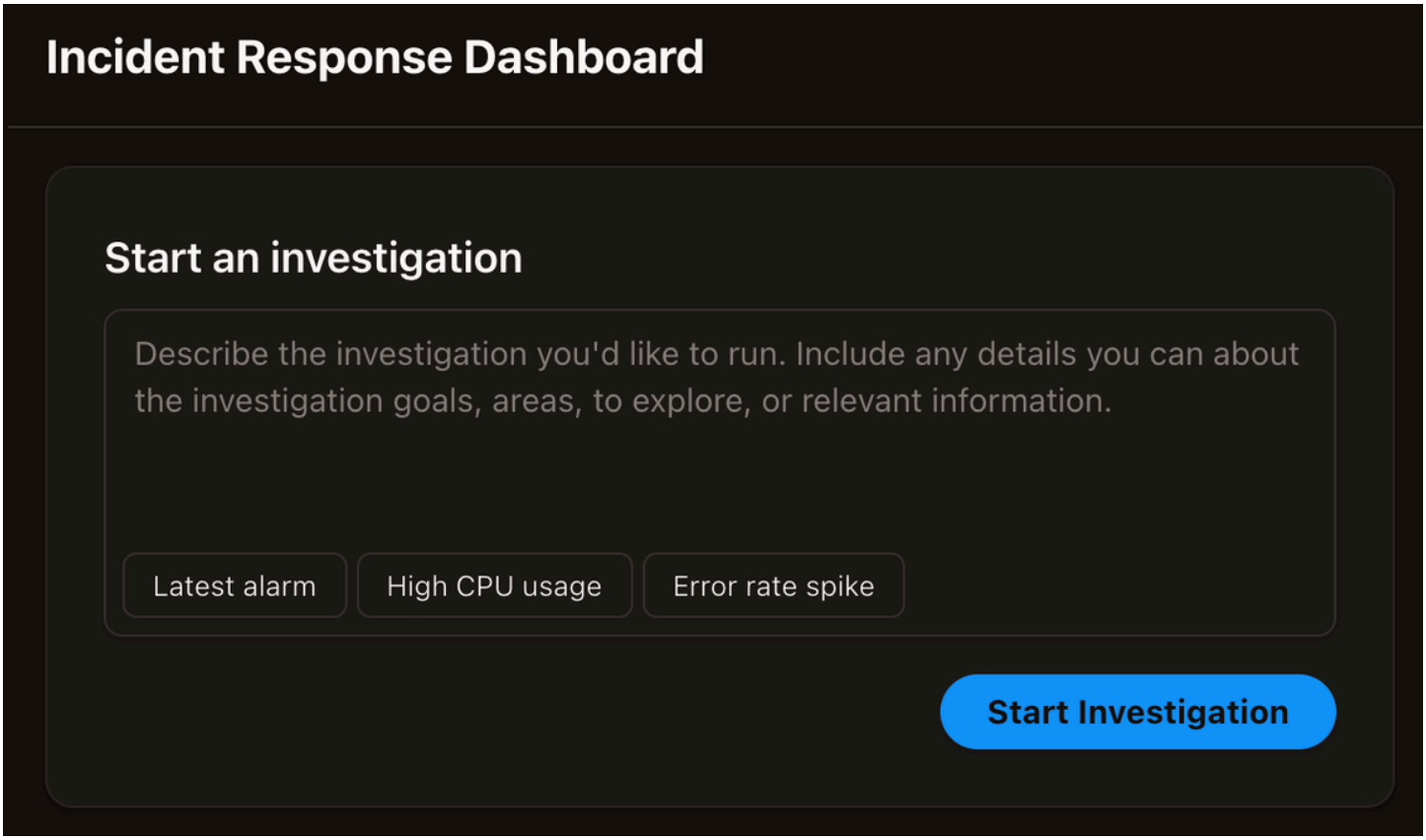
Autonome Reaktion auf Vorfälle

Ermittlungen einleiten

Untersuchungen zur Reaktion auf Vorfälle können auf drei Arten eingeleitet werden.

- Built-in Integrationen — Sie können einen DevOps Agent Space mit Ticketsystemen verbinden, z. B. ServiceNow mithilfe integrierter Integrationen. Sobald die Verbindung hergestellt ist, DevOps werden anhand von Support-Tickets automatisch Untersuchungen zur Reaktion auf Vorfälle durch den DevOps Agenten ausgelöst, und Ihr Mitarbeiter wird die wichtigsten Ergebnisse, Ursachenanalysen und Pläne zur Schadensbegrenzung in das ursprüngliche Ticket aufnehmen.
- Webhooks — Sie können Webhooks verwenden, um Ereignisse an den Agenten zu senden. AWS DevOps Sie können Webhooks beispielsweise verwenden, um Incident-Response-Untersuchungen anhand von PagerDuty Tickets oder Grafana-Alarmen auszulösen.
- Manuell — Sie können Untersuchungen zur Reaktion auf Vorfälle manuell über den Tab „Incident Response“ einer beliebigen DevOps Agent Space-Web-App starten. Sie können entweder Text in freier Form eingeben, der den Vorfall beschreibt, den Ihr DevOps Agent untersuchen soll. Dieser erstellt dann einen Untersuchungsplan, sammelt Ergebnisse, ermittelt die Ursache und bietet an,

einen Plan zur Schadensbegrenzung zu erstellen. Sie können auch aus mehreren vorkonfigurierten Startpunkten wählen, um schnell mit Ihrer Untersuchung zu beginnen: Letzter Alarm, um Ihren zuletzt ausgelösten Alarm zu untersuchen und die zugrunde liegenden Metriken und Protokolle zu analysieren, um die Ursache zu ermitteln, Hohe CPU-Auslastung, um Kennzahlen zu hoher CPU-Auslastung in Ihren Rechenressourcen zu untersuchen und festzustellen, welche Prozesse oder Dienste übermäßig Ressourcen verbrauchen, oder Error Rate Spike, um den jüngsten Anstieg der Anwendungsfehlerraten zu untersuchen, indem Metriken und Anwendungsprotokolle analysiert und die Fehlerquelle identifiziert werden.



Sobald Sie auf „Untersuchung starten“ klicken, werden Sie aufgefordert, einige zusätzliche Informationen anzugeben, damit sich der Agent auf seine Arbeit konzentrieren kann. Das Dialogfeld „Untersuchung“ umfasst die folgenden Felder:

- Einzelheiten der Untersuchung — Pre-filled mit Ihrer Beschreibung. Sie können dies bearbeiten, um den Umfang der Untersuchung zu verfeinern.
- Ausgangspunkt der Untersuchung — Beschreiben Sie optional einen bestimmten Alarm, eine Metrik, einen Protokollausschnitt oder einen anderen Ausgangspunkt für den Agenten.

- Datum und Uhrzeit des Vorfalls — Auto-filled mit der aktuellen Uhrzeit im UTC-Format. Passen Sie an, ob der Vorfall früher aufgetreten ist.
- Benennen Sie Ihre Untersuchung — Auto-generated mit einem Zeitstempel. Sie können dies anpassen (maximal 400 Zeichen).
- Priorität — Wählen Sie die Ermittlungspriorität aus der Dropdownliste aus (Medium ist die Standardeinstellung).

Überprüfen Sie diese Felder und passen Sie sie nach Bedarf an. Klicken Sie dann auf „Untersuchung starten...“, um zu beginnen. Sie werden dann zur Seite mit den Ermittlungsdetails weitergeleitet, auf der Sie Ihren DevOps Agenten in Aktion sehen können!

Triage von Vorfällen

Die Triage-Phase ist die erste Phase des Incident-Response-Systems des AWS DevOps Agenten. Wenn ein externes Ereignis ausgelöst wird, z. B. ein Alarm von Datadog, ein Incident-Ticket von ServiceNow oder ein Problem von Dynatrace, verarbeitet der AWS DevOps Agent es automatisch innerhalb von Sekunden, um zu entscheiden, ob es unabhängig untersucht oder mit einer bestehenden Untersuchung verknüpft werden sollte.

Die Hauptfunktion der Triage-Phase ist die Korrelation von Vorfällen. Dabei werden zusammenhängende Vorfälle identifiziert und in einer einzigen Untersuchung zusammengefasst, um Doppelarbeit und Ressourcenverschwendung zu vermeiden. Wenn ein neuer Vorfall eintrifft, analysiert der AWS DevOps Agent ihn zusammen mit aktiven Untersuchungen innerhalb eines Rückblickfensters (in der Regel 20 Minuten). Mithilfe von AI-powered Analysen werden Faktoren wie Ähnlichkeiten der Komponenten, geografische Region und Zeitmuster untersucht, um Zusammenhänge zwischen Vorfällen zu ermitteln.

AWS DevOps Der Agent trifft eine von drei Entscheidungen:

- Verknüpft — Korreliert den Vorfall mit einer bestehenden Untersuchung und sendet eine Leitbotschaft an diese Untersuchung mit dem Kontext des neuen Vorfalls.
- Übersprungen — Der Vorfall entspricht den in einem Skill definierten Übersprungkriterien und wird automatisch ohne Untersuchung abgewiesen. Weitere Informationen finden Sie unter the section called “DevOps Fähigkeiten der Agenten”.
- Fortfahren — Plant eine neue unabhängige Untersuchung des Vorfalls.

Triage-Entscheidungen anzeigen

Wenn Vorfälle miteinander verknüpft sind, erhält die primäre Untersuchung eine Steuerungsnachricht, die die Einzelheiten des verknüpften Vorfalls und die Korrelationsgründe enthält. In Ihrer AWS DevOps Agent Space-Web-App sehen Sie den Status VERLINKT sowie eine Korrelationsbegründung, die erklärt, warum die Vorfälle verknüpft wurden. In der ersten Untersuchung wird eine Liste aller verknüpften Vorfälle angezeigt, sodass Sie den gesamten Umfang der verwandten Probleme, die gemeinsam untersucht werden, einsehen können. Ihr externes Ticketsystem (ServiceNow PagerDuty, usw.) und Ihr Kommunikationskanal (Slack) erhalten eine Benachrichtigung, dass der Vorfall verknüpft wurde, zusammen mit einer Begründung für den Zusammenhang.

Wenn Vorfälle übersprungen werden, zeigt die AWS DevOps Agent Space-Web-App den Status ÜBERSPRUNGEN an, zusammen mit dem Grund, warum der Vorfall gefiltert wurde. Ihr externes Ticketsystem und Ihr Kommunikationskanal erhalten außerdem eine Benachrichtigung, dass der Vorfall übersprungen wurde, zusammen mit dem Grund für das Überspringen.

Korrektur von Triage-Entscheidungen

Wenn der AWS DevOps Agent einen Vorfall falsch verknüpft, können Sie die Verknüpfung manuell über die AWS DevOps Agent Space-Web-App aufheben. Dadurch wird der nicht verknüpfte Vorfall in eine unabhängige Untersuchung verschoben. Sie können auch benutzerdefinierte Korrelationsregeln bereitstellen, indem Sie einen AWS DevOps Agent-Skill erstellen, der Ihre Korrelationslogik enthält, und ihn der Triage-Phase zuordnen.

Wenn der AWS DevOps Agent einen Incident fälschlicherweise überspringt, können Sie ihn über die AWS DevOps Agent Space-Web-App manuell aufheben. Dadurch wird die Untersuchung des Vorfalls verschoben. Um festzulegen, welche Vorfälle übersprungen werden, ändern oder deaktivieren Sie den Skill, der die Übersprungskriterien definiert.

Bitten Sie um menschliche Unterstützung

AWS DevOps Der Mitarbeiter kann sich direkt mit dem AWS Support in Verbindung setzen, um Ihren Prozess zur Reaktion auf Vorfälle zu optimieren. Wenn Sie zusätzliche Hilfe vom AWS Support benötigen, können Sie in Ihrer DevOps Agent Space-Web-App Supportfälle erstellen, die automatisch den Untersuchungskontext mit den AWS Support-Technikern teilen, sodass Sie weniger Zeit für die Erläuterung Ihres Problems benötigen.

Funktionsweise

Bei der Untersuchung eines Vorfalls erstellt der AWS DevOps Agent ein umfassendes Analyseprotokoll, das Folgendes umfasst:

- Ergebnisse der Untersuchung der Grundursache
- Analysierte Metriken, Protokolle und Traces
- Die Codeänderungen und der Verlauf der Implementierung wurden überprüft
- Es werden Abhilfemaßnahmen empfohlen
- Zeitleiste der Ereignisse und des Systemverhaltens

Sie können Ihre Untersuchung direkt über die AWS DevOps Agent Space-Web-App an den AWS Support weiterleiten. Wenn Sie dies tun, leitet der AWS DevOps Agent sein Ermittlungsprotokoll automatisch an den AWS Support weiter, sodass der Support-Techniker den vollständigen Kontext Ihrer Untersuchung erhält, ohne dass Sie die Details manuell sammeln und erläutern müssen.

Chatten mit AWS Support

Sobald Sie einen Support-Fall erstellt haben, können Sie in einem separaten Chat-Fenster in Ihrer AWS DevOps Agent Space-Web-App mit dem AWS Support kommunizieren. Das ermöglicht Ihnen Folgendes:

- Besprechen Sie Ihr Problem mit den AWS Support-Technikern und besprechen Sie den Zeitplan für die Untersuchung durch Ihren AWS DevOps Agenten
- Sehen Sie sich sowohl die automatisierte Analyse des AWS DevOps Agenten als auch die fachkundige Beratung des AWS Supports auf derselben Oberfläche an
- Geben Sie bei Bedarf problemlos zusätzliche Informationen oder Erläuterungen weiter

Dank des Chat-Erlebnisses haben Sie jederzeit Zugriff auf Ihre AWS DevOps AWS Kundenanfragen und Support-Konversationen, was eine schnellere Zusammenarbeit und Problemlösung ermöglicht.

Sprache der Support-Anfrage

Wenn Sie einen Support-Fall über AWS DevOps Agent erstellen, wird der Fall automatisch in der Sprache erstellt, die in der Spracheinstellung für Agent-Antworten in Ihrem Agent Space konfiguriert ist. Dadurch wird sichergestellt, dass Ihr Support-Fall an einen Support-Techniker weitergeleitet wird, der Ihre bevorzugte Sprache spricht.

Wenn Ihre Agent Space-Sprache beispielsweise auf Japanisch eingestellt ist, wird Ihr Support-Fall an einen Japanese-speaking Support-Techniker weitergeleitet. Wenn keine Sprache konfiguriert ist oder wenn die konfigurierte Sprache vom AWS Support für die ausgewählte Fallkategorie nicht unterstützt wird, ist der Fall standardmäßig auf Englisch eingestellt.

AWS Der Support unterstützt derzeit die folgenden Sprachen für die Fallweiterleitung: Chinesisch, Englisch, Französisch, Japanisch, Koreanisch, Portugiesisch und Spanisch. Um die für Supportanfragen verwendete Sprache zu ändern, aktualisieren Sie die Einstellung für die Antwortsprache des Agenten in Ihrer Agent Space-Konfiguration. Weitere Informationen finden Sie unter the section called “Einen Agentenbereich erstellen”.

Anforderungen an den Supportplan

Ihre Fähigkeit, Supportfälle über AWS DevOps Agent zu erstellen und mit ihnen zu interagieren, hängt von Ihrem AWS Support-Plan ab. Weitere Informationen zu Ihren Ansprüchen finden Sie im Benutzerhandbuch für Supportpläne.

Hinweis Basic Support-Kunden können keine technischen Support-Fälle erstellen und können daher keine Anfragen von Mitarbeitern an den AWS Support weiterleiten. Developer Support Kunden können Fälle über AWS DevOps Agent erstellen, müssen jedoch das AWS Support Center besuchen, um mit den Support-Technikern zu kommunizieren, da der Entwicklersupport keinen chatbasierten Support beinhaltet. Alle anderen Pläne können die integrierte Chat-Funktion innerhalb von Agent nutzen. AWS DevOps AWS DevOps Vollständige Informationen zu den Ansprüchen auf Supportpläne, einschließlich Reaktionszeiten und verfügbarer Schweregrade von Fällen, finden Sie im Benutzerhandbuch AWS für Supportpläne.

Mit welchen Informationen werden sie geteilt AWS Support

Wenn Sie in der AWS DevOps Agent Space-Web-App einen Support-Fall erstellen, werden die folgenden Informationen automatisch an den AWS Support weitergegeben:

- Zeitplan für die Untersuchung: Chronologische Aufzeichnung der Analyse durch den AWS DevOps Agenten
- Informationen zur Ressource: Betroffene AWS Ressourcen
- Beobachtungsdaten: Relevante Metriken, Protokolle und Traces aus Ihren integrierten Überwachungstools
- Aktuelle Änderungen: Codebereitstellungen, Infrastrukturänderungen und Konfigurationsupdates
- Behebungsversuche: Actions AWS DevOps Agent wird empfohlen

- Folgenabschätzung: Umfang und Schwere des Vorfalls

Alle mit dem AWS Support geteilten Daten folgen Ihren bestehenden AWS Datenresidenz- und Sicherheitskonfigurationen. AWS DevOps Der Mitarbeiter gibt nur Informationen weiter, die sich auf Ihre spezifische Untersuchung beziehen, und respektiert die Datenverwaltungsrichtlinien Ihres Unternehmens.

Erste Schritte

So verwenden Sie die AWS Support-Integration für AWS DevOps Agenten:

1. Stellen Sie sicher, dass Sie über einen aktiven AWS Support-Plan verfügen.
2. Stellen Sie sicher, dass die IAM-Berechtigungen Ihres AWS DevOps Agenten die Erstellung von Support-Fällen beinhalten (Support:CreateCase, Support:DescribeCases).
3. Wenn der AWS DevOps Agent ein Problem untersucht und Sie Unterstützung vom AWS Support benötigen, wählen Sie in Ihrer DevOps Agent Space-Web-App die Option Um menschlichen Support bitten aus.
4. Lesen Sie die Zusammenfassung der Untersuchung, die dem AWS Support zur Verfügung gestellt wird.
5. Wählen Sie den entsprechenden Schweregrad der Fälle auf der Grundlage Ihrer Support-Rechte aus.
6. Fall einreichen — Der AWS DevOps Agent fügt automatisch Ihr Ermittlungsprotokoll hinzu.

Das Chat-Fenster wird automatisch geöffnet, sodass Sie sofort mit der Zusammenarbeit mit dem AWS Support beginnen können.

Proaktive Prävention von Zwischenfällen

AWS DevOps Der Agent analysiert Muster bei Ihren Vorfalluntersuchungen, um gezielte Empfehlungen zu geben, mit denen Sie Ihre betriebliche Situation kontinuierlich verbessern und future Vorfälle verhindern können. Über die Seite „Verbesserungen“ in der Operator Web App können Sie auf die proaktive Prävention von Vorfällen zugreifen.

So funktioniert die proaktive Prävention von Zwischenfällen

AWS DevOps Der Mitarbeiter bewertet die jüngsten Untersuchungen von Vorfällen, um dauerhafte Verbesserungen zu ermitteln, um future Vorfälle zu verhindern und die mittlere Erkennungszeit

(MTTD) zu verkürzen. Der Agent analysiert mehrere Vorfälle, um Empfehlungen zu identifizieren, mit denen ganze Gruppen von Vorfällen in future verhindert werden können. Dabei konzentriert er sich auf die wirksamsten Empfehlungen, um sicherzustellen, dass sie umsetzbar sind.

Standardmäßig führt der Agent wöchentlich automatisch Evaluierungen durch. Sie können den Zeitplan unterbrechen, wenn Sie es vorziehen, Evaluierungen nur bei Bedarf durchzuführen. Manuelle Bewertungen sind immer verfügbar. Dies ist nützlich, wenn eine kürzlich durchgeführte Untersuchung eine schnelle Bearbeitung der empfohlenen Verbesserungen rechtfertigt.

Der Mitarbeiter identifiziert Verbesserungen in vier Kategorien, die in der Tabelle zur Kategorisierung von Empfehlungen auf der Seite „Verbesserungen“ dargestellt sind:

- Beobachtbarkeit — Empfehlungen zur Verbesserung von Überwachung, Warnmeldungen, Protokollierung und Systemtransparenz, sodass Probleme schneller und genauer erkannt werden können.
- Infrastruktur — Empfehlungen zur Optimierung der Ressourcenkonfigurationen, der Kapazitätsoptimierung und der Widerstandsfähigkeit der Architektur.
- Verwaltung — Empfehlungen zur Stärkung der Bereitstellungsprozesse, der Verbesserung der Pipeline, der Testpraktiken und der Betriebskontrollen.
- Codeoptimierung — Empfehlungen zur Verbesserung der Qualität des Anwendungscodes, der Fehlerbehandlung und der Widerstandsfähigkeit des Codes.

Diese Kategorisierung hilft Ihnen zu verstehen, wo Ihre betrieblichen Verbesserungen am dringendsten erforderlich sind, und ermöglicht es Ihnen, Empfehlungen auf der Grundlage der Schwerpunktbereiche Ihres Teams zu priorisieren.

Vorteile

- Vermeiden Sie wiederkehrende Vorfälle — Gehen Sie systematisch auf die Grundursachen ein, anstatt immer wieder auf dieselben Probleme zu reagieren
- Reduzieren Sie den betrieblichen Aufwand — Befreien Sie Ihr Team von wiederholter Brandbekämpfung, sodass es sich auf Innovationen und strategische Verbesserungen konzentrieren kann
- Verbessern Sie die Systemstabilität — Stärken Sie Ihre Infrastruktur, Beobachtbarkeit und Bereitstellungsprozesse auf der Grundlage realer Vorfalldaten
- Lernen Sie aus historischen Mustern — Nutzen Sie Erkenntnisse aus vergangenen Vorfällen, um gezielte Verbesserungen vorzunehmen, die die größte Wirkung haben

Zusammenfassung der Agenten

Die Agentenübersicht auf der Seite „Verbesserungen“ der Web-App enthält eine Beschreibung der Ergebnisse der letzten Bewertung der jüngsten Vorfälle. In der Zusammenfassung wird die Anzahl der analysierten Vorfalluntersuchungen erläutert, welche Vorfälle früheren ähnlich sind und welche Empfehlungen erstellt oder mit neuen Informationen aktualisiert wurden.

Die Zusammenfassung hilft Ihnen dabei, schnell zu verstehen, was der Mitarbeiter bei seiner letzten Bewertung herausgefunden hat, und hebt die wichtigsten Empfehlungen hervor, die sich am stärksten auf Ihre betriebliche Situation auswirken könnten.

Kontrolle von Evaluierungen

Sie können steuern, wann der AWS DevOps Agent Vorfälle bewertet und Empfehlungen generiert:

- Manuelles Ausführen von Evaluierungen — Klicken Sie auf der Seite mit den Verbesserungen auf die Schaltfläche Jetzt ausführen, um sofort eine Bewertung zu starten. Dies ist nützlich, wenn eine kürzlich durchgeführte Untersuchung eine schnelle Bearbeitung der empfohlenen Verbesserungen rechtfertigt.
- Aktive Evaluierungen beenden — Klicken Sie auf der Seite mit den Verbesserungen auf die Schaltfläche Bewertung beenden, um eine Evaluierung zu beenden, die gerade läuft.

Empfehlungen verwalten

AWS DevOps Der Agent stellt Empfehlungen auf der Seite „Verbesserungen“ bereit, wo Sie sie überprüfen und verwalten können:

- Empfehlungsdetails anzeigen — Klicken Sie auf eine Empfehlung, um die Seite mit den Empfehlungsdetails zu öffnen. Dort finden Sie weitere Informationen zu der vorgeschlagenen Verbesserung, einschließlich der Vorfälle, die der Empfehlung zugrunde lagen, der erwarteten Auswirkungen und der nächsten Schritte. Empfehlungen mit Codeänderungen finden Sie auch in der für den Agenten geeigneten Spezifikation, die Sie einem Programmierer zur Implementierung aushändigen können.
- Beibehalten — Klicken Sie auf „Beibehalten“, um eine Empfehlung zur Nachverfolgung in Ihrem Backlog beizubehalten. Auf diese Weise können Sie überwachen, welche Verbesserungen Sie umsetzen möchten, und deren Fortschritt verfolgen.
- Verwerfen — Klicken Sie auf „Verwerfen“, um eine Empfehlung aus Ihrem Backlog zu entfernen. Wenn Sie eine Empfehlung verwerfen, können Sie in natürlicher Sprache erklären, warum sie nicht

Ihren Bedürfnissen entspricht. Der Mitarbeiter lernt aus diesem Feedback und verwendet es als Grundlage für future Empfehlungen, um sicherzustellen, dass diese im Laufe der Zeit besser an Ihre betrieblichen Prioritäten und Anforderungen angepasst werden.

- Implementiert — Klicken Sie auf „Implementiert“, um eine Empfehlung als abgeschlossen zu markieren. Auf diese Weise können Sie nachverfolgen, welche Verbesserungen vorgenommen wurden, und der Berater kann die Wirksamkeit seiner Empfehlungen im Laufe der Zeit messen.
- Automatisches Entfernen — Empfehlungen, die nicht als „Beibehalten“ oder „Implementiert“ gekennzeichnet wurden, können nach etwa 6 Wochen entfernt werden, sofern durch die Umsetzung der Empfehlung keine neuen Vorfälle verhindert worden wären. Dadurch wird sichergestellt, dass sich die Seite mit den Verbesserungen auf die wichtigsten Verbesserungen für Ihre betrieblichen Herausforderungen konzentriert.
- Aktualisierungen von Empfehlungen — Bestehende Empfehlungen werden aktualisiert, wenn neuere Vorfälle gefunden werden, die durch die Empfehlung verhindert worden wären. Aktualisierungen können die Priorität der Empfehlung ändern oder die Empfehlung auf der Grundlage neuer Erkenntnisse verfeinern.

Priorisierung der Empfehlung

AWS DevOps Der Agent ordnet Ihre Empfehlungen automatisch nach Priorität, sodass Sie sich zuerst auf die wichtigsten Verbesserungen konzentrieren können. Die Rangfolge berücksichtigt den spezifischen Kontext Ihres Teams, die Betriebsmuster und den Schweregrad der Probleme, auf die sich die einzelnen Empfehlungen beziehen.

Wie funktioniert die Priorisierung

In jedem Bewertungszyklus bewertet der Agent Ihre aktiven Empfehlungen (diejenigen, die sich in einem vorgeschlagenen oder beibehaltenen Zustand befinden) anhand einer Kombination aus:

- AI-powered Rangfolge — Der Mitarbeiter bewertet die relative Bedeutung Ihrer wichtigsten Empfehlungen anhand der Relevanz der Kategorie, der Schwere des Vorfalls und der Auswirkungen auf den Betrieb.
- Deterministische Bewertung — Bei größeren Rückständen wendet der Mitarbeiter eine Prioritätsbewertung an, die auf der Häufigkeit, dem Schweregrad und der Aktualität der Vorfälle basiert, um eine konsistente Reihenfolge zu gewährleisten, die über die am besten bewerteten Punkte hinausgeht.

Die Rangliste wird auf der Seite „Verbesserungen“ mit einer numerischen Rangposition angezeigt (1 steht für höchste Priorität). Empfehlungen, die verworfen oder umgesetzt wurden, werden nicht eingestuft.

Prioritäten anpassen

Sie können beeinflussen, wie der Agent Empfehlungen bewertet, indem Sie die Prioritäten Ihres Teams über die Chat-Oberfläche kommunizieren:

- Kategoriepräferenzen festlegen — Teilen Sie dem Agenten mit, welche Empfehlungskategorien für Ihr Team am wichtigsten sind (z. B. „Verbesserungen der Beobachtbarkeit haben Vorrang vor Infrastrukturänderungen“). Der Agent speichert diese Präferenzen und verwendet sie für future Ranking-Bewertungen.
- Bereitstellung von Kontext — Teilen Sie Informationen über bevorstehende Projekte, Compliance-Anforderungen oder Schwerpunktbereiche des Teams. Der Mitarbeiter berücksichtigt diesen Kontext, wenn er festlegt, welche Empfehlungen priorisiert werden sollten.

Um Ihre Einstellungen zu aktualisieren, verwenden Sie die Chat-Oberfläche und beschreiben Sie die Prioritäten Ihres Teams in natürlicher Sprache. Der Mitarbeiter bestätigt, dass er Ihre Präferenzen verstanden hat und wird sie im nächsten Bewertungszyklus anwenden.

Stabilität im Rang

Die Rangfolge der Empfehlungen kann sich zwischen den Bewertungszyklen ändern, wenn:

- Es werden neue Empfehlungen hinzugefügt, die eine höhere Priorität als bestehende haben
- Die angegebenen Präferenzen Ihres Teams ändern sich
- Neue Vorfalldaten stärken oder schwächen die Argumente für eine Empfehlung

Empfehlungen, die Sie bereits als „Behalten“ markiert haben, behalten unabhängig von Rangänderungen ihren Platz in Ihrem Backlog. So wird sichergestellt, dass Ihr Arbeitsablauf nicht gestört wird.

Agent-ready Spezifikationen

Für Empfehlungen, die Code- oder Konfigurationsänderungen beinhalten, kann der AWS DevOps Agent eine für den Agenten geeignete Spezifikation generieren. Diese Spezifikation bietet ein

strukturiertes Dokument, das zur Implementierung direkt an einen Codierungsagenten übergeben werden kann.

Die Spezifikation beinhaltet:

- Problemstellung — Eine Zusammenfassung des Problems und seiner Ursache
- Lösungszusammenfassung — Eine allgemeine Beschreibung des empfohlenen Ansatzes
- Ziel-Repositorys — Die spezifischen Repositorys, an denen Änderungen vorgenommen werden müssen
- Codeänderungen — Detaillierte Beschreibungen dessen, was geändert werden muss und warum, mit spezifischen Dateipfaden und Überlegungen zur Implementierung
- Testanforderungen — Welche Szenarien müssen getestet werden
- Implementierungsplan — Ein schrittweiser Ansatz zur Umsetzung der Änderungen

Agent-ready Spezifikationen beschleunigen die Implementierung, indem sie den Programmierern den Kontext bieten, den sie benötigen, um produktionsreife Änderungen vorzunehmen, ohne dass ein umfangreiches Hin und Her mit den Technikern erforderlich ist.

Umsetzung von Empfehlungen

Um den Nutzen proaktiver Empfehlungen zur Prävention von Zwischenfällen zu maximieren, sollten Sie die folgenden Methoden zur Umsetzung dieser Empfehlungen in Betracht ziehen:

- Verwendung einsatzbereiter Spezifikationen — Verwenden Sie für Empfehlungen zu Codeänderungen die generierte Spezifikation, um die Implementierung zu beschleunigen, indem Sie sie einem Codierungsagenten übergeben oder sie als detaillierten Leitfaden für die manuelle Implementierung verwenden.
- Empfehlungen zu Ihrem Ticket-Backlog hinzufügen — Kopieren Sie Empfehlungen in das Ticketsystem oder das Projektmanagement-Tool Ihres Teams, um sicherzustellen, dass sie neben anderen technischen Arbeiten priorisiert werden.
- Priorisierung von Empfehlungen auf der Grundlage ihrer Auswirkungen — Konzentrieren Sie sich zunächst auf Empfehlungen, die sich auf die häufigsten oder schwerwiegendsten Arten von Vorfällen beziehen, oder auf solche, die kritische Systeme betreffen.
- Nachverfolgung des Umsetzungsfortschritts — Überwachen Sie, welche Empfehlungen umgesetzt wurden, und messen Sie deren Wirksamkeit, indem Sie beobachten, ob ähnliche Vorfälle im Laufe der Zeit abnehmen.

- Abstimmung mit den Entwicklungsteams — Teilen Sie Empfehlungen mit den entsprechenden Teams, denen die betroffenen Systeme gehören, und stellen Sie sicher, dass sie über den Kontext und die Ressourcen verfügen, die für die Umsetzung von Verbesserungen erforderlich sind.

DevOps Aufgaben auf Abruf

AWS DevOps Agent On Demand Tasks ist ein auf generativer künstlicher Intelligenz (KI) basierender Konversationsassistent, der es Betriebsteams ermöglicht, ihre Anwendungsarchitektur abzufragen, den Systemzustand zu analysieren und in natürlicher Sprache auf Ermittlungsergebnisse zuzugreifen. Sie können Fragen zu Ihren AWS Ressourcen, Systemmetriken, Alarmstatus, Bereitstellungsverlauf und Vorfallmustern stellen. Der Chat bietet sofortige Antworten, die auf Ihren tatsächlichen Infrastruktur- und Betriebsdaten basieren, sodass Sie nicht mehr zwischen mehreren AWS Konsolen oder Überwachungstools hin- und herschalten müssen.

Chat ist in die DevOps Agent Space-Web-App integriert und bietet kontextsensitive Antworten, die auf der von Ihnen aufgerufenen Seite basieren. Die Benutzeroberfläche verwaltet den Konversationsverlauf, sodass Sie frühere Diskussionen fortsetzen und auf früheren Anfragen aufbauen können.

Aufgaben, Funktionen

AWS DevOps Agent On Demand Tasks bietet umfassende Funktionen, die Ihnen helfen, Ihre Infrastruktur zu verwalten und zu verstehen:

Ressourcenabfragen — Fragen Sie nach AWS Ressourcen in Ihrem Agent Space, einschließlich Lambda-Funktionen, DynamoDB-Tabellen, EKS-Bereitstellungen, Zertifikaten und Infrastrukturkonfigurationen. Chat kann Ressourcen anhand von Attributen wie Laufzeitversionen, Kapazitätseinstellungen oder Bereitstellungsstatus filtern und analysieren. Fragen Sie zum Beispiel: „Wie viele Lambdas verwenden Python 3.8?“ oder „Habe ich irgendwelche Zertifikate, die bald ablaufen?“

Systemintegritätsanalyse — Fragen Sie aktuelle und historische Systemintegritätskennzahlen ab, einschließlich Alarmstatus, Fehlerraten, CPU-Auslastung und Serviceverfügbarkeit. Chat kann Zusammenfassungen des Systemzustands für bestimmte Zeiträume erstellen und Trends im Systemverhalten identifizieren. Stellen Sie Fragen wie „Welche Alarme wurden in den letzten 24 Stunden ausgelöst?“ oder „Gab es in der letzten Stunde 5xx-Fehler?“

Erkenntnisse zu Untersuchungen — Greifen Sie auf Informationen aus abgeschlossenen und laufenden Untersuchungen zu, einschließlich Ursachenanalysen, untersuchten Hypothesen,

überprüften Protokollen und Lösungsmustern. Der Chat kann häufige Vorfallursachen identifizieren und Empfehlungen auf der Grundlage historischer Daten geben. Abfrage „Was ist die häufigste Ursache für Vorfälle im letzten Monat?“ oder „Was ist die durchschnittliche Lösungszeit für abgeschlossene Untersuchungen?“

Steuerung der Untersuchung — Wenn Sie sich eine Seite mit Ermittlungsdetails ansehen, leiten Sie die Untersuchung, indem Sie den Agenten anweisen, sich auf bestimmte Protokolle zu konzentrieren, bestimmte Hypothesen zu untersuchen oder die Ursachenanalyse zu aktualisieren. Geben Sie Steuerungsinformationen wie „Konzentrieren Sie sich auf die Protokolle für den Zahlungsdienst und aktualisieren Sie Ihre RCA“ oder „Untersuchen Sie die Hypothese, dass DynamoDB-Drosselung das Problem verursacht hat“.

Chat-Artefakte — Generieren Sie strukturierte Berichte und Dokumente, z. B. Zusammenfassungen des betrieblichen Zustands, Fehlerberichte und Vorfallanalysen. Artefakte werden in einem speziellen Bereich angezeigt und unterstützen die versionierte Bearbeitung innerhalb der Konversation.

Dateianhänge — Hängen Sie Bilder, Dokumente und Codedateien an Ihre Nachrichten an, damit Chat sie im Kontext analysieren kann. Hängen Sie beispielsweise einen Screenshot eines Alarm-Dashboards, eine YAML-Konfigurationsdatei oder ein Runbook-PDF an und fragen Sie Chat, was als Nächstes zu tun ist. Einzelheiten finden Sie unter Senden von Dateianhängen.

Filterung von Empfehlungen — Fragen Sie Empfehlungen zur Vorfallprävention anhand bestimmter Kriterien ab, z. B. Empfehlungen zu bestimmten Diensten oder betrieblichen Belangen. Im Chat werden die Auswirkungen und Überlegungen zur Umsetzung der einzelnen Empfehlungen erläutert. Zum Beispiel „Zeigen Sie mir Empfehlungen zur Vermeidung von Vorfällen mit DynamoDB“ oder „Welche Empfehlungen würden mir helfen, Probleme mit der Anforderungslatenz schneller zu erkennen?“

Auf Chat zugreifen

Chat ist als persistenter Bereich auf der linken Seite der DevOps Agent Space-Web-App verfügbar. Die linke Seitenleiste enthält die Schaltfläche „+ Neuer Chat“, einen Bereich „Seiten“, in dem Sie zu „Vorfälle“, „Verbesserungen“ und „Topologie“ navigieren können, und einen Bereich „Chats“, in dem Ihre letzten Konversationen angezeigt werden. Wählen Sie „Alle anzeigen“, um Ihren vollständigen Konversationsverlauf zu sehen.

Der Chat bietet kontextsensitive Antworten, je nachdem, wo Sie darauf zugreifen:

Topologie — Stellen Sie allgemeine Fragen zu Ihren Agent Space-Ressourcen, Ihrer Architektur und Ihrem Betriebsstatus. Chat bietet vollen Einblick in alle verbundenen Konten und Dienste. In diesem

Kontext können Sie Ressourcenkonfigurationen, den Bereitstellungsverlauf, Topologieinformationen und Integrationen von Observability-Tools abfragen.

Reaktion auf Vorfälle — Stellen Sie auf der Seite zur Reaktion auf Vorfälle Fragen zu Ermittlungstrends, Lösungszeiten und Vorfallmustern in Ihrem gesamten Agentenbereich. Chat kann historische Ermittlungsdaten analysieren, um häufige Ursachen und Verbesserungsmöglichkeiten zu identifizieren.

Einzelheiten der Untersuchung — Während Sie sich eine bestimmte Untersuchung ansehen, gibt Chat kontextbezogene Antworten zu dieser Untersuchung. Fragen Sie nach überprüften Protokollen, untersuchten Hypothesen, Schlussfolgerungen zu den Ursachen und Plänen zur Schadensbegrenzung. Sie können auch Anregungen zur Steuerung geben, um den Schwerpunkt der Untersuchung zu bestimmen.

Prävention — Auf der Präventionsseite können Sie Empfehlungen mit Filtern abfragen, herausfinden, warum Empfehlungen ausgesprochen wurden, und Umsetzungsansätze untersuchen. Chat hilft Ihnen dabei, Prioritäten zu setzen und die Auswirkungen von Empfehlungen zur Vorfallprävention zu verstehen.

Die Chat-Oberfläche bleibt verfügbar, wenn Sie zwischen den Seiten wechseln, aber der Kontext ändert sich, um relevante Informationen für Ihre aktuelle Ansicht bereitzustellen. Wenn Sie eine neue Konversation beginnen, beginnt sie ohne vorherigen Kontext. Wenn Sie eine bestehende Konversation fortsetzen, behält Chat den vollständigen Konversationsverlauf für Folgefragen bei.

Context-aware Antworten

Chat passt seine Antworten an die Seite an, die Sie in der DevOps Agent Space-Web-App aufrufen. Diese Kontextsensitivität stellt sicher, dass Sie relevante Informationen erhalten, ohne angeben zu müssen, nach welcher Untersuchung oder welchem Ressourcenumfang Sie fragen.

Wenn Sie sich eine Detailseite zu einer Untersuchung ansehen, versteht Chat automatisch, dass Sie zu dieser bestimmten Untersuchung fragen. Fragen wie „Welche Protokolle haben Sie sich angesehen?“ oder „Welche Hypothesen haben Sie untersucht?“ beziehen sich auf die aktuell angezeigte Untersuchung. Wenn Sie Steuerungseingaben geben, wendet Chat diese auf die aktive Untersuchung an und erstellt gegebenenfalls eine neue Version der Ursache.

Auf der Seite „Prävention“ geht Chat davon aus, dass Sie an Empfehlungen zur Prävention von Vorfällen interessiert sind. Abfragen filtern und analysieren automatisch Empfehlungen in Ihrem Agent Space-Kontext. Das System erkennt, ob Sie nach allgemeinen Empfehlungen oder nach spezifischen Empfehlungsdetails fragen.

Wenn Sie von der Topologieseite aus auf Chat zugreifen, bietet Chat einen umfassenden Überblick über alle Ressourcen, Metriken und historischen Daten in Ihrem Agentenbereich. Sie können Fragen zu allen Ressourcen-, Service- oder Betriebsproblemen stellen, ohne den Kontext der Untersuchung oder Empfehlung angeben zu müssen.

Durch diese Kontexterkennung müssen Sie nicht wiederholt angeben, auf welche Untersuchung, Empfehlung oder welchen Ressourcenbereich Sie sich beziehen, wodurch ein natürlicherer Gesprächsablauf entsteht.

Verwalten von Konversationen

Chat speichert den Konversationsverlauf, sodass Sie frühere Diskussionen fortsetzen und auf frühere Anfragen verweisen können.

Neue Konversationen erstellen — Klicken Sie im Chat-Panel auf die Schaltfläche „Neue Sitzung“, um eine neue Konversation ohne vorherigen Kontext zu beginnen. In neue Konversationen werden keine Informationen aus früheren Chats übernommen, sodass Sie ohne Verwirrung Fragen stellen können, die nichts miteinander zu tun haben.

Auf den Konversationsverlauf zugreifen — Klicken Sie auf „Verlauf“, um alle vorherigen Konversationen in Ihrem Agentenbereich einzusehen. Konversationen sind chronologisch mit Zeitstempeln und Vorschautext organisiert. Der Konversationsverlauf wird 90 Tage lang aufbewahrt und ist nur für Ihr Benutzerkonto im Agentenbereich sichtbar.

Konversationen fortsetzen — Wählen Sie eine Konversation aus Ihrem Verlauf aus, um sie dort fortzusetzen, wo Sie aufgehört haben. Der Chat behält den vollständigen Kontext früherer Nachrichten bei, sodass Sie Folgefragen stellen können, die sich auf frühere Teile der Konversation beziehen. Wenn Sie während der Anzeige einer Konversation zwischen den Seiten wechseln, bleibt der Konversationskontext erhalten, aber der seitenspezifische Kontext wird je nach Ihrem aktuellen Standort aktualisiert.

Beachten Sie, dass der Konversationsverlauf innerhalb der einzelnen Agentenbereiche isoliert ist. Konversationen in einem Agent Space sind von anderen Agent Spaces aus nicht sichtbar oder zugänglich. Diese Isolierung stellt sicher, dass vertrauliche Informationen gemäß Ihren Unternehmensgrenzen getrennt bleiben.

Artefakte werden generiert

AWS DevOps Der Agent unterstützt Chat-Artefakte — strukturierte, versionierte Dokumente, die vom Agenten während einer Konversation generiert werden. Artefakte bieten ein spezielles, interaktives

Fenster in der Chat-Benutzeroberfläche, in dem AI-generated Inhalte wie Betriebsberichte, Fehlerzusammenfassungen und Zustandsbeurteilungen überprüft und bearbeitet werden können.

Sie können Artefakte von jeder Seite der DevOps Agent Space-Web-App aus anfordern. Chat verwendet den aktuellen Seitenkontext, um den Artefaktinhalt einzugrenzen.

Wie funktionieren Artefakte

Wenn Sie Chat bitten, Inhalte zu erstellen oder zu aktualisieren, generiert Chat ein Artefakt — in der Regel ein formatiertes Dokument — und zeigt es zusammen mit der Konversation im Artefaktfenster an.

Generieren — Senden Sie eine Anfrage in natürlicher Sprache, um einen Bericht oder ein Dokument zu erstellen. Fragen Sie zum Beispiel: „Generieren Sie einen wöchentlichen Bericht über den Betriebsstatus meines Agentenbereichs“ oder „Zeigen Sie mir einen Bericht über meine 4xx-Fehler der letzten Woche“.

Überprüfung — Das Artefakt wird in einem speziellen Bereich neben der Konversation angezeigt. Sie können den gesamten Inhalt überprüfen, während Sie weiterhin mit dem Chat interagieren.

Bearbeiten — Fordere Änderungen am Artefakt über den Chat an. Fragen Sie beispielsweise „Fügen Sie einen Abschnitt über Lambda-Kaltstarts hinzu“ oder „Aktualisieren Sie den Bericht so, dass er die Daten des letzten Monats enthält“. Chat erstellt eine neue Version des Artefakts mit den von Ihnen gewünschten Änderungen.

Senden von Dateianhängen

Sie können Dateien an Ihre Chat-Nachrichten anhängen, sodass Chat sie zusammen mit Ihrer Frage lesen kann. Verwenden Sie Anlagen, um zu teilen, was Sie gerade sehen — einen Screenshot eines Dashboards oder Alarms, eine Konfigurationsdatei, einen Quellcode, ein Runbook für den Betrieb — und bitten Sie den Agenten, direkt darüber nachzudenken.

Dateien sind auf Ihren Agent Space beschränkt — sie sind von anderen Agent Spaces aus nicht sichtbar, und der Zugriff ist durch dieselben IAM-Berechtigungen begrenzt, die auch für den Rest von Chat gelten. Dateien werden in den verwalteten Agent Space-Speicher hochgeladen, sobald Sie sie anhängen.

Wie hängen Sie Dateien an

Sie können einer Nachricht auf drei Arten Dateien hinzufügen:

- Wählen Sie das Upload-Symbol in der Chat-Eingabesymbolleiste und wählen Sie eine oder mehrere Dateien von Ihrem Gerät aus.
- Ziehen Sie eine oder mehrere Dateien per Drag & Drop in den Chat-Eingabebereich.
- Fügen Sie ein Bild direkt aus Ihrer Zwischenablage ein, beispielsweise nach dem Aufnehmen eines Screenshots.

Jede Datei, die Sie anhängen, erscheint als Chip in der Chat-Eingabe mit einer Fortschrittsanzeige für den Upload. Um eine Vorschau einer Datei anzuzeigen, wählen Sie ihren Chip aus. Um eine Datei zu entfernen, wählen Sie das X auf dem Chip. Die Schaltfläche „Senden“ bleibt deaktiviert, solange eine angehängte Datei noch hochgeladen wird.

Unterstützte Dateitypen

Chat akzeptiert die folgenden drei Kategorien von Dateien:

- Bilder —`png`,`jpeg`,`jpg`,`gif`, `webp`
- Dokumente —`pdf`, `csv`doc,`docx`,`xls`,`xlsx`,`html`,`txt`, `md`
- Text- und Codedateien — `json` `yaml`yml,`xml`,`js`,`ts`,`py`,`java`,`rb`, `gors`,`sh`,`bash`,`log`,`cfg`,`ini`, `toml`

Dateien außerhalb dieser Kategorien werden vor dem Hochladen abgelehnt.

Einschränkungen

Für jede Nachricht gelten die folgenden Beschränkungen:

Limit	Wert
Maximale Dateigröße	3,75 MB
Anlagen pro Nachricht (beliebiger Typmix)	20
Davon sind binäre Dokumente (PDF, DOC, DOCX, XLS, XLSX)	bis zu 5

Darüber hinaus müssen Ihr Nachrichtentext und der Inhalt der Anlage zusammen in das für jede Nachricht spezifische Kontextfenster des Modells passen. Wenn eine Nachricht und ihre Anlagen zu

groß sind, lehnt Chat die Nachricht ab und fordert Sie auf, die Größe oder Anzahl der Anlagen vor dem Senden zu reduzieren.

Anwendungsfälle

Übliche Methoden zur Verwendung von Dateianhängen mit dem DevOps Agenten:

- Hängen Sie einen Screenshot eines Alarm- oder Fehler-Dashboards an und bitten Sie Chat, zu interpretieren, was fehlschlägt und wo als Nächstes gesucht werden muss.
- Hängen Sie den Service-Quellcode an und bitten Sie Chat, die Änderung zu überprüfen, Korrekturen vorzuschlagen oder ihr Verhalten zu erläutern.
- Hängen Sie eine Konfigurationsdatei an (z. B. eine YAML-, JSON- oder TOML-Konfiguration) und bitten Sie Chat, zu ermitteln, warum sich eine Bereitstellung, ein Alarm oder eine Integration schlecht verhält.
- Hängen Sie ein operatives Runbook oder ein PDF mit einem Bericht nach einem Vorfall an und bitten Sie Chat, es in einen Skill umzuwandeln. Der Agent extrahiert das Verfahren und speichert es in Ihrem Agentenbereich, sodass future Untersuchungen es automatisch anwenden können.

Beispielabfragen

Die folgenden Beispiele zeigen, welche Arten von Fragen Sie im Chat stellen können. Diese Beispiele sind nach Anwendungsfall und Kontext gegliedert.

Abfragen zur Artefaktgenerierung

Von einer beliebigen Seite in der DevOps Agent Space-Web-App aus:

- Generieren Sie eine wöchentliche Zusammenfassung des Betriebszustands für meinen Agent Space
- Erstellen Sie einen Bericht über alle 4xx-Fehler der letzten Woche
- Erstellen Sie einen zusammenfassenden Bericht über die Vorfälle der letzten 30 Tage
- Erstellen Sie eine Zusammenfassung der Alarmaktivitäten für den Zahlungsdienst in dieser Woche
- Generieren Sie einen Bericht über den Bereitstellungsverlauf der letzten 7 Tage
- Fassen Sie alle offenen Empfehlungen in einem Bericht zusammen

Abfragen zu Ressourceninformationen

Von einer beliebigen Seite in der DevOps Agent Space-Web-App aus:

- Wie viele Lambda-Funktionen verwenden Python 3.8?
- Habe ich irgendwelche Zertifikate, die bald ablaufen?
- Alle DynamoDB-Tabellen mit Abrechnung auf Abruf auflisten
- Zeigen Sie mir EKS-Cluster in der Produktion
- Welche Lambda-Funktionen wurden in den letzten 90 Tagen nicht bereitgestellt?
- S3-Buckets ohne aktivierte Versionierung auflisten
- Auf welchen RDS-Instances wird Datenbankversion X ausgeführt?

Abfragen zur Systemintegrität

Auf den Seiten Topologie oder Incident Response:

- Welche Alarme wurden in den letzten 24 Stunden ausgelöst?
- Gab es in der letzten Stunde 5xx-Fehler?
- Zeige mir Lambda-Fehlertrends für den Zahlungsservice
- Wie hoch ist die CPU-Auslastung für meinen ECS-Cluster?
- Gibt es fehlerhafte Ziele in meinen Load Balancern?
- Zeig mir Drosselungsereignisse für API Gateway von gestern
- Welche Dienste hatten letzte Woche die höchste Fehlerrate?
- Geben Sie mir einen allgemeinen Gesundheitsbericht der letzten 24 Stunden

Fragen zum Observability-Tool

Aus Topology:

- Splunk-Loggruppen auflisten
- Zeige mir Prometheus-Metriken und ihre Alarmschwellen
- Welche Datadog-Monitore sind für diesen Dienst konfiguriert?
- Notieren Sie die New Relic Alert-Richtlinien
- Zeigen Sie mir die Dynatrace-Dashboard-Konfigurationen

Ermittlungen, Erkenntnisse und Fragen

Von der Seite „Incident Response“:

- Was ist die häufigste Ursache für Vorfälle im letzten Monat?
- Was ist die durchschnittliche Lösungszeit für abgeschlossene Untersuchungen?
- Fassen Sie die Untersuchungen der letzten Woche und deren RCA zusammen
- Wie viele Vorfälle wurden durch DynamoDB-Drosselung verursacht?
- Zeigen Sie mir die Untersuchungstrends des letzten Quartals
- Bei welchen Diensten treten die häufigsten Vorfälle auf?

Anfragen zu Einzelheiten der Untersuchung

Von der Seite mit den Ermittlungsdetails:

- Welche Protokolle haben Sie sich angesehen?
- Welche Hypothesen haben Sie untersucht?
- Wie riskant ist die von Ihnen vorgeschlagene Abhilfemaßnahme?
- Was war der Zeitplan der Ereignisse während dieses Vorfalls?
- Warum kamen Sie zu dem Schluss, dass dies die Hauptursache war?
- Welche Beweise stützen Ihre Ursachenanalyse?
- Wer hat während Ihrer Untersuchung die Leitung übernommen?
- Geben Sie mir eine Zusammenfassung dieser Untersuchung des Vorfalls

Fragen zur Untersuchung und Steuerung

Von der Seite mit den Ermittlungsdetails:

- Konzentrieren Sie sich auf die Protokolle für den Zahlungsdienst zwischen 14:00 und 15:00 Uhr UTC und aktualisieren Sie Ihre RCA
- Untersuchen Sie die Hypothese, dass DynamoDB-Drosselung das Problem verursacht hat
- Überprüfen Sie die ECS-Clusterkonfiguration, um festzustellen, ob dadurch der Alarm ausgelöst wurde
- Überprüfen Sie nur die Protokolle der letzten 2 Stunden, nicht des gesamten Tages

- Untersuchen Sie den Anstieg der Fehler um 15 Uhr
- Schauen Sie sich die API-Gateway-Protokolle anstelle der Lambda-Protokolle an

Fragen zu Empfehlungen zur Prävention

Von der Präventionsseite aus:

- Was sind meine drei wichtigsten Empfehlungen zur Unfallverhütung?
- Zeigen Sie mir Empfehlungen, die Vorfälle mit DynamoDB verhindern
- Welche Empfehlungen würden mir helfen, Probleme mit der Latenz von Anfragen schneller zu erkennen?
- Führen Sie Verbesserungen der Beobachtbarkeit auf, durch die ähnliche Vorfälle verhindert werden könnten
- Zeigen Sie mir Infrastrukturempfehlungen für den Zahlungsdienst
- Welche Empfehlungen haben den größten Einfluss auf die Widerstandsfähigkeit des Systems?

Chat in Ihrem Agentenbereich aktivieren

Chat ist in allen DevOps Agent Space-Web-Apps verfügbar. Der Einrichtungsvorgang hängt davon ab, ob Sie über einen neuen oder bestehenden Agent Space verfügen.

Neue Agentenbereiche

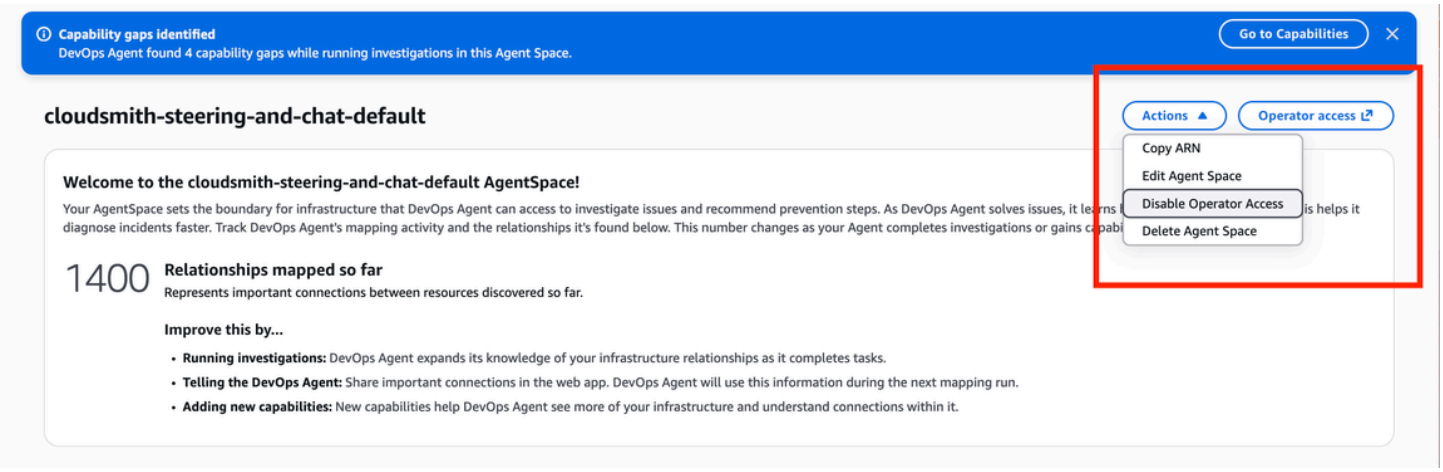
Der Chat wird automatisch aktiviert, wenn Sie einen neuen Agentenbereich erstellen. Es ist keine zusätzliche Konfiguration oder Einrichtung von IAM-Berechtigungen erforderlich. Nachdem Sie Ihre DevOps Agent Space-Web-App konfiguriert haben, ist Chat sofort als persistenter Bereich auf der linken Seite jeder Seite verfügbar.

Bestehende Agentenbereiche

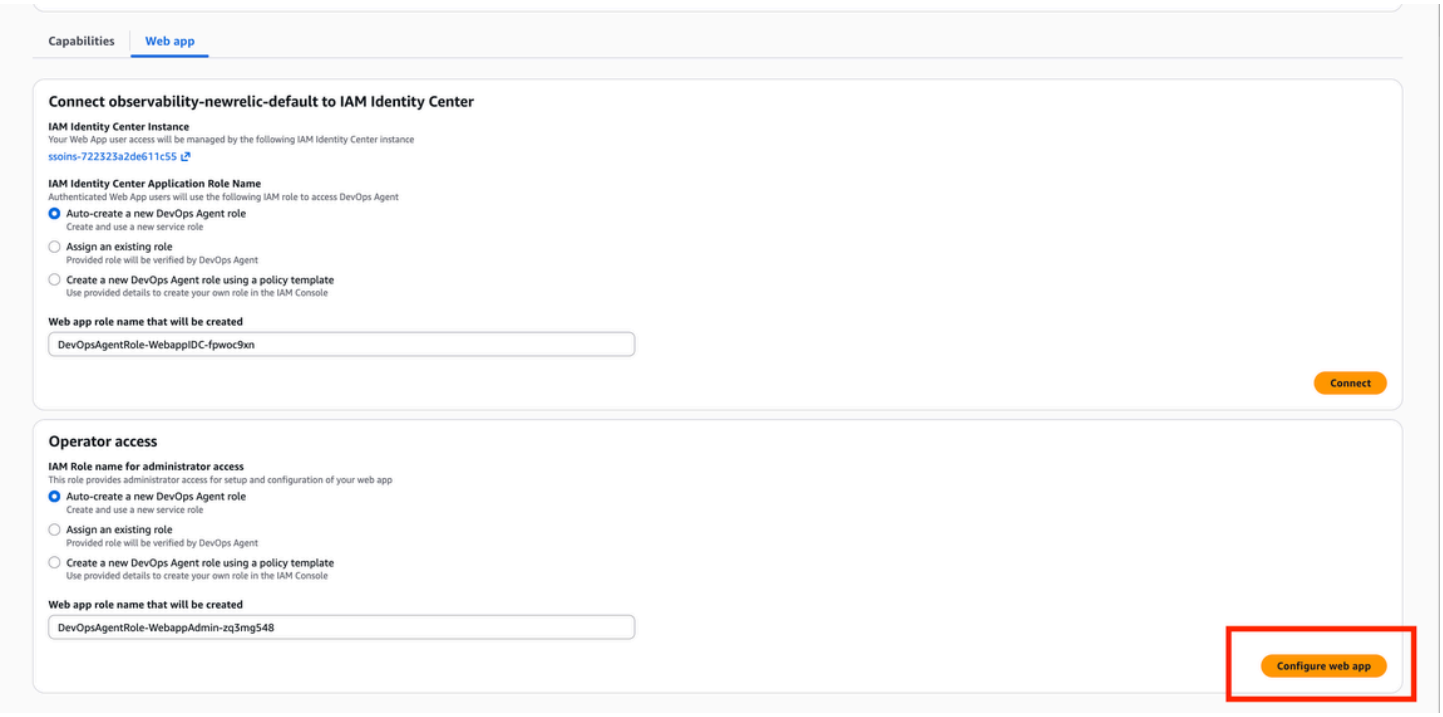
Wenn Sie Ihren Agent Space vor der Veröffentlichung von Chat erstellt haben, müssen Sie die erforderlichen IAM-Berechtigungen aktivieren. Sie haben zwei Optionen:

Option 1: Den Zugriff auf die Operator-App widerrufen und erneut aktivieren

Navigieren Sie zur Administratorkonsole für AWS DevOps Agenten, suchen Sie in der oberen rechten Ecke nach dem Drop-down-Menü „Aktion“ und deaktivieren Sie die aktuelle Konfiguration für den Bedienerzugriff.



Aktivieren Sie dann die Option zur automatischen Erstellung für den Bedienerzugriff.



Dadurch werden automatisch die erforderlichen IAM-Berechtigungen für Chat zusammen mit allen anderen aktuellen Operatorberechtigungen angewendet.

Option 2: Manuelles Hinzufügen von IAM-Berechtigungen

Fügen Sie Ihrer bestehenden Operator-Zugriffsrolle die folgenden IAM-Berechtigungen hinzu:

- `aidevops:ListChats`— Chat-Konversationsverlauf anzeigen
- `aidevops:CreateChat`— Neue Chat-Konversationen erstellen
- `aidevops:SendMessage`— Nachrichten senden und Antworten erhalten

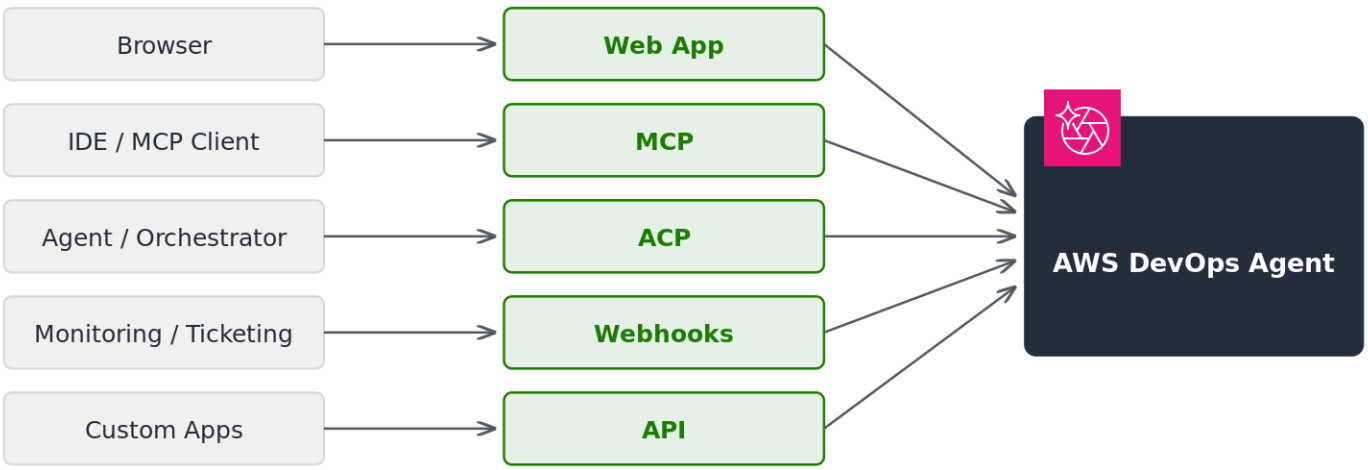
Navigieren Sie zur AWS IAM-Konsole, suchen Sie Ihre DevOps Agent-Operator-Rolle und fügen Sie diese Berechtigungen der Rollenrichtlinie hinzu. Der Chat ist sofort verfügbar, nachdem die Berechtigungen hinzugefügt wurden.

Nachdem Sie eine der Optionen abgeschlossen haben, aktualisieren Sie Ihre DevOps Agent Space-Web-App. Das Chat-Panel wird dann auf jeder beliebigen Seite links angezeigt.

Schnittstelle zum Agenten DevOps

AWS DevOps Der Agent unterstützt fünf Zugriffsmethoden: die Web-App-Konsole, die Integration des Model Context Protocol (MCP), die Integration des Agent Client Protocol (ACP), Webhooks für ereignisgesteuerte Automatisierung und direkten API-Zugriff. Wählen Sie die Methode, die am besten zu Ihrem Arbeitsablauf und Ihren technischen Anforderungen passt.

Das folgende Diagramm veranschaulicht diese Zugriffsmethoden und wie sie eine Verbindung zum DevOps Agent-Dienst herstellen.



DevOps Web-App für Agenten

Die Web-App ist die primäre Schnittstelle für DevOps Agent. Verwenden Sie den Konversations-Chat, um Vorfälle zu untersuchen, Ihre Infrastruktur abzufragen und Empfehlungen zu verwalten. Weitere Informationen finden Sie unter the section called “Was ist eine DevOps Agent-Web-App?”.

Integration des Model Context Protocol (MCP)

Sie können direkt von MCP-compatible Clients und IDEs aus auf die Funktionen der AWS DevOps Agenten zugreifen. Verwenden Sie den AWS MCP-Server, um eine Verbindung herzustellen. Sie können Vorfälle untersuchen, Kosten optimieren, die Architektur überprüfen und die Topologie abbilden, ohne Ihre Entwicklungsumgebung verlassen zu müssen.

Für Kiro-Benutzer steht im Kiro Powers-Repository eine spezielle AWS-Devops-Agent-Leistung zur Verfügung. Diese Stromversorgung verbindet Kiro über den MCP-Server mit dem Agenten. AWS DevOps AWS Es liefert AI-powered betriebliche Informationen direkt in Ihrer IDE.

Für Claude Code-Benutzer bietet das sample-aws-devops-agent-claude-plugin ein vorkonfiguriertes Plugin, das Claude Code über den MCP-Server mit dem Agenten verbindet. AWS DevOps AWS

Integration des Agent Client Protocol (ACP)

Sie können den AWS DevOps Agenten programmgesteuert mithilfe des Agent Client Protocol (ACP) aufrufen. Eine Beispielimplementierung finden Sie im Repository sample-aws-devops-agent-acp-mcp unter. GitHub

Webhooks

Webhooks ermöglichen es externen Systemen, automatisch Untersuchungen durch AWS DevOps Agenten auszulösen. Externe Systeme wie Ticketplattformen und Überwachungstools können HTTP-Anfragen senden, wenn Vorfälle auftreten. Weitere Informationen finden Sie unter the section called “DevOps Agent über Webhook aufrufen”.

AWS DevOps Agenten-API

AWS DevOps Der Agent stellt APIs für den programmatischen Zugriff auf Agentenfunktionen bereit. Sie können Agent Spaces erstellen und verwalten, Untersuchungen auslösen und Ergebnisse abrufen. Weitere Informationen finden Sie in der AWS DevOps Agenten-API-Referenz.

Konfiguration von Funktionen für AWS DevOps Agent

AWS DevOps Die Funktionen von Agenten erweitern die Funktionalität Ihres Agenten, indem sie ihn mit Ihren vorhandenen Tools und Ihrer Infrastruktur verbinden. Konfigurieren Sie diese Funktionen, um eine umfassende Untersuchung von Vorfällen, automatisierte Reaktionsabläufe und eine nahtlose Integration in Ihr DevOps Ökosystem zu ermöglichen.

Die folgenden Funktionen helfen Ihnen dabei, die Effektivität Ihres DevOps Agenten zu maximieren:

- AWS EKS Access Setup — Ermöglicht die Introspektion von Kubernetes-Clustern, Pod-Logs und Cluster-Ereignissen für öffentliche und private EKS-Umgebungen
- Azure-Integration — Connect Azure-Abonnements und DevOps Azure-Organisationen, um Azure-Ressourcen zu untersuchen und DevOps Azure-Bereitstellungen mit Vorfällen zu korrelieren
- CI/CD Pipeline-Integration — Connect GitHub und führen Sie GitLab Pipelines durch, um Bereitstellungen mit Vorfällen zu korrelieren und Codeänderungen bei Untersuchungen nachzuverfolgen
- MCP-Serververbindungen — Erweitern Sie die Untersuchungsmöglichkeiten, indem Sie externe Beobachtungstools und benutzerdefinierte Überwachungssysteme über das Model Context Protocol verbinden
- Multi-Account AWS Zugriff — Konfigurieren Sie sekundäre AWS Konten, um bei der Reaktion auf Vorfälle Ressourcen in Ihrem gesamten Unternehmen zu untersuchen
- Telemetrie-Quellenintegration — Connect Monitoring-Plattformen wie Datadog, Dynatrace, Grafana, New Relic und Splunk für umfassenden Zugriff auf Observability-Daten
- Ticketing- und Chat-Integration — Connect ServiceNow, und Slack PagerDuty, um Workflows zur Reaktion auf Vorfälle zu automatisieren und die Zusammenarbeit im Team zu ermöglichen
- Webhook-Konfiguration — Erlaubt externen Systemen, automatisch DevOps Agentenuntersuchungen über HTTP-Anfragen auszulösen. Einzelheiten zur Einrichtung von Webhooks, zu den Authentifizierungsmethoden und zum Anforderungsformat finden Sie unter. the section called “DevOps Agent über Webhook aufrufen”
- EventBridge Amazon-Integration — Integrieren Sie AWS DevOps Agent in ereignisgesteuerte Anwendungen, indem Sie Ereignisse im Lebenszyklus von Untersuchungen und Abhilfemaßnahmen an Amazon-Ziele weiterleiten EventBridge

Sie können jede Funktion unabhängig auf der Grundlage der spezifischen Bedürfnisse Ihres Teams und des vorhandenen Tool-Stacks konfigurieren. Beginnen Sie mit den Integrationen, die für

Ihren Incident-Response-Workflow am wichtigsten sind, und erweitern Sie sie dann bei Bedarf um zusätzliche Funktionen.

Migration von der öffentlichen Vorversion zur allgemeinen Verfügbarkeit

Wenn Sie AWS DevOps Agent während der öffentlichen Vorschauversion verwendet haben, müssen Sie Ihre IAM-Rollen vor der GA-Version aktualisieren. In dieser Anleitung erfahren Sie, wie Sie die Überwachungs- und Operatorrollen in Ihren Konten aktualisieren.

Was ändert sich

1. Auf On-Demand-Chatverläufe während der Vorschauversion kann nicht mehr zugegriffen werden
2. Neue verwaltete Richtlinien ersetzen die in der Vorschauversion verfügbaren Richtlinien
3. Agent Spaces hat möglicherweise einen veralteten IAM Identity Center-Anwendungszugriffsbereich

On-Demand-Chatverlauf aus der öffentlichen Vorschau

Die GA-Version führt zusätzliche Sicherheitsmaßnahmen ein, um die Zugriffskontrollen für Chat-Verläufe zu verschärfen. Aufgrund dieser Änderungen sind On-Demand-Chatverläufe aus der öffentlichen Vorschauphase (vor dem 30. März 2026) nicht mehr zugänglich. Untersuchungszeitschriften und Ergebnisse, die während der öffentlichen Vorschauphase erstellt wurden, sind davon nicht betroffen. Diese Änderung gilt nur für On-Demand-Chat-Konversationen.

Neue verwaltete Richtlinien

AWS Stellt für GA neue verwaltete Richtlinien bereit, die die Richtlinien aus der Vorschauzeit ersetzen:

Art der Rolle	Remove	Addition
Überwachen	`AIOpsAssistantPolicy` -verwaltete Richtlinie	`AIDevOpsAgentAccessPolicy` -verwaltete Richtlinie

Art der Rolle	Remove	Addition
Betreiber (IAM und IDC)	Online-Richtlinie	`AIDevOpsOperatorAppAccessPolicy` -verwaltete Richtlinie

Darüber hinaus erfordern Operatorrollen aktualisierte Vertrauensrichtlinien, und IDC-Operatorrollen erfordern eine neue Inline-Richtlinie.

Voraussetzungen

- Zugriff auf die AWS Konten, in denen Ihre DevOps Agentenrollen konfiguriert sind (primäre und alle sekundären Konten)
- IAM-Berechtigungen zum Ändern von Rollen, Richtlinien und Vertrauensbeziehungen
- Ihre Agent Space-ID, AWS Konto-ID und Region (sichtbar in der DevOps Agentenkonsole)

Schritt 1: Monitoring-Rollen aktualisieren

Aktualisieren Sie die Überwachungsrolle in Ihrem primären Konto und in jedem sekundären Konto. Dies sind die Primary/Secondary Quellrollen, die auf der Registerkarte Funktionen in Ihrem Agentenbereich konfiguriert sind (primary/secondary Beispielrolle:`DevOpsAgentRole-AgentSpace-3xj2396z`).

1. Gehen Sie in der DevOps Agentenkonsole zu Ihrem Agentenbereich und wählen Sie die Registerkarte Funktionen.
2. Suchen Sie die Monitoring-Rolle für Ihre Primary/Secondary Quellen (z. B.`DevOpsAgentRole-AgentSpace-3xj2396z`) und wählen Sie Bearbeiten.
3. Entfernen Sie unter Berechtigungsrichtlinien die `AIOpsAssistantPolicy` AWS verwaltete Richtlinie.
4. Wählen Sie Berechtigungen hinzufügen, Richtlinien anhängen und fügen Sie die `AIDevOpsAgentAccessPolicy` verwaltete Richtlinie an.
5. Bearbeiten Sie die Inline-Richtlinie und ersetzen Sie ihren Inhalt durch Folgendes, wobei Sie Ihre Konto-ID ersetzen:

```
{
```

```
    "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. Die Vertrauensrichtlinie für die Überwachungsrolle erfordert keine Änderungen. Stellen Sie sicher, dass sie den folgenden Kriterien entspricht:

```
{
    "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/*"
                }
            }
        }
    ]
}
```

- Wiederholen Sie die Schritte 2—6 für die Überwachungsrolle in jedem sekundären Konto.

Schritt 2: Aktualisieren Sie die Operatorrolle (IAM)

1. Wählen Sie in der DevOps Agentenkonsole die Registerkarte Zugriff und suchen Sie nach der Operatorrolle.
2. Entfernen Sie in der IAM-Konsole die vorhandene Inline-Richtlinie aus der Operatorrolle.
3. Wählen Sie „Berechtigungen hinzufügen“, „Richtlinien anhängen“ und fügen Sie die `AIDevOpsOperatorAppAccessPolicy` verwaltete Richtlinie an.
4. Wählen Sie die Registerkarte Vertrauensbeziehungen und dann Vertrauensrichtlinie bearbeiten aus. Ersetzen Sie die Vertrauensrichtlinie durch Folgendes und ersetzen Sie dabei Ihre Konto-ID, Region und Agent Space-ID:

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

Schritt 3: Aktualisieren Sie die Operatorrollen (IDC)

Wenn Sie IAM Identity Center mit DevOps Agent verwenden, aktualisieren Sie jede IDC-Operatorrolle.

1. Gehen Sie in der IAM-Konsole zu Rollen und suchen Sie `WebappIDC` nach Ihren IDC-Rollen für DevOps Agenten (z. B.). `DevOpsAgentRole-WebappIDC-<id>`
2. Gehen Sie für jede IDC-Rolle wie folgt vor:

a. Entfernen Sie die bestehende Inline-Richtlinie.

b. Wählen Sie „Berechtigungen hinzufügen“, „Richtlinien anhängen“ und fügen Sie die `AIDevOpsOperatorAppAccessPolicy` verwaltete Richtlinie an.

c. Wählen Sie die Registerkarte Vertrauensbeziehungen und dann Vertrauensrichtlinie bearbeiten aus. Ersetzen Sie die Vertrauensrichtlinie durch Folgendes und ersetzen Sie dabei Ihre Konto-ID, Region und Agent Space-ID:

```
{
    "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. Erstellen Sie eine neue Inline-Richtlinie mit den folgenden Berechtigungen und ersetzen Sie dabei Ihre Konto-ID:

```
{
    "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/*"
            ]
        }
    ]
```

```
}
```

Stellen Sie die Verbindung zu IAM Identity Center erneut her (falls zutreffend)

In Agent Spaces, die während der öffentlichen Vorschau erstellt wurden, ist möglicherweise eine IAM Identity Center-Anwendung mit einem veralteten Zugriffsbereich konfiguriert. Für GA ist **`aidevops:read_write`**der richtige Bereich. Wenn Ihre IAM Identity Center-Anwendung den vorherigen Geltungsbereich (**`awsaidevops:read_write`**) hat, müssen Sie die Verbindung zu IAM Identity Center trennen und erneut verbinden.

So überprüfen Sie den Geltungsbereich Ihrer IAM Identity Center-Anwendung

Führen Sie den folgenden AWS CLI-Befehl aus, um den Bereich in Ihrer IAM Identity Center-Anwendung zu überprüfen. Sie finden den Anwendungs-ARN in der IAM Identity Center-Konsole unter Anwendungen.

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

Die Ausgabe sollte den richtigen Bereich **`aidevops:read_write`**anzeigen:

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

Wenn der Bereich angezeigt wird **`awsaidevops:read_write`**, ist er veraltet. Gehen Sie wie folgt vor, um ihn zu aktualisieren.

Wie stelle ich die Verbindung zu IAM Identity Center wieder her

Der Zugriffsbereich einer AWS verwalteten IAM Identity Center-Anwendung kann nicht direkt aktualisiert werden. Sie müssen die Verbindung trennen und erneut verbinden:

1. Gehen Sie in der AWS DevOps Agent-Konsole zu Ihrem Agent-Bereich und wählen Sie die Registerkarte Zugriff.
2. Wählen Sie neben der IAM Identity Center-Konfiguration die Option Trennen aus.
3. Bestätigen Sie die Trennung der Verbindung.
4. Wählen Sie Connect, um IAM Identity Center erneut einzurichten. Der Dienst erstellt eine neue IAM Identity Center-Anwendung mit dem richtigen Umfang.
5. Weisen Sie der neuen Anwendung in der IAM Identity Center-Konsole Benutzer und Gruppen neu zu.

> **Important**
>
> Durch das Trennen der Verbindung werden der Chat- und Artefaktverlauf einzelner Benutzer entfernt, die mit IAM Identity Center-Benutzerkonten verknüpft sind. Benutzer müssen sich nach der Wiederverbindung erneut anmelden.

Verifizierung

Nach Abschluss aller Schritte:

1. Kehren Sie zur DevOps Agent-Konsole zurück und stellen Sie sicher, dass auf der Registerkarte Agent Space Access keine Berechtigungsfehler angezeigt werden.
2. Testen Sie die Operator-Web-App, um sicherzustellen, dass sie geladen wird und ordnungsgemäß funktioniert.
3. Wenn Sie IDC verwenden, stellen Sie sicher, dass sich Benutzer authentifizieren und auf das Bedienerlebnis zugreifen können.

Fehlerbehebung

Fehler nach der Migration „Zugriff verweigert“

- Stellen Sie sicher, dass es entfernt `AIOpsAssistantPolicy` wurde und den Überwachungsrollen zugeordnet `AIDevOpsAgentAccessPolicy` ist.
- Stellen Sie sicher, dass alte Inline-Richtlinien entfernt wurden und `AIDevOpsOperatorAppAccessPolicy` den Operatorrollen zugeordnet sind.

- Vergewissern Sie sich, dass die Vertrauensrichtlinien für Betreiber Folgendes `sts:TagSession` enthalten:
- Vergewissern Sie sich, dass Sie alle Platzhalterwerte (`<account-id>`,`<region>`,`<agentspace-id>`) durch tatsächliche Werte ersetzt haben.

Sekundäre Konten funktionieren nicht

- Die Überwachungsrolle jedes sekundären Kontos muss unabhängig aktualisiert werden. Melden Sie sich bei jedem Konto an und wiederholen Sie Schritt 1.

Fehler bei der IDC-Authentifizierung

- Stellen Sie sicher, dass die IDC-Vertrauensrichtlinie sowohl die `sts:TagSession` Anweisung`sts:AssumeRole`/als auch die `TrustedIdentityPropagation` Anweisung enthält.
- Bestätigen Sie, dass die Inline-Richtlinie mit `sso:ListInstancessso:DescribeInstance`, und erstellt `identitystore:DescribeUser` wurde.

Der On-Demand-Chat-Verlauf fehlt nach der Migration

- Auf On-Demand-Chatverläufe aus der öffentlichen Vorschauphase kann nach der Veröffentlichung der öffentlichen Version nicht mehr zugegriffen werden. Dieses Verhalten ist aufgrund der in GA eingeführten erweiterten Sicherheitsmaßnahmen zu erwarten. Untersuchungszeitschriften und Ergebnisse aus der öffentlichen Vorschau sind davon nicht betroffen.

AWS Einrichtung des EKS-Zugangs

Sie können es dem AWS DevOps Agenten ermöglichen, Probleme in Ihren Amazon EKS-Clustern zu untersuchen, indem Sie schreibgeschützte `kubectl` Befehle sowohl für öffentliche als auch für private Cluster ausführen. Sie können eine beliebige Anzahl von EKS-Clustern mit demselben Agent Space verbinden.

Sobald die Verbindung hergestellt ist, kann der Agent bei der Diagnose von Betriebsproblemen in Ihren Clustern helfen. Er beschreibt Ressourcen, ruft Pod-Logs ab, überprüft Cluster-Ereignisse, überprüft den Zustand der Knoten und vieles mehr. Der Agent kann keine Ressourcen in Ihrem Cluster erstellen, ändern oder löschen.

Voraussetzungen

Stellen Sie vor der Einrichtung des EKS-Zugriffs sicher, dass der Authentifizierungsmodus Ihres EKS-Clusters die EKS-API enthält. Sie können dies auf der Registerkarte Zugriff in der Amazon EKS-Konsole überprüfen. Wenn der Modus die EKS-API nicht enthält, wählen Sie einen Modus aus, in dem dies der Fall ist, bevor Sie fortfahren.

Einrichtung

Diese Schritte müssen von der Amazon EKS-Konsole aus für jeden Cluster ausgeführt werden, für den Sie einen Zugriffseintrag erstellen möchten. Sie finden den ARN Ihrer IAM-Rolle in Ihrem Agent Space (siehethe section called “Einen Agentenbereich erstellen”) unter Capabilities > Cloud > Primary Source > Edit.

1. Gehen Sie zur Registerkarte Zugriff. Wenn im Authentifizierungsmodus bereits EKS-API angegeben ist, können Sie Zugriffseinträge hinzufügen. Wählen Sie andernfalls einen Modus aus, der die EKS-API enthält.
2. Erstellen Sie auf der Registerkarte Zugriff einen neuen IAM-Zugriffseintrag. Kopieren Sie den ARN Ihrer primären Cloud-Quell-IAM-Rolle und geben Sie ihn als IAM-Prinzipal für den Zugriffseintrag ein. Klicken Sie auf Weiter.
3. Wählen Sie die AWS Managed AIOpsAssistantPolicyAmazon-Zugriffsrichtlinie und als Zugriffsbereich Cluster aus. (Wenn Sie möchten, dass der Agent nur auf bestimmte Namespaces zugreift, wählen Sie alternativ die gewünschten Kubernetes-Namespaces aus). Klicken Sie auf Richtlinie hinzufügen und dann auf Weiter.
4. Überprüfen Sie die Änderungen und vergewissern Sie sich, dass die richtige Zugriffsrichtlinie und die richtige IAM-Rolle ausgewählt wurden, und erstellen Sie Ihren Zugriffseintrag, indem Sie auf „Erstellen“ klicken.

Um zu überprüfen, ob der EKS-Zugriff korrekt konfiguriert wurde, navigieren Sie zur Operator-App und starten Sie eine neue Untersuchung. Stellen Sie dem Agenten eine Frage zu Ihrem Cluster, z. B. „Alle Pods im Standard-Namespace auflisten“ oder „Zeige mir die letzten Ereignisse in meinem Cluster“.

Fehlerbehebung

Wenn der Agent Ihren Cluster nicht erreichen kann, überprüfen Sie, ob der Zugriffseintrag den richtigen IAM-Rollen-ARN verwendet, der im Einrichtungsdialog angezeigt wird, und ob die AIOpsAssistantPolicyAmazon-Zugriffsrichtlinie beigefügt ist.

Azure verbinden

Die Azure-Integration ermöglicht es dem AWS DevOps Agenten, Ressourcen in Ihrer Azure-Umgebung zu untersuchen und DevOps Azure-Pipeline-Bereitstellungen mit betrieblichen Vorfällen zu korrelieren. Durch die Verbindung von Azure erhält der Agent Einblick in Ihre Azure-Infrastruktur und kann Ursachenanalysen sowohl für Azure-Ressourcen als auch für AWS Azure-Ressourcen durchführen.

Die Azure-Integration besteht aus zwei unabhängigen Funktionen:

- Azure-Ressourcen — Ermöglicht es dem Agenten, Azure-Cloudressourcen wie virtuelle Maschinen, Azure Kubernetes Service (AKS) -Cluster, Datenbanken und Netzwerkkomponenten zu erkennen und zu untersuchen. Der Agent verwendet Azure Resource Graph, um Ihre Ressourcen bei der Untersuchung von Vorfällen abzufragen.
- Azure DevOps — Ermöglicht dem Agenten den Zugriff auf DevOps Azure-Repositorys und den Pipeline-Ausführungsverlauf. Der Agent kann Codeänderungen und Bereitstellungen mit Vorfällen korrelieren, um mögliche Ursachen zu identifizieren.

Jede Funktion wird auf AWS Kontoebene registriert und kann dann einzelnen Agent Spaces zugeordnet werden.

Methoden der Registrierung

AWS DevOps Der Agent unterstützt zwei Methoden für die Verbindung mit Azure:

- Zustimmung durch den Administrator — Ein optimierter, auf Zustimmung basierender Ablauf, bei dem Sie die AWS DevOps Agent Entra-Anwendung in Ihrem Azure-Mandanten autorisieren. In der Konsole wird dies als Option Admin Consent angezeigt. Für diese Methode müssen Sie sich mit einem Konto anmelden, das über die Berechtigung verfügt, die Zustimmung des Administrators in Microsoft Entra ID zu erteilen.
- App-Registrierung — Ein selbstverwalteter Ansatz, bei dem Sie mithilfe von Outbound Identity Federation Ihre eigene Entra-Anwendung mit föderierten Identitätsanmeldedaten erstellen. In der

Konsole wird dies als Option für die App-Registrierung angezeigt. Diese Methode eignet sich, wenn Sie mehr Kontrolle über die Anwendungskonfiguration benötigen oder wenn die Zustimmung des Administrators nicht verfügbar ist.

Beide Methoden bieten dieselben Funktionen. Sie können eine oder beide Methoden innerhalb desselben AWS Kontos verwenden.

Bekannte Beschränkungen

- Zustimmung des Administrators: ein AWS Konto pro Azure-Mandant — Jedem Azure-Mandanten kann die AWS DevOps Agent Entra-App jeweils nur einem AWS Konto zugeordnet werden. Um denselben Mandanten einem anderen AWS Konto zuzuordnen, müssen Sie zuerst die bestehende Registrierung abmelden.
- App-Registrierung: einzigartiger Antrag pro Registrierung — Für jede App-Registrierung muss eine andere Anwendung (Client-ID) verwendet werden. Sie können nicht mehrere Konfigurationen mit derselben Client-ID registrieren.
- Azure DevOps: Quellcodezugriff — Die DevOps Azure-Integration bietet Zugriff auf den Pipeline-Ausführungsverlauf, unabhängig davon, wo der Quellcode gehostet wird. Um auf den eigentlichen Quellcode zugreifen zu können, muss das Repository jedoch separat über einen unterstützten Quellanbieter verbunden werden (z. B.the section called “Verbindung herstellen GitHub”). Auf den in Bitbucket gehosteten Quellcode kann über die DevOps Azure-Integration nicht direkt zugegriffen werden.

Topics

Azure-Ressourcen verbinden

Die Azure-Ressourcenintegration ermöglicht es dem AWS DevOps Agenten, bei der Untersuchung von Vorfällen Ressourcen in Ihren Azure-Abonnements zu erkennen und zu untersuchen. Der Agent verwendet Azure Resource Graph für die Ressourcenerkennung und kann auf Metriken, Protokolle und Konfigurationsdaten in Ihrer Azure-Umgebung zugreifen.

Diese Integration folgt einem zweistufigen Prozess: Registrieren Sie Azure auf AWS Kontoebene und verknüpfen Sie dann bestimmte Azure-Abonnements mit einzelnen Agent Spaces.

Voraussetzungen

Bevor Sie Azure-Ressourcen verbinden, stellen Sie sicher, dass Sie über Folgendes verfügen:

- Zugriff auf die AWS DevOps Agentenkonsole
- Ein Azure-Konto mit Zugriff auf das Zielabonnement
- Für die Admin-Zustimmungsmethode: ein Konto mit der Berechtigung, die Admin-Zustimmung in Microsoft Entra ID zu erteilen
- Für die App-Registrierungsmethode: eine Entra-Anwendung mit Berechtigungen zur Konfiguration von föderierten Identitätsdaten und aktivierter Outbound Identity Federation in Ihrem Konto AWS

> Hinweis: Sie können die Registrierung auch von einem Agent Space aus starten. Navigieren Sie zu Sekundäre Quellen, klicken Sie auf Hinzufügen und wählen Sie Azure aus. Wenn Azure Cloud noch nicht registriert ist, führt Sie die Konsole zunächst durch die Registrierung.

Registrierung von Azure-Ressourcen über die Zustimmung des Administrators

Die Admin-Zustimmungsmethode verwendet einen auf Zustimmung basierenden Ablauf mit der vom AWS DevOps Agenten verwalteten Anwendung.

Schritt 1: Starten Sie die Registrierung

1. Melden Sie sich bei der AWS Management Console an und navigieren Sie zur AWS DevOps Agent-Konsole
2. Gehen Sie zur Seite Capability Providers
3. Suchen Sie den Bereich Azure Cloud und klicken Sie auf Registrieren
4. Wählen Sie die Registrierungsmethode Admin Consent aus

Schritt 2: Vervollständigen Sie die Zustimmung des Administrators

1. Überprüfen Sie die angeforderten Berechtigungen
2. Klicken Sie hier, um fortzufahren — Sie werden zur Microsoft Entra-Administrator-Zustimmungsseite weitergeleitet

3. Melden Sie sich mit einem Benutzer-Hauptkonto an, das berechtigt ist, die Zustimmung des Administrators zu erteilen
4. Überprüfen Sie den AWS DevOps Agent-Antrag und erteilen Sie ihm die Zustimmung

Schritt 3: Vollständige Benutzerautorisierung

1. Nach der Zustimmung des Administrators werden Sie zur Benutzerautorisierung aufgefordert, um Ihre Identität als Mitglied des autorisierten Mandanten zu überprüfen
2. Melden Sie sich mit einem Konto an, das demselben Azure-Mandanten gehört
3. Nach der Autorisierung werden Sie mit einem Erfolgsstatus zurück zur AWS DevOps Agent-Konsole weitergeleitet

Schritt 4: Rollen zuweisen

Weitere Informationen finden Sie weiter unten unter Zuweisen von Azure-Rollen. Suchen Sie bei der Auswahl von Mitgliedern nach AWS DevOps Agent.

Registrierung von Azure-Ressourcen über die App-Registrierung

Die Methode zur App-Registrierung verwendet Ihre eigene Entra-Anwendung mit föderierten Identitätsanmeldedaten.

Schritt 1: Starten Sie die Registrierung

1. Rufen Sie in der AWS DevOps Agent-Konsole die Seite Capability Providers auf
2. Suchen Sie den Bereich Azure Cloud und klicken Sie auf Registrieren
3. Wählen Sie die Methode zur App-Registrierung aus

Schritt 2: Erstellen und konfigurieren Sie Ihre Entra-Anwendung

Folgen Sie den Anweisungen in der Konsole, um:

1. Aktivieren Sie Outbound Identity Federation in Ihrem AWS Konto (gehen Sie in der IAM-Konsole zu Kontoeinstellungen → Outbound Identity Federation)
2. Erstellen Sie eine Entra-Anwendung in Ihrer Microsoft Entra-ID oder verwenden Sie eine vorhandene

3. Konfigurieren Sie Anmeldeinformationen für föderierte Identitäten in der Anwendung

Schritt 3: Geben Sie die Registrierungsdetails an

Füllen Sie das Anmeldeformular aus mit:

- Mandanten-ID — Ihre Azure-Mandanten-ID
- Mandantenname — Ein Anzeigename für den Mandanten
- Client-ID — Die Anwendungs- (Client-) ID der von Ihnen erstellten Entra-Anwendung
- Zielgruppe — Die Zielgruppen-ID für die Verbundanmeldedaten

Schritt 4: Erstellen Sie die IAM-Rolle

Eine IAM-Rolle wird automatisch erstellt, wenn Sie die Registrierung über die Konsole einreichen. Sie ermöglicht dem AWS DevOps Agenten, Anmeldeinformationen anzunehmen und aufzurufen`sts:GetWebIdentityToken`.

Schritt 5: Rollen zuweisen

Weitere Informationen finden Sie weiter unten unter Zuweisen von Azure-Rollen. Suchen Sie nach der Entra-Anwendung, die Sie bei der Mitgliederauswahl erstellt haben.

Schritt 6: Schließen Sie die Registrierung ab

1. Bestätigen Sie die Konfiguration in der AWS DevOps Agentenkonsole
2. Klicken Sie auf Senden, um die Registrierung abzuschließen

Zuweisen von Azure-Rollen

Gewähren Sie der Anwendung nach der Registrierung Lesezugriff auf Ihr Azure-Abonnement. Dieser Schritt ist für die Methoden Admin Consent und App Registration identisch.

1. Navigieren Sie im Azure-Portal zu Ihrem Zielabonnement
2. Gehen Sie zu Access Control (IAM)
3. Klicken Sie auf Hinzufügen > Rollenzuweisung hinzufügen
4. Wählen Sie die Rolle Leser aus und klicken Sie auf Weiter

5. Klicken Sie auf Mitglieder auswählen und suchen Sie nach der Anwendung (entweder AWS DevOps Agent for Admin Consent oder Ihre eigene Entra-Anwendung für die App-Registrierung)
6. Wählen Sie die Anwendung aus und klicken Sie auf Überprüfen + Zuweisen
7. (Optional) Damit der Agent auf Azure Kubernetes Service (AKS) -Cluster zugreifen kann, führen Sie die folgende Einrichtung für den AKS-Zugriff durch.

> Sicherheitsanforderung: Dem Dienstprinzipal darf nur die Rolle Reader (und optional die unten aufgeführten schreibgeschützten AKS-Rollen) zugewiesen werden. Die Rolle Reader dient als Sicherheitsgrenze, die den Agenten auf schreibgeschützte Operationen beschränkt und die Auswirkungen indirekter Prompt-Injection-Angriffe begrenzt. Durch die Zuweisung von Rollen mit Schreib- oder Aktionsberechtigungen wird der Explosionsradius von Prompt Injection erheblich erhöht, was zu einer Beeinträchtigung der Azure-Ressourcen führen kann. AWS DevOps Der Agent führt nur Lesevorgänge durch. Der Agent ändert, erstellt oder löscht keine Azure-Ressourcen.

Einrichtung des AKS-Zugriffs (optional)

Schritt 1: Zugriff auf Azure Resource Manager (ARM) -Ebene

Weisen Sie der Anwendung die Azure Kubernetes Service Cluster-Benutzerrolle zu.

Gehen Sie im Azure-Portal zu Abonnements → wählen Sie Abonnement → Zugriffskontrolle (IAM) → Rollenzuweisung hinzufügen → wählen Sie Azure Kubernetes Service Cluster-Benutzerrolle aus → weisen Sie sie der Anwendung zu (entweder AWS DevOps Agent for Admin Consent oder Ihre eigene Entra-Anwendung für die App-Registrierung).

Dies deckt alle AKS-Cluster im Abonnement ab. Um den Bereich auf bestimmte Cluster zu beschränken, weisen Sie ihn stattdessen auf der Ebene der Ressourcengruppe oder des einzelnen Clusters zu.

Schritt 2: Kubernetes-API-Zugriff

Wählen Sie eine Option, die auf der Authentifizierungskonfiguration Ihres Clusters basiert:

Option A: Azure Role-Based Access Control (RBAC) für Kubernetes (empfohlen)

1. Aktivieren Sie Azure RBAC auf dem Cluster, falls es nicht bereits aktiviert ist: Azure-Portal → AKS-Cluster → Einstellungen → Sicherheitskonfiguration → Authentifizierung und Autorisierung → wählen Sie Azure RBAC

2. Weisen Sie eine schreibgeschützte Rolle zu: Azure-Portal → Abonnements → Abonnement auswählen → Zugriffskontrolle (IAM) → Rollenzuweisung hinzufügen → Azure Kubernetes Service RBAC Reader auswählen → der Anwendung zuweisen

Dies deckt alle AKS-Cluster im Abonnement ab.

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

Verwenden Sie diese Option, wenn Ihr Cluster bereits die standardmäßige Azure AD-Authentifizierungskonfiguration verwendet und Sie Azure RBAC nicht aktivieren möchten. Dies erfordert eine Einrichtung pro Cluster`kubectl`.

1. Speichern Sie das folgende Manifest unter: `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. <SERVICE_PRINCIPAL_OBJECT_ID>Ersetzen Sie es durch die Objekt-ID Ihres Service Principals. Um sie zu finden: Azure-Portal → Entra ID → Unternehmensanwendungen → suchen Sie nach dem Namen der Anwendung (entweder AWS DevOps Agent for Admin Consent oder Ihre eigene Entra-Anwendung für die App-Registrierung).
2. Auf jeden Cluster anwenden:

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

> Hinweis: Cluster, die nur lokale Konten (ohne Azure AD) verwenden, werden nicht unterstützt. Wir empfehlen, die Azure AD-Integration in Ihrem Cluster zu aktivieren, um diese Funktion nutzen zu können.

Benutzerdefinierte Rolle mit den geringsten Rechten (optional)

Für eine strengere Zugriffskontrolle können Sie eine benutzerdefinierte Azure-Rolle erstellen, die nur auf die Ressourcenanbieter beschränkt ist, die der AWS DevOps Agent verwendet, und nicht auf die allgemeine Leserrolle:

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

Ein Abonnement einem Agent Space zuordnen

Nachdem Sie Azure auf Kontoebene registriert haben, ordnen Sie Ihren Agent Spaces bestimmte Abonnements zu:

1. Wählen Sie in der AWS DevOps Agent-Konsole Ihren Agent Space aus
2. Gehen Sie zur Registerkarte Funktionen
3. Klicken Sie im Abschnitt Sekundäre Quellen auf Hinzufügen
4. Wählen Sie Azure aus
5. Geben Sie die Abonnement-ID für das Azure-Abonnement an, das Sie verknüpfen möchten
6. Klicken Sie auf Hinzufügen, um die Zuordnung abzuschließen

Sie können mehrere Abonnements demselben Agent Space zuordnen, um dem Agenten Transparenz in Ihrer Azure-Umgebung zu bieten.

Verwaltung von Azure Resources-Verbindungen

- Verbundene Abonnements anzeigen — Auf der Registerkarte Funktionen werden im Abschnitt Sekundäre Quellen alle verbundenen Azure-Abonnements aufgeführt.
- Abonnement entfernen — Um ein Abonnement von einem Agent Space zu trennen, wählen Sie es in der Liste Sekundäre Quellen aus und klicken Sie auf Entfernen. Dies hat keine Auswirkungen auf die Registrierung auf Kontoebene.
- Registrierung entfernen — Um die Azure Cloud-Registrierung vollständig zu entfernen, rufen Sie die Seite Capability Providers auf und löschen Sie die Registrierung. Alle Agent Space-Verknüpfungen müssen zuerst entfernt werden.

Azure verbinden DevOps

DevOps Die Azure-Integration ermöglicht es dem AWS DevOps Agenten, auf Repositorys und den Pipeline-Ausführungsverlauf in Ihrer DevOps Azure-Organisation zuzugreifen. Der Agent kann Codeänderungen und Bereitstellungen mit betrieblichen Vorfällen korrelieren, um mögliche Ursachen zu identifizieren.

> Hinweis: DevOps Azure-Pipelines können Quellcode aus Azure Repos oder Bitbucket verwenden. GitHub Die DevOps Azure-Integration bietet unabhängig vom Quellanbieter Zugriff auf den Pipeline-Ausführungsverlauf. Um bei Untersuchungen auf den eigentlichen Quellcode zugreifen zu können, muss das Repository jedoch separat über eine unterstützte Integration verbunden werden, z. the section called “Verbindung herstellen GitHub” B. Auf den Quellcode in Bitbucket kann über diese Integration nicht direkt zugegriffen werden.

Diese Integration folgt einem zweistufigen Prozess: Registrieren Sie Azure DevOps auf AWS Kontoebene und ordnen Sie dann bestimmte Projekte einzelnen Agent Spaces zu.

Voraussetzungen

Bevor Sie Azure verbinden, stellen Sie sicher DevOps, dass Sie über Folgendes verfügen:

- Zugriff auf die AWS DevOps Agentenkonsole
- Eine DevOps Azure-Organisation mit mindestens einem Projekt, das einen Repository- und Pipeline-Verlauf enthält
- Berechtigungen zum Hinzufügen von Benutzern zu Ihrer DevOps Azure-Organisation
- Für die Admin-Zustimmungsmethode: ein Konto mit der Berechtigung, die Admin-Zustimmung in Microsoft Entra ID zu erteilen
- Für die Methode der App-Registrierung: eine Entra-Anwendung mit Berechtigungen zur Konfiguration von föderierten Identitätsanmeldedaten und aktivierter Outbound Identity Federation in Ihrem Konto AWS

> Hinweis: Sie können die Registrierung auch von einem Agent Space aus starten. Navigieren Sie zum Abschnitt Pipelines, klicken Sie auf Hinzufügen und wählen Sie Azure DevOps aus. Wenn Azure noch nicht registriert DevOps ist, führt Sie die Konsole zunächst durch die Registrierung.

Azure DevOps über Admin Consent registrieren

Die Admin Consent-Methode verwendet einen auf Zustimmung basierenden Ablauf mit der vom AWS DevOps Agenten verwalteten Anwendung.

Schritt 1: Starten Sie die Registrierung

1. Melden Sie sich bei der AWS Management Console an und navigieren Sie zur AWS DevOps Agent-Konsole
2. Gehen Sie zur Seite Capability Providers
3. Suchen Sie den DevOpsAzure-Bereich und klicken Sie auf Registrieren
4. Geben Sie den Namen Ihrer DevOps Azure-Organisation ein, wenn Sie dazu aufgefordert werden

Schritt 2: Schließen Sie die Zustimmung des Administrators ab

1. Klicken Sie hier, um fortzufahren — Sie werden zur Microsoft Entra-Administrator-Zustimmungsseite weitergeleitet
2. Melden Sie sich mit einem Hauptbenutzerkonto an, das berechtigt ist, die Zustimmung des Administrators zu erteilen
3. Überprüfen Sie die AWS DevOps Agent-Anwendung und erteilen Sie die Zustimmung

Schritt 3: Vollständige Benutzerautorisierung

1. Nach der Zustimmung des Administrators werden Sie zur Benutzerautorisierung aufgefordert, um Ihre Identität als Mitglied des autorisierten Mandanten zu überprüfen
2. Melden Sie sich mit einem Konto an, das demselben Azure-Mandanten gehört
3. Nach der Autorisierung werden Sie mit einem Erfolgsstatus zurück zur AWS DevOps Agent-Konsole weitergeleitet

Schritt 4: Gewähren Sie Zugriff in Azure DevOps

Weitere Informationen finden Sie DevOps weiter unten unter Zugriff in Azure gewähren. Suchen Sie beim Hinzufügen von Benutzern nach AWS DevOps Agent.

Azure DevOps über die App-Registrierung registrieren

Die App-Registrierung wird von Azure Resources und Azure gemeinsam genutzt DevOps. Wenn Sie die App-Registrierung für Azure-Ressourcen bereits abgeschlossen haben, können Sie mit dem Abschnitt Zugriff in Azure gewähren fortfahren DevOps.

Schritt 1: Starten Sie die ADO-App-Registrierung

1. Rufen Sie in der AWS DevOps Agent-Konsole die Seite Capability Providers auf
2. Suchen Sie den Bereich Azure Cloud und klicken Sie auf Registrieren
3. Wählen Sie die Methode zur App-Registrierung

Schritt 2: Erstellen und konfigurieren Sie Ihre Entra-Anwendung

Folgen Sie den in der Konsole angezeigten Anweisungen, um:

1. Aktivieren Sie Outbound Identity Federation in Ihrem AWS Konto (gehen Sie in der IAM-Konsole zu Kontoeinstellungen → Outbound Identity Federation)
2. Erstellen Sie eine Entra-Anwendung in Ihrer Microsoft Entra-ID oder verwenden Sie eine vorhandene
3. Konfigurieren Sie die Anmeldeinformationen für föderierte Identitäten in der Anwendung

Schritt 3: Geben Sie die Registrierungsdetails an

Füllen Sie das Anmeldeformular aus mit:

- Mandanten-ID — Ihre Azure-Mandanten-ID
- Mandantenname — Ein Anzeigename für den Mandanten
- Client-ID — Die Anwendungs- (Client-) ID der Entra-Anwendung
- Zielgruppe — Die Zielgruppen-ID für die Verbundanmeldedaten

Schritt 4: Erstellen Sie die IAM-Rolle

Eine IAM-Rolle wird automatisch erstellt, wenn Sie die Registrierung über die Konsole einreichen. Sie ermöglicht dem AWS DevOps Agenten, Anmeldeinformationen anzunehmen und aufzurufen`sts:GetWebIdentityToken`.

Schritt 5: Schließen Sie die Registrierung ab

1. Bestätigen Sie die Konfiguration in der AWS DevOps Agentenkonsole
2. Klicken Sie auf Senden, um die Registrierung abzuschließen

Schritt 6: Gewähren Sie Zugriff in Azure DevOps

Weitere Informationen finden Sie DevOps weiter unten unter Zugriff in Azure gewähren. Suchen Sie beim Hinzufügen von Benutzern nach der Entra-Anwendung, die Sie bei der App-Registrierung erstellt haben.

Zugriff in Azure gewähren DevOps

Gewähren Sie der Anwendung nach der Registrierung Zugriff auf Ihre DevOps Azure-Organisation. Dieser Schritt ist für die Methoden Admin Consent und App Registration identisch.

1. Gehen Sie in Azure DevOps zu Organisationseinstellungen > Benutzer > Benutzer hinzufügen
2. Suchen Sie nach der Anwendung (entweder AWS DevOps Agent for Admin Consent oder Ihre eigene Entra-Anwendung für die App-Registrierung)
3. Stellen Sie die Zugriffsebene auf Basic ein
4. Wählen Sie unter Zu Projekten hinzufügen die Projekte aus, auf die der Agent zugreifen soll
5. Wählen Sie unter DevOps Azure-Gruppen die Option Project Readers aus
6. Klicken Sie auf Hinzufügen, um den Vorgang abzuschließen

> Sicherheitsanforderung: Weisen Sie nur der Gruppe „Projektleser“ zu. Der schreibgeschützte Zugriff dient als Sicherheitsgrenze, die den Agenten auf schreibgeschützte Operationen beschränkt und die Auswirkungen indirekter Prompt-Injection-Angriffe begrenzt. Durch die Zuweisung von Gruppen mit Schreib- oder Aktionsberechtigungen wird der Explosionsradius von Prompt Injection erheblich erhöht, was zu einer Beeinträchtigung der Azure-Ressourcen führen kann. DevOps

Ein Projekt einem Agent Space zuordnen

Nachdem Sie Azure DevOps auf Kontoebene registriert haben, ordnen Sie Ihren Agent Spaces bestimmte Projekte zu:

1. Wählen Sie in der AWS DevOps Agent-Konsole Ihren Agent Space aus

2. Gehen Sie zur Registerkarte Funktionen
3. Klicken Sie im Abschnitt Pipelines auf Hinzufügen
4. Wählen Sie Azure DevOps aus der Liste der verfügbaren Anbieter aus
5. Wählen Sie das Projekt aus der Dropdownliste der verfügbaren Projekte aus
6. Klicken Sie auf Hinzufügen, um die Zuordnung abzuschließen

Verwaltung von DevOps Azure-Verbindungen

- Verbundene Projekte anzeigen — Auf der Registerkarte Funktionen werden im Abschnitt Pipelines alle verbundenen DevOps Azure-Projekte aufgeführt.
- Projekt entfernen — Um ein Projekt von einem Agent Space zu trennen, wählen Sie es im Abschnitt Pipelines aus und klicken Sie auf Entfernen.
- Registrierung entfernen — Um die DevOps Azure-Registrierung vollständig zu entfernen, rufen Sie die Seite Capability Providers auf und löschen Sie die Registrierung. Alle Agent Space-Verknüpfungen müssen zuerst entfernt werden.

Anschluss an CI/CD Rohrleitungen

Die CI/CD-Pipeline-Integration ermöglicht es dem AWS DevOps Agenten, Bereitstellungen zu überwachen und bei Untersuchungen Codeänderungen mit Betriebsvorfällen zu korrelieren. Durch die Verbindung Ihrer CI/CD Anbieter kann der Agent Bereitstellungsereignisse verfolgen und sie mit AWS Ressourcen verknüpfen, um bei der Reaktion auf Vorfälle mögliche Ursachen zu identifizieren.

AWS DevOps Agent unterstützt die Integration mit gängigen CI/CD Plattformen in einem zweistufigen Prozess:

1. Registrierung auf Kontoebene — Registrieren Sie Ihren CI/CD Anbieter einmal auf Kontoebene AWS
2. Agent Space-Verbindung — Connect spezifische Projekte oder Repositorys mit einzelnen Agent Spaces, je nach den Bedürfnissen deiner Organisation

Dieser Ansatz ermöglicht es Ihnen, CI/CD Anbieterregistrierungen für mehrere Agent Spaces gemeinsam zu nutzen und gleichzeitig die genaue Kontrolle darüber zu behalten, welche Projekte von jedem Space überwacht werden.

Unterstützte Anbieter CI/CD

AWS DevOps Agent unterstützt die folgenden CI/CD Plattformen:

- GitHub— Stellen Sie mithilfe der AWS DevOps GitHub Agent-App eine Verbindung zu Repositorys von GitHub.com her.
- GitLab— Connect Projekte über GitLab.com, verwaltete GitLab Instanzen oder öffentlich zugängliche, selbst gehostete GitLab Bereitstellungen.

Topics

Verbindung herstellen GitHub

GitHub Durch die Integration kann der AWS DevOps Agent bei der Untersuchung von Vorfällen auf Code-Repositorys zugreifen und Bereitstellungsereignisse empfangen. Diese Integration folgt einem zweistufigen Prozess: Registrierung von auf Kontoebene GitHub, gefolgt von der Verbindung bestimmter Repositorys mit einzelnen Agent Spaces.

AWS DevOps Der Agent unterstützt sowohl GitHub .com- (SaaS) als auch GitHub Enterprise Server-Instanzen (selbst gehostet).

Voraussetzungen

Stellen Sie vor der Verbindung sicher GitHub, dass Sie über Folgendes verfügen:

- Zugriff auf die Administratorkonsole des AWS DevOps Agenten
- Ein GitHub Benutzerkonto oder eine Organisation mit Administratorrechten
- Autorisierung zur Installation von GitHub Apps in Ihrem Konto oder Ihrer Organisation

Für GitHub Enterprise Server benötigen Sie außerdem:

- Eine GitHub Enterprise Server-Instanz (Version 3.x oder höher), auf die über HTTPS zugegriffen werden kann

- Die HTTPS-URL Ihrer GitHub Enterprise Server-Instanz (zum Beispiel`https://github.example.com`)
- (Optional) Eine private Verbindung, wenn Ihre GitHub Enterprise Server-Instanz nicht öffentlich zugänglich ist

Registrierung GitHub (auf Kontoebene)

GitHub wird auf AWS Kontoebene registriert und von allen Agent Spaces in diesem Konto gemeinsam genutzt. Sie müssen sich nur GitHub einmal pro AWS Konto registrieren.

Schritt 1: Navigieren Sie zu den Pipeline-Anbietern

1. Melden Sie sich bei der AWS Management Console an
2. Navigieren Sie zur AWS DevOps Agent-Konsole
3. Gehen Sie zur Registerkarte Funktionen
4. Klicken Sie im Abschnitt Pipeline auf Hinzufügen
5. Wählen Sie GitHubaus der Liste der verfügbaren Anbieter

Wenn es GitHub noch nicht registriert wurde, werden Sie aufgefordert, es zuerst zu registrieren.

Schritt 2: Wählen Sie den Verbindungstyp

Wählen Sie auf dem Bildschirm „ GitHub Konto/Organisation registrieren“ aus, ob Sie sich als Benutzer oder Organisation verbinden möchten:

- Benutzer — Ihr persönliches GitHub Konto mit einem Benutzernamen und einem Profil
- Organisation — Ein gemeinsam genutzter GitHub Account, über den mehrere Personen an vielen Projekten gleichzeitig zusammenarbeiten können

Wenn Sie eine Verbindung zu einer GitHub Enterprise Server-Instanz herstellen, aktivieren Sie das Kontrollkästchen GitHub Enterprise Server verwenden und geben Sie die HTTPS-URL Ihrer Instanz ein (z. B.`https://github.example.com`).

Wenn Ihre GitHub Enterprise Server-Instanz nicht öffentlich zugänglich ist, können Sie optional eine private Verbindung konfigurieren, damit der AWS DevOps Agent Ihre Instanz sicher erreichen kann. Weitere Informationen finden Sie unter the section called “Verbindung zu privat gehosteten Tools herstellen”.

> **Note**
>
> Fügen `/api/v3` Sie in der URL keinen nachfolgenden Pfad ein — geben Sie nur die Basis-URL ein.

Schritt 3: Richten Sie die GitHub App ein

Klicken Sie auf Senden, um mit der Einrichtung der App zu beginnen. Die nächsten Schritte unterscheiden sich je nachdem, ob Sie eine Verbindung zu GitHub .com oder GitHub Enterprise Server herstellen.

Für GitHub .com

1. Sie werden GitHub zur Installation der AWS DevOps GitHub Agent-App weitergeleitet.
2. Wählen Sie aus, in welchem Konto oder welcher Organisation die App installiert werden soll.
3. Die App ermöglicht es dem AWS DevOps Agenten, Ereignisse von verbundenen Repositorys zu empfangen, einschließlich Bereitstellungsereignisse.

Für GitHub Enterprise Server

GitHub Enterprise Server verwendet einen GitHub App-Manifest-Flow, der automatisch eine neue GitHub App auf Ihrer Instanz einrichtet. Dies beinhaltet zwei Weiterleitungen zu Ihrer GitHub Enterprise Server-Instanz.

1. Ihr Browser wird auf die Seite „ GitHub App erstellen“ Ihrer GitHub Enterprise Server-Instanz weitergeleitet.
2. Sie werden sehen, dass der Name der App vorausgefüllt ist. Sie können den Namen jederzeit nach Bedarf ändern. Klicken Sie auf GitHub App erstellen.
3. Sie werden zurück zum AWS DevOps Agenten weitergeleitet, der den Manifestcode gegen App-Anmeldeinformationen eintauscht.

Schritt 4: Wählen Sie Repositorys aus und schließen Sie die Installation ab

1. Sie sehen die Seite „Installieren und autorisieren“ für die GitHub App.
2. Wählen Sie aus, auf welche Repositorys die App zugreifen darf:
   - Alle Repositorien — Gewähren Sie Zugriff auf alle aktuellen und future Repositorien

- Nur Repositorys auswählen — Wählen Sie bestimmte Repositorys aus Ihrem Konto oder Ihrer Organisation

3. Klicken Sie auf Installieren und autorisieren.
4. Sie werden zurück zur AWS DevOps Agent-Konsole weitergeleitet, wo GitHub Sie auf Kontoebene als registriert angezeigt werden.

Repositorys mit einem Agent Space verbinden

Nach der Registrierung GitHub auf Kontoebene können Sie bestimmte Repositorys mit einzelnen Agent Spaces verbinden:

1. Wählen Sie in der AWS DevOps Agent-Konsole Ihren Agent Space aus
2. Gehen Sie zur Registerkarte Funktionen
3. Klicken Sie im Abschnitt Pipeline auf Hinzufügen
4. Wählen Sie GitHubaus der Liste der verfügbaren Anbieter
5. Wählen Sie die Teilmenge der Repositorys aus, die für diesen Agent Space relevant sind
6. Klicken Sie auf Hinzufügen, um die Verbindung herzustellen

Sie können je nach den Anforderungen Ihres Unternehmens verschiedene Gruppen von Repositorys mit verschiedenen Agent Spaces verbinden.

Die GitHub App verstehen

Die AWS DevOps GitHub Agent-App:

- Fordert Zugriff auf Ihre Repositorys an — Sie können die spezifischen Berechtigungen während der Installation der GitHub App überprüfen
- Empfängt Bereitstellungsereignisse und andere Repository-Ereignisse
- Ermöglicht dem AWS DevOps Agenten, Codeänderungen mit betrieblichen Vorfällen zu korrelieren
- Kann jederzeit über Ihre GitHub Einstellungen deinstalliert werden

Für GitHub Enterprise Server wird die GitHub App bei der Registrierung automatisch auf Ihrer Instanz erstellt. Sie können den Repository-Zugriff der App verwalten oder sie über Einstellungen > Anwendungen > Installierte GitHub Apps deinstallieren. Um die App-Definition vollständig zu löschen, gehen Sie zu Einstellungen > Entwicklereinstellungen > GitHub Apps.

GitHub Aktualisierungen der App-Berechtigungen

AWS DevOps Der Agent kann nach der Installation der GitHub App Aktualisierungen der Berechtigungen anfordern, um neue Funktionen zu unterstützen. Wenn das passiert:

1. Sie erhalten eine Benachrichtigung von GitHub bezüglich der Anfrage zur Aktualisierung der Genehmigung.
2. Lesen Sie die Aktualisierungsdetails, um zu erfahren, welche neuen Berechtigungen angefordert werden.
3. Akzeptieren Sie die Anfrage zur Erteilung der aktualisierten Berechtigungen.

An Ihrem Service oder Ihrer Anwendung sind keine Änderungen erforderlich. Sobald Sie die aktualisierten Berechtigungen akzeptiert haben, enthält das nächste Zugriffstoken für die Installation, das der AWS DevOps Agent anfordert, GitHub automatisch die neuen Berechtigungen.

> **Note**
>
> Bis Sie ein Berechtigungsupdate akzeptieren, arbeitet der AWS DevOps Agent weiterhin mit den zuvor erteilten Berechtigungen. Neue Funktionen, die von den aktualisierten Berechtigungen abhängen, sind erst verfügbar, wenn Sie die Anfrage genehmigen.

GitHub Verbindungen verwalten

- Repository-Zugriff aktualisieren — Um zu ändern, auf welche Repositorys die GitHub App zugreifen kann, gehen Sie zu Ihren GitHub Konto- oder Organisationseinstellungen (oder zu den Einstellungen Ihrer GitHub Enterprise Server-Instanz), navigieren Sie zu den installierten GitHub Apps und ändern Sie die Konfiguration der AWS DevOps Agent-App.
- Verbundene Repositorys anzeigen — Wählen Sie in der AWS DevOps Agent-Konsole Ihren Agent Space aus und wechseln Sie zur Registerkarte Funktionen, um die verbundenen Repositorys im Abschnitt Pipeline anzuzeigen.
- GitHub Verbindung entfernen — Um die Verbindung zu einem Agent Space zu GitHub trennen, wählen Sie die Verbindung im Abschnitt Pipeline aus und klicken Sie auf Entfernen. Um die GitHub App vollständig zu deinstallieren, deinstallieren Sie sie in Ihren GitHub Konto- oder Organisationseinstellungen. Da die GitHub App bei GitHub Enterprise Server während der Registrierung direkt auf Ihrer Instanz erstellt wird, können Sie die App optional vollständig bereinigen, indem Sie die beiden folgenden Schritte ausführen:

- App deinstallieren — Gehen Sie zu Einstellungen > Anwendungen > Installierte GitHub Apps, klicken Sie in der App auf Konfigurieren und deinstallieren Sie sie anschließend.
- App löschen — Wähle „Einstellungen“ > „Entwicklereinstellungen“ > „ GitHub Apps“, wähle die App aus, gehe zum Tab „Erweitert“ und wähle „ GitHub App löschen“. Warnung: Das Löschen der GitHub App ist dauerhaft und kann nicht rückgängig gemacht werden. Wenn Sie sie löschen, müssen Sie GitHub Enterprise Server von Anfang an in der AWS DevOps Agent-Konsole neu registrieren, um eine neue App zu erstellen.

Verbindung herstellen GitLab

GitLab Die Integration ermöglicht es dem AWS DevOps Agenten, Bereitstellungen von GitLab Pipelines aus zu überwachen, um bei der Reaktion auf Vorfälle als Grundlage für Ursachenuntersuchungen zu dienen. Diese Integration folgt einem zweistufigen Prozess: Registrierung von auf Kontoebene GitLab, gefolgt von der Verbindung bestimmter Projekte mit einzelnen Agent Spaces.

Registrierung GitLab (auf Kontoebene)

GitLab wird auf AWS Kontoebene registriert und von allen Agent Spaces in diesem Konto gemeinsam genutzt. Einzelne Agent Spaces können dann wählen, welche spezifischen Projekte für ihren Agent Space gelten.

Schritt 1: Navigieren Sie zu den Pipeline-Anbietern

1. Melden Sie sich bei der AWS Management Console an
2. Navigieren Sie zur AWS DevOps Agent-Konsole
3. Gehen Sie zur Seite Capability Providers (zugänglich über die Seitennavigation)
4. Suchen Sie GitLabim Abschnitt Verfügbare Anbieter unter Pipeline und klicken Sie auf Registrieren

Schritt 2: GitLab Verbindung konfigurieren

Konfigurieren Sie auf der GitLab Registrierungsseite Folgendes:

Verbindungstyp — Wählen Sie aus, ob Sie als Person oder als Gruppe eine Verbindung herstellen möchten:

- Persönlich (Standard) — Ihr individuelles GitLab Benutzerkonto mit einem Benutzernamen und einem Profil

- Gruppe — In verwenden Sie Gruppen GitLab, um ein oder mehrere verwandte Projekte gleichzeitig zu verwalten

GitLab Instanztyp — Wählen Sie aus, mit welchem GitLab Instanztyp Sie eine Verbindung herstellen möchten:

- GitLab.com (Standard) — Der öffentliche GitLab Dienst
- Öffentlich zugänglich, selbst gehostet GitLab — Markieren Sie das Kästchen GitLab Self Hosted Endpoint verwenden und geben Sie die URL zu Ihrer GitLab Instance ein

> **Note**
>
> Derzeit werden nur öffentlich zugängliche GitLab Instanzen unterstützt.

Zugriffstoken — Stellen Sie ein GitLab persönliches Zugriffstoken bereit:

1. Melden Sie sich in einem separaten Browser-Tab bei Ihrem GitLab Konto an
2. Navigieren Sie zu Ihren Benutzereinstellungen und wählen Sie Access Tokens
3. Erstellen Sie ein neues persönliches Zugriffstoken mit den folgenden Berechtigungen:
   - `read_repository`— Erforderlich für den Zugriff auf Repository-Inhalte
   - `read_virtual_registry`— Erforderlich für den Zugriff auf virtuelle Registrierungsinformationen
   - `read_registry`— Erforderlich für den Zugriff auf Registrierungsinformationen
   - `api`— Erforderlich für den Lese- und Schreibzugriff auf die API
   - `self_rotate`- Erforderlich für rotierende Tokens. Diese Funktion wird derzeit von AWS DevOps Agent nicht unterstützt, wird aber zu einem späteren Zeitpunkt unterstützt. Wenn Sie es jetzt hinzufügen, müssen Sie in future kein neues Token erstellen.
4. Legen Sie den Ablauf des Tokens auf maximal 365 Tage ab dem aktuellen Datum fest
5. Kopieren Sie das generierte Token
6. Kehren Sie zur AWS DevOps Agentenkonsole zurück
7. Fügen Sie das Token in das Feld „Zugriffstoken“ ein

Schritt 3: Schließen Sie die Registrierung ab

(Optional) Stichwörter — Fügen Sie der GitLab Registrierung aus organisatorischen Gründen AWS Stichwörter hinzu.

Klicken Sie auf Weiter, um Ihre Konfiguration zu überprüfen, und klicken Sie dann auf Senden, um den GitLab Registrierungsprozess abzuschließen. Das System validiert Ihr Zugriffstoken und stellt die Verbindung her.

Projekte mit einem Agent Space verbinden

Nachdem Sie sich auf GitLab Kontoebene registriert haben, können Sie bestimmte Projekte mit einzelnen Agent Spaces verbinden:

1. Wählen Sie in der AWS DevOps Agent-Konsole Ihren Agent Space aus
2. Gehen Sie zur Registerkarte Funktionen
3. Klicken Sie im Abschnitt Pipeline auf Hinzufügen
4. Wählen Sie GitLabaus der Liste der verfügbaren Anbieter
5. Wählen Sie die GitLab Projekte aus, die für Ihren Agent Space relevant sind
6. Klicken Sie auf Speichern

AWS DevOps Der Agent überwacht diese Projekte im Hinblick auf Implementierungen von GitLab Pipelines, um anhand von Ursachenuntersuchungen Informationen zu liefern.

Verbindungen verwalten GitLab

- Aktualisierung des Zugriffstokens — Wenn Ihr Zugriffstoken abläuft oder aktualisiert werden muss, können Sie es in der AWS DevOps Agent-Konsole aktualisieren, indem Sie die GitLab Registrierung auf Kontoebene ändern.
- Verbundene Projekte anzeigen — Wählen Sie in der AWS DevOps Agent-Konsole Ihren Agent-Bereich aus und wechseln Sie zur Registerkarte Funktionen, um die verbundenen Projekte im Abschnitt Pipeline anzuzeigen.
- GitLab Verbindung entfernen — Um GitLab Projekte von einem Agent Space zu trennen, wählen Sie die Verbindung im Abschnitt Pipeline aus und klicken Sie auf Entfernen. Um die GitLab Registrierung vollständig zu entfernen, entfernen Sie sie zunächst aus allen Agent Spaces und löschen Sie dann die Registrierung auf Kontoebene.

MCP-Server verbinden

Model Context Protocol (MCP) -Server erweitern die Ermittlungsmöglichkeiten von AWS DevOps Agent, indem sie Zugriff auf Daten aus Ihren externen Observability-Tools, benutzerdefinierten Überwachungssystemen und betrieblichen Datenquellen bieten. In diesem Handbuch wird erklärt, wie Sie einen MCP-Server mit dem Agenten verbinden. AWS DevOps

Voraussetzungen

Bevor Sie einen MCP-Server verbinden, stellen Sie sicher, dass Ihr Server die folgenden Anforderungen erfüllt:

- Streamables HTTP-Transportprotokoll — Nur MCP-Server, die das Streamable HTTP-Transportprotokoll implementieren, werden unterstützt.
- Authentifizierungsunterstützung — Ihr MCP-Server muss eine der folgenden Authentifizierungsmethoden unterstützen: OAuth 2.0 (Client Credentials oder 3LO), API-Schlüssel-/Token-basierte Authentifizierung oder Signature Version 4 (Sigv4). AWS

Sicherheitsüberlegungen

Beachten Sie bei der Verbindung von MCP-Servern mit Agent die folgenden Sicherheitsaspekte: AWS DevOps

- Zulassungsliste für Tools — Sie sollten nur die spezifischen Tools zulassen, die Ihr Agent Space benötigt, anstatt alle Tools von Ihrem MCP-Server verfügbar zu machen. Informationen dazu, wie Sie das Auflisten von Tools pro Agent Space zulassen, finden Sie unter Konfiguration von MCP-Tools in einem Agent Space.

Bitte beachten Sie, dass die maximale Werkzeuglänge eines MCP-Tools 64 beträgt.

- Prompt-Injection-Risiken — Benutzerdefinierte MCP-Server können ein zusätzliches Risiko von Prompt-Injection-Angriffen mit sich bringen. Weitere Informationen finden Sie unter Prompt-Injection-Schutz: AWS DevOps Agent Security.
- Schreibgeschützte Tools und Zugriff — Setzen Sie nur schreibgeschützte MCP-Tools auf die Liste und stellen Sie sicher, dass Authentifizierungsdaten nur Lesezugriff haben.

Weitere Informationen zu AWS DevOps Agentensicherheit Prompt Injection und dem Modell der gemeinsamen Verantwortung finden Sie unter.

> **Note**
>
> Wenn sich Ihr MCP-Server in einem privaten Netzwerk befindet, finden Sie unter the section called “Verbindung zu privat gehosteten Tools herstellen”

Registrierung eines MCP-Servers (auf Kontoebene)

MCP-Server werden auf AWS Kontoebene registriert und von allen Agent Spaces in diesem Konto gemeinsam genutzt. Einzelne Agent Spaces können dann auswählen, welche spezifischen Tools sie von jedem MCP-Server benötigen.

Schritt 1: Details zum MCP-Server

1. Melden Sie sich bei der AWS Management Console an
2. Navigieren Sie zur AWS DevOps Agent-Konsole
3. Gehen Sie zur Seite Capability Providers (zugänglich über die Seitennavigation)
4. Suchen Sie im Bereich Verfügbare Anbieter nach MCP Server und klicken Sie auf Registrieren
5. Geben Sie auf der Seite mit den MCP-Serverdetails die folgenden Informationen ein:
   - Name — Geben Sie einen aussagekräftigen Namen für Ihren MCP-Server ein
   - Endpunkt-URL — Geben Sie die vollständige HTTPS-URL Ihres MCP-Serverendpunkts ein
   - Beschreibung (optional) — Fügen Sie eine Beschreibung hinzu, um den Zweck des Servers zu identifizieren
   - Dynamische Client-Registrierung aktivieren — Aktivieren Sie dieses Kontrollkästchen, wenn Sie möchten, dass sich der AWS DevOps Agent automatisch beim Autorisierungsserver Ihres MCP-Servers registriert
   - Connect zum Endpunkt über private Verbindung herstellen — Aktivieren Sie dieses Kontrollkästchen, wenn der AWS DevOps Agent privat Anfragen an Ihren MCP-Server stellen soll. Sie können eine bestehende private Verbindung auswählen oder eine neue erstellen. Wenn Sie die OAuth Authentifizierung verwenden, gilt die private Verbindung sowohl für den MCP-Serverendpunkt als auch für den Token-Exchange-Endpunkt. Stellen Sie sicher, dass die private Verbindung mit einer Hostadresse konfiguriert ist, die den Datenverkehr an beide Endpunkte

weiterleiten kann. Weitere Informationen finden Sie unter the section called “Verbindung zu privat gehosteten Tools herstellen”.

6. Klicken Sie auf Weiter

> **Note**
>
> Die URL des MCP-Serverendpunkts wird in den AWS CloudTrail Protokollen Ihres Kontos angezeigt.

Schritt 2: Autorisierungsablauf

Wählen Sie die Authentifizierungsmethode für Ihren MCP-Server aus:

OAuth Client-Anmeldeinformationen — Wenn Ihr MCP-Server den OAuth Client Credentials Flow verwendet:

1. Wählen Sie OAuth Client-Anmeldeinformationen
2. Klicken Sie auf Weiter

OAuth 3LO (Three-Legged OAuth) — Wenn Ihr MCP-Server 3LO zur Authentifizierung verwendet OAuth :

1. OAuth Wählen Sie 3LO
2. Klicken Sie auf Weiter

API-Schlüssel — Wenn Ihr MCP-Server die API-Schlüsselauthentifizierung verwendet:

1. Wählen Sie API-Schlüssel
2. Klicken Sie auf Weiter

AWS SigV4 — Wenn Ihr MCP-Server die Authentifizierung mit AWS Signature Version 4 verwendet:

1. Wählen Sie SigV4 AWS
2. Klicken Sie auf Weiter

Schritt 3: Konfiguration der Autorisierung

Konfigurieren Sie zusätzliche Autorisierungsparameter basierend auf der ausgewählten Authentifizierungsmethode:

Für OAuth Kundenanmeldedaten:

1. Client-ID — Geben Sie die Client-ID des OAuth Kunden ein
2. Geheimer Client-Schlüssel — Geben Sie den geheimen Client-Schlüssel des OAuth Clients ein
3. Exchange-URL — Geben Sie die URL des OAuth Token-Exchange-Endpunkts ein
4. Exchange-Parameter — Geben Sie OAuth Token-Austauschparameter für die Authentifizierung beim Dienst ein
5. Bereich hinzufügen — Fügen Sie OAuth Bereiche für die Authentifizierung hinzu
6. Klicken Sie auf Weiter

Für OAuth 3LO:

1. Client-ID — Geben Sie die Client-ID des OAuth Kunden ein
2. Kundengeheimnis — Geben Sie das geheime Kundengeheimnis des OAuth Kunden ein, falls dies von Ihrem OAuth Kunden verlangt wird
3. Exchange-URL — Geben Sie die URL des OAuth Token-Exchange-Endpunkts ein
4. Autorisierungs-URL — Geben Sie die URL des OAuth Autorisierungsendpunkts ein
5. Code Challenge-Support — Markieren Sie dieses Kontrollkästchen, wenn Ihr OAuth Client Code Challenge unterstützt
6. Bereich hinzufügen — Fügen Sie OAuth Bereiche für die Authentifizierung hinzu
7. Klicken Sie auf Weiter

Für den API-Schlüssel:

1. Geben Sie einen API-Schlüsselnamen ein
2. Geben Sie den Namen des Headers ein, der den API-Schlüssel in der Anfrage enthalten soll
3. Geben Sie den Wert Ihres API-Schlüssels ein
4. Klicken Sie auf Weiter

Für AWS SigV4:

AWS Die SigV4-Authentifizierung ermöglicht es dem AWS DevOps Agenten, eine Verbindung zu MCP-Servern herzustellen, die AWS Signature Version 4 für die Signierung von Anfragen verwenden. Dies ist nützlich für MCP-Server, die hinter Amazon API Gateway oder anderen AWS Diensten gehostet werden, die die SigV4-Authentifizierung unterstützen.

> Hinweis: Private Verbindungen werden für MCP-Server, die die SigV4-Authentifizierung verwenden, nicht unterstützt. Ihr MCP-Serverendpunkt muss öffentlich zugänglich sein. Informationen zu MCP-Servern in privaten Netzwerken, die andere Authentifizierungsmethoden verwenden, finden Sie unter. the section called "Verbindung zu privat gehosteten Tools herstellen"

1. IAM-Rolle konfigurieren — Wählen Sie eine der folgenden Optionen:
   - Eine bestehende Rolle verwenden — Wählen Sie eine vorhandene IAM-Rolle aus der Dropdownliste aus. Die Rolle muss über eine Vertrauensrichtlinie verfügen, die es dem AWS DevOps Agent-Dienstprinzipal ermöglicht, sie zu übernehmen (siehe Erstellen einer IAM-Rolle für die SigV4-Authentifizierung).
   - Manuell eine neue Rolle erstellen — Folgen Sie den step-by-step Anweisungen in der Konsole, um eine neue IAM-Rolle mit der richtigen Vertrauensrichtlinie zu erstellen.
2. AWS Region — Geben Sie die AWS Region für die Sigv4-Signatur ein (z. B.`us-east-1`). Um die SigV4a-Signatur für mehrere Regionen zu verwenden, geben Sie ein. *
3. Dienstname — Geben Sie den AWS Dienstnamen für die SigV4-Signatur ein (z. B. `execute-api` für API Gateway).
4. Benutzerdefinierte Header (optional) — Fügen Sie bis zu 10 benutzerdefinierte Schlüssel-Wert-Header-Paare hinzu, die jeder signierten Anfrage beigefügt werden sollen.
5. Klicken Sie auf Weiter

Schritt 4: Überprüfen und abschicken

1. Überprüfen Sie alle Konfigurationsdetails des MCP-Servers
2. Klicken Sie auf Senden, um die Registrierung abzuschließen
3. AWS DevOps Der Agent validiert die Verbindung zu Ihrem MCP-Server
4. Nach erfolgreicher Validierung wird Ihr MCP-Server auf Kontoebene registriert

Konfiguration von MCP-Tools in einem Agent Space

Nachdem Sie einen MCP-Server auf Kontoebene registriert haben, können Sie konfigurieren, welche Tools von diesem Server für bestimmte Agent Spaces verfügbar sind:

1. Wählen Sie in der AWS DevOps Agent-Konsole Ihren Agent Space aus
2. Gehen Sie zur Registerkarte Funktionen
3. Klicken Sie im Bereich MCP-Server auf Hinzufügen
4. Wählen Sie den registrierten MCP-Server aus, den Sie mit diesem Agent Space verbinden möchten
5. Konfigurieren Sie, welche Tools von diesem MCP-Server für den Agent Space verfügbar sein sollen:
   - Alle Tools zulassen — Macht alle Tools vom MCP-Server verfügbar
   - Bestimmte Tools auswählen — Ermöglicht es Ihnen, auszuwählen, welche Tools zugelassen werden sollen
6. Klicken Sie auf Hinzufügen, um den MCP-Server mit Ihrem Agent Space zu verbinden

AWS DevOps Der Agent kann nun bei Untersuchungen in diesem Agent Space die Tools auf der Zulassungsliste Ihres MCP-Servers verwenden.

MCP-Serververbindungen verwalten

Aktualisierung der Authentifizierungsdaten — Wenn Ihre Authentifizierungsdaten aktualisiert werden müssen, müssen Sie Ihren MCP-Server erneut registrieren. Navigieren Sie in der AWS DevOps Agent-Konsole zur Seite Capability Providers, suchen Sie Ihren MCP-Server, entfernen Sie alle aktiven Verknüpfungen und klicken Sie auf Abmelden. Als Nächstes registrieren Sie Ihren MCP-Server mit den neuen Authentifizierungsdaten und stellen Sie alle erforderlichen Verknüpfungen mit Ihrem Agent Space erneut her.

Verbundene MCP-Server anzeigen — Um alle MCP-Server zu sehen, die mit Ihrem Agent Space verbunden sind, wählen Sie Ihren Agent Space aus, wechseln Sie zur Registerkarte Funktionen und überprüfen Sie den Abschnitt MCP-Server. Sie können hier auch ausgewählte Tools aktualisieren.

MCP-Serververbindungen entfernen — Um einen MCP-Server von einem Agent Space zu trennen, wählen Sie den Server im Abschnitt MCP-Server aus und klicken Sie auf Entfernen. Um eine MCP-Serverregistrierung vollständig zu löschen, entfernen Sie sie zuerst aus allen Agent Spaces und löschen Sie dann die Registrierung auf Kontoebene.

Eine IAM-Rolle für die SigV4-Authentifizierung erstellen

Wenn Sie die AWS SigV4-Authentifizierung verwenden, nimmt der AWS DevOps Agent eine IAM-Rolle in Ihrem Konto ein, um Anfragen an Ihren MCP-Server zu signieren. Diese Rolle muss über eine Vertrauensrichtlinie verfügen, die es dem AWS DevOps Agent-Dienstprinzipal (`aidevops.amazonaws.com`) ermöglicht, diese Rolle zu übernehmen, wobei der stellvertretende Schutz nicht gewährleistet ist.

Vertrauensrichtlinie

Erstellen Sie eine IAM-Rolle mit der folgenden Vertrauensrichtlinie. `REGION`Ersetzen Sie sie durch Ihre AWS Region (z. B.`us-east-1`) und `ACCOUNT_ID` durch Ihre AWS Konto-ID.

```
{
  "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:service/*"
        }
      }
    }
  ]
}
```

Die Vertrauensrichtlinie umfasst die folgenden Bedingungen, um das Problem des verwirrten Stellvertreters zu verhindern:

- `aws:SourceAccount`— Beschränkt die Übernahme der Rolle auf Anfragen, die von Ihrem AWS Konto stammen.
- `aws:SourceArn`— Beschränkt die Rollenübernahme auf Anfragen, die von den Serviceressourcen der AWS DevOps Agenten in Ihrem Konto stammen.

Berechtigungsrichtlinie

Fügen Sie der Rolle eine Berechtigungsrichtlinie hinzu, die die Mindestberechtigungen gewährt, die zum Aufrufen Ihres MCP-Servers erforderlich sind. Wenn Ihr MCP-Server beispielsweise hinter Amazon API Gateway gehostet wird, sollte die Rolle über `execute-api:Invoke` Berechtigungen für die API Gateway-Ressource verfügen.

Signierung in mehreren Regionen (SigV4a)

Wenn Ihr MCP-Server in mehreren AWS Regionen eingesetzt wird, können Sie SigV4a (Signature Version 4a) für das Signieren mehrerer Regionen verwenden. Um dies zu aktivieren, geben Sie bei der Konfiguration der * AWS SigV4-Autorisierung als Region ein. SigV4a verwendet eine asymmetrische Signatur, die es ermöglicht, dass eine einzelne signierte Anfrage für mehrere Regionen gültig ist.

Verwandte Themen

- Sicherheit im Agenten AWS DevOps
- Einen Agent-Bereich einrichten
- Schutz vor sofortiger Injektion

Mehrere AWS Konten verbinden

Sekundäre AWS Konten ermöglichen es dem AWS DevOps Agenten, Ressourcen mehrerer AWS Konten in Ihrer Organisation zu untersuchen. Wenn sich Ihre Anwendungen auf mehrere Konten erstrecken, stellt das Hinzufügen von Sekundärkonten sicher, dass der Agent bei der Untersuchung von Vorfällen Einblick in alle relevanten Ressourcen hat. Ein besserer Zugriff auf die Konten und Ressourcen, aus denen sich eine Anwendung zusammensetzt, gewährleistet eine höhere Genauigkeit der Ermittlungen.

Voraussetzungen

Bevor Sie ein sekundäres AWS Konto hinzufügen, stellen Sie sicher, dass Sie über Folgendes verfügen:

- Zugriff auf die AWS DevOps Agent-Konsole im primären Konto
- Administratorzugriff auf das sekundäre AWS Konto

- IAM-Berechtigungen zum Erstellen von Rollen im sekundären Konto

Ein AWS sekundäres Konto hinzufügen

Zusätzlich zu den unten aufgeführten Schritten können Sie das verwenden, the section called “AWS DevOps Leitfaden für das CLI Onboarding von Agenten” um programmgesteuert sekundäre Konten hinzuzufügen.

Schritt 1: Starten Sie die Konfiguration des sekundären Kontos

1. Melden Sie sich bei der AWS Management Console an und navigieren Sie zur AWS DevOps Agent-Konsole
2. Wählen Sie Ihren Agent Space
3. Gehen Sie zur Registerkarte Funktionen
4. Suchen Sie im Bereich Cloud den Unterabschnitt Sekundäre Quellen
5. Klicken Sie auf Hinzufügen

Schritt 2: Geben Sie den Rollennamen an

1. Geben Sie im Feld Name Ihrer Rolle einen Namen für die Rolle ein, die Sie im sekundären Konto erstellen werden
2. Notieren Sie sich diesen Namen — Sie werden ihn erneut verwenden, wenn Sie die Rolle im sekundären Konto erstellen
3. Kopieren Sie die in der Konsole bereitgestellte Vertrauensrichtlinie und speichern Sie sie in einem Scratch-Bereich

Schritt 3: Erstellen Sie die Rolle im sekundären Konto

1. Öffnen Sie einen neuen Browser-Tab und melden Sie sich im sekundären AWS Konto bei der IAM-Konsole an
2. Navigieren Sie zu IAM > Rollen > Rolle erstellen
3. Wählen Sie Benutzerdefinierte Vertrauensrichtlinie
4. Fügen Sie die Vertrauensrichtlinie ein, die Sie aus Schritt 2 kopiert haben
5. Klicken Sie auf Weiter

Schritt 4: Hängen Sie die AWS verwaltete Richtlinie an

1. Suchen Sie im Abschnitt Berechtigungsrichtlinien nach AIDevOpsAgentAccessPolicy
2. Aktivieren Sie das Kontrollkästchen neben der AIDevOpsAgentAccessPolicyverwalteten Richtlinie
3. Klicken Sie auf Weiter

Schritt 5: Benennen und erstellen Sie die Rolle

1. Geben Sie im Feld Rollenname denselben Rollennamen ein, den Sie in Schritt 2 angegeben haben
2. (Optional) Fügen Sie eine Beschreibung hinzu, um den Zweck der Rolle besser zu identifizieren
3. Überprüfen Sie die Vertrauensrichtlinie und die beigefügten Berechtigungen
4. Klicken Sie auf Rolle erstellen

Schritt 6: Hängen Sie die Inline-Richtlinie an

1. Suchen Sie in der IAM-Konsole die Rolle, die Sie gerade erstellt haben, und wählen Sie sie aus
2. Gehen Sie zur Registerkarte „Berechtigungen“
3. Klicken Sie auf Berechtigungen hinzufügen > Inline-Richtlinie erstellen
4. Wechseln Sie zur Registerkarte JSON
5. Fügen Sie die Richtlinie ein, die Sie in Schritt 2 gespeichert haben
6. Fügen Sie die Richtlinie in den JSON-Editor in der IAM-Konsole ein
7. Klicken Sie auf Weiter
8. Geben Sie einen Namen für die Inline-Richtlinie ein (z. B. "DevOpsAgentInlinePolicy„)
9. Klicken Sie auf Richtlinie erstellen

Schritt 7: Schließen Sie die Konfiguration ab

1. Kehren Sie im Hauptkonto zur AWS DevOps Agentenkonsole zurück
2. Klicken Sie auf Weiter, um die Konfiguration des sekundären Kontos abzuschließen
3. Stellen Sie sicher, dass der Verbindungsstatus als Aktiv angezeigt wird

Grundlegendes zu den erforderlichen Richtlinien

AWS DevOps Der Agent benötigt drei Richtlinienkomponenten, um auf Ressourcen in einem sekundären Konto zuzugreifen:

- Vertrauensrichtlinie — Ermöglicht es dem AWS DevOps Agenten im primären Konto, die Rolle im sekundären Konto zu übernehmen. Dadurch wird das Vertrauensverhältnis zwischen den Konten hergestellt.
- AIDevOpsAgentAccessPolicy (AWS verwaltete Richtlinie) — Stellt die grundlegenden Leseberechtigungen bereit, die der AWS DevOps Agent benötigt, um Ressourcen im sekundären Konto zu untersuchen. Diese Richtlinie wird von beibehalten AWS und aktualisiert, sobald neue Funktionen hinzugefügt werden.
- Inline-Richtlinie — Bietet zusätzliche Berechtigungen, die für Ihre Agent Space-Konfiguration spezifisch sind. Diese Richtlinie wird auf der Grundlage Ihrer Agent Space-Einstellungen generiert und kann Berechtigungen für bestimmte Integrationen oder Funktionen beinhalten.

Im primären Konto muss die AWS DevOps Agent-IAM-Rolle die Rolle übernehmen können, die im sekundären Konto erstellt wurde.

Verwaltung sekundärer Konten

- Verbundene Konten anzeigen — Auf der Registerkarte Funktionen werden im Unterabschnitt Sekundäre Quellen alle verbundenen sekundären Konten mit ihrem Verbindungsstatus aufgeführt.
- Aktualisierung der IAM-Rolle — Wenn Sie die Berechtigungen ändern müssen, aktualisieren Sie die Inline-Richtlinie, die der Rolle im sekundären Konto zugewiesen ist. Änderungen werden sofort wirksam.
- Ein sekundäres Konto entfernen — Um die Verbindung zu einem sekundären Konto zu trennen, wählen Sie es in der Liste Sekundäre Quellen aus und klicken Sie auf Entfernen. Dadurch wird die IAM-Rolle im sekundären Konto nicht gelöscht.

Telemetriequellen anschließen

AWS DevOps Der Agent bietet drei Möglichkeiten, eine Verbindung zu Ihren Telemetriequellen herzustellen.

Integrierte 2-Wege-Integration

Derzeit unterstützt AWS DevOps Agent Dynatrace-Benutzer mit einer integrierten 2-Wege-Integration, die Folgendes ermöglicht:

- Zuordnung der Topologieressourcen — AWS DevOps Der Agent erweitert Ihre DevOps Agent Space-Topologie um Entitäten und Beziehungen, die ihm über einen vom Agenten gehosteten Dynatrace MCP-Server zur Verfügung stehen. AWS DevOps
- Automatisierte Auslösung von Untersuchungen — Dynatrace-Workflows können so konfiguriert werden, dass sie Incident Solution Investigations aufgrund von Dynatrace-Problemen auslösen.
- Telemetrie-Introspektion — AWS DevOps Der Agent kann die Dynatrace-Telemetrie überprüfen, während er ein Problem über den vom Agenten gehosteten Dynatrace MCP-Server untersucht. AWS DevOps
- Status-Updates — Der AWS DevOps Agent veröffentlicht wichtige Untersuchungsergebnisse, Ursachenanalysen und generierte Pläne zur Schadensbegrenzung auf der Dynatrace-Benutzeroberfläche.

Weitere Informationen zu bidirektionalen Integrationen finden Sie unter

- the section called “Dynatrace verbinden”

Integrierte 1-Wege-Integration

Derzeit unterstützt AWS DevOps Agent Amazon S3 AWS CloudWatch -, Datadog-, Grafana-, New Relic- und Splunk-Benutzer mit integrierten 1-Wege-Integrationen.

Bewährte Sicherheitspraxis: Bei der Konfiguration von Anmeldeinformationen für integrierte einseitige Integrationen empfehlen wir, API-Schlüssel und Token auf schreibgeschützten Zugriff zu beschränken. AWS DevOps Der Agent verwendet diese Anmeldeinformationen nur für die Telemetrie-Introspektion und benötigt keinen Schreibzugriff auf Ihren Telemetrieanbieter.

Die AWS CloudWatch integrierte 1-Wege-Integration erfordert keine zusätzliche Einrichtung und ermöglicht Folgendes:

- Topologie-Ressourcenzuweisung — AWS DevOps Der Agent erweitert Ihre DevOps Agent Space-Topologie um Entitäten und Beziehungen, die ihm über Ihre konfigurierten primären und sekundären Cloud-Konten zur Verfügung stehen. AWS

- Telemetrie-Introspektion — AWS DevOps Der Agent kann bei der Untersuchung eines Problems mithilfe der IAM-Rolle (n), die bei der Konfiguration des primären und sekundären Cloud-Kontos bereitgestellt wurden, die AWS CloudWatch Telemetrie überprüfen. AWS

Die integrierte 1-Wege-Integration von Amazon S3 ermöglicht Folgendes:

- Telemetrie-Introspektion — AWS DevOps Der Agent kann Objekte aus Amazon S3 S3-Buckets lesen, während er ein Problem untersucht. Dies ist nützlich für den Zugriff auf Protokolle, Konfigurationsdateien und andere in S3 gespeicherte Artefakte.

Um die Amazon S3 S3-Integration zu verwenden, fügen Sie der IAM-Rolle des DevOps Agenten die `s3:ListBucket` Berechtigungen `s3:GetObject` und hinzu. Halten Sie sich an das Prinzip der geringsten Rechte und beschränken Sie diese Berechtigungen nur auf die spezifischen S3-Buckets, auf die der Agent zugreifen muss. Weitere Informationen zur Konfiguration von IAM-Berechtigungen finden Sie unter. the section called “DevOps IAM-Berechtigungen für Agenten”

Die integrierten 1-Wege-Integrationen von Datadog, Grafana, New Relic und Splunk müssen eingerichtet werden und ermöglichen Folgendes:

- Automatisierte Auslösung von Ermittlungen — Datadog-, Grafana-, New Relic- und Splunk-Ereignisse können so konfiguriert werden, dass sie Untersuchungen zur Behebung von Agentenvorfällen über AWS DevOps Agent-Webhooks auslösen. AWS DevOps
- Telemetrie-Introspektion — AWS DevOps Der Agent kann die Telemetrie von Datadog, Grafana, New Relic und Splunk überprüfen, während er ein Problem über den Remote-MCP-Server jedes Anbieters untersucht.

Weitere Informationen zu einseitigen Integrationen finden Sie im Folgenden:

- the section called “Verbindung herstellen DataDog”
- the section called “Grafana verbinden”
- the section called “New Relic verbinden”
- the section called “Splunk verbinden”

Bring-your-own Telemetriequellen

Für jede andere Telemetriequelle, einschließlich Prometheus-Metriken, können Sie die Unterstützung von AWS DevOps Agent sowohl für die Webhook- als auch für die MCP-Serverintegration nutzen.

Weitere Informationen bring-your-own zu Integrationen finden Sie im Folgenden

- the section called “DevOps Agent über Webhook aufrufen”
- the section called “MCP-Server verbinden”

Dynatrace verbinden

Built-in, 2-Wege-Integration

Derzeit unterstützt AWS DevOps Agent Dynatrace-Benutzer mit einer integrierten 2-Wege-Integration, die Folgendes ermöglicht:

- Zuordnung von Topologie-Ressourcen — AWS DevOps Agent erweitert Ihre DevOps Agent Space-Topologie um Entitäten und Beziehungen, die ihm in Ihrer Dynatrace-Umgebung zur Verfügung stehen.
- Automatisierte Auslösung von Untersuchungen — Dynatrace-Workflows können so konfiguriert werden, dass sie die Lösung von Vorfällen auslösen. Untersuchungen aufgrund von Dynatrace-Problemen.
- Telemetrie-Introspektion — AWS DevOps Der Agent kann die Dynatrace-Telemetrie überprüfen, während er ein Problem über den Dynatrace MCP-Server untersucht. AWS DevOps Agent-hosted
- Status-Updates — Der AWS DevOps Agent veröffentlicht wichtige Untersuchungsergebnisse, Ursachenanalysen und generierte Pläne zur Schadensbegrenzung auf der Dynatrace-Benutzeroberfläche.

Voraussetzungen

Für die AWS DevOps Agentenintegration mit Dynatrace ist Dynatrace SaaS erforderlich. Die Integration hängt von den Funktionen der Dynatrace-Plattform (Workflows, AppEngine Apps einschließlich der SRE Agents-App und OAuth-Clients) ab, die nur in Dynatrace SaaS-Umgebungen verfügbar sind.

Dynatrace Managed (lokal) wird nicht unterstützt, und Dynatrace hat nicht vor, diese Plattformfunktionen in Managed zu integrieren. Wenn Sie Dynatrace Managed verwenden, müssen Sie ein Upgrade auf Dynatrace SaaS durchführen, bevor Sie es mit Agent verbinden können. AWS DevOps Siehe Upgrade von Dynatrace Managed auf SaaS.

Onboarding

Onboarding-Prozess

Das Onboarding Ihres Dynatrace-Observability-Systems umfasst drei Phasen:

1. Connect — Stellen Sie eine Verbindung zu Dynatrace her, indem Sie die Zugangsdaten für Ihr Konto konfigurieren, mit allen Umgebungen, die Sie benötigen
2. Aktivieren — Aktivieren Sie Dynatrace in bestimmten Agentenbereichen mit bestimmten Dynatrace-Umgebungen
3. Konfigurieren Sie Ihre Dynatrace-Umgebung — verwenden Sie die Dynatrace SRE Agents App, um die Verbindung mit 2 Klicks herzustellen

Schritt 1: Connect

Stellen Sie eine Verbindung zu Ihrer Dynatrace-Umgebung her

Konfiguration

1. Gehen Sie zur Seite Capability Providers (zugänglich über die Seitennavigation)
2. Suchen Sie im Bereich Verfügbare Anbieter unter Telemetrie nach Dynatrace und klicken Sie auf Registrieren
3. Erstellen Sie einen OAuth-Client in Dynatrace mit den detaillierten Berechtigungen.
   a. Weitere Informationen finden Sie in der Dynatrace-Dokumentation
   b. Wenn Sie bereit sind, drücken Sie Weiter
   c. Sie können mehrere Dynatrace-Umgebungen verbinden und später für jeden DevOps Agent Space, über den Sie verfügen, bestimmte Umgebungen zuordnen.
4. Geben Sie Ihre Dynatrace-Daten aus dem OAuth-Client-Setup ein:
   - Name des Kunden
   - Kunden-ID
   - Geheimes Kundenkonto
   - Konto-URN

5. Klicken Sie auf Weiter
6. Überprüfen und hinzufügen

Schritt 2: Aktivieren

Aktivieren Sie Dynatrace in einem bestimmten Agent-Bereich und konfigurieren Sie das entsprechende Scoping

Konfiguration

1. Wählen Sie auf der Seite „Agentenbereiche“ einen Agentenbereich aus und klicken Sie auf „Details anzeigen“
2. Wählen Sie die Registerkarte Funktionen
3. Suchen Sie den Bereich Telemetrie und drücken Sie auf Hinzufügen
4. Sie werden feststellen, dass Dynatrace den Status „Registriert“ hat. Klicken Sie auf Hinzufügen, um dies zu Ihrem Agentenbereich hinzuzufügen
5. Dynatrace-Umgebungs-ID — Geben Sie die Dynatrace-Umgebungs-ID ein, die Sie diesem Agentenbereich zuordnen möchten. DevOps
6. Geben Sie eine oder mehrere Dynatrace-Entitäts-IDs ein — diese helfen dem DevOps Agenten dabei, Ihre wichtigsten Ressourcen zu finden, beispielsweise Dienste oder Anwendungen. Wenn Sie sich nicht sicher sind, können Sie auf Entfernen klicken.
7. Überprüfe und drücke auf Speichern
8. Kopieren Sie die Webhook-URL und das Webhook-Secret. Sie werden diese in der Dynatrace SRE Agents App verwenden, um die Verbindung herzustellen. Einzelheiten finden Sie im Abschnitt Erste Schritte.

Schritt 3: Konfigurieren Sie Ihre Dynatrace-Umgebung

Um Ihr Dynatrace-Setup abzuschließen, verwenden Sie die Dynatrace SRE Agents App, um die Dynatrace-Seite der Integration mit 2 Klicks zu konfigurieren — es ist keine manuelle Workflow-Einrichtung erforderlich. Einzelheiten finden Sie im Abschnitt Erste Schritte.

Unterstützte Ereignisschemas

AWS DevOps Der Agent unterstützt zwei Arten von Ereignissen von Dynatrace mithilfe von Webhooks. Die unterstützten Ereignisschemas sind unten dokumentiert:

Ereignis (Ereignis)

Vorfallsereignisse werden verwendet, um eine Untersuchung auszulösen. Das Ereignisschema lautet:

```
{
    "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[];
}
```

Ereignis zur Schadensbegrenzung

Minderungsereignisse werden verwendet, um die Erstellung eines Minderungsberichts für die Untersuchung der nächsten Schritte auszulösen. Das Ereignisschema lautet:

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

Entfernung

Die Telemetriequelle ist auf zwei Ebenen miteinander verbunden, auf der Ebene des Agentenbereichs und auf der Kontoebene. Um sie vollständig zu entfernen, müssen Sie sie zunächst aus allen Agentenbereichen entfernen, in denen sie verwendet wird. Anschließend kann die Registrierung aufgehoben werden.

Schritt 1: Aus dem Agentenbereich entfernen

1. Wählen Sie auf der Seite „Agentenbereiche“ einen Agentbereich aus und klicken Sie auf „Details anzeigen“
2. Wählen Sie die Registerkarte Funktionen
3. Scrollen Sie nach unten zum Abschnitt Telemetrie
4. Wählen Sie Dynatrace

5. Drücken Sie auf Entfernen

Schritt 2: Vom Konto abmelden

1. Gehen Sie zur Seite Capability Providers (zugänglich über die Seitennavigation)
2. Scrollen Sie zum Abschnitt Aktuell registriert.
3. Vergewissern Sie sich, dass die Anzahl der Agentenplätze Null ist (falls nicht, wiederholen Sie Schritt 1 oben in Ihren anderen Agentenbereichen)
4. Klicken Sie neben Dynatrace auf Abmelden

Verbindung herstellen DataDog

Built-in, 1-Wege-Integration

Derzeit unterstützt AWS DevOps Agent Datadog-Benutzer mit einer integrierten 1-Wege-Integration, die Folgendes ermöglicht:

- Automatisierte Auslösung von Ermittlungen — Datadog-Ereignisse können so konfiguriert werden, dass sie Untersuchungen zur Behebung von AWS DevOps Agentenvorfällen über Agent-Webhooks auslösen. AWS DevOps
- Telemetrie-Introspektion — AWS DevOps Der Agent kann die Datadog-Telemetrie überprüfen, während er ein Problem über den Remote-MCP-Server jedes Anbieters untersucht.

Onboarding

Schritt 1: Connect

Stellen Sie mit den Zugangsdaten für Ihr Konto eine Verbindung zu Ihrem DataDog-Remote-MCP-Endpunkt her

Konfiguration

1. Gehen Sie zur Seite Capability Providers (zugänglich über die Seitennavigation)
2. Suchen Sie Datadog im Bereich Verfügbare Anbieter unter Telemetrie und klicken Sie auf Registrieren
3. Geben Sie Ihre Datadog MCP-Serverdetails ein:
   - Servername — Eindeutiger Bezeichner (z. B. my-datadog-server)

- Endpunkt-URL — Ihr Datadog MCP-Serverendpunkt. Die Endpunkt-URL variiert je nach Ihrer Datadog-Site. Die Endpunkttabelle der Datadog-Site finden Sie unten.
- Beschreibung — Optionale Serverbeschreibung

4. Klicken Sie auf Weiter
5. Überprüfen und Einreichen

Endpunkte der Datadog-Site

Die MCP-Endpunkt-URL variiert je nach Ihrer Datadog-Site. Um Ihre Site zu identifizieren, überprüfen Sie die URL in Ihrem Browser, wenn Sie bei Datadog angemeldet sind, oder lesen Sie auf die Datadog-Website zugreifen.

Datadog-Seite	Domäne der Website	MCP-Endpunkt-URL
US1 (Standard)	`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`
UNS 5	`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`
AP 1	`ap1.datadoghq.com`	`https://mcp.ap1.datadoghq.com/api/unstable/mcp-server/mcp`

Datadog-Seite	Domäne der Website	MCP-Endpunkt-URL
AP 2	`ap2.datadoghq.com`	`https://mcp.ap2.datadoghq.com/api/unstable/mcp-server/mcp`

Autorisierung

Vollständige OAuth-Autorisierung durch:

- Autorisieren Sie sich als Ihr Benutzer auf der Datadog OAuth-Seite
- Wenn Sie nicht angemeldet sind, klicken Sie auf Zulassen, Anmelden und dann auf Autorisieren

Nach der Konfiguration ist Datadog in allen Agentenbereichen verfügbar.

Schritt 2: Aktivieren

DataDog In einem bestimmten Agent-Bereich aktivieren und den entsprechenden Geltungsbereich konfigurieren

Konfiguration

1. Wählen Sie auf der Seite für Agentenbereiche einen Agentenbereich aus und klicken Sie auf Details anzeigen (falls Sie noch keinen Agentenbereich erstellt haben, siehe) the section called “Einen Agentenbereich erstellen”
2. Wählen Sie die Registerkarte Funktionen
3. Scrollen Sie nach unten zum Abschnitt Telemetrie
4. Drücken Sie auf Hinzufügen
5. Wählen Sie Datadog
6. Next
7. Überprüfe und drücke Speichern
8. Kopieren Sie die Webhook-URL und den API-Schlüssel

Schritt 3: Webhooks konfigurieren

Mithilfe der Webhook-URL und des API-Schlüssels können Sie Datadog so konfigurieren, dass Ereignisse gesendet werden, um eine Untersuchung auszulösen, beispielsweise aufgrund eines Alarms.

Datadog-Webhooks verwenden die Bearer-Token-Authentifizierung. Das vollständige Webhook-Anforderungsformat, das Payload-Schema und den Beispielcode finden Sie unter. the section called "DevOps Agent über Webhook aufrufen" Verwenden Sie die Beispiele für Version 2 (Bearer-Token-Authentifizierung) und legen Sie den `Authorization: Bearer <Token>` Header mit dem API-Schlüssel aus Schritt 2 fest.

Senden Sie Webhooks mit Datadog https://docs.datadoghq.com/integrations/webhooks/(beachten Sie, dass Sie keine Autorisierung auswählen und stattdessen die benutzerdefinierte Header-Option verwenden).

Erfahren Sie mehr: Datadog Remote MCP Server

Entfernung

Die Telemetriequelle ist auf zwei Ebenen miteinander verbunden, auf der Ebene des Agentenbereichs und auf der Kontoebene. Um sie vollständig zu entfernen, müssen Sie sie zunächst aus allen Agentenbereichen entfernen, in denen sie verwendet wird. Anschließend kann die Registrierung aufgehoben werden.

Schritt 1: Aus dem Agentenbereich entfernen

1. Wählen Sie auf der Seite „Agentenbereiche“ einen Agentbereich aus und klicken Sie auf „Details anzeigen“
2. Wählen Sie die Registerkarte Funktionen
3. Scrollen Sie nach unten zum Abschnitt Telemetrie
4. Wählen Sie Datadog
5. Drücken Sie auf Entfernen

Schritt 2: Vom Konto abmelden

1. Gehen Sie zur Seite Capability Providers (zugänglich über die Seitennavigation)
2. Scrollen Sie zum Abschnitt Aktuell registriert.

3. Vergewissern Sie sich, dass die Anzahl der Agentenplätze Null ist (falls nicht, wiederholen Sie Schritt 1 oben in Ihren anderen Agentenbereichen)
4. Drücken Sie neben Datadog auf Abmelden

Grafana verbinden

Die Grafana-Integration ermöglicht es dem AWS DevOps Agenten, bei der Untersuchung von Vorfällen Metriken, Dashboards und Warndaten aus Ihrer Grafana-Instanz abzufragen. Diese Integration folgt einem zweistufigen Prozess: Registrierung von Grafana auf Kontoebene, gefolgt von der Verbindung mit einzelnen Agent Spaces.

Um die Sicherheit zu verbessern, ermöglicht die Grafana-Integration nur schreibgeschützte Tools. Schreibwerkzeuge sind deaktiviert und können nicht aktiviert werden. Das bedeutet, dass der Agent Daten aus Ihrer Grafana-Instanz abfragen und lesen kann, jedoch keine Grafana-Ressourcen wie Dashboards, Benachrichtigungen oder Anmerkungen erstellen, ändern oder löschen kann. Weitere Informationen finden Sie unter Sicherheit im Agent. AWS DevOps

Grafana-Anforderungen

Bevor Sie Grafana verbinden, stellen Sie sicher, dass Sie über Folgendes verfügen:

- Grafana-Version 9.0 oder höher. Einige Funktionen, insbesondere datenquellenbezogene Operationen, funktionieren in früheren Versionen aufgrund fehlender API-Endpunkte möglicherweise nicht richtig.
- Eine Grafana-Instanz, auf die über HTTPS zugegriffen werden kann. Sowohl öffentliche als auch private Netzwerkendpunkte werden unterstützt. Mit privater Netzwerkkonnektivität kann Ihre Grafana-Instanz in einer VPC ohne öffentlichen Internetzugang gehostet werden. Details hierzu finden Sie unter the section called “Verbindung zu privat gehosteten Tools herstellen”.
- Ein Grafana-Dienstkonto mit einem Zugriffstoken, das über entsprechende Leseberechtigungen verfügt

Registrierung von Grafana (Kontoebene)

Grafana ist auf AWS Kontoebene registriert und wird von allen Agent Spaces in diesem Konto gemeinsam genutzt.

Schritt 1: Grafana konfigurieren

1. Melden Sie sich bei der AWS Management Console an
2. Navigieren Sie zur AWS DevOps Agent-Konsole
3. Gehen Sie zur Seite Capability Providers (zugänglich über die Seitennavigation)
4. Suchen Sie Grafana im Bereich Verfügbare Anbieter unter Telemetrie und klicken Sie auf Registrieren
5. Geben Sie auf der Seite Grafana konfigurieren die folgenden Informationen ein:
   - Dienstname (erforderlich) — Geben Sie einen beschreibenden Namen für Ihren Grafana-Server ein, der nur alphanumerische Zeichen, Bindestriche und Unterstriche verwendet. Beispiel, `my-grafana-server`.
   - Grafana-URL (erforderlich) — Geben Sie die vollständige HTTPS-URL Ihrer Grafana-Instanz ein. Beispiel, `https://myinstance.grafana.net`.
   - Zugriffstoken für Dienstkonten (erforderlich) — Geben Sie ein Zugriffstoken für das Grafana-Dienstkonto ein. Token beginnen normalerweise mit`glsa_`. Um ein Dienstkonto-Token zu erstellen, navigieren Sie zu Ihrer Grafana-Instanz, gehen Sie zu Administration > Dienstkonten, erstellen Sie ein Dienstkonto mit Viewer-Rolle und generieren Sie ein Token.
   - Beschreibung (optional) — Fügen Sie eine Beschreibung hinzu, um den Zweck des Servers zu identifizieren. Beispiel, `Production Grafana server for monitoring`.
6. (Optional) Fügen Sie der Registrierung aus organisatorischen Gründen AWS Tags hinzu.
7. Klicken Sie auf Weiter

Schritt 2: Grafana-Registrierung überprüfen und einreichen

1. Überprüfen Sie alle Grafana-Konfigurationsdetails
2. Klicken Sie auf Senden, um die Registrierung abzuschließen
3. Nach erfolgreicher Registrierung wird Grafana im Bereich Aktuell registriert auf der Seite Capability Providers angezeigt

Grafana zu einem Agent Space hinzufügen

Nachdem Sie Grafana auf Kontoebene registriert haben, können Sie es mit einzelnen Agent Spaces verbinden:

1. Wählen Sie in der AWS DevOps Agent-Konsole Ihren Agent Space aus

2. Gehen Sie zur Registerkarte Funktionen
3. Klicken Sie im Bereich Telemetrie auf Hinzufügen
4. Wählen Sie Grafana aus der Liste der verfügbaren Anbieter
5. Klicken Sie auf Speichern

Konfiguration von Grafana-Alert-Webhooks

Sie können Grafana so konfigurieren, dass automatisch AWS DevOps Agentenuntersuchungen ausgelöst werden, wenn Warnmeldungen ausgelöst werden, indem Webhooks über Grafana-Kontaktpunkte gesendet werden. Einzelheiten zu Webhook-Authentifizierungsmethoden und zur Verwaltung von Anmeldeinformationen finden Sie unter. the section called “DevOps Agent über Webhook aufrufen”

Schritt 1: Erstellen Sie eine benutzerdefinierte Benachrichtigungsvorlage

Navigieren Sie in Ihrer Grafana-Instanz zu Alerting > Kontaktstellen > Benachrichtigungsvorlagen und erstellen Sie eine neue Vorlage mit dem folgenden Inhalt:

```
{{ 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 }}
```

Diese Vorlage formatiert Grafana-Benachrichtigungen in die vom Agenten erwartete Webhook-Payload-Struktur. AWS DevOps Sie ordnet Warnbezeichnungen, Anmerkungen und Status den entsprechenden Feldern zu und enthält alle Warnmeldungsbeschriftungen als Metadaten.

> Hinweis: Diese Vorlage verarbeitet nur die erste Warnung in einer Gruppe. Grafana gruppiert standardmäßig mehrere Feuerwarnungen in einer einzigen Benachrichtigung. Um sicherzustellen, dass jede Warnung einzeln gesendet wird, konfigurieren Sie Ihre Benachrichtigungsrichtlinien so, dass sie nach Gruppen gruppiert werden`alertname`. Darüber hinaus enthält diese Vorlage keine JSON-Sonderzeichen in Labelwerten oder Anmerkungen. Stellen Sie sicher, dass Warnungsbeschriftungen und die `summary` Anmerkung keine Zeichen wie doppelte Anführungszeichen oder Zeilenumbrüche enthalten, da dies zu ungültigem JSON führen würde.

Schritt 2: Erstellen Sie eine Webhook-Kontaktstelle

1. Navigieren Sie in Grafana zu Alerting > Kontaktstellen und klicken Sie auf Kontaktstelle hinzufügen
2. Wählen Sie Webhook als Integrationstyp
3. Stellen Sie die URL auf Ihren AWS DevOps Agent-Webhook-Endpunkt ein
4. Konfigurieren Sie unter Optionale Webhook-Einstellungen die Authentifizierungs-Header basierend auf Ihrem Webhook-Typ. Einzelheiten finden Sie unter Webhook-Authentifizierungsmethoden.
5. Stellen Sie das Feld Benutzerdefinierter Payload so ein, dass Ihre benutzerdefinierte Vorlage verwendet wird: `{{ template "devops-agent-payload" . }}`
6. Klicken Sie auf Kontaktstelle speichern

Schritt 3: Weisen Sie den Kontaktpunkt einer Benachrichtigungsrichtlinie zu

1. Navigieren Sie zu Alerting > Benachrichtigungsrichtlinien
2. Bearbeiten Sie eine bestehende Richtlinie oder erstellen Sie eine neue
3. Stellen Sie den Kontaktpunkt auf den Webhook-Kontaktpunkt ein, den Sie erstellt haben
4. Klicken Sie auf Richtlinie speichern

Wenn eine passende Warnung ausgelöst wird, sendet Grafana die formatierte Nutzlast an den AWS DevOps Agenten, der automatisch eine Untersuchung einleitet.

Einschränkungen

- ClickHouse Datenquellentools — ClickHouse Datenquellentools werden derzeit nicht unterstützt.
- Proaktive Prävention von Vorfällen — verwendet derzeit the section called “Proaktive Prävention von Zwischenfällen” keine Grafana-Tools. Support ist für eine future Version geplant.

Überlegungen zu Amazon Managed Grafana

Wenn Sie Amazon Managed Grafana (AMG) verwenden, beachten Sie die folgenden Einschränkungen:

- Webhook-Kontaktpunkte werden nicht unterstützt — AMG unterstützt derzeit keine Webhook-Kontaktpunkte in seiner Warnkonfiguration. Sie können AMG nicht verwenden, um Warnungs-Webhooks direkt an den Agenten zu senden. AWS DevOps Einzelheiten finden Sie unter Benachrichtigungen an Kontaktstellen in Amazon Managed Grafana.
- Ablauf der Dienstkonto-Tokens — AMG-Servicekonto-Token haben eine maximale Gültigkeitsdauer von 30 Tagen. Sie müssen die Tokens rotieren und Ihre Grafana-Registrierung in AWS DevOps Agent aktualisieren, bevor sie ablaufen. Informationen zum Aktualisieren der Anmeldeinformationen finden Sie unter Grafana-Verbindungen verwalten. Einzelheiten zu AMG-Token-Limits finden Sie unter Dienstkonten in Amazon Managed Grafana.

Verwaltung von Grafana-Verbindungen

- Anmeldeinformationen aktualisieren — Wenn Ihr Dienstkonto-Token abläuft oder aktualisiert werden muss, melden Sie Grafana auf der Seite Capability Providers ab und registrieren Sie sich erneut mit dem neuen Token.
- Verbundene Instanzen anzeigen — Wählen Sie in der AWS DevOps Agentenkonsole Ihren Agent Space aus und wechseln Sie zur Registerkarte Funktionen, um die verbundenen Telemetriequellen anzuzeigen.
- Grafana entfernen — Um Grafana von einem Agent Space zu trennen, wählen Sie es im Abschnitt Telemetrie aus und klicken Sie auf Entfernen. Um die Registrierung vollständig zu entfernen, entfernen Sie sie zuerst aus allen Agent Spaces und melden Sie sich dann auf der Seite Capability Providers ab.

New Relic verbinden

Built-in, Einwegintegration

Derzeit unterstützt AWS DevOps Agent New Relic-Benutzer mit einer integrierten 1-Wege-Integration, die Folgendes ermöglicht:

- Automatisierte Auslösung von Ermittlungen — New Relic-Ereignisse können so konfiguriert werden, dass über Agent-Webhooks Untersuchungen zur Behebung von Vorfällen durch AWS DevOps AWS DevOps Agenten ausgelöst werden.
- Telemetrie-Introspektion — AWS DevOps Der Agent kann die New Relic-Telemetrie überprüfen, während er ein Problem über den Remote-MCP-Server jedes Anbieters untersucht.

Onboarding

Schritt 1: Connect

Stellen Sie mit den Zugangsdaten für Ihr Konto eine Verbindung zu Ihrem New Relic Remote-MCP-Endpunkt her

Bitte verwenden Sie einen Vollplattformbenutzer (nicht Basic/Core) in New Relic, um die New Relic MCP-Tools zu aktivieren.

Konfiguration

1. Gehen Sie zur Seite Capability Providers (zugänglich über die Seitennavigation)
2. Suchen Sie New Relic im Bereich Verfügbare Anbieter unter Telemetrie und klicken Sie auf Registrieren
3. Folgen Sie den Anweisungen, um Ihren New Relic API-Schlüssel zu erhalten
4. Geben Sie die Details Ihres New Relic MCP Server-API-Schlüssels ein:
   - Konto-ID: Geben Sie Ihre New Relic-Konto-ID ein, die Sie oben erhalten haben
   - API-Schlüssel: Geben Sie den oben erhaltenen API-Schlüssel ein
   - Wählen Sie je nachdem, wo sich Ihr New Relic-Konto befindet, die Region USA oder EU aus.
5. Klicken Sie auf Hinzufügen

Schritt 2: Aktivieren

Aktivieren Sie New Relic in einem bestimmten Agent-Bereich und konfigurieren Sie das entsprechende Scoping

Konfiguration

1. Wählen Sie auf der Seite für Agentenbereiche einen Agentenbereich aus und klicken Sie auf Details anzeigen (falls Sie noch keinen Agentenbereich erstellt haben, siehe) the section called "Einen Agentenbereich erstellen"
2. Wählen Sie die Registerkarte Funktionen
3. Scrollen Sie nach unten zum Abschnitt Telemetrie
4. Drücken Sie auf Hinzufügen
5. Wählen Sie New Relic
6. Next
7. Überprüfe es und drücke auf Speichern
8. Kopieren Sie die Webhook-URL und den API-Schlüssel

Schritt 3: Webhooks konfigurieren

Mithilfe der Webhook-URL und des API-Schlüssels können Sie New Relic so konfigurieren, dass Ereignisse gesendet werden, um eine Untersuchung auszulösen, beispielsweise aufgrund eines Alarms. Weitere Informationen zur Einrichtung von Webhooks finden Sie unter Tracking-Webhooks ändern.

New Relic Webhooks verwenden die Bearer-Token-Authentifizierung. Das vollständige Webhook-Anforderungsformat, das Payload-Schema und den Beispielcode finden Sie unter. the section called "DevOps Agent über Webhook aufrufen" Verwenden Sie die Beispiele für Version 2 (Bearer-Token-Authentifizierung) und legen Sie den `Authorization: Bearer <Token>` Header mit dem API-Schlüssel aus Schritt 2 fest.

Senden Sie Webhooks mit New Relic. https://newrelic.com/instant-observability/webhook-notifications Sie können entweder Bearer-Token als Autorisierungstyp auswählen oder keine Autorisierung auswählen und stattdessen das `Authorization: Bearer <Token>` als benutzerdefinierten Header hinzufügen.

Erfahren Sie mehr: https://docs.newrelic.com/docs/agentic-ai/mcp/overview/

Entfernung

Die Telemetriequelle ist auf zwei Ebenen miteinander verbunden, auf der Ebene des Agentenbereichs und auf der Kontoebene. Um sie vollständig zu entfernen, müssen Sie sie zunächst aus allen Agentenbereichen entfernen, in denen sie verwendet wird. Anschließend kann die Registrierung aufgehoben werden.

Schritt 1: Aus dem Agentenbereich entfernen

1. Wählen Sie auf der Seite „Agentenbereiche“ einen Agentbereich aus und klicken Sie auf „Details anzeigen“
2. Wählen Sie die Registerkarte Funktionen
3. Scrollen Sie nach unten zum Abschnitt Telemetrie
4. Wählen Sie New Relic
5. Drücken Sie auf Entfernen

Schritt 2: Vom Konto abmelden

1. Gehen Sie zur Seite Capability Providers (zugänglich über die Seitennavigation)
2. Scrollen Sie zum Abschnitt Aktuell registriert.
3. Vergewissern Sie sich, dass die Anzahl der Agentenplätze Null ist (falls nicht, wiederholen Sie Schritt 1 oben in Ihren anderen Agentenbereichen)
4. Klicken Sie neben New Relic auf Abmelden

Splunk verbinden

Built-in, Einweg-Integration

Derzeit unterstützt AWS DevOps Agent Splunk-Benutzer mit einer integrierten 1-Wege-Integration, die Folgendes ermöglicht:

- Automatisierte Auslösung von Ermittlungen — Splunk-Ereignisse können so konfiguriert werden, dass sie Untersuchungen zur Behebung von AWS DevOps Agentenvorfällen über Agenten-Webhooks auslösen. AWS DevOps
- Telemetrie-Introspektion — AWS DevOps Der Agent kann die Splunk-Telemetrie während der Untersuchung eines Problems über den Remote-MCP-Server jedes Anbieters überprüfen.

Voraussetzungen

Ein Splunk-API-Token abrufen

Sie benötigen eine MCP-URL und ein Token, um Splunk zu verbinden.

Schritte des Splunk-Administrators

Ihr Splunk-Administrator muss die folgenden Schritte ausführen:

- REST-API-Zugriff aktivieren
- aktivieren Sie die Token-Authentifizierung für die Bereitstellung.
- erstellen Sie eine neue Rolle 'mcp_user', die neue Rolle muss keine Funktionen haben.
- weisen Sie die Rolle 'mcp_user' allen Benutzern in der Bereitstellung zu, die berechtigt sind, den MCP-Server zu verwenden.
- erstellen Sie das Token für die autorisierten Benutzer mit der Zielgruppe „mcp“ und legen Sie das entsprechende Ablaufdatum fest, falls der Benutzer nicht berechtigt ist, selbst Token zu erstellen.

Schritte für Splunk-Benutzer

Ein Splunk-Benutzer muss die folgenden Schritte ausführen:

- Besorgen Sie sich ein entsprechendes Token vom Splunk-Administrator oder erstellen Sie selbst eines, sofern er die entsprechende Erlaubnis hat. Die Zielgruppe für das Token muss „mcp“ sein.

Onboarding

Schritt 1: Connect

Stellen Sie mit den Zugangsdaten für Ihr Konto eine Verbindung zu Ihrem Splunk-Remote-MCP-Endpunkt her

Konfiguration

1. Gehen Sie zur Seite Capability Providers (zugänglich über die Seitennavigation)
2. Suchen Sie Splunk im Bereich Verfügbare Anbieter unter Telemetrie und klicken Sie auf Registrieren

3. Geben Sie Ihre Splunk MCP-Serverdetails ein:
   - Servername — Eindeutiger Bezeichner (z. B. my-splunk-server)
   - Endpunkt-URL — Ihr Splunk MCP-Serverendpunkt:

```
https://<YOUR_SPLUNK_DEPLOYMENT_NAME>.api.scs.splunk.com/
<YOUR_SPLUNK_DEPLOYMENT_NAME>/mcp/v1/
```

- Beschreibung — Optionale Serverbeschreibung
- Tokenname — Der Name des Inhaber-Tokens für die Authentifizierung: `my-splunk-token`
- Token-Wert: Der Wert des Bearer-Tokens für die Authentifizierung

Schritt 2: Aktivieren

Aktivieren Sie Splunk in einem bestimmten Agent-Bereich und konfigurieren Sie das entsprechende Scoping

Konfiguration

1. Wählen Sie auf der Seite „Agentenbereiche“ einen Agentenbereich aus und klicken Sie auf „Details anzeigen“ (falls Sie noch keinen Agentenbereich erstellt haben, finden Sie unter) the section called “Einen Agentenbereich erstellen”
2. Wählen Sie die Registerkarte Funktionen
3. Scrollen Sie nach unten zum Abschnitt Telemetrie
4. Drücken Sie auf Hinzufügen
5. Wählen Sie Splunk
6. Next
7. Überprüfen Sie es und drücken Sie auf Speichern
8. Kopieren Sie die Webhook-URL und den API-Schlüssel

Schritt 3: Webhooks konfigurieren

Mithilfe der Webhook-URL und des API-Schlüssels können Sie Splunk so konfigurieren, dass Ereignisse gesendet werden, um eine Untersuchung auszulösen, beispielsweise aufgrund eines Alarms.

Splunk-Webhooks verwenden die Bearer-Token-Authentifizierung. Das vollständige Webhook-Anforderungsformat, das Payload-Schema und den Beispielcode finden Sie unter. the section called "DevOps Agent über Webhook aufrufen" Verwenden Sie die Beispiele für Version 2 (Bearer-Token-Authentifizierung) und legen Sie den `Authorization: Bearer <Token>` Header mit dem API-Schlüssel aus Schritt 2 fest.

Senden Sie Webhooks mit Splunk https://help.splunk.com/en/splunk-enterprise/alert-and-respond/alerting-manual/9.4/configure-alert-actions/use-a-webhook-alert-action(beachten Sie, dass Sie „Keine Autorisierung" auswählen und stattdessen die Option „Benutzerdefinierter Header" verwenden)

Weitere Informationen:

- Dokumentation zum MCP-Server von Splunk: https://help.splunk.com/en/splunk-cloud-platform/mcp-server-for-splunk-platform/about-mcp-server-for-splunk-platform
- Zugriffsanforderungen und Einschränkungen für die Splunk Cloud Platform REST API: https://docs.splunk.com/Documentation/SplunkCloud/latest/RESTTUT/RESTandCloud
- Authentifizierungstoken auf der Splunk Cloud Platform verwalten: 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
- Rollen mit Splunk Web erstellen und verwalten: https://docs.splunk.com/Documentation/SplunkCloud/latest/Security/Addandeditroles

Entfernung

Die Telemetriequelle ist auf zwei Ebenen miteinander verbunden, auf der Ebene des Agentenbereichs und auf der Kontoebene. Um sie vollständig zu entfernen, müssen Sie sie zunächst aus allen Agentenbereichen entfernen, in denen sie verwendet wird. Anschließend kann die Registrierung aufgehoben werden.

Schritt 1: Aus dem Agentenbereich entfernen

1. Wählen Sie auf der Seite „Agentenbereiche" einen Agentbereich aus und klicken Sie auf „Details anzeigen"
2. Wählen Sie die Registerkarte Funktionen
3. Scrollen Sie nach unten zum Abschnitt Telemetrie
4. Wählen Sie Splunk
5. Drücken Sie auf Entfernen

Schritt 2: Vom Konto abmelden

1. Gehen Sie zur Seite Capability Providers (zugänglich über die Seitennavigation)
2. Scrollen Sie zum Abschnitt Aktuell registriert.
3. Vergewissern Sie sich, dass die Anzahl der Agentenplätze Null ist (falls nicht, wiederholen Sie Schritt 1 oben in Ihren anderen Agentenbereichen)
4. Klicken Sie neben Splunk auf Abmelden

Verbindung zu Ticketing und Chat herstellen

AWS DevOps Der Agent ist so konzipiert, dass er als Mitglied Ihres Teams auftritt, indem er an den bestehenden Kommunikationskanälen Ihres Teams teilnimmt. Sie können DevOps Agent mit Ihren Ticket- und Alarmsystemen verbinden, z. B. um anhand von Incident-Tickets automatisch Untersuchungen einzuleiten ServiceNow und PagerDuty so die Reaktion auf Vorfälle innerhalb Ihrer bestehenden Workflows zu beschleunigen und so die Mean Time to Recovery (MTTR) zu reduzieren. Du kannst deinen DevOps Agenten auch mit deinen Teamzusammenarbeitssystemen wie Slack verbinden, um von deinem Agenten in einem Chat-Kanal Zusammenfassungen der Aktivitäten zu erhalten. DevOps

Weitere Informationen zur Verbindung von Ticketing- und Chat-Integrationen findest du im Folgenden:

- the section called “Verbindung herstellen PagerDuty”
- the section called “Verbindung herstellen ServiceNow”
- the section called “Slack verbinden”

Verbindung herstellen PagerDuty

PagerDuty Die Integration ermöglicht es dem AWS DevOps Agenten, während der Untersuchung von Vorfällen und der automatisierten Reaktion auf Vorfalldaten, Bereitschaftszeiten und Serviceinformationen von Ihrem PagerDuty Konto aus zuzugreifen und diese zu aktualisieren. Diese Integration verwendet OAuth 2.0 für die sichere Authentifizierung.

> **Important**
>
> AWS DevOps Der Agent unterstützt nur die neuere Version PagerDuty OAuth 2.0 (Scoped OAuth). Legacy PagerDuty OAuth mit Umleitungs-URI wird nicht unterstützt.

PagerDuty Anforderungen

Stellen Sie vor dem Herstellen PagerDuty der Verbindung sicher, dass Sie über Folgendes verfügen

- Ein PagerDuty Konto mit Ihrer OAuth Kunden-ID und Ihrem geheimen Kundengeheimnis
- Ihre PagerDuty Konto-Subdomain (wenn Ihre PagerDuty URL beispielsweise lautet`https://your-company.pagerduty.com`, ist die Subdomain) `your-company`

Registrierung PagerDuty

PagerDuty wird auf AWS Kontoebene registriert und von allen Agent Spaces in diesem Konto gemeinsam genutzt.

Schritt 1: Konfigurieren Sie den Zugriff in PagerDuty

1. Melden Sie sich bei der AWS Management Console an
2. Navigieren Sie zur AWS DevOps Agent-Konsole
3. Gehen Sie zur Seite Capability Providers (zugänglich über die Seitennavigation)
4. Suchen Sie PagerDutyim Bereich Verfügbare Anbieter unter Kommunikation nach und klicken Sie auf Registrieren
5. Folgen Sie den Anweisungen zur Einrichtung auf der PagerDuty Seite Zugriff konfigurieren in:

Überprüfen Sie Ihre Serviceregion und Subdomain:

- Kontobereich — Wählen Sie Ihre PagerDuty Region (USA oder EU) aus und geben Sie Ihre PagerDuty Subdomain ein. Wenn Ihre PagerDuty URL beispielsweise lautet`https://your-company.pagerduty.com`, geben Sie ein`your-company`.

Erstellen Sie eine neue App in PagerDuty:

- Melden Sie sich in einem separaten Browser-Tab an PagerDuty und navigieren Sie zu Integrationen > App-Registrierung
- Erstellen Sie eine neue App mit OAuth 2.0 Scoped OAuth
- Gewähren Sie unter Berechtigungen die folgenden erforderlichen Mindestbereiche:`incidents.read`,, und `incidents.write` `services.read`
- Aktivieren Sie die Ereignisintegration, um eine bidirektionale Kommunikation zwischen Agent und AWS DevOps PagerDuty

OAuth Anmeldeinformationen konfigurieren:

- Berechtigungsbereich — Die erforderlichen Mindestbereiche sind:`incidents.read`,, `incidents.write` `services.read`
- Kundenname — Geben Sie einen aussagekräftigen Namen für Ihren Kunden ein OAuth
- Kunden-ID — Geben Sie die OAuth Kunden-ID aus Ihrer PagerDuty App-Registrierung ein
- Kundengeheimnis — Geben Sie das OAuth Client-Geheimnis aus Ihrer PagerDuty App-Registrierung ein

Schritt 2: Überprüfen Sie die PagerDuty Registrierung und senden Sie sie ab

1. Überprüfen Sie alle PagerDuty Konfigurationsdetails
2. Klicken Sie auf Senden, um die Registrierung abzuschließen
3. Wird nach erfolgreicher Registrierung im Bereich Aktuell registriert auf der Seite Capability Providers angezeigt PagerDuty

PagerDuty Zu einem Agentenbereich hinzufügen

Nachdem Sie sich auf PagerDuty Kontoebene registriert haben, können Sie es mit einzelnen Agent Spaces verbinden:

1. Wählen Sie in der AWS DevOps Agent-Konsole Ihren Agent Space aus
2. Gehen Sie zur Registerkarte Funktionen
3. Klicken Sie im Bereich Kommunikation auf Hinzufügen
4. Wählen Sie PagerDutyaus der Liste der verfügbaren Anbieter
5. Klicken Sie auf Speichern

PagerDuty Verbindungen verwalten

- Anmeldeinformationen aktualisieren — Wenn Ihre OAuth Anmeldeinformationen aktualisiert werden müssen, melden Sie sich auf der Seite Capability Providers PagerDuty ab und registrieren Sie sich erneut mit den neuen Anmeldeinformationen.
- Verbindungen anzeigen — Wählen Sie in der AWS DevOps Agent-Konsole Ihren Agent-Bereich aus und wechseln Sie zur Registerkarte Funktionen, um die verbundenen Kommunikationsintegrationen anzuzeigen.
- Entfernen PagerDuty — Um die Verbindung zu einem Agent-Space zu PagerDuty trennen, wählen Sie ihn im Bereich Kommunikation aus und klicken Sie auf Entfernen. Um die Registrierung vollständig zu entfernen, entfernen Sie sie zunächst aus allen Agent Spaces und melden Sie sich dann auf der Seite Capability Providers ab.

Webhook-Unterstützung

AWS DevOps Der Agent unterstützt nur PagerDuty V3-Webhooks. Frühere Webhook-Versionen werden nicht unterstützt.

Weitere Informationen zu PagerDuty V3-Webhook-Abonnements finden Sie in der Entwicklerdokumentation unter Webhooks Overview. PagerDuty

Verbindung herstellen ServiceNow

In diesem Tutorial erfahren Sie, wie Sie eine ServiceNow Instanz mit dem AWS DevOps Agenten verbinden, damit dieser bei der Erstellung eines Tickets automatisch Untersuchungen zur Reaktion auf Vorfälle einleiten und die wichtigsten Ergebnisse im ursprünglichen Ticket veröffentlichen kann. Es enthält auch Beispiele dafür, wie Sie Ihre ServiceNow Instanz so konfigurieren, dass nur bestimmte Tickets an einen DevOps Agent Space gesendet werden, und wie Sie das Ticket-Routing über mehrere DevOps Agent Spaces hinweg orchestrieren.

Ersteinrichtung

Der erste Schritt besteht darin, ServiceNow einen OAuth-Anwendungsclient zu erstellen, mit dem Sie auf Ihre ServiceNow Instanz zugreifen AWS DevOps können.

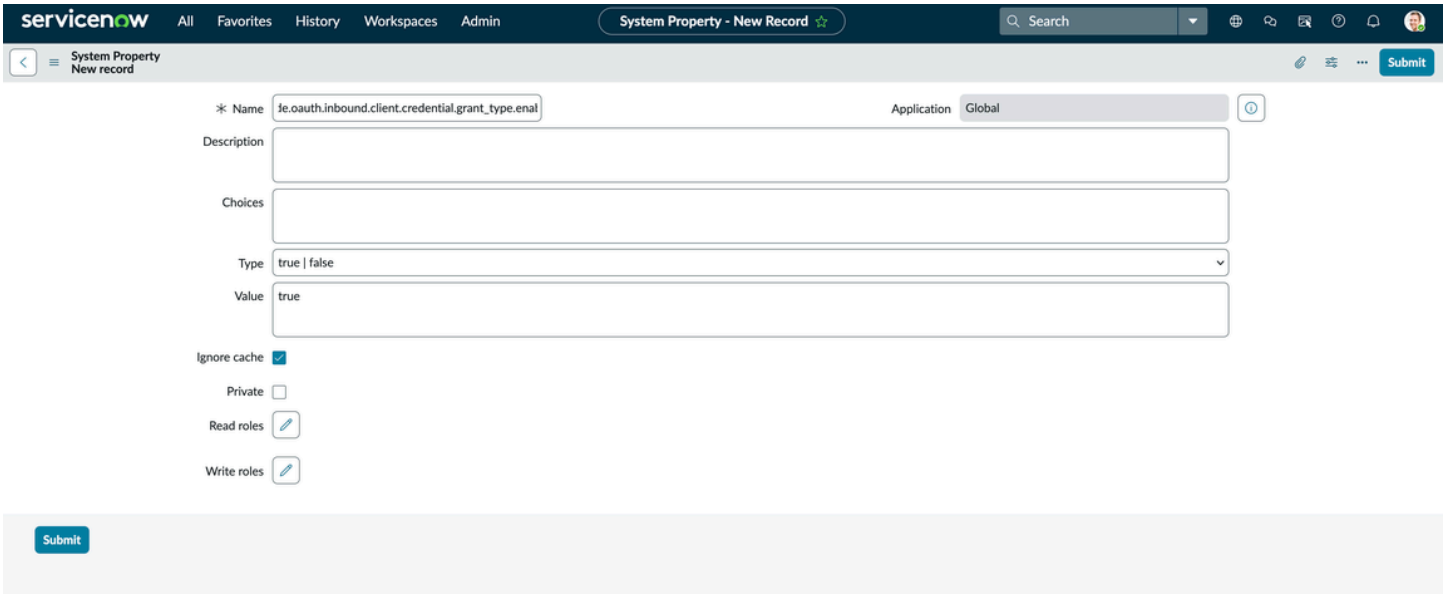
Erstellen Sie einen ServiceNow OAuth-Anwendungsclient

1. Aktivieren Sie die Systemeigenschaft für Client-Anmeldeinformationen Ihrer Instanz

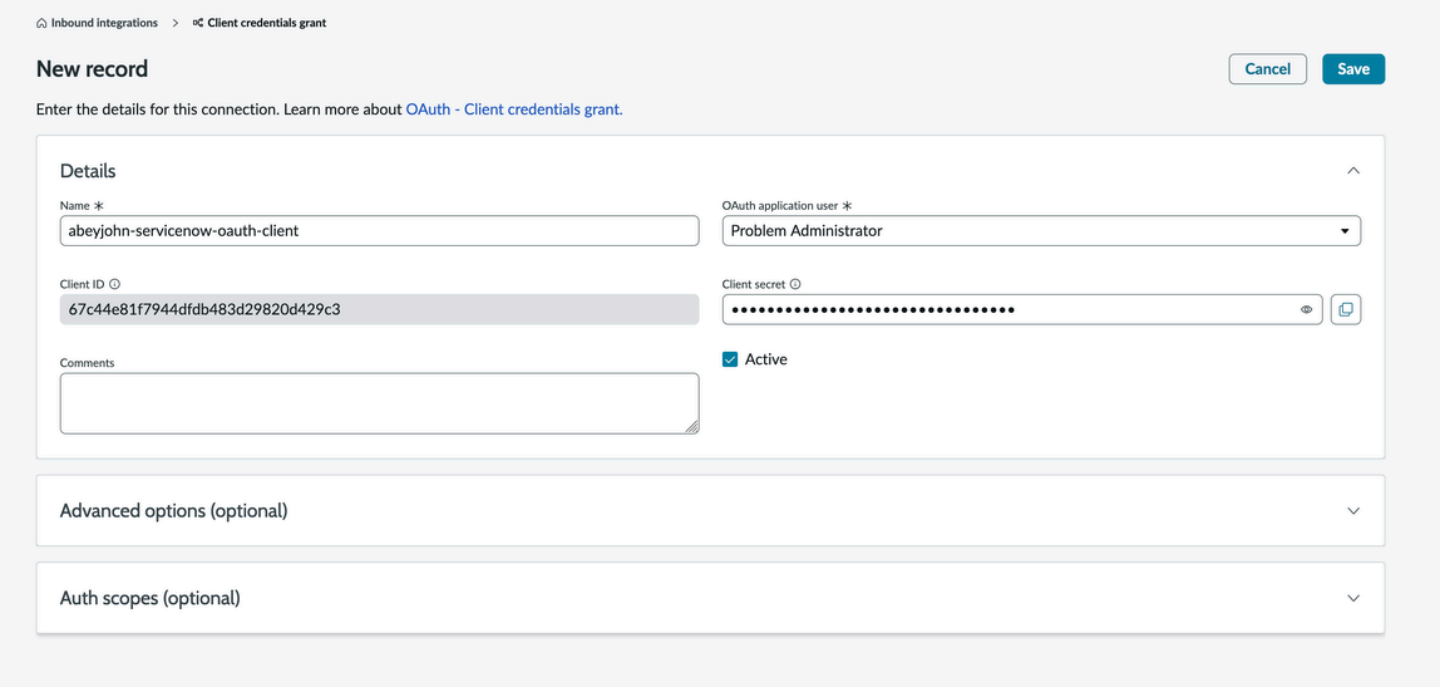
a. Suchen Sie `sys_properties.list` im Filter-Suchfeld und drücken Sie dann die Eingabetaste (die Option wird nicht angezeigt, aber das Drücken der Eingabetaste funktioniert)

b. Wählen Sie „Neu“

c. Fügen Sie den Namen als `glide.oauth.inbound.client.credential.grant_type.enabled` und den Wert zu true hinzu und geben Sie den Typ true | false ein



1. Navigieren Sie im Filtersuchfeld zu System OAuth > Anwendungsregistrierung
2. Wählen Sie „Neu“ > „Neues Integrationserlebnis für eingehenden Datenverkehr“ > „Neue Integration“ > „OAuth — Client Credentials Grant“
3. Wählen Sie einen Namen und setzen Sie den Benutzer der OAuth-Anwendung auf „Problem Administrator“ und klicken Sie auf „Speichern“

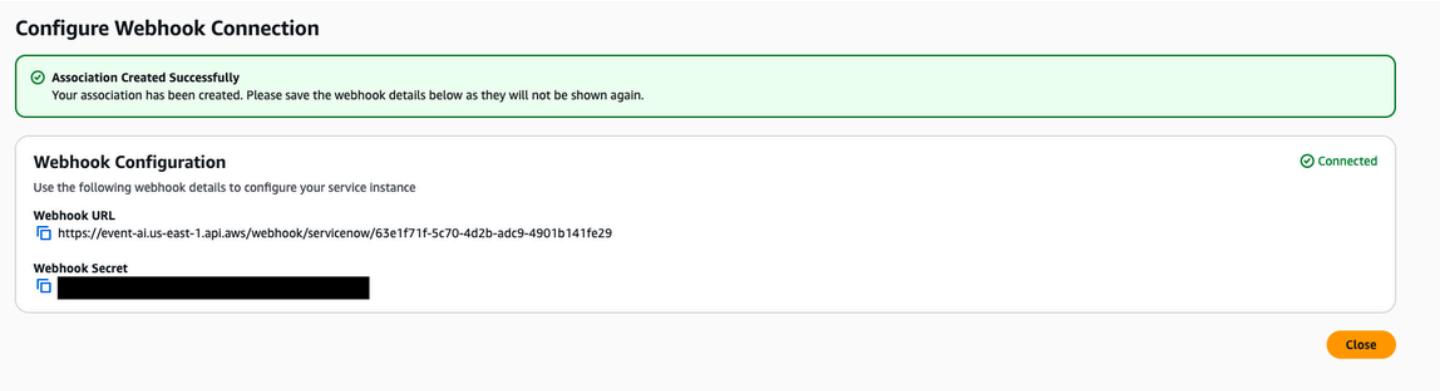


Connect Sie Ihren ServiceNow OAuth-Client mit AWS DevOps Agent

1. Sie können diesen Vorgang an zwei Stellen starten. Rufen Sie zunächst die Seite Capability Providers auf, suchen Sie ServiceNowunter Kommunikation und klicken Sie dann auf Registrieren. Alternativ können Sie einen beliebigen DevOps Agent-Bereich auswählen, den Sie möglicherweise erstellt haben, und zu Funktionen → Kommunikation → Hinzufügen → navigieren ServiceNow und auf Registrieren klicken.
2. Autorisieren Sie als Nächstes den DevOps Agenten, mithilfe des OAuth-Anwendungsclients, den Sie gerade erstellt haben, auf Ihre ServiceNow Instanz zuzugreifen.

- Folgen Sie den nächsten Schritten und speichern Sie die resultierenden Informationen über den Webhook

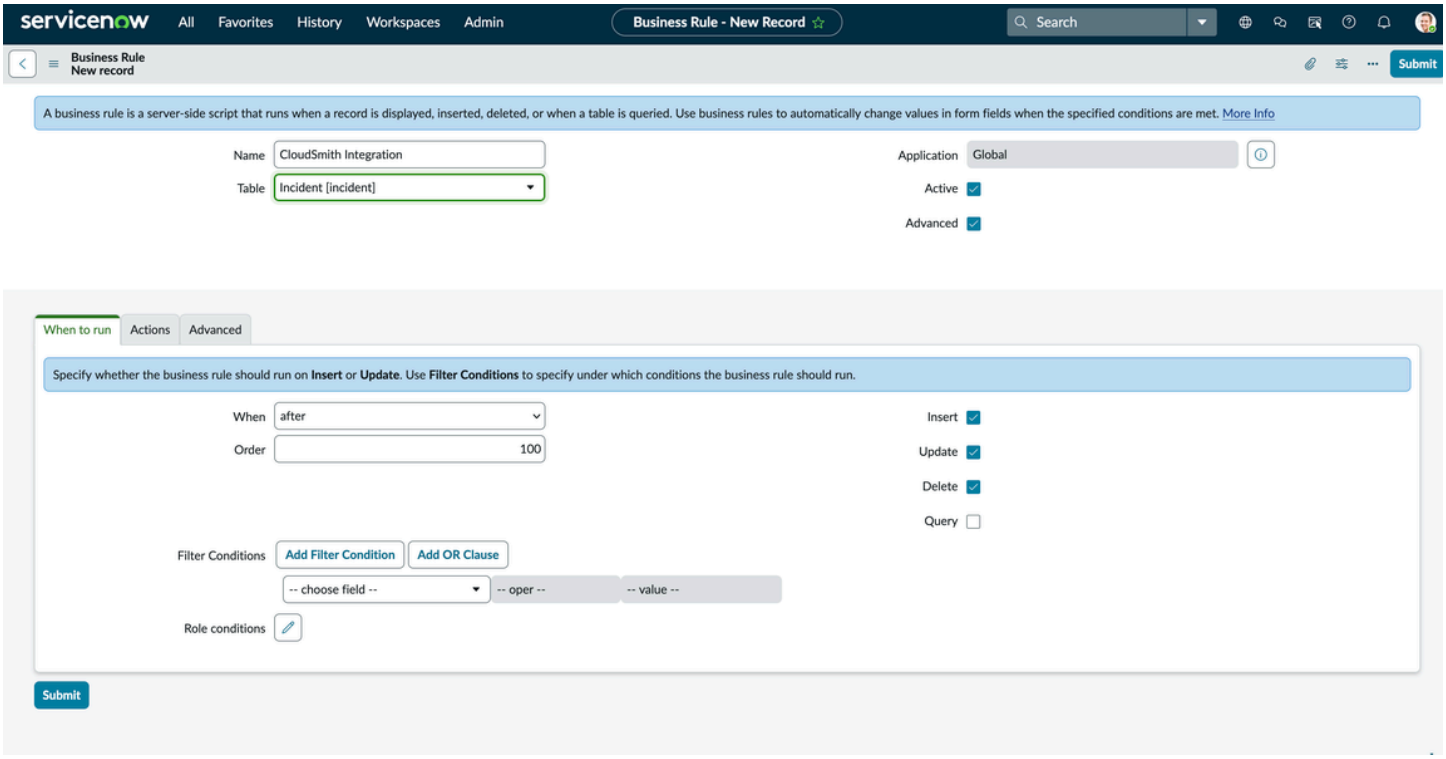
> **Important**
>
> Sie werden diese Informationen nicht wieder sehen



Konfigurieren Sie Ihre Geschäftsregel ServiceNow

Sobald Sie die Konnektivität hergestellt haben, müssen Sie eine Geschäftsregel konfigurieren ServiceNow , um Tickets an Ihre DevOps Agentenbereiche zu senden.

1. Navigieren Sie zu Activity Subscriptions → Administration → Business Rules und klicken Sie auf Neu.
2. Stellen Sie das Feld „Tabelle“ auf „Vorfall [Vorfall]“ ein, aktivieren Sie das Kästchen „Erweitert“ und legen Sie fest, dass die Regel nach Einfügen, Aktualisieren und Löschen ausgeführt wird.



1. Navigieren Sie zur Registerkarte „Erweitert“ und fügen Sie das folgende Webhook-Skript hinzu, fügen Sie Ihr Webhook-Geheimnis und Ihre URL an der angegebenen Stelle ein und klicken Sie auf Senden.

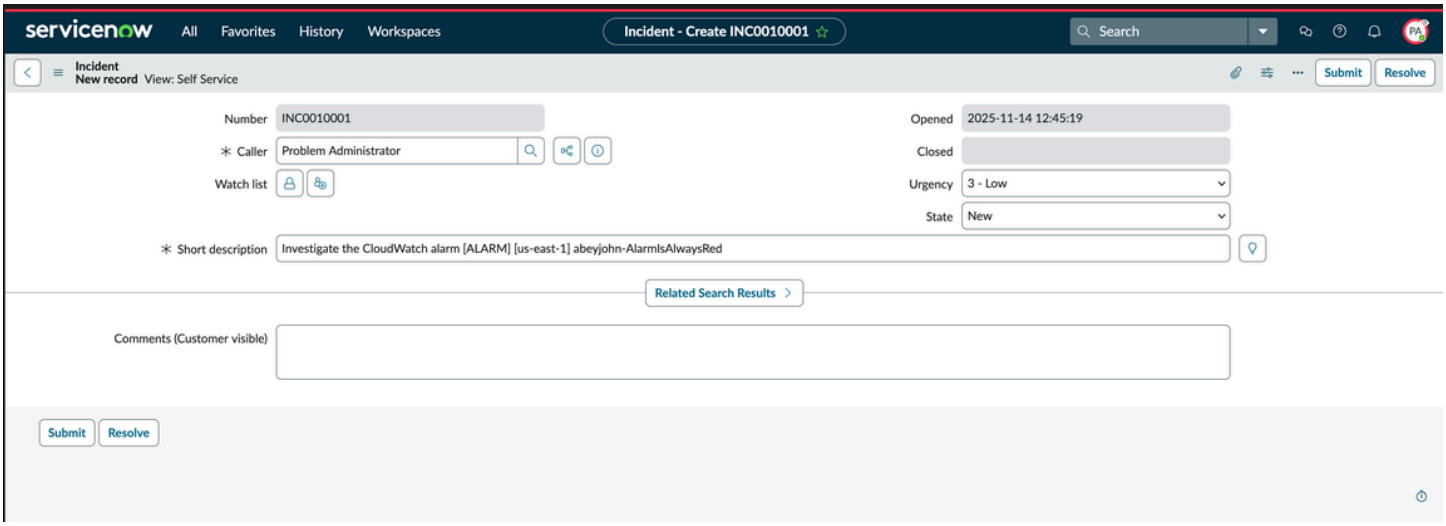
```
(function executeRule(current, previous /*null when async*/ ) {

    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);
```

Wenn Sie sich dafür entschieden haben, Ihre ServiceNow Verbindung auf der Seite Capability Providers zu registrieren, müssen Sie jetzt zu dem DevOps Agentenbereich navigieren, in dem Sie ServiceNow Incident-Tickets untersuchen möchten, Capabilities → Communications auswählen und dann die ServiceNow Instance registrieren, die Sie auf der Capability Provider-Seite registriert haben. Jetzt sollte alles eingerichtet sein, und alle Vorfälle, bei denen der Anrufer auf „Problem Administrator“ eingestellt ist (um die Berechtigungen nachzuahmen, die Sie dem AWS DevOps OAuth-Client erteilt haben), lösen im konfigurierten Agent Space eine Incident-Response-Untersuchung aus. DevOps Sie können dies testen, indem Sie einen neuen Incident erstellen ServiceNow und das Anrufer-Feld des Incidents auf „Problem Administrator“ setzen.



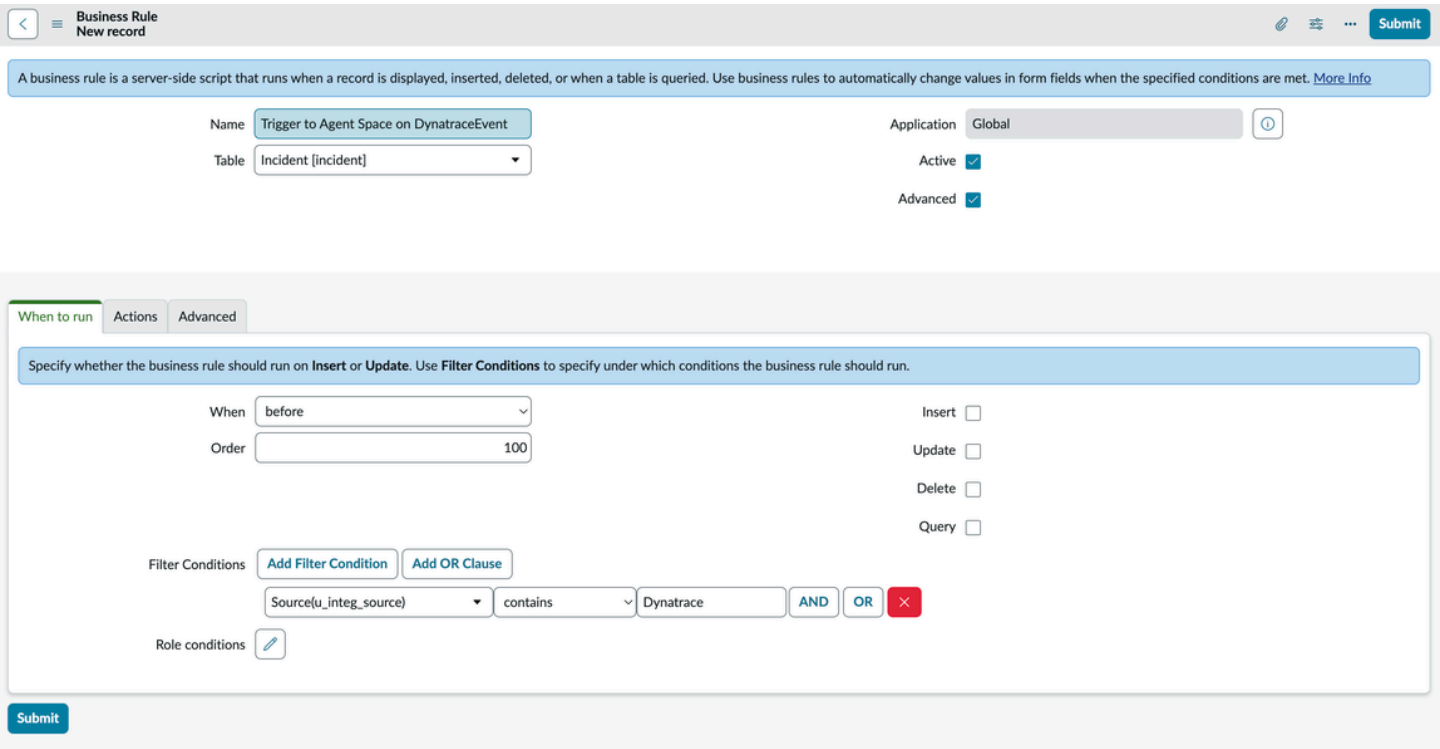
ServiceNow Ticketaktualisierungen

Während aller Untersuchungen zur Reaktion auf Vorfälle aktualisiert DevOps Ihr Mitarbeiter die wichtigsten Ergebnisse, Ursachenanalysen und Pläne zur Schadensbegrenzung in das ursprüngliche Ticket. Die Ergebnisse des Sachbearbeiters werden in den Kommentaren zu einem Vorfall veröffentlicht. Derzeit veröffentlichen wir nur Agentendatensätze vom Typ`finding`,, `cause` `investigation_summary``mitigation_summary`, und Aktualisierungen zum Ermittlungsstatus (z. B.`AWS DevOps Agent started/finished its investigation`).

Beispiele für die Weiterleitung und Orchestrierung von Tickets

Szenario: Filtern, welche Incidents an einen DevOps Agent Space gesendet werden

Dies ist ein einfaches Szenario, für das jedoch einige Konfigurationen erforderlich sind ServiceNow , um ein Feld zur Nachverfolgung der Vorfallquelle ServiceNow zu erstellen. Für dieses Beispiel erstellen Sie mit dem SNOW Form Builder ein neues Quellfeld (u_source). Auf diese Weise können Sie die Quelle des Vorfalls nachverfolgen und anhand dieser Informationen Anfragen von einer bestimmten Quelle an einen DevOps Agent Space weiterleiten. Die Weiterleitung erfolgt, indem eine Service Now-Geschäftsregel erstellt wird und auf der Registerkarte „Wann ausgeführt“ die Einstellungen „Wann“ und „Filterbedingungen“ ausgelöst werden. In diesem Beispiel sind die Filterbedingungen wie folgt festgelegt:

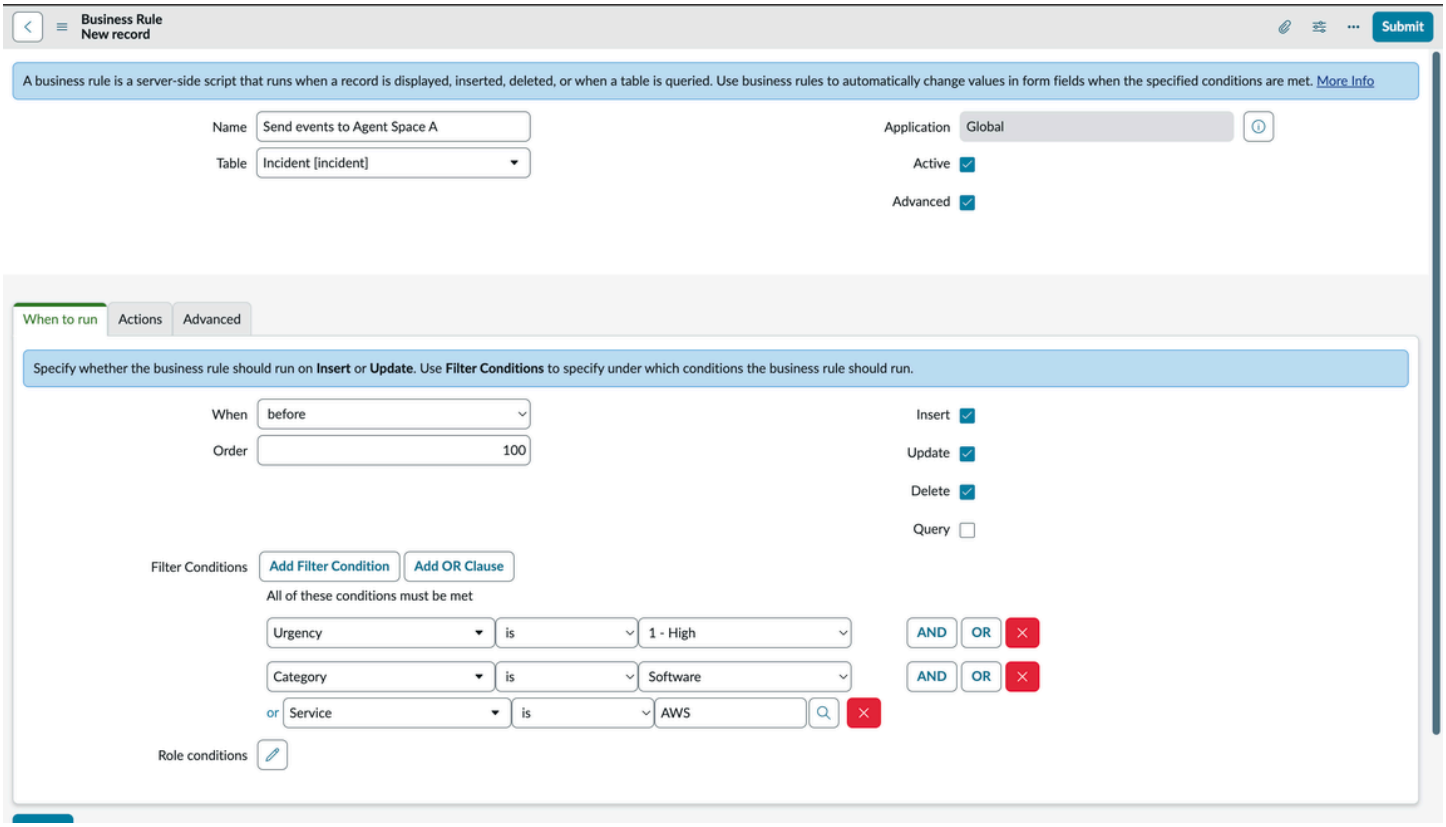


Szenario: Weiterleitung von Vorfällen über mehrere Agent Spaces DevOps

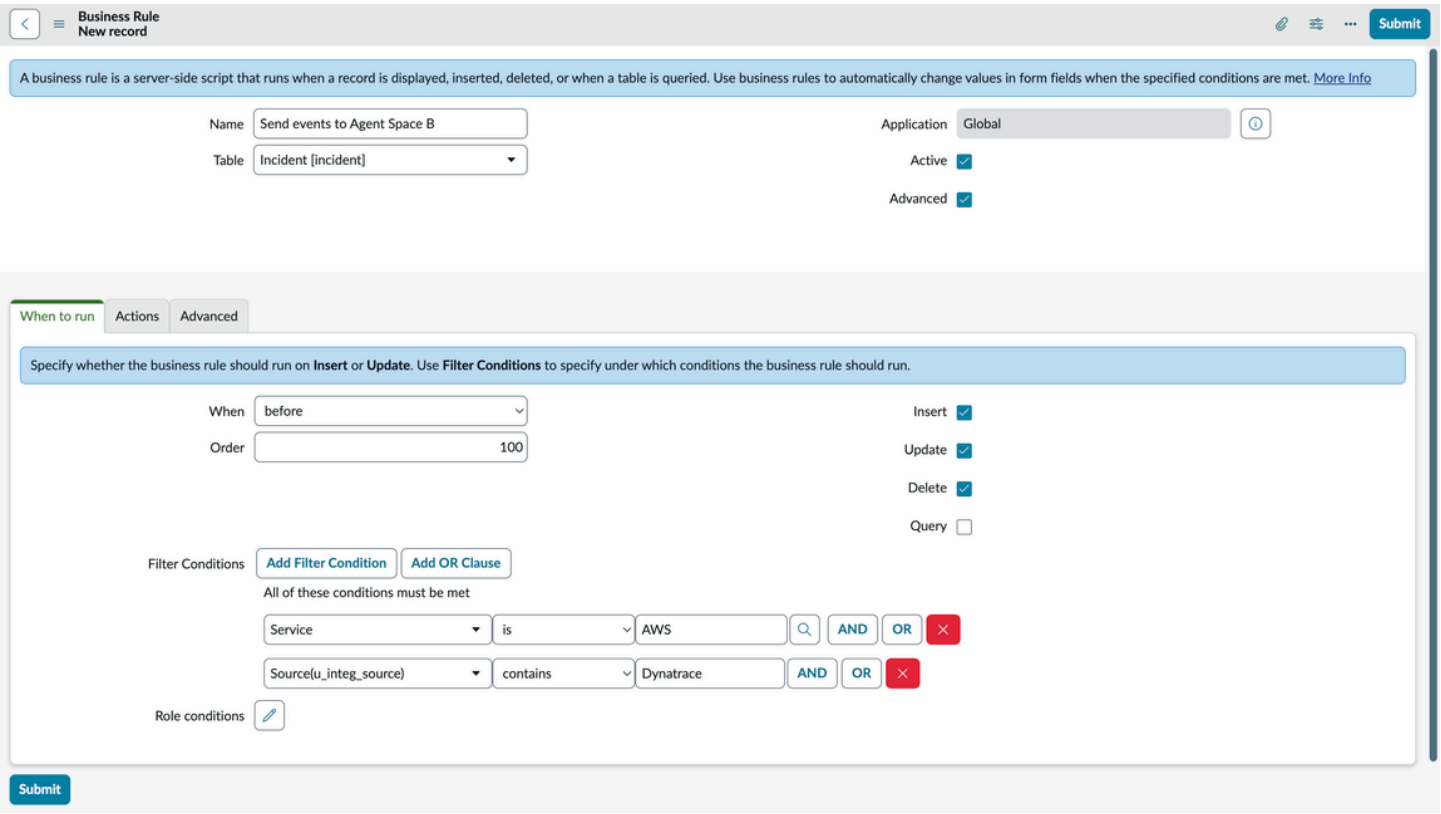
Dieses Beispiel zeigt, wie eine Untersuchung in DevOps Agent Space B ausgelöst wird, wenn die Dringlichkeit1, Kategorie oder Service ist`AWS`, und eine Untersuchung in DevOps Agent Space A ausgelöst wird`AWS`, wenn der Service und die Quelle ist`Dynatrace`. `Software`

Dieses Szenario kann auf zwei Arten erreicht werden. Das Webhook-Skript selbst kann aktualisiert werden, um diese Geschäftslogik einzubeziehen. In diesem Szenario werden wir zeigen, wie dies mit einer ServiceNow Geschäftsregel erreicht werden kann, um Transparenz zu gewährleisten und das Debuggen zu vereinfachen. Das Routing erfolgt durch die Erstellung von zwei Service Now-Geschäftsregeln.

- Erstellen Sie eine Geschäftsregel ServiceNow für DevOps Agent Space A und erstellen Sie mithilfe des Condition Builders eine Bedingung, sodass nur Ereignisse gesendet werden, die auf unserer angegebenen Bedingung basieren.



- Erstellen Sie als Nächstes eine weitere Geschäftsregel in ServiceNow für AgentSpace B, für die die Geschäftsregel nur ausgelöst wird, wenn Service aktiviert ist AWS und die Quelle Dynatrace ist.



Wenn Sie nun einen neuen Incident erstellen, der der angegebenen Bedingung entspricht, löst er entweder eine Untersuchung in DevOps Agent Space A oder DevOps Agent Space B aus, sodass Sie eine genaue Kontrolle über die Weiterleitung von Vorfällen haben.

Slack verbinden

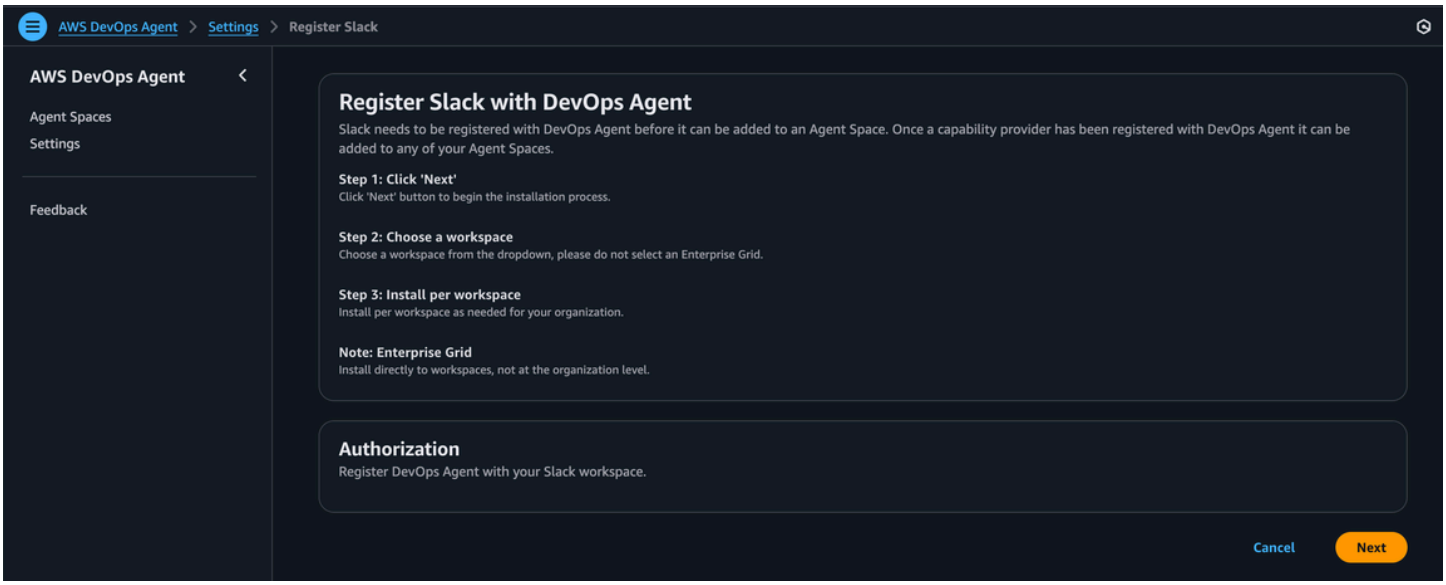
Du kannst den AWS DevOps Agenten so konfigurieren, dass er einen von dir ausgewählten Slack-Channel mit den wichtigsten Ergebnissen der Untersuchung von Vorfällen, Ursachenanalysen und generierten Plänen zur Schadensbegrenzung aktualisiert.

Bevor Sie beginnen

Slack muss bei DevOps Agent registriert sein, bevor es einem Agent Space hinzugefügt werden kann. Um AWS DevOps Agent mit Slack zu integrieren, musst du die folgenden Anforderungen erfüllen:

- Du hast Zugriff auf einen Slack-Workspace mit der Möglichkeit, Anwendungen von Drittanbietern zu installieren und zu autorisieren
- Hast du die Slack-Channels identifiziert, über die der AWS DevOps Agent Benachrichtigungen senden soll

Registrieren Sie die Slack-Integration mit Agent AWS DevOps



1. Suche auf der Seite Capability Providers in der AWS DevOps Agent-Konsole im Abschnitt Verfügbare Anbieter unter Kommunikation nach Slack und klicke auf Registrieren.
2. Wähle die Schaltfläche „Registrieren“.
3. Du wirst zu Slack weitergeleitet, um die AWS DevOps Agenten-Anwendung für deinen Workspace zu autorisieren.
4. Installiere auf der Autorisierungsseite von Slack direkt in Workspaces, nicht auf Organisationsebene.
5. Wähle einen Workspace aus dem Drop-down-Menü aus. Wählen Sie kein Enterprise Grid aus.
6. Installieren Sie je nach Bedarf pro Workspace, wie es für Ihre Organisation erforderlich ist.
7. Überprüfen Sie die angeforderten Bereiche und klicken Sie auf Zulassen, um die Integration zu autorisieren.
8. Nach der Autorisierung kehren Sie zur AWS DevOps Agentenkonsole zurück.

Verbinde Slack mit deinen DevOps Agent Space (s)

Nachdem du Slack in deinem DevOps Agent Space registriert hast, kannst du es mit deinen DevOps Agent Space (s) verknüpfen:

1. Navigiere auf dem Tab Funktionen in deiner Konfiguration AgentSpace zu Communications > Slack.
2. Wähle Slack hinzufügen

3. Gib die Kanal-ID ein
4. Wähle Create, um die Slack-Konfiguration abzuschließen.

> **Note**
>
> Der Bot-Benutzer des Agenten muss zu privaten Channels hinzugefügt werden, bevor er Nachrichten posten kann.

> **Important**
>
> Die Deinstallation der Slack-App kann dazu führen, dass die Slack-App nicht erneut installiert werden kann. Bitte vermeide es, die Slack-App zu deinstallieren.

DevOps Agent über Webhook aufrufen

Webhooks ermöglichen es externen Systemen, automatisch Agentenuntersuchungen auszulösen AWS DevOps . Dies ermöglicht die Integration mit Ticketsystemen, Überwachungstools und anderen Plattformen, die HTTP-Anfragen senden können, wenn Vorfälle auftreten.

Voraussetzungen

Bevor Sie den Webhook-Zugriff konfigurieren, stellen Sie sicher, dass Sie über Folgendes verfügen:

- Ein im Agent konfigurierter Agent-Space AWS DevOps
- Zugriff auf die AWS DevOps Agent-Konsole
- Das externe System, das Webhook-Anfragen sendet

Webhook-Typen

AWS DevOps Agent unterstützt die folgenden Arten von Webhooks:

- Integration-specific Webhooks — Wird automatisch generiert, wenn Sie Integrationen von Drittanbietern wie Dynatrace, Splunk, Datadog, New Relic oder Slack konfigurieren. ServiceNow Diese Webhooks sind der jeweiligen Integration zugeordnet und verwenden Authentifizierungsmethoden, die vom Integrationstyp bestimmt werden

- Generische Webhooks — Können manuell erstellt werden, um Untersuchungen aus beliebigen Quellen auszulösen, die nicht durch eine bestimmte Integration abgedeckt sind. Generische Webhooks verwenden derzeit die HMAC-Authentifizierung (Bearer-Token derzeit nicht verfügbar).
- Grafana-Alert-Webhooks — Grafana kann Warnmeldungen über Webhook-Kontaktpunkte direkt an den AWS DevOps Agenten senden. Anweisungen zur Einrichtung, einschließlich einer benutzerdefinierten Benachrichtigungsvorlage, finden Sie unter Grafana verbinden.

Webhook-Authentifizierungsmethoden

Die Authentifizierungsmethode für Ihren Webhook hängt davon ab, mit welcher Integration er verknüpft ist:

HMAC-Authentifizierung — Wird verwendet von:

- Webhooks zur Dynatrace-Integration
- Generische Webhooks (nicht mit einer bestimmten Drittanbieter-Integration verknüpft)

Bearer-Token-Authentifizierung — Wird verwendet von:

- Webhooks zur Splunk-Integration
- Webhooks zur Datadog-Integration
- Webhooks zur Integration von New Relic
- ServiceNow Webhooks für die Integration
- Webhooks zur Integration von Slack
- Webhooks zur Grafana-Integration

Die HMAC-Authentifizierung verstehen

HMAC (Hash-based Message Authentication Code) ist ein kryptografischer Mechanismus, der sowohl die Integrität als auch die Authentizität einer Webhook-Anfrage überprüft. Wenn Sie einen Webhook mit HMAC-Authentifizierung senden, generieren Sie eine Signatur, indem Sie den Zeitstempel und die Nutzdaten der Anfrage zusammen mit Ihrem geheimen Schlüssel und dem Algorithmus hashen. SHA-256 AWS DevOps Der Agent berechnet auf seiner Seite unabhängig denselben Hash und vergleicht die beiden Signaturen. Wenn sie übereinstimmen, wird die Anfrage akzeptiert.

Da der Zeitstempel in der Signatur enthalten ist, bietet HMAC auch Schutz vor Wiederholungen. Der AWS DevOps Agent kann Anfragen zurückweisen, deren Zeitstempel zu weit in der Vergangenheit liegen, wodurch verhindert wird, dass ein Angreifer eine gültige Anfrage abfängt und erneut sendet.

Wahl zwischen HMAC und Bearer-Token

Überlegungen	HMAC	Inhaber-Token
Komplexität der Einrichtung	Komplexer — Ihr Kunde muss für jede Anfrage anhand des Zeitstempels und der Nutzdaten eine Signatur berechnen	Einfacher — füge ein statische s Token in den Header ein `Authorization`
Integrität der Nutzlast	Verifiziert — Jede Änderung der Payload nach dem Signieren macht die Signatur ungültig	Nicht verifiziert — das Token authentifiziert den Absender, schützt aber nicht den Inhalt der Nutzlast
Schutz bei wiederholter Wiedergabe	Built-in — Der Zeitstempel in der Signatur ermöglich t es dem Server, veraltete Anfragen abzulehnen	Nicht integriert — ein erfasstes Token kann wiederverwendet werden, bis es rotiert wird
Geheimes Expositionsrisiko	Niedriger — das Geheimnis wird in der Anfrage nie übertragen, sondern nur die berechnete Signatur wird gesendet	Höher — Das Token wird in jedem Anforderungsheader gesendet, was die Gefahr erhöht, wenn der Datenverk ehr abgefangen wird
Wann sollte dies verwendet werden?	Wird empfohlen, wenn Sie stärkere Sicherheitsgarantien benötigen, z. B. für generisch e Webhooks oder Umgebunge n mit strengen Compliance-Anforderungen	Geeignet, wenn einfache Integration Priorität hat und Ihr Netzwerktransport vertrauen swürdig ist, z. B. für verwaltet e SaaS-Integrationen über HTTPS

Hinweis: Die Authentifizierungsmethode wird durch den Integrationstyp bestimmt. Integration-specific Webhooks (Splunk, Datadog, New Relic, Slack, Grafana) verwenden die ServiceNow Bearer-Token-Authentifizierung. Dynatrace und generische Webhooks verwenden die HMAC-Authentifizierung. Sie können die Authentifizierungsmethode für einen integrationsspezifischen Webhook nicht ändern.

Webhook-Zugriff konfigurieren

Schritt 1: Navigieren Sie zur Webhook-Konfiguration

1. Melden Sie sich bei der AWS Management Console an und navigieren Sie zur AWS DevOps Agent-Konsole
2. Wählen Sie Ihren Agent Space aus
3. Gehen Sie zur Registerkarte Funktionen
4. Klicken Sie im Abschnitt Webhook auf Configure

Schritt 2: Generieren Sie Webhook-Anmeldeinformationen

Für integrationsspezifische Webhooks:

Webhooks werden automatisch generiert, wenn Sie die Konfiguration einer Drittanbieter-Integration abschließen. Die Webhook-Endpunkt-URL und die Anmeldeinformationen werden am Ende des Integrations-Setup-Prozesses bereitgestellt.

Für generische Webhooks:

1. Klicken Sie auf Webhook generieren
2. Das System generiert ein HMAC-Schlüsselpaar
3. Speichern Sie den generierten Schlüssel und das Geheimnis sicher — Sie können sie nicht erneut abrufen
4. Kopieren Sie die angegebene Webhook-Endpunkt-URL

Schritt 3: Konfigurieren Sie Ihr externes System

Verwenden Sie die Webhook-Endpunkt-URL und die Anmeldeinformationen, um Ihr externes System so zu konfigurieren, dass es Anfragen an den AWS DevOps Agenten sendet. Die spezifischen Konfigurationsschritte hängen von Ihrem externen System ab.

Verwaltung von Webhook-Anmeldeinformationen

Anmeldeinformationen entfernen — Um Webhook-Anmeldeinformationen zu löschen, gehen Sie zum Abschnitt mit der Webhook-Konfiguration und klicken Sie auf Entfernen. Nach dem Entfernen der Anmeldeinformationen akzeptiert der Webhook-Endpunkt keine Anfragen mehr, bis Sie neue Anmeldeinformationen generieren.

Anmeldeinformationen neu generieren — Um neue Anmeldeinformationen zu generieren, entfernen Sie zuerst die vorhandenen Anmeldeinformationen und generieren Sie dann ein neues key pair oder Token.

Den Webhook verwenden

Webhook-Anforderungsformat

Um eine Untersuchung auszulösen, sollte Ihr externes System eine HTTP-POST-Anfrage an die Webhook-Endpunkt-URL senden.

Für Version 1 (HMAC-Authentifizierung):

Kopfzeilen:

- `Content-Type: application/json`
- `x-amzn-event-signature: <HMAC signature>`
- `x-amzn-event-timestamp: <+%Y-%m-%dT%H:%M:%S.000Z>`

Die HMAC-Signatur wird generiert, indem Sie den Anforderungstext mit Ihrem geheimen Schlüssel signieren. SHA-256

Für Version 2 (Bearer-Token-Authentifizierung):

Kopfzeilen:

- `Content-Type: application/json`
- `Authorization: Bearer <your-token>`

Hauptteil der Anfrage:

Der Hauptteil der Anfrage sollte Informationen über den Vorfall enthalten:

```
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"
    }
  }
}
```

Payload-Schema:

```
{
    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;
}
```

Beispiel-Code

Version 1 (HMAC-Authentifizierung) -: 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);
```

```
});
```

Version 1 (HMAC-Authentifizierung) — 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"
```

Version 2 (Bearer-Token-Authentifizierung) -: 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),
  });
}
```

Version 2 (Bearer-Token-Authentifizierung) — 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"
```

Fehlerbehebung bei Webhooks

Wenn Sie keine 200 erhalten

Eine 200 und eine Meldung wie Webhook erhalten bedeuten, dass die Authentifizierung bestanden wurde und die Nachricht in die Warteschlange gestellt wurde, damit das System sie überprüfen und verarbeiten kann. Wenn Sie keine 200, sondern eine 4xx erhalten, stimmt höchstwahrscheinlich etwas mit der Authentifizierung oder den Headern nicht. Versuchen Sie, manuell zu senden, indem Sie die Curl-Optionen verwenden, um die Authentifizierung zu debuggen.

Wenn Sie eine 200 erhalten, aber eine Untersuchung nicht gestartet wird

Die wahrscheinliche Ursache ist eine falsch formatierte Nutzlast.

1. Vergewissern Sie sich, dass sowohl der Zeitstempel als auch die Vorfall-ID aktualisiert und eindeutig sind. Doppelte Nachrichten werden dedupliziert.
2. Prüfen Sie, ob die Nachricht gültig ist (JSON)
3. Prüfen Sie, ob das Format korrekt ist

Wenn Sie eine 200 erhalten und die Untersuchung sofort abgebrochen wird

Höchstwahrscheinlich haben Sie das Limit für den Monat erreicht. Bitte wenden Sie sich an Ihren AWS Ansprechpartner, um gegebenenfalls eine Änderung des Ratenlimits zu beantragen.

Verwandte Themen

- the section called “Einen Agentenbereich erstellen”
- the section called “Was ist eine DevOps Agent-Web-App?”
- the section called “DevOps IAM-Berechtigungen für Agenten”

Integration AWS DevOps Agent bei Amazon EventBridge

Sie können AWS DevOps Agent in Ihre ereignisgesteuerten Anwendungen integrieren, indem Sie Ereignisse verwenden, die während der Inspektions- und Schadensbegrenzungszyklen auftreten. AWS DevOps Der Agent sendet Ereignisse an Amazon, EventBridge wenn sich der Status einer Untersuchung oder Schadensbegrenzung ändert. Anschließend können Sie EventBridge Regeln erstellen, die auf der Grundlage dieser Ereignisse Maßnahmen ergreifen.

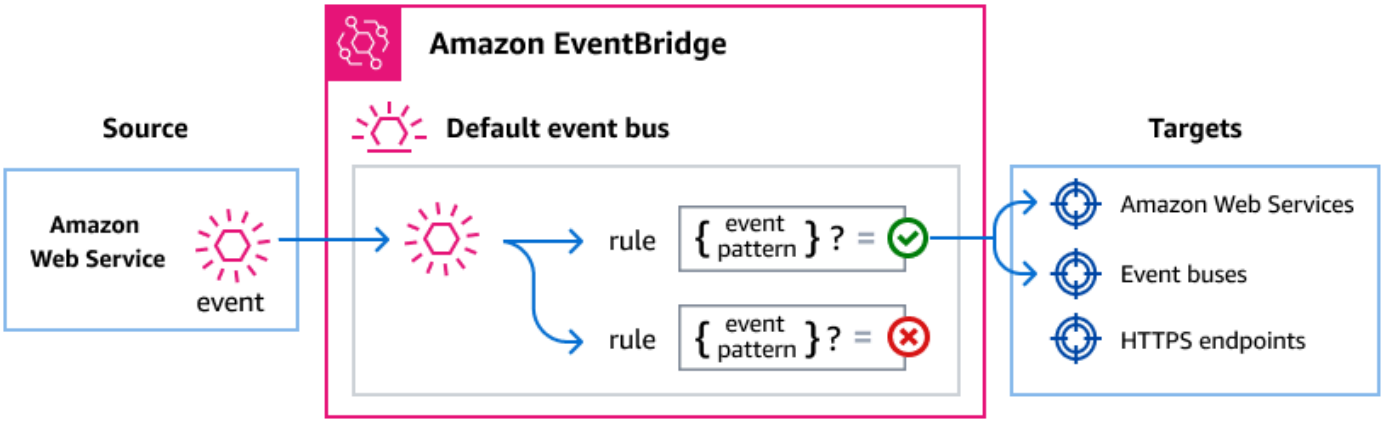
Sie können beispielsweise Regeln erstellen, die die folgenden Aktionen ausführen:

- Rufen Sie eine AWS Lambda-Funktion auf, um Untersuchungsergebnisse zu verarbeiten, wenn eine Untersuchung abgeschlossen ist.
- Senden Sie eine Amazon SNS SNS-Benachrichtigung, wenn eine Untersuchung fehlschlägt oder eine Zeitüberschreitung eintritt.
- Aktualisieren Sie ein Ticketsystem, wenn eine neue Untersuchung erstellt wird.
- Starten Sie einen AWS Step Functions Functions-Workflow, wenn eine Abhilfemaßnahme abgeschlossen ist.

Wie Routen EventBridge AWS DevOps Ereignisse für Agenten

AWS DevOps Der Agent sendet Ereignisse an den EventBridge Standard-Event-Bus. EventBridge bewertet die Ereignisse dann anhand der von Ihnen erstellten Regeln. Wenn ein Ereignis mit dem Ereignismuster einer Regel übereinstimmt, wird das Ereignis an die angegebenen Ziele EventBridge gesendet.

Das folgende Diagramm zeigt, wie EventBridge AWS DevOps Agent-Ereignisse weitergeleitet werden.



1. AWS DevOps Der Agent sendet ein Ereignis an den EventBridge Standard-Event-Bus, wenn sich der Lebenszyklusstatus einer Untersuchung oder Schadensbegrenzung ändert.
2. EventBridge bewertet das Ereignis anhand der Regeln, die Sie erstellt haben.
3. Wenn das Ereignis mit dem Ereignismuster einer Regel übereinstimmt, wird das Ereignis an die in der Regel angegebenen Ziele EventBridge gesendet.

AWS DevOps Agentenereignisse

AWS DevOps Der Agent sendet die folgenden Ereignisse an EventBridge. Alle Ereignisse verwenden die Quelle`aws.aidevops`.

Unterstützte Ermittlungsereignisse

detail-type	Description
`Investigation Created`	Im Agentenbereich wurde eine Untersuchung eingeleitet.
`Investigation Priority Updated`	Die Priorität einer Untersuchung wurde geändert.
`Investigation In Progress`	Mit einer Untersuchung wurde eine aktive Analyse eingeleitet.

detail-type	Description
`Investigation Completed`	Eine Untersuchung wurde mit Ergebnissen erfolgreich abgeschlossen.
`Investigation Failed`	Bei einer Untersuchung ist ein Fehler aufgetreten und sie konnte nicht abgeschlossen werden.
`Investigation Timed Out`	Bei einer Untersuchung wurde die zulässige Höchstdauer überschritten.
`Investigation Cancelled`	Eine Untersuchung wurde vor Abschluss abgebrochen.
`Investigation Pending Triage`	Eine Untersuchung wartet noch auf die Triage, bevor die aktive Analyse beginnt.
`Investigation Linked`	Eine Untersuchung wurde mit einem ähnlichen Vorfall oder Ticket verknüpft.
`Investigation Skipped`	Eine Untersuchung wurde übersprungen, weil sie den in einem Skill definierten Übersprungkriterien entsprach.

Unterstützte Minderungsereignisse

detail-type	Description
`Mitigation In Progress`	Eine Minderungsmaßnahme wurde gestartet.
`Mitigation Completed`	Eine Minderungsmaßnahme wurde erfolgreich abgeschlossen.
`Mitigation Failed`	Bei einer Minderungsaktion ist ein Fehler aufgetreten und sie konnte nicht abgeschlossen werden.

detail-type	Description
`Mitigation Timed Out`	Eine Minderungsmaßnahme hat die maximal zulässige Dauer überschritten.
`Mitigation Cancelled`	Eine Minderungsmaßnahme wurde vor Abschluss abgebrochen.

Ausführliche Feldbeschreibungen und Beispielereignisse finden Sie unterthe section called “AWS DevOps Detailreferenz zu Agentenereignissen”.

Passende Ereignismuster erstellen AWS DevOps Agentenereignisse

EventBridge Regeln verwenden Ereignismuster, um Ereignisse auszuwählen und sie an Ziele weiterzuleiten. Ein Ereignismuster entspricht der Struktur der Ereignisse, die es verarbeitet. Sie erstellen Ereignismuster, um AWS DevOps Agentenereignisse anhand der Ereignisfelder zu filtern.

Die folgenden Beispiele zeigen Ereignismuster für gängige Anwendungsfälle.

Ordnen Sie allen AWS DevOps Agentenereignissen zu

Das folgende Ereignismuster entspricht allen Ereignissen von AWS DevOps Agent.

```
{
  "source": ["aws.aidevops"]
}
```

Nur Ermittlungsereignisse zuordnen

Das folgende Ereignismuster verwendet eine Präfixübereinstimmung, um nur Ereignisse im Untersuchungszyklus auszuwählen.

```
{
  "source": ["aws.aidevops"],
  "detail-type": [{"prefix": "Investigation"}]
}
```

Ordnet nur Abschlusses- und Fehlschlagsereignisse zu

Das folgende Ereignismuster ordnet Ereignisse für abgeschlossene oder fehlgeschlagene Untersuchungen und Abhilfemaßnahmen zu.

```
{
  "source": ["aws.aidevops"],
  "detail-type": [
    "Investigation Completed",
    "Investigation Failed",
    "Mitigation Completed",
    "Mitigation Failed"
  ]
}
```

Ordnet Ereignisse für einen bestimmten Agentenbereich zu

Das folgende Ereignismuster entspricht Ereignissen aus einem bestimmten Agentenbereich.

```
{
  "source": ["aws.aidevops"],
  "detail": {
    "metadata": {
      "agent_space_id": ["your-agent-space-id"]
    }
  }
}
```

Weitere Informationen zu Ereignismustern finden Sie unter EventBridge Amazon-Ereignismuster im EventBridge Amazon-Benutzerhandbuch.

EventBridge Amazon-Berechtigungen

AWS DevOps Der Agent benötigt keine zusätzlichen Berechtigungen, um Ereignisse an zu senden EventBridge. Die Ereignisse werden automatisch an den Standard-Event-Bus gesendet.

Abhängig von den Zielen, die Sie für Ihre EventBridge Regeln konfigurieren, müssen Sie möglicherweise bestimmte Berechtigungen hinzufügen. Weitere Informationen zu den für Ziele erforderlichen Berechtigungen finden Sie unter Using Resource Based Policies for Amazon EventBridge im EventBridge Amazon-Benutzerhandbuch.

Zusätzliche Ressourcen EventBridge

Weitere Informationen zu EventBridge Konzepten und Konfiguration finden Sie in den folgenden Themen im EventBridge Amazon-Benutzerhandbuch:

- EventBridge Busse für Veranstaltungen
- EventBridge Ereignisse
- EventBridge Ereignismuster
- EventBridge Regeln
- EventBridge Ziele

AWS DevOps Detailreferenz zu Agentenereignissen

Ereignisse von AWS Diensten haben gemeinsame Metadatenfelder`source`, darunter`detail-type`,`account`,`region`, und`time`. Diese Ereignisse enthalten auch ein `detail` Feld mit dienstspezifischen Daten. Bei AWS DevOps Agent-Ereignissen `source` ist das immer `aws.aidevops` und das `detail-type` identifiziert das spezifische Ereignis.

Ermittlungsereignisse

Die folgenden `detail-type` Werte identifizieren Ermittlungsereignisse:

- `Investigation Created`
- `Investigation Priority Updated`
- `Investigation In Progress`
- `Investigation Completed`
- `Investigation Failed`
- `Investigation Timed Out`
- `Investigation Cancelled`
- `Investigation Pending Triage`
- `Investigation Linked`
- `Investigation Skipped`

Die `detail-type` Felder `source` und sind unten aufgeführt, da sie spezifische Werte für AWS DevOps Agentenereignisse enthalten. Definitionen der anderen Metadatenfelder, die in allen

Ereignissen enthalten sind, finden Sie unter Event-Struktur in der Amazon EventBridge Events-Referenz.

Im Folgenden finden Sie die JSON-Struktur für Ermittlungsereignisse.

```
{
  . . .,
  "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`Identifiziert den Ereignistyp. Bei Ermittlungsereignissen ist dies einer der oben aufgeführten Ereignisnamen.

`source`Identifiziert den Dienst, der das Ereignis generiert hat. Für AWS DevOps Agent-Ereignisse ist dieser Wert`aws.aidevops`.

`detail`Ein JSON-Objekt, das ereignisspezifische Daten enthält. Das `detail` Objekt umfasst die folgenden Felder:

- `version`(string) — Die Schemaversion des Ereignisdetails. Derzeit`1.0.0`.
- `metadata.agent_space_id`(string) — Die eindeutige Kennung des Agentenbereichs, in dem das Ereignis seinen Ursprung hat.
- `metadata.task_id`(string) — Die eindeutige Kennung der Aufgabe.

- `metadata.execution_id`(string) — Die eindeutige Kennung des Ausführungslaufs. Vorhanden, wenn der Untersuchung eine Hinrichtung zugewiesen wurde.
- `data.task_type`(string) — Der Typ der Aufgabe. Wert:`INVESTIGATION`.
- `data.priority`(string) — Die Prioritätsstufe. Werte:`CRITICAL`,`HIGH`,`MEDIUM`,`LOW`,`MINIMAL`.
- `data.status`(string) — Der aktuelle Status. Werte:`PENDING_START`,`IN_PROGRESS`,`COMPLETED`,`FAILED`,`TIMED_OUT`,`CANCELLED`,`PENDING_TRIAGE`
- `data.created_at`(string) — ISO 8601-Zeitstempel, als die Aufgabe erstellt wurde.
- `data.updated_at`(string) — ISO 8601-Zeitstempel, als die Aufgabe zuletzt aktualisiert wurde.
- `data.summary_record_id`(string) — Die Kennung des zusammenfassenden Datensatzes, der die Untersuchungsergebnisse enthält. Ist enthalten, wenn eine Zusammenfassung für die abgeschlossene Untersuchung generiert wird. Sie können den Inhalt der Zusammenfassung über die AWS DevOps Agenten-API abrufen, indem Sie diese Kennung verwenden, um nach dem Journaldatensatz mit dem Datensatztyp zu suchen`investigation_summary_md`.

Beispiel: Ereignis „Untersuchung abgeschlossen“

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

Beispiel: Ereignis „Untersuchung fehlgeschlagen“

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

Ereignisse zur Schadensbegrenzung

Die folgenden `detail-type` Werte identifizieren Minderungsereignisse:

- `Mitigation In Progress`

- `Mitigation Completed`
- `Mitigation Failed`
- `Mitigation Timed Out`
- `Mitigation Cancelled`

Die `detail-type` Felder `source` und sind unten aufgeführt, da sie spezifische Werte für AWS DevOps Agentenereignisse enthalten. Definitionen der anderen Metadatenfelder, die in allen Ereignissen enthalten sind, finden Sie unter Event-Struktur in der Amazon EventBridge Events-Referenz.

Im Folgenden finden Sie die JSON-Struktur für Minderungsereignisse.

```
{
  . . .,
  "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`Identifiziert den Ereignistyp. Für Minderungsereignisse ist dies einer der zuvor aufgeführten Ereignisnamen.

`source`Identifiziert den Dienst, der das Ereignis generiert hat. Für AWS DevOps Agent-Ereignisse ist dieser Wert`aws.aidevops`.

`detail`Ein JSON-Objekt, das ereignisspezifische Daten enthält. Das `detail` Objekt umfasst die folgenden Felder:

- `version`(string) — Die Schemaversion des Ereignisdetails. Derzeit`1.0.0`.
- `metadata.agent_space_id`(string) — Die eindeutige Kennung des Agentenbereichs, in dem das Ereignis seinen Ursprung hat.
- `metadata.task_id`(string) — Die eindeutige Kennung der Aufgabe.
- `metadata.execution_id`(string) — Die eindeutige Kennung des Ausführungslaufs. Vorhanden, wenn der Schadensbegrenzung eine Ausführung zugewiesen wurde.
- `data.task_type`(string) — Der Typ der Aufgabe. Wert:`INVESTIGATION`.
- `data.priority`(string) — Die Prioritätsstufe. Werte:`CRITICAL`,`HIGH`,`MEDIUM`,`LOW`,`MINIMAL`.
- `data.status`(string) — Der aktuelle Status. Werte:`IN_PROGRESS`,`COMPLETED`,`FAILED`,`TIMED_OUT`,`CANCELLED`.
- `data.created_at`(string) — ISO 8601-Zeitstempel, als die Aufgabe erstellt wurde.
- `data.updated_at`(string) — ISO 8601-Zeitstempel, als die Aufgabe zuletzt aktualisiert wurde.
- `data.summary_record_id`(string) — Die Kennung des zusammenfassenden Datensatzes, der die Ergebnisse zur Risikominderung enthält. Ist enthalten, wenn eine Zusammenfassung für die abgeschlossene Abhilfemaßnahme generiert wird. Sie können den Inhalt der Zusammenfassung über die AWS DevOps Agenten-API abrufen, indem Sie diese Kennung verwenden, um nach dem Journaldatensatz mit dem Datensatztyp zu suchen. `mitigation_summary_md`

Beispiel: Ereignis „Schadensbegrenzung abgeschlossen“

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

Beispiel: Ereignis „Schadensbegrenzung fehlgeschlagen“

```
{
  "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"
```

```
    }
  }
}
```

Verkaufte Logs und Metriken

Sie können Ihre Agentenbereiche und Servicebetriebe anhand der von Amazon angebotenen CloudWatch Kennzahlen und Protokolle überwachen. In diesem Thema werden die CloudWatch Metriken beschrieben, die der AWS DevOps Agent automatisch auf Ihrem Konto veröffentlicht, und die versendeten Protokolle, die Sie für die Lieferung an Ihre bevorzugten Ziele konfigurieren können.

CloudWatch Verkaufte Metriken

AWS DevOps Der Agent veröffentlicht automatisch Metriken auf Amazon CloudWatch in Ihrem Konto. Diese Metriken sind ohne Konfiguration verfügbar. Sie können sie verwenden, um die Nutzung zu überwachen, betriebliche Aktivitäten zu verfolgen und Alarme zu erstellen.

Serviceverknüpfte Rolle

Damit CloudWatch Amazon-Metriken in Ihrem Konto für diesen Service veröffentlicht werden, erstellt der AWS DevOps Agent automatisch die serviceverknüpfte Rolle AWSServiceRoleForAIDevOps Service-Linked Role für Sie. Wenn die IAM-Rolle, die die API aufruft, nicht über die entsprechenden Berechtigungen verfügt, schlägt die Ressourcenerstellung mit einem fehl. InvalidParameterException

> **Important**
>
> Kunden, die ihre Rolle AgentSpace vor dem 13. März 2026 erstellt haben, müssen die mit dem AWSServiceRoleForAIDevOps-Service verknüpfte Rolle manuell erstellen, damit die CloudWatch Metriken für den AWS DevOps Agenten in ihrem Konto veröffentlicht werden.

Manuell eine serviceverknüpfte Rolle erstellen (für Bestandskunden)

Führen Sie eine der folgenden Aktionen aus:

- Erstellen Sie in der IAM-Konsole die AWSServiceRoleForAIDevOps-Rolle unter dem AWS DevOps Agent-Dienst.
- Führen Sie in der AWS CLI den folgenden Befehl aus:

```
aws iam create-service-linked-role --aws-service-name aidevops.amazonaws.com
```

Namespace

Alle Metriken werden unter dem `AWS/AIDevOps` Namespace veröffentlicht.

Dimensionen

Alle Metriken enthalten die folgende Dimension.

Dimension	Description
`AgentSpaceUUID`	Die eindeutige Kennung des Agentenbereichs. Verwenden Sie CloudWatch mathematische Ausdrücke oder lassen Sie den Dimension sfilter weg, um Metriken für alle Agentenbe reiche in Ihrem Konto zu aggregieren.

Referenz zu den Metriken

Metrikname	Description	Einheit	Häufigkeit der Veröffentlichung	Nützliche Statistiken
ConsumedC hatRequests	Die Anzahl der Chat-Anfr agen, die ein Agentenbereich verbraucht hat. Verwenden Sie die `SUM` Statistik für alle `AgentSpac eUUID` Dimension en, um die Gesamtzahl für	Anzahl	Alle 5 Minuten	Summe, Durchschnitt

Metrikname	Description	Einheit	Häufigkeit der Veröffentlichung	Nützliche Statistiken
	Ihr Konto zu ermitteln.			
ConsumedInvestigationTime	Die Zeit, die für die Durchführung von Ermittlungen in einem Agentenbereich aufgewendet wurde.	Sekunden	Alle 5 Minuten	Summe, Durchschnitt, Maximum
ConsumedEvaluationTime	Die Zeit, die für die Durchführung von Evaluierungen in einem Agentenbereich aufgewendet wurde.	Sekunden	Alle 5 Minuten	Summe, Durchschnitt, Maximum

Metrikname	Description	Einheit	Häufigkeit der Veröffentlichung	Nützliche Statistiken
TopologyCompletionCount	Die Anzahl der Abschlüsse der Topologieverarbeitung. AWS DevOps Der Agent gibt diese Metrik aus, wenn die Verarbeitung einer Topologie abgeschlossen ist, unabhängig davon, ob es sich um die erste Erstellung während des Onboardings, um ein manuelles Update oder um eine geplante tägliche Aktualisierung handelt.	Anzahl	Ereignisgesteuert (wird bei jedem Abschluss ausgegeben)	Summe, SampleCount

Metriken in der CloudWatch Konsole anzeigen

1. Öffnen Sie die CloudWatch -Konsole.
2. Wählen Sie im Navigationsbereich Metrics (Metriken) und dann All metrics (Alle Metriken) aus.
3. Wählen Sie den Namespace AWS/AIDevOps.
4. Wählen Sie Von AgentSpace, um die Kennzahlen für Ihre Agentenbereiche anzuzeigen.

> **Note**
>
> Sie können CloudWatch Alarme für diese Messwerte einrichten, um Benachrichtigungen zu erhalten, wenn die Nutzung einen Schwellenwert überschreitet. Erstellen Sie beispielsweise einen Alarm, `ConsumedChatRequests` um den Verbrauch von Chat-Anfragen zu überwachen.

Voraussetzungen

Bevor Sie die Protokollzustellung konfigurieren, stellen Sie sicher, dass Sie über Folgendes verfügen:

- Ein aktives AWS Konto mit Zugriff auf die AWS DevOps Agent-Konsole
- Ein IAM-Prinzipal mit Berechtigungen für die Übermittlung von CloudWatch Protokollen APIs
- (Optional) Ein Amazon S3 S3-Bucket oder Amazon Data Firehose-Lieferstream, falls Sie diese als Protokollziele verwenden möchten

Vended-Protokolle

AWS DevOps Der Agent unterstützt versendete Logs, die Einblick in Ereignisse bieten, die in Ihren Agenten-Spaces und Service-Registrierungen verarbeitet werden. Verkaufte Logs verwenden die Amazon CloudWatch Logs-Infrastruktur, um Logs an Ihr bevorzugtes Ziel zu liefern.

Um verkaufte Logs verwenden zu können, müssen Sie ein Lieferziel konfigurieren. Die folgenden Ziele werden unterstützt:

- Amazon CloudWatch Logs — Eine Protokollgruppe in Ihrem Konto
- Amazon S3 — Ein S3-Bucket in Ihrem Konto
- Amazon Data Firehose — Ein Firehose-Lieferstream in Ihrem Konto

Unterstützte Protokolltypen

Ein einziger Protokolltyp wird unterstützt:. `APPLICATION_LOGS` Dieser Protokolltyp deckt alle Betriebsereignisse ab, die der Dienst ausgibt.

Ereignistypen protokollieren

In der folgenden Tabelle sind die Ereignisse zusammengefasst, die der AWS DevOps Agent protokolliert.

Veranstaltung	Description	Protokollebene
Eingehendes Agentenereignis empfangen	Ein Agent wird durch eine integrierte Quelle ausgelöst und empfängt ein eingehend es Ereignis (z. B. ein PagerDuty Vorfallereignis).	INFO
Eingehendes Agentenereignis wurde gelöscht	Ein eingehendes Ereignis wurde gelöscht, bevor der Agent es verarbeitet hat. Das Protokoll enthält den Grund (z. B. falsch formatierte Daten).	TBD
Fehler bei der ausgehend en Kommunikation mit dem Agenten	Eine ausgehende Kommunika tion mit einer Drittanbieter-Inte gration ist fehlgeschlagen. Das Protokoll enthält die Aufgaben-ID und die Ziel-ID (z. B. einen Authentifizierungsfehler).	TBD
Topologieerstellung in Warteschlange	Ein Auftrag zur Erstellung einer Topologie wurde zur Verarbeitung in die Warteschl ange gestellt.	INFO
Die Erstellung der Topologie wurde gestartet	Die Verarbeitung eines Jobs zur Erstellung einer Topologie wurde gestartet.	INFO
Die Erstellung der Topologie ist abgeschlossen	Die Verarbeitung eines Jobs zur Erstellung einer Topologie wurde abgeschlossen. Dieses	INFO

Veranstaltung	Description	Protokollebene
	Ereignis gilt für Ersterste llungen, Aktualisierungen und tägliche Aktualisierungen.	
Die Ressourcenerkennung ist fehlgeschlagen	Bei der Ressourcenerkennung bei der Topologieerstellung ist ein Fehler aufgetreten.	ERROR
Die Registrierung des Dienstes ist fehlgeschlagen	Bei der Registrierung des Dienstes ist ein Fehler aufgetreten, der nicht behebbar ist	ERROR
Die Webhook-Validierung schlägt fehl	Wenn der vom DevOps-Ag enten empfangene Webhook nicht dem erwarteten Schema entspricht	ERROR
Statusaktualisierungen zur Bestätigung der Assoziation	Wenn eine Agent-Space-Zuordnung (typisches primary/secondary Konto) erfolgt, ändert sich der Validieru ngsstatus von „gültig“ zu „ungültig“ und umgekehrt (z. B. aufgrund einer falsch formatierten Rolle, von der der Dienst nicht ausgehen kann).	FEHLER/INFORMATION

Berechtigungen

AWS DevOps Der Agent verwendet CloudWatch verkaufte Protokolle (V2-Berechtigungen), um Protokolle zu übermitteln. Um die Protokollzustellung einzurichten, muss die IAM-Rolle, die die Übermittlung konfiguriert, über die folgenden Berechtigungen verfügen:

- `aidevops:AllowVendedLogDeliveryForResource`— Erforderlich, um die Protokollzustellung für die Agent-Space-Ressource zuzulassen.

- Berechtigungen für die CloudWatch Übertragung von Protokollen APIs (`logs:PutDeliverySourcelogs:PutDeliveryDestination`,`logs:CreateDelivery`, und verwandte Vorgänge).
- Spezifische Berechtigungen für das von Ihnen gewählte Lieferziel.

Die vollständige IAM-Richtlinie, die für jeden Zieltyp erforderlich ist, finden Sie in den folgenden Themen im Amazon CloudWatch Logs-Benutzerhandbuch:

- An Logs gesendete Protokolle CloudWatch
- An Amazon S3 gesendete Protokolle
- An Firehose gesendete Protokolle

Konfigurieren Sie die Protokollzustellung (Konsole)

AWS DevOps Der Agent stellt in der AWS Management Console zwei Speicherorte für die Konfiguration der Protokollzustellung zur Verfügung:

- Seite mit den Einstellungen für die Dienstregistrierung — Konfigurieren Sie die Protokollzustellung für Ereignisse auf Service-Ebene. Diese Protokolle verwenden den Dienst ARN (`arn:aws:aidevops:<region>:<account-id>:service/<account-id>`) als Ressource.
- Agent-Space-Seite — Konfigurieren Sie die Protokollzustellung für Ereignisse, die für einen einzelnen Agentenbereich spezifisch sind. Diese Protokolle verwenden den ARN (`arn:aws:aidevops:<region>:<account-id>:agentspace/<agent-space-id>`) des Agentenbereichs als Ressource.

Um die Protokollzustellung für eine Dienstregistrierung zu konfigurieren

1. Öffnen Sie die AWS DevOps Agent-Konsole in der AWS Management-Konsole.
2. Wählen Sie im Navigationsbereich Settings (Einstellungen).
3. Wählen Sie auf der Registerkarte Capability Providers > Logs die Option Configure aus.
4. Wählen Sie als Zieltyp eine der folgenden Optionen aus:
5. CloudWatch Protokolle — Wählen Sie eine Protokollgruppe aus oder erstellen Sie sie.
6. Amazon S3 — Geben Sie den S3-Bucket-ARN ein.
7. Amazon Data Firehose — Wählen oder erstellen Sie einen Firehose-Lieferstream.

8. Für zusätzliche Einstellungen — optional — können Sie die folgenden Optionen angeben:
   a. Wählen Sie unter Feldauswahl die Namen der Protokollfelder aus, die Sie an Ihr Ziel senden möchten. Sie können Zugriffsprotokollfelder und eine Teilmenge von Echtzeitzugriffsprotokollfeldern auswählen.
   b. (Nur Amazon S3) Geben Sie für die Partitionierung den Pfad zur Partitionierung Ihrer Protokolldateidaten an.
   c. (Nur Amazon S3) Für ein Hive-kompatibles Dateiformat können Sie das Kontrollkästchen aktivieren, um Hive-kompatible S3-Pfade zu verwenden. Dies vereinfacht das Laden neuer Daten in Ihre Hive-kompatiblen Tools.
   d. Geben Sie unter Ausgabeformat Ihr bevorzugtes Format an.
   e. Geben Sie unter Feldtrennzeichen an, wie Protokollfelder getrennt werden sollen.
9. Wählen Sie Speichern.
10. Stellen Sie sicher, dass der Lieferstatus Aktiv lautet.

Um die Protokollzustellung für einen Agentbereich zu konfigurieren

1. Öffnen Sie die AWS DevOps Agent-Konsole in der AWS Management-Konsole.
2. Wählen Sie den Agent-Bereich aus, den Sie konfigurieren möchten.
3. Wählen Sie auf der Registerkarte Konfiguration die Option Konfigurieren aus.
4. Wählen Sie als Zieltyp eine der folgenden Optionen aus:
5. CloudWatch Protokolle — Wählen Sie eine Protokollgruppe aus oder erstellen Sie sie.
6. Amazon S3 — Geben Sie den S3-Bucket-ARN ein.
7. Amazon Data Firehose — Wählen oder erstellen Sie einen Firehose-Lieferstream.
8. Für Zusätzliche Einstellungen — *optional* können Sie die folgenden Optionen angeben:
   a. Wählen Sie unter Feldauswahl die Namen der Protokollfelder aus, die Sie an Ihr Ziel senden möchten. Sie können Zugriffsprotokollfelder und eine Teilmenge von Echtzeitzugriffsprotokollfeldern auswählen.
   b. (Nur Amazon S3) Geben Sie für die Partitionierung den Pfad zur Partitionierung Ihrer Protokolldateidaten an.
   c. (Nur Amazon S3) Für ein Hive-kompatibles Dateiformat können Sie das Kontrollkästchen aktivieren, um Hive-kompatible S3-Pfade zu verwenden. Dies vereinfacht das Laden neuer Daten in Ihre Hive-kompatiblen Tools.
   d. Geben Sie unter Ausgabeformat Ihr bevorzugtes Format an.

e. Geben Sie unter Feldtrennzeichen an, wie Protokollfelder getrennt werden sollen.

9. Wählen Sie Speichern.
10. Stellen Sie sicher, dass der Lieferstatus Aktiv lautet.

Konfigurieren Sie die Protokollzustellung (CloudWatch API)

Sie können die CloudWatch Logs-API auch verwenden, um die Protokollzustellung programmgesteuert zu konfigurieren. Eine funktionierende Protokollzustellung besteht aus drei Elementen:

- A DeliverySource— Stellt die AWS DevOps Agent-Space-Ressource dar, die die Protokolle generiert.
- A DeliveryDestination— Stellt das Ziel dar, in das Protokolle geschrieben werden.
- Eine Lieferung — Verbindet eine Lieferquelle mit einem Lieferziel.

Schritt 1: Erstellen Sie eine Lieferquelle

Verwenden Sie den PutDeliverySourceVorgang, um eine Lieferquelle zu erstellen. Übergeben Sie den ARN Ihrer AWS DevOps Agent-Space-Ressource und geben Sie `APPLICATION_LOGS` ihn als Protokolltyp an.

Im folgenden Beispiel wird eine Zustellungsquelle für einen Agentenbereich erstellt:

```
{
    "name": "my-agent-space-delivery-source",
    "resourceArn": "arn:aws:aidevops:us-east-1:123456789012:agentspace/my-agent-space-
id",
    "logType": "APPLICATION_LOGS"
}
```

Im folgenden Beispiel wird eine Zustellungsquelle für den Service erstellt:

```
{
    "name": "my-service-delivery-source",
    "resourceArn": "arn:aws:aidevops:us-east-1:123456789012:service",
    "logType": "APPLICATION_LOGS"
}
```

Schritt 2: Erstellen Sie ein Lieferziel

Verwenden Sie den PutDeliveryDestinationVorgang, um zu konfigurieren, wo Protokolle gespeichert werden. Sie können Amazon CloudWatch Logs, Amazon S3 oder Amazon Data Firehose wählen.

Im folgenden Beispiel wird ein CloudWatch Logs-Ziel erstellt:

```
{
    "name": "my-cwl-destination",
    "deliveryDestinationConfiguration": {
        "destinationResourceArn": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/
aidevops/my-agent-space"
    },
    "outputFormat": "json"
}
```

Das folgende Beispiel erstellt ein Amazon S3 S3-Ziel:

```
{
    "name": "my-s3-destination",
    "deliveryDestinationConfiguration": {
        "destinationResourceArn": "arn:aws:s3:::my-aidevops-logs-bucket"
    },
    "outputFormat": "json"
}
```

Im folgenden Beispiel wird ein Amazon Data Firehose erstellt:

```
{
    "name": "my-firehose-destination",
    "deliveryDestinationConfiguration": {
        "destinationResourceArn": "arn:aws:firehose:us-
east-1:123456789012:deliverystream/my-aidevops-log-stream"
    },
    "outputFormat": "json"
}
```

> **Note**
>
> Wenn Sie Logs kontoübergreifend versenden, müssen Sie dies PutDeliveryDestinationPolicyim Zielkonto verwenden, um die Lieferung zu autorisieren.

Wenn Sie verwenden möchten CloudFormation, können Sie Folgendes verwenden:

- Delivery
- DeliveryDestination
- DeliverySource

Der `ResourceArn` ist der `AgentSpaceArn` und `LogType` muss `APPLICATION_LOGS` als unterstützten Protokolltyp aufweisen.

Schritt 3: Eine Lieferung erstellen

Verwenden Sie den CreateDeliveryVorgang, um die Lieferquelle mit dem Lieferziel zu verknüpfen.

```
{
    "deliverySourceName": "my-agent-space-delivery-source",
    "deliveryDestinationArn": "arn:aws:logs:us-east-1:123456789012:delivery-
destination:my-cwl-destination"
}
```

AWS CloudFormation

Sie können die Protokollzustellung auch AWS CloudFormation mithilfe der folgenden Ressourcen konfigurieren:

- AWS: :Protokolle:: DeliverySource
- AWS: :Protokolle:: DeliveryDestination
- AWS: :Logs: :Lieferung

Stellen Sie `ResourceArn` den AWS DevOps Agent-Bereich oder den Dienst-ARN ein und legen Sie ihn `LogType` auf fest`APPLICATION_LOGS`.

Referenz zum Protokollschema

AWS DevOps Der Agent verwendet ein gemeinsames Protokollschema für alle Ereignistypen. Nicht jedes Protokollereignis verwendet jedes Feld.

In der folgenden Tabelle werden die Felder im Protokollschema beschrieben.

Feld	Typ	Description
event_timestamp	Long	Unix-Zeitstempel, wann das Ereignis eingetreten ist
resource_arn	Zeichenfolge	ARN der Ressource, die das Ereignis generiert hat
optional_account_id	Zeichenfolge	AWS Konto-ID, die dem Protokoll zugeordnet ist.
optional_level	Zeichenfolge	Protokollebene:,, `INFO` `WARN` `ERROR`
optional_agent_space_id	Zeichenfolge	Bezeichner des Agentenbereichs.
optional_association_id	Zeichenfolge	Zuordnungs-ID für das Protokoll.
optional_status	Zeichenfolge	Status des Topologievorgangs.
optional_webhook_id	Zeichenfolge	Webhook-Kennung.
optional_mcp_endpoint_url	Zeichenfolge	URL des MCP-Serverendpunkts
optional_service_type	Zeichenfolge	Art des Dienstes:`DYNATRACE` ,,,,. `DATADOG` `GITHUB` `SLACK` `SERVICENOW`
optional_service_endpoint_url	Zeichenfolge	Endpunkt-URL für Integrationen von Drittanbietern.
optional_service_id	Zeichenfolge	Bezeichner der Quelle.

Feld	Typ	Description
request_id	Zeichenfolge	Anforderungs-ID für die Korrelation mit AWS CloudTrai l oder für Support-Tickets.
optional_Operation	Zeichenfolge	Name der Operation, die ausgeführt wurde.
optional_task_type	Zeichenfolge	Aufgabentyp für das Agenten-Backlog: oder `INVESTIGA TION EVALUATION`
optional_task_id	Zeichenfolge	ID der Backlog-Aufgabe für die Backlog-Aufgabe des Agenten. IDAgent
optional_reference	Zeichenfolge	Referenz aus einer Agentenau fgabe (z. B. einem Jira-Ticket).
optional_error_type	Zeichenfolge	Fehlertyp
optionale_Fehlermeldung	Zeichenfolge	Beschreibung des Fehlers, wenn ein Vorgang fehlschlägt.
optional_details	Zeichenfolge (JSON)	Dienstspezifische Event-Pay load, die Betriebsparameter und Ergebnisse enthält.

Verwaltung und Deaktivierung der Protokollzustellung

Sie können die Protokollzustellung jederzeit über die AWS DevOps Agentenkonsole in der AWS Management Console oder mithilfe der CloudWatch Logs-API ändern oder entfernen.

Protokollzustellung verwalten (Konsole)

1. Öffnen Sie die AWS DevOps Agent-Konsole in der AWS Management-Konsole.
2. Navigieren Sie zur Seite „Einstellungen“ (für Protokolle auf Service-Ebene) oder zur entsprechenden Seite auf Agent Space-Ebene (für Protokolle auf Agent Space-Ebene).

3. Wählen Sie auf der Registerkarte Konfiguration (für Protokolle auf Agent Space-Ebene) oder auf der Registerkarte Capability Providers > Logs (für Logs auf Service-Ebene) die zu ändernde Lieferung aus.
4. Aktualisieren Sie die Konfiguration nach Bedarf und wählen Sie Speichern.

> Hinweis: Sie können den Zieltyp einer bestehenden Lieferung nicht ändern. Um den Zieltyp zu ändern, löschen Sie die aktuelle Lieferung und erstellen Sie eine neue.

Deaktivieren Sie die Protokollzustellung (Konsole)

1. Öffnen Sie die AWS DevOps Agent-Konsole in der AWS Management-Konsole.
2. Navigieren Sie zur Seite „Einstellungen“ (für Protokolle auf Service-Ebene) oder zur entsprechenden Seite auf Agent Space-Ebene (für Protokolle auf Agent Space-Ebene).
3. Wählen Sie auf der Registerkarte Konfiguration (für Protokolle auf Agent Space-Ebene) oder auf der Registerkarte Capability Providers > Logs (für Logs auf Service-Ebene) die Lieferung aus, die entfernt werden soll.
4. Wählen Sie Löschen und bestätigen Sie.

Deaktivieren Sie die Protokollzustellung (API)

Um eine Protokollzustellung mithilfe der API zu entfernen, löschen Sie die Ressourcen in der folgenden Reihenfolge:

1. Löschen Sie die Lieferung mithilfe von DeleteDelivery.
2. Löschen Sie die Lieferquelle mithilfe von DeleteDeliverySource.
3. (Optional) Wenn das Lieferziel nicht mehr benötigt wird, löschen Sie es mit DeleteDeliveryDestination.

> **Important**
>
> Sie sind dafür verantwortlich, die Ressourcen für die Protokollzustellung zu entfernen, nachdem Sie die Agentenbereichsressource gelöscht haben, die die Protokolle generiert (z. B. nachdem Sie einen Agentbereich gelöscht haben). Wenn Sie diese Ressourcen nicht entfernen, bleiben möglicherweise verwaiste Übermittlungskonfigurationen bestehen.

Preisgestaltung

Der AWS DevOps Agent berechnet keine Gebühren für die Aktivierung von versendeten Protokollen. Je nach dem von Ihnen ausgewählten Ziel für die Protokollbereitstellung können jedoch Gebühren für die Bereitstellung, Erfassung, Speicherung oder den Zugriff anfallen. Preisinformationen finden Sie unter Verkaufte Logs auf der Registerkarte Logs bei Amazon CloudWatch Pricing.

Die länderspezifischen Preise finden Sie im Folgenden:

- Amazon CloudWatch Logs — Preise
- Amazon S3 – Preise
- Amazon Data Firehose – Preise

Verbindung zu privat gehosteten Tools herstellen

Übersicht über private Verbindungen

AWS DevOps Der Agent kann mit benutzerdefinierten Model Context Protocol (MCP) -Tools und anderen Integrationen erweitert werden, die dem Agenten Zugriff auf interne Systeme wie private Paketregister, selbst gehostete Observability-Plattformen, interne Dokumentations-APIs und Quellcodeverwaltungsinstanzen gewähren (siehe:). Konfiguration von Funktionen für AWS DevOps Agent Diese Dienste werden häufig in einer Amazon Virtual Private Cloud (Amazon VPC) mit eingeschränktem oder keinem öffentlichen Internetzugang ausgeführt, was bedeutet, dass der AWS DevOps Agent sie standardmäßig nicht erreichen kann.

Private Verbindungen für AWS DevOps Agenten ermöglichen es Ihnen, Ihren Agent Space sicher mit Diensten zu verbinden, die in Ihrer VPC ausgeführt werden, ohne sie dem öffentlichen Internet auszusetzen. Private Verbindungen funktionieren mit jeder Integration, die einen privaten Endpunkt erreichen muss, einschließlich MCP-Servern, selbst gehosteten Grafana- oder Splunk-Instanzen und Quellcodeverwaltungssystemen wie Enterprise Server und. GitHub GitLab Self-Managed

> **Note**
>
> Wenn Ihre privat gehosteten Tools von Ihrer VPC aus ausgehende Anfragen an den AWS DevOps Agenten senden, kann dieser Datenverkehr auch mithilfe eines VPC-Endpunkts gesichert werden, sodass er im Netzwerk bleibt. AWS Dies kann beispielsweise mit Tools verwendet werden, die den DevOps Agenten über Webhook-Ereignisse auslösen (siehe:).

the section called “DevOps Agent über Webhook aufrufen” Weitere Informationen finden Sie unter the section called “VPC-Endpunkte (AWS PrivateLink)”.

Wie funktionieren private Verbindungen

Eine private Verbindung erstellt einen sicheren Netzwerkpfad zwischen dem AWS DevOps Agenten und einer Zielressource in Ihrer VPC. Unter der Haube verwendet AWS DevOps Agent Amazon VPC Lattice, um diesen sicheren privaten Konnektivitätspfad einzurichten. VPC Lattice ist ein Anwendungsnetzwerkdienst, mit dem Sie die Kommunikation zwischen Anwendungen über VPCs, Konten und Rechnungstypen hinweg verbinden, sichern und überwachen können, ohne die zugrunde liegende Netzwerkinfrastruktur verwalten zu müssen.

Wenn Sie eine private Verbindung herstellen, passiert Folgendes:

- Sie stellen die VPC, Subnetze und (optional) Sicherheitsgruppen bereit, die Netzwerkkonnektivität zu Ihrem Zieldienst haben.
- AWS DevOps Der Agent erstellt ein vom Service verwaltetes Ressourcen-Gateway und stellt seine Elastic Network Interfaces (ENIs) in den von Ihnen angegebenen Subnetzen bereit.
- Der Agent verwendet das Ressourcen-Gateway, um den Datenverkehr über den privaten Netzwerkpfad an die IP-Adresse oder den DNS-Namen Ihres Zieldienstes weiterzuleiten.

Das Ressourcen-Gateway wird vollständig vom AWS DevOps Agenten verwaltet und erscheint als schreibgeschützte Ressource in Ihrem Konto (benannt`aidevops-{your-private-connection-name}`). Sie müssen es nicht konfigurieren oder verwalten. Die einzigen Ressourcen, die in Ihrer VPC erstellt werden, sind ENIs in den von Ihnen angegebenen Subnetzen. Diese ENIs dienen als Einstiegspunkt für privaten Datenverkehr und werden vollständig vom Service verwaltet. Sie akzeptieren keine eingehenden Verbindungen aus dem Internet, und Sie behalten über Ihre eigenen Sicherheitsgruppen die volle Kontrolle über ihren Datenverkehr.

Sicherheit

Private Verbindungen sind mit mehreren Sicherheitsebenen konzipiert:

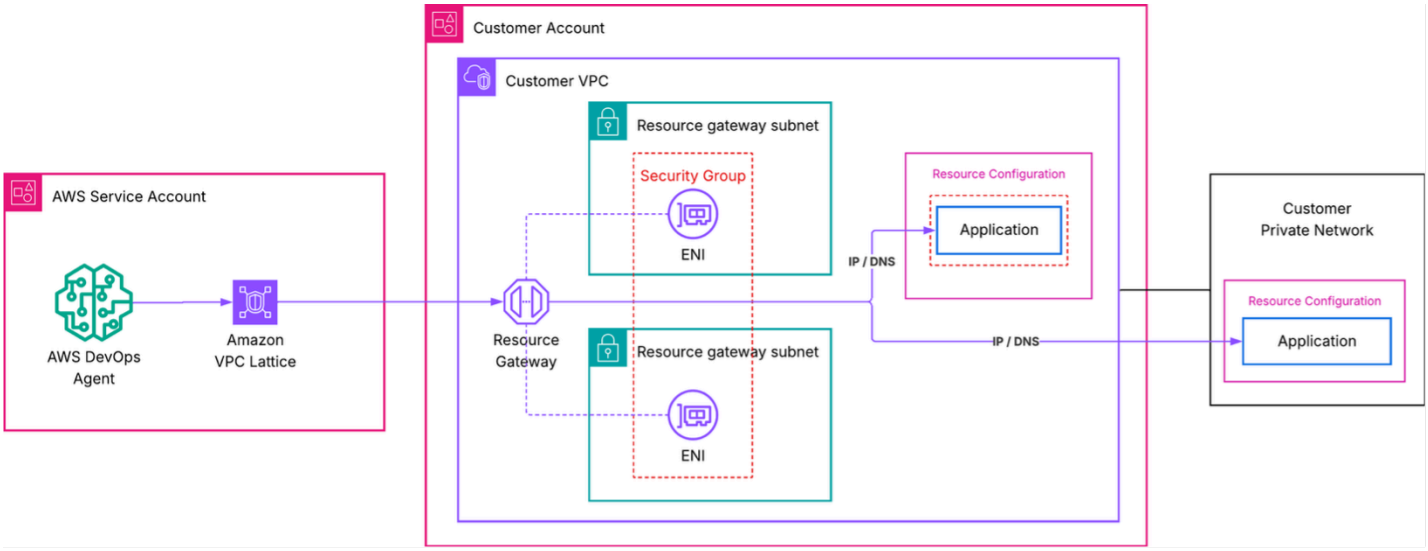
- Keine Gefahr für das öffentliche Internet — Der gesamte Datenverkehr zwischen dem AWS DevOps Agenten und Ihrem Zieldienst verbleibt im AWS Netzwerk. Ihr Service benötigt niemals eine öffentliche IP-Adresse oder ein Internet-Gateway.

- Service-controlled Resource Gateway — Das vom Service verwaltete Resource Gateway ist in Ihrem Konto schreibgeschützt. Es kann nur vom AWS DevOps Agenten verwendet werden, und kein anderer Dienst oder Principal kann den Datenverkehr über es weiterleiten. Sie können dies in AWS CloudTrailProtokollen überprüfen, in denen alle VPC Lattice API-Aufrufe aufgezeichnet werden.
- Ihre Sicherheitsgruppen, Ihre Regeln — Sie kontrollieren den ein- und ausgehenden Datenverkehr zu den ENIs über Sicherheitsgruppen, die Ihnen gehören und die Sie verwalten. Wenn Sie keine Sicherheitsgruppen angeben, erstellt der AWS DevOps Agent eine Standardsicherheitsgruppe, die auf die von Ihnen definierten Ports beschränkt ist.
- Service-linked Rollen mit den geringsten Rechten — AWS DevOps Der Agent verwendet eine serviceverknüpfte Rolle, um nur die erforderlichen VPC Lattice- und Amazon EC2 EC2-Ressourcen zu erstellen. Diese Rolle ist auf Ressourcen beschränkt, die mit markiert sind, `AWSAIDevOpsManaged` und kann nicht auf andere Ressourcen in Ihrem Konto zugreifen.

> **Note**
>
> Wenn Ihre Organisation über Service Control Policies (SCPs) verfügt, die VPC Lattice API-Aktionen einschränken, wird das vom Service verwaltete Ressourcen-Gateway über eine serviceverknüpfte Rolle erstellt. Stellen Sie sicher, dass Ihre SCPs die erforderlichen Aktionen für die serviceverknüpfte Rolle zulassen.

Architektur

Das folgende Diagramm zeigt den Netzwerkpfad für eine private Verbindung.



In dieser Architektur:

- AWS DevOps Der Agent initiiert eine Anfrage an Ihren Zieldienst.
- Amazon VPC Lattice leitet die Anfrage über das vom Service verwaltete Ressourcen-Gateway in Ihrer VPC weiter. Informationen zu erweiterten Setups, die Ihre eigenen VPC-Lattice-Ressourcen verwenden, finden Sie unter Erweiterte Einrichtung mithilfe vorhandener VPC-Lattice-Ressourcen.
- Eine ENI in Ihrer VPC empfängt den Datenverkehr und leitet ihn an die IP-Adresse oder den DNS-Namen Ihres Zieldienstes weiter.
- Ihre Sicherheitsgruppen regeln, welcher Datenverkehr über die ENIs zugelassen wird.
- Aus Sicht Ihres Zieldienstes stammt die Anfrage von privaten IP-Adressen von ENIs innerhalb Ihrer VPC.

Erstellen Sie eine private Verbindung

Sie können mit der AWS Management Console oder der AWS CLI eine private Verbindung herstellen.

> **Note**
>
> Die folgenden Availability Zones werden von VPC Lattice nicht unterstützt:`use1-az3`,`usw1-az2`,`apne1-az3`,`apne2-az2`,, `euc1-az2euw1-az4`,`cac1-az3`. `ilc1-az2`

Voraussetzungen

Bevor Sie eine private Verbindung herstellen, stellen Sie sicher, dass Sie über Folgendes verfügen:

- Ein aktiver Agent Space — Sie benötigen einen vorhandenen Agent Space in Ihrem Konto. Falls Sie noch keines haben, beachten Sie die Informationen unter Erste Schritte mit AWS DevOps Agent.
- Ein privat erreichbarer Zieldienst — Ihr MCP-Server, Ihre Observability-Plattform oder ein anderer Dienst muss über eine bekannte private IP-Adresse oder einen DNS-Namen von der VPC aus erreichbar sein, in der das Ressourcen-Gateway bereitgestellt ist. Der Dienst kann in derselben VPC, einer Peer-VPC oder lokal ausgeführt werden, sofern er von den Subnetzen des Ressourcen-Gateways aus routbar ist. Der Dienst muss HTTPS-Verkehr mit einer TLS-Mindestversion von 1.2 an einem Port bereitstellen, den Sie beim Herstellen der Verbindung angeben.
- Subnetze in Ihrer VPC — Identifizieren Sie 1—20 Subnetze, in denen die ENIs erstellt werden. Wir empfehlen, Subnetze in mehreren Availability Zones auszuwählen, um eine hohe Verfügbarkeit zu gewährleisten. Diese Subnetze müssen über eine Netzwerkverbindung zu Ihrem Zieldienst verfügen. Ein Subnetz pro Availability Zone kann von VPC Lattice verwendet werden.
- (Optional) Sicherheitsgruppen — Wenn Sie den Datenverkehr mit bestimmten Regeln kontrollieren möchten, bereiten Sie bis zu fünf Sicherheitsgruppen-IDs vor, die an die ENIs angehängt werden sollen. Wenn Sie Sicherheitsgruppen weglassen, erstellt der AWS DevOps Agent eine Standardsicherheitsgruppe.

Private Verbindungen sind Ressourcen auf Kontoebene. Nachdem Sie eine private Verbindung erstellt haben, können Sie sie für mehrere Integrationen und Agent Spaces wiederverwenden, die denselben Host erreichen müssen.

Erstellen Sie mithilfe der Konsole eine private Verbindung

1. Öffnen Sie die AWS DevOps Agent-Konsole.
2. Wählen Sie im Navigationsbereich Capability Providers und anschließend Private Verbindungen aus.
3. Wählen Sie Neue Verbindung erstellen aus.
4. Geben Sie unter Name einen beschreibenden Namen für die Verbindung ein, z. B. `my-mcp-tool-connection`
5. Wählen Sie für VPC die VPC aus, auf der die Resource Gateway-ENIs bereitgestellt werden.

6. Wählen Sie für Subnetze ein oder mehrere Subnetze (bis zu 20) aus. Wir empfehlen, Subnetze in mindestens zwei Availability Zones auszuwählen.
7. Wählen Sie als IP-Adresstyp den IP-Adresstyp Ihres Zieldienstes (`IPv4`IPv6, oder`DualStack`) aus.
8. (Optional) Wenn Sie für Anzahl der IPv4-Adressen IPv4 oder Dualstack als IP-Adresstyp ausgewählt haben, können Sie die Anzahl der IPv4-Adressen pro ENI für Ihr Ressourcen-Gateway eingeben. Die Standardeinstellung ist 16 IPv4-Adressen pro ENI.
9. (Optional) Wählen Sie für Sicherheitsgruppen bestehende Sicherheitsgruppen (bis zu 5) aus, um einzuschränken, welcher Datenverkehr Ihren Zieldienst erreichen darf. Wenn Sie keine auswählen, wird eine Standardsicherheitsgruppe erstellt.
10. (Optional) Geben Sie für Portbereiche die TCP-Ports an, auf denen Ihre Zielanwendung lauscht (z. B. `443` oder`8080-8090`). Sie können bis zu 11 Portbereiche angeben.
11. Geben Sie als Hostadresse die IP-Adresse oder den DNS-Namen Ihres Zieldienstes ein (z. B. `mcp.internal.example.com` oder`10.0.1.50`). Der Service muss von der ausgewählten VPC aus erreichbar sein. Wenn Sie einen DNS-Namen wählen, muss dieser öffentlich auflösbar sein.
12. (Optional) Geben Sie für Certificate public key den öffentlichen Schlüssel des Zertifikats ein, wenn die angegebene Hostadresse TLS-Zertifikate verwendet, die PEM-encoded von einer privaten Zertifizierungsstelle ausgestellt wurden. Dadurch kann der AWS DevOps Agent der TLS-Verbindung zu Ihrem Zieldienst vertrauen.
13. Wählen Sie Create Connection (Verbindung erstellen) aus.

Der Verbindungsstatus ändert sich zu Create in progress. Dieser Vorgang kann bis zu 10 Minuten dauern. Wenn der Status auf Aktiv wechselt, ist der Netzwerkpfad bereit.

Wenn sich der Status in Create failed (Create failed) ändert, überprüfen Sie Folgendes:

- Die von Ihnen angegebenen Subnetze verfügen über verfügbare IP-Adressen.
- Ihr Konto hat die VPC Lattice-Dienstkontingente nicht erreicht.
- Es gibt keine restriktiven IAM-Richtlinien, die verhindern, dass die serviceverknüpfte Rolle Ressourcen erstellt.

> **Note**
>
> Diese Schritte können auch ausgeführt werden, indem Sie `Create a new private connection` bei der Registrierung einen Capability Provider auswählen. Weitere

Informationen finden Sie unter Verwenden einer privaten Verbindung mit einem Capability Provider.

Erstellen Sie eine private Verbindung mit dem AWS CLI

Führen Sie den folgenden Befehl aus, um eine private Verbindung herzustellen. Ersetzen Sie die Platzhalterwerte durch Ihre eigenen.

```
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"]
        }
    }'
```

Die Antwort enthält den Verbindungsnamen und den folgenden Status: `CREATE_IN_PROGRESS`

```
{
    "name": "my-mcp-tool-connection",
    "status": "CREATE_IN_PROGRESS",
    "resourceGatewayId": "rgw-0123456789abcdef0",
    "hostAddress": "mcp.internal.example.com",
    "vpcId": "vpc-0123456789abcdef0"
}
```

Verwenden Sie den folgenden `describe-private-connection` Befehl, um den Verbindungsstatus zu überprüfen:

```
aws devops-agent describe-private-connection \
```

```
    --name my-mcp-tool-connection
```

Wenn der Status lautet`ACTIVE`, ist Ihre private Verbindung einsatzbereit.

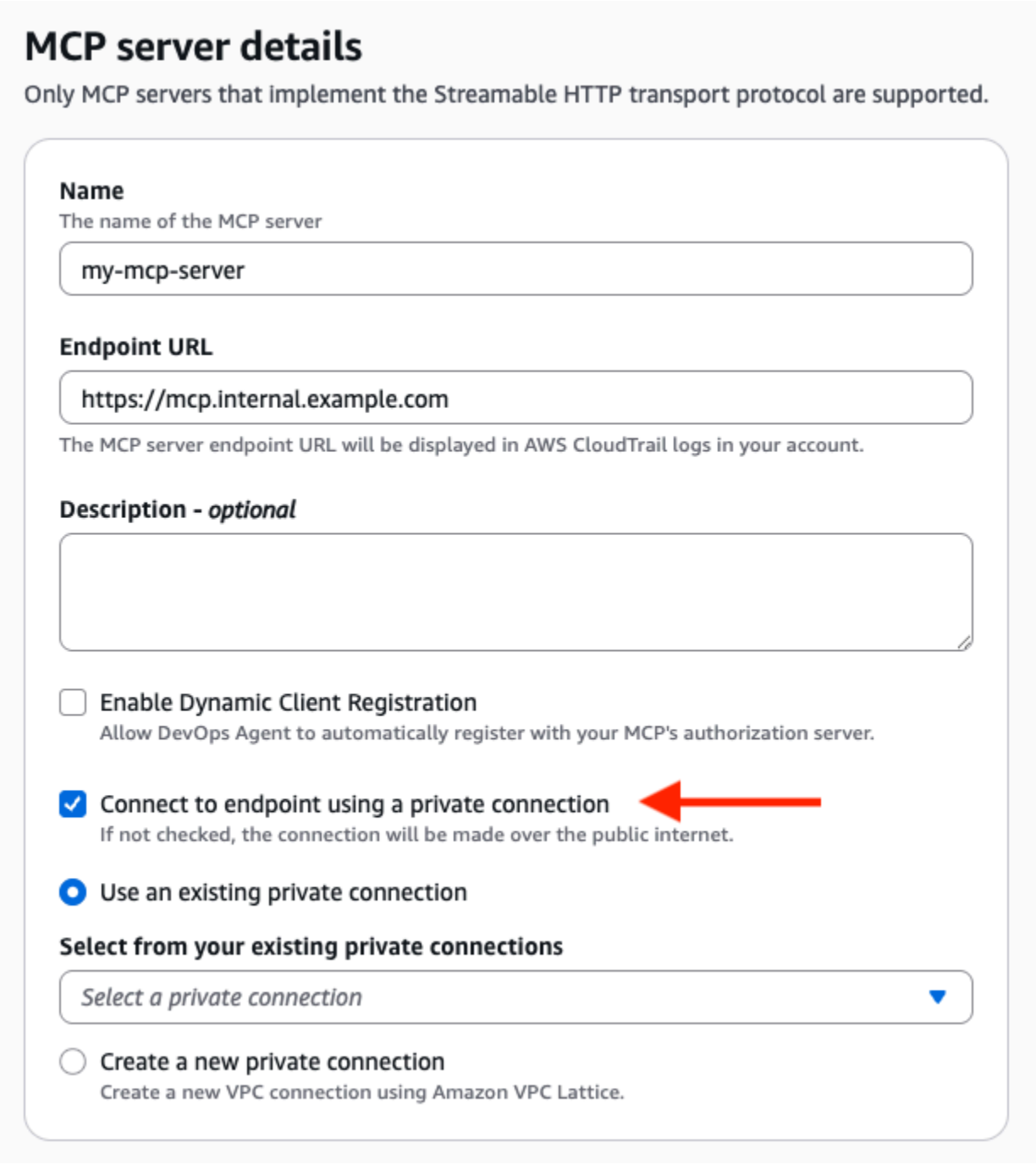
Verwenden Sie eine private Verbindung mit einem Capability Provider

Um eine private Verbindung zu verwenden, können Sie bei der Registrierung eines Capability Providers eine Verbindung zu dieser herstellen. Zu den unterstützten Funktionen, die mit privaten Verbindungen verwendet werden können`GitHub`, gehören:`GitLab`,`MCP Server`, und`Grafana`. Sie können diesen Schritt mit der AWS Management Console oder der AWS CLI ausführen.

> **Note**
>
> Bei der Registrierung eines Capability Providers überprüft der AWS DevOps Agent, ob der Endpunkt erreichbar ist und reagiert. Stellen Sie sicher, dass Ihr Zieldienst läuft und Verbindungen akzeptiert, bevor Sie die Registrierung abschließen.

Verwenden Sie über die Konsole eine private Verbindung mit einem Capability Provider

In der AWS DevOps Agentenkonsole können private Verbindungen während der Registrierung mit einer Funktion verknüpft werden, indem die Option „Über eine private Verbindung mit Endpunkt Connect“ ausgewählt wird.



1. Öffnen Sie die AWS DevOps Agent-Konsole und navigieren Sie zu Ihrem Agent Space.
2. Wählen Sie im Bereich Capability Providers die Option Registrierung aus.
3. Wählen Sie Registrieren für den Funktionstyp aus, den Sie mit der privaten Verbindung verwenden möchten.

4. Geben Sie in der Ansicht mit den Registrierungsdetails die Endpunkt-URL ein, zu der Sie über die private Verbindung eine Verbindung herstellen möchten (z. B.`https://mcp.internal.example.com`).
5. Wählen Sie Über eine private Verbindung mit Endpunkt Connect aus.
6. Wählen Sie entweder eine bestehende private Verbindung aus, die der Endpunkt-URL entspricht, zu der Sie eine Verbindung herstellen möchten, oder wählen Sie Neue private Verbindung erstellen, um eine Verbindung zu erstellen.
7. Schließen Sie den Registrierungsprozess für den Capability Provider ab.

> **Note**
>
> Wenn Sie eine private Verbindung für einen Capability Provider auswählen, der die OAuth-Authentifizierung (Client Credentials oder 3LO) verwendet, gilt die private Verbindung sowohl für den Capability Provider-Endpunkt als auch für den Token-Exchange-Endpunkt. Stellen Sie sicher, dass die private Verbindung mit einer Hostadresse konfiguriert ist, die den Datenverkehr an beide Endpunkte weiterleiten kann.

Verwenden Sie eine private Verbindung mit einem Capability Provider unter Verwendung des AWS CLI

Sie können Capabilities mit einer privaten Verbindung registrieren, indem Sie das `private-connection-name` Argument angeben. Im Folgenden finden Sie ein Beispiel für die Registrierung eines MCP-Servers mit API-Schlüsselautorisierung über die `my-mcp-tool-connection` private Verbindung. Ersetzen Sie die Platzhalterwerte durch Ihre eigenen.

```
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
```

Überprüfen Sie eine private Verbindung

Nachdem die private Verbindung den Status Aktiv erreicht hat und von einem Capability Provider genutzt wurde, stellen Sie sicher, dass der AWS DevOps Agent Ihren Zieldienst erreichen kann:

1. Öffnen Sie die AWS DevOps Agent-Konsole und navigieren Sie zu Ihrem Agent-Bereich.
2. Starten Sie eine neue Chat-Sitzung.
3. Rufen Sie einen Befehl auf, der die Integration verwendet, die von Ihrer privaten Verbindung unterstützt wird. Wenn Ihr MCP-Tool beispielsweise Zugriff auf eine interne Wissensdatenbank bietet, stellen Sie dem Agenten eine Frage, für die diese Wissensdatenbank erforderlich ist.
4. Vergewissern Sie sich, dass der Agent Ergebnisse aus dem privaten Dienst zurückgibt.

Wenn die Verbindung fehlschlägt, überprüfen Sie Folgendes:

- VPC-Lattice-Grenzwerte — Stellen Sie sicher, dass Sie keine Ressourcen-Gateway- oder andere VPC-Lattice-Kontingentgrenzen erreicht haben
- Sicherheitsgruppenregeln — Stellen Sie sicher, dass die Sicherheitsgruppen, die den ENIs zugeordnet sind, ausgehenden Datenverkehr an dem Port zulassen, den Ihr Dienst überwacht. Stellen Sie außerdem sicher, dass die Sicherheitsgruppe Ihres Dienstes eingehenden Datenverkehr auf dem Zielport zulässt. Der Datenverkehr kommt von VPC-Lattice-Datenebenen-IPs innerhalb Ihres VPC-CIDR-Bereichs. Sie können die Sicherheitsgruppenreferenzierung verwenden (die ENI-Sicherheitsgruppe als Quelle zulassen) oder eingehende Daten aus der VPC CIDR zulassen.
- Subnetzkonnektivität — Stellen Sie sicher, dass die ausgewählten Subnetze den Datenverkehr an Ihren Service weiterleiten können. Wenn der Dienst in einem anderen Subnetz ausgeführt wird, stellen Sie sicher, dass die Routing-Tabellen den Verkehr zwischen ihnen zulassen.
- Dienstverfügbarkeit — Stellen Sie sicher, dass Ihr Dienst läuft und Verbindungen über den erwarteten Port akzeptiert.
- Nicht unterstützte Availability Zone — Stellen Sie sicher, dass sich Ihre Subnetze in unterstützten Availability Zones befinden. Führen Sie die Ausführung aus `aws ec2`

`describe-subnets --subnet-ids <your-subnet-ids> --query 'Subnets[*].[SubnetId,AvailabilityZoneId]'` und vergleichen Sie sie mit den oben aufgeführten Availability Zones, die nicht unterstützt werden.

Löschen Sie eine private Verbindung

Sie können ungenutzte private Verbindungen mit der AWS Management Console oder der AWS CLI löschen.

Löschen Sie eine private Verbindung mithilfe der Konsole

1. Öffnen Sie die AWS DevOps Agent-Konsole.
2. Wählen Sie im Navigationsbereich Capability Providers und anschließend Private Verbindungen aus.
3. Wählen Sie das Aktionsmenü für die private Verbindung aus, die Sie löschen möchten, und wählen Sie Entfernen aus.

Die private Verbindung wird mit dem Status „Verbindung wird entfernt“ angezeigt, während der AWS DevOps Agent das verwaltete Ressourcen-Gateway und die ENIs von Ihrer VPC entfernt. Nach Abschluss des Löschvorgangs erscheint die Verbindung nicht mehr in Ihrer Liste der privaten Verbindungen.

Löschen Sie eine private Verbindung mit AWS CLI

```
aws devops-agent delete-private-connection \
    --name my-mcp-tool-connection
```

Die Antwort gibt den Status von zurück`DELETE_IN_PROGRESS`. AWS DevOps Der Agent entfernt das Managed Resource Gateway und die ENIs aus Ihrer VPC. Nach Abschluss des Löschvorgangs erscheint die Verbindung nicht mehr in Ihrer Liste der privaten Verbindungen.

Erweitertes Setup unter Verwendung vorhandener VPC-Lattice-Ressourcen

Wenn Ihre Organisation bereits Amazon VPC Lattice verwendet und Ihre eigenen Ressourcenkonfigurationen verwaltet, können Sie im selbstverwalteten Modus eine private Verbindung erstellen. Anstatt den AWS DevOps Agenten ein Ressourcen-Gateway für Sie

erstellen zu lassen, geben Sie den Amazon-Ressourcennamen (ARN) einer vorhandenen Ressourcenkonfiguration an, die auf Ihren Zielservice verweist.

Dieser Ansatz ist nützlich, wenn Sie:

- Sie möchten die volle Kontrolle über das Ressourcen-Gateway und den Lebenszyklus der Ressourcenkonfiguration haben.
- Sie müssen Ressourcenkonfigurationen für mehrere AWS Konten oder Dienste gemeinsam nutzen.
- Für eine detaillierte Verkehrsüberwachung sind VPC Lattice-Zugriffsprotokolle erforderlich.
- Führen Sie eine Hub-and-Spoke-Netzwerkarchitektur aus.

So erstellen Sie eine selbstverwaltete private Verbindung mit der AWS CLI:

```
aws devops-agent create-private-connection \
    --name my-advanced-connection \
    --mode '{
        "selfManaged": {
            "resourceConfigurationId": "arn:aws:vpc-lattice:us-
east-1:123456789012:resourceconfiguration/rcfg-0123456789abcdef0"
        }
    }'
```

Weitere Informationen zur Einrichtung von VPC Lattice-Ressourcen-Gateways und Ressourcenkonfigurationen finden Sie im Amazon VPC Lattice-Benutzerhandbuch.

Verwandte Themen

- the section called “VPC-Endpunkte (AWS PrivateLink)”
- the section called “MCP-Server verbinden”
- Konfiguration von Funktionen für AWS DevOps Agent
- AWS DevOps Agentensicherheit
- the section called “DevOps IAM-Berechtigungen für Agenten”

AWS DevOps Agentensicherheit

Dieses Dokument enthält Informationen zu Sicherheitsaspekten, Datenschutz, Zugriffskontrollen und Compliance-Funktionen für AWS DevOps Agent. Anhand dieser Informationen erfahren Sie, wie AWS DevOps Agent so konzipiert ist, dass er Ihre Sicherheits- und Compliance-Anforderungen erfüllt.

Multi-layered Sicherheit

AWS DevOps Der Agent implementiert Sicherheit auf mehreren Ebenen. Selbst wenn der IAM-Rolle des Agenten umfassendere Berechtigungen gewährt werden, setzt der Agent seine eigenen internen Zugriffskontrollen durch, um den Umfang seiner Aktionen einzuschränken.

Wir empfehlen, bei der Konfiguration der IAM-Berechtigungen für den AWS DevOps Agenten das Prinzip der geringsten Rechte zu beachten und die Sicherheit auf mehreren Ebenen zu implementieren. Eine umfassende Abwehr stellt sicher, dass keine einzige Fehlkonfiguration die Sicherheit Ihrer Umgebung gefährden kann.

Bereiche für Agenten

Agent Spaces dienen als primäre Sicherheitsgrenze in AWS DevOps Agent. Jeder Agentenbereich:

- Arbeitet unabhängig mit eigenen Konfigurationen und Berechtigungen
- Definiert, AWS auf welche Konten und Ressourcen der Agent zugreifen kann
- Stellt Verbindungen zu Plattformen von Drittanbietern her

Agent Spaces sind strikt isoliert, um die Sicherheit zu gewährleisten und unbeabsichtigten Zugriff über verschiedene Umgebungen oder Teams hinweg zu verhindern.

Regionale Verarbeitung und Datenfluss

AWS DevOps Agent ist weltweit tätig und verfügt über regionale Verarbeitungskapazitäten. Der Agent ruft Betriebsdaten aus AWS Regionen aller AWS Konten ab, denen innerhalb des konfigurierten Agentenbereichs Zugriff gewährt wurde. Diese kontenübergreifende Datenerfassung für mehrere Regionen gewährleistet eine umfassende Analyse von Vorfällen und berücksichtigt gleichzeitig die geografischen Grenzen für die Inferenzverarbeitung.

Nutzung von Amazon Bedrock und regionsübergreifende Inferenz

AWS DevOps Der Agent wählt automatisch die optimale Region innerhalb Ihrer Region aus, um Ihre Inferenzanfragen zu bearbeiten. Dies maximiert die verfügbaren Rechenressourcen und die Modellverfügbarkeit und sorgt für ein optimales Kundenerlebnis. Ihre Daten bleiben nur in der Region gespeichert, in der Ihr Agent Space erstellt wurde. Eingabeaufforderungen und Ausgabeergebnisse können jedoch außerhalb dieser Region verarbeitet werden, wie in der folgenden Liste beschrieben. Alle Daten werden bei der Übertragung über das sichere Netzwerk von Amazon verschlüsselt.

AWS DevOps Der Agent leitet Ihre Inferenzanfragen wie folgt sicher an verfügbare Rechenressourcen in dem geografischen Gebiet weiter, aus dem die Anfrage stammt:

- Inferenzanfragen mit Ursprung in der Europäischen Union werden innerhalb der Europäischen Union bearbeitet.
- Inferenzanfragen mit Ursprung in den Vereinigte Staaten werden in den Vereinigte Staaten bearbeitet.
- Inferenzanfragen mit Ursprung in Australien werden innerhalb Australien bearbeitet.
- Inferenzanfragen mit Ursprung in Japan werden innerhalb Japan bearbeitet.
- Wenn eine Inferenzanfrage aus einem Gebiet stammt, das nicht aufgeführt ist, wird sie standardmäßig in den Vereinigte Staaten bearbeitet.
- DevOps Agent und Bedrock sind nicht von Kundenrichtlinien in Service Control Policies (SCPs) oder Control Tower betroffen, die Kundeninhalte auf bestimmte Regionen beschränken.
- Bedrock kann andere Regionen als die Ursprungsregion innerhalb Ihrer Region verwenden, um staatenlose Inferenzen durchzuführen, um Leistung und Verfügbarkeit zu optimieren

Identity and Access Management

Authentifizierungsmethoden

AWS DevOps Agent bietet zwei Authentifizierungsmethoden für die Anmeldung bei der AWS DevOps Agent Space-Web-App:

- AWS Identity Center-Integration — Die primäre Authentifizierungsmethode verwendet OAuth 2.0 mit sitzungsbasierter Authentifizierung mithilfe von Cookies. HTTP-only AWS Identity Center kann sich über Standard-OIDC- und SAML-Protokolle mit externen Identitätsanbietern verbinden, darunter Anbieter wie Okta, Ping Identity und Microsoft Entra ID. Diese Methode unterstützt

die Multi-Faktor-Authentifizierung über Ihren Identitätsanbieter. AWS Identity Center verwendet standardmäßig eine Sitzungsdauer von bis zu 12 Stunden und kann auf eine gewünschte Dauer konfiguriert werden.

- IAM-Authentifizierungslink — Eine alternative Methode ermöglicht den direkten Zugriff auf die Web-App von der AWS Management Console aus mithilfe von JWT-based Tokens, die aus einer bestehenden AWS Management Console-Sitzung stammen. Diese Option ist nützlich, um den AWS DevOps Agenten vor der Implementierung der vollständigen Identity Center-Integration zu testen und um Administratorzugriff zu erhalten, falls auf die AWS DevOps Agent-Web-App über die Identity Center-basierte Authentifizierung nicht mehr zugegriffen werden kann. Die Sitzungen sind auf 10 Minuten begrenzt.

IAM-Rollen

AWS DevOps Der Agent verwendet IAM-Rollen, um Zugriffsberechtigungen zu definieren:

- Primäre Kontorolle — Gewährt dem Agenten Zugriff auf Ressourcen in dem AWS Konto, in dem Sie den Agent Space erstellen, sowie Zugriff auf sekundäre Kontorollen.
- Sekundäre Kontorollen — Gewährt dem Agenten Zugriff auf Ressourcen in zusätzlichen AWS Konten, die mit dem Agent Space verbunden sind.
- Web-App-Rolle — Gewährt Benutzern Zugriff auf die Ermittlungsdaten und Ergebnisse des AWS DevOps Agenten in der Web-App.

Diese Rollen sollten nach dem Prinzip der geringsten Rechte konfiguriert werden, sodass nur die für Untersuchungen erforderlichen Leseberechtigungen gewährt werden.

Datenschutz

Datenverschlüsselung

AWS DevOps Der Agent verschlüsselt alle Kundendaten:

- Verschlüsselung im Ruhezustand — Alle Daten werden mit AWS verwalteten Schlüsseln verschlüsselt.
- Verschlüsselung bei der Übertragung — Alle abgerufenen Protokolle, Metriken, Wissenselemente, Ticket-Metadaten und andere Daten werden bei der Übertragung innerhalb des privaten Netzwerks des Agenten und in externe Netzwerke verschlüsselt.

Speicherung und Aufbewahrung von Daten

Daten werden in der Region gespeichert, in der Ihr Agent Space erstellt wurde. Die Verarbeitung von Inferenzen kann jedoch in Ihrer Region erfolgen, wie im Abschnitt Nutzung von Amazon Bedrock oben beschrieben.

Persönlich identifizierbare Informationen (PII)

AWS DevOps Der Agent filtert keine personenbezogenen Daten, wenn er Daten zusammenfasst, die im Rahmen von Untersuchungen, Empfehlungsauswertungen oder Chat-Antworten gesammelt wurden. Es wird empfohlen, PII-Daten zu redigieren, bevor sie in Observability-Logs gespeichert werden.

Agentenjournal und Auditprotokollierung

Agenten-Journal

Sowohl die Funktionen Incident Investigation als auch Prevention führen ausführliche Journale, die:

- Protokollieren Sie alle Argumentationsschritte und ergriffenen Maßnahmen
- Schaffen Sie vollständige Transparenz in den Entscheidungsprozessen der Agenten
- Kann von den Agenten nach der Aufzeichnung nicht geändert werden, wodurch Angriffe, wie z. B. die sofortige Injektion, durch das Verbergen wichtiger Aktionen minimiert werden
- Schließt alle Chat-Nachrichten von der Investigation-Seite ein

AWS CloudTrail Integration

Alle AWS DevOps Agenten-API-Aufrufe werden automatisch AWS CloudTrail innerhalb des AWS Hosting-Kontos erfasst. Anhand der von CloudTrail gesammelten Informationen können Sie Folgendes ermitteln:

- Die Anfrage, die an den Agenten gestellt wurde
- Die IP-Adresse, von der die Anforderung erfolgt ist
- Wer die Anforderung gestellt hat
- Wann sie gestellt wurde.

Sofortiger Injektionsschutz

Ein Prompt-Injection-Angriff liegt vor, wenn ein Angreifer böswillige Anweisungen in externe Daten wie eine Webseite oder ein Dokument einbettet, die später von einem generativen KI-System verarbeitet werden. AWS DevOps Der Agent nutzt im Rahmen seines normalen Betriebs nativ viele Datenquellen, darunter Protokolle, Ressourcen-Tags und andere Betriebsdaten. AWS DevOps Der Agent schützt mit den unten aufgeführten Sicherheitsvorkehrungen vor Prompt-Injection-Angriffen. Es ist jedoch wichtig, sicherzustellen, dass alle verbundenen Datenquellen und der Benutzerzugriff auf diese Datenquellen vertrauenswürdig sind. Weitere Informationen finden Sie im Abschnitt Modell der geteilten Verantwortung.

Sicherheitsvorkehrungen für eine schnelle Injektion:

- Eingeschränkte Schreibmöglichkeiten — Die Tools, die dem Agenten zur Verfügung stehen, sind nicht in der Lage, Ressourcen zu verändern, mit Ausnahme von Tickets und Supportanfragen. Dadurch wird verhindert, dass böswillige Anweisungen Ihre Infrastruktur oder Anwendungen verändern.
- Durchsetzung von Kontogrenzen — Der AWS DevOps Agent arbeitet nur innerhalb der Grenzen, die durch die Rollen, die dem Agenten in den primären und verbundenen sekundären AWS Konten zugewiesen wurden, zulässig sind. Der Agent kann nicht auf Ressourcen außerhalb seines konfigurierten Bereichs zugreifen oder diese ändern.
- KI-Sicherheitsschutz — AWS DevOps Der Agent verwendet Modelle mit KI-Sicherheitsstufe 3 (ASL-3). Zu diesen Schutzmaßnahmen gehören Klassifikatoren, die Prompt-Injection-Angriffe erkennen und verhindern, bevor sie das Verhalten der Agenten beeinflussen können.
- Unveränderlicher Prüfpfad — Das Agenten-Journal protokolliert alle Argumentationsschritte und ergriffenen Maßnahmen. Journaleinträge können vom Agenten nicht mehr geändert werden, sobald sie einmal aufgezeichnet wurden, wodurch verhindert wird, dass Prompt-Injection-Angriffe böswillige Aktionen verbergen.

AWS DevOps Der Agent bietet zwar mehrere Schutzebenen vor Prompt-Injection-Angriffen, aber bestimmte Konfigurationen können das Risiko erhöhen:

- Benutzerdefinierte MCP-Servertools — Die Bring-Your-Own-MCP-Funktion ermöglicht es Ihnen, dem Agenten benutzerdefinierte Tools vorzustellen, was zusätzliche Möglichkeiten für eine sofortige Injektion bieten kann. Benutzerdefinierte Tools verfügen möglicherweise nicht über dieselben Sicherheitskontrollen wie native AWS DevOps Agent-Tools, und böswillige Anweisungen

könnten diese Tools möglicherweise auf unbeabsichtigte Weise nutzen. Weitere Informationen finden Sie im Abschnitt Modell der geteilten Verantwortung.

- Angriffe durch autorisierte Benutzer — Benutzer, die autorisiert sind, innerhalb der AWS Kontogrenzen oder verbundener Tools zu agieren, haben ein höheres Risiko, einen Angriff auf den Agenten zu versuchen. Diese Benutzer haben möglicherweise die Möglichkeit, Datenquellen zu ändern, die der Agent verwendet, z. B. Protokolle oder Ressourcen-Tags, wodurch es einfacher wird, bösartige Anweisungen einzubetten, die der Agent verarbeitet.

Um diese Risiken zu minimieren, gehen Sie wie folgt vor:

1. Prüfen und testen Sie die benutzerdefinierten MCP-Server sorgfältig, bevor Sie sie in Agent Spaces einsetzen.
   a. Stellen Sie sicher, dass sie nur schreibgeschützte Aktionen ausführen dürfen
   b. Stellen Sie sicher, dass es sich bei Benutzern externer Tools, auf die von MCP-Servern zugegriffen wird, um vertrauenswürdige Entitäten handelt, da AWS DevOps Agenten, die eine Schnittstelle zu MCP herstellen, auf der impliziten Vertrauensbeziehung zwischen diesen Tool-Benutzern und dem Agenten basieren AWS DevOps
2. Wenden Sie das Prinzip der geringsten Rechte an, wenn Sie Benutzern Zugriff auf Systeme gewähren, die dem Agenten Daten zur Verfügung stellen
3. Prüfen Sie regelmäßig, welche MCP-Server mit Ihren Agent Spaces verbunden sind
4. Da jeder Inhalt, der von URLs auf der Zulassungsliste abgerufen wird, versuchen könnte, das Verhalten des Agenten zu manipulieren, sollten Sie nur vertrauenswürdige Quellen in Ihre Zulassungsliste aufnehmen.

Sicherheit bei der Integration

AWS DevOps Der Agent unterstützt mehrere Integrationstypen mit jeweils eigenem Sicherheitsmodell:

- Native bidirektionale Integrationen — Built-in Integrationen, die Daten an den Agenten senden und Updates vom Agenten empfangen können. Dabei werden die Authentifizierungsmethoden des Anbieters verwendet
- MCP-Server — Remote Model Context Protocol-Server, die OAuth 2.0-Authentifizierungsabläufe und API-Schlüssel verwenden, um sicher mit externen Systemen zu kommunizieren.

- Webhook-Trigger — Ermittlungsauslöser, die von Remotediensten wie Tickets oder Observability-Systemen ausgelöst werden. Webhooks verwenden aus Sicherheitsgründen den Hash-based Message Authentication Code (HMAC).
- Ausgehende Kommunikation — Integrationen wie Slack und Ticketsysteme erhalten Updates vom Agenten, unterstützen aber noch keine bidirektionale Kommunikation.

Anbieter von Registrierungen

Einige externe Tools werden auf Kontoebene authentifiziert und von allen Agent Spaces im Konto gemeinsam genutzt. Wenn Sie diese Tools registrieren, authentifizieren Sie sich einmal auf Kontoebene, und dann kann jeder Agent Space innerhalb dieser registrierten Verbindung eine Verbindung zu bestimmten Ressourcen herstellen.

Die folgenden Tools verwenden die Registrierung auf Kontoebene:

- GitHub— Verwendet den OAuth-Flow für die Authentifizierung. Nach der Registrierung GitHub auf Kontoebene kann jeder Agent Space eine Verbindung zu bestimmten Repositorys innerhalb Ihrer Organisation herstellen. GitHub
- Dynatrace — Verwendet die OAuth-Token-Authentifizierung. Nach der Registrierung von Dynatrace auf Kontoebene kann sich jeder Agent Space mit bestimmten Dynatrace-Umgebungen oder Überwachungskonfigurationen verbinden.
- Slack — Verwendet die OAuth-Token-Authentifizierung. Nach der Registrierung von Slack auf Kontoebene kann sich jeder Agent Space mit bestimmten Slack-Channel-Kanälen verbinden.
- Datadog — Verwendet MCP mit OAuth-Flow zur Authentifizierung. Nach der Registrierung von Datadog auf Kontoebene kann jeder Agent Space eine Verbindung zu bestimmten Datadog-Überwachungsressourcen herstellen.
- New Relic — Verwendet die API-Schlüsselauthentifizierung. Nach der Registrierung von New Relic auf Kontoebene kann jeder Agent Space eine Verbindung zu bestimmten New Relic-Überwachungskonfigurationen herstellen.
- Splunk — Verwendet die Bearer-Token-Authentifizierung. Nach der Registrierung von Splunk auf Kontoebene kann jeder Agent Space eine Verbindung zu bestimmten Splunk-Datenquellen herstellen.
- GitLab— Verwendet die Zugriffstoken-Authentifizierung. Nach der Registrierung GitLab auf Kontoebene kann jeder Agent Space eine Verbindung zu bestimmten GitLab Repositorys herstellen.

- ServiceNow— Verwendet die OAuth-Client-Authentifizierung. key/token Nach der Registrierung ServiceNow auf Kontoebene kann sich jeder Agent Space mit bestimmten ServiceNow Instanzen oder Ticketwarteschlangen verbinden.
- Allgemein zugängliche Remote-MCP-Server — Verwenden Sie den OAuth-Flow für die Authentifizierung. Nach der Registrierung eines Remote-MCP-Servers auf Kontoebene kann jeder Agent Space eine Verbindung zu bestimmten Ressourcen herstellen, die von diesem Server verfügbar gemacht werden.

Netzwerkkonnektivität

AWS DevOps Der Agent stellt eine Verbindung zu Ihren Drittanbietersystemen und Remote-MCP-Servern her, um Untersuchungen und andere Operationen durchzuführen.

Eingehender Verkehr von AWS DevOps Agent für Ihre Systeme

AWS DevOps Der Agent initiiert ausgehende Verbindungen zu Ihren Drittanbietersystemen und Remote-MCP-Servern, die als eingehender Datenverkehr in Ihre Infrastruktur gelangen. Wie Sie diesen Datenverkehr schützen, hängt davon ab, wie Ihre Tools gehostet werden:

- Privat gehostete Tools — Wenn Ihre Tools von einer AWS VPC aus erreichbar sind, können Sie private AWS DevOps Agentenverbindungen verwenden, um den Datenverkehr zu AWS Netzwerken zu isolieren und vom öffentlichen Internet fernzuhalten. Weitere Informationen finden Sie unter the section called “Verbindung zu privat gehosteten Tools herstellen”.
- Öffentlich gehostete Tools — Wenn Ihre Tools über das öffentliche Internet erreichbar sind und IP-Allowlisting- oder Firewallregeln verwenden, müssen Sie eingehenden Datenverkehr von den folgenden AWS DevOps Agent-Quell-IP-Adressen zulassen:
  - Asien-Pazifik (Sydney): (ap-southeast-2)
    - `13.237.95.197`
    - `13.238.84.102`
    - `52.64.174.242`
    - `13.211.249.13`
    - `15.134.235.54`
    - `3.107.145.226`
  - Asien-Pazifik (Tokyo) (ap-northeast-1)
    - `13.192.12.233`

- 35.74.181.230
- 57.183.50.158
- 13.114.228.89
- 54.150.140.28
- 46.51.224.121

- Europa (Frankfurt) (eu-central-1)
  - 18.158.110.140
  - 52.57.96.160
  - 52.59.55.56
  - 63.183.67.111
  - 63.184.95.132
  - 63.184.36.38
- Europa (Irland) (eu-west-1)
  - 34.251.85.24
  - 52.30.157.157
  - 52.51.192.222
  - 99.81.41.52
  - 54.246.170.103
  - 52.212.224.65
- USA Ost (Nord-Virginia): (us-east-1)
  - 34.228.181.128
  - 44.219.176.187
  - 54.226.244.221
  - 100.56.22.59
  - 3.234.39.4
  - 44.215.92.10
- USA West (Oregon): (us-west-2)
  - 34.212.16.133
  - 52.89.67.212
  - 54.187.135.61

- 34.209.115.89
- 44.224.219.86
- 54.201.89.243

Ausgehender Verkehr von Ihrer VPC zu AWS DevOps Agent

Für ausgehenden Datenverkehr von Ihrer AWS VPC zum AWS DevOps Agenten (z. B. überthe section called “DevOps Agent über Webhook aufrufen”) können Sie VPC-Endpunkte verwenden, um diesen Netzwerkverkehr von Netzwerken zu isolieren. AWS Weitere Informationen finden Sie unter the section called “VPC-Endpunkte (AWS PrivateLink)”.

Modell der geteilten Verantwortung

AWS Verantwortlichkeiten

AWS ist verantwortlich für:

- Aufrechterhaltung der Sicherheit der vom Agenten abgerufenen Daten
- Sicherung systemeigener Tools, die vom Agenten verwendet werden können
- Schutz der Infrastruktur, auf der der AWS DevOps Agent ausgeführt wird

Pflichten des Kunden

Kunden sind verantwortlich für:

- Verwaltung des Benutzerzugriffs auf den Agentenbereich
- Beschränkung des Zugriffs auf vertrauenswürdige Benutzer externer Systeme, die Eingaben für den Agenten bereitstellen, z. B. Dienste und Ressourcen, die Protokolle, CloudTrail Ereignisse, Tickets und mehr erstellen, die für böswillige Prompt-Injection-Versuche verwendet werden können.
- Stellen Sie sicher, dass alle verbundenen Datenquellen über vertrauenswürdige Daten verfügen, die wahrscheinlich nicht für Prompt-Injection-Angriffe verwendet werden
- Stellen Sie sicher, dass die MCP-Serverintegrationen von Bring-Your-Own sicher funktionieren
- Stellen Sie sicher, dass die dem Agenten zugewiesenen IAM-Rollen den richtigen Umfang haben

- Bearbeitung personenbezogener Daten vor der Speicherung in Observability-Logs und anderen Agentendatenquellen
- Wir halten uns an die empfohlene Vorgehensweise, verbundenen Datenquellen, einschließlich MCP-Servern, bei denen Sie Ihre eigenen Daten verwenden, nur Leseberechtigungen zu gewähren

Datennutzung

AWS verwendet keine Agentendaten, Chat-Nachrichten oder Daten aus integrierten Datenquellen, um Modelle zu trainieren oder das Produkt zu verbessern. Der AWS DevOps Agent Space verwendet das Feedback von Kunden innerhalb des Produkts, um die Antworten und Untersuchungen des Kundendienstmitarbeiters zu verbessern, verwendet es jedoch AWS nicht, um den Service selbst zu verbessern.

DevOps IAM-Berechtigungen für Agenten

AWS DevOps Der Agent verwendet dienstspezifische AWS Identity and Access Management (IAM) -Aktionen, um den Zugriff auf seine Funktionen und Fähigkeiten zu kontrollieren. Diese Aktionen bestimmen, was Benutzer in der AWS DevOps Agentenkonsole und der Operator Web App tun können. Dies ist unabhängig von den AWS Service-API-Berechtigungen, die der Agent selbst verwendet, um Ihre Ressourcen zu untersuchen.

Weitere Informationen zur Beschränkung des Agentenzugriffs finden Sie unter Beschränken des Agentenzugriffs in einem AWS Konto.

Aktionen zur Verwaltung des Agentenbereichs

Diese Aktionen steuern den Zugriff auf die Konfiguration und Verwaltung von Agent Space:

- aidevops: GetAgentSpace — Ermöglicht Benutzern das Anzeigen von Details zu einem Agent Space, einschließlich seiner Konfiguration, seines Status und der zugehörigen Konten. Benutzer benötigen diese Berechtigung, um auf einen Agent Space in der AWS Management Console zuzugreifen.
- aidevops: GetAssociation — Ermöglicht Benutzern das Anzeigen von Details zu einer bestimmten Kontoverknüpfung, einschließlich der IAM-Rollenkonfiguration und des Verbindungsstatus.
- aidevops: ListAssociations — Ermöglicht Benutzern, alle für einen Agent Space konfigurierten AWS Kontozuordnungen aufzulisten, einschließlich primärer und sekundärer Konten.

Untersuchungs- und Ausführungsmaßnahmen

Mit diesen Aktionen wird der Zugriff auf Funktionen zur Untersuchung von Vorfällen gesteuert:

- aidevops: ListExecutions — Ermöglicht es Benutzern, Ausführungsmetadaten — einschließlich ID, Status und mehr — für Untersuchungen, Abhilfemaßnahmen, Bewertungen und Chat-Konversationen im Zusammenhang mit einer Aufgabe einzusehen.
- aidevops: ListJournalRecords — Ermöglicht Benutzern den Zugriff auf detaillierte Protokolle, in denen die Argumentation des Agenten, die ergriffenen Maßnahmen und die während einer Untersuchung, Abhilfemaßnahme, Bewertung und Chat-Konversation konsultierten Datenquellen aufgeführt sind. Dies ist nützlich, um zu verstehen, wie der Agent zu seinen Schlussfolgerungen gelangt ist.

Aktionen zur Chat-Verwaltung

Chat benötigt die folgenden IAM-Berechtigungen, um zu funktionieren:

- aidevops: ListChats — Ermöglicht Benutzern, den Chat-Konversationsverlauf aufzulisten und darauf zuzugreifen.
- aidevops: CreateChat — Ermöglicht Benutzern, neue Chat-Konversationen zu erstellen.
- aidevops: SendMessage — Ermöglicht Benutzern, Anfragen einzureichen und Streaming-Antworten zu erhalten.

Topologie und Erkennungsaktionen

Diese Aktionen steuern den Zugriff auf Funktionen zur Zuordnung von Anwendungsressourcen:

- aidevops: DiscoverTopology — Ermöglicht Benutzern, die Topologieerkennung und -zuordnung für einen Agent Space auszulösen. Diese Aktion initiiert den Prozess des Scannens von AWS Konten und des Aufbaus der Anwendungsressourcentopologie.

Präventions- und Empfehlungsmaßnahmen

Diese Aktionen steuern den Zugriff auf die Präventionsfunktion:

- aidevops: ListGoals — Ermöglicht es Benutzern, sich anhand der aktuellen Vorfallmuster die Präventionsziele und -ziele anzusehen, auf die der Agent hinarbeitet.

- aidevops: ListRecommendations — Ermöglicht Benutzern, alle Empfehlungen einzusehen, die durch die Präventionsfunktion generiert wurden, einschließlich ihrer Priorität und Kategorie.
- aidevops: GetRecommendation — Ermöglicht Benutzern, detaillierte Informationen zu einer bestimmten Empfehlung einzusehen, einschließlich der Vorfälle, die dadurch verhindert worden wären, und Anleitungen zur Umsetzung.

Aktionen zur Verwaltung von Backlog-Aufgaben

Diese Aktionen steuern die Fähigkeit, Empfehlungen als Backlog-Aufgaben zu verwalten:

- aidevops: CreateBacklogTask — Ermöglicht Benutzern, eine Aufgabe zur Untersuchung von Vorfällen oder zur Bewertung von Präventionsmaßnahmen zu erstellen.
- aidevops: UpdateBacklogTask — Ermöglicht Benutzern, einen Plan zur Schadensbegrenzung zu genehmigen oder eine aktive Untersuchung oder Bewertung abzubrechen.
- aidevops: GetBacklogTask — Ermöglicht Benutzern das Abrufen von Details zu einer bestimmten Aufgabe.
- aidevops: ListBacklogTasks — Ermöglicht Benutzern, Aufgaben für einen Agent Space aufzulisten, gefiltert nach Aufgabentyp, Status, Priorität oder Erstellungszeit.

Maßnahmen zum Wissensmanagement

Diese Aktionen steuern die Fähigkeit, benutzerdefiniertes Wissen hinzuzufügen und zu verwalten, das der Agent bei Untersuchungen verwenden kann:

- aidevops: CreateKnowledgeItem — Ermöglicht Benutzern das Hinzufügen benutzerdefinierter Wissenselemente wie Fähigkeiten, Anleitungen zur Fehlerbehebung oder anwendungsspezifische Informationen, auf die der Agent zurückgreifen sollte.
- aidevops: ListKnowledgeItems — Ermöglicht Benutzern, alle für einen Agent Space konfigurierten Wissenselemente einzusehen.
- aidevops: GetKnowledgeItem — Ermöglicht Benutzern das Abrufen der Details eines bestimmten Wissenselements.
- aidevops: UpdateKnowledgeItem — Ermöglicht Benutzern, bestehende Wissenselemente zu ändern, um die Informationen auf dem neuesten Stand zu halten.
- aidevops: DeleteKnowledgeItem — Ermöglicht Benutzern, Wissenselemente zu entfernen, die nicht mehr relevant sind.

AWS Support von Integrationsmaßnahmen

Diese Aktionen steuern die Integration mit AWS Support-Fällen:

- aidevops: InitiateChatForCase — Ermöglicht Benutzern, direkt von einer Untersuchung aus eine Chat-Sitzung mit dem AWS Support zu starten, wobei automatisch der Kontext zum Vorfall bereitgestellt wird.
- aidevops: EndChatForCase — Ermöglicht Benutzern, eine aktive Chat-Sitzung mit AWS Support-Anfragen zu beenden.
- aidevops: DescribeSupportLevel — Ermöglicht es Benutzern, die AWS Support-Stufe für das Konto zu überprüfen, um die verfügbaren Support-Optionen zu ermitteln.

Nutzungs- und Überwachungsaktionen

Diese Aktionen steuern den Zugriff auf Nutzungsinformationen:

- aidevops: GetAccountUsage — Ermöglicht es Benutzern, das monatliche Kontingent des AWS DevOps Agenten für Untersuchungsstunden, Teststunden zur Prävention und Chat-Anfragen sowie die Nutzung des aktuellen Monats einzusehen.

Allgemeine Beispiele für IAM-Richtlinien

Administrator-Richtlinie

Diese Richtlinie gewährt vollen Zugriff auf alle AWS DevOps Agent-Funktionen:

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

Richtlinie für Betreiber

Diese Richtlinie gewährt Zugriff auf Ermittlungs- und Präventionsfunktionen ohne Verwaltungsaufwand:

```
{
  "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": "*"
    }
  ]
}
```

Read-only Richtlinie

Diese Richtlinie gewährt nur Lesezugriff auf Untersuchungen und Empfehlungen:

```
{
  "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": "*"
    }
  ]
}
```

Verwenden von dienstbezogenen Rollen für AWS DevOps Agent

AWS DevOps Der Agent verwendet dienstverknüpfte Rollen für AWS Identity and Access Management (IAM). Eine dienstverknüpfte Rolle ist ein einzigartiger Typ von IAM-Rolle, die direkt mit dem Agenten verknüpft ist. AWS DevOps Service-linked Rollen werden vom AWS DevOps Agenten vordefiniert und beinhalten alle Berechtigungen, die der Dienst benötigt, um andere AWS Dienste in Ihrem Namen aufzurufen.

Service-linked Rollenberechtigungen

Die serviceverknüpfte `AWSServiceRoleForAIDevOps`-Rolle vertraut dem `aidevops.amazonaws.com`-Service-Prinzipal im Rahmen der Übernahme der Rolle.

Die Rolle verwendet die verwaltete Richtlinie `AWSServiceRoleForAIDevOpsPolicy` mit den folgenden Berechtigungen:

- `cloudwatch:PutMetricData`— Veröffentlichen Sie Nutzungsmetriken im `AWS/AIDevOps` CloudWatch Namespace. Gültig durch eine `cloudwatch:namespace` Bedingung, dass nur der Namespace zugelassen wird. `AWS/AIDevOps`
- `vpc-lattice:CreateResourceGateway`— Erstellen Sie VPC Lattice Resource Gateways für private Verbindungen. Gültig durch eine `aws:RequestTag/AWSAIDevOpsManaged` Bedingung, sodass der Dienst nur Ressourcen-Gateways erstellen kann, die das Tag tragen. `AWSAIDevOpsManaged`
- `vpc-lattice:TagResource`— Markieren Sie VPC Lattice Resource Gateways. Fällt unter eine Bedingung. `aws:RequestTag/AWSAIDevOpsManaged`
- `vpc-lattice:DeleteResourceGateway`— Löscht VPC Lattice Resource Gateways. Gültig durch eine `aws:ResourceTag/AWSAIDevOpsManaged` Bedingung, sodass der Service nur Ressourcen-Gateways löschen kann, die er erstellt hat.
- `vpc-lattice:GetResourceGateway`— Ruft Informationen über VPC Lattice Resource Gateways ab. Unterliegt einer `aws:ResourceTag/AWSAIDevOpsManaged` Bedingung, sodass der Dienst nur die von ihm erstellten Ressourcen-Gateways lesen kann.
- `ec2:DescribeVpcs`,`ec2:DescribeSubnets`, `ec2:DescribeSecurityGroups` — Ruft Informationen über VPC-Netzwerkressourcen ab, die für die Konfiguration von Ressourcen-Gateways erforderlich sind. Diese schreibgeschützten Aktionen gelten für alle VPC-Ressourcen, da die EC2-API keine Berechtigungen auf Ressourcenebene für Describe-Aufrufe unterstützt.
- `iam:CreateServiceLinkedRole`— Erstellen Sie die serviceverknüpfte VPC Lattice-Rolle, die für den Betrieb des Ressourcen-Gateways erforderlich ist. Diese Berechtigung gilt nur für den `vpc-lattice.amazonaws.com` Dienstprinzipal und kann nicht zum Erstellen von dienstbezogenen Rollen für andere Dienste verwendet werden.

Erstellen der serviceverknüpfte -Rolle

Sie müssen die serviceverknüpfte Rolle `AWSServiceRoleForAIDevOps` nicht manuell erstellen. Wenn Sie den AWS DevOps Agenten verwenden, erstellt der Dienst die dienstverknüpfte Rolle für Sie.

Damit der Dienst die Rolle in Ihrem Namen erstellen kann, benötigen Sie die entsprechende `iam:CreateServiceLinkedRole` Genehmigung. Wir empfehlen, diese Berechtigung mit der `iam:AWSServiceName` Bedingung `aidevops.amazonaws.com` zu verknüpfen, dass der

Grundsatz der geringsten Rechte eingehalten wird. Weitere Informationen finden Sie unter Service-linked Rollenberechtigungen.

Bearbeiten der serviceverknüpften Rolle

Sie können die serviceverknüpfte Rolle `AWSServiceRoleForAIDevOps` nicht bearbeiten. Nachdem die Rolle erstellt wurde, können Sie den Namen der Rolle nicht mehr ändern, da verschiedene Entitäten möglicherweise namentlich auf die Rolle verweisen. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle.

Löschen der serviceverknüpften -Rolle

Wenn Sie den AWS DevOps Agenten nicht mehr verwenden müssen, empfehlen wir, die `AWSServiceRoleForAIDevOps` dienstverknüpfte Rolle zu löschen. Bevor Sie die Rolle löschen können, müssen Sie zunächst alle privaten Verbindungen entfernen, die in Ihrem Agent Space konfiguriert sind. Durch das Löschen der dienstverknüpften Rolle werden die mit dem Dienst markierten VPC Lattice-Ressourcen-Gateways, die zuvor vom Dienst erstellt wurden`AWSAIDevOpsManaged`, nicht automatisch entfernt. Sie sollten diese Ressourcen-Gateways manuell löschen, wenn sie nicht mehr benötigt werden. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle.

AWS Verwaltete Richtlinien für AWS DevOps Agent

AWS adressiert viele gängige Anwendungsfälle durch die Bereitstellung eigenständiger IAM-Richtlinien, die von erstellt und verwaltet AWS werden. Diese AWS verwalteten Richtlinien gewähren die erforderlichen Berechtigungen für allgemeine Anwendungsfälle, sodass Sie nicht erst untersuchen müssen, welche Berechtigungen benötigt werden. Weitere Informationen finden Sie unter AWS Verwaltete Richtlinien im _IAM-Benutzerhandbuch_.

Die folgenden AWS verwalteten Richtlinien, die Sie Benutzern in Ihrem Konto zuordnen können, gelten nur für Agent. AWS DevOps

AIDevOpsAgentReadOnlyAccess

Ermöglicht Lesezugriff auf Amazon DevOps Agent über die AWS Management Console

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

```
    "Sid": "AIDevOpsAgentReadOnlyAccess",
    "Effect": "Allow",
    "Action": [
     "aidevops:Get*",
     "aidevops:List*",
     "aidevops:SearchServiceAccessibleResource"
    ],
    "Resource": "*"
   }
  ]
}
```

AIDevOpsAgentFullAccess

Bietet vollen Zugriff auf Amazon DevOps Agent über die AWS Management Console

```
{
 "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:DescribeServices",
  "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

Ermöglicht den Zugriff auf die AWS DevOps Operator-Web-App für einen Agent Space.

```
{
 "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",
    "aidevops:DescribeServices"
```

```
    ],
    "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": "${aws:PrincipalAccount}"
     }
    }
   },
   {
    "Sid": "AllowSupportOperatorActions",
    "Effect": "Allow",
    "Action": [
     "support:DescribeCases",
     "support:DescribeServices",
     "support:InitiateChatForCase",
     "support:DescribeSupportLevel"
    ],
    "Resource": "*",
    "Condition": {
     "StringEquals": {
      "aws:ResourceAccount": "${aws:PrincipalAccount}"
     }
    }
   },
   {
    "Sid": "AllowSecretsManagerOperatorActions",
    "Effect": "Allow",
    "Action": [
     "secretsmanager:CreateSecret",
     "secretsmanager:ListSecrets"
    ],
```

```
   "Resource": "*",
   "Condition": {
    "StringEquals": {
     "aws:ResourceAccount": "${aws:PrincipalAccount}"
    }
   }
  }
 ]
}
```

AIDevOpsAgentAccessPolicy

Stellt die vom AWS DevOps Agenten benötigten Berechtigungen zur Durchführung von Untersuchungen und Analysen der AWS Kundenressourcen bereit.

```
{
    "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:Get*",
                "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:Describe*",
                "pi:Get*",
                "pi:List*",
                "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",
                "servicequotas:ListServiceQuotas",
                "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/*"
            ]
        }
    ]
}
```

Beschränken des Agentenzugriffs in einem AWS Account

AWS DevOps Der Agent verwendet IAM-Rollen, um AWS Ressourcen bei der Untersuchung von Vorfällen und präventiven Bewertungen zu ermitteln und zu beschreiben. Sie können die

Zugriffsebene des Agenten steuern, indem Sie IAM-Richtlinien konfigurieren, die diesen Rollen zugeordnet sind. Die Anwendungstopologie zeigt nicht alles, worauf der Agent Zugriff hat. IAM-Richtlinien sind die einzige Möglichkeit, wirklich einzuschränken, auf welche AWS Service-APIs und Ressourcen der Agent zugreifen kann.

Grundlegendes zu den IAM-Rollen für AWS DevOps Agent

AWS DevOps Der Agent verwendet IAM-Rollen, um auf Ressourcen in zwei Arten von Konten zuzugreifen:

- Primäre Kontorolle — Gewährt dem Agenten Zugriff auf Ressourcen in dem AWS Konto, in dem Sie den Agent Space erstellen.
- Sekundäre Kontorollen — Gewährt dem Agenten Zugriff auf Ressourcen in zusätzlichen AWS Konten, die Sie mit dem Agent Space verbinden.

Für beide Kontotypen können Sie einschränken, auf welche AWS Dienste der Agent zugreifen kann, den Zugriff auf bestimmte Ressourcen innerhalb dieser Dienste einschränken und steuern, in welchen Regionen der Agent tätig sein kann.

Grundlegendes zu Genehmigungsrichtlinien

AWS DevOps Der Agent wendet auf jede Sitzung, die er beim Zugriff auf Ihre Ressourcen erstellt, einen Berechtigungsschutz an. AWS Diese Leitplanke dient als Obergrenze — sie definiert die maximale Anzahl an Berechtigungen, die der Agent jemals verwenden kann, unabhängig davon, welche Berechtigungen Sie für die IAM-Rolle gewähren.

Funktionsweise

Wenn der Agent Ihre IAM-Rolle übernimmt, übergibt er eine Sitzungsrichtlinie, die die effektiven Berechtigungen für diese Sitzung einschränkt. Die effektiven Berechtigungen sind der Schnittpunkt von:

1. Ihre IAM-Rollenrichtlinien — Die verwaltete Richtlinie und alle Inline-Richtlinien, die Sie der Rolle zuordnen.
2. Die Berechtigungs-Guardrail — Eine Sitzungsrichtlinie, die vom AWS DevOps Agenten bei der Übernahme der Rolle angewendet wird.

Eine Genehmigung muss auf beiden Ebenen vorhanden sein, um wirksam zu werden. Wenn Sie Ihrer Rolle eine Berechtigung hinzufügen, die nicht in der Guardrail enthalten ist, kann der Agent sie nicht verwenden.

Standardberechtigungen

Die `AIDevOpsAgentAccessPolicy` verwaltete Richtlinie stellt die standardmäßigen Leseberechtigungen bereit, die der Agent für Untersuchungen verwendet. Diese Berechtigungen sind in der Guardrail enthalten, sodass sie ohne zusätzliche Konfiguration funktionieren.

Erweiterung der Berechtigungen über die Standardberechtigungen hinaus

AWS DevOps Der Agent unterstützt einen kuratierten Satz zusätzlicher Berechtigungen, die über die standardmäßige verwaltete Richtlinie hinausgehen. Diese Berechtigungen sind in der Guardrail enthalten, aber standardmäßig nicht aktiviert. Um sie zu verwenden, fügen Sie Ihrer Rolle die spezifischen Berechtigungen als Inline-Richtlinie hinzu.

Um es dem Agenten beispielsweise zu ermöglichen, bei Untersuchungen Objekte aus Ihren S3-Buckets zu lesen, fügen Sie Ihrer Rolle eine Inline-Richtlinie hinzu:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::my-application-bucket",
        "arn:aws:s3:::my-application-bucket/*"
      ]
    }
  ]
}
```

Da sie `s3:GetObject` in der Leitplanke enthalten `s3:ListBucket` sind, ist diese Inline-Richtlinie wirksam. Sie können den Bereich `Resource` auf bestimmte Bereiche beschränken, um dem Prinzip der geringsten Rechte zu folgen.

Zusätzliche Berechtigungen werden unterstützt

Die folgenden Berechtigungen sind in der Guardrail enthalten und können aktiviert werden, indem Sie sie Ihrer Rolle als Inline-Richtlinie hinzufügen. Diese werden standardmäßig nicht gewährt — Sie müssen sich ausdrücklich dafür entscheiden.

Service	Aktionen	Anwendungsfall
Amazon S3	`s3:GetObject` , `s3:ListBucket`	Lesen Sie in S3 gespeicherte Anwendungsdaten, Protokolle oder Konfigurationen
AWS Direkte Connect	`directconnect:DescribeConnections` , `directconnect:DescribeDirectConnectGatewayAssociations` , `directconnect:DescribeDirectConnectGateways` , `directconnect:DescribeLags` , `directconnect:DescribeVirtualInterfaces`	Untersuchen Sie Probleme mit der Netzwerkverbindung

> Hinweis: Diese Liste kann im Laufe der Zeit erweitert werden, wenn dem AWS DevOps Agenten neue Funktionen hinzugefügt werden. Berechtigungen, die hier oder in der `AIDevOpsAgentAccessPolicy` verwalteten Richtlinie nicht aufgeführt sind, werden durch die Leitplanke blockiert.

Durch die Leitplanke blockierte Berechtigungen

Wenn Sie Ihrer Rolle eine Berechtigung hinzufügen, die nicht in der Guardrail enthalten ist, kann der Agent sie nicht verwenden. Das ist beabsichtigt — die Leitplanke verhindert, dass der Agent Aktionen außerhalb seines vorgesehenen Bereichs ausführt, selbst wenn die Rolle dies andernfalls zulassen würde.

Schreiboperationen wies3:PutObject, oder dynamodb:DeleteItem sind beispielsweise nicht in der Guardrail enthalten. ec2:TerminateInstances Selbst wenn Ihre Rolle diese Berechtigungen gewährt, kann der Agent diese Aktionen nicht ausführen.

Zusammenfassung

Ebene	Wer kontrolliert es	Zweck
IAM-Rollenrichtlinien	Sie	Definieren Sie, wozu der Agent in der Lage sein soll
Leitplanke für Genehmigungen	AWS DevOps Agentin	Definiert das Maximum, das der Agent jemals erreichen kann
Effektive Berechtigungen	Schnittmenge von beiden	Was der Agent tatsächlich tun kann

Dieses Modell stellt sicher, dass der Agent innerhalb einer klar definierten Sicherheitsgrenze arbeitet, und bietet Ihnen gleichzeitig die Flexibilität, seine Funktionen für Ihren speziellen Anwendungsfall zu erweitern.

Wählen Sie Ihre Ressourcengrenzen

Wenn Sie den Ressourcenzugriff einschränken, müssen Sie genügend Berechtigungen angeben, damit der Agent Anwendungsvorfälle erfolgreich untersuchen kann. Dies umfasst:

- Alle Ressourcen für in den Anwendungsbereich fallende Anwendungen, die der Agent überwachen und untersuchen sollte
- Die gesamte unterstützende Infrastruktur, von der diese Anwendungen abhängen

Die unterstützende Infrastruktur kann Folgendes umfassen:

- Netzwerkkomponenten (VPCs, Subnetze, Load Balancer, API-Gateways)
- Datenspeicher (Datenbanken, Caches, Objektspeicher)
- Rechenressourcen (EC2-Instances, Lambda-Funktionen, Container)
- Überwachungs- und Protokollierungsdienste (CloudWatch,) CloudTrail

- Ressourcen für die Identitäts- und Zugriffsverwaltung sind erforderlich, um die Berechtigungen zu verstehen

Wenn Sie den Zugriff zu eng einschränken, ist der Agent möglicherweise nicht in der Lage, die Hauptursachen zu identifizieren, die auf die unterstützende Infrastruktur außerhalb Ihrer definierten Grenzen zurückzuführen sind.

Einschränken des Servicezugriffs

Sie können einschränken, auf welche AWS Dienste der Agent zugreifen kann, indem Sie die IAM-Richtlinien ändern, die den Rollen des Agenten zugeordnet sind. Beachten Sie bei der Erstellung benutzerdefinierter Richtlinien die folgenden bewährten Methoden:

- Nur Leseberechtigungen gewähren — Der Agent muss bei Untersuchungen Ressourcenkonfigurationen, Metriken und Protokolle lesen. Vermeiden Sie es, dem Agenten Berechtigungen zu erteilen, die es ihm ermöglichen, Ressourcen zu ändern oder zu löschen.
- Beschränken Sie sich auf notwendige Dienste — Schließen Sie nur die AWS Dienste ein, die Ressourcen enthalten, die für Ihre Anwendungen relevant sind. Wenn Ihre Anwendung beispielsweise Amazon RDS nicht verwendet, nehmen Sie keine RDS-Berechtigungen in die Richtlinie auf.
- Verwenden Sie bestimmte Aktionen anstelle von Platzhaltern — Anstatt `service:*` Berechtigungen zu gewähren, geben Sie einzelne Aktionen wie `cloudwatch:GetMetricData` oder `ec2:DescribeInstances` an.

Beispiel für eine Richtlinie, die auf bestimmte Dienste beschränkt ist:

```
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": "*"
    }
  ]
}
```

Beschränkung des Ressourcenzugriffs

Um den Agenten auf bestimmte Ressourcen innerhalb eines Dienstes zu beschränken, verwenden Sie in Ihren IAM-Richtlinien Berechtigungen auf Ressourcenebene. Auf diese Weise können Sie nur Ressourcen Zugriff gewähren, die bestimmten Mustern entsprechen.

Verwenden von Ressourcen-ARN-Mustern:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lambda:GetFunction",
        "lambda:GetFunctionConfiguration"
      ],
      "Resource": "arn:aws:lambda:*:*:function:production-*"
    }
  ]
}
```

In diesem Beispiel wird der Agent darauf beschränkt, nur auf Lambda-Funktionen zuzugreifen, deren Namen mit „production-“ beginnen.

Verwendung von tagbasierten Einschränkungen:

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

```
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeInstanceStatus"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Environment": "production"
        }
      }
    }
  ]
}
```

In diesem Beispiel wird der Agent darauf beschränkt, nur auf EC2-Instances zuzugreifen, die mit gekennzeichnet sind. `Environment=production`

Beschränkung des regionalen Zugriffs

Um einzuschränken, auf welche AWS Regionen der Agent zugreifen kann, verwenden Sie den `aws:RequestedRegion` Bedingungsschlüssel in Ihren IAM-Richtlinien:

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

```
}
```

In diesem Beispiel wird der Agent darauf beschränkt, nur auf Ressourcen in den Regionen us-east-1 und us-west-2 zuzugreifen.

Benutzerdefinierte IAM-Richtlinien erstellen

Wenn Sie einen Agent Space erstellen oder sekundäre Konten hinzufügen, haben Sie die Möglichkeit, mithilfe einer Richtlinienvorlage eine benutzerdefinierte IAM-Rolle zu erstellen. Auf diese Weise können Sie das Prinzip der geringsten Rechte implementieren.

Beim Erstellen eines Agent Space

Von der DevOps Agent-Konsole in der AWS Management-Konsole aus...

- Wählen Sie Neue DevOps Agentenrolle mithilfe eines Richtliniendokuments erstellen und folgen Sie den Anweisungen

Beim Bearbeiten eines Agentenbereichs

Von der DevOps Agent-Konsole in der AWS Management-Konsole aus...

- Wählen Sie die Registerkarte Funktionen
- Wählen Sie im Cloud-Bereich das sekundäre Konto aus, das Sie bearbeiten möchten, und klicken Sie auf Bearbeiten
- Wählen Sie Neue DevOps Agentenrichtlinie mithilfe einer Vorlage erstellen und folgen Sie den Anweisungen

Bewährte Methoden für benutzerdefinierte Richtlinien

- Nur Leseberechtigungen gewähren — Vermeiden Sie Berechtigungen, die das Ändern oder Löschen von Ressourcen ermöglichen
- Verwenden Sie nach Möglichkeit Berechtigungen auf Ressourcenebene — Beschränken Sie den Zugriff auf bestimmte Ressourcen mithilfe von ARN-Mustern oder -Tags
- Regelmäßige Überprüfung und Prüfung der Berechtigungen — Überprüfen Sie regelmäßig die IAM-Richtlinien des Agenten, um sicherzustellen, dass sie weiterhin Ihren Sicherheitsanforderungen entsprechen

Einrichtung der IAM Identity Center-Authentifizierung

Die IAM Identity Center-Authentifizierung bietet eine zentrale Möglichkeit, den Benutzerzugriff auf die AWS DevOps Agent Space-Webanwendung zu verwalten. In diesem Handbuch wird erklärt, wie die IAM Identity Center-Authentifizierung konfiguriert und Benutzer verwaltet werden.

Voraussetzungen

Bevor Sie die IAM Identity Center-Authentifizierung einrichten, stellen Sie sicher, dass Sie über Folgendes verfügen:

- IAM Identity Center ist in Ihrer Organisation oder Ihrem Konto aktiviert
- Administratorberechtigungen in Agent AWS DevOps
- Ein Agent-Space ist konfiguriert oder bereit zur Erstellung

Authentifizierungsoptionen

AWS DevOps Der Agent bietet zwei Authentifizierungsmethoden für den Zugriff auf die Agent Space-Web-App:

IAM Identity Center-Authentifizierung — Für Produktionsumgebungen empfohlen. Bietet zentrale Benutzerverwaltung, Integration mit externen Identitätsanbietern und Sitzungen von bis zu 12 Stunden.

Administratorzugriff (IAM-Authentifizierung) — Bietet Administratoren bei der Ersteinrichtung und Konfiguration schnellen Zugriff. Die Sitzungen sind auf 30 Minuten begrenzt.

Konfiguration von IAM Identity Center während der Erstellung des Agent Space

Wenn Sie einen Agent Space erstellen, können Sie die IAM Identity Center-Authentifizierung auf der Registerkarte Access konfigurieren:

Schritt 1: Navigieren Sie zur Konfiguration der Web-App

1. Nachdem Sie Ihre Agent Space-Details und den AWS Kontozugriff konfiguriert haben, wechseln Sie zur Registerkarte Zugriff

2. Sie sehen zwei Abschnitte: „Connect IAM Identity Center“ und „Admin-Zugriff“

Schritt 2: Konfigurieren Sie die IAM Identity Center-Integration

Gehen Sie im Abschnitt [Agent Space] mit IAM Identity Center verbinden wie folgt vor:

1. Überprüfen Sie die IAM Identity Center-Instanz — In der Konsole wird angezeigt, welche Identity Center-Instanz den Web-App-Benutzerzugriff verwaltet (z. B.`ssoins-7223a9580931edbe`). Ihre nächstgelegene IAM Identity Center-Instanz wird automatisch vorab aufgefüllt.
2. Wählen Sie die Option Rollenname der IAM Identity Center-Anwendung aus — Wählen Sie eine von drei Optionen:

Automatische Erstellung einer neuen DevOps Agentenrolle (empfohlen):

- Das System erstellt automatisch eine neue Servicerolle mit den entsprechenden Berechtigungen
- Dies ist die einfachste Option und funktioniert für die meisten Anwendungsfälle

Weisen Sie eine bestehende Rolle zu:

- Verwenden Sie eine bestehende IAM-Rolle, die Sie bereits erstellt haben
- Das System überprüft, ob die Rolle über die erforderlichen Berechtigungen verfügt
- Wählen Sie diese Option, wenn Ihre Organisation vorab erstellte Rollen für AWS DevOps Agenten hat

Erstellen Sie mithilfe einer Richtlinienvorlage eine neue DevOps Agentenrolle:

- Verwenden Sie die bereitgestellten Richtliniendetails, um Ihre eigene benutzerdefinierte Rolle in der IAM-Konsole zu erstellen
- Wählen Sie diese Option, wenn Sie die Rollenberechtigungen anpassen müssen

Nachdem Sie auf Connect geklickt haben, führt das System automatisch:

- Erstellt oder konfiguriert die angegebene IAM-Rolle
- Richtet eine IAM Identity Center-Anwendung für Ihren Agent Space ein
- Stellt Vertrauensbeziehungen zwischen IAM Identity Center und der Agent Space-Web-App her

- Konfiguriert OAuth 2.0-Authentifizierungsabläufe für sicheren Benutzerzugriff

Alternative: Verwenden des Administratorzugriffs

Wenn Sie sofort auf die Agent Space-Web-App zugreifen möchten, ohne IAM Identity Center einzurichten:

1. Notieren Sie sich im Abschnitt Admin-Zugriff den ARN für die IAM-Rolle, der Administratorzugriff bietet (z. B.`arn:aws:iam::440491339484:role/service-role/DevOpsAgentRole-WebappAdmin-15ppoc42`)
2. Klicken Sie auf die blaue Schaltfläche für Administratorzugriff, um die Agent Space-Web-App mit IAM-Authentifizierung zu starten
3. Sitzungen, die diese Methode verwenden, sind auf 30 Minuten begrenzt

> **Note**
>
> Der Administratorzugriff ist für die Ersteinrichtung und Konfiguration vorgesehen. Für den produktiven Einsatz und den laufenden Betrieb konfigurieren Sie die IAM Identity Center-Authentifizierung.

Hinzufügen von Benutzern und Gruppen

Nach der Konfiguration der IAM Identity Center-Authentifizierung müssen Sie bestimmten Benutzern und Gruppen Zugriff auf die Agent Space-Web-App gewähren:

Schritt 1: Greifen Sie auf die Benutzerverwaltung zu

1. Wählen Sie in der AWS DevOps Agent-Konsole Ihren Agent Space aus
2. Gehen Sie zur Registerkarte Zugriff
3. Klicken Sie unter Benutzerzugriff auf Benutzer und Gruppen verwalten

Schritt 2: Fügen Sie Benutzer oder Gruppen hinzu

1. Wählen Sie Benutzer oder Gruppen hinzufügen
2. Suchen Sie in Ihrem IAM Identity Center-Verzeichnis nach Benutzern oder Gruppen

3. Wählen Sie die Kontrollkästchen neben den Benutzern oder Gruppen aus, die Sie hinzufügen möchten
4. Klicken Sie auf Hinzufügen, um ihnen Zugriff zu gewähren

Die ausgewählten Benutzer können jetzt mit ihren IAM Identity Center-Anmeldeinformationen auf die Agent Space-Web-App zugreifen.

Mit externen Identitätsanbietern arbeiten

Wenn Sie einen externen Identitätsanbieter (wie Okta, Microsoft Entra ID oder Ping Identity) mit IAM Identity Center verwenden:

- Benutzer und Gruppen werden von Ihrem externen Identitätsanbieter mit IAM Identity Center synchronisiert
- Wenn Sie der Agent Space-Web-App Benutzer und Gruppen hinzufügen, wählen Sie aus dem synchronisierten Verzeichnis
- Benutzerattribute und Gruppenmitgliedschaften werden von Ihrem externen Identitätsanbieter verwaltet
- Änderungen an Ihrem Identitätsanbieter werden nach der Synchronisation automatisch in IAM Identity Center übernommen

So greifen Benutzer auf die Agent Space-Web-App zu

Nachdem Sie Benutzer zu Ihrem Agent Space hinzugefügt haben:

1. Teilen Sie die URL der Agent Space-Web-App mit autorisierten Benutzern
2. Wenn Benutzer zu der URL navigieren, werden sie zur IAM Identity Center-Anmeldeseite weitergeleitet
3. Nach Eingabe ihrer Anmeldeinformationen (und Abschluss der MFA, falls konfiguriert), werden sie zurück zur Agent Space-Web-App umgeleitet
4. Ihre Sitzung ist standardmäßig für 8 Stunden gültig (vom Identity Center-Administrator konfigurierbar)

Verwalten des Benutzerzugriffs

Sie können den Benutzerzugriff jederzeit aktualisieren:

Weitere Benutzer oder Gruppen hinzufügen:

- Gehen Sie wie oben beschrieben vor, um weitere Benutzer oder Gruppen hinzuzufügen

Zugriff entfernen:

1. Suchen Sie im Abschnitt Benutzerzugriff nach dem Benutzer oder der Gruppe, den Sie entfernen möchten
2. Klicken Sie neben dem Namen auf die Schaltfläche Entfernen
3. Bestätigen Sie das Entfernen

Entfernte Benutzer verlieren sofort den Zugriff, aktive Sitzungen können jedoch fortgesetzt werden, bis sie ablaufen.

Verwaltung von Sitzungen

IAM Identity Center-Sitzungen für die Agent Space-Web-App weisen die folgenden Merkmale auf:

- Standard-Sitzungsdauer — 8 Stunden
- Sitzungssicherheit — Nur HTTP-Cookies für verbesserten Schutz
- Multi-Faktor-Authentifizierung — Wird unterstützt, wenn sie in IAM Identity Center konfiguriert ist
- API-Anmeldeinformationen — SigV4-Anmeldeinformationen mit kurzer Dauer (15 Minuten) werden für API-Aufrufe ausgestellt und automatisch erneuert

So konfigurieren Sie die Sitzungsdauer:

1. Navigieren Sie zur IAM Identity Center-Konsole
2. Gehen Sie zu Einstellungen > Authentifizierung
3. Konfigurieren Sie unter Sitzungsdauer Ihre bevorzugte Dauer (von 1 Stunde bis 12 Stunden)
4. Wählen Sie Save Changes (Änderungen speichern)

Identity Center wird getrennt

1. Klicken Sie in der Konsole Ihres Agent Space oben rechts auf Aktionen und wählen Sie Verbindung zum IAM Identity Center trennen

2. Bestätigen Sie im Bestätigungsdialogfeld

Authentifizierung durch externen Identitätsanbieter (IdP) einrichten

Die Authentifizierung durch einen externen Identitätsanbieter (IdP) ermöglicht es Ihrer Organisation, einen vorhandenen OIDC-kompatiblen Identitätsanbieter wie Okta oder Microsoft Entra ID zu verwenden, um den Benutzerzugriff auf die Agent Space-Webanwendung zu verwalten. AWS DevOps Benutzer melden sich mit ihren Unternehmensanmeldedaten direkt über Ihren IdP an, ohne dass AWS IAM Identity Center erforderlich ist.

Voraussetzungen

Bevor Sie die externe IdP-Authentifizierung einrichten, stellen Sie sicher, dass Sie über Folgendes verfügen:

- Ein OIDC-kompatibler Identitätsanbieter (Okta oder Microsoft Entra ID)
- Administratorzugriff auf Ihren Identitätsanbieter
- Administratorberechtigungen für den Zugriff auf die AWS DevOps Agentenkonsole
- Ein Agent-Space ist konfiguriert oder bereit zur Erstellung

Funktionsweise

Wenn Sie die externe IdP-Authentifizierung konfigurieren:

- Benutzer navigieren zur URL der Agent Space-Web-App
- Sie werden auf die Anmeldeseite Ihres Identitätsanbieters weitergeleitet
- Nachdem sie sich mit ihren Unternehmensanmeldedaten authentifiziert haben, werden sie zurück zur Web-App weitergeleitet
- Die Web-App tauscht das Authentifizierungstoken gegen kurzlebige AWS Anmeldeinformationen aus, die auf den Agent Space beschränkt sind

Sitzungen sind bis zu 8 Stunden gültig. Anmeldeinformationen werden automatisch mithilfe von OIDC-Aktualisierungstoken aktualisiert, ohne dass sich Benutzer erneut authentifizieren müssen.

Konfiguration der externen IdP-Authentifizierung

Schritt 1: Registrieren Sie eine Anwendung bei Ihrem Identitätsanbieter

Wählen Sie Ihren Identitätsanbieter und folgen Sie den entsprechenden Einrichtungsanweisungen.

Option A: Okta

1. Navigieren Sie in der Okta Admin Console zu Anwendungen > Anwendungen und wählen Sie Create App Integration
2. Wählen Sie OIDC — OpenID Connect als Anmeldemethode und Webanwendung als Anwendungstyp. Wählen Sie Weiter
3. Geben Sie einen aussagekräftigen Namen für die Anwendung ein (z. B.) `AWS DevOps Agent`
4. Stellen Sie sicher, dass unter Grant-Typ die folgenden Optionen aktiviert sind:
   - Autorisierungscode (Standard)
   - Token aktualisieren — Dies ist für die Sitzungsaktualisierung erforderlich. Wenn diese Option nicht aktiviert ist, können Benutzer keine Sitzungen aufrechterhalten.

> **Note**
>
> Okta aktiviert den Gewährungstyp „Aktualisierungstoken“ standardmäßig nicht. Sie müssen ihn explizit aktivieren.

1. Behalten Sie die Anmeldeumleitung vorerst URIs als Standardwert bei. Sie werden sie aktualisieren, nachdem Sie den Agent Space konfiguriert haben
2. Weisen Sie unter Zuweisungen die Benutzer oder Gruppen zu, die Zugriff haben sollen
3. Wählen Sie Speichern aus.
4. Notieren Sie sich auf der Registerkarte Allgemein der Anwendung die folgenden Werte:
   - Kunden-ID
   - Geheimer Client-Schlüssel — Wählen Sie Kopieren, um diesen Wert sicher zu speichern
5. Notieren Sie sich Ihre Okta-Domain — das ist Ihre Aussteller-URL (z. B.`https://dev-12345678.okta.com`).

> **Note**
>
> Stellen Sie auf der Registerkarte Anmelden sicher, dass der Aussteller auf Okta-URL (nicht dynamisch) eingestellt ist. Dadurch wird eine stabile Aussteller-URL gewährleistet.

> **Note**
>
> Fügen Sie dem ID-Token auf der Registerkarte Ansprüche Ihres Autorisierungsservers keinen Gruppenanspruch hinzu. AWS DevOps Der Agent verwendet keine Gruppenmitgliedschaft von Ihrem IdP.

Option B: Microsoft Entra ID

1. Navigieren Sie im Azure-Portal zu Microsoft Entra ID > App-Registrierungen > Neue Registrierung
2. Geben Sie einen aussagekräftigen Namen ein (z. B.) `AWS DevOps Agent`
3. Wählen Sie unter Unterstützte Kontotypen die für Ihre Organisation geeignete Option aus (normalerweise nur Konten in diesem Organisationsverzeichnis)
4. Lassen Sie die Umleitungs-URI vorerst leer. Wählen Sie Registrieren
5. Notieren Sie sich auf der Seite mit der Anwendungsübersicht die folgenden Werte:
   - Anwendungs-ID (Client) — wird bei der Konfiguration des Agent Space als Client-ID verwendet
   - Verzeichnis-ID (Mandanten-ID) — wird zur Erstellung der Aussteller-URL verwendet
6. Navigieren Sie zu Certificates & Secrets > New Client Secret
   - Lege eine Beschreibung und einen Ablaufzeitraum fest
   - Wählen Sie Hinzufügen und kopieren Sie den geheimen Wert sofort — er wird nicht erneut angezeigt
7. Die Aussteller-URL für Entra ID folgt diesem Format. `{tenant-id}`Ersetzen Sie sie durch Ihre Verzeichnis-ID (Mandanten-ID) aus Schritt 5:
   - `https://login.microsoftonline.com/{tenant-id}/v2.0`

> **Note**
>
> Aktivieren Sie in der Token-Konfiguration nicht den optionalen Anspruch „Gruppen“. AWS DevOps Der Agent verwendet keine Gruppenmitgliedschaft von Ihrem IdP.

Schritt 2: Aktivieren Sie die Operator App mit IdP-Authentifizierung

1. Wählen Sie in der AWS DevOps Agent-Konsole Ihren Agent Space aus
2. Gehen Sie zur Registerkarte Zugriff
3. Wählen Sie unter Benutzerzugriff die Option Externer Identitätsanbieter
4. Konfigurieren Sie im Konfigurationsformular Folgendes:
   - Identitätsanbieter — Wählen Sie Ihren Identitätsanbieter (Okta oder Microsoft Entra ID)
   - Aussteller-URL — Die OIDC-Aussteller-URL Ihres Identitätsanbieters
   - Client-ID — Die Client-ID aus der OIDC-Anwendung, die Sie erstellt haben
   - Geheimer Client-Schlüssel — Der geheime Client-Schlüssel aus Ihrer OIDC-Anwendung
5. Wählen Sie unter Rollenname der Identity Provider-Anwendung eine von drei Optionen aus:
   - Automatische Erstellung einer neuen DevOps Agentenrolle (empfohlen) — Erstellt eine neue Servicerolle mit den entsprechenden Berechtigungen
   - Eine bestehende Rolle zuweisen — Verwenden Sie eine bestehende IAM-Rolle, die Sie bereits erstellt haben
   - Eine neue DevOps Agentenrolle mithilfe einer Richtlinienvorlage erstellen — Verwenden Sie die bereitgestellten Details, um Ihre eigene Rolle in der IAM-Konsole zu erstellen
6. Lesen Sie die Warnmeldung zur Callback-URL, die am Ende des Formulars angezeigt wird. Kopieren Sie diese URL — Sie müssen sie der erlaubten Weiterleitung Ihres Identitätsanbieters hinzufügen, URIs bevor sich Benutzer anmelden können.
7. Wählen Sie Connect

Nachdem Sie Connect ausgewählt haben, zeigt die Konsole die Konfiguration des externen Identitätsanbieters mit den folgenden Details an:

- Anbieter — Der Identitätsanbieter, den Sie ausgewählt haben
- Aussteller-URL — Die konfigurierte OIDC-Aussteller-URL
- Client-ID — Die konfigurierte Client-ID

- IAM-Rolle ARN — Die für den Benutzerzugriff verwendete IAM-Rolle
- Callback-URL — Konfigurieren Sie diese URL in Ihrem Identity Provider als zulässige Umleitungs-URI
- Anmelde-URL — Verwenden Sie diese URL, um über Ihren Identitätsanbieter auf die Web-App zuzugreifen

Schritt 3: Fügen Sie die Rückruf-URL zu Ihrem Identitätsanbieter hinzu

Okta

1. Navigieren Sie in der Okta Admin Console zum Tab Allgemein Ihrer Anwendung
2. Wählen Sie unter Anmelden die Option Bearbeiten
3. Fügen Sie die Callback-URL als Anmelde-Umleitungs-URI hinzu:
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/callback`
4. (Optional) Legen Sie die Initiate login URI fest, um die IDP-initiierte Anmeldung über das Okta-Dashboard zu aktivieren:
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/login`
5. (Empfohlen) Fügen Sie eine Abmelde-Umleitungs-URI hinzu, um Benutzer nach dem Abmelden zurück zur Web-App umzuleiten. Andernfalls wird Benutzern beim Abmelden möglicherweise eine Fehlerseite angezeigt:
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome`
6. Wählen Sie Speichern aus.

Microsoft Entra ID

1. Navigieren Sie im Azure-Portal zur Authentifizierungsseite Ihrer Anwendung
2. Wählen Sie unter Plattformkonfigurationen die Option Plattform hinzufügen > Web
3. Geben Sie die Rückruf-URL als Umleitungs-URI ein:
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/callback`
4. (Optional) Fügen Sie eine Abmelde-Umleitungs-URI hinzu, um Benutzer nach dem Abmelden zurück zur Web-App umzuleiten:
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome`
5. Wählen Sie „Konfigurieren“

Schritt 4: Überprüfen Sie die Konfiguration

1. Navigieren Sie zu der in der Konsole angezeigten Anmelde-URL:
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/login`
2. Sie sollten zur Anmeldeseite Ihres Identitätsanbieters weitergeleitet werden
3. Melden Sie sich mit Ihren Unternehmensanmeldedaten an
4. Nach erfolgreicher Authentifizierung werden Sie zurück zur Agent Space-Web-App weitergeleitet

Aktualisierung der IdP-Konfiguration

Sie können den geheimen Client-Schlüssel rotieren, ohne die Verbindung zu trennen:

1. Wählen Sie in der AWS DevOps Agent-Konsole Ihren Agent-Bereich
2. Gehen Sie zur Registerkarte Zugriff
3. Wählen Sie unter Konfiguration des externen Identitätsanbieters die Option Rotate client secret
4. Geben Sie den neuen geheimen Clientschlüssel ein
5. Wählen Sie Speichern aus.

Um ein anderes IdP-Konfigurationsfeld (wie Aussteller-URL, Client-ID oder Identitätsanbieter) zu ändern, müssen Sie die Verbindung zum vorhandenen IdP trennen und einen neuen konfigurieren.

Wie Benutzer auf die Agent Space-Web-App zugreifen

Nach der Konfiguration der externen IdP-Authentifizierung:

- Teilen Sie die URL der Agent Space-Web-App mit autorisierten Benutzern
- Wenn Benutzer zu der URL navigieren, werden sie zur Anmeldeseite Ihres Identitätsanbieters weitergeleitet
- Nach Eingabe ihrer Anmeldeinformationen (und Abschluss der MFA, falls von Ihrem IdP konfiguriert), werden sie zurück zur Agent Space-Web-App umgeleitet
- Sitzungen werden automatisch aktualisiert — Einzelheiten finden Sie unter Sitzungsverwaltung

Sitzungsverwaltung

Externe IdP-Sitzungen für die Agent Space-Web-App haben die folgenden Eigenschaften:

- Sitzungsdauer — Browsersitzungen dauern bis zu 8 Stunden. Dies ist im AWS DevOps Agent nicht konfigurierbar. Wenn die Sitzungsdauer Ihres IdP 8 Stunden überschreitet, können Benutzer bei ihrem nächsten Besuch automatisch erneut authentifiziert werden, ohne dass Anmeldeinformationen eingegeben werden müssen. Konfigurieren Sie die Sitzungs- und Token-Gültigkeitsdauer Ihres IdP gemäß den Sicherheitsanforderungen Ihres Unternehmens.
- Aktualisierung der Anmeldeinformationen — Sitzungen werden automatisch mithilfe von OIDC-Aktualisierungstoken aktualisiert, ohne dass sich Benutzer erneut authentifizieren müssen
- Multi-Faktor-Authentifizierung — Wird unterstützt, wenn sie in Ihrem Identitätsanbieter konfiguriert ist. Der IdP verarbeitet MFA während der Anmeldung — es ist keine zusätzliche Konfiguration im Agent erforderlich. AWS DevOps

Verhalten beim Abmelden

Wenn ein Benutzer in der Web-App auf Abmelden klickt:

1. Alle Sitzungscookies werden sofort gelöscht
2. Der Benutzer wird zum OIDC-Abmeldeendpunkt des Identitätsanbieters weitergeleitet, um die SSO-Sitzung zu beenden
3. Wenn eine Abmelde-Umleitungs-URI konfiguriert ist, wird der Benutzer zurück zur Willkommensseite der Web-App weitergeleitet

Benutzerzugriff wird widerrufen

Um einem Benutzer sofort den Zugriff zu entziehen, können Sie seine Sitzungen direkt im Admin-Portal Ihres Identitätsanbieters widerrufen:

- Okta — Navigieren Sie in der Okta Admin-Konsole zu Verzeichnis > Personen, wählen Sie den Benutzer aus und wählen Sie Weitere Aktionen > Benutzersitzungen löschen
- Microsoft Entra ID — Navigieren Sie im Azure-Portal zu Benutzer, wählen Sie den Benutzer aus und klicken Sie auf Sitzungen widerrufen

Sicherheitsüberlegungen

Geheimer Client-Schlüssel — Der geheime Client-Schlüssel, den Sie bei der Einrichtung angeben, wird mit Ihrem vom Kunden verwalteten KMS-Schlüssel verschlüsselt, sofern Sie einen beim Erstellen des Agent Space angegeben haben, oder andernfalls mit einem diensteigenen Schlüssel.

Es wird weder in API-Antworten zurückgegeben noch nach der Erstkonfiguration in der Konsole angezeigt.

Rotation der geheimen Client-Schlüssel — Entra-Clientgeheimnisse haben ein konfigurierbares Ablaufdatum. Richten Sie mithilfe der Option „Client-Schlüssel rotieren“ in der AWS DevOps Agent-Konsole eine Erinnerung ein, sodass der geheime Schlüssel rotiert werden soll, bevor er abläuft. Wenn der geheime Schlüssel abläuft, können sich Benutzer erst anmelden, wenn er rotiert wird.

Verwaltung der Token-Lebensdauer — Die Lebensdauer der von Ihrem Identitätsanbieter ausgegebenen Token (Zugriffstoken, Aktualisierungstoken) wird durch die Konfiguration Ihres IdP gesteuert. Wir empfehlen, die entsprechenden Token-Lebensdauern in Ihrem IdP zu konfigurieren:

- Okta — Konfigurieren Sie die Gültigkeitsdauer von Token unter Sicherheit > API > Autorisierungsserver > Zugriffsrichtlinien
- Microsoft Entra ID — Konfigurieren Sie die Gültigkeitsdauer von Token mithilfe von Richtlinien zur Tokenlebensdauer

Gruppenanspruch — Aktivieren Sie den Gruppenanspruch nicht in der Token-Konfiguration Ihres Identitätsanbieters. AWS DevOps Der Agent verwendet derzeit keine Gruppenmitgliedschaft von Ihrem IdP.

Benutzer-ID — Der AWS DevOps Agent verwendet einen anbieterspezifischen Anspruch, um Benutzer eindeutig zu identifizieren:

- Okta — Verwendet den `sub` Anspruch aus dem ID-Token
- Microsoft Entra ID — Verwendet den Anspruch `oid` (Objektbezeichner) aus dem ID-Token

Diese Kennungen sind unveränderlich und erscheinen zu Prüfzwecken in CloudTrail Protokollen.

Trennen der Verbindung zum externen IdP

1. Wählen Sie in der AWS DevOps Agent-Konsole Ihren Agent Space aus
2. Gehen Sie zur Registerkarte Zugriff
3. Wählen Sie unter Benutzerzugriff die Option Trennen
4. Überprüfen Sie die im Bestätigungsdialogfeld aufgeführten Auswirkungen und bestätigen Sie

Durch das Trennen der Verbindung wird:

- Entfernen Sie die IdP-Konfiguration aus dem Agent Space
- Verhindern Sie, dass sich Benutzer über den externen Identitätsanbieter anmelden
- Entfernen Sie den individuellen Chat- und Artefaktverlauf, der mit IdP-Benutzerkonten verknüpft ist

Aktive Benutzersitzungen werden fortgesetzt, bis sie ablaufen oder die nächste Aktualisierung der Anmeldeinformationen fehlschlägt.

Fehlerbehebung

- Die Weiterleitung zum IdP schlägt fehl — Stellen Sie sicher, dass die Aussteller-URL mit dem OIDC-Discovery-Endpunkt Ihres IdP übereinstimmt. Stellen Sie für Okta sicher, dass der Aussteller auf der Registerkarte Anmelden auf Okta-URL (nicht dynamisch) eingestellt ist. Verwenden Sie für Entra das Format. `https://login.microsoftonline.com/{tenant-id}/v2.0`
- Zugriff verweigert oder Richtlinienfehler (Okta) — Stellen Sie sicher, dass der Benutzer oder seine Gruppe der Anwendung unter Zuweisungen zugewiesen ist. Klicken Sie auf Anmelden **>** Regeln für die Anmelderichtlinie.
- IdP-Konfigurationsfehler nach der Anmeldung — Ihr Identitätsanbieter hat kein Aktualisierungstoken zurückgegeben. Stellen Sie sicher, dass der `offline_access` Geltungsbereich und der Gewährungstyp für das Aktualisierungstoken aktiviert sind:
  - Okta — Gehen Sie zur Registerkarte „Allgemein“ Ihrer Anwendung und aktivieren Sie unter „Art der Gewährung“ das Kontrollkästchen „Token aktualisieren“
  - Entra — Gehen Sie zu den API-Berechtigungen und stellen Sie sicher, dass sie unter `offline_access` Delegierte Berechtigungen aufgeführt sind
- Die Authentifizierung ist erfolgreich, aber die Web-App zeigt einen Fehler an — Stellen Sie sicher, dass die Umleitungs-URI in Ihrem IdP genau mit der Callback-URL übereinstimmt, die in der AWS DevOps Agentenkonsole angezeigt wird.
- Authentifizierungsfehler — Wenn der optionale Gruppenanspruch in Ihrem IdP aktiviert ist, deaktivieren Sie ihn. AWS DevOps Der Agent verwendet keine Gruppenansprüche.
- Die Anmeldung schlägt nach der IdP-Authentifizierung fehl — Für Entra **`requestedAccessTokenVersion`** ist verify **`null`** im Anwendungsmanifest nicht auf eingestellt. Stellen Sie für Okta sicher, dass die Aussteller-URL korrekt ist.
- Fehlerseite nach dem Klicken auf Abmelden (Okta) — Wenn Sie nach dem Abmelden einen **`post_logout_redirect_uri`** Fehler sehen, fügen Sie auf der Registerkarte Allgemein Ihrer **`https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome`** Okta-Anwendung als Umleitungs-URI für die Abmeldung hinzu.

- Benutzer bleiben nach dem Abmelden auf der Identity-Provider-Seite (Entra) — Um Benutzer nach dem Abmelden zurück zur Web-App umzuleiten, fügen Sie auf der Authentifizierungsseite Ihrer Entra-Anwendung **`https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome`** als Umleitungs-URI hinzu.

Verschlüsselung im Ruhezustand für den AWS DevOps Agenten

AWS DevOps Der Agent verschlüsselt alle gespeicherten Kundendaten. Standardmäßig verwendet der AWS DevOps Agent AWS eigene Schlüssel, um Ihre Daten ohne zusätzliche Kosten automatisch zu verschlüsseln. Sie können die Verwendung AWS eigener Schlüssel nicht einsehen, verwalten oder überprüfen. Sie müssen jedoch keine Maßnahmen ergreifen, um diese Schlüssel zu schützen. Ihre Daten werden automatisch gesichert.

Sie können Ihre Daten mit einem symmetrischen, vom Kunden verwalteten Schlüssel verschlüsseln, den Sie im AWS Key Management Service (AWS KMS) erstellen, besitzen und verwalten. Da Sie die volle Kontrolle über diese Verschlüsselungsebene haben, können Sie Aufgaben wie die folgenden ausführen:

- Festlegung und Pflege wichtiger Richtlinien
- Aktivieren und Deaktivieren wichtiger Richtlinien
- Kryptographisches Material mit rotierendem Schlüssel
- Hinzufügen von -Tags
- Erstellen von Schlüsselaliasen
- Schlüssel für das Löschen von Schlüsseln planen

Weitere Informationen finden Sie unter Vom Kunden verwaltete Schlüssel im AWS Key Management Service Developer Guide.

> **Note**
>
> AWS DevOps Der Agent aktiviert automatisch die Verschlüsselung im Ruhezustand mithilfe AWS eigener Schlüssel, um Kundendaten kostenlos zu schützen. Wenn Sie einen vom Kunden verwalteten Schlüssel verwenden, fallen die standardmäßigen AWS KMS-Gebühren an. Weitere Informationen zur Preisgestaltung finden Sie unter Preise für den AWS Key Management Service.

Kundenseitig verwaltete Schlüssel

Von Kunden verwaltete Schlüssel sind KMS-Schlüssel in Ihrem AWS Konto, die Sie erstellen, besitzen und verwalten. Sie haben die volle Kontrolle über diese KMS-Schlüssel, einschließlich der Festlegung und Verwaltung ihrer wichtigsten Richtlinien.

Wenn Sie einen vom Kunden verwalteten Schlüssel konfigurieren, verwendet der AWS DevOps Agent ihn zum Schutz sensibler Ressourcendaten. AWS DevOps Der Agent verwendet die Envelope-Verschlüsselung mit dem hierarchischen Schlüsselbund des AWS Encryption SDK. Ihr KMS-Schlüssel wird verwendet, um Filialschlüssel zu generieren, die wiederum Ihre Daten schützen.

Sie können einen vom Kunden verwalteten Schlüssel angeben, wenn Sie die folgenden Ressourcen erstellen:

- Agent Space — Verschlüsselt Agent Space-Details und Inhalte, die mit der DevOps Agent Web App erstellt wurden und sich auf Untersuchungen, Fähigkeiten und Chat beziehen.
- Service — Verschlüsselt die Anmeldeinformationen für Dienste von Drittanbietern im Ruhezustand.

Gehen Sie wie folgt vor, um einen vom Kunden verwalteten Schlüssel in AWS DevOps Agent zu konfigurieren.

Schritt 1: Erstellen eines kundenverwalteten Schlüssels

Sie können einen symmetrischen, vom Kunden verwalteten Schlüssel mithilfe der AWS KMS-Konsole oder der AWS KMS-API erstellen. Der Schlüssel muss die folgenden Anforderungen erfüllen:

Eigenschaft	Anforderung
Schlüsseltyp	Symmetrisch
Schlüsselspezifikation	`SYMMETRIC_DEFAULT`
Schlüsselnutzung	`ENCRYPT_DECRYPT`

> **Note**
>
> AWS DevOps Der Agent unterstützt nur KMS-Schlüssel zur symmetrischen Verschlüsselung mit der `SYMMETRIC_DEFAULT` Schlüsselspezifikation und der `ENCRYPT_DECRYPT`

Schlüsselverwendung. Schlüssel für mehrere Regionen und asymmetrische Schlüssel werden derzeit nicht unterstützt.

Weitere Informationen finden Sie unter Creating a symmetric customer managed key im AWS Key Management Service Developer Guide.

Schritt 2: Legen Sie die Schlüsselrichtlinie fest

Schlüsselrichtlinien steuern den Zugriff auf den vom Kunden verwalteten Schlüssel. Jeder vom Kunden verwaltete Schlüssel muss über genau eine Schlüsselrichtlinie verfügen, die aussagt, wer den Schlüssel wie verwenden kann.

Ihre Schlüsselrichtlinie muss sowohl dem aufrufenden Prinzipal (Ihre IAM-Identität) als auch dem AWS DevOps Agent-Dienst Berechtigungen gewähren. AWS DevOps Der Agent greift mit zwei Anmeldeinformationen auf Ihren Schlüssel zu:

1. Ihre Anruferanmeldedaten — Werden für alle synchronen Vorgänge verwendet, einschließlich der Schlüsselvalidierung, Verschlüsselung bei der Ressourcenerstellung und für jeden API-Aufruf, der eine direkte Antwort an den Anrufer zurückgibt.
2. AWS DevOps Agent Service Principal — Wird für asynchrone Vorgänge verwendet, die im Hintergrund ausgeführt werden, z. B. betriebliche Untersuchungen, Vorfallanalysen, Ereigniskorrelation und Generierung von Ursachenanalysen.

In der folgenden Tabelle sind die erforderlichen KMS-Aktionen aufgeführt:

KMS-Aktion	Description
`kms:DescribeKey`	Überprüfen Sie die Schlüsselkonfiguration bei der Erstellung der Ressource
`kms:GenerateDataKey`	Generieren Sie Datenverschlüsselungsschlüssel für die Umschlagverschlüsselung
`kms:Decrypt`	Daten entschlüsseln
`kms:Encrypt`	Daten verschlüsseln

KMS-Aktion	Description
`kms:ReEncrypt`	Verschlüsseln Sie Daten erneut mit demselben oder einem anderen Schlüssel

AWS DevOps Der Agent validiert all diese Berechtigungen bei der Konfiguration mithilfe von Testläufen. Wenn eine Berechtigung fehlt, schlägt die Anfrage mit einer Ausnahme fehl.

Es folgt eine Beispielschlüsselrichtlinie. Ersetzen Sie die Platzhalterwerte durch Ihre eigenen.

```
{
  "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/*"
        }
      }
    }
  ]
}
```

Die Richtlinie enthält die folgenden Aussagen:

- AllowKeyAdministration— Gewährt dem Account-Root vollen administrativen Zugriff auf den Schlüssel. Ersetze es `111122223333` durch deine AWS Konto-ID.
- AllowCallerAccessViaService— Gewährt Ihren IAM-Prinzipalen die KMS-Berechtigungen, die für alle synchronen AWS DevOps Agentenvorgänge erforderlich sind. Dazu gehören die Schlüsselvalidierung bei der Erstellung der Ressource sowie Verschlüsselungs- und Entschlüsselungsvorgänge für jeden API-Aufruf, der eine direkte Antwort an den Aufrufer zurückgibt. Diese `kms:ViaService` Bedingung stellt sicher, dass Sie den Schlüssel nur über den AWS DevOps Agent-Dienst verwenden können. `111122223333`Ersetzen Sie es durch Ihre AWS Konto-ID und `us-east-1` durch Ihre AWS Region.
- AllowDevOpsAgentServiceAccessForAgentSpace/AllowDevOpsAgentServiceAccessForService— Gewährt dem `aidevops.amazonaws.com` Dienstprinzipal die für asynchrone Operationen erforderlichen KMS-Berechtigungen. AWS DevOps Der Agent verwendet diesen Dienstprinzipal, um Ihre Daten zu verschlüsseln und zu entschlüsseln, wenn er Hintergrundoperationen durchführt, z. B. Betriebsuntersuchungen durchführt, Vorfälle analysiert, Ereignisse dienstübergreifend korreliert und Ursachenanalysen erstellt. Ohne diesen Zugriff kann der AWS DevOps Agent die verschlüsselten Daten, die für die Durchführung von Untersuchungen in Ihrem Namen benötigt werden, nicht lesen. Die `aws:SourceArn` Bedingung schränkt den Zugriff auf Anfragen ein, die von Ihren AWS DevOps Agentenressourcen stammen, und die `kms:EncryptionContext` Bedingung stellt sicher, dass der Verschlüsselungskontext mit Ihrer Ressource ARNs übereinstimmt. `111122223333`Ersetzen Sie es durch Ihre AWS Konto-ID und `us-east-1` durch Ihre AWS Region.

Weitere Informationen zu wichtigen Richtlinien finden Sie unter Wichtige Richtlinien in AWS KMS im AWS Key Management Service Developer Guide.

Schritt 3: Geben Sie den Schlüssel an, wenn Sie eine Ressource erstellen

Nachdem Sie Ihren Schlüssel erstellt und die Schlüsselrichtlinie konfiguriert haben, können Sie den Schlüssel bei der Erstellung von AWS DevOps Agentenressourcen angeben.

Konsole

So konfigurieren Sie einen vom Kunden verwalteten Schlüssel bei der Erstellung eines Agentenbereichs in der Konsole:

1. Öffnen Sie die AWS DevOps Agent-Konsole.
2. Wählen Sie Create Agent Space oder Service registrieren.
3. Geben Sie die Details des Agentenbereichs ein (Name, Beschreibung und IAM-Rolle).
4. Erweitern Sie den Abschnitt Erweiterte Konfiguration.
5. Wählen Sie unter Verschlüsselungsschlüsseltyp die Option Vom Kunden verwalteter Schlüssel aus.
6. Wählen Sie einen KMS-Schlüssel aus der Dropdownliste aus, oder geben Sie einen KMS-Schlüssel-ARN ein.
7. Überprüfen Sie die Schlüsselrichtlinie, die im erweiterbaren Abschnitt Schlüsselrichtlinie angezeigt wird. Stellen Sie sicher, dass Sie diese Richtlinie an Ihren KMS-Schlüssel angehängt haben. Sie können die Schaltfläche „Kopieren“ verwenden, um die Richtlinie zu kopieren.
8. Schließen Sie die verbleibende Konfiguration ab und wählen Sie Create.

> **Note**
>
> Wenn Sie Ihren KMS-Schlüssel nicht in der Dropdownliste sehen, überprüfen Sie, ob der Schlüssel die Anforderungen in Schritt 1 erfüllt und ob Sie über die `kms:DescribeKey` erforderlichen Berechtigungen verfügen`kms:ListKeys`.

API

Einen Agent Space mit einem vom Kunden verwalteten Schlüssel erstellen

Geben Sie den `kmsKeyArn` Parameter an, wenn Sie einen Agentenbereich erstellen. Der Wert muss der vollständige KMS-Schlüssel-ARN sein.

```
{
```

```
  "name": "my-agent-space",
  "description": "An encrypted agent space",
  "kmsKeyArn": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

Registrierung eines Dienstes mit einem vom Kunden verwalteten Schlüssel

Geben Sie den `kmsKeyArn` Parameter bei der Registrierung eines Dienstes an. Der Wert muss der vollständige KMS-Schlüssel-ARN sein. Dieser Parameter wird für alle Diensttypen unterstützt, einschließlich Dynatrace-,,, ServiceNow PagerDuty GitLab GitHub, und MCP-Servern.

```
{
  "service": "dynatrace",
  "kmsKeyArn": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "serviceDetails": { ... }
}
```

> **Note**
>
> Sie müssen den vom Kunden verwalteten Schlüssel bei der Erstellung der Ressource angeben. Sie können den vom Kunden verwalteten Schlüssel für eine vorhandene Ressource nicht hinzufügen oder ändern.

AWS DevOps Kontext der Agentenverschlüsselung

Ein Verschlüsselungskontext ist ein Satz nicht geheimer Schlüssel-Wert-Paare, die zusätzliche kontextbezogene Informationen zu den Daten enthalten. AWS KMS verwendet den Verschlüsselungskontext als zusätzliche authentifizierte Daten, um die authentifizierte Verschlüsselung zu unterstützen. Wenn Sie einen Verschlüsselungskontext in eine Anforderung zur Datenverschlüsselung aufnehmen, bindet AWS KMS den Verschlüsselungskontext an die verschlüsselten Daten. Um Daten zu entschlüsseln, müssen Sie denselben Verschlüsselungskontext in die Anfrage aufnehmen.

AWS DevOps Der Agent verwendet den folgenden Verschlüsselungskontext für alle kryptografischen Operationen:

```
{
```

```
  "aws-crypto-ec:aws:aidevops:arn": "arn:aws:aidevops:{region}:{accountId}:
{resourceType}/{resourceId}"
}
```

Der Wert für den Verschlüsselungskontext ist der ARN der AWS DevOps Agentenressource, die verschlüsselt wird. Sie können diesen Verschlüsselungskontext in Ihren wichtigsten Richtlinienbedingungen und in AWS CloudTrail Protokollen verwenden, um zu überprüfen, wie Ihr Schlüssel verwendet wird.

Schlüsselverwaltung

Wenn Sie das Löschen Ihres KMS-Schlüssels deaktivieren oder planen, kann der AWS DevOps Agent Ihre Daten nicht entschlüsseln. Dies führt zu `AccessDeniedException` Fehlern bei Vorgängen, die verschlüsselte Daten lesen.

> **Important**
>
> Wenn Sie sich dafür entscheiden, einen vom Kunden verwalteten Schlüssel zu verwenden, sind Sie für die Verwaltung des Schlüssels und seiner Berechtigungen verantwortlich. Wenn der Schlüssel deaktiviert oder gelöscht wird oder wenn der AWS DevOps Agent die Berechtigung zur Verwendung des Schlüssels verliert, verlieren Sie den Zugriff auf die verschlüsselten Daten.

In der folgenden Tabelle werden die häufigsten Fehlerszenarien beschrieben:

Action	Auswirkung
Wichtige Richtlinienberechtigungen wurden widerrufen	`AccessDeniedException` bei Verschlüsselungs- und Entschlüsselungsvorgängen
Der KMS-Schlüssel ist deaktiviert	`DisabledException` bei Verschlüsselungs- und Entschlüsselungsvorgängen
Der KMS-Schlüssel ist für die Löschung geplant	`KMSInvalidStateException` bei Verschlüsselungs- und Entschlüsselungsvorgängen

Action	Auswirkung
Der KMS-Schlüssel ist gelöscht	Dauerhafter Datenverlust — verschlüsselte Daten können nicht wiederhergestellt werden

Bevor Sie einen Schlüssel deaktivieren oder löschen:

1. Stellen Sie sicher, dass keine aktiven AWS DevOps Agentenressourcen von dem Schlüssel abhängen.
2. Erwägen Sie, zuerst den Schlüssel zu deaktivieren, um die Auswirkungen zu testen, bevor Sie den Löschvorgang planen.
3. AWS KMS erzwingt eine Mindestwartezeit vor dem Löschen des Schlüssels, sodass Sie bei Bedarf Zeit haben, den Vorgang zu kündigen.

> Hinweis: AWS DevOps Der Agent verschlüsselt Daten nicht automatisch erneut unter einem neuen Schlüssel. Wenn Sie zu einem neuen, vom Kunden verwalteten Schlüssel wechseln müssen, müssen Sie eine neue Ressource mit dem neuen Schlüssel erstellen.

Überwachen Ihrer Verschlüsselungsschlüssel

Wenn Sie einen vom Kunden verwalteten Schlüssel mit AWS DevOps Agent verwenden, können Sie AWS CloudTraildamit Anfragen verfolgen, die der AWS DevOps Agent an AWS KMS sendet.

Sie können CloudTrail Ereignisse nach folgenden Kriterien filtern:

- Quelle des Ereignisses — `kms.amazonaws.com`
- Schlüssel für den Verschlüsselungskontext — `aws-crypto-ec:aws:aidevops:arn`
- Schlüssel-ARN — Ihr vom Kunden verwalteter Schlüssel-ARN in den Anforderungsparametern

Weitere Informationen finden Sie unter Protokollieren von AWS KMS-API-Aufrufen mit AWS CloudTrail im AWS Key Management Service Developer Guide.

VPC-Endpunkte (AWS PrivateLink)

Sie können AWS PrivateLink es verwenden, um eine private Verbindung zwischen Ihrer VPC und Ihrem AWS DevOps Agenten herzustellen. Sie können auf den AWS DevOps Agenten zugreifen, als ob er sich in Ihrer VPC befände, ohne ein Internet-Gateway, ein NAT-Gerät, eine VPN-Verbindung oder eine Direct Connect-Verbindung verwenden zu müssen. Instances in Ihrer VPC benötigen keine öffentlichen IP-Adressen, um auf AWS DevOps Agent zuzugreifen.

Sie stellen diese private Verbindung her, indem Sie einen Schnittstellenendpunkt erstellen, der von AWS PrivateLink betrieben wird. Wir erstellen eine Endpunkt-Netzwerkschnittstelle in jedem Subnetz, das Sie für den Schnittstellen-Endpunkt aktivieren. Dabei handelt es sich um vom Anforderer verwaltete Netzwerkschnittstellen, die als Einstiegspunkt für den Datenverkehr dienen, der für den Agenten bestimmt ist. AWS DevOps

Weitere Informationen finden Sie unter Access AWS Services through AWS PrivateLink im _AWS Guide_. PrivateLink

Überlegungen zu VPC-Endpunkten für AWS DevOps Agenten

Bevor Sie einen Schnittstellen-Endpunkt für AWS DevOps Agenten einrichten, lesen Sie die Überlegungen im PrivateLink _AWS Guide_.

AWS DevOps Der Agent unterstützt API-Aufrufe über die folgenden VPC-Endpunkte.

Kategorie	Endpunktsuffix
AWS DevOps API-Aktionen auf der Agentenst euerungsebene	`aidevops`
AWS DevOps Runtime-Operationen für Agenten	`aidevops-dataplane`
AWS DevOps Webhook-Ereignisse für Agenten	`event-ai`

Erstellen Sie einen Schnittstellen-Endpunkt für Agent AWS DevOps

Sie können einen Schnittstellenendpunkt für AWS DevOps Agent entweder mit der Amazon VPC-Konsole oder der AWS Befehlszeilenschnittstelle (AWS CLI) erstellen. Weitere Informationen finden Sie unter Erstellen eines Schnittstellenendpunkts im _AWS PrivateLink Guide_.

Erstellen Sie einen Schnittstellenendpunkt für AWS DevOps Agent mit den folgenden Servicenamen:

- com.amazonaws. {Region} .aidevops
- com.amazonaws. {Region} .aidevops-Datenebene
- com.amazonaws. {Region} .event-ai

Wenn der Endpunkt erstellt wurde, haben Sie die Möglichkeit, einen privaten DNS-Hostnamen zu aktivieren. Aktivieren Sie diese Einstellung, indem Sie Privaten DNS-Namen aktivieren in der VPC-Konsole auswählen, wenn Sie den VPC-Endpunkt erstellen.

Wenn Sie privates DNS für den Schnittstellenendpunkt aktivieren, können Sie API-Anfragen an den AWS DevOps Agenten stellen, indem Sie dessen standardmäßigen regionalen DNS-Namen verwenden. Das folgende Beispiel zeigt das Format des standardmäßigen regionalen DNS-Namens.

- cp.aidevops. {Region} .api.aws
- dp.aidevops. {Region} .api.aws
- ereignis-AI. {Region} .api.aws

Erstellen einer Endpunktrichtlinie für Ihren Schnittstellen-Endpunkt

Eine Endpunktrichtlinie ist eine IAM-Ressource, die Sie an einen Schnittstellen-Endpunkt anfügen können. Die standardmäßige Endpunktrichtlinie ermöglicht den vollen Zugriff auf den AWS DevOps Agenten über den Schnittstellenendpunkt. Um den Zugriff zu kontrollieren, der dem AWS DevOps Agenten von Ihrer VPC aus gewährt wird, fügen Sie dem Schnittstellenendpunkt eine benutzerdefinierte Endpunktrichtlinie hinzu.

Eine Endpunktrichtlinie gibt die folgenden Informationen an:

- Die Principals, die Aktionen ausführen können (AWS Konten, IAM-Benutzer und IAM-Rollen).
- Aktionen, die ausgeführt werden können
- Die Ressourcen, auf denen die Aktionen ausgeführt werden können.

Weitere Informationen finden Sie unter Steuern des Zugriffs auf Services mithilfe von Endpunktrichtlinien im _AWS Guide_. PrivateLink

Konformitätsprüfung für den AWS DevOps Agenten

Externe Prüfer bewerten die Sicherheit und Konformität der AWS Dienste im Rahmen mehrerer AWS Compliance-Programme. AWS DevOps Der Beauftragte fällt in den Geltungsbereich der folgenden Compliance-Programme: BIO, C5, CISPE, CPSTIC, ENS High, FINMA, GNS, GSMA, HITRUST, IRAP, ISMAP, ISO (ISO/IEC 27001, 27017, 27018, 27701, 22301, 20000, 9001), CSA STAR, MTCS, OSPAR, PCI, Pinakes und SOC. PiTuKri Darüber hinaus ist der AWS DevOps Agent HIPAA-fähig. Unsere externen Prüfer werden AWS DevOps Agent in den nächsten Prüfungszyklen im Hinblick auf diese Compliance-Programme überprüfen und testen.

Eine Liste der AWS Dienstleistungen im Rahmen bestimmter Compliance-Programme finden Sie unter AWS Dienstleistungen im Umfang der einzelnen Compliance-Programme. Allgemeine Informationen finden Sie unter AWS -Compliance-Programme.

Mit AWS Artifact können Sie Prüfberichte von Drittanbietern herunterladen. Weitere Informationen finden Sie unter Berichte in AWS Artifact herunterladen.

Ihre Verantwortung für die Einhaltung von Vorschriften bei der Verwendung von AWS DevOps Agent hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS bietet die folgenden Ressourcen zur Unterstützung bei der Einhaltung von Vorschriften:

- Schnellstartanleitungen zu Sicherheit und Compliance — In diesen Bereitstellungsleitfäden werden architektonische Überlegungen erörtert und Schritte für die Implementierung von Umgebungen beschrieben, auf denen auf Sicherheit und Compliance ausgerichtete Basisumgebungen eingerichtet werden. AWS
- AWS Ressourcen zur Einhaltung von Vorschriften — Eine Sammlung von Arbeitsmappen und Leitfäden, die möglicherweise auf Ihre Branche und Ihren Standort zutreffen.
- AWS Config — Dieser AWS Service bewertet, wie gut Ihre Ressourcenkonfigurationen internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- AWS Security Hub — Dieser AWS Service bietet einen umfassenden Überblick über Ihren Sicherheitsstatus innerhalb AWS. Security Hub verwendet Sicherheitskontrollen, um Ihre AWS Ressourcen zu bewerten und Ihre Einhaltung der Sicherheitsstandards und Best Practices der Sicherheitsbranche zu überprüfen.

Kontingente

AWS DevOps Die Agentenkontingente umfassen die Anzahl der Agentenplätze, gleichzeitige Untersuchungen und mehr. Sie können für einige Kontingente eine Erhöhung beantragen, aber nicht alle Kontingente können erhöht werden. Diese Erhöhungen werden nicht sofort gewährt, sodass es einige Stunden bis Tage dauern kann, bis Ihre Erhöhung wirksam wird. Sofern nicht anders angegeben, gilt jedes Kontingent spezifisch für eine Region.

In der folgenden Tabelle werden die Kontingente für AWS DevOps Agent beschrieben.

Name	Standard	Anpassbar	Description
Agentenplätze pro Konto pro Region	100	Ja	Die maximale Anzahl von Agentenräumen, die Sie pro Konto in jeder AWS Region einrichten können.
Gleichzeitige Untersuchungen pro Agentenbereich	3	Ja	Die maximale Anzahl von Untersuch ungen zur Behebung von Vorfällen, die gleichzeitig in einem einzigen Agentenbe reich ausgeführt werden können.
Gleichzeitige Evaluierungen pro Agentenbereich	1	Nein	Die maximale Anzahl von Evaluierungen zur Verhinderung von Zwischenfällen, die gleichzeitig in einem einzelnen Agentenbe reich ausgeführt werden können.

Name	Standard	Anpassbar	Description
Gleichzeitige On-Demand-Aufrufe pro Agentenbereich	10	Ja	Die maximale Anzahl von DevOps On-Demand-Aufrufen, die gleichzeitig in einem einzelnen Agentenbereich ausgeführt werden können.

Beantragen einer Kontingenterhöhung

Sie können eine Erhöhung des Kontingents beantragen, indem Sie eine der folgenden Optionen verwenden:

- Von der AWS Management Console aus — Öffnen Sie die Service Quotas Quotas-Konsole. Wählen Sie im Navigationsbereich AWS -Services. Wählen Sie DevOps Agent aus, wählen Sie ein Kontingent aus und folgen Sie den Anweisungen, um eine Erhöhung des Kontingents zu beantragen. Weitere Informationen finden Sie unter Beantragen einer Kontingenterhöhung im Service-Quotas-Benutzerhandbuch.
- Über die AWS CLI — Verwenden Sie den request-service-quota-increase AWS CLI-Befehl. Weitere Informationen finden Sie unter Beantragen einer Kontingenterhöhung im Service-Quotas-Benutzerhandbuch.

Dokumentverlauf

In der folgenden Tabelle werden die wichtigen Änderungen an der Dokumentation seit der letzten Version von AWS DevOps Agent beschrieben. Um Benachrichtigungen über Aktualisierungen dieser Dokumentation zu erhalten, können Sie einen RSS-Feed abonnieren.

Änderungen	Beschreibung	Date
Anweisungen für den Agenten	Es wurden globale und agentenspezifische Anweisungen (AGENTS.md) hinzugefügt, die für jede Sitzung gelten.	21. Mai 2026
Triage-Status „Skill überspringen“ und „SKIPPED“	Es wurden ein Beispiel für eine Fähigkeit zur Filterung von Vorfällen (Überspringen des Wartungsfensters), die Entscheidung für die Triage-Entscheidung ÜBERSPRUNGEN, Anweisungen zur Korrektur von Triage-Entscheidungen und das Ereignis Investigation Skipped hinzugefügt. EventBridge	20. Mai 2026
Senden von Dateianhängen	Dokumentation zum Anhängen von Bildern, Dokumenten und Codedateien an Chat-Nachrichten hinzugefügt. Dazu gehören unterstützte Dateitypen, Beschränkungen und Anwendungsfälle.	19. Mai 2026
Priorisierung von Empfehlungen	Es wurde AI-powered ein Backlog-Ranking hinzugefügt,	13. Mai 2026

Änderungen	Beschreibung	Date
	einschließlich der Anpassung von Prioritäten per Chat und Rangstabilität.	
Claude Code-Plugin für MCP	Im Abschnitt MCP-Integration wurde ein Verweis auf das Claude Code-Beispiel-Plugin hinzugefügt.	12. Mai 2026
Leitplanken für Genehmigungen	Es wurde ein Guardrail-Modell für Sitzungsrichtlinien hinzugefügt, das Standardberechtigungen, unterstützte zusätzliche Berechtigungen und durch die Guardrail blockierte Berechtigungen abdeckt.	7. Mai 2026
Neue statische IPs	Neue statische IP-Adressen für ausgehende Verbindungen in allen unterstützten Regionen hinzugefügt.	7. Mai 2026
Dokumentverlauf	Es wurde eine Seite mit dem Dokumentenverlauf hinzugefügt, um neue Dokumentation nachzuverfolgen.	5. Mai 2026
Verbindung mit dem Agenten DevOps	Dokumentation für fünf Zugriffsmethoden hinzugefügt: Web-App, MCP, ACP, Webhooks und API.	28. April 2026

Änderungen	Beschreibung	Date
Compliance-Validierung	Eine spezielle Seite zur Überprüfung der Einhaltung von Vorschriften wurde hinzugefügt.	15. April 2026
Erste Schritte mit AWS CloudFormation	Anleitung für CloudFormation die ersten Schritte hinzugefügt.	29. März 2026
Verbindung zu privat gehosteten Tools herstellen	Dokumentation für private Verbindungen hinzugefügt.	29. März 2026
Endpunkte der VPC-Schnittstelle	Dokumentation zu VPC-Endpunkt (AWS PrivateLink) hinzugefügt.	29. März 2026
EventBridge Amazon-Integration	EventBridge Integrationsleitfaden für ereignisgesteuerte Anwendungen hinzugefügt.	28. März 2026
EventBridge Referenz zu den Einzelheiten der Ereignisse	Referenz zu Ereignisdetails für die EventBridge Integration hinzugefügt.	28. März 2026
Kontingente	Seite mit Servicekontingenten hinzugefügt.	28. März 2026
Grafana verbinden	Dokumentation zur Grafana-Telemetrie-Integration hinzugefügt.	27. März 2026
Azure verbinden	Die Dokumentation zur Azure-Integration wurde hinzugefügt.	27. März 2026
Azure-Ressourcen verbinden	Verbindungsleitfaden für Azure Resources hinzugefügt.	27. März 2026

Änderungen	Beschreibung	Date
Azure verbinden DevOps	DevOps Azure-Verbindungsleitfaden hinzugefügt.	27. März 2026
Verbindung herstellen PagerDuty	Dokumentation zur PagerDuty Kommunikationsintegration hinzugefügt.	27. März 2026
Migration von Public Preview zu GA	Migrationsleitfaden für die Public Preview zur allgemeinen Verfügbarkeit hinzugefügt.	27. März 2026
Allgemeine Verfügbarkeit	Dies ist die erste allgemein verfügbare Version von AWS DevOps Agent.	30. März 2026

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.