Guida per l'utente

AWS DevOps Agente

AWS DevOps Agente: Guida per l’utente

Table of Contents

Informazioni AWS DevOps Agente

AWS DevOps Agent è un agente di frontiera che risolve e previene in modo proattivo gli incidenti, migliorando continuamente l'affidabilità e le prestazioni.

AWS DevOps Agent indaga sugli incidenti e identifica i miglioramenti operativi in qualità di ingegnere esperto. DevOps

L'agente lavora tramite:

- Imparare le tue risorse e le loro relazioni.
- Lavorare con gli strumenti, le competenze, gli archivi di codice e CI/CD le pipeline di osservabilità.
- Correlazione dei dati di telemetria, codice e distribuzione per comprendere le relazioni tra le risorse dell'applicazione.
- Supporto di applicazioni in ambienti multicloud e ibridi.

Funzionalità principali

AWS DevOps Agent offre funzionalità complete di risposta e prevenzione degli incidenti attraverso le seguenti funzionalità:

Always-on, risposta autonoma agli incidenti

AWS DevOps L'agente indaga autonomamente sui problemi nel momento in cui si verificano:

- Indagine automatizzata sugli incidenti: inizia a indagare immediatamente quando arriva un avviso o un ticket di assistenza
- AWS DevOps Agent Chat: interroga l'infrastruttura, analizza lo stato del sistema e guida le indagini utilizzando il linguaggio naturale in tutta l'app web DevOps Agent Space. Chat fornisce risposte sensibili al contesto in base alla pagina che stai visualizzando, ad esempio chiedendo informazioni sulle risorse in Topologia, indirizzando un'indagine o filtrando i consigli in Prevenzione.
- Piani di mitigazione dettagliati: forniscono azioni specifiche per risolvere gli incidenti, convalidare il successo e ripristinare le modifiche, se necessario
- Coordinamento automatizzato degli incidenti: indirizza le osservazioni, i risultati e le misure di mitigazione attraverso i tuoi canali di comunicazione preferiti come Slack e ServiceNow

- AWS Integrazione del AWS supporto: crea casi di supporto direttamente da un'indagine con un contesto immediato fornito agli esperti del AWS supporto

Prevenire incidenti futuri

AWS DevOps L'agente analizza i modelli relativi agli incidenti storici per aiutarti a passare dalla lotta antincendio reattiva al miglioramento operativo proattivo:

- Raccomandazioni mirate: offre miglioramenti specifici e attuabili che rafforzano quattro aree chiave: osservabilità (monitoraggio, avvisi, registrazione), ottimizzazione dell'infrastruttura (scalabilità automatica, ottimizzazione della capacità) e miglioramento della pipeline di implementazione (test, convalida).
- Apprendimento continuo: perfeziona i consigli in base al feedback del team

Ottieni di più dai tuoi strumenti DevOps

AWS DevOps Agent si integra con gli strumenti esistenti senza modificare i flussi di lavoro:

- Mappatura delle risorse applicative: crea un grafico topologico delle risorse dell'applicazione e delle relative relazioni
- Built-in integrazioni: funziona con i più diffusi strumenti di osservabilità (Amazon CloudWatch, Dynatrace, Datadog, New Relic e Splunk), repository di codice e CI/CD pipeline (azioni e repository, flussi di lavoro e repository) GitHub GitLab
- Integrazione personalizzata con strumenti: estendi le funzionalità connettendoti ai tuoi server Model Context Protocol (MCP) per strumenti aggiuntivi
- Interrogazioni conversazionali sull'infrastruttura: utilizza il linguaggio naturale per interrogare AWS risorse, metriche di sistema e stato degli allarmi senza dover navigare su più console. La chat comprende il contesto e conserva la cronologia delle conversazioni per le domande successive.

In che modo AWS DevOps L'agente funziona

AWS DevOps L'agente opera tramite un'architettura a doppia console. Gli amministratori utilizzano la console di AWS gestione per creare e gestire gli spazi degli agenti, configurare le integrazioni e configurare i controlli di accesso. I team operativi utilizzano l'app web AWS DevOps Agent per le attività quotidiane di risposta agli incidenti e di indagine. L'app Web è il luogo in cui gli operatori

possono interagire con le indagini degli agenti, esplorare la topologia delle applicazioni tra più account e conoscere i miglioramenti preventivi dell'osservabilità, del codice, delle pipeline e delle architetture dell'infrastruttura. Per ulteriori informazioni, consulta the section called "Prevenzione proattiva degli incidenti".

Il servizio è organizzato in base agli Agent Spaces, che sono contenitori logici che definiscono a cosa l'agente può accedere e a cosa può indagare. AWS DevOps Ogni Agent Space contiene le configurazioni AWS dell'account, le integrazioni di strumenti di terze parti e le autorizzazioni di accesso. Per ulteriori informazioni, consulta the section called "Cosa sono gli DevOps Agent Spaces?".

AWS DevOps Agent crea automaticamente una topologia applicativa che mappa le risorse e le relative relazioni. Questa topologia aiuta il servizio a comprendere l'architettura dell'applicazione durante le indagini. Per ulteriori informazioni, consulta the section called "Cos'è una topologia ad DevOps agenti?".

Vantaggi

- Riduzione del tempo medio di risoluzione (MTTR): l'indagine autonoma inizia immediatamente, accelerando la risoluzione degli incidenti da ore a minuti
- Prevenzione degli incidenti ricorrenti: le raccomandazioni mirate affrontano le cause profonde e rafforzano la resilienza del sistema
- Migliora l'efficienza operativa: libera il tuo team da attività di indagine ripetitive per concentrarsi sull'innovazione
- Lavora all'interno dei flussi di lavoro esistenti: si integra con gli strumenti e i processi esistenti senza interruzioni

Cos'è un'app Web per DevOps agenti?

AWS DevOps L'agente utilizza un'architettura a doppia console che separa le funzioni amministrative dalle day-to-day attività operative. Questo design consente agli amministratori di configurare il servizio mentre i team operativi si concentrano sulla risposta e sulla prevenzione degli incidenti.

Console

AWS DevOps Agent fornisce due interfacce distinte:

- AWS Console di gestione: gli amministratori utilizzano la console di AWS gestione per configurare e gestire AWS DevOps l'agente. In questa console, puoi the section called "Creazione di uno spazio per agenti" connettere AWS servizi e strumenti di terze parti e gestire le autorizzazioni di accesso per la tua organizzazione.
- DevOps App web per agenti: i team operativi utilizzano le app web di DevOps Agent Space per le attività quotidiane di risposta agli incidenti. Questa applicazione autonoma fornisce un'interfaccia in cui gli ingegneri a chiamata possono avviare indagini, interagire con l'agente tramite chat in linguaggio naturale, visualizzare le topologie delle applicazioni e rivedere i consigli sulla prevenzione degli incidenti.

Funzionalità delle app Web

L'app Web DevOps Agent offre le seguenti funzionalità principali:

- Risposta agli incidenti: la pagina consente di creare e tenere traccia delle indagini sugli incidenti, nonché di generare piani di mitigazione per risolvere gli incidenti.
- Prevenzione degli incidenti: nella pagina Prevenzione, qui troverai consigli per migliorare la tua posizione di osservabilità, i processi di consegna e l'architettura dell'infrastruttura per prevenire incidenti futuri.
- Topologia: la pagina Topologia fornisce una rappresentazione visiva interattiva delle risorse dell'account e delle loro relazioni tra tutte le risorse degli account collegati. È possibile visualizzare la topologia con diversi livelli di dettaglio utilizzando il menu a discesa «Mostra» per passare dalla visualizzazione Sistema, Contenitore e Risorse.
- Competenze: set di istruzioni modulari che estendono AWS DevOps Agent con funzionalità specializzate. Le competenze includono conoscenze di settore, metodologie di indagine e configurazioni di strumenti personalizzate per l'infrastruttura. Ogni abilità abilita strumenti specifici e fornisce la divulgazione progressiva delle istruzioni solo se pertinenti all'indagine.
- Interfaccia di chat in linguaggio naturale: disponibile in tutta l'app web, Chat è un assistente conversazionale basato sull'intelligenza artificiale che consente di interrogare l'infrastruttura, analizzare lo stato del sistema e svolgere indagini utilizzando il linguaggio naturale. La chat fornisce risposte sensibili al contesto in base alla pagina che stai visualizzando.

Autenticazione

AWS DevOps Agent supporta metodi di autenticazione flessibili per soddisfare diversi requisiti organizzativi:

- Integrazione con IAM Identity Center (accesso utente): le organizzazioni possono utilizzare AWS Identity Center (IAM Identity Center) per gestire centralmente l'accesso degli utenti alle app Web di DevOps Agent Space. IAM Identity Center può federarsi con provider di identità esterni tramite protocolli OIDC e SAML standard, inclusi provider come Okta, Ping Identity e Microsoft Entra ID. Questo metodo supporta l'autenticazione a più fattori da parte del tuo provider di identità.
- Autenticazione con provider di identità esterni (IdP): le organizzazioni possono connettere un provider di identità compatibile con OIDC, come Okta o Microsoft Entra ID, direttamente all'app Web Agent Space senza richiedere IAM Identity Center. Gli utenti accedono con le proprie credenziali aziendali tramite l'IdP. Per le istruzioni di configurazione, consulta. the section called "Configurazione dell'autenticazione tramite provider di identità esterno (IdP)"
- Link di autenticazione IAM (accesso amministratore): un metodo alternativo fornisce l'accesso diretto all'app Web dalla console di AWS gestione utilizzando la sessione di console esistente. Questa opzione è utile prima di implementare l'integrazione completa di Identity Center, ma le sessioni sono limitate a 10 minuti.

Cosa sono gli DevOps Agent Spaces?

Un DevOps Agent Space è un contenitore logico che definisce gli strumenti e l'infrastruttura a cui l' AWS DevOps agente ha accesso. Ogni Agent Space opera in modo indipendente con accesso al proprio AWS account, integrazioni di terze parti e autorizzazioni utente.

Un Agent Space rappresenta il limite di ciò a cui l' AWS DevOps agente può accedere e indagare durante la risposta agli incidenti. Quando si crea un Agent Space, si definisce a quali AWS account l'agente può accedere, a quali strumenti esterni può connettersi e quali utenti dell'organizzazione possono interagire con l'agente.

Ogni Agent Space funziona come una distribuzione indipendente di AWS DevOps Agent. Agent Space viene configurato tramite la console di AWS gestione, mentre i team operativi utilizzano l'app web di Agent Space per condurre indagini e rivedere i consigli all'interno di tale spazio.

Come vengono isolati gli Agent Spaces

Agent Spaces mantiene l'isolamento per garantire la sicurezza e prevenire accessi involontari tra diversi ambienti o team:

- AWS isolamento degli account: ogni Agent Space utilizza ruoli IAM dedicati che garantiscono l'accesso solo ad AWS account e risorse specifici. L'agente non può accedere a AWS risorse al di fuori di quelle configurate esplicitamente per Agent Space.
- Isolamento dell'accesso degli utenti: è possibile controllare quali utenti o gruppi possono accedere a ogni Agent Space. Ciò consente di allineare le autorizzazioni di accesso alla struttura organizzativa, garantendo che i team interagiscano solo con gli Agent Spaces designati.
- Isolamento dei dati: i dati delle indagini, la cronologia degli incidenti e le raccomandazioni vengono conservati separatamente all'interno di ciascun Agent Space. Le informazioni provenienti da un Agent Space non sono visibili o accessibili da un altro Agent Space.
- Isolamento dei dati della chat: anche la cronologia delle conversazioni in chat è isolata all'interno di ogni Agent Space. Le conversazioni e le domande in un Agent Space non sono visibili o accessibili da un altro Agent Space.

App Web Agent Space

Ogni Agent Space dispone di un'app Web dedicata accessibile dall'esterno della console di AWS gestione. Vedi the section called “Cos'è un'app Web per DevOps agenti?” per saperne di più sull'app web.

Quando utilizzare più Agent Spaces

Prendi in considerazione la possibilità di creare più Agent Spaces per supportare diverse esigenze organizzative:

- Separazione dei team: crea Agent Spaces dedicati per diversi team applicativi o unità aziendali per mantenere chiari i confini di proprietà nell'Agent Space.
- Isolamento dell'ambiente: separa gli ambienti di produzione e non di produzione in diversi Agent Spaces per evitare accessi accidentali tra ambienti.
- Limiti del servizio: allinea Agent Spaces a servizi o applicazioni specifici per mantenere le indagini mirate e pertinenti.

- Requisiti di conformità: configura Agent Spaces separati con diversi controlli di accesso o impostazioni di residenza dei dati per soddisfare i requisiti normativi.

> **Note**
>
> Quando si creano più Agent Spaces, è possibile utilizzare un AWS account dedicato come account principale per un Agent Space e collegare account applicativi distinti come account secondari. Questo approccio consente di mantenere controlli di accesso granulari garantendo al contempo che ogni Agent Space possa accedere solo alle risorse specifiche per l'ambito previsto, anche quando si utilizza la creazione automatica di ruoli.

Cos'è una topologia ad DevOps agenti?

AWS DevOps Agent's rileva e visualizza automaticamente le risorse e le relazioni all'interno delle applicazioni e utilizza la topologia risultante per comprendere l'infrastruttura durante le indagini sugli incidenti e quando formula raccomandazioni preventive.

Come vengono creati i grafici topologici

AWS DevOps Agent crea grafici di topologia attraverso diversi processi automatizzati:

- Rilevamento delle risorse: l'agente analizza automaticamente gli AWS account per identificare risorse come istanze di calcolo, servizi di storage, componenti di rete e database che fanno parte delle applicazioni.
- Rilevamento delle relazioni: l'agente analizza i dati di configurazione, gli CloudFormation stack e i tag delle risorse per determinare in che modo le risorse si relazionano tra loro.
- Mappatura del codice e dell'implementazione: quando è connesso alle CI/CD pipeline, l'agente collega le risorse dell'infrastruttura ai relativi processi di implementazione e modifica il codice dell'applicazione e dell'infrastruttura.
- Mappatura del comportamento di osservabilità: i dati provenienti da sistemi di osservabilità come Amazon CloudWatch Application Signals e Dynatrace vengono utilizzati per identificare i comportamenti osservati che indicano le relazioni tra le risorse.

Funzionalità chiave

La mappatura delle risorse offre diverse funzionalità che migliorano l'indagine e la prevenzione degli incidenti:

- Visualizzazione interattiva: esplora la topologia dell'applicazione tramite un grafico interattivo nell'app Web Operator. È possibile ingrandire e navigare nella topologia per comprendere relazioni complesse tra le risorse. Puoi anche usare Chat per interrogare informazioni sulla topologia utilizzando il linguaggio naturale, ad esempio «Mostrami tutte le funzioni Lambda connesse a questa tabella DynamoDB» o «Quali risorse sono interessate da questo allarme?».
- Indagine contestuale: durante le indagini sugli incidenti, l' AWS DevOps agente è assistito dalla topologia delle risorse per identificare i componenti interessati, comprendere il raggio di esplosione e tracciare il percorso dell'impatto attraverso i sistemi.
- Analisi delle cause principali: la comprensione dettagliata delle relazioni tra le risorse aiuta a individuare l'origine dei problemi, anche in sistemi distribuiti complessi con molte interdipendenze.
- Valutazione dell'impatto: durante l'analisi degli incidenti, l'agente può determinare meglio quali servizi a valle potrebbero essere interessati identificando le catene di dipendenza nella topologia.
- Consigli preventivi: l'agente utilizza le informazioni sulla topologia per formulare raccomandazioni mirate per il miglioramento della resilienza, suggerendo le modifiche che avranno l'impatto più significativo sulla stabilità del sistema.

Visualizzazioni topologiche

La visualizzazione della topologia nella pagina Topologia dell'Operator Web App offre diversi livelli di dettaglio:

- Appresa: la visualizzazione predefinita, generata dall'abilità Agent Space Understanding. Visualizza un riepilogo strutturato dell'infrastruttura organizzato per servizi logici e percorsi di richiesta.
- Sistema: mostra i confini di account e regioni di alto livello.
- Contenitore: visualizza gli stack di distribuzione come gli CloudFormation stack che contengono risorse correlate.
- Componenti: mostra i singoli componenti all'interno dei contenitori e le relative relazioni.
- Tutte le risorse: mostra la visualizzazione completa con tutte le risorse scoperte e le relative relazioni.

Scoperta delle risorse

Le risorse vengono scoperte attraverso due metodi:

- CloudFormation pile: l'agente elenca tutti gli CloudFormation stack e le relative risorse nell' AWS account principale e in tutti gli account secondari collegati. Questa funzionalità è supportata per tutti gli infrastructure-as-code strumenti utilizzati CloudFormation per la distribuzione, incluso il AWS Cloud Development Kit (AWS CDK).
- Resource Explorer: per le risorse non distribuite da CloudFormation, le risorse con tag vengono scoperte da AWS Resource Explorer. L' AWS account di destinazione deve avere Resource Explorer abilitato. Ciò è utile per identificare i limiti delle applicazioni per le risorse distribuite tramite la console di AWS gestione, il AWS servizio APIs o altri infrastructure-as-code framework.

Ambito di indagine che va oltre la topologia

Sebbene la topologia dell'applicazione fornisca un contesto importante durante le indagini, AWS DevOps Agent non si limita a esaminare solo le risorse mostrate nella topologia. L'agente può utilizzare fonti di dati aggiuntive, come AWS servizi APIs o strumenti di osservabilità connessi, per esaminare risorse che non rientrano nella topologia dell'applicazione.

Per limitare le risorse a cui l'agente ha accesso, limita la politica relativa al ruolo assegnato all'agente per l'accesso alle risorse tra account. Per ulteriori informazioni, consulta the section called "Limitazione dell'accesso degli agenti in un AWS Account".

Topologia e abilità Agent Space Understanding

Il grafico della topologia alimenta l'abilità acquisita da Agent Space Understanding, che codifica un riepilogo strutturato dell'infrastruttura da utilizzare durante le indagini. Al termine dell'individuazione della topologia per un nuovo spazio agente, il sistema genera automaticamente la skill Agent Space Understanding. Per ulteriori informazioni sulle competenze acquisite, vedere. the section called "Competenze apprese"

DevOps Competenze degli agenti

AWS DevOps Agent Skills sono set di istruzioni modulari che estendono le capacità dell'agente con conoscenze specialistiche del settore e metodologie di indagine personalizzate in base all'infrastruttura e ai flussi di lavoro operativi.

Cosa sono le competenze

Le competenze sono directory autonome contenenti istruzioni Markdown che forniscono funzionalità specializzate ad Agent. AWS DevOps AWS DevOps Agent supporta un sottoinsieme della specifica Agent Skills, uno standard aperto per le istruzioni e le risorse degli agenti di packaging, che supporta solo documenti non eseguibili: istruzioni Markdown, PDF, immagini e file di dati.

Ogni competenza richiede un SKILL.md file contenente le istruzioni che desideri fornire al tuo agente. AWS DevOps Oltre al SKILL.md file richiesto, le competenze possono includere:

- Flussi di lavoro di indagine per scenari o tipi di infrastruttura specifici.
- Materiali di riferimento, inclusi modelli di architettura e procedure operative.
- Targeting per tipo di agente: le competenze possono essere indirizzate a tipi di agenti specifici (Generic, Incident Triage On-demand, Incident RCA, Incident Mitigation, Evaluation) per ridurre il consumo di contesto e migliorare l'attenzione degli agenti.

Perché usare Skills

Le competenze trasformano AWS DevOps Agent da assistente generico a specialista per l'infrastruttura e i flussi di lavoro operativi. A differenza delle istruzioni monouso fornite in un messaggio di chat, le Skills sono funzionalità riutilizzabili che si caricano automaticamente quando sono pertinenti alle attività eseguite dall'agente. AWS DevOps

Principali vantaggi:

- Specializza il tuo agente: personalizza AWS DevOps l'agente con procedure di indagine, best practice e conoscenze organizzative specifiche per la tua infrastruttura e i tuoi modelli operativi.
- Riduci le ripetizioni: crea flussi di lavoro di indagine una sola volta e AWS DevOps Agent li utilizza automaticamente in tutte le indagini pertinenti, eliminando la necessità di fornire ripetutamente le stesse indicazioni.
- Composizione di funzionalità: combina più competenze per creare flussi di lavoro di indagine completi. AWS DevOps L'agente acquisisce diverse competenze durante l'esecuzione, ad esempio l'abilità di recuperare le distribuzioni dalla CI/CD pipeline personalizzata e l'abilità di cercare negli archivi di codice.
- Amplifica gli strumenti personalizzati: crea competenze che AWS DevOps guidino l'agente nell'utilizzo efficace degli strumenti server MCP personalizzati. Le competenze possono

documentare quando richiamare strumenti specifici, quali parametri utilizzare per diversi scenari e come interpretare i risultati per realizzare flussi di lavoro specifici per l'infrastruttura.

Come funzionano le competenze

Quando AWS DevOps l'Agente affronta un compito rilevante, acquisisce le competenze appropriate e segue le istruzioni per guidarne l'indagine. Ad esempio, una competenza di «Database Performance Investigation» potrebbe includere procedure dettagliate per l'analisi dei problemi di limitazione RDS, che consentano all'agente di controllare sistematicamente lo stato degli allarmi, analizzare le metriche di connessione e identificare le query lente.

Struttura delle competenze

Un'abilità è organizzata come una cartella contenente:

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

SKILL.md

Il `SKILL.md` è l'unico file obbligatorio. Contiene le istruzioni di base scritte in formato Markdown. Questo file dovrebbe:

- Descrivi quando e come usare l'abilità.
- Fornisci procedure di indagine dettagliate.
- Includi alberi decisionali per diversi scenari.
- Documenta i risultati attesi e i criteri di successo.

Parte introduttiva

Frontmatter è il blocco di metadati nella parte superiore di un `SKILL.md` file, racchiuso tra delimitatori. `---` Contiene i `description` campi `name` e utilizzati dall' AWS DevOps Agente per determinare quando attivare l'abilità durante un'indagine o un'attività.

```
---
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.
---
```

nome: un identificatore univoco per l'abilità. Utilizza solo lettere minuscole, numeri e trattini (massimo 64 caratteri). Non deve iniziare o terminare con un trattino.

descrizione — Una spiegazione dettagliata di quando e perché AWS DevOps l'Agente dovrebbe usare questa Skill. AWS DevOps L'agente valuta questo campo per decidere se l'abilità è rilevante per l'attività corrente. Una descrizione vaga o mancante può far sì che l'agente salti completamente l'abilità, anche se le istruzioni sono scritte bene.

Importante: scrivi la descrizione dal punto di vista dell'agente. Includi gli scenari, i servizi, i tipi di errore o i sintomi specifici che dovrebbero attivare la Skill. Ad esempio, «Usa questa abilità per analizzare la latenza del database, gli errori di connessione o i timeout delle query per le istanze Amazon RDS» è più efficace di «abilità RDS».

Quando crei una Skill nell'interfaccia utente, il sistema genera automaticamente una parte introduttiva a partire dal nome e dalla descrizione che fornisci. Le abilità caricate come file zip devono includere il frontespizio nel file. `SKILL.md`

Esempio: abilità completa

L'esempio seguente mostra un'abilità completa e ben formata per l'analisi dei problemi di prestazioni RDS. Illustra la struttura delle cartelle, la parte introduttiva SKILL.md , le procedure di indagine utilizzabili e un file di riferimenti supplementare.

Struttura delle directory:

```
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% |
```

Esempio: abilità di filtraggio degli incidenti

Le competenze mirate al tipo di agente Incident Triage possono definire criteri per saltare automaticamente gli incidenti. Utilizzalo per filtrare gli incidenti che non richiedono indagini. Quando un nuovo incidente soddisfa i criteri di ignoramento, l' AWS DevOps agente lo contrassegna come ignorato. Il sistema fornisce un motivo che spiega perché è stato filtrato.

L'esempio seguente mostra un'abilità che ignora gli incidenti a bassa priorità durante la manutenzione programmata:

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.
```

Quando crei questa abilità, seleziona Incident Triage come tipo di agente. Ciò garantisce che l'abilità venga valutata solo durante la fase di triage.

Creazione di competenze

Prima di creare abilità, devi disporre di un Agent Space. Per ulteriori informazioni, consulta the section called "Creazione di uno spazio per agenti".

È possibile creare competenze in due modi, a seconda delle preferenze del flusso di lavoro e della complessità delle competenze:

Creazione di un'abilità nell'interfaccia utente

Le competenze create nell'app Web AWS DevOps Agent Operator contengono un nome, una descrizione e le istruzioni in un unico SKILL.md file.

Per creare un'abilità nell'interfaccia utente:

- Vai alla pagina Competenze nella tua app Web Agent Space Operator.
- Fai clic su «Aggiungi abilità».
- Seleziona «Crea abilità» dalla modalità modale.
- Compila il modulo di abilità:
  - Nome: solo lettere minuscole, numeri e trattini (massimo 64 caratteri). Non deve iniziare o terminare con un trattino. Ad esempio: `rds-throttling-investigation`
  - Descrizione: breve spiegazione di quando utilizzare questa abilità (minimo 100 caratteri consigliati, massimo 1.024 caratteri). Questo aiuta l'agente a determinare quando attivare l'abilità.
  - Stato: impostato su Attivo (impostazione predefinita) o Inattivo. Le abilità inattive non vengono utilizzate dall'agente.
  - Tipo di agente: seleziona uno o più tipi di agenti che possono utilizzare questa abilità. L'opzione Generica è selezionata per impostazione predefinita e rende l'abilità disponibile per tutti i tipi di agenti. Per scegliere come target agenti specifici, deselezionate Generico e scegliete tra: On-demand, Incident Triage, Incident RCA, Incident Mitigation o Evaluation.
  - Istruzioni: Step-by-step procedure in formato Markdown. Sii specifico e fattibile.
- Fai clic su «Crea» per salvare l'abilità.

Il sistema genera automaticamente un SKILL.md file con la struttura del frontespizio corretta.

Per modificare un'abilità creata nell'interfaccia utente:

- Vai all'abilità nell'elenco Competenze e fai clic sull'abilità per aprirla.
- Fare clic su Edit (Modifica).
- Modifica il nome, la descrizione o le istruzioni.
- Fai clic su Salva per aggiornare l'abilità.

Caricamento di un'abilità

Le competenze caricate come file zip contengono un SKILL.md file e risorse aggiuntive come materiali di riferimento o risorse.

Struttura delle abilità:

```
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 requisiti frontali:

Le competenze caricate come file zip devono includere frontespizi e campi. SKILL.md `name` `description` AWS DevOps L'agente utilizza questi campi per determinare quando attivare l'abilità. Per informazioni dettagliate sulla scrittura di frontespizi efficaci, consultate la sezione Frontmatter più avanti in questo argomento.

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

Per creare un'abilità tramite caricamento in formato zip:

- Crea una cartella con i file delle tue abilità seguendo la struttura sopra riportata.
- Assicurati che SKILL.md includa il frontespizio corretto (nome e descrizione).
- Comprimi la directory in un file.zip.
- Vai alla pagina Skills nella tua app Web Agent Space Operator.
- Fai clic su «Aggiungi abilità».
- Seleziona «Carica abilità» dalla modalità modale.
- Trascina e rilascia il file.zip o fai clic per sfogliarlo (solo file ZIP, massimo 6 MB).
- Seleziona uno o più tipi di agenti che possono utilizzare questa abilità (l'opzione Generico è selezionata per impostazione predefinita e si applica a tutti i tipi di agente; deseleziona in particolare Target On-demand, Incident Triage, Incident RCA, Incident Mitigation o Evaluation).
- Esamina i requisiti del file zip e i risultati della convalida.
- Fai clic su «Carica» per aggiungere l'abilità al tuo Agent Space.

Restrizioni importanti per le competenze caricate come file zip:

- Gli script non sono attualmente supportati: le abilità che contengono script nella `scripts/` directory verranno rifiutate durante il caricamento. L'esecuzione degli script verrà abilitata in una versione futura non appena gli agenti avranno accesso a un ambiente di codifica sicuro.
- Limite di dimensione: la dimensione totale del file zip non deve superare i 6 MB (inclusi tutti i file).
- SKILL.md obbligatorio: il file zip deve contenere un SKILL.md file con frontespizio valido.

Le migliori pratiche per le competenze di denominazione:

Utilizza nomi chiari e descrittivi come «rds-throttling-investigation» anziché nomi generici. Un buon nome di abilità riflette lo scenario o il servizio specifico a cui si rivolge, rendendo più facile identificare la competenza giusta a colpo d'occhio.

Gestione delle competenze

AWS DevOps Agent offre funzionalità complete di gestione delle competenze tramite l'app Web Operator:

Elenco delle competenze: visualizza tutte le competenze nel tuo Agent Space. La pagina Competenze mostra il nome della competenza, lo stato Attivo o Inattivo, la data di creazione, la data dell'ultimo aggiornamento e le azioni disponibili.

Competenze di visualizzazione: fai clic su una competenza per visualizzarne i dettagli. Le abilità create nell'interfaccia utente mostrano contenuti modificabili in cui puoi modificare il nome, la descrizione o le istruzioni direttamente nell'interfaccia utente e fare clic su «Salva» per aggiornare. Le abilità caricate come file zip mostrano un albero di file SKILL.md e tutte le cartelle aggiuntive come references/ e assets/. Fate clic sui file nell'albero per visualizzarne il contenuto in modalità di sola lettura.

Selezione degli agenti per un'abilità: configura quali tipi di agenti possono utilizzare ciascuna abilità durante la creazione o la modifica. Nel menu a discesa Tipo di agente, seleziona uno o più tipi di agenti utilizzando le caselle di controllo: Generico (impostazione predefinita, si applica a tutti i tipi di agenti), On-demand(query conversazionali), Incident Triage (valutazione iniziale dell'incidente), Incident RCA (analisi della causa principale), Incident Mitigation (risposta automatizzata agli incidenti) o Valutazione (consigli proattivi). Generico è selezionato per impostazione predefinita e rende la competenza disponibile per tutti i tipi di agenti. Le competenze destinate a agenti specifici riducono il consumo di contesto e migliorano la concentrazione degli agenti.

Attivazione e disattivazione delle abilità: disattiva temporaneamente le abilità senza eliminarle utilizzando l'interruttore. Active/Inactive Apri la visualizzazione dei dettagli delle abilità e imposta l'interruttore su «Inattivo» per impedire all'agente di caricarlo per nuove indagini, preservando al contempo tutti i contenuti e le configurazioni. In-progress le indagini continuano utilizzando l'abilità. Torna a «Attiva» per rendere immediatamente nuovamente disponibile l'abilità.

Aggiornamento delle competenze: modifica le competenze esistenti in base a come sono state create. Per le competenze create nell'interfaccia utente, fai clic su «Modifica» nella visualizzazione dei dettagli delle competenze, modifica il nome, la descrizione o le istruzioni e fai clic su «Salva» per aggiornare. Per le competenze caricate come file zip, modifica i file localmente, crea un nuovo file zip e carica una nuova versione.

Eliminazione delle competenze: rimuovi definitivamente le abilità dal tuo Agent Space. Apri la visualizzazione dell'elenco delle abilità, fai clic sul menu Altre opzioni (⋮) e seleziona «Elimina», leggi l'avviso relativo all'eliminazione permanente, digita il nome dell'abilità per confermare e fai clic su «Elimina abilità». L'eliminazione non può essere annullata. In-progress le indagini potrebbero essere compromesse se tentano di caricare l'abilità eliminata. Per le abilità caricate come file zip, scaricate il file zip prima di eliminarle come backup. Prendi in considerazione la possibilità di disattivare l'abilità invece di eliminarla se potresti averne bisogno di nuovo.

Migrazione da Runbook

I Runbook esistenti vengono migrati automaticamente a Skills senza che sia richiesta alcuna azione da parte del cliente. Quando Agent Space passa al modello Skills, tutti i Runbook vengono convertiti in Skills e vengono visualizzati nell'interfaccia utente Skills. Dopo la migrazione, puoi:

- Rivedi le competenze migrate: verifica che la migrazione automatica abbia convertito correttamente i tuoi Runbook.
- Aggiorna se necessario: modifica le competenze direttamente nell'interfaccia utente per perfezionare le istruzioni, aggiornare le descrizioni o configurare il targeting per tipo di agente.
- Espandi con riferimenti: per le competenze che potrebbero trarre vantaggio da materiali di riferimento o diagrammi di architettura aggiuntivi, ricreale come abilità di caricamento zip con una directory references/ o assets/.
- Crea nuove competenze: aggiungi nuove competenze per flussi di lavoro investigativi non inclusi in precedenza nei Runbooks.

Contatta l' AWS assistenza se riscontri problemi con la migrazione automatica di Skills o hai bisogno di assistenza con gli aggiornamenti successivi alla migrazione.

Competenze apprese

Cosa sono le abilità apprese?

Le competenze acquisite sono file di conoscenza strutturati che l' DevOps agente genera dai dati di Agent Space. Ogni abilità appresa codifica un tipo specifico di conoscenza che l' AWS DevOps agente utilizza per eseguire le attività. Al momento del lancio, sono disponibili due competenze acquisite: Agent Space Understanding e Tool Use Best Practices.

Agent Space Understanding

L'abilità Agent Space Understanding (`understanding-agent-space`) analizza gli account cloud connessi, gli archivi di codice e le integrazioni di telemetria per creare una mappa delle risorse e delle relazioni in un Agent Space.

L'abilità produce un `SKILL.md` file principale e un set di file di riferimento. Il file principale contiene una panoramica del sistema in linguaggio semplice con i concetti chiave del dominio, gli ambienti di distribuzione (coppie di AWS account e regioni, sottoscrizioni e aree di Azure e così via), un diagramma di architettura a livello di contenitore che mostra come i servizi logici si connettono, i

percorsi di richiesta che sono fondamentali per l'applicazione con i componenti che attraversano e una mappatura degli archivi di codice ai contenitori.

Ogni contenitore logico riceve un file di riferimento dedicato che descrive i suoi componenti interni (elaborazione, dati, messaggistica, rete e altri) con tipi di risorse e identificatori fisici come ARNs nomi di tabelle e coda URLs. Il file di riferimento acquisisce anche la copertura dell'osservabilità, inclusi gli allarmi, i dashboard e i monitor collegati a ciascun componente. Inoltre, associa ogni componente agli archivi di codice, ai pacchetti e alle infrastructure-as-code definizioni associati, fornendo una catena di tracciabilità completa dal codice sorgente alle risorse distribuite.

Ogni percorso di richiesta critico riceve un file di riferimento dedicato che descrive l'intero flusso di end-to-end richieste in base alla granularità dei componenti, dal punto di ingresso fino a ogni servizio intermedio, archivio dati e dipendenza esterna. Il file include un diagramma di flusso in sequenza che mostra l'ordine delle operazioni e i meccanismi di interazione tra i componenti, insieme alla responsabilità di ciascun partecipante. Inoltre, cataloga i segnali di osservabilità pertinenti al percorso: modelli di gruppi di log per ogni hop, metriche chiave (latenza, tassi di errore, throttling, quote di token) con i relativi nomi e dimensioni degli allarmi e intervalli di traccia distribuiti che possono essere correlati tra servizi e account.

Best practice per l'uso degli strumenti

L'abilità Tool Use Best Practices analizza gli usi passati degli strumenti di indagine per estrarre modelli di utilizzo efficaci, modalità di errore comuni e linee guida sui parametri. Questo aiuta l' DevOps agente a evitare insidie note e a condurre indagini con meno passaggi inutili. L'abilità produce un file principale e una serie di file di riferimento per strumento. Il file principale funge da indice di routing che elenca ogni strumento con gli scenari di indagine supportati e i collegamenti al file di riferimento corrispondente.

Ogni file di riferimento per strumento può includere fino a tre sezioni:

- Best practice: tecniche basate sull'indagine estratte dall'utilizzo efficace degli strumenti, come i modelli di query di CloudWatch Logs Insights, i namespace e le dimensioni delle metriche specifici dell'ambiente e i filtri di origine degli eventi. CloudTrail Ogni voce è organizzata in base a uno scenario di indagine e include valori di parametri concreti ed esempi osservati nelle indagini precedenti.
- Errori comuni: modalità di errore ricorrenti e relative correzioni. Ogni voce descrive una condizione di errore specifica, ad esempio l'interrogazione di un account inaccessibile o la creazione di una query di aggregazione non valida, e fornisce un'azione correttiva in modo che l'agente possa evitare o ripristinare l'errore senza sprecare le fasi di indagine.

- Gestione dell'output: linee guida per le chiamate agli strumenti che tendono a restituire risposte di grandi dimensioni. Ogni voce descrive una modifica dei parametri o una strategia di elaborazione che riduce le dimensioni dell'output preservando al contempo il valore diagnostico.

Quando è disponibile l'accesso in tempo reale all'infrastruttura, la skill convalida i modelli rispetto all'ambiente in uso prima di includerli. I modelli confermati vengono dichiarati con sicurezza, i modelli non confermati utilizzano un linguaggio cauto e i modelli smentiti sono esclusi. Ciò mantiene le competenze in linea con lo stato attuale dell'infrastruttura.

Gestione delle competenze acquisite

Aggiornamenti: l' DevOps agente genera e aggiorna automaticamente le competenze acquisite in base all'attività svolta nell'Agent Space. Di seguito viene descritto quando ogni abilità viene aggiornata.

L' DevOps agente genera un'abilità Tool Use Best Practices aggiornata ogni 30 indagini.

La skill Agent Space Understanding viene generata dal learning agent, che viene eseguita ogni volta che aggiungi, aggiorni o rimuovi una funzionalità o un'integrazione di Agent Space.

Per rigenerare manualmente le abilità apprese, scegli il pulsante Rigenera nella pagina Topologia dell'app dell'operatore oppure chatta con l'agente e chiedigli di aggiornare le competenze apprese.

Disattivazione: le abilità apprese sono attive per impostazione predefinita. Quando sono attive, l' DevOps agente le carica all'inizio di ogni attività dell' DevOps agente. Per impedire che un'abilità appresa venga applicata, disattivala dal visualizzatore delle abilità nell'app dell'operatore. La disattivazione di un'abilità non la elimina. L'abilità viene mantenuta e può essere riattivata in qualsiasi momento. Quando un'abilità viene disattivata, l' DevOps Agente opera all'insaputa dell'abilità.

Vista topologica: la pagina Topologia dell'app Web di Agent Space utilizza l'Agent Space Understanding Skill per visualizzare visivamente l'ambiente Agent Space come contenitori e componenti logici. Fai clic su qualsiasi contenitore per visualizzarne i componenti, gli identificatori di risorse e la telemetria.

Istruzioni per l'agente

Utilizza le istruzioni dell'agente per fornire una guida sempre attiva che AWS DevOps l'agente applica a ogni sessione. Una sessione è una singola conversazione o indagine con un agente. Nella pagina Agenti della tua app Web Agent Space Operator, puoi impostare istruzioni globali che si applicano a

tutti gli agenti o impostare istruzioni per uno specifico agente gestito, come Chat o Incident Triage. Queste istruzioni vengono archiviate come AGENTS.md file. A differenza the section called "DevOps Competenze degli agenti" di quelle che vengono caricate su richiesta quando l'agente abbina una descrizione delle competenze all'attività corrente, le istruzioni dell'agente sono sempre presenti dall'inizio di ogni sessione, indipendentemente da ciò su cui l'agente sta lavorando.

Cosa sono le istruzioni per l'agente

Le istruzioni per gli agenti forniscono agli agenti una guida incondizionata e sempre attiva. All'inizio di ogni sessione, il servizio agente recupera le istruzioni configurate per Agent Space e ne inserisce il contenuto direttamente nel prompt del sistema dell'agente. L'agente non decide se caricarle; sono sempre presenti.

Ogni sessione dell'agente riceve istruzioni sia dalle istruzioni globali che dalle istruzioni specifiche dell'agente pertinenti, ad esempio Chat.

Le istruzioni dell'agente vengono archiviate come AGENTS.md file e differiscono tra loro the section called "DevOps Competenze degli agenti" in diversi modi importanti:

Aspetto	Abilità	Istruzioni per l'agente (AGENTS.md)
Nome e descrizione	Richiesto	Non applicabile
Formato del contenuto	Pacchetto Markdown o ZIP	Solo Markdown
File di risorse	Supportata	Non supportata
Iniezione di contesto	Su richiesta (l'agente decide tramite l'abbinamento della descrizione delle competenze)	Sempre (incondizionato, ogni sessione)
Univocità	Spazio multiplo per agente	Uno per agente (uno per le istruzioni globali, uno per agente gestito)

Le istruzioni dell'agente non hanno un nome o un campo di descrizione. Il AGENTS.md file sottostante contiene solo markdown senza frontespizio, senza supporto per pacchetti ZIP e senza file di risorse.

Perché usare le istruzioni per gli agenti

Le istruzioni per gli agenti offrono un modo affidabile per garantire che determinate indicazioni siano sempre contestualizzate, senza dipendere dalle decisioni che comportano un carico di competenze dell'agente.

Principali vantaggi:

- Prevedibilità: le istruzioni sono sempre presenti, indipendentemente dall'attività su cui sta lavorando l'agente. Non è richiesta alcuna corrispondenza tra le descrizioni e l'agente non può ignorare il contenuto.
- Copertura garantita: a differenza di Skills, che l'agente può caricare o meno a seconda della rilevanza dell'attività, le istruzioni dell'agente vengono sempre fornite all'inizio di ogni sessione.
- Politiche permanenti: utilizza le istruzioni degli agenti per le politiche operative permanenti, le linee guida di sicurezza, gli standard di codifica o qualsiasi altra guida che debba essere applicata a ogni sessione senza eccezioni.
- Ambito mirato: è possibile applicare le istruzioni a tutti i tipi di agente contemporaneamente utilizzando istruzioni globali o limitare le istruzioni a un tipo specifico di agente quando le linee guida riguardano solo il lavoro di quell'agente.

Come funzionano le istruzioni per gli agenti

All'avvio di una sessione, il servizio dell'agente recupera le istruzioni configurate per Agent Space e ne inserisce il contenuto nel prompt del sistema dell'agente prima dell'inizio della sessione. Ciò avviene automaticamente per ogni sessione. L'agente non valuta se caricarli; inietta sempre il contenuto.

Ogni nuova sessione carica le istruzioni fresche all'avvio. Se aggiorni le istruzioni, la modifica ha effetto immediato per le sessioni che iniziano dopo il salvataggio. Le sessioni già in corso continuano a utilizzare il contenuto caricato al momento dell'avvio.

L'ambito determina le istruzioni ricevute da una sessione. Le istruzioni globali si applicano a tutti i tipi di agenti presenti nell'Agent Space, quindi ogni sessione le riceve. Agent-specific le istruzioni si applicano solo alle sessioni di quel particolare tipo di agente. Una sessione riceve istruzioni sia dalle istruzioni globali che dalle istruzioni pertinenti specifiche dell'agente.

Ambito del tipo di agente

L'ambito controlla quali sessioni di agenti ricevono un determinato set di istruzioni. Sono disponibili due opzioni di ambito:

- Istruzioni globali: si applicano a tutti i tipi di agenti presenti nell'Agent Space. Ogni sessione dell'agente riceve questo contenuto.
- Agent-specific: si applica solo alle sessioni del tipo di agente selezionato.

Gli agenti gestiti disponibili per istruzioni specifiche per agente sono:

- Chat: Ad-hoc domande e richieste durante le sessioni di chat.
- Valutazione degli incidenti: filtraggio degli allarmi, classificazione della gravità e definizione iniziale dell'ambito.
- RCA degli incidenti: analisi delle cause principali con raccolta e convalida delle prove.
- Attenuazione degli incidenti: raccomandazioni per la correzione e Short-term la correzione a lungo termine.
- Valutazione: valutazione delle prestazioni degli agenti e controlli di conformità alle politiche.

Indicazioni sulla dimensione dei contenuti

Ogni volta che inizi una conversazione o un'indagine, l'agente legge tutte le tue istruzioni prima di fare qualsiasi altra cosa. L'agente dispone di una quantità fissa di memoria di lavoro per sessione e le istruzioni dell'utente ne utilizzano una parte. Più grande è il file, meno spazio rimane per le domande, le indagini, i registri letti dall'agente e il suo ragionamento. Istruzioni più brevi e mirate offrono all'agente una maggiore capacità di risolvere il problema.

- Limite rigido: 25 KB
- Dimensione consigliata: 120 righe (consigliata per la maggior parte delle configurazioni)

Concentra le tue istruzioni sulla guida che deve essere presente in ogni sessione. Per le procedure di indagine specializzate che si applicano solo a compiti specifici, considera the section called "DevOps Competenze degli agenti" invece l'utilizzo.

Esempio

L'esempio seguente mostra istruzioni ben formate per gli agenti con linee guida per le indagini, standard di formattazione delle risposte e requisiti di sicurezza che si applicano a ogni sessione dell'agente.

```
# 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.
```

Impostazione delle istruzioni per l'agente

Prima di impostare le istruzioni per l'agente, è necessario disporre di un Agent Space. Per ulteriori informazioni, consulta the section called “Creazione di uno spazio per agenti”.

Ogni agente ha esattamente un set di istruzioni. Quando si salvano nuovi contenuti, questi sovrascrivono i contenuti esistenti per quell'agente.

Per impostare istruzioni globali (si applica a tutti gli agenti):

1. Vai alla pagina Agenti nella tua app Web Agent Space Operator.
2. Scegli Visualizza accanto alle istruzioni globali.
3. Inserisci le istruzioni di markdown nell'editor.
4. Scegli Save (Salva).

Per impostare le istruzioni per un agente specifico:

1. Vai alla pagina Agenti nella tua app Web Agent Space Operator.

2. In Agenti gestiti, scegli Visualizza accanto all'agente che desideri configurare: Chat, Incident triage, Incident RCA, Incident mitigation o Evaluation.
3. Inserisci le istruzioni di markdown nell'editor.
4. Scegli Save (Salva).

Istruzioni per gli agenti di gestione

AWS DevOps L'agente fornisce funzionalità di gestione per le istruzioni degli agenti tramite l'Operator Web App.

Visualizzazione delle istruzioni: accedi alla pagina Agenti e scegli Visualizza accanto a Istruzioni globali o all'agente gestito specifico. L'editor mostra il contenuto corrente. Usa la scheda Anteprima per vedere il markdown renderizzato o la scheda Codice per vedere il markdown non elaborato.

Istruzioni di modifica: apri l'agente come descritto sopra, modifica il contenuto nell'editor e scegli Salva.

Istruzioni per il caricamento da un file: apri l'agente, quindi scegli il pulsante Carica nell'editor per caricare un file markdown dal tuo computer.

Istruzioni per il download: apri l'agente, quindi scegli il pulsante Download nell'editor per scaricare il contenuto corrente come file.

Istruzioni per l'eliminazione: apri l'agente, scegli il pulsante Elimina nell'editor e conferma l'eliminazione. Questa operazione non può essere annullata. Prendi in considerazione la possibilità di scaricare prima il contenuto se potresti averne bisogno di nuovo.

Regioni supportate

Questo argomento descrive le AWS regioni in cui è possibile utilizzare AWS DevOps Agent. Per ulteriori informazioni sulle AWS regioni, consulta Specificare AWS le regioni che il tuo account può utilizzare nella Guida di riferimento per la gestione degli AWS account.

Monitoraggio delle risorse in più regioni

AWS DevOps L'agente può monitorare e analizzare le risorse AWS degli account ubicati in qualsiasi AWS regione, indipendentemente dalla regione supportata in cui viene creato lo spazio per agenti. Quando associ un AWS account a un Agent Space, l'agente scopre e mappa le risorse in tutte le

regioni all'interno di quell'account. Ciò significa che non è necessario un Agent Space in ogni regione in cui vengono eseguiti i carichi di lavoro.

Scegli una regione supportata in base alla residenza dei dati preferita, alla vicinanza al team operativo o ai requisiti organizzativi.

Regioni supportate

AWS DevOps L'agente è disponibile nelle seguenti AWS regioni.

Nome della regione	Codice regione	Collegamento alla console
Stati Uniti orientali (Virginia settentrionale)	us-east-1	Console aperta
Stati Uniti occidentali (Oregon)	us-west-2	Console aperta
Asia Pacifico (Sydney)	ap-southeast-2	Console aperta
Asia Pacifico (Tokyo)	ap-northeast-1	Console aperta
Europa (Francoforte)	eu-central-1	Console aperta
Europa (Irlanda)	eu-west-1	Console aperta

Endpoint del servizio

Nome della regione	Codice regione	Endpoint	Protocollo
Stati Uniti orientali (Virginia settentrionale)	us-east-1	aidevops.us-east-1.amazonaws.com	HTTPS
Stati Uniti occidentali (Oregon)	us-west-2	aidevops.us-west-2.amazonaws.com	HTTPS
Asia Pacifico (Sydney)	ap-southeast-2	aidevops.ap-southeast-2.amazonaws.com	HTTPS

Nome della regione	Codice regione	Endpoint	Protocollo
Asia Pacifico (Tokyo)	ap-northeast-1	aidevops.ap-northeast-1.amazonaws.com	HTTPS
Europa (Francoforte)	eu-central-1	aidevops.eu-central-1.amazonaws.com	HTTPS
Europa (Irlanda)	eu-west-1	aidevops.eu-west-1.amazonaws.com	HTTPS

Considerazioni

- Selezione della regione di Agent Space: un Agent Space e i relativi dati (indagini,

topologia, consigli) vengono archiviati nella regione in cui vengono creati. Scegli una regione che soddisfi i requisiti di residenza dei dati.

- Monitoraggio interregionale: risorse disponibili negli AWS account associati a un agente

Lo spazio viene monitorato indipendentemente dalla regione in cui vengono distribuite le risorse. Non è necessario creare Agent Spaces separati in ogni regione in cui vengono eseguiti i carichi di lavoro.

- Integrazioni di terze parti: connessioni ai CI/CD provider (GitHub,), GitLab

gli strumenti di osservabilità (Dynatrace, Datadog, New Relic, Splunk) e i server MCP sono configurati per Agent Space e non dipendono dalla regione.

Guida introduttiva a AWS DevOps Agent

In questa guida introduttiva, creerai un Agent Space di base, configurerai autorizzazioni minime e condurrai la tua prima indagine basata sull'intelligenza artificiale.

Argomenti:

Creazione di uno spazio per agenti

Un Agent Space definisce gli strumenti e l'infrastruttura a cui l' AWS DevOps agente ha accesso. Questa guida illustra come creare un Agent Space, configurare l'accesso all'account principale e abilitare l' DevOps Agent Web App. Vedi «Cos'è un Agent Space» per saperne di più sul concetto di Agent Space.

Creazione di un Agent Space

Accedi alla console AWS DevOps dell'agente

1. Accedi alla console AWS di gestione
2. Vai alla console dell' AWS DevOps agente

Assegna un nome all'Agent Space

1. Fai clic su Crea spazio agente

Nella sezione dei dettagli di Agent Space, fornisci:

1. Nel campo Nome, inserisci un nome per il tuo Agent Space

2. (Facoltativo) Nel campo Descrizione, aggiungi dettagli sullo scopo di Agent Space
3. (Facoltativo) Dal menu a discesa Lingua di risposta dell'agente, seleziona la lingua utilizzata dall'agente per generare risposte, risultati e risultati di indagine. Le opzioni includono: indonesiano, cinese (Simplified/PRC), Chinese (Traditional/Taiwan), inglese (Regno Unito), francese (Francia), tedesco (Germania), italiano (Italia), giapponese (Giappone), coreano (Corea), portoghese (Brasile), spagnolo (America Latina), turco (Turchia), arabo (Arabia Saudita), tailandese (Thailandia) e vietnamita (Vietnam). Se non è selezionata alcuna lingua, l'agente risponde nella lingua di input. Questa impostazione viene utilizzata anche per determinare la lingua per i casi di AWS supporto creati tramite la funzione Ask for human support.

Configurazione dell'accesso all'account principale

Nella sezione Concedi l'accesso alle AWS risorse di Agent Space, imposterai un ruolo IAM per concedere a Agent Space l'accesso all' AWS account principale. L'account principale è l' AWS account in cui crei il tuo Agent Space. AWS DevOps L'agente richiede un ruolo IAM per scoprire e accedere alle AWS risorse di questo account durante le indagini.

Scegli un metodo di configurazione del ruolo. Seleziona una delle seguenti opzioni:

Opzione 1: creazione automatica di un nuovo ruolo di AWS DevOps agente (consigliato)

Questa opzione crea automaticamente un ruolo con le autorizzazioni appropriate per consentire all' AWS DevOps agente di esaminare le risorse presenti nell'account.

> **Note**
>
> È necessario disporre delle autorizzazioni IAM per creare nuovi ruoli per utilizzare questa opzione.

1. Seleziona Crea automaticamente un nuovo ruolo di agente AWS DevOps
2. (Facoltativo) Aggiorna il nome del ruolo Agent Space da creare

Opzione 2: assegnare un ruolo esistente

Utilizzate questa opzione quando un altro amministratore ha precedentemente creato un ruolo specifico per AWS DevOps Agente.

1. Seleziona Assegna un ruolo esistente
2. Dal menu a discesa, seleziona un ruolo esistente con le autorizzazioni appropriate

Opzione 3: creare un nuovo ruolo di AWS DevOps agente utilizzando un modello di policy

Utilizza questa opzione quando devi limitare i servizi e le risorse a cui l'agente può accedere nell'account principale.

1. Seleziona Crea un nuovo ruolo di AWS DevOps agente utilizzando un modello di policy
2. Segui le istruzioni per creare la politica di fiducia e la politica in linea del nuovo ruolo.

Abilitazione dell'app Web Agent Space

L'app Web è il luogo in cui il personale interagisce con AWS DevOps l'agente per le indagini sugli incidenti e la revisione delle raccomandazioni. Per ulteriori informazioni, consulta AWS DevOps Agent Console Architecture [link]. Se abilitata, gli utenti possono accedere all'app Web Agent Space tramite un link di autenticazione IAM dalla console di AWS gestione.

Seleziona una delle seguenti opzioni:

Opzione 1: creazione automatica di un nuovo ruolo di AWS DevOps agente (consigliato)

Questa opzione crea automaticamente un ruolo con le autorizzazioni appropriate per l'accesso all' DevOps Agent Web App.

> **Note**
>
> È necessario disporre delle autorizzazioni IAM per creare nuovi ruoli per utilizzare questa opzione.

1. Seleziona Crea automaticamente un nuovo ruolo di agente AWS DevOps
2. Rivedi le autorizzazioni che verranno concesse al ruolo

Opzione 2: assegna un ruolo esistente

Utilizzate questa opzione quando un altro amministratore ha precedentemente creato un ruolo operatore.

1. Seleziona Assegna un ruolo esistente
2. Dal menu a discesa, seleziona un ruolo esistente con le autorizzazioni appropriate

Opzione 3: creare un nuovo ruolo di AWS DevOps agente utilizzando un modello di policy

Utilizza questa opzione quando devi personalizzare le autorizzazioni per l'accesso alle app Web.

1. Seleziona Crea un nuovo ruolo di AWS DevOps agente utilizzando un modello di policy
2. Segui le istruzioni per creare la politica di fiducia e la politica in linea del nuovo ruolo.

Aggiungere tag (opzionale)

Puoi aggiungere AWS tag al tuo Agent Space durante la creazione. I tag sono coppie chiave-valore che consentono di organizzare e identificare le risorse. Puoi aggiungere fino a 50 tag per Agent Space. Per aggiungere tag, espandi la sezione Tag nella pagina Create Agent Space e fai clic su Aggiungi nuovo tag.

Completa la creazione dello spazio per gli agenti

Una volta compilate tutte le sezioni, fai clic su Crea

Verifica della configurazione di Agent Space

Una volta configurato, il pulsante di accesso dell'operatore verrà visualizzato nella pagina dei dettagli di Agent Space. Facendo clic su di esso si aprirà l'app Web in una nuova scheda e l'autenticazione verrà eseguita correttamente.

Fasi successive

Dopo aver configurato Agent Space, considera questi passaggi successivi:

- Aggiungi account secondari se le tue applicazioni si estendono su più AWS account
- Configura integrazioni di terze parti come strumenti di osservabilità o sistemi di ticketing
- Configura l'autenticazione AWS Identity Center per gli ambienti di produzione
- Esplora la mappatura delle risorse applicative per aiutare AWS DevOps Agent a comprendere la tua infrastruttura

AWS DevOps Guida all'onboarding CLI per agenti

Panoramica di

Con AWS DevOps Agent, puoi monitorare e gestire la tua infrastruttura. AWS Questa guida illustra come configurare l' AWS DevOps agente utilizzando l'interfaccia a riga di AWS comando (AWS CLI). Puoi creare ruoli IAM, configurare uno spazio per agenti e associare il tuo AWS account. Puoi anche abilitare l'app dell'operatore e, facoltativamente, collegare integrazioni di terze parti. Il completamento di questa guida richiede circa 20 minuti.

AWS DevOps L'agente è disponibile in sei AWS regioni: Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon), Asia Pacifico (Sydney), Asia Pacifico (Tokyo), Europa (Francoforte) ed Europa (Irlanda). Per ulteriori informazioni sulle regioni supportate, consulta. the section called "Regioni supportate"

Prerequisiti

Prima di iniziare, assicurati di disporre di:

- AWS CLI versione 2 installata e configurata
- Autenticazione al tuo account AWS di monitoraggio
- Autorizzazioni per creare ruoli AWS Identity and Access Management (IAM) e allegare policy
- Un AWS account da utilizzare come account di monitoraggio
- Familiarità con la AWS CLI e la sintassi JSON

In questa guida, sostituisci i seguenti valori segnaposto con i tuoi:

- `<MONITORING_ACCOUNT_ID>`— L'ID dell'account a 12 cifre per l' AWS account di monitoraggio (principale)
- `<EXTERNAL_ACCOUNT_ID>`— L'ID dell'account a 12 cifre dell' AWS account secondario da monitorare (utilizzato nella fase 4)
- `<REGION>`— Il codice AWS regionale dello spazio riservato all'agente (ad esempio, `us-east-1` o) `eu-central-1`
- `<AGENT_SPACE_ID>`— L'identificatore dello spazio dell'agente restituito dal comando `create-agent-space`

Configurazione dei ruoli IAM

1. Crea il ruolo DevOps Agent Space

Crea la policy di fiducia IAM eseguendo il seguente comando:

```
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
```

Crea il ruolo IAM:

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

Salvate il ruolo ARN eseguendo il seguente comando:

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

Allega la policy AWS gestita:

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

Crea e allega una policy in linea per consentire la creazione del ruolo collegato al servizio Resource Explorer:

```
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. Crea il ruolo IAM dell'app operatore

Crea la policy di fiducia IAM eseguendo il seguente comando:

```
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
```

Crea il ruolo IAM:

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

Salvate il ruolo ARN eseguendo il seguente comando:

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

Allega la policy dell'app per operatori AWS gestiti:

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

Questa politica gestita concede all'app dell'operatore le autorizzazioni per accedere alle funzionalità dello spazio agente. Queste funzionalità includono indagini, consigli, gestione delle conoscenze,

chat e integrazione con AWS Support. La policy prevede l'accesso allo spazio specifico dell'agente utilizzando la `aws:PrincipalTag/AgentSpaceId` condizione. Per ulteriori informazioni sull'elenco completo delle azioni, vederethe section called "DevOps Autorizzazioni Agent IAM".

Fasi di onboarding

1. Crea uno spazio per agenti

Eseguite il seguente comando per creare uno spazio agente:

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

Facoltativamente, specificare `--kms-key-arn` di utilizzare una chiave AWS KMS gestita dal cliente per la crittografia. Puoi anche utilizzarla `--tags` per aggiungere tag di risorsa e `--locale` impostare la lingua per le risposte degli agenti.

Salva il file `agentSpaceId` dalla risposta (che si trova in`agentSpace.agentSpaceId`).

Per elencare gli spazi degli agenti in un secondo momento, esegui il seguente comando:

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

2. Associa il tuo AWS account

Associa il tuo AWS account per attivare l'individuazione della topologia. `accountType`Imposta uno dei seguenti valori:

- `monitor`— L'account principale in cui esiste lo spazio dell'agente. Questo account ospita l'agente e viene utilizzato per il rilevamento della topologia.
- `source`— Un account aggiuntivo monitorato dall'agente. Utilizza questo tipo quando associ account esterni nel passaggio 4.

```
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. Abilita l'app dell'operatore

I flussi di autenticazione possono utilizzare IAM, IAM Identity Center (IDC) o un provider di identità esterno (IdP). Esegui il comando seguente per abilitare l'app dell'operatore per lo spazio del tuo agente:

```
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>
```

Per l'autenticazione IAM Identity Center, usa `--auth-flow idc` e fornisci`--idc-instance-arn`. Per un provider di identità esterno, utilizza `--auth-flow idp` e fornisci `--issuer-url` e`--idp-client-secret`. `--idp-client-id` Per ulteriori informazioni, consultare the section called "Configurazione dell'autenticazione IAM Identity Center" e the section called "Configurazione dell'autenticazione tramite provider di identità esterno (IdP)".

> Nota: se in precedenza hai creato un ruolo dell'app operatore per un altro spazio agente nel tuo account, puoi riutilizzare l'ARN di quel ruolo.

4. (Facoltativo) Associa account di origine aggiuntivi

Per monitorare account aggiuntivi con AWS DevOps Agent, crea un ruolo IAM tra account.

Crea il ruolo tra account diversi nell'account esterno

Passa all'account esterno e crea la politica di fiducia. `MONITORING_ACCOUNT_ID`È l'account principale che ospita lo spazio agente configurato nel passaggio 2. Questa configurazione consente

al servizio AWS DevOps Agent di assumere un ruolo negli account di origine secondari per conto dell'account di monitoraggio.

Esegui il comando seguente per creare la politica di fiducia:

```
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
```

Crea il ruolo IAM tra account:

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

Salvate il ruolo ARN eseguendo il seguente comando:

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

Allega la policy AWS gestita:

```
aws iam attach-role-policy \
  --role-name DevOpsAgentCrossAccountRole \
```

```
  --policy-arn arn:aws:iam::aws:policy/AIDevOpsAgentAccessPolicy
```

Allega la policy in linea per consentire la creazione del ruolo collegato al servizio Resource Explorer nell'account esterno:

```
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
```

Associa l'account esterno

Torna al tuo account di monitoraggio, quindi esegui il seguente comando per associare l'account esterno:

```
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. (Facoltativo) Associa GitHub

> Nota: è necessario prima registrarsi GitHub tramite la console dell' AWS DevOps agente utilizzando il OAuth flusso prima di poterlo associare tramite la CLI.

Per istruzioni sulla registrazione GitHub tramite la console, consulta. the section called "Connessione alle CI/CD tubazioni"

Elenca i servizi registrati:

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

Salva il <SERVICE_ID> per ServiceType:. `github`

Dopo esserti registrato GitHub nella console, associa i GitHub repository eseguendo il seguente comando:

```
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. (Facoltativo) Registrazione e associazione ServiceNow

Innanzitutto, registra il ServiceNow servizio con OAuth le credenziali:

```
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>
```

Salva il file restituito<SERVICE_ID>, quindi associa 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. (Facoltativo) Registrare e associare Dynatrace

Innanzitutto, registra il servizio Dynatrace con le credenziali: 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>
```

Salva il file restituito<SERVICE_ID>, quindi associa Dynatrace. Le risorse sono facoltative. L'ambiente specifica a quale ambiente Dynatrace associarsi.

```
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>
```

La risposta include informazioni sui webhook per l'integrazione. Puoi utilizzare questo webhook per avviare un'indagine da parte di Dynatrace. Per ulteriori informazioni, consulta the section called "Connessione di Dynatrace".

8. (Facoltativo) Registrati e associa Splunk

Innanzitutto, registra il servizio Splunk con BearerToken le credenziali.

L'endpoint utilizza il seguente formato: `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>
```

Salva il file restituito`<SERVICE_ID>`, quindi associa 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>
```

La risposta include informazioni sui webhook per l'integrazione. Puoi utilizzare questo webhook per avviare un'indagine da Splunk. Per ulteriori informazioni, consulta the section called “Connessione a Splunk”.

9. (Facoltativo) Registrati e associa New Relic

Innanzitutto, registrate il servizio New Relic con le credenziali della chiave API.

Regione: o. `US` `EU`

Campi opzionali:`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>
```

Salva il risultato restituito<SERVICE_ID>, quindi associa 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>
```

La risposta include informazioni sui webhook per l'integrazione. Puoi utilizzare questo webhook per avviare un'indagine da parte di New Relic. Per ulteriori informazioni, consulta the section called "Collegamento di New Relic".

10. (Facoltativo) Registrati e associa Datadog

È necessario innanzitutto registrare Datadog tramite la console dell' AWS DevOps agente utilizzando il OAuth flusso prima di poterlo associare tramite la CLI. Per ulteriori informazioni, consulta the section called "Connessione DataDog".

Elenca i servizi registrati:

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

Salva il <SERVICE_ID> per ServiceType:. mcpserverdatadog

Quindi associa Datadog:

```
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>
```

La risposta include informazioni sui webhook per l'integrazione. Puoi utilizzare questo webhook per avviare un'indagine da Datadog. Per ulteriori informazioni, consulta the section called "Connessione DataDog".

11. (Facoltativo) Eliminare uno spazio per agenti

L'eliminazione di uno spazio agente rimuove tutte le associazioni, le configurazioni e i dati di indagine relativi a tale spazio agente. Questa azione non può essere annullata.

Per eliminare uno spazio agente, esegui il seguente comando:

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

Verifica

Per verificare la configurazione, esegui i seguenti comandi:

```
# 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>
```

Fasi successive

- Per connettere integrazioni aggiuntive, consultaConfigurazione delle funzionalità per AWS DevOps Agente.

- Per ulteriori informazioni sulle competenze e le funzionalità degli agenti, consultathe section called "DevOps Competenze degli agenti".
- Per comprendere l'app web dell'operatore, consultathe section called "Cos'è un'app Web per DevOps agenti?".

Note

- Sostituisci `<AGENT_SPACE_ID><MONITORING_ACCOUNT_ID>,<EXTERNAL_ACCOUNT_ID>,<REGION>`, e così via con i tuoi valori effettivi.
- Per un elenco delle regioni supportate, consulta the section called "Regioni supportate".

Creazione di un ambiente di test

Questa guida fornisce test pratici per convalidare la funzionalità di risposta agli incidenti di AWS DevOps Agent utilizzando un'architettura di esempio. Utilizzate questo supplemento se desiderate testare DevOps Agent prima di connettere i vostri sistemi di produzione.

Prerequisiti

- AWS account con accesso amministrativo
- AWS DevOps Agent Space creato e configurato utilizzando il flusso di ruoli Auto create DevOps Agent
- Per il test EC2: un VPC esistente con almeno una sottorete nella regione in cui effettuerai la distribuzione.

Panoramica dei costi e della sicurezza

Protezione dei costi

- Test EC2: GRATUITO (livello AWS gratuito) o ~$0,02 per 2 ore
- Test Lambda: GRATUITO (livello gratuito di 1 milione requests/month)
- CloudWatch: GRATUITO (10 allarmi, metriche di base incluse)
- Costo totale stimato previsto: 0,00 USD - 0,05 USD per il test completo

Caratteristiche di sicurezza in questi test

- Auto-termination: Built-in spegnimento automatico
- Idoneo al piano gratuito: utilizza i tipi di istanze più piccoli
- Ambito limitato: risorse di test minime e isolate
- Pulizia semplice: semplici passaggi da console per rimuovere tutto
- Nessun impatto sulla produzione: ambiente di test completamente separato

Configura il tuo AWS account per i test

> **Important**
>
> Le risorse dell'infrastruttura devono essere distribuite nell' AWS account in cui hai creato l'account cloud principale di DevOps Agent Space. La regione specifica non ha importanza.

1. Accedi alla AWS console: https://console.aws.amazon.com
2. Assicurati di lavorare con lo stesso AWS account in cui si trova il tuo DevOps Agent Space
3. Puoi utilizzare qualsiasi regione per le tue risorse di test

> **Note**
>
> La mappatura 1:1 tra l'account principale del vostro DevOps agente e le risorse dell'ambiente di test che state creando semplifica la configurazione del test. Puoi facilmente estendere il tuo DevOps Agent Space per includere account secondari e abilitare indagini su più account.

Scegli il tuo test

Puoi eseguire entrambi i test indipendentemente o entrambi insieme:

Opzione di test A: test della capacità della CPU EC2

Scopo: convalidare la capacità dell' AWS DevOps agente di rilevare e analizzare i problemi di prestazioni di EC2

Tempo stimato: 5 minuti di configurazione+10 minuti di esecuzione automatica

Difficoltà: Completamente automatizzato (non sono richiesti passaggi manuali)

Opzione di test B: test del tasso di errore Lambda

Scopo: convalidare la capacità dell' AWS DevOps agente di rilevare e analizzare gli errori della funzione Lambda

Tempo stimato: 10 minuti di configurazione+ 2 minuti di attivazione

Difficoltà: Molto facile

Opzione di test A: test della capacità della CPU EC2

Fase 1: Implementazione CloudFormation dello stack per il test EC2

Le utilizzeremo CloudFormation per creare le nostre risorse di test, che consentiranno ad AWS DevOps Agent di monitorarle e analizzarle correttamente.

1. Vai a CloudFormation:
   a. In AWS Console, cerca "CloudFormation" e fai clic su CloudFormation
   b. Fai clic su Crea pila > Con nuove risorse (standard)
2. Carica modello:
   a. Crea un nuovo file locale chiamato `AWS-DevOpsAgent-ec2-test.yaml`
   b. Copia e incolla questo CloudFormation modello nel file:
      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. Nella CloudFormation console, seleziona Carica un file modello

d. Fai clic su Scegli file

e. Seleziona il `AWS-DevOpsAgent-ec2-test.yaml` file

f. Fai clic su Avanti

3. Configura lo stack:

   a. Nome dello stack: `AWS-DevOpsAgent-EC2-Test`

   b. Parametri:

      i. VpcId: seleziona un VPC esistente dal menu a discesa.

   ii. SubnetId: Seleziona una sottorete all'interno del VPC che hai scelto. Per l'accesso SSH, la sottorete deve essere instradata verso un gateway Internet e all'istanza deve essere associato un indirizzo IPv4 pubblico. In caso contrario, l'`SSHCommand`output sarà vuoto e le connessioni SSH non avranno esito positivo.
   iii. MyIP: lascia come impostazione predefinita `0.0.0.0/0` (puoi proteggerlo in seguito, se necessario)
   c. Fai clic su Avanti
4. Configura le opzioni dello stack:
   a. Lascia le impostazioni predefinite, fai clic su Avanti
5. Revisione e creazione:
   a. Seleziona Riconosco che AWS CloudFormation potrebbe creare risorse IAM
   b. Fai clic su Invia
6. Attendi il completamento:
   a. La creazione dello stack richiede 3-5 minuti
   b. Lo stato cambierà da a `CREATE_IN_PROGRESS` `CREATE_COMPLETE`
   c. Importante: la tua istanza EC2 fa ora parte di uno CloudFormation stack in AWS DevOpsAgent grado di tracciare!

Opzionale: accesso SSH sicuro (solo se prevedi di connetterti all'istanza)

Salta questo passaggio se desideri solo eseguire il test automatico

1. Individua il gruppo di sicurezza:
   a. In AWS Console, vai a CloudFormatione seleziona lo `AWS-DevOpsAgent-EC2-Test` stack
   b. Apri la scheda Output e copia il valore di `SecurityGroupId` (inizia con) `sg-`
   c. Vai a EC2 → Gruppi di sicurezza e incolla l'ID nella barra di ricerca per aprire il gruppo di sicurezza
2. Aggiorna la regola SSH:
   a. Seleziona il gruppo di sicurezza → scheda Regole in entrata → Modifica regole in entrata
   b. Trova la regola SSH (porta 22)
   c. Cambia la fonte dal `0.0.0.0/0` tuo IP: `[YOUR_IP]/32`
   d. Ottieni il tuo IP da https://whatismyipaddress.com
   e. Fai clic su Salva regole

Fase 2: Attendi l'esecuzione automatica del test

1. Esecuzione automatica del test:
   - Lo stress test della CPU verrà avviato automaticamente 5 minuti dopo l'avvio dell'istanza
   - Non è richiesto alcun intervento manuale: basta attendere, il test viene eseguito completamente in background
2. Monitora il test:
   - L'istanza si avvia e prepara il test automaticamente
   - Lo script verrà eseguito per 5 minuti e genererà un utilizzo della CPU superiore al 70%
   - CloudWatch l'allarme dovrebbe attivarsi entro 8-10 minuti in totale (5 minuti di ritardo+ 3-5 minuti di allarme)
3. Opzionale: riesecuzione manuale (per test aggiuntivi):
   - Connettiti alla tua istanza: console EC2 → → Connect **`AWS-DevOpsAgent-Test-Instance`** → Session Manager
   - Esegui nuovamente lo stress test: `./cpu-stress-test.sh`
   - Perfetto per AWS DevOpsAgent la risposta dei test più volte

Opzione di test B: test del tasso di errore Lambda

Fase 1: Implementazione CloudFormation dello stack per il test Lambda

1. Passa a: CloudFormation
   a. In AWS Console, vai a CloudFormation
   b. Fai clic su Crea pila → Con nuove risorse (standard)
2. Carica modello:
   a. Crea un nuovo file locale chiamato `AWS-DevOpsAgent-lambda-test.yaml`
   b. Copia e incolla questo CloudFormation modello nel file:
      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. Nella CloudFormation console, seleziona Carica un file modello

d. Fai clic su Scegli file

e. Seleziona il `AWS-DevOpsAgent-lambda-test.yaml` file

f. Fai clic su Avanti

3. Configura lo stack:

a. Nome dello stack: `AWS-DevOpsAgent-Lambda-Test`

b. Fai clic su Avanti

4. Configura le opzioni dello stack:

a. Lascia le impostazioni predefinite, fai clic su Avanti

5. Revisione e creazione:

a. Seleziona Riconosco che AWS CloudFormation potrebbe creare risorse IAM

b. Fai clic su Invia

6. Attendi il completamento:

a. La creazione dello stack richiede 2-3 minuti

b. Lo stato cambierà in `CREATE_COMPLETE`

Fase 2: Attivazione degli errori Lambda

1. Vai alla console Lambda:

a. Vai alla console AWS Lambda

b. Trova la tua funzione `AWS-DevOpsAgent-test-lambda`

2. Prova la funzione:

a. Fai clic sulla scheda Test

b. Fai clic su Crea nuovo evento

c. Nome dell'evento: `AWS-DevOpsAgent-test-event`

d. Usa questo payload JSON:

i.

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

```
}
```

e. Fai clic su Salva

3. Genera errori:

a. Fai clic sul pulsante Test 3 volte (attendi 10 secondi tra una e l'altra)

b. Ogni test genererà un errore intenzionale

c. CloudWatch l'allarme dovrebbe attivarsi entro 2-3 minuti

d. AWS DevOpsAgentora dovrebbe essere in grado di rilevare l'allarme con un'indagine nell'app Operator che configurerai in seguito.

Convalida AWS DevOps Rilevamento degli agenti

Fase 1: CloudWatch allarmi Sanity Check (opzionale)

Questo passaggio serve a garantire che i test precedenti siano ora in uno stato di allarme.

Per il test EC2:

- Nella CloudWatch console, vai a Allarmi
- Attendere 3-5 minuti dopo l'inizio dello stress test
- La sveglia dovrebbe apparire in stato di allarme
- Se è ancora «OK»: attendi altri 2-3 minuti (le CloudWatch metriche possono subire ritardi)

Per il test Lambda:

- Controlla l'allarme `AWS-DevOpsAgent-Lambda-Error-Test`
- Dovrebbe apparire In alarm entro 2-3 minuti dall'esecuzione dei test

Fase 2: Avviare un AWS DevOps Indagine sugli agenti

1. Apri il tuo AWS DevOps agente AgentSpace
2. Fai clic su Accesso amministratore. Si aprirà l'app web DevOps Agent Space in una nuova finestra
3. Fai clic sul pulsante Avvia indagine sul lato destro dello schermo
4. Compila il seguente modulo:

a. Dettagli dell'indagine: descrivi l'indagine che desideri condurre. Includi tutti i dettagli possibili sugli obiettivi dell'indagine, sulle aree da esplorare o sulle informazioni pertinenti.
b. Punto di inizio dell'indagine: descrivi le informazioni da cui desideri iniziare l'indagine. Puoi menzionare un allarme, una metrica, un frammento di registro o qualsiasi altra cosa per dare all' DevOps Agent un punto di partenza su cui lavorare. In questo caso, fornisci un riepilogo degli allarmi che hai appena creato.
c. Data e ora dell'incidente (preferibilmente ISO 8601): YYYY-MM-DDTHH:MMZ
d. Assegna un nome alla tua indagine: esempio: `Oncall_investigation_1:2025-10-27`
e. AWS ID dell'account per l'incidente
f. Regione in cui si è verificato l'incidente
g. Priorità: AWS DevOpsAgent consente due indagini simultanee. La priorità consente di definire l'ordine di esecuzione delle indagini.

5. Fate clic su Indagate per avviare l'indagine.
6. Fai clic sulla tua indagine elencata nella dashboard. Verrai indirizzato alla schermata Dettagli dell'indagine, dove potrai visualizzare i passaggi dettagliati che l' DevOps agente sta eseguendo.

Risultati attesi

Risultati del test EC2:

- Rileva l'allarme della CPU EC2
- Identifica la causa principale: «carico di lavoro dello stress test della CPU»
- Mostra la sequenza temporale: Stress test → Picco della CPU → Allarme
- Fornisce consigli per il monitoraggio e la scalabilità

Risultati del test Lambda:

- Rileva il picco del tasso di errore Lambda
- Identifica la causa principale: «eccezioni intenzionali nei test»
- Mostra la sequenza temporale: Invocazioni di funzioni → Errori → Allarme
- Fornisce consigli per la gestione e il monitoraggio degli errori

Istruzioni per la pulizia

Test di pulizia A (test EC2)

Pulizia automatica

- L'istanza verrà terminata automaticamente dopo 2 ore (integrata nel CloudFormation modello)

Pulizia manuale (immediata)

1. Elimina CloudFormation stack:
   a. Vai alla console CloudFormation
   b. Seleziona `AWS-DevOpsAgent-EC2-Test` pila
   c. Fai clic su Elimina
   d. Conferma l'eliminazione
   e. Questo eliminerà automaticamente tutte le risorse: istanza EC2, gruppo di sicurezza, key pair e alarm CloudWatch

Test di pulizia B (test Lambda)

1. Elimina stack CloudFormation :
   a. Vai alla console CloudFormation
   b. Seleziona `AWS-DevOpsAgent-Lambda-Test` pila
   c. Fai clic su Elimina
   d. Conferma l'eliminazione
   e. Questo eliminerà automaticamente tutte le risorse: funzione Lambda, ruolo IAM e allarme CloudWatch

Risoluzione dei problemi

Problemi comuni

«Impossibile connettersi all'istanza EC2»

- Verifica il gruppo di sicurezza: assicurati che SSH (porta 22) sia aperto al tuo IP

- Controlla le autorizzazioni chiave: Esegui `chmod 400 AWS-DevOpsAgent-test-key.pem`
- Verifica IP pubblico: all'istanza deve essere assegnato un IP pubblico
- Attendi l'istanza: assicurati che l'istanza sia nello stato «In esecuzione»

«Allarme non si attiva»

- Attendi la visualizzazione delle metriche: la CloudWatch visualizzazione delle metriche può richiedere 2-5 minuti
- Controlla il carico della CPU: invia SSH all'istanza ed esegui per verificare che la CPU `top` sia superiore al 70%
- Verifica lo stress test: esegui `ps aux | grep yes` per vedere se i processi di caricamento sono in esecuzione
- Attesa prolungata: a volte occorrono fino a 7-8 minuti per attivare il primo allarme

Convalida dei test

Il test AWS DevOp dell'agente ha esito positivo quando:

Validazione tecnica

- Precisione dell'indagine: i risultati del test EC2 dovrebbero indicare correttamente che l'allarme è stato attivato a causa del carico della CPU. Il risultato del test Lambda dovrebbe indicare che si è trattato di un errore intenzionale.
- Precisione della sequenza temporale: viene mostrata la sequenza corretta degli eventi
- Qualità della raccomandazione: vengono forniti suggerimenti attuabili

Guida introduttiva all'utilizzo di AWS DevOps Agent con AWS CDK

Panoramica di

Questa guida mostra come utilizzare il AWS Cloud Development Kit (AWS CDK) per creare e distribuire risorse per AWS DevOps agenti. L'applicazione AWS CDK automatizza la creazione di uno spazio agente, ruoli AWS Identity and Access Management (IAM), un'app operatore e AWS le associazioni di account tramite. AWS CloudFormation

L'approccio AWS CDK automatizza i passaggi manuali descritti nella guida all'onboarding della CLI definendo tutte le risorse richieste come infrastruttura come codice.

AWS DevOps L'agente è disponibile nelle seguenti 6 AWS regioni: Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon), Asia Pacifico (Sydney), Asia Pacifico (Tokyo), Europa (Francoforte) ed Europa (Irlanda). Per ulteriori informazioni sulle regioni supportate, consulta. the section called "Regioni supportate"

Prerequisiti

Prima di iniziare, assicurati di disporre di:

- AWS Interfaccia a riga di comando (AWS CLI) installata e configurata con le credenziali appropriate
- Node.js versione 18 o successiva
- AWS Interfaccia a riga di comando (CLI) CDK installata a livello globale. Per installare la CLI AWS CDK, esegui il seguente comando:

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

- Un AWS account per l'account di monitoraggio (primario)
- (Facoltativo) Un secondo AWS account se desideri configurare il monitoraggio tra più account

Cosa tratta questa guida

Questa guida è divisa in due parti:

- Parte 1: implementa uno spazio per agenti con un'app per operatori e un' AWS associazione nel tuo account di monitoraggio. Dopo aver completato questa parte, l'agente può monitorare i problemi relativi all'account.
- Parte 2 (facoltativa): aggiungi un' AWS associazione di origine per un account di servizio e distribuisci un ruolo IAM tra account in quell'account. Questa configurazione consente allo spazio dell'agente di monitorare le risorse tra gli account.

Risorse create

Parte 1: DevOpsAgentStack (account di monitoraggio)

- Ruolo IAM (`DevOpsAgentRole-AgentSpace`): assunto dal servizio DevOps Agent per monitorare l'account. Include la policy `AIDevOpsAgentAccessPolicy` gestita e una policy in linea che consente la creazione del ruolo collegato al servizio Resource Explorer.
- IAM role (`DevOpsAgentRole-WebappAdmin`): ruolo dell'operatore nell'app con policy `AIDevOpsOperatorAppAccessPolicy` gestita per le operazioni degli agenti.
- Agent space (`MyCDKAgentSpace`): lo spazio centrale degli agenti, creato utilizzando la `AWS::DevOpsAgent::AgentSpace` CloudFormation risorsa. Include la configurazione dell'app per l'operatore.
- Associazione (AWS monitor): collega l'account di monitoraggio allo spazio degli agenti utilizzando la `AWS::DevOpsAgent::Association` CloudFormation risorsa.
- Associazione (AWS fonte): (Facoltativo) Collega l'account di servizio allo spazio dell'agente per il monitoraggio tra account.

Parte 2: ServiceStack (account di servizio, opzionale)

- IAM role (`DevOpsAgentRole-SecondaryAccount`): ruolo interaccount con un nome fisso. Scelto dallo spazio degli agenti nell'account di monitoraggio. Include la policy `AIDevOpsAgentAccessPolicy` gestita e una policy in linea che consente la creazione del ruolo collegato al servizio Resource Explorer.
- Funzione Lambda (`echo-service`): un semplice servizio di esempio che riproduce gli eventi di input di ritorno.

Configurazione

Fase 1: clonare il repository di esempio

Esegui i seguenti comandi per clonare il repository e passare alla directory del progetto:

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

Fase 2: Installare le dipendenze

Esegui il seguente comando per installare le dipendenze del progetto:

```
npm install
```

Parte 1: Distribuisci lo spazio degli agenti

In questa sezione, crei lo spazio degli agenti, i ruoli IAM, l'app operatore e un' AWS associazione nel tuo account di monitoraggio.

Passaggio 1: configura l'ID dell'account di monitoraggio

Apri `lib/constants.ts` e imposta l'ID del tuo account di monitoraggio:

L'esempio seguente mostra la costante da aggiornare:

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

Fase 2: Avvia l'ambiente AWS CDK

Se non hai avviato il AWS CDK nel tuo account di monitoraggio, esegui il seguente comando:

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

Fase 3: Compila e distribuisci

Esegui i seguenti comandi per creare il TypeScript codice e distribuire lo stack:

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

Fase 4: Registrare gli output dello stack

Al termine della distribuzione, il AWS CDK stampa gli output dello stack. Registra questi valori per un uso successivo.

L'esempio seguente mostra l'output previsto:

```
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
```

Se intendi completare la Parte 2, salva il `AgentSpaceArn` valore. È necessario per configurare lo stack di account del servizio.

Fase 5: Verificare la distribuzione

Per verificare che lo spazio agente sia stato creato correttamente, esegui il seguente comando AWS CLI:

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

A questo punto, lo spazio per gli agenti viene distribuito con l'app operatore abilitata e l'account di monitoraggio associato. L'agente può monitorare i problemi relativi a questo account.

Parte 2 (facoltativa): aggiungi il monitoraggio tra account

In questa sezione, estendete la configurazione in modo che lo spazio dell'agente possa monitorare le risorse in un secondo AWS account (l'account di servizio). Ciò comporta due azioni:

1. Aggiungere un' AWS associazione di origine DevOpsAgentStack che punta all'account di servizio.
2. Implementazione dell'account ServiceStack nell'account di servizio con un ruolo IAM che si fida dello spazio dell'agente.

> **Important**
>
> È necessario completare la parte 1 prima di procedere. ServiceStack Richiede l'`AgentSpaceArn`output dell' DevOpsAgentStack implementazione.

Fase 1: Configurare l'ID dell'account del servizio

Apri `lib/constants.ts` e imposta l'ID dell'account di servizio:

L'esempio seguente mostra la costante da aggiornare:

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

DevOpsAgentStack crea un' AWS associazione di origine utilizzando questo ID di account. Se hai distribuito DevOpsAgentStack prima di impostare questo valore, ridistribuisci per creare l'associazione:

Esegui i seguenti comandi per ridistribuire:

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

Fase 2: Impostare lo spazio dell'agente (ARN)

Copia il `AgentSpaceArn` valore dall' DevOpsAgentStack output (Parte 1, Fase 4) e impostalo in`lib/constants.ts`:

L'esempio seguente mostra la costante da aggiornare:

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

ServiceStack utilizza questo valore per definire l'ambito della politica di fiducia sul ruolo secondario dell'account. ServiceStack Viene sintetizzato solo quando questo valore è impostato.

Fase 3: Avvia l'account del servizio

Se non hai avviato il AWS CDK nel tuo account di servizio, esegui il seguente comando:

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

Fase 4: Implementare ServiceStack

Esegui i seguenti comandi per crearlo e distribuirlo ServiceStack utilizzando le credenziali per l'account del servizio:

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

In questo modo vengono create le seguenti risorse nell'account di servizio:

- Un ruolo IAM (`DevOpsAgentRole-SecondaryAccount`) che affida lo spazio dell'agente nell'account di monitoraggio
- Una funzione echo Lambda `echo-service` () come servizio di esempio

Fase 5: Verificare la distribuzione

Per confermare che la funzione Lambda è stata implementata correttamente, esegui i seguenti comandi per testare il servizio echo:

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

Risoluzione dei problemi

Questa sezione descrive i problemi più comuni e come risolverli.

CloudFormation tipo di risorsa non trovato

- Verifica di eseguire la distribuzione in unthe section called “Regioni supportate”.
- Verifica che la tua AWS CLI sia configurata con le autorizzazioni appropriate.

Creazione del ruolo IAM non riuscita

- Verifica che il tuo ruolo di distribuzione disponga delle autorizzazioni per creare ruoli IAM.
- Verifica che le condizioni della politica di fiducia corrispondano all'ID del tuo account.

La distribuzione tra account non riesce e viene visualizzato il messaggio «Impossibile assumere il ruolo nell'account di destinazione»

- Ogni stack deve essere distribuito con le credenziali per l'account di destinazione. Utilizzate il `--profile` flag per specificare il profilo AWS CLI corretto.
- Verifica che il AWS CDK sia stato avviato nell'account di destinazione.

Ritardi di propagazione IAM

- La propagazione delle modifiche ai ruoli IAM può richiedere alcuni minuti. Se la creazione dello spazio dell'agente fallisce subito dopo la creazione del ruolo, attendi qualche minuto e ridistribuisci.

Pulizia

Per rimuovere tutte le risorse, distruggi le pile in ordine inverso.

Esegui i seguenti comandi per distruggere gli stack:

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

Avviso: questa azione elimina definitivamente lo spazio dell'agente e tutti i dati associati. Questa azione non può essere annullata. Assicurati di aver eseguito il backup di tutte le informazioni importanti prima di procedere.

Considerazioni relative alla sicurezza

- L'applicazione AWS CDK crea ruoli IAM con policy di fiducia che consentono solo al responsabile del `aidevops.amazonaws.com` servizio di assumerli.
- Le politiche di fiducia includono condizioni che limitano l'accesso al tuo AWS account specifico e allo spazio ARN dello spazio agente.
- Tutte le politiche seguono il principio del privilegio minimo. Rivedi e personalizza le policy IAM in base ai requisiti di sicurezza della tua organizzazione.
- Il ruolo tra account (`DevOpsAgentRole-SecondaryAccount`) utilizza un nome fisso ed è limitato a uno spazio di agenti ARN specifico.

Fasi successive

Dopo aver distribuito l' AWS DevOps agente utilizzando il CDK: AWS

1. Scopri l'intera gamma di funzionalità dell' DevOps agente nella Guida per l'utente dell'AWS DevOps agente.
2. Prendi in considerazione l'integrazione dell'implementazione AWS CDK nelle tue CI/CD pipeline per la gestione automatizzata dell'infrastruttura.

Risorse aggiuntive

- AWS DevOps Guida per l'utente dell'agente
- Esempio di repository CDK sul sito Web GitHub
- Guida all'onboarding CLI

Guida introduttiva all'utilizzo di AWS DevOps Agent AWS CloudFormation

Panoramica di

Questa guida mostra come utilizzare i AWS CloudFormation modelli per creare e distribuire le risorse degli AWS DevOps agenti. I modelli automatizzano la creazione di uno spazio agente, ruoli AWS Identity and Access Management (IAM), un'app operatore e associazioni di AWS account come infrastruttura come codice.

L' CloudFormation approccio automatizza i passaggi manuali descritti nella guida all'onboarding della CLI definendo tutte le risorse richieste in modelli YAML dichiarativi.

AWS DevOps L'agente è disponibile nelle seguenti 6 AWS regioni: Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon), Asia Pacifico (Sydney), Asia Pacifico (Tokyo), Europa (Francoforte) ed Europa (Irlanda). Per ulteriori informazioni sulle regioni supportate, consulta. the section called “Regioni supportate”

Prerequisiti

Prima di iniziare, assicurati di disporre di:

- AWS Interfaccia a riga di comando (AWS CLI) installata e configurata con le credenziali appropriate
- Autorizzazioni per creare ruoli e stack IAM CloudFormation
- Un AWS account per l'account di monitoraggio (principale)
- (Facoltativo) Un secondo AWS account se desideri configurare il monitoraggio tra più account

Cosa tratta questa guida

Questa guida è divisa in due parti:

- Parte 1: implementa uno spazio per agenti con un'app per operatori e un' AWS associazione nel tuo account di monitoraggio. Dopo aver completato questa parte, l'agente può monitorare i problemi relativi all'account.
- Parte 2 (facoltativa): implementa un ruolo IAM tra account diversi in un account secondario e aggiungi un'associazione di origine AWS . Questa configurazione consente allo spazio dell'agente di monitorare le risorse tra gli account.

Parte 1: Distribuisci lo spazio degli agenti

In questa sezione, crei un CloudFormation modello che fornisce lo spazio degli agenti, i ruoli IAM, l'app operatore e un' AWS associazione nel tuo account di monitoraggio.

Fase 1: Creare il CloudFormation modello

Salva il seguente modello come`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
```

Fase 2: Distribuire lo stack

Esegui il comando seguente per distribuire lo stack. Sostituisci <REGION> con a the section called "Regioni supportate" (ad esempio,us-east-1).

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

Fase 3: Registrare gli output dello stack

Al termine della distribuzione, esegui il comando seguente per recuperare gli output dello stack. Registra questi valori per un uso successivo.

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

L'esempio seguente mostra l'output previsto:

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

Se intendi completare la Parte 2, salva il `AgentSpaceArn` valore. Ne hai bisogno per configurare il ruolo tra account.

Fase 4: Verifica della distribuzione

Per verificare che lo spazio agente sia stato creato correttamente, esegui il seguente comando AWS CLI:

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

A questo punto, lo spazio per gli agenti viene distribuito con l'app operatore abilitata e l'account di monitoraggio associato. L'agente può monitorare i problemi relativi a questo account.

Parte 2 (facoltativa): aggiungi il monitoraggio tra account

In questa sezione, estendete la configurazione in modo che lo spazio dell'agente possa monitorare le risorse in un secondo AWS account (l'account di servizio). Ciò comporta due azioni:

1. Implementazione di un ruolo IAM nell'account di servizio che affida lo spazio degli agenti.
2. Aggiungere un' AWS associazione di origine nell'account di monitoraggio che punti all'account di servizio.

> Nota: è necessario completare la Parte 1 prima di procedere. Il modello di account di servizio richiede gli `AgentSpaceArn` output dello stack della Parte 1.

Fase 1: Creare il modello di account di servizio

Salva il seguente modello come`devops-agent-service-account.yaml`. Questo modello crea un ruolo IAM tra account diversi nell'account secondario.

```
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
```

Fase 2: Implementazione dello stack di account di servizio

Utilizzando le credenziali per l'account di servizio, esegui il comando seguente:

```
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>
```

Passaggio 3: Aggiungere l'associazione di origine AWS

Torna all'account di monitoraggio e crea un' AWS associazione di origine. Puoi farlo creando uno stack separato o aggiornando il modello originale. L'esempio seguente utilizza un modello autonomo.

Salva il seguente modello come`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
```

Distribuisci lo stack di associazione utilizzando le credenziali dell'account di monitoraggio:

```
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>
```

Verifica

Verifica la configurazione eseguendo i seguenti comandi AWS CLI:

```
# 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>
```

Risoluzione dei problemi

Questa sezione descrive i problemi più comuni e come risolverli.

CloudFormation tipo di risorsa non trovato

- Verifica di eseguire la distribuzione in unthe section called "Regioni supportate".
- Verifica che la tua AWS CLI sia configurata con le autorizzazioni appropriate.

Creazione del ruolo IAM non riuscita

- Verifica che le tue credenziali di distribuzione dispongano delle autorizzazioni per creare ruoli IAM con nomi personalizzati ()`CAPABILITY_NAMED_IAM`.
- Verifica che le condizioni della politica di fiducia corrispondano all'ID del tuo account.

La distribuzione tra account non riesce

- Ogni stack deve essere distribuito con le credenziali per l'account di destinazione. Utilizzate il `--profile` flag per specificare il profilo AWS CLI corretto.
- Verificate che il `AgentSpaceArn` parametro corrisponda all'ARN esatto degli output dello stack Parte 1.

Ritardi di propagazione IAM

- La propagazione delle modifiche ai ruoli IAM può richiedere alcuni minuti. Se la creazione dello spazio dell'agente fallisce subito dopo la creazione del ruolo, attendi qualche minuto e ridistribuisci.

Pulizia

Per rimuovere tutte le risorse, elimina gli stack in ordine inverso.

Avviso: questa azione elimina definitivamente lo spazio dell'agente e tutti i dati associati. Questa azione non può essere annullata. Assicurati di aver eseguito il backup di tutte le informazioni importanti prima di procedere.

Esegui i seguenti comandi per eliminare gli stack:

```
# 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>
```

Fasi successive

Dopo aver distribuito l' AWS DevOps agente utilizzando: AWS CloudFormation

- Per connettere integrazioni aggiuntive, consulta. Configurazione delle funzionalità per AWS DevOps Agente
- Per ulteriori informazioni sulle competenze e le funzionalità degli agenti, consultathe section called "DevOps Competenze degli agenti".
- Per comprendere l'app web dell'operatore, consultathe section called "Cos'è un'app Web per DevOps agenti?".

Guida introduttiva a AWS DevOps Agent utilizzando Terraform

Panoramica di

Questa guida mostra come utilizzare Terraform per creare e distribuire AWS DevOps risorse Agent. La configurazione Terraform automatizza la creazione di uno spazio per agenti, ruoli IAM, un'app operatore e associazioni di account. AWS

L'approccio Terraform automatizza i passaggi manuali descritti nella guida all'onboarding della CLI definendo tutte le risorse richieste come infrastruttura come codice.

AWS DevOps L'agente è disponibile nelle seguenti 6 AWS regioni: Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon), Asia Pacifico (Sydney), Asia Pacifico (Tokyo), Europa (Francoforte) ed Europa (Irlanda). Per ulteriori informazioni sulle regioni supportate, consulta. the section called "Regioni supportate"

Prerequisiti

Prima di iniziare, assicurati di disporre di:

- Terraform >= 1.0 installato
- AWS CLI installata e configurata con credenziali appropriate
- Un AWS account per l'account di monitoraggio (primario)
- (Facoltativo) Un secondo AWS account se desideri configurare il monitoraggio tra più account

Cosa tratta questa guida

Questa guida è divisa in due parti:

- Parte 1: implementa uno spazio per agenti con un'app per operatori e un' AWS associazione nel tuo account di monitoraggio. Dopo aver completato questa parte, l'agente può monitorare i problemi di quell'account.
- Parte 2 (facoltativa): aggiungi un' AWS associazione di origine per un account di servizio e distribuisci un ruolo IAM tra account più un echo Lambda in quell'account. Ciò consente allo spazio dell'agente di monitorare le risorse tra gli account.

Risorse create

Parte 1: Monitoraggio dell'account

- Ruolo IAM (`DevOpsAgentRole-AgentSpace-*`): assunto dal servizio DevOps Agent per monitorare l'account. Include la policy `AIDevOpsAgentAccessPolicy` gestita e una policy in linea che consente la creazione del ruolo collegato al servizio Resource Explorer.
- IAM role (`DevOpsAgentRole-WebappAdmin-*`): ruolo dell'operatore nell'app con policy `AIDevOpsOperatorAppAccessPolicy` gestita per le operazioni degli agenti.
- Spazio agente (nome configurabile): lo spazio centrale dell'agente, creato utilizzando la `awscc_devopsagent_agent_space` risorsa. Include la configurazione dell'app per l'operatore.
- Associazione (AWS monitor): collega l'account di monitoraggio allo spazio dell'agente utilizzando la `awscc_devopsagent_association` risorsa.
- Associazione (AWS fonte): (Facoltativo) Collega l'account di servizio allo spazio dell'agente per il monitoraggio tra account.

Parte 2: Account di servizio (opzionale)

- IAM role (`DevOpsAgentRole-SecondaryAccount-TF`): ruolo interaccount con un nome fisso. Scelto dallo spazio degli agenti nell'account di monitoraggio. Include la policy `AIDevOpsAgentAccessPolicy` gestita e una policy in linea che consente la creazione del ruolo collegato al servizio Resource Explorer.
- Funzione Lambda (`echo-service-tf`): un semplice servizio di esempio che riproduce gli eventi di input di ritorno.

Configurazione

Fase 1: clonare il repository di esempio

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

Fase 2: Configurare le variabili

Copia il file delle variabili di esempio e personalizzalo per il tuo ambiente:

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

Modifica `terraform.tfvars` con il nome e la descrizione dello spazio dell'agente:

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

Parte 1: Distribuisci lo spazio degli agenti

In questa sezione, crei lo spazio degli agenti, i ruoli IAM, l'app operatore e un' AWS associazione nel tuo account di monitoraggio.

Fase 1: Implementazione con automazione (consigliata)

Utilizza lo script di distribuzione fornito per una configurazione semplificata:

```
./deploy.sh
```

Questo script automaticamente:

- Verifica i prerequisiti (Terraform, AWS CLI, credenziali)
- Crea `terraform.tfvars` dall'esempio se necessario
- Inizializza, convalida, pianifica e applica Terraform

In alternativa, se preferisci il controllo manuale:

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

Digita `yes` quando richiesto per confermare la distribuzione.

Fase 2: Registrare le uscite

Al termine della distribuzione, Terraform stampa gli output. Registra questi valori per un uso successivo:

```
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"
```

Se intendi completare la Parte 2, salva il `agent_space_arn` valore. Ti servirà per configurare le risorse dell'account di servizio.

Fase 3: Verificare la distribuzione

Esegui lo script di verifica post-implementazione:

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

Oppure usa la AWS CLI per verificare che lo spazio dell'agente sia stato creato correttamente:

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

A questo punto, lo spazio per gli agenti viene distribuito con l'app dell'operatore abilitata e l'account di monitoraggio associato. L'agente può monitorare i problemi relativi a questo account.

Parte 2 (facoltativa): aggiungi il monitoraggio tra account

In questa sezione, estendete la configurazione in modo che lo spazio dell'agente possa monitorare le risorse in un secondo AWS account (l'account di servizio). Ciò comporta due azioni:

1. Aggiungere un' AWS associazione di origine che punti all'account del servizio.
2. Implementazione di un ruolo IAM su più account e di una funzione echo Lambda nell'account del servizio.

> **Important**
>
> È necessario completare la parte 1 prima di procedere. Le risorse dell'account di servizio richiedono l'output `agent_space_arn` di distribuzione della Parte 1.

Fase 1: Configurare l'ID dell'account di servizio

In `terraform.tfvars`, imposta l'ID del tuo account di servizio:

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

Fase 2: Impostare lo spazio dell'agente (ARN)

Copiate il `agent_space_arn` valore dall'output della Parte 1 (Fase 2) e impostatelo `interraform.tfvars`:

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

Le risorse dell'account di servizio utilizzano questo valore per definire la politica di fiducia relativa al ruolo secondario dell'account. Queste risorse vengono create solo quando questo valore è impostato.

Passaggio 3: configurare il provider `aws.service`

In `main.tf`, configura l'alias del `aws.service` provider con le credenziali per l'account di servizio. Puoi utilizzare un profilo denominato o assumere un ruolo:

Utilizzando un profilo:

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

O usando assume il ruolo:

```
provider "aws" {
```

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

Fase 4: Implementazione

Applica la configurazione aggiornata:

```
terraform apply
```

In questo modo vengono create le seguenti risorse nell'account del servizio:

- Un ruolo IAM (`DevOpsAgentRole-SecondaryAccount-TF`) che affida lo spazio dell'agente nell'account di monitoraggio
- Una funzione echo Lambda `echo-service-tf` () come servizio di esempio

Crea inoltre un' AWS associazione di origine nell'account di monitoraggio che collega l'account di servizio.

Fase 5: Verificare la distribuzione

Prova il servizio echo per confermare che la funzione Lambda è stata implementata correttamente:

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

Risoluzione dei problemi

Ritardi di propagazione IAM

- La configurazione include 30 secondi `time_sleep` tra la creazione del ruolo IAM e la creazione di Agent Space. Il servizio DevOps Agent convalida la politica di fiducia del ruolo dell'operatore

durante la creazione di Agent Space, e questa operazione può fallire se IAM non viene propagato completamente. Se continui a riscontrare errori nelle policy di fiducia, attendi un minuto ed esegui di `terraform apply` nuovo: i ruoli IAM esisteranno già e l'applicazione riprenderà dal punto in cui era stata interrotta.

Errori di autorizzazione

- Verifica che AWS le tue credenziali dispongano delle autorizzazioni IAM necessarie per creare ruoli e policy.
- Verifica che le condizioni della politica di fiducia corrispondano all'ID del tuo account.

La distribuzione tra account non riesce

- Il `aws.service` provider deve essere configurato con le credenziali per l'account di servizio. Utilizza un profilo denominato o un blocco Assume Role.
- Verificate che il `agent_space_arn` valore corrisponda all'ARN dell'output della Parte 1.

Tipo di risorsa Terraform non trovato

- Verifica di avere la versione del `awscc` provider `~> 1.0` o successiva. Le `awscc_devopsagent_association` risorse `awscc_devopsagent_agent_space` e richiedono il provider AWS Cloud Control.

Pulizia

Per rimuovere tutte le risorse, distruggile in ordine inverso se hai distribuito la Parte 2:

```
./cleanup.sh
```

O manualmente:

```
terraform destroy
```

Avviso: questa operazione elimina definitivamente lo spazio dell'agente e tutti i dati associati. Assicurati di aver eseguito il backup di tutte le informazioni importanti prima di procedere.

Considerazioni relative alla sicurezza

- La configurazione Terraform crea ruoli IAM con politiche di fiducia che consentono solo al responsabile del `aidevops.amazonaws.com` servizio di assumerli.
- Le politiche di fiducia includono condizioni che limitano l'accesso al tuo AWS account specifico e allo spazio ARN dello spazio agente.
- Tutte le politiche seguono il principio del privilegio minimo. Rivedi e personalizza le policy IAM in base ai requisiti di sicurezza della tua organizzazione.
- Il ruolo tra account (`DevOpsAgentRole-SecondaryAccount-TF`) utilizza un nome fisso ed è limitato a uno spazio di agenti ARN specifico.

Fasi successive

Dopo aver distribuito il tuo AWS DevOps agente utilizzando Terraform:

1. Scopri l'intera gamma di funzionalità dell' DevOps agente nella Guida per l'utente dell'AWS DevOps agente.
2. Prendi in considerazione l'integrazione dell'implementazione di Terraform nelle tue CI/CD pipeline per la gestione automatizzata dell'infrastruttura.

Risorse aggiuntive

- AWS DevOps Guida per l'utente dell'agente
- Esempio di repository Terraform
- Guida all'onboarding CLI

Lavorare con DevOps l'agente

Lavorare con DevOps l'agente

AWS DevOps L'agente collabora con il team operativo per l'intero ciclo di vita degli incidenti, dal rilevamento alle indagini, al ripristino e alla prevenzione. I seguenti argomenti descrivono come utilizzare DevOps Agent per gestire ogni fase di questo ciclo di vita.

Risposta autonoma agli incidenti

Quando viene rilevato un incidente, tramite un'integrazione integrata con il sistema di ticketing, un webhook degli strumenti di monitoraggio o un'attivazione manuale, l' DevOps agente avvia automaticamente un'indagine. L'agente analizza metriche, log, tracce, modifiche al codice e cronologia di implementazione per determinare la causa principale e proporre un piano di mitigazione. Se hai bisogno di ulteriore assistenza, puoi passare direttamente a AWS Support dall'app web DevOps Agent Space, che condivide automaticamente il contesto dell'indagine con i tecnici dell'assistenza in modo da non dover ripetere ciò che l'agente ha già scoperto. Per ulteriori informazioni, consulta the section called “Risposta autonoma agli incidenti”.

Attività su richiesta DevOps

In qualsiasi momento del ciclo di vita dell'incidente, puoi interagire con DevOps Agent tramite un'interfaccia di chat conversazionale. Poni domande sulle AWS risorse, sullo stato del sistema, sullo stato degli allarmi e sulla cronologia delle implementazioni utilizzando il linguaggio naturale. La chat è sensibile al contesto: quando stai visualizzando un'indagine specifica, puoi indirizzare l'agente a esplorare ipotesi particolari, concentrarsi su log specifici o aggiornare l'analisi della causa principale. Puoi anche interrogare le configurazioni delle risorse, le tendenze degli errori e gli approfondimenti delle indagini in tutto il tuo ambiente senza navigare tra le console. Per ulteriori informazioni, consulta the section called “DevOps Attività su richiesta”.

Prevenzione proattiva degli incidenti

Dopo aver risolto gli incidenti, DevOps Agent analizza i modelli della cronologia delle indagini per generare raccomandazioni che prevengano incidenti futuri e riducano il tempo medio di rilevamento. Le raccomandazioni riguardano quattro aree: posizione di osservabilità, lacune nei test, modifiche al

codice e architettura dell'infrastruttura. L'agente esegue le valutazioni settimanalmente e aggiorna i consigli man mano che si verificano nuovi incidenti. Puoi accettare, rifiutare o tenere traccia dei consigli e l'agente impara dal tuo feedback per affinare i suggerimenti futuri. Per ulteriori informazioni, consulta the section called "Prevenzione proattiva degli incidenti".

Interfacciamento con l'agente DevOps

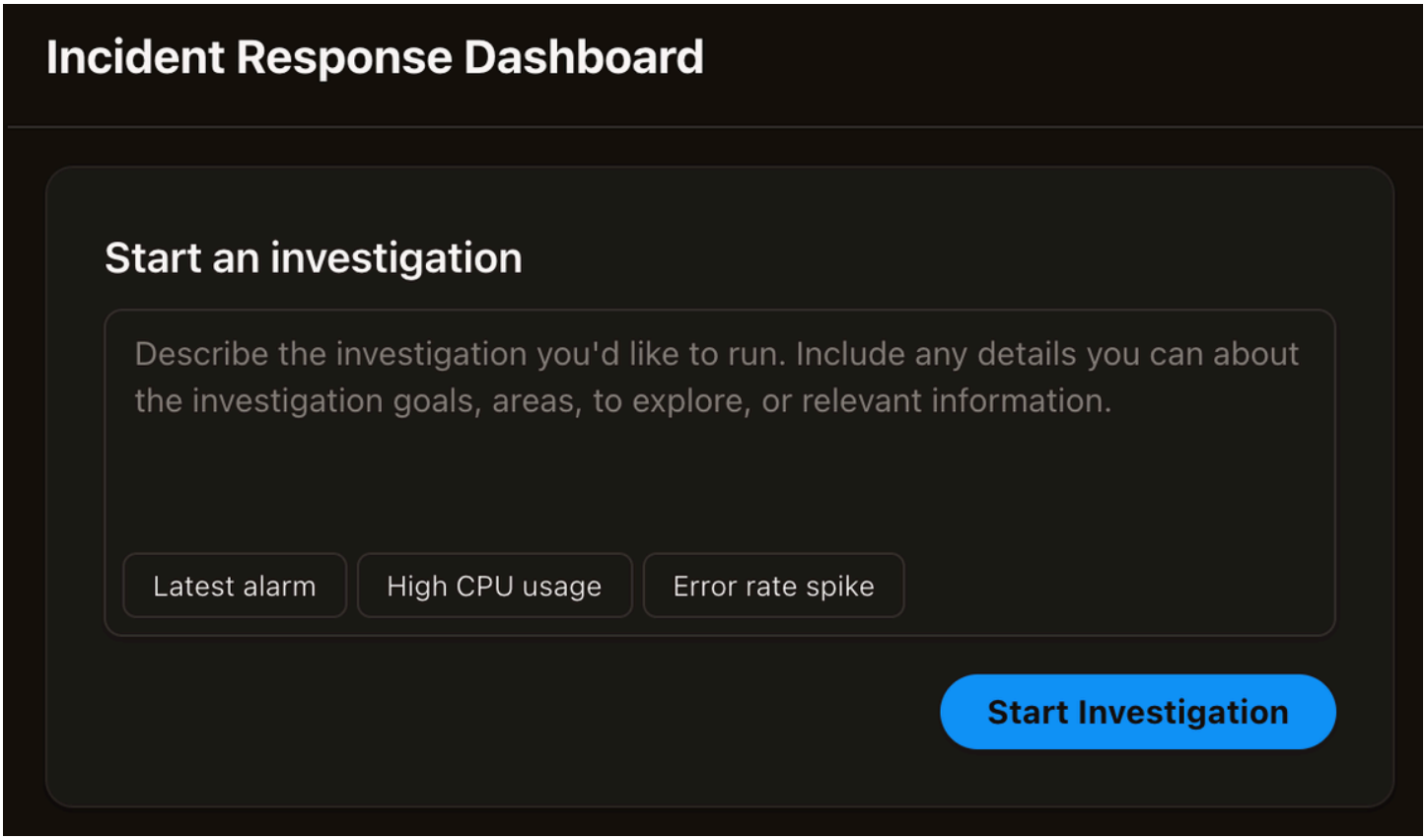
AWS DevOps L'agente supporta diversi metodi di accesso, tra cui la console dell'app Web, l'integrazione MCP per IDEs, l'Agent Client Protocol (ACP), i webhook per l'automazione basata sugli eventi e l'accesso diretto alle API. Per ulteriori informazioni, consulta the section called "Interfacciamento con l'agente DevOps".

Risposta autonoma agli incidenti

Avvio delle indagini

Le indagini sulla risposta agli incidenti possono essere avviate in tre modi.

- Built-in integrazioni: puoi connettere un DevOps Agent Space ai sistemi di ticketing ServiceNow utilizzando ad esempio integrazioni integrate. Una volta connesso, gli DevOps agenti avvieranno automaticamente le indagini sulla risposta agli incidenti dai ticket di assistenza e l' DevOps agente fornirà aggiornamenti sui risultati principali, sulle analisi delle cause principali e sui piani di mitigazione nel ticket di origine.
- Webhook: è possibile utilizzare i webhook per inviare eventi all'agente. AWS DevOps Ad esempio, puoi utilizzare i webhook per avviare indagini sulla risposta agli incidenti dai ticket PagerDuty o dagli allarmi Grafana.
- Manualmente: è possibile avviare manualmente le indagini sulla risposta agli incidenti dalla scheda Incident Response di qualsiasi app web di Agent Space. DevOps Puoi inserire un testo in formato libero che descriva l'incidente su cui desideri che il tuo DevOps agente indaghi e che creerà un piano di indagine, raccoglierà i risultati, determinerà la causa principale e si offrirà di generare un piano di mitigazione. Puoi anche scegliere tra diversi punti di partenza preconfigurati per iniziare rapidamente l'indagine: Allarme più recente per esaminare l'allarme attivato più recente e analizzare le metriche e i log sottostanti per determinarne la causa principale, Utilizzo elevato della CPU per esaminare i parametri di utilizzo elevato della CPU nelle risorse di calcolo e identificare quali processi o servizi stanno consumando risorse eccessive, oppure Error rate spike per indagare sul recente aumento dei tassi di errore delle applicazioni analizzando metriche, registri delle applicazioni e identificando il fonte di guasti.



Dopo aver fatto clic su «Avvia indagine», ti verrà chiesto di fornire alcuni dettagli aggiuntivi per aiutare l'agente a concentrare il suo lavoro. La finestra di dialogo di indagine include i seguenti campi:

- Dettagli dell'indagine, Pre-filled con la tua descrizione. Puoi modificarlo per affinare l'ambito dell'indagine.
- Punto di inizio dell'indagine: descrivi facoltativamente un allarme, una metrica, un frammento di registro o un altro punto di partenza specifico per l'agente.
- Data e ora dell'incidente, Auto-filled con l'ora corrente in formato UTC. Modifica se l'incidente si è verificato prima.
- Assegna un nome alla tua indagine, Auto-generated con un timestamp. Puoi personalizzarlo (massimo 400 caratteri).
- Priorità: seleziona la priorità dell'indagine dal menu a discesa (Media è l'impostazione predefinita).

Rivedi e modifica questi campi secondo necessità, quindi fai clic su «Inizia a indagare...» per iniziare. Verrai quindi indirizzato alla pagina dei dettagli dell'indagine dove potrai vedere il tuo DevOps agente in azione!

Triage degli incidenti

La fase di triage è la prima fase del sistema di risposta agli incidenti di AWS DevOps Agent. Quando si verifica un evento esterno, ad esempio un allarme di Datadog, un ticket di emergenza o un problema di Dynatrace ServiceNow, AWS DevOps Agent lo elabora automaticamente in pochi secondi per determinare se debba essere esaminato in modo indipendente o collegato a un'indagine esistente.

La funzione principale della fase di triage è la correlazione degli incidenti: identificare gli incidenti correlati e consolidarli in un'unica indagine per evitare la duplicazione del lavoro e lo spreco di risorse. Quando si verifica un nuovo incidente, AWS DevOps Agent lo analizza parallelamente alle indagini attive all'interno di una finestra riepilogativa (in genere 20 minuti). Utilizzando AI-powered l'analisi, esamina fattori come le somiglianze dei componenti, la regione geografica e gli schemi temporali per determinare le relazioni tra gli incidenti.

AWS DevOps L'agente prende una delle tre decisioni:

- Collegato: mette in correlazione l'incidente a un'indagine esistente e invia un messaggio orientativo a tale indagine contestualizzando il nuovo incidente.
- Ignorato: l'incidente soddisfa i criteri di salto definiti in un'abilità e viene automaticamente respinto senza indagini. Per ulteriori informazioni, consulta the section called “DevOps Competenze degli agenti”.
- Procedi: pianifica una nuova indagine indipendente sull'incidente.

Visualizzazione delle decisioni di triage

Quando gli incidenti sono collegati, l'indagine principale riceve un messaggio orientativo contenente i dettagli dell'incidente collegato e il ragionamento della correlazione. Sulla tua app web AWS DevOps Agent Space, vedrai lo stato LINKED insieme a un ragionamento di correlazione che spiega perché gli incidenti sono stati collegati. L'indagine principale mostra un elenco di tutti gli incidenti collegati, che consente di visualizzare l'intera gamma dei problemi correlati oggetto di indagine congiunta. Il tuo sistema di ticket esterno (ServiceNow PagerDuty, ecc.) e il tuo canale di comunicazione (Slack) riceveranno una notifica che indica che l'incidente è stato collegato insieme a un ragionamento di correlazione.

Quando gli incidenti vengono ignorati, l'app web AWS DevOps Agent Space mostra lo stato SKIPPED insieme al motivo che spiega perché l'incidente è stato filtrato. Il sistema di ticket esterno

e il canale di comunicazione ricevono inoltre una notifica che indica che l'incidente è stato ignorato insieme al motivo del salto.

Correzione delle decisioni di triage

Se AWS DevOps Agent collega erroneamente un incidente, puoi scollegarlo manualmente tramite l'app web AWS DevOps Agent Space. In questo modo l'incidente non collegato viene riprogrammato come indagine indipendente. Puoi anche fornire regole di correlazione personalizzate creando una AWS DevOps Agent Skill contenente la tua logica di correlazione e associandola alla fase di triage.

Se AWS DevOps Agent salta erroneamente un incidente, puoi annullarlo manualmente tramite l'app web Agent Space. AWS DevOps In questo modo l'incidente viene riprogrammato per le indagini. Per stabilire quali incidenti vengono ignorati, modifica o disattiva l'abilità che definisce i criteri di salto.

Richiedi supporto umano

AWS DevOps L'agente può connettersi direttamente con AWS Support per semplificare il processo di risposta agli incidenti. Se hai bisogno di ulteriore assistenza da parte di AWS Support, dalla tua app web DevOps Agent Space puoi creare casi di supporto che condividono automaticamente il contesto dell'indagine con i tecnici dell' AWS assistenza, riducendo il tempo necessario per spiegare il problema.

Come funziona

Quando indaga su un incidente, AWS DevOps Agent crea un registro completo delle sue analisi, che include:

- Risultati delle indagini sulla causa principale
- Metriche, log e tracce analizzati
- Revisione delle modifiche al codice e della cronologia di implementazione
- Azioni correttive consigliate
- Cronologia degli eventi e del comportamento del sistema

Puoi inoltrare la tua indagine a AWS Support direttamente dall'app web di AWS DevOps Agent Space. Quando lo fai, AWS DevOps l'agente trasmette automaticamente il registro delle indagini a AWS Support, fornendo al tecnico dell'assistenza un contesto completo sull'indagine senza che tu debba raccogliere e spiegare manualmente i dettagli.

Chiacchierando con AWS Supporto

Dopo aver creato una richiesta di supporto, puoi comunicare con l' AWS assistenza in una finestra di chat separata all'interno della tua app web AWS DevOps Agent Space. Ciò consente di:

- Discutete il problema con i tecnici del AWS Support insieme alla tempistica delle indagini del vostro AWS DevOps agente
- Visualizza sia AWS DevOps l'analisi automatizzata dell'agente che la guida esperta dell' AWS assistenza nella stessa interfaccia
- Condividi senza interruzioni informazioni o chiarimenti aggiuntivi, se necessario

L'esperienza di chat consente di accedere facilmente alle indagini con AWS DevOps l'Agente e alla conversazione di AWS Support, consentendo una collaborazione e una risoluzione più rapide.

Support case language

Quando crei un caso di supporto tramite AWS DevOps Agent, il caso viene creato automaticamente nella lingua configurata nell'impostazione della lingua di risposta Agent Space di Agent Space. Ciò garantisce che la richiesta di supporto venga indirizzata a un tecnico dell'assistenza che parli la lingua preferita.

Ad esempio, se la lingua di Agent Space è impostata sul giapponese, la richiesta di assistenza verrà indirizzata a un Japanese-speaking tecnico dell'assistenza. Se non è configurata alcuna lingua o se la lingua configurata non è supportata da AWS Support per la categoria di casi selezionata, il valore predefinito del caso è l'inglese.

AWS L'assistenza attualmente supporta le seguenti lingue per l'instradamento dei casi: cinese, inglese, francese, giapponese, coreano, portoghese e spagnolo. Per modificare la lingua utilizzata per i casi di supporto, aggiorna l'impostazione della lingua di risposta dell'agente nella configurazione di Agent Space. Per ulteriori informazioni, consulta the section called “Creazione di uno spazio per agenti”.

Requisiti del piano di supporto

La tua capacità di creare e interagire con i casi di supporto tramite AWS DevOps Agent dipende dal tuo piano di AWS supporto. Consulta la guida per l'utente dei piani di supporto per ulteriori informazioni sui tuoi diritti.

Nota I clienti di Basic Support non possono creare casi di supporto tecnico e quindi non possono inoltrare le indagini degli AWS DevOps agenti a Support AWS Developer Support I clienti possono creare casi tramite AWS DevOps Agente, ma devono visitare il Centro assistenza AWS per comunicare con i tecnici dell'assistenza, poiché il supporto per gli sviluppatori non include il supporto basato sulla chat Tutti gli altri piani possono utilizzare l'esperienza di chat integrata all'interno di Agent. AWS DevOps Per i dettagli completi sui diritti ai piani di supporto, inclusi i tempi di risposta e la gravità dei casi disponibili, consulta la Guida per l'utente dei piani di AWS supporto.

Con quali informazioni vengono condivise AWS Supporto

Quando si crea una richiesta di supporto dall'app Web AWS DevOps Agent Space, le seguenti informazioni vengono condivise automaticamente con AWS Support:

- Cronologia dell'indagine: registrazione cronologica dell'analisi dell' AWS DevOps agente
- Informazioni sulle risorse: risorse interessate AWS
- Dati di osservabilità: metriche, log e tracce pertinenti provenienti dagli strumenti di monitoraggio integrati
- Modifiche recenti: implementazioni di codice, modifiche all'infrastruttura e aggiornamenti della configurazione
- Tentativi di riparazione: Actions Agent consigliato AWS DevOps
- Valutazione dell'impatto: portata e gravità dell'incidente

Tutti i dati condivisi con AWS Support seguono le configurazioni di residenza e sicurezza AWS dei dati esistenti. AWS DevOps L'agente condivide solo le informazioni relative all'indagine specifica e rispetta le politiche di governance dei dati dell'organizzazione.

Nozioni di base

Per utilizzare l'integrazione AWS DevOps Agent's AWS Support:

1. Assicurati di avere un piano di AWS Support attivo.
2. Verifica che le autorizzazioni IAM del tuo AWS DevOps agente includano la creazione di casi di supporto (support:CreateCase, support:DescribeCases).
3. Quando AWS DevOps l'agente sta esaminando un problema e hai bisogno AWS di assistenza, scegli Chiedi supporto umano dalla tua app web DevOps Agent Space.
4. Consulta il riepilogo dell'indagine che verrà condiviso con AWS Support.
5. Seleziona la gravità del caso appropriata in base ai diritti concessi al tuo piano di supporto.

6. Invia il caso: l' AWS DevOps agente include automaticamente il registro delle indagini.

La finestra di chat si apre automaticamente e ti consente di iniziare subito a collaborare con AWS Support.

Prevenzione proattiva degli incidenti

AWS DevOps L'agente analizza i modelli delle indagini sugli incidenti per fornire raccomandazioni mirate che migliorano continuamente la postura operativa e prevengono incidenti futuri. Accedi alla prevenzione proattiva degli incidenti tramite la pagina Miglioramenti dell'app Web Operator.

Come funziona la prevenzione proattiva degli incidenti

AWS DevOps L'agente valuta le indagini recenti sugli incidenti per identificare miglioramenti duraturi per prevenire incidenti futuri e accelerare il tempo medio di rilevamento (MTTD). L'agente analizza più incidenti per identificare le raccomandazioni che potrebbero prevenire intere classi di incidenti in futuro, concentrandosi sulle raccomandazioni più efficaci per garantire che siano attuabili.

Per impostazione predefinita, l'agente esegue automaticamente le valutazioni settimanali. Puoi mettere in pausa la pianificazione se preferisci eseguire le valutazioni solo su richiesta. Le valutazioni manuali sono sempre disponibili, il che è utile quando un'indagine recente giustifica una rapida risposta ai miglioramenti consigliati.

L'agente identifica i miglioramenti in quattro categorie, mostrati nella tabella di categorizzazione dei consigli nella pagina Miglioramenti:

- Osservabilità: raccomandazioni per migliorare il monitoraggio, gli avvisi, la registrazione e la visibilità del sistema per rilevare i problemi in modo più rapido e preciso.
- Infrastruttura: consigli per ottimizzare le configurazioni delle risorse, l'ottimizzazione della capacità e la resilienza dell'architettura.
- Governance: raccomandazioni per rafforzare i processi di implementazione, i miglioramenti della pipeline, le pratiche di test e i controlli operativi.
- Ottimizzazione del codice: raccomandazioni per migliorare la qualità del codice delle applicazioni, la gestione degli errori e la resilienza del codice.

Questa categorizzazione ti aiuta a capire dove sono più necessari i miglioramenti operativi e ti consente di dare priorità ai consigli in base alle aree di interesse del tuo team.

Vantaggi

- Prevenzione degli incidenti ricorrenti: affronta le cause alla radice in modo sistematico anziché rispondere ripetutamente allo stesso tipo di problemi
- Riduci la fatica operativa: libera il tuo team da interventi antincendio ripetitivi per concentrarsi sull'innovazione e sui miglioramenti strategici
- Migliora la resilienza del sistema: rafforza l'infrastruttura, l'osservabilità e i processi di implementazione sulla base di dati reali sugli incidenti
- Impara dai modelli storici: sfrutta gli approfondimenti degli incidenti passati per apportare miglioramenti mirati che abbiano il maggiore impatto

Riepilogo degli agenti

Il riepilogo dell'agente nella pagina Miglioramenti dell'app Web fornisce una descrizione dei risultati dell'ultima valutazione degli incidenti recenti. Il riepilogo spiega il numero di indagini sugli incidenti analizzate, quali incidenti sono simili a quelli passati e quali raccomandazioni sono state create o aggiornate con nuove informazioni.

Il riepilogo aiuta a comprendere rapidamente ciò che l'agente ha scoperto durante la sua valutazione più recente ed evidenzia le raccomandazioni più importanti che potrebbero avere il maggiore impatto sulla postura operativa.

Controllo delle valutazioni

È possibile controllare quando AWS DevOps Agent valuta gli incidenti e genera raccomandazioni:

- Esecuzione manuale delle valutazioni: fai clic sul pulsante Esegui ora nella pagina Miglioramenti per avviare immediatamente una valutazione. Ciò è utile quando un'indagine recente giustifica una rapida risposta ai miglioramenti consigliati.
- Interruzione delle valutazioni attive: fate clic sul pulsante Interrompi valutazione nella pagina Miglioramenti per interrompere una valutazione attualmente in corso.

Gestione dei consigli

AWS DevOps L'agente fornisce consigli nella pagina Miglioramenti in cui è possibile esaminarli e gestirli:

- Visualizzazione dei dettagli del consiglio: fai clic su un consiglio per aprire la pagina dei dettagli del consiglio, dove puoi visualizzare ulteriori informazioni sul miglioramento suggerito, inclusi gli incidenti che hanno determinato la raccomandazione, gli impatti previsti e le fasi successive. Per consigli sulle modifiche al codice, puoi anche visualizzare le specifiche pronte per l'uso con l'agente che possono essere consegnate a un agente di codifica per l'implementazione.
- Conserva: fai clic su «Mantieni» per conservare una raccomandazione nel backlog a fini di tracciamento. In questo modo puoi monitorare i miglioramenti che intendi implementare e monitorarne i progressi.
- Ignora: fai clic su «Ignora» per rimuovere una raccomandazione dal backlog. Quando scartate un consiglio, potete fornire una spiegazione in linguaggio naturale del motivo per cui non soddisfa le vostre esigenze. L'agente apprende da questo feedback e lo utilizza per fornire raccomandazioni future, assicurando che diventino più allineate con le priorità e i requisiti operativi nel tempo.
- Implementato: fai clic su «Implementato» per contrassegnare una raccomandazione come completata. Ciò consente di tenere traccia dei miglioramenti applicati e consente all'agente di misurare l'efficacia dei suoi consigli nel tempo.
- Rimozione automatica: i consigli che non sono stati contrassegnati come Mantieni o Implementati possono essere rimossi dopo circa 6 settimane se non fosse stato possibile prevenire nuovi incidenti implementando la raccomandazione. Ciò garantisce che la pagina Miglioramenti si concentri sui miglioramenti più pertinenti per le sfide operative.
- Aggiornamenti dei consigli: i consigli esistenti vengono aggiornati quando vengono rilevati nuovi incidenti che sarebbero stati evitati dalla raccomandazione. Gli aggiornamenti possono modificare la priorità della raccomandazione o perfezionarla in base a nuove informazioni.

Assegnazione di priorità ai consigli

AWS DevOps Agent classifica automaticamente i tuoi consigli in base alla priorità per aiutarti a concentrarti prima sui miglioramenti più efficaci. La classifica considera il contesto specifico del team, i modelli operativi e la gravità dei problemi affrontati da ciascuna raccomandazione.

Come funziona la definizione delle priorità

In ogni ciclo di valutazione, l'agente classifica i consigli attivi (quelli in uno stato proposto o mantenuto) utilizzando una combinazione di:

- AI-powered classificazione: l'agente valuta l'importanza relativa dei migliori consigli in base alla pertinenza della categoria, alla gravità degli incidenti e all'impatto operativo.

- Punteggio deterministico: in caso di arretrati più consistenti, l'agente applica un punteggio di priorità basato sulla frequenza degli incidenti, sui modelli di gravità e sull'attualità per garantire un ordinamento coerente oltre agli articoli di primo livello.

L'elenco classificato viene visualizzato nella pagina Miglioramenti con una posizione numerica (1 indica la priorità più alta). I consigli che sono stati scartati o implementati non vengono classificati.

Personalizzazione delle priorità

Puoi influenzare il modo in cui l'agente classifica le raccomandazioni comunicando le priorità del tuo team tramite l'interfaccia di chat:

- Impostazione delle preferenze di categoria: indica all'agente quali categorie di raccomandazioni sono più importanti per il tuo team (ad esempio, «Diamo priorità ai miglioramenti dell'osservabilità rispetto alle modifiche dell'infrastruttura»). L'agente memorizza queste preferenze e le utilizza nelle future valutazioni del posizionamento.
- Fornire un contesto: condividi informazioni sui progetti imminenti, sui requisiti di conformità o sulle aree di interesse del team. L'agente incorpora questo contesto per determinare a quali raccomandazioni dare la priorità.

Per aggiornare le tue preferenze, usa l'interfaccia di chat e descrivi le priorità del tuo team in linguaggio naturale. L'agente confermerà di aver compreso e applicherà le tue preferenze nel prossimo ciclo di valutazione.

Stabilità del rango

I gradi delle raccomandazioni possono cambiare tra i cicli di valutazione quando:

- Vengono aggiunte nuove raccomandazioni che hanno una priorità maggiore rispetto a quelle esistenti
- Le preferenze dichiarate dal tuo team cambiano
- I nuovi dati sugli incidenti rafforzano o indeboliscono la necessità di una raccomandazione

I consigli che hai già contrassegnato come Keep mantengono la loro posizione nel backlog indipendentemente dalle variazioni di rango, assicurando che il flusso di lavoro non venga interrotto.

Agent-ready specifiche

Per suggerimenti che comportano modifiche al codice o alla configurazione, AWS DevOps Agent può generare una specifica pronta per l'agente. Questa specifica fornisce un documento strutturato che può essere consegnato direttamente a un agente di codifica per l'implementazione.

La specifica include:

- Dichiarazione del problema: un riepilogo del problema e della sua causa principale
- Riepilogo della soluzione: una descrizione di alto livello dell'approccio consigliato
- Archivi di destinazione: i repository specifici in cui è necessario apportare modifiche
- Modifiche al codice: descrizioni dettagliate di cosa è necessario modificare e perché, con percorsi di file specifici e considerazioni sull'implementazione
- Requisiti dei test: quali scenari devono essere testati
- Piano di implementazione: un approccio graduale all'implementazione delle modifiche

Agent-ready le specifiche accelerano l'implementazione fornendo agli agenti di codifica il contesto necessario per apportare modifiche pronte per la produzione senza richiedere un ampio scambio di informazioni con gli ingegneri.

Consigli di implementazione

Per massimizzare il valore delle raccomandazioni proattive sulla prevenzione degli incidenti, prendete in considerazione le seguenti pratiche per agire di conseguenza:

- Utilizzo di specifiche pronte per l'uso con agenti: per suggerimenti relativi alle modifiche al codice, utilizzate le specifiche generate per accelerare l'implementazione consegnandole a un agente di codifica o utilizzandole come guida dettagliata per l'implementazione manuale.
- Aggiungere consigli al backlog dei ticket: copia i consigli nel sistema di ticketing o nello strumento di gestione dei progetti del team per assicurarti che abbiano la priorità rispetto ad altri lavori di ingegneria.
- Dare priorità ai consigli in base all'impatto: concentrati innanzitutto sui consigli che riguardano i tipi di incidenti più frequenti o gravi o quelli che interessano i sistemi critici.
- Monitoraggio dei progressi nell'implementazione: monitora quali raccomandazioni sono state implementate e misurane l'efficacia osservando se incidenti simili diminuiscono nel tempo.

- Coordinamento con i team di sviluppo: condividi le raccomandazioni con i team appropriati che possiedono i sistemi interessati, assicurandoti che dispongano del contesto e delle risorse necessari per implementare i miglioramenti.

DevOps Attività su richiesta

AWS DevOps Agent On Demand Tasks è un assistente conversazionale generativo basato sull'intelligenza artificiale (AI) che consente ai team operativi di interrogare l'architettura delle applicazioni, analizzare lo stato del sistema e accedere agli approfondimenti delle indagini utilizzando il linguaggio naturale. Puoi porre domande sulle AWS risorse, sulle metriche del sistema, sullo stato degli allarmi, sulla cronologia delle implementazioni e sui modelli degli incidenti. La chat fornisce risposte immediate basate sui dati effettivi dell'infrastruttura e delle operazioni, eliminando la necessità di navigare tra più AWS console o strumenti di monitoraggio.

La chat è integrata nell'app web di DevOps Agent Space e fornisce risposte sensibili al contesto in base alla pagina che stai visualizzando. L'interfaccia conserva la cronologia delle conversazioni, consentendoti di continuare le discussioni precedenti e di basarti sulle domande precedenti.

Attività e funzionalità

AWS DevOps Agent On Demand Tasks offre funzionalità complete per aiutarti a gestire e comprendere la tua infrastruttura:

Domande sulle risorse: chiedi informazioni sulle AWS risorse disponibili nel tuo Agent Space, tra cui funzioni Lambda, tabelle DynamoDB, distribuzioni EKS, certificati e configurazioni dell'infrastruttura. Chat può filtrare e analizzare le risorse in base ad attributi come le versioni di runtime, le impostazioni di capacità o lo stato dell'implementazione. Ad esempio, chiedi «Quanti Lambda stanno usando Python 3.8?» o «Ho dei certificati che stanno per scadere?»

Analisi dello stato del sistema: interroga le metriche di integrità del sistema attuali e storiche, tra cui lo stato degli allarmi, i tassi di errore, l'utilizzo della CPU e la disponibilità del servizio. La chat può generare riepiloghi sullo stato di salute relativi a periodi di tempo specifici e identificare le tendenze nel comportamento del sistema. Poni domande come «Quali allarmi sono stati attivati nelle ultime 24 ore?» o «Qualche errore 5xx nell'ultima ora?»

Informazioni dettagliate sulle indagini: accedi alle informazioni delle indagini completate e in corso, tra cui l'analisi delle cause principali, le ipotesi esplorate, i registri esaminati e i modelli di risoluzione. La chat può identificare le cause più comuni degli incidenti e fornire consigli basati su dati storici.

Query «Qual è la causa più comune degli incidenti del mese scorso?» o «Qual è il tempo medio di risoluzione delle indagini completate?»

Gestione delle indagini: quando visualizzi una pagina dei dettagli dell'indagine, guida l'indagine indicando all'agente di concentrarsi su registri specifici, esplorare ipotesi particolari o aggiornare l'analisi delle cause alla radice. Fornisci indicazioni come «Concentrati sui log del servizio di pagamento e aggiorna il tuo RCA» o «Esplora l'ipotesi che il problema sia stato causato dal throttling di DynamoDB».

Elementi della chat: genera report e documenti strutturati, come riepiloghi sullo stato operativo, segnalazioni di errori e analisi degli incidenti. Gli artefatti vengono visualizzati in un pannello dedicato e supportano la modifica delle versioni all'interno della conversazione.

File allegati: allega immagini, documenti e file di codice ai tuoi messaggi in modo che Chat possa analizzarli nel contesto. Ad esempio, allega uno screenshot di una dashboard di allarme, un file di configurazione YAML o un PDF di runbook e chiedi a Chat cosa fare dopo. Vedi Invio di file allegati per i dettagli.

Filtraggio dei consigli: consulta i consigli per la prevenzione degli incidenti con criteri specifici, ad esempio consigli relativi a particolari servizi o problemi operativi. La chat spiega le considerazioni sull'impatto e sull'implementazione di ogni raccomandazione. Ad esempio, «Mostrami consigli per prevenire incidenti che coinvolgono DynamoDB» o «Quali consigli mi aiuterebbero a rilevare più rapidamente i problemi di latenza delle richieste?»

Accedere alla chat

La chat è disponibile come pannello permanente sul lato sinistro dell'app web DevOps Agent Space. La barra laterale sinistra include un pulsante + Nuova chat, una sezione Pagine per accedere a Incidenti, Miglioramenti e Topologia e una sezione Chat che mostra le conversazioni recenti. Scegli Visualizza tutto per vedere la cronologia completa delle conversazioni.

La chat fornisce risposte sensibili al contesto in base a dove accedi:

Topologia: poni domande generali sulle risorse, sull'architettura e sullo stato operativo di Agent Space. La chat offre una visibilità completa su tutti gli account e i servizi connessi. Da questo contesto, puoi interrogare le configurazioni delle risorse, la cronologia di implementazione, le informazioni sulla topologia e le integrazioni degli strumenti di osservabilità.

Risposta agli incidenti: quando visualizzi la pagina di risposta agli incidenti, fai domande sulle tendenze delle indagini, sui tempi di risoluzione e sui modelli degli incidenti in tutto il tuo Agent

Space. Chat può analizzare i dati storici delle indagini per identificare cause comuni e opportunità di miglioramento.

Dettagli dell'indagine: durante la visualizzazione di un'indagine specifica, Chat fornisce risposte sensibili al contesto a tale indagine. Chiedi informazioni sui registri esaminati, sulle ipotesi esplorate, sulle conclusioni sulle cause principali e sui piani di mitigazione. Puoi anche fornire suggerimenti orientativi per orientare l'attenzione dell'indagine.

Prevenzione: dalla pagina sulla prevenzione, consulta le raccomandazioni utilizzando i filtri, scopri perché sono state formulate le raccomandazioni ed esplora gli approcci di implementazione. La chat ti aiuta a stabilire le priorità e a comprendere l'impatto delle raccomandazioni sulla prevenzione degli incidenti.

L'interfaccia di chat rimane disponibile quando si passa da una pagina all'altra, ma il contesto cambia per fornire informazioni pertinenti alla visualizzazione corrente. Quando inizi una nuova conversazione, questa inizia senza un contesto precedente. Quando continui una conversazione esistente, Chat conserva la cronologia completa delle conversazioni per le domande successive.

Context-aware risposte

Chat adatta le sue risposte in base alla pagina visualizzata nell'app web DevOps Agent Space. Questa consapevolezza del contesto ti assicura di ricevere informazioni pertinenti senza dover specificare a quale indagine o ambito di risorse stai chiedendo.

Quando visualizza la pagina dei dettagli di un'indagine, Chat capisce automaticamente che stai chiedendo informazioni su quella specifica indagine. Domande come «Quali registri hai esaminato?» o «Quali ipotesi hai esplorato?» fai riferimento all'indagine attualmente visualizzata. Quando fornisci un input fondamentale, Chat lo applica all'indagine in corso e, se del caso, crea una nuova versione della causa principale.

Nella pagina sulla prevenzione, Chat capisce che sei interessato ai consigli sulla prevenzione degli incidenti. Le query filtrano e analizzano automaticamente i consigli all'interno del contesto di Agent Space. Il sistema riconosce se stai chiedendo consigli generali o dettagli specifici.

Quando si accede a Chat dalla pagina Topologia, Chat offre un'ampia visibilità su tutte le risorse, le metriche e i dati storici presenti nell'Agent Space. Puoi chiedere informazioni su qualsiasi risorsa, servizio o problema operativo senza specificare il contesto dell'indagine o della raccomandazione.

Questa consapevolezza del contesto elimina la necessità di specificare ripetutamente a quale indagine, raccomandazione o ambito di risorse si fa riferimento, creando un flusso di conversazione più naturale.

Gestione delle conversazioni

La chat conserva la cronologia delle conversazioni per consentirti di continuare le discussioni precedenti e fare riferimento alle domande precedenti.

Creazione di nuove conversazioni: fai clic sul pulsante «Nuova sessione» nel pannello della chat per iniziare una nuova conversazione senza il contesto precedente. Le nuove conversazioni non riprendono le informazioni delle chat precedenti, consentendoti di porre domande non correlate senza confusione.

Accesso alla cronologia delle conversazioni: fai clic su «Cronologia» per visualizzare tutte le conversazioni precedenti all'interno di Agent Space. Le conversazioni sono organizzate cronologicamente con timestamp e testo di anteprima. La cronologia delle conversazioni viene conservata per 90 giorni ed è privata dell'account utente all'interno di Agent Space.

Conversazioni continue: seleziona una conversazione dalla cronologia per riprendere da dove l'avevi interrotta. La chat mantiene il contesto completo dei messaggi precedenti, consentendoti di porre domande di follow-up che fanno riferimento a parti precedenti della conversazione. Quando cambi pagina durante la visualizzazione di una conversazione, il contesto della conversazione rimane invariato, ma il contesto specifico della pagina viene aggiornato in base alla posizione corrente.

Tieni presente che la cronologia delle conversazioni è isolata all'interno di ogni Agent Space. Le conversazioni in un Agent Space non sono visibili o accessibili da altri Agent Space. Questo isolamento garantisce che le informazioni sensibili rimangano compartimentate in base ai confini organizzativi.

Generazione di artefatti

AWS DevOps L'agente supporta gli artefatti della chat, ovvero documenti strutturati e con versioni diverse generati dall'agente durante una conversazione. Gli artefatti forniscono un pannello interattivo dedicato nell'interfaccia utente della chat per la revisione e la modifica AI-generated dei contenuti, come report operativi, riepiloghi degli errori e valutazioni dello stato di salute.

Puoi richiedere artefatti da qualsiasi pagina dell'app web Agent Space. DevOps Chat utilizza il contesto della pagina corrente per definire il contenuto degli artefatti.

Come funzionano gli artefatti

Quando chiedi a Chat di creare o aggiornare contenuti, Chat genera un artefatto, in genere un documento formattato, e lo visualizza nel pannello degli artefatti accanto alla conversazione.

Genera: invia una richiesta in linguaggio naturale per creare un rapporto o un documento. Ad esempio, chiedi «Genera un rapporto settimanale sullo stato di funzionamento per il mio Agent Space» o «Mostrami un rapporto per i miei errori 4xx della settimana scorsa».

Recensione: l'artefatto appare in un pannello dedicato accanto alla conversazione. Puoi rivedere il contenuto completo continuando a interagire con Chat.

Modifica: richiedi modifiche all'artefatto tramite Chat. Ad esempio, chiedi «Aggiungi una sezione sugli avviamenti a freddo Lambda» o «Aggiorna il rapporto per includere i dati del mese scorso». Chat crea una nuova versione dell'artefatto con le modifiche richieste.

Invio di file allegati

Puoi allegare file ai tuoi messaggi di chat in modo che Chat possa leggerli insieme alla tua domanda. Usa gli allegati per condividere ciò che stai guardando (uno screenshot di una dashboard o di un allarme, un file di configurazione, un codice sorgente, un runbook operativo) e chiedi all'agente di ragionare direttamente al riguardo.

I file rientrano nell'ambito del tuo Agent Space: non sono visibili da altri Agent Spaces e l'accesso è limitato dalle stesse autorizzazioni IAM che valgono per il resto di Chat. I file vengono caricati nello spazio di archiviazione gestito di Agent Space non appena vengono allegati.

Come allegare file

È possibile aggiungere file a un messaggio in tre modi:

- Scegli l'icona di caricamento nella barra degli strumenti di input della chat e seleziona uno o più file dal tuo dispositivo.
- Trascina e rilascia uno o più file nell'area di immissione della chat.
- Incolla un'immagine direttamente dagli appunti, ad esempio dopo aver scattato uno screenshot.

Ogni file allegato appare come un chip nell'input della chat con un indicatore di avanzamento del caricamento. Per visualizzare l'anteprima di un file, scegli il relativo chip. Per rimuovere un file, scegli la X sul chip. Il pulsante Invia rimane disabilitato durante il caricamento di qualsiasi file allegato.

Tipi di file supportati

Chat accetta le seguenti tre categorie di file:

- Immagini — `png`,`jpeg`,`jpg`,`gif`, `webp`
- Documenti — `pdf`,`csv`,`doc`,`docx`,`xls`,`xlsx`,`html`,`txt`, `md`
- File di testo e di codice — `json` `yaml`,`yml`,`xml`, `js`,`ts`,`py`,`java`,`rb`,`go`,`rs`,`sh`,`bash`,`log`,`cfg`,`ini`, `toml`

I file che non rientrano in queste categorie vengono rifiutati prima del caricamento.

Limits

I seguenti limiti si applicano a ciascun messaggio:

Limite	Valore
Dimensione massima dei file	3,75 MB
Allegati per messaggio (qualsiasi combinazione di tipi)	20
Di questi, documenti binari (PDF, DOC, DOCX, XLS, XLSX)	fino a 5

Inoltre, il testo del messaggio e il contenuto degli allegati devono rientrare insieme nella finestra contestuale per messaggio del modello. Se un messaggio e i relativi allegati sono troppo grandi, Chat lo rifiuta e ti chiede di ridurre le dimensioni o il numero di allegati prima dell'invio.

Casi d'uso

Metodi comuni per utilizzare i file allegati con l'agente: DevOps

- Allega uno screenshot di un pannello di allarme o di errore e chiedi a Chat di interpretare cosa non va e dove cercare successivamente.
- Allega il codice sorgente del servizio e chiedi a Chat di esaminare la modifica, suggerire correzioni o spiegarne il comportamento.

- Allega un file di configurazione (ad esempio, una configurazione YAML, JSON o TOML) e chiedi a Chat di risolvere il motivo per cui una distribuzione, un allarme o un'integrazione si comporta male.
- Allega un runbook operativo o un PDF di report post-incidente e chiedi a Chat di convertirlo in una competenza: l'agente estrae la procedura e la salva nell'Agent Space in modo che le indagini future possano applicarla automaticamente.

Query di esempio

Gli esempi seguenti mostrano i tipi di domande che puoi porre a Chat. Questi esempi sono organizzati per caso d'uso e contesto.

Interrogazioni sulla generazione di Artifact

Da qualsiasi pagina dell'app web DevOps Agent Space:

- Genera un riepilogo dello stato operativo settimanale per my Agent Space
- Crea un rapporto di tutti gli errori 4xx della settimana scorsa
- Crea un rapporto di riepilogo degli incidenti degli ultimi 30 giorni
- Crea un riepilogo dell'attività di allarme per il servizio di pagamento di questa settimana
- Genera un rapporto sulla cronologia delle implementazioni degli ultimi 7 giorni
- Riassumi tutte le raccomandazioni aperte in un rapporto

Richieste di informazioni sulle risorse

Da qualsiasi pagina dell'app web DevOps Agent Space:

- Quante funzioni Lambda usano Python 3.8?
- Ho dei certificati che stanno per scadere?
- Elenca tutte le tabelle DynamoDB con fatturazione su richiesta
- Mostrami i cluster EKS in produzione
- Quali funzioni Lambda non sono state implementate negli ultimi 90 giorni?
- Elenca i bucket S3 senza il controllo delle versioni abilitato
- Su quali istanze RDS è in esecuzione la versione X del database?

Domande sullo stato del sistema

Dalle pagine Topology o Incident Response:

- Quali allarmi sono stati attivati nelle ultime 24 ore?
- Qualche errore 5xx nell'ultima ora?
- Mostrami le tendenze degli errori Lambda per il servizio di pagamento
- Qual è l'utilizzo della CPU per il mio cluster ECS?
- I miei sistemi di bilanciamento del carico presentano obiettivi non idonei?
- Mostrami gli eventi di limitazione di API Gateway di ieri
- Quali servizi hanno registrato il tasso di errore più elevato la settimana scorsa?
- Datemi un rapporto sullo stato di salute generale delle ultime 24 ore

Domande sullo strumento di osservabilità

Dalla topologia:

- Elenca i gruppi di log Splunk
- Mostrami le metriche di Prometheus e le relative soglie di allarme
- Quali monitor Datadog sono configurati per questo servizio?
- Elenca le politiche di avviso di New Relic
- Mostrami le configurazioni della dashboard di Dynatrace

Domande e approfondimenti sulle indagini

Dalla pagina Incident Response:

- Qual è la causa più comune di incidenti del mese scorso?
- Qual è il tempo medio di risoluzione delle indagini completate?
- Riassumi le indagini della settimana scorsa e il relativo RCA
- Quanti incidenti sono stati causati dal throttling di DynamoDB?
- Mostrami le tendenze delle indagini nell'ultimo trimestre
- Quali servizi presentano gli incidenti più frequenti?

Domande sui dettagli dell'indagine

Dalla pagina dei dettagli dell'indagine:

- Quali registri hai esaminato?
- Quali ipotesi hai esplorato?
- Quanto è rischiosa l'azione mitigante che proponete?
- Qual era la cronologia degli eventi durante questo incidente?
- Perché hai concluso che questa era la causa principale?
- Quali prove supportano la tua analisi della causa principale?
- Chi ha fornito le indicazioni necessarie durante le vostre indagini?
- Dammi un riepilogo di questa indagine sull'incidente

Domande sulla gestione delle indagini

Dalla pagina dei dettagli dell'indagine:

- Concentrati sui registri del servizio di pagamento tra le 14:00 e le 15:00 UTC e aggiorna il tuo RCA
- Esplora l'ipotesi che la limitazione di DynamoDB abbia causato il problema
- Controlla la configurazione del cluster ECS per vedere se ciò ha causato l'allarme
- Controlla solo i log delle ultime 2 ore, non dell'intera giornata
- Esamina il picco di errori alle 15:00
- Guarda i log dell'API Gateway anziché i log Lambda

Domande di consigli sulla prevenzione

Dalla pagina Prevenzione:

- Quali sono i miei 3 consigli principali per la prevenzione degli incidenti?
- Mostrami consigli per prevenire incidenti che coinvolgono DynamoDB
- Quali consigli mi aiuterebbero a rilevare più rapidamente i problemi di latenza delle richieste?
- Elenca i miglioramenti dell'osservabilità che potrebbero prevenire incidenti simili
- Mostrami i consigli sull'infrastruttura per il servizio di pagamento
- Quali raccomandazioni hanno il maggiore impatto sulla resilienza del sistema?

Attivazione della chat nell'area riservata agli agenti

La chat è disponibile in tutte le app web di DevOps Agent Space. Il processo di configurazione dipende dal fatto che si disponga di un Agent Space nuovo o esistente.

Nuovi spazi per agenti

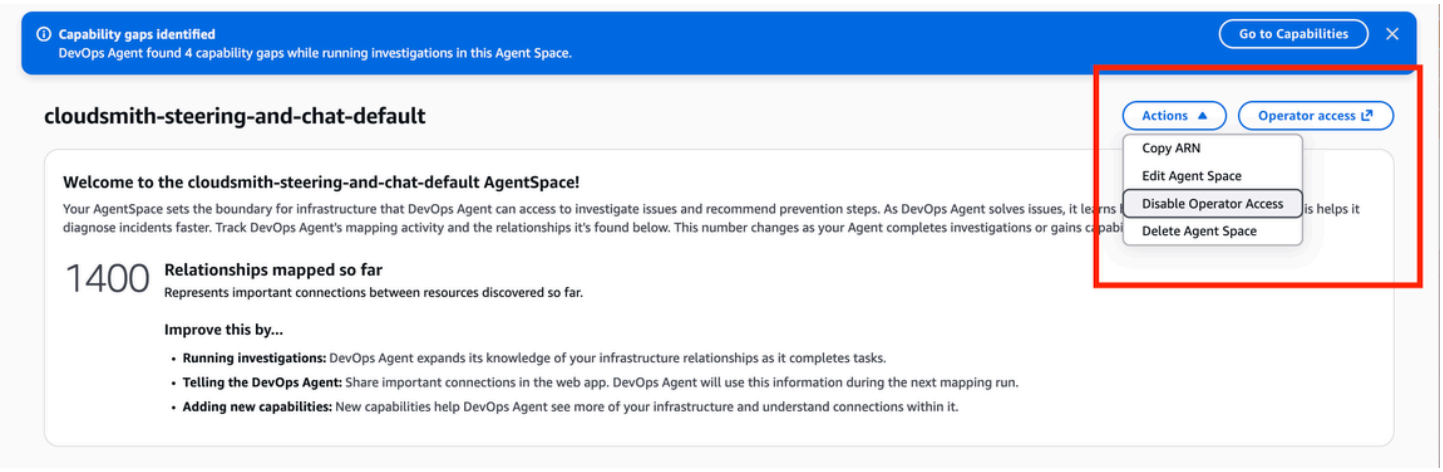
La chat viene abilitata automaticamente quando crei un nuovo Agent Space. Non è richiesta alcuna configurazione aggiuntiva o configurazione delle autorizzazioni IAM. Dopo aver configurato l'app web DevOps Agent Space, Chat è immediatamente disponibile come pannello persistente sul lato sinistro di qualsiasi pagina.

Agent Spaces esistenti

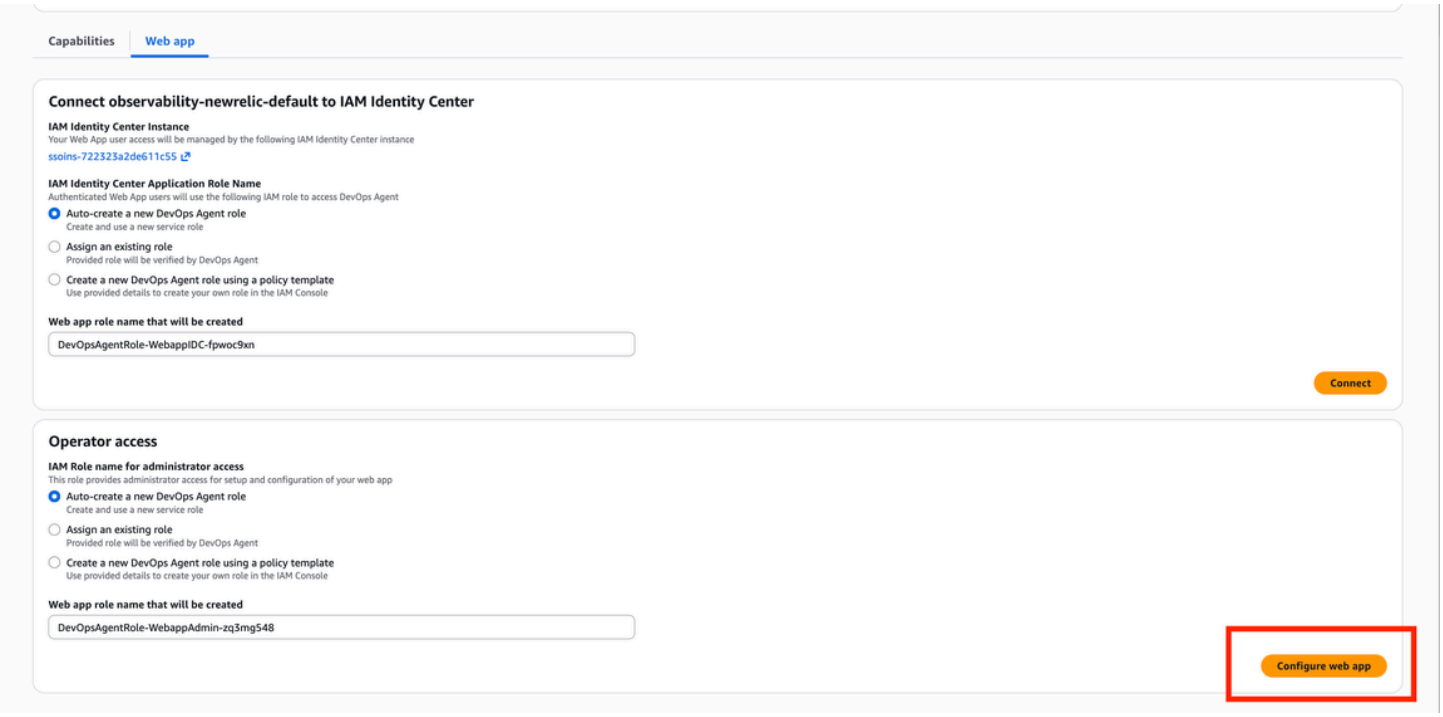
Se hai creato Agent Space prima del rilascio di Chat, devi abilitare le autorizzazioni IAM richieste. Sono disponibili due opzioni:

Opzione 1: revoca e riattiva l'accesso all'app dell'operatore

Accedi alla Console di amministrazione dell' AWS DevOps agente, individua il menu a discesa Azione nell'angolo in alto a destra e disabilita la configurazione corrente dell'accesso dell'operatore.



Quindi abilita l'opzione di creazione automatica per l'accesso dell'operatore.



Ciò applica automaticamente le autorizzazioni IAM richieste per Chat insieme a tutte le altre autorizzazioni attuali dell'operatore.

Opzione 2: aggiungere manualmente le autorizzazioni IAM

Aggiungi le seguenti autorizzazioni IAM al tuo ruolo di accesso operatore esistente:

- `aidevops:ListChats`— Visualizza la cronologia delle conversazioni in chat
- `aidevops:CreateChat`— Crea nuove conversazioni in chat
- `aidevops:SendMessage`— Invia messaggi e ricevi risposte

Accedi alla console AWS IAM, individua il tuo ruolo di operatore DevOps agente e aggiungi queste autorizzazioni alla politica del ruolo. La chat diventa disponibile immediatamente dopo l'aggiunta delle autorizzazioni.

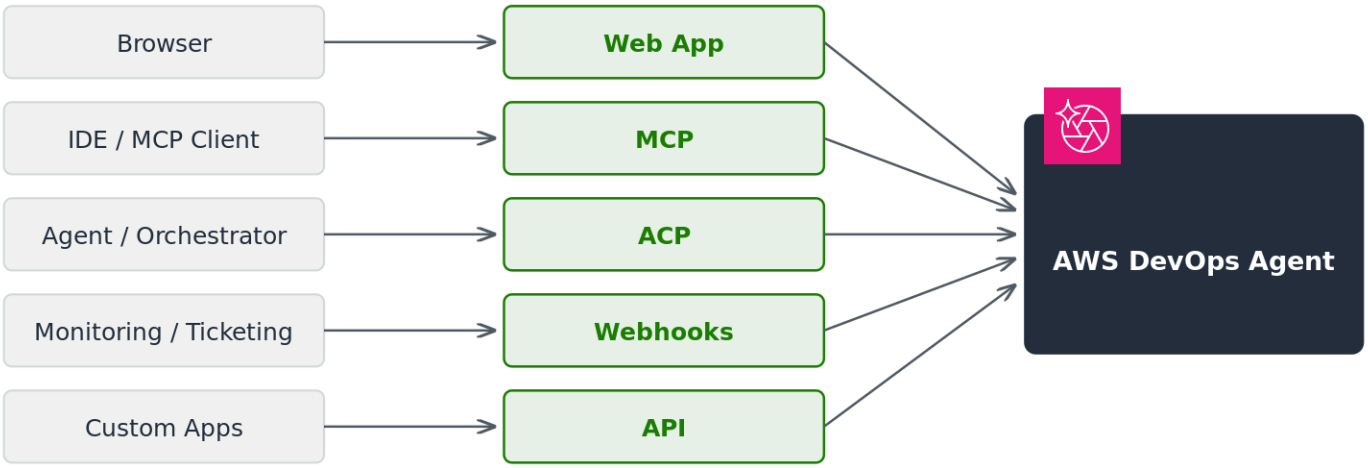
Dopo aver completato una delle due opzioni, aggiorna l'app web di DevOps Agent Space e il pannello della chat viene visualizzato sul lato sinistro di qualsiasi pagina.

Interfacciamento con l'agente DevOps

AWS DevOps Agent supporta cinque metodi di accesso: la console dell'app Web, l'integrazione con Model Context Protocol (MCP), l'integrazione con Agent Client Protocol (ACP), i webhook per

l'automazione basata sugli eventi e l'accesso diretto alle API. Scegliete il metodo più adatto al vostro flusso di lavoro e ai vostri requisiti tecnici.

Il diagramma seguente illustra questi metodi di accesso e il modo in cui si connettono al servizio DevOps Agent.



DevOps App web dell'agente

L'app Web è l'interfaccia principale per DevOps Agent. Utilizza la chat conversazionale per indagare sugli incidenti, interrogare l'infrastruttura e gestire i consigli. Per ulteriori informazioni, consulta the section called "Cos'è un'app Web per DevOps agenti?".

Integrazione con Model Context Protocol (MCP)

È possibile accedere alle funzionalità AWS DevOps dell'agente direttamente dai MCP-compatible client e dagli IDE. Utilizzate il server AWS MCP per connettervi. È possibile analizzare gli incidenti, ottimizzare i costi, rivedere l'architettura e mappare la topologia senza uscire dall'ambiente di sviluppo.

Per gli utenti Kiro, una potenza dedicata di aws-devops-agent è disponibile nel repository Kiro powers. Questa alimentazione collega Kiro ad Agent tramite il server MCP. AWS DevOps AWS Fornisce intelligenza AI-powered operativa direttamente nel tuo IDE.

Per gli utenti di Claude Code, il sample-aws-devops-agent-claude-plugin fornisce un plug-in preconfigurato che collega Claude Code ad Agent tramite il server MCP. AWS DevOps AWS

Integrazione con Agent Client Protocol (ACP)

È possibile richiamare AWS DevOps Agent a livello di codice utilizzando l'Agent Client Protocol (ACP). Per un'implementazione di esempio, consulta il repository sample-aws-devops-agent-acp-mcp su. GitHub

Webhook

I webhook consentono ai sistemi esterni di avviare automaticamente le indagini degli agenti. AWS DevOps I sistemi esterni, come le piattaforme di ticketing e gli strumenti di monitoraggio, possono inviare richieste HTTP in caso di incidenti. Per ulteriori informazioni, consulta the section called "Richiamo DevOps dell'agente tramite Webhook".

AWS DevOps API dell'agente

AWS DevOps Agent fornisce API per l'accesso programmatico alle funzionalità degli agenti. È possibile creare e gestire Agent Spaces, avviare indagini e recuperare i risultati. Per ulteriori informazioni, consulta l'AWS DevOps Agent API Reference.

Configurazione delle funzionalità per AWS DevOps Agente

AWS DevOps Le funzionalità degli agenti estendono le funzionalità dell'agente collegandolo agli strumenti e all'infrastruttura esistenti. Configura queste funzionalità per consentire un'indagine completa sugli incidenti, flussi di lavoro di risposta automatizzati e una perfetta integrazione con il tuo DevOps ecosistema.

Le seguenti funzionalità ti aiutano a massimizzare l'efficacia del tuo DevOps agente:

- AWS EKS Access Setup: abilita l'introspezione dei cluster Kubernetes, dei pod log e degli eventi del cluster per ambienti EKS pubblici e privati
- Integrazione con Azure: collega le sottoscrizioni di Azure e le DevOps organizzazioni di Azure per esaminare le risorse di Azure e correlare le distribuzioni di Azure agli incidenti DevOps
- CI/CD Integrazione della pipeline: Connect GitHub e GitLab pipeline per correlare le implementazioni agli incidenti e tenere traccia delle modifiche al codice durante le indagini
- Connessioni al server MCP: estendi le capacità di indagine collegando strumenti di osservabilità esterni e sistemi di monitoraggio personalizzati tramite Model Context Protocol
- Multi-Account AWS Accesso: configura AWS gli account secondari per esaminare le risorse dell'intera organizzazione durante la risposta agli incidenti
- Integrazione delle fonti di telemetria: collega piattaforme di monitoraggio come Datadog, Dynatrace, Grafana, New Relic e Splunk per un accesso completo ai dati di osservabilità
- Integrazione di ticket e chat: Connect ServiceNow e Slack per automatizzare i flussi di lavoro di risposta agli incidenti e consentire la collaborazione in team PagerDuty
- Configurazione Webhook: consenti ai sistemi esterni di attivare automaticamente le indagini degli DevOps agenti tramite richieste HTTP. Per informazioni dettagliate sulla configurazione dei webhook, sui metodi di autenticazione e sul formato delle richieste, vedere. the section called "Richiamo DevOps dell'agente tramite Webhook"
- EventBridge Integrazione con Amazon: incorpora AWS DevOps Agent in applicazioni basate sugli eventi indirizzando gli eventi del ciclo di vita di indagine e mitigazione verso obiettivi Amazon EventBridge

Puoi configurare ogni funzionalità in modo indipendente in base alle esigenze specifiche del tuo team e allo stack di strumenti esistente. Inizia con le integrazioni più importanti per il flusso di lavoro di risposta agli incidenti, quindi espandi le funzionalità aggiuntive se necessario.

Migrazione dall'anteprima pubblica alla disponibilità generale

Se hai utilizzato AWS DevOps Agent durante l'anteprima pubblica, devi aggiornare i ruoli IAM prima della versione GA. Questa guida illustra l'aggiornamento dei ruoli di monitoraggio e dei ruoli di operatore nei tuoi account.

Cosa sta cambiando

1. Le cronologie delle chat su richiesta durante l'anteprima non sono più accessibili
2. Le nuove politiche gestite sostituiscono le politiche disponibili durante l'anteprima
3. Agent Spaces potrebbe avere un ambito di accesso alle applicazioni IAM Identity Center obsoleto

Cronologia delle chat su richiesta dall'anteprima pubblica

La versione GA introduce misure di sicurezza aggiuntive per rafforzare i controlli di accesso alle cronologie delle chat. A seguito di queste modifiche, le cronologie delle chat su richiesta del periodo di anteprima pubblica (prima del 30 marzo 2026) non sono più accessibili. I diari di indagine e i risultati creati durante l'anteprima pubblica non sono interessati. Questa modifica si applica solo alle conversazioni in chat su richiesta.

Nuove politiche gestite

Per GA, AWS fornisce nuove politiche gestite che sostituiscono le politiche dell'era di anteprima:

Tipo di ruolo	Rimuovi	Add (Aggiungi)
Monitoraggio	Policy gestita di `AIOpsAssistantPolicy`	Policy gestita di `AIDevOpsAgentAccessPolicy`
Operatore (IAM e IDC)	Politica in linea	Policy gestita di `AIDevOpsOperatorAppAccessPolicy`

Inoltre, i ruoli dell'operatore richiedono politiche di fiducia aggiornate e i ruoli dell'operatore IDC richiedono una nuova politica in linea.

Prerequisiti

- Accesso agli AWS account in cui sono configurati i ruoli di DevOps agente (account primari e tutti gli account secondari)
- Autorizzazioni IAM per modificare ruoli, politiche e relazioni di fiducia
- L'ID Agent Space, l'ID AWS dell'account e la regione (visibili nella console dell' DevOps agente)

Fase 1: Aggiornare i ruoli di monitoraggio

Aggiorna il ruolo di monitoraggio nel tuo account principale e in ogni account secondario. Questi sono i ruoli di Primary/Secondary origine configurati nella scheda Capacità nello spazio degli agenti (esempio di primary/secondary ruolo:`DevOpsAgentRole-AgentSpace-3xj2396z`).

1. Nella console dell' DevOps agente, vai al tuo Agent Space e scegli la scheda Funzionalità.
2. Trova il ruolo di monitoraggio per le tue Primary/Secondary Fonti (ad esempio`DevOpsAgentRole-AgentSpace-3xj2396z`) e scegli Modifica.
3. In Politiche di autorizzazione, rimuovi la politica `AIOpsAssistantPolicy` AWS gestita.
4. Scegli Aggiungi autorizzazioni, Allega politiche e allega la politica `AIDevOpsAgentAccessPolicy` gestita.
5. Modifica la politica in linea e sostituisci il suo contenuto con quanto segue, sostituendo l'ID del tuo account:

```
{
    "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. La politica di fiducia per il ruolo di monitoraggio non richiede modifiche. Verifica che corrisponda a quanto segue:

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

- Ripeti i passaggi da 2 a 6 per il ruolo di monitoraggio in ogni account secondario.

Fase 2: Aggiornare il ruolo dell'operatore (IAM)

1. Nella console dell' DevOps agente, scegli la scheda Accesso e trova il ruolo dell'operatore.
2. Nella console IAM, rimuovi la policy in linea esistente dal ruolo dell'operatore.
3. Scegli Aggiungi autorizzazioni, Allega politiche e allega la politica `AIDevOpsOperatorAppAccessPolicy` gestita.
4. Scegli la scheda Relazioni di fiducia e scegli Modifica politica di fiducia. Sostituisci la politica di fiducia con la seguente, sostituendo l'ID dell'account, la regione e l'ID di Agent Space:

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

Fase 3: Aggiornamento dei ruoli degli operatori (IDC)

Se utilizzi IAM Identity Center with DevOps Agent, aggiorna ogni ruolo di operatore IDC.

1. Nella console IAM, vai su Ruoli e cerca per **`WebappIDC`** trovare i ruoli di DevOps Agent IDC (ad esempio,`DevOpsAgentRole-WebappIDC-<id>`).
2. Per ogni ruolo IDC:

a. Rimuovi la politica in linea esistente.

b. Scegli Aggiungi autorizzazioni, Allega politiche e allega la politica `AIDevOpsOperatorAppAccessPolicy` gestita.

c. Scegli la scheda Relazioni di fiducia e scegli Modifica politica di fiducia. Sostituisci la politica di fiducia con la seguente, sostituendo l'ID dell'account, la regione e l'ID di Agent Space:

```
{
    "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. Crea una nuova politica in linea con le seguenti autorizzazioni, sostituendo l'ID del tuo account:

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

Ricollega IAM Identity Center (se applicabile)

Gli Agent Spaces creati durante l'anteprima pubblica possono avere un'applicazione IAM Identity Center configurata con un ambito di accesso obsoleto. Per GA, l'ambito corretto è **`aidevops:read_write`**. Se la tua applicazione IAM Identity Center ha l'ambito precedente (**`awsaidevops:read_write`**), devi disconnettere e ricollegare IAM Identity Center.

Come verificare l'ambito dell'applicazione IAM Identity Center

Esegui il seguente comando AWS CLI per controllare l'ambito sulla tua applicazione IAM Identity Center. Puoi trovare l'ARN dell'applicazione nella console IAM Identity Center alla voce Applicazioni.

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

L'output dovrebbe mostrare l'ambito **`aidevops:read_write`**corretto:

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

Se l'ambito viene visualizzato **`awsaidevops:read_write`**, è obsoleto. Segui i passaggi seguenti per aggiornarlo.

Come riconnettere IAM Identity Center

L'ambito di accesso su un'applicazione IAM Identity Center AWS gestita non può essere aggiornato direttamente. È necessario disconnettersi e riconnettersi:

1. Nella console dell' AWS DevOps agente, vai al tuo Agent Space e scegli la scheda Accesso.
2. Scegli Disconnect accanto alla configurazione di IAM Identity Center.
3. Conferma la disconnessione.
4. Scegli Connect per configurare nuovamente IAM Identity Center. Il servizio crea una nuova applicazione IAM Identity Center con l'ambito corretto.
5. Riassegna utenti e gruppi alla nuova applicazione nella console IAM Identity Center.

> **Important**
>
> La disconnessione rimuove la chat dei singoli utenti e la cronologia degli artefatti associati agli account utente IAM Identity Center. Gli utenti dovranno effettuare nuovamente l'accesso dopo la riconnessione.

Verifica

Dopo aver completato tutti i passaggi:

1. Tornate alla console dell' DevOps agente e verificate che non compaiano errori di autorizzazione nella scheda Agent Space Access.

2. Testa l'app web dell'operatore per confermare che si carichi e funzioni correttamente.
3. Se utilizzi IDC, verifica che gli utenti possano autenticarsi e accedere all'esperienza dell'operatore.

Risoluzione dei problemi

Errori di autorizzazione negata dopo la migrazione

- Verifica che sia `AIOpsAssistantPolicy` stato rimosso e `AIDevOpsAgentAccessPolicy` sia associato ai ruoli di monitoraggio.
- Verifica che le vecchie politiche in linea siano state rimosse e che `AIDevOpsOperatorAppAccessPolicy` siano associate ai ruoli dell'operatore.
- Verifica che le politiche di fiducia degli operatori includano`sts:TagSession`.
- Conferma di aver sostituito tutti i valori segnaposto (`<account-id>`,`<region>`,`<agentspace-id>`) con valori effettivi.

Gli account secondari non funzionano

- Il ruolo di monitoraggio di ogni account secondario deve essere aggiornato in modo indipendente. Accedi a ciascun account e ripeti il passaggio 1.

Errori di autenticazione IDC

- Verifica che la policy di fiducia di IDC includa sia l'`sts:TagSession`istruzione`sts:AssumeRole`/ che l'istruzione. `TrustedIdentityPropagation`
- Conferma la politica in linea con `sso:ListInstancessso:DescribeInstance`, ed `identitystore:DescribeUser` è stata creata.

Manca la cronologia delle chat su richiesta dopo la migrazione

- Le cronologie delle chat su richiesta del periodo di anteprima pubblica non sono accessibili dopo il rilascio della versione GA. Questo è un comportamento previsto dovuto alle misure di sicurezza avanzate introdotte in GA. Le riviste investigative e i risultati dell'anteprima pubblica non sono interessati.

AWS Configurazione dell'accesso EKS

Puoi consentire ad AWS DevOps Agent di esaminare i problemi nei tuoi cluster Amazon EKS eseguendo `kubectl` comandi di sola lettura su cluster pubblici e privati. Puoi connettere un numero qualsiasi di cluster EKS allo stesso Agent Space.

Una volta connesso, l'agente può aiutare a diagnosticare i problemi operativi nei cluster, descrivendo le risorse, recuperando i log dei pod, ispezionando gli eventi del cluster, controllando lo stato dei nodi e altro ancora. L'agente non può creare, modificare o eliminare alcuna risorsa nel cluster.

Prerequisiti

Prima di configurare l'accesso EKS, assicuratevi che la modalità di autenticazione del cluster EKS includa l'API EKS. Puoi verificarlo nella scheda Accesso nella console Amazon EKS. Se la modalità non include l'API EKS, seleziona una modalità che lo includa prima di procedere.

Configurazione

Questi passaggi devono essere completati dalla console Amazon EKS per ogni cluster per cui desideri creare una voce di accesso. Puoi trovare l'ARN del tuo ruolo IAM nel tuo Agent Space (vedithe section called "Creazione di uno spazio per agenti") in Capabilities > Cloud > Primary Source > Modifica.

1. Vai alla scheda Accesso. Se la modalità di autenticazione dice già EKS API, puoi aggiungere voci di accesso. Altrimenti, seleziona una modalità che includa l'API EKS.
2. Dalla scheda Accesso, crea una nuova voce di accesso IAM. Copia l'ARN del ruolo IAM di origine cloud principale e inseriscilo come principale IAM per la voce di accesso. Fare clic su Avanti.
3. Seleziona la policy di AIOps AssistantPolicy accesso AWS Managed Amazon e seleziona Cluster per l'ambito di accesso. (In alternativa, se desideri che l'agente acceda solo a determinati namespace, seleziona i namespace Kubernetes desiderati). Fai clic su Aggiungi politica, quindi su Avanti.
4. Rivedi le modifiche e conferma che sono stati scelti la politica di accesso e il ruolo IAM corretti, quindi crea la voce di accesso facendo clic su «Crea».

Per verificare che l'accesso EKS sia stato configurato correttamente, accedi all'app Operator e avvia una nuova indagine, ponendo all'agente una domanda sul tuo cluster, ad esempio «elenca tutti i pod nel namespace predefinito» o «mostrami gli eventi recenti nel mio cluster».

Risoluzione dei problemi

Se l'agente non riesce a raggiungere il tuo cluster, verifica che l'accesso utilizzi l'ARN del ruolo IAM corretto mostrato nella finestra di dialogo di configurazione e che la policy di AIOps AssistantPolicy accesso Amazon sia allegata.

Connessione ad Azure

L'integrazione con Azure consente all' AWS DevOps agente di esaminare le risorse nell'ambiente Azure e di correlare le distribuzioni della DevOps pipeline di Azure con gli incidenti operativi. Connettendo Azure, l'agente ottiene visibilità sull'infrastruttura di Azure e può eseguire l'analisi della causa principale su entrambe le risorse di Azure. AWS

L'integrazione con Azure è costituita da due funzionalità indipendenti:

- Risorse di Azure: consente all'agente di scoprire e analizzare le risorse cloud di Azure come macchine virtuali, cluster Azure Kubernetes Service (AKS), database e componenti di rete. L'agente usa Azure Resource Graph per interrogare le tue risorse durante le indagini sugli incidenti.
- Azure DevOps: consente all'agente di accedere agli DevOps archivi di Azure e alla cronologia di esecuzione della pipeline. L'agente può correlare le modifiche e le distribuzioni del codice agli incidenti per aiutare a identificare le potenziali cause principali.

Ogni funzionalità è registrata a livello di AWS account e può quindi essere associata a singoli Agent Spaces.

Metodi di registrazione

AWS DevOps L'agente supporta due metodi per la connessione ad Azure:

- Consenso dell'amministratore: un flusso semplificato basato sul consenso in cui autorizzi l'applicazione AWS DevOps Agent Entra nel tuo tenant di Azure. Nella console, viene visualizzata come opzione di consenso dell'amministratore. Questo metodo richiede l'accesso con un account che dispone dell'autorizzazione per eseguire il consenso dell'amministratore in Microsoft Entra ID.
- Registrazione delle app: un approccio autogestito in cui è possibile creare un'applicazione Entra personalizzata con credenziali di identità federate utilizzando Outbound Identity Federation. Nella console, questa opzione appare come opzione di registrazione dell'app. Questo metodo è adatto

quando è necessario un maggiore controllo sulla configurazione dell'applicazione o quando le autorizzazioni di consenso dell'amministratore non sono disponibili.

Entrambi i metodi offrono le stesse funzionalità. È possibile utilizzare uno o entrambi i metodi all'interno dello stesso AWS account.

Limiti noti

- Consenso dell'amministratore: un AWS account per tenant di Azure: ogni tenant di Azure può avere la propria app AWS DevOps Agent Entra associata a un solo AWS account alla volta. Per associare lo stesso tenant a un AWS account diverso, devi prima annullare la registrazione esistente.
- Registrazione dell'app: applicazione unica per registrazione — Ogni registrazione all'app deve utilizzare un'applicazione diversa (ID cliente). Non è possibile registrare più configurazioni con lo stesso ID client.
- Azure DevOps: accesso al codice sorgente: l' DevOps integrazione con Azure fornisce l'accesso alla cronologia di esecuzione della pipeline indipendentemente da dove è ospitato il codice sorgente. Tuttavia, per accedere al codice sorgente effettivo, il repository deve essere connesso separatamente tramite un provider di sorgenti supportato (ad esempio,). the section called "Connessione GitHub" Il codice sorgente ospitato in Bitbucket non è accessibile direttamente tramite l'integrazione con Azure. DevOps

Argomenti

- the section called "Connessione delle risorse di Azure"
- the section called "Connessione ad Azure DevOps"

Connessione delle risorse di Azure

L'integrazione con Azure Resources consente all' AWS DevOps agente di scoprire e analizzare le risorse nelle sottoscrizioni di Azure durante le indagini sugli incidenti. L'agente usa Azure Resource Graph per l'individuazione delle risorse e può accedere a metriche, log e dati di configurazione nell'ambiente Azure.

Questa integrazione segue un processo in due fasi: registrare Azure a livello di AWS account, quindi associare sottoscrizioni Azure specifiche a singoli Agent Spaces.

Prerequisiti

Prima di connettere Azure Resources, assicurati di avere:

- Accesso alla console dell' AWS DevOps agente
- Un account Azure con accesso alla sottoscrizione di destinazione
- Per il metodo di consenso dell'amministratore: un account con l'autorizzazione a eseguire il consenso dell'amministratore in Microsoft Entra ID
- Per il metodo di registrazione delle app: un'applicazione Entra con autorizzazioni per configurare credenziali di identità federate e Outbound Identity Federation abilitata nell'account AWS

> Nota: è possibile avviare la registrazione anche dall'interno di un Agent Space. Passa a Fonti secondarie, fai clic su Aggiungi e seleziona Azure. Se Azure Cloud non è ancora registrato, la console ti guida prima nella registrazione.

Registrazione delle risorse di Azure tramite il consenso dell'amministratore

Il metodo Admin Consent utilizza un flusso basato sul consenso con l'applicazione gestita dall' AWS DevOps agente.

Fase 1: Avviare la registrazione

1. Accedi alla console di AWS gestione e vai alla console dell' AWS DevOps agente
2. Vai alla pagina Capability Provider
3. Individua la sezione Azure Cloud e fai clic su Registra
4. Seleziona il metodo di registrazione Admin Consent

Fase 2: Completa il consenso dell'amministratore

1. Verifica le autorizzazioni richieste
2. Fai clic per procedere: verrai reindirizzato alla pagina di consenso dell'amministratore di Microsoft Entra
3. Accedi con un account utente principale che dispone dell'autorizzazione a fornire il consenso dell'amministratore
4. Rivedi e concedi il consenso per l'applicazione AWS DevOps Agent

Fase 3: Autorizzazione utente completa

1. Dopo il consenso dell'amministratore, ti viene richiesta l'autorizzazione dell'utente per verificare la tua identità come membro del tenant autorizzato
2. Accedi con un account appartenente allo stesso tenant di Azure
3. Dopo l'autorizzazione, verrai reindirizzato nuovamente alla console dell' AWS DevOps agente con lo stato di successo

Fase 4: Assegnazione dei ruoli

Vedi Assegnazione dei ruoli di Azure di seguito. Cerca AWS DevOps Agent quando selezioni i membri.

Registrazione delle risorse di Azure tramite la registrazione dell'app

Il metodo di registrazione dell'app utilizza la tua applicazione Entra con credenziali di identità federate.

Fase 1: Avviare la registrazione

1. Nella console dell' AWS DevOps agente, vai alla pagina Capability Provider
2. Individua la sezione Azure Cloud e fai clic su Registra
3. Seleziona il metodo di registrazione dell'app

Fase 2: Creare e configurare l'applicazione Entra

Segui le istruzioni visualizzate nella console per:

1. Abilita Outbound Identity Federation nel tuo AWS account (nella console IAM, vai a Impostazioni account → Outbound Identity Federation)
2. Crea un'applicazione Entra nel tuo ID Microsoft Entra o usane una esistente
3. Configura le credenziali di identità federate sull'applicazione

Fase 3: Fornire i dettagli di registrazione

Compila il modulo di registrazione con:

- ID tenant: il tuo identificatore del tenant di Azure

- Nome tenant: un nome visualizzato per il tenant
- ID client: l'ID dell'applicazione (client) dell'applicazione Entra che hai creato
- Pubblico: l'identificatore del pubblico per la credenziale federata

Fase 4: Creare il ruolo IAM

Un ruolo IAM verrà creato automaticamente quando invii la registrazione tramite la console. Consente all' AWS DevOps agente di assumere le credenziali e di richiamare. `sts:GetWebIdentityToken`

Fase 5: Assegnazione dei ruoli

Vedi Assegnazione dei ruoli di Azure di seguito. Cerca l'applicazione Entra che hai creato durante la selezione dei membri.

Fase 6: Completare la registrazione

1. Conferma la configurazione nella console AWS DevOps dell'agente
2. Fate clic su Invia per completare la registrazione

Assegnazione dei ruoli di Azure

Dopo la registrazione, concedi all'applicazione l'accesso in lettura alla tua sottoscrizione di Azure. Questo passaggio è lo stesso per i metodi di consenso dell'amministratore e di registrazione dell'app.

1. Nel portale di Azure, accedi alla sottoscrizione di Target
2. Vai a Access Control (IAM)
3. Fai clic su Aggiungi > Aggiungi assegnazione di ruolo
4. Seleziona il ruolo Reader e fai clic su Avanti
5. Fai clic su Seleziona membri, cerca l'applicazione (AWS DevOps Agent for Admin Consent o la tua applicazione Entra per la registrazione delle app)
6. Selezionate l'applicazione e fate clic su Review + assign
7. (Facoltativo) Per consentire all'agente di accedere ai cluster di Azure Kubernetes Service (AKS), completa la seguente configurazione di accesso AKS.

> Requisito di sicurezza: al responsabile del servizio deve essere assegnato solo il ruolo Reader (e facoltativamente i ruoli di sola lettura AKS elencati di seguito). Il ruolo

Reader funge da limite di sicurezza che limita l'agente alle operazioni di sola lettura e limita l'impatto degli attacchi indiretti di prompt injection. L'assegnazione di ruoli con autorizzazioni di scrittura o azione aumenta in modo significativo il raggio di risposta del prompt injection e può comportare una compromissione delle risorse di Azure. AWS DevOps L'agente esegue solo operazioni di lettura. L'agente non modifica, crea o elimina risorse di Azure.

Configurazione dell'accesso AKS (opzionale)

Fase 1: accesso a livello di Azure Resource Manager (ARM)

Assegna il ruolo utente del cluster di servizio Azure Kubernetes all'applicazione.

Nel portale di Azure, vai a Sottoscrizioni → seleziona abbonamento → Controllo di accesso (IAM) → Aggiungi assegnazione di ruolo → seleziona Ruolo utente del cluster di servizio Azure Kubernetes → assegna all'applicazione (AWS DevOps Agent for Admin Consent o la tua applicazione Entra per la registrazione delle app).

Questo copre tutti i cluster AKS inclusi nell'abbonamento. Per limitarti a cluster specifici, assegnali invece a livello di gruppo di risorse o di singolo cluster.

Fase 2: accesso all'API Kubernetes

Scegli un'opzione in base alla configurazione di autenticazione del cluster:

Opzione A: Azure Role-Based Access Control (RBAC) per Kubernetes (consigliato)

1. Abilita Azure RBAC sul cluster se non è già abilitato: Azure Portal → AKS cluster → Impostazioni → Configurazione di sicurezza → Autenticazione e autorizzazione → seleziona Azure RBAC
2. Assegna un ruolo di sola lettura: Azure Portal → Subscriptions → seleziona abbonamento → Access Control (IAM) → Aggiungi l'assegnazione del ruolo → seleziona Azure Kubernetes Service RBAC Reader → assegna all'applicazione

Questo copre tutti i cluster AKS nell'abbonamento.

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

Usala se il tuo cluster usa già la configurazione di autenticazione predefinita di Azure AD e preferisci non abilitare Azure RBAC. Ciò richiede una configurazione per cluster. `kubectl`

1. Salva il seguente manifesto come: `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. Sostituiscilo <SERVICE_PRINCIPAL_OBJECT_ID> con l'Object ID del responsabile del servizio. Per trovarlo: Azure Portal → Entra ID → Enterprise Applications → cerca il nome dell'applicazione (AWS DevOps Agent for Admin Consent o la tua applicazione Entra per la registrazione delle app).
2. Applica a ciascun cluster:

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

> Nota: i cluster che utilizzano solo account locali (senza Azure AD) non sono supportati. Ti consigliamo di abilitare l'integrazione di Azure AD nel tuo cluster per usare questa funzionalità.

Ruolo personalizzato con privilegi minimi (opzionale)

Per un controllo più rigoroso degli accessi, puoi creare un ruolo di Azure personalizzato riservato solo ai provider di risorse utilizzati da AWS DevOps Agent, anziché il ruolo generale Reader:

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

Associazione di una sottoscrizione a un Agent Space

Dopo aver registrato Azure a livello di account, associa sottoscrizioni specifiche ai tuoi Agent Spaces:

1. Nella console dell' AWS DevOps agente, seleziona il tuo Agent Space
2. Vai alla scheda Funzionalità

3. Nella sezione Fonti secondarie, fai clic su Aggiungi
4. Seleziona Azure
5. Fornisci l'ID di sottoscrizione per la sottoscrizione di Azure che desideri associare
6. Fai clic su Aggiungi per completare l'associazione

Puoi associare più sottoscrizioni allo stesso Agent Space per offrire all'agente visibilità nel tuo ambiente Azure.

Gestione delle connessioni di Azure Resources

- Visualizzazione delle sottoscrizioni connesse: nella scheda Funzionalità, la sezione Fonti secondarie elenca tutte le sottoscrizioni di Azure connesse.
- Rimozione di un abbonamento: per disconnettere un abbonamento da un Agent Space, selezionalo nell'elenco delle fonti secondarie e fai clic su Rimuovi. Ciò non influisce sulla registrazione a livello di account.
- Rimozione della registrazione: per rimuovere completamente la registrazione sul cloud di Azure, vai alla pagina Provider di capacità ed elimina la registrazione. Tutte le associazioni di Agent Space devono prima essere rimosse.

Connessione ad Azure DevOps

DevOps L'integrazione con Azure consente all' AWS DevOps agente di accedere agli archivi e alla cronologia di esecuzione della pipeline nella tua organizzazione Azure. DevOps L'agente può correlare le modifiche e le distribuzioni del codice con gli incidenti operativi per aiutare a identificare le potenziali cause principali.

> Nota: le DevOps pipeline di Azure possono usare il codice sorgente di Azure Repos o Bitbucket. GitHub L' DevOps integrazione con Azure fornisce l'accesso alla cronologia di esecuzione della pipeline indipendentemente dal provider di origine. Tuttavia, per accedere al codice sorgente effettivo durante le indagini, il repository deve essere connesso separatamente tramite un'integrazione supportata come. the section called "Connessione GitHub" Il codice sorgente in Bitbucket non è direttamente accessibile tramite questa integrazione.

Questa integrazione segue un processo in due fasi: registrare Azure DevOps a livello di AWS account, quindi associare progetti specifici a singoli Agent Spaces.

Prerequisiti

Prima di connettere Azure DevOps, assicurati di avere:

- Accesso alla console dell' AWS DevOps agente
- Un' DevOps organizzazione Azure con almeno un progetto contenente un repository e una cronologia della pipeline
- Autorizzazioni per aggiungere utenti alla tua organizzazione Azure DevOps
- Per il metodo di consenso dell'amministratore: un account con l'autorizzazione a eseguire il consenso dell'amministratore in Microsoft Entra ID
- Per il metodo di registrazione delle app: un'applicazione Entra con autorizzazioni per configurare credenziali di identità federate e Outbound Identity Federation abilitata nell'account AWS

> Nota: è possibile avviare la registrazione anche dall'interno di un Agent Space. Vai alla sezione Pipelines, fai clic su Aggiungi e seleziona Azure DevOps. Se Azure non DevOps è ancora registrato, la console ti guida prima nella registrazione.

Registrazione di Azure DevOps tramite il consenso dell'amministratore

Il metodo Admin Consent utilizza un flusso basato sul consenso con l'applicazione gestita dall' AWS DevOps agente.

Fase 1: Avviare la registrazione

1. Accedi alla console di AWS gestione e vai alla console dell' AWS DevOps agente
2. Vai alla pagina Capability Provider
3. Individua la DevOps sezione Azure e fai clic su Registra
4. Inserisci il nome della tua DevOps organizzazione Azure quando richiesto

Passaggio 2: Completa il consenso dell'amministratore

1. Fai clic per procedere: verrai reindirizzato alla pagina di consenso dell'amministratore di Microsoft Entra
2. Accedi con un account utente principale che dispone dell'autorizzazione a fornire il consenso dell'amministratore
3. Rivedi e concedi il consenso per l'applicazione AWS DevOps Agent

Fase 3: Autorizzazione utente completa

1. Dopo il consenso dell'amministratore, ti viene richiesta l'autorizzazione dell'utente per verificare la tua identità come membro del tenant autorizzato
2. Accedi con un account appartenente allo stesso tenant di Azure
3. Dopo l'autorizzazione, verrai reindirizzato nuovamente alla console dell' AWS DevOps agente con lo stato di successo

Passaggio 4: concedere l'accesso in Azure DevOps

Vedi Concessione dell'accesso in Azure DevOps di seguito. Cerca AWS DevOps Agent quando aggiungi utenti.

Registrazione di Azure DevOps tramite la registrazione dell'app

La registrazione dell'app è condivisa tra Azure Resources e Azure. DevOps Se hai già completato la registrazione dell'app per Azure Resources, puoi passare alla sezione Concessione dell'accesso in Azure. DevOps

Passaggio 1: avviare la registrazione dell'app ADO

1. Nella console dell' AWS DevOps agente, vai alla pagina Capability Provider
2. Individua la sezione Azure Cloud e fai clic su Registra
3. Seleziona il metodo di registrazione dell'app

Fase 2: Creare e configurare l'applicazione Entra

Segui le istruzioni visualizzate nella console per:

1. Abilita Outbound Identity Federation nel tuo AWS account (nella console IAM, vai a Impostazioni account → Outbound Identity Federation)
2. Crea un'applicazione Entra nel tuo ID Microsoft Entra o usane una esistente
3. Configura le credenziali di identità federate sull'applicazione

Fase 3: Fornire i dettagli di registrazione

Compila il modulo di registrazione con:

- ID tenant: il tuo identificatore del tenant di Azure
- Nome tenant: un nome visualizzato per il tenant
- ID client: l'ID dell'applicazione (client) dell'applicazione Entra
- Pubblico: l'identificatore del pubblico per la credenziale federata

Fase 4: Creare il ruolo IAM

Un ruolo IAM verrà creato automaticamente quando invii la registrazione tramite la console. Consente all' AWS DevOps agente di assumere le credenziali e di richiamare. `sts:GetWebIdentityToken`

Fase 5: Completare la registrazione

1. Conferma la configurazione nella console AWS DevOps dell'agente
2. Fate clic su Invia per completare la registrazione

Passaggio 6: concedere l'accesso in Azure DevOps

Vedi Concessione dell'accesso in Azure DevOps di seguito. Cerca l'applicazione Entra che hai creato durante la registrazione dell'app quando aggiungi utenti.

Concessione dell'accesso in Azure DevOps

Dopo la registrazione, concedi all'applicazione l'accesso alla tua organizzazione Azure DevOps . Questo passaggio è lo stesso per i metodi di consenso dell'amministratore e di registrazione dell'app.

1. In Azure DevOps, vai a Impostazioni dell'organizzazione > Utenti > Aggiungi utenti
2. Cerca l'applicazione (AWS DevOps Agent for Admin Consent o la tua applicazione Entra per la registrazione delle app)
3. Imposta il livello di accesso su Basic
4. In Aggiungi ai progetti, seleziona i progetti a cui desideri che l'agente acceda
5. In DevOps Gruppi di Azure, seleziona Project Readers
6. Fai clic su Aggiungi per completare

> Requisito di sicurezza: assegna solo il gruppo Project Readers. L'accesso in sola lettura funge da limite di sicurezza che limita l'agente alle operazioni di sola lettura e

limita l'impatto degli attacchi indiretti di prompt injection. L'assegnazione ai gruppi di autorizzazioni di scrittura o azione aumenta in modo significativo il raggio di risposta del prompt injection e può comportare una compromissione delle risorse di Azure. DevOps

Associazione di un progetto a un Agent Space

Dopo aver registrato Azure DevOps a livello di account, associa progetti specifici ai tuoi Agent Spaces:

1. Nella console dell' AWS DevOps agente, seleziona il tuo Agent Space
2. Vai alla scheda Funzionalità
3. Nella sezione Pipeline, fai clic su Aggiungi
4. Seleziona Azure DevOps dall'elenco dei provider disponibili
5. Seleziona il progetto dal menu a discesa dei progetti disponibili
6. Fai clic su Aggiungi per completare l'associazione

Gestione delle connessioni Azure DevOps

- Visualizzazione dei progetti connessi: nella scheda Capacità, la sezione Pipelines elenca tutti i progetti DevOps Azure connessi.
- Rimozione di un progetto: per disconnettere un progetto da un Agent Space, selezionalo nella sezione Pipelines e fai clic su Rimuovi.
- Rimozione della registrazione: per rimuovere completamente la DevOps registrazione di Azure, vai alla pagina Capability Provider ed elimina la registrazione. Tutte le associazioni di Agent Space devono prima essere rimosse.

Connessione alle CI/CD tubazioni

L'integrazione della pipeline CI/CD consente ad AWS DevOps Agent di monitorare le implementazioni e correlare le modifiche al codice con gli incidenti operativi durante le indagini. Collegando i CI/CD provider, l'agente può tenere traccia degli eventi di implementazione e associarli a AWS risorse per aiutare a identificare le potenziali cause alla radice durante la risposta agli incidenti.

AWS DevOps Agent supporta l'integrazione con CI/CD le piattaforme più diffuse attraverso un processo in due fasi:

1. Registrazione a livello di account: registra il tuo CI/CD provider una volta a livello di account AWS
2. Connessione Agent Space: collega progetti o repository specifici a singoli Agent Spaces in base alle esigenze organizzative

Questo approccio consente di condividere le registrazioni dei CI/CD provider su più Agent Spaces mantenendo al contempo un controllo granulare su quali progetti vengono monitorati da ogni spazio.

Fornitori supportati CI/CD

AWS DevOps Agent supporta le seguenti CI/CD piattaforme:

- GitHub— Connect i repository GitHubda.com utilizzando l' GitHub app AWS DevOps Agent.
- GitLab— Connect progetti da GitLab.com, GitLab istanze gestite o implementazioni self-hosted GitLab accessibili pubblicamente.

Argomenti

- the section called "Connessione GitHub"
- the section called "Connessione GitLab"

Connessione GitHub

GitHub l'integrazione consente all' AWS DevOps agente di accedere agli archivi di codice e ricevere eventi di implementazione durante le indagini sugli incidenti. Questa integrazione segue un processo in due fasi: registrazione a livello di account GitHub, seguita dal collegamento di repository specifici a singoli Agent Spaces.

AWS DevOps Agent supporta istanze GitHub .com (SaaS) ed GitHub Enterprise Server (ospitate autonomamente).

Prerequisiti

Prima di connetterti GitHub, assicurati di avere:

- Accesso alla console di amministrazione AWS DevOps dell'agente
- Un account GitHub utente o un'organizzazione con autorizzazioni di amministratore
- Autorizzazione a installare GitHub app nell'account o nell'organizzazione

Per GitHub Enterprise Server, sono inoltre necessari:

- Un'istanza di GitHub Enterprise Server (versione 3.x o successiva) accessibile tramite HTTPS
- L'URL HTTPS dell'istanza di GitHub Enterprise Server (ad esempio,`https://github.example.com`)
- (Facoltativo) Una connessione privata, se l'istanza di GitHub Enterprise Server non è accessibile pubblicamente

Registrazione GitHub (a livello di account)

GitHub è registrato a livello di AWS account e condiviso tra tutti gli Agent Space di quell'account. È sufficiente registrarsi GitHub una sola volta per AWS account.

Passaggio 1: accedi ai fornitori di pipeline

1. Accedi alla console di AWS gestione
2. Vai alla console dell' AWS DevOps agente
3. Vai alla scheda Funzionalità
4. Nella sezione Pipeline, fai clic su Aggiungi
5. Seleziona GitHubdall'elenco dei fornitori disponibili

Se GitHub non è ancora stato registrato, ti verrà prima richiesto di registrarlo.

Passaggio 2: Scegli il tipo di connessione

Nella schermata «Registra GitHub account /organizzazione», seleziona se ti stai connettendo come utente o organizzazione:

- Utente: il tuo GitHub account personale con nome utente e profilo
- Organizzazione: un GitHub account condiviso in cui più persone possono collaborare su più progetti contemporaneamente

Se ti connetti a un'istanza di GitHub Enterprise Server, seleziona la casella di controllo Usa GitHub Enterprise Server e inserisci l'URL HTTPS dell'istanza (ad esempio,`https://github.example.com`).

Se l'istanza di GitHub Enterprise Server non è accessibile pubblicamente, è possibile configurare facoltativamente una connessione privata per consentire ad AWS DevOps Agent di raggiungere l'istanza in modo sicuro. Per ulteriori informazioni, consulta the section called "Connessione a strumenti ospitati privatamente".

> **Note**
>
> Non includete `/api/v3` alcun percorso finale nell'URL: immettete solo l'URL di base.

Passaggio 3: configura l'app GitHub

Fai clic su Invia per iniziare il processo di configurazione dell'app. I passaggi successivi variano a seconda che ci si stia connettendo GitHub a.com o GitHub Enterprise Server.

Per GitHub .com

1. Verrai reindirizzato GitHub a installare l' GitHub app AWS DevOps Agent.
2. Seleziona l'account o l'organizzazione in cui installare l'app.
3. L'app consente all' AWS DevOps agente di ricevere eventi dagli archivi connessi, inclusi gli eventi di distribuzione.

Per GitHub Enterprise Server

GitHub Enterprise Server utilizza un flusso GitHub App Manifest, che configura automaticamente una nuova GitHub app sull'istanza. Ciò comporta due reindirizzamenti all'istanza di GitHub Enterprise Server.

1. Il browser verrà reindirizzato alla pagina «Crea GitHub app» dell'istanza GitHub Enterprise Server.
2. Verrà visualizzato il nome dell'app precompilato. Sentiti libero di cambiare il nome se necessario. Fai clic su Crea GitHub app.
3. Verrai reindirizzato nuovamente ad AWS DevOps Agent, che scambia il codice manifesto con le credenziali dell'app.

Passaggio 4: Seleziona i repository e completa l'installazione

1. Verrà visualizzata la pagina di installazione e autorizzazione dell' GitHub app.
2. Seleziona a quali repository consentire l'accesso all'app:

- Tutti gli archivi: concedi l'accesso a tutti gli archivi attuali e futuri
- Seleziona solo i repository: scegli repository specifici dal tuo account o dalla tua organizzazione

3. Fai clic su Installa e autorizza.
4. Verrai reindirizzato nuovamente alla console dell' AWS DevOps agente, dove GitHub apparirà come registrato a livello di account.

Connessione dei repository a un Agent Space

Dopo la registrazione GitHub a livello di account, puoi connettere repository specifici a singoli Agent Spaces:

1. Nella console dell' AWS DevOps agente, seleziona il tuo Agent Space
2. Vai alla scheda Funzionalità
3. Nella sezione Pipeline, fai clic su Aggiungi
4. Seleziona GitHubdall'elenco dei fornitori disponibili
5. Seleziona il sottoinsieme di repository pertinenti a questo Agent Space
6. Fate clic su Aggiungi per completare la connessione

È possibile collegare diversi set di repository a diversi Agent Spaces in base alle esigenze organizzative.

Comprendere l'app GitHub

L' GitHub app AWS DevOps Agent:

- Richiede l'accesso ai tuoi repository: puoi rivedere le autorizzazioni specifiche durante GitHub l'installazione dell'app
- Riceve eventi di distribuzione e altri eventi del repository
- Consente all' AWS DevOps agente di correlare le modifiche al codice con gli incidenti operativi
- Può essere disinstallato in qualsiasi momento tramite le impostazioni GitHub

Per GitHub Enterprise Server, l' GitHub app viene creata automaticamente sull'istanza durante la registrazione. È possibile gestire l'accesso all'archivio dell'app o disinstallarla tramite Impostazioni > Applicazioni > GitHub App installate. Per eliminare completamente la definizione dell'app, vai a Impostazioni > Impostazioni sviluppatore > GitHub App.

GitHub Aggiornamenti delle autorizzazioni delle app

AWS DevOps L'agente può richiedere aggiornamenti delle autorizzazioni dopo l'installazione GitHub dell'App per supportare nuove funzionalità. Quando ciò accade:

1. Riceverai una notifica GitHub relativa alla richiesta di aggiornamento delle autorizzazioni.
2. Controlla i dettagli dell'aggiornamento per capire quali nuove autorizzazioni vengono richieste.
3. Accetta la richiesta di concessione delle autorizzazioni aggiornate.

Non sono richieste modifiche al servizio o all'applicazione. Una volta accettate le autorizzazioni aggiornate, il token di accesso all'installazione successiva richiesto dall' AWS DevOps Agente GitHub includerà automaticamente le nuove autorizzazioni.

> **Note**
>
> Finché non accetti un aggiornamento delle autorizzazioni, AWS DevOps Agent continua a funzionare con le autorizzazioni concesse in precedenza. Le nuove funzionalità che dipendono dalle autorizzazioni aggiornate non saranno disponibili fino all'approvazione della richiesta.

Gestione delle connessioni GitHub

- Aggiornamento dell'accesso all'archivio: per modificare i repository a cui l' GitHub app può accedere, accedi alle impostazioni dell' GitHub account o dell'organizzazione (o alle impostazioni dell'istanza GitHub Enterprise Server), accedi alle GitHub app installate e modifica la configurazione dell'app AWS DevOps Agent.
- Visualizzazione degli archivi collegati: nella console dell' AWS DevOps agente, seleziona Agent Space e vai alla scheda Funzionalità per visualizzare gli archivi collegati nella sezione Pipeline.
- Rimozione della GitHub connessione: per disconnetterti GitHub da un Agent Space, seleziona la connessione nella sezione Pipeline e fai clic su Rimuovi. Per disinstallare completamente l' GitHub app, disinstallala dalle impostazioni dell' GitHub account o dell'organizzazione. Per GitHub Enterprise Server, poiché l' GitHub app viene creata direttamente sull'istanza durante la registrazione, è possibile opzionalmente ripulire completamente l'app eseguendo entrambe le seguenti operazioni:
  - Disinstalla l'app: vai su Impostazioni > Applicazioni > GitHub App installate, fai clic su Configura sull'app, quindi disinstallala.

- Elimina l'app: vai su Impostazioni > Impostazioni sviluppatore > GitHub App, seleziona l'app, vai alla scheda Avanzate e scegli Elimina GitHub app. Avviso: l'eliminazione dell' GitHub app è permanente e non può essere annullata. Se la elimini, dovrai registrare nuovamente GitHub Enterprise Server dall'inizio nella console di AWS DevOps Agent per creare una nuova app.

Connessione GitLab

GitLab l'integrazione consente all' AWS DevOps agente di monitorare le implementazioni da GitLab Pipelines per fornire informazioni sulle indagini causali durante la risposta agli incidenti. Questa integrazione segue un processo in due fasi: registrazione a livello di account GitLab, seguita dal collegamento di progetti specifici a singoli Agent Spaces.

Registrazione GitLab (a livello di account)

GitLab è registrato a livello di AWS account e condiviso tra tutti gli Agent Space di quell'account. I singoli Agent Spaces possono quindi scegliere quali progetti specifici applicare al proprio Agent Space.

Passaggio 1: accedi ai fornitori di pipeline

1. Accedi alla console di AWS gestione
2. Vai alla console dell' AWS DevOps agente
3. Vai alla pagina Capability Provider (accessibile dalla barra di navigazione laterale)
4. Cerca GitLabnella sezione Provider disponibili sotto Pipeline e fai clic su Registra

Fase 2: Configurare GitLab la connessione

Nella pagina GitLab di registrazione, configura quanto segue:

Tipo di connessione: seleziona se ti stai connettendo come persona o come gruppo:

- Personale (impostazione predefinita): il tuo account GitLab utente individuale con nome utente e profilo
- Gruppo: in GitLab, utilizzi i gruppi per gestire uno o più progetti correlati contemporaneamente

GitLab tipo di istanza: scegli a quale tipo di GitLab istanza ti stai connettendo:

- GitLab.com (impostazione predefinita) — Il GitLab servizio pubblico

- Accessibile pubblicamente, ospitato autonomamente GitLab: seleziona la casella Usa endpoint con hosting GitLab autonomo e fornisci l'URL alla tua istanza GitLab

> **Note**
>
> Attualmente sono supportate solo le GitLab istanze accessibili al pubblico.

Token di accesso: fornisci un token di accesso GitLab personale:

1. In una scheda separata del browser, accedi al tuo GitLab account
2. Vai alle impostazioni utente e seleziona Access Tokens
3. Crea un nuovo token di accesso personale con le seguenti autorizzazioni:
   - `read_repository`— Necessario per accedere ai contenuti del repository
   - `read_virtual_registry`— Necessario per accedere alle informazioni del registro virtuale
   - `read_registry`— Necessario per accedere alle informazioni del registro
   - `api`— Richiesto per l'accesso alle API di lettura e scrittura
   - `self_rotate`- Necessario per la rotazione dei token. Questa funzionalità non è attualmente supportata da AWS DevOps Agent, ma lo sarà in un secondo momento. L'aggiunta di ora evita la necessità di creare un nuovo token in futuro.
4. Imposta la scadenza del token su un massimo di 365 giorni dalla data corrente
5. Copia il token generato
6. Torna alla console dell' AWS DevOps agente
7. Incolla il token nel campo «Token di accesso»

Fase 3: Completa la registrazione

(Facoltativo) Tag: aggiungi AWS tag alla GitLab registrazione per scopi organizzativi.

Fai clic su Avanti per rivedere la configurazione, quindi fai clic su Invia per completare il processo di GitLab registrazione. Il sistema convaliderà il token di accesso e stabilirà la connessione.

Collegamento dei progetti a un Agent Space

Dopo la registrazione GitLab a livello di account, puoi collegare progetti specifici a singoli Agent Spaces:

1. Nella console dell' AWS DevOps agente, seleziona il tuo Agent Space
2. Vai alla scheda Funzionalità
3. Nella sezione Pipeline, fai clic su Aggiungi
4. Seleziona GitLabdall'elenco dei fornitori disponibili
5. Seleziona i GitLab progetti pertinenti al tuo Agent Space
6. Fai clic su Salva

AWS DevOps L'agente monitorerà questi progetti per individuare eventuali implementazioni da parte di GitLab Pipelines al fine di fornire informazioni sulle indagini causali.

Gestione delle connessioni GitLab

- Aggiornamento del token di accesso: se il token di accesso scade o deve essere aggiornato, puoi aggiornarlo nella console dell' AWS DevOps agente modificando la GitLab registrazione a livello di account.
- Visualizzazione dei progetti collegati: nella console dell' AWS DevOps agente, seleziona Agent Space e vai alla scheda Funzionalità per visualizzare i progetti collegati nella sezione Pipeline.
- Rimozione della GitLab connessione: per disconnettere GitLab i progetti da un Agent Space, seleziona la connessione nella sezione Pipeline e fai clic su Rimuovi. Per rimuovere completamente la GitLab registrazione, rimuovila prima da tutti gli Agent Spaces, quindi elimina la registrazione a livello di account.

Connessione dei server MCP

I server Model Context Protocol (MCP) estendono le capacità di indagine di AWS DevOps Agent fornendo l'accesso ai dati provenienti da strumenti di osservabilità esterni, sistemi di monitoraggio personalizzati e fonti di dati operative. Questa guida spiega come connettere un server MCP ad Agent. AWS DevOps

Requisiti

Prima di connettere un server MCP, assicuratevi che il server soddisfi questi requisiti:

- Protocollo di trasporto HTTP Streamable: sono supportati solo i server MCP che implementano il protocollo di trasporto HTTP Streamable.

- Supporto per l'autenticazione: il server MCP deve supportare uno dei seguenti metodi di autenticazione: OAuth 2.0 (Client Credentials o 3LO), autenticazione basata su chiave API/token o Signature Version 4 (SIGv4). AWS

Considerazioni relative alla sicurezza

Quando connetti i server MCP ad Agent, considera questi aspetti di sicurezza: AWS DevOps

- Elenco degli strumenti consentiti: è consigliabile inserire nell'elenco consentito solo gli strumenti specifici necessari ad Agent Space, anziché esporre tutti gli strumenti del server MCP. Vedi Configurazione degli strumenti MCP in un Agent Space per sapere come consentire gli strumenti di elenco per Agent Space.

Si noti che la lunghezza massima dell'utensile MCP è 64.

- Rischi di iniezione rapida: i server MCP personalizzati possono comportare un ulteriore rischio di attacchi di iniezione rapida. Per ulteriori informazioni, vedere Prompt injection protection: AWS DevOps Agent Security.
- Strumenti e accesso di sola lettura: consente solo gli strumenti MCP di sola lettura e garantisce che alle credenziali di autenticazione sia consentito solo l'accesso in sola lettura.

AWS DevOps Sicurezza degli agentiPer ulteriori informazioni sulla pronta iniezione e sul modello di responsabilità condivisa, vedere.

> **Note**
>
> Se il server MCP si trova su una rete privata, vedere the section called "Connessione a strumenti ospitati privatamente"

Registrazione di un server MCP (a livello di account)

I server MCP sono registrati a livello di AWS account e condivisi tra tutti gli Agent Spaces di quell'account. I singoli Agent Spaces possono quindi scegliere gli strumenti specifici di cui hanno bisogno da ciascun server MCP.

Fase 1: Dettagli del server MCP

1. Accedere alla console di AWS gestione
2. Vai alla console dell' AWS DevOps agente
3. Vai alla pagina Capability Provider (accessibile dalla barra di navigazione laterale)
4. Trova MCP Server nella sezione Provider disponibili e fai clic su Registra
5. Nella pagina dei dettagli del server MCP, inserite le seguenti informazioni:
   - Nome: immettete un nome descrittivo per il server MCP
   - URL dell'endpoint: immettete l'URL HTTPS completo dell'endpoint del server MCP
   - Descrizione (opzionale): aggiungi una descrizione per aiutare a identificare lo scopo del server
   - Abilita la registrazione dinamica del client: selezionare questa casella di controllo se si desidera consentire all' AWS DevOps agente di registrarsi automaticamente sul server di autorizzazione del server MCP
   - Connetti all'endpoint utilizzando una connessione privata: seleziona questa casella di controllo se desideri che AWS DevOps l'agente effettui richieste al tuo server MCP in privato. È possibile selezionare una connessione privata esistente o crearne una nuova. Se si utilizza l' OAuth autenticazione, la connessione privata si applica sia all'endpoint del server MCP che all'endpoint di scambio di token. Assicurati che la connessione privata sia configurata con un indirizzo host in grado di indirizzare il traffico verso entrambi gli endpoint. Per ulteriori informazioni, consulta the section called "Connessione a strumenti ospitati privatamente".
6. Fai clic su Avanti

> **Note**
>
> L'URL dell'endpoint del server MCP verrà visualizzato nei AWS CloudTrail log del tuo account.

Fase 2: Flusso di autorizzazione

Seleziona il metodo di autenticazione per il tuo server MCP:

OAuth Credenziali client: se il server MCP utilizza il flusso OAuth Client Credentials:

1. Seleziona Credenziali client OAuth

2. Fai clic su Avanti

OAuth 3LO (Three-Legged OAuth): se il server MCP utilizza OAuth 3LO per l'autenticazione:

1. Seleziona 3LO OAuth
2. Fai clic su Avanti

Chiave API: se il server MCP utilizza l'autenticazione tramite chiave API:

1. Seleziona la chiave API
2. Fai clic su Avanti

AWS SigV4 — Se il server MCP utilizza l'autenticazione AWS Signature Version 4:

1. AWS Seleziona SigV4
2. Fai clic su Avanti

Fase 3: Configurazione dell'autorizzazione

Configura parametri di autorizzazione aggiuntivi in base al metodo di autenticazione selezionato:

Per le credenziali OAuth del client:

1. ID cliente: immettere l'ID client del OAuth client
2. Segreto del cliente: immettere il segreto del OAuth client
3. URL di scambio: immettere l'URL dell'endpoint di scambio di OAuth token
4. Parametri di Exchange: OAuth immettete i parametri di scambio di token per l'autenticazione con il servizio
5. Aggiungi ambito: aggiunge OAuth ambiti per l'autenticazione
6. Fai clic su Avanti

Per OAuth 3LO:

1. ID client: immettere l'ID client del OAuth client
2. Segreto del cliente: inserisci il segreto del OAuth cliente, se richiesto dal OAuth cliente

3. URL di Exchange: inserisci l'URL dell'endpoint di scambio di OAuth token
4. URL di autorizzazione: inserisci l'URL dell'endpoint di OAuth autorizzazione
5. Code Challenge Support: seleziona questa casella di controllo se il tuo OAuth client supporta Code Challenge
6. Aggiungi ambito: aggiungi OAuth ambiti per l'autenticazione
7. Fai clic su Avanti

Per API Key:

1. Inserisci il nome di una chiave API
2. Inserisci il nome dell'intestazione che conterrà la chiave API nella richiesta
3. Inserisci il valore della tua chiave API
4. Fai clic su Avanti

Per AWS SigV4:

AWS L'autenticazione SIGv4 consente all' AWS DevOps agente di connettersi ai server MCP che utilizzano la versione 4 di Signature per AWS la firma delle richieste. Ciò è utile per i server MCP ospitati su Amazon API Gateway o altri AWS servizi che supportano l'autenticazione SigV4.

> Nota: le connessioni private non sono supportate per i server MCP che utilizzano l'autenticazione SigV4. L'endpoint del server MCP deve essere accessibile al pubblico. Per i server MCP su reti private che utilizzano altri metodi di autenticazione, vedere. the section called "Connessione a strumenti ospitati privatamente"

1. Configura il ruolo IAM: scegli una delle seguenti opzioni:
   - Usa un ruolo esistente: seleziona un ruolo IAM esistente dal menu a discesa. Il ruolo deve avere una politica di fiducia che consenta al responsabile del servizio AWS DevOps Agent di assumerlo (vedi Creazione di un ruolo IAM per l'autenticazione SigV4).
   - Crea un nuovo ruolo manualmente: segui le step-by-step istruzioni visualizzate nella console per creare un nuovo ruolo IAM con la policy di fiducia corretta.
2. AWS Regione: inserisci la AWS regione per la firma SIGv4 (ad esempio,`us-east-1`). Per utilizzare la firma multiregionale SigV4A, inserisci. *
3. Nome servizio: immettere il nome del AWS servizio per la firma SIGv4 (ad esempio, `execute-api` per API Gateway).

4. Intestazioni personalizzate (facoltative): aggiungi fino a 10 coppie di intestazioni chiave-valore personalizzate da includere in ogni richiesta firmata.
5. Fai clic su Avanti

Fase 4: Rivedi e invia

1. Rivedi tutti i dettagli di configurazione del server MCP
2. Fate clic su Invia per completare la registrazione
3. AWS DevOps L'agente convaliderà la connessione al server MCP
4. Una volta completata la convalida, il server MCP verrà registrato a livello di account

Configurazione degli strumenti MCP in un Agent Space

Dopo aver registrato un server MCP a livello di account, è possibile configurare quali strumenti di quel server sono disponibili per specifici Agent Spaces:

1. Nella console dell' AWS DevOps agente, seleziona il tuo Agent Space
2. Vai alla scheda Funzionalità
3. Nella sezione Server MCP, fai clic su Aggiungi
4. Seleziona il server MCP registrato che desideri connettere a questo Agent Space
5. Configura quali strumenti di questo server MCP devono essere disponibili per Agent Space:
   - Consenti tutti gli strumenti: rende disponibili tutti gli strumenti del server MCP
   - Seleziona strumenti specifici: consente di scegliere quali strumenti consentire nell'elenco
6. Fate clic su Aggiungi per connettere il server MCP a Agent Space

AWS DevOps L'agente sarà ora in grado di utilizzare gli strumenti consentiti dal server MCP durante le indagini in questo Agent Space.

Gestione delle connessioni al server MCP

Aggiornamento delle credenziali di autenticazione: se è necessario aggiornare le credenziali di autenticazione, sarà necessario registrare nuovamente il server MCP. Passate alla pagina Capability Provider nella console dell' AWS DevOps agente, individuate il server MCP, rimuovete eventuali

associazioni attive e fate clic su Annulla registrazione. Successivamente, registrate il server MCP con le nuove credenziali di autenticazione e ricreate le associazioni necessarie con Agent Space.

Visualizzazione dei server MCP connessi: per vedere tutti i server MCP collegati a Agent Space, seleziona Agent Space, vai alla scheda Funzionalità e controlla la sezione Server MCP. Puoi anche aggiornare gli strumenti selezionati qui.

Rimozione delle connessioni al server MCP: per disconnettere un server MCP da un Agent Space, selezionate il server nella sezione Server MCP e fate clic su Rimuovi. Per eliminare completamente una registrazione del server MCP, rimuovila prima da tutti gli Agent Spaces, quindi elimina la registrazione a livello di account.

Creazione di un ruolo IAM per l'autenticazione SigV4

Quando utilizza l'autenticazione AWS SigV4, l' AWS DevOps agente assume un ruolo IAM nell'account per firmare le richieste al server MCP. Questo ruolo deve avere una politica di fiducia che consenta all' AWS DevOps agente service principal (`aidevops.amazonaws.com`) di assumerlo, con una confusa protezione sostitutiva.

Policy di attendibilità

Crea un ruolo IAM con la seguente politica di fiducia. Sostituiscilo `REGION` con la tua AWS regione (ad esempio`us-east-1`) e `ACCOUNT_ID` con l'ID AWS del tuo account.

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

```
    }
  ]
}
```

La politica di fiducia include le seguenti condizioni per prevenire il confuso problema del vice:

- `aws:SourceAccount`— Limita l'assunzione del ruolo alle richieste provenienti dal tuo account. AWS
- `aws:SourceArn`— Limita l'assunzione del ruolo alle richieste provenienti dalle risorse del servizio AWS DevOps Agent presenti nell'account.

Policy delle autorizzazioni

Allega al ruolo una politica di autorizzazioni che conceda le autorizzazioni minime necessarie per richiamare il server MCP. Ad esempio, se il tuo server MCP è ospitato su Amazon API Gateway, il ruolo deve avere l'`execute-api:Invoke`autorizzazione sulla risorsa API Gateway.

Firma multiregionale (SigV4a)

Se il server MCP è distribuito in più AWS regioni, è possibile utilizzare Sigv4a (Signature Version 4a) per la firma multiregionale. Per abilitare ciò, inserisci * come Regione durante la configurazione dell' AWS autorizzazione SigV4. SigV4a utilizza la firma asimmetrica, che consente a una singola richiesta firmata di essere valida in più regioni.

Argomenti correlati

- AWS DevOps Sicurezza in Agent
- Configurazione di uno spazio per agenti
- Protezione da iniezione rapida

Connessione di più AWS account

AWS Gli account secondari consentono all' AWS DevOps agente di esaminare le risorse di più AWS account dell'organizzazione. Quando le applicazioni si estendono su più account, l'aggiunta di account secondari garantisce all'agente la visibilità di tutte le risorse pertinenti durante le indagini sugli incidenti. Un maggiore accesso agli account e alle risorse che compongono un'applicazione garantisce una maggiore precisione delle indagini.

Prerequisiti

Prima di aggiungere un AWS account secondario, assicurati di avere:

- Accesso alla console AWS DevOps dell'agente nell'account principale
- Accesso amministrativo all' AWS account secondario
- Autorizzazioni IAM per creare ruoli nell'account secondario

Aggiungere un account secondario AWS

Oltre ai passaggi seguenti, puoi utilizzare il the section called "AWS DevOps Guida all'onboarding CLI per agenti" per aggiungere account secondari a livello di codice.

Passaggio 1: avviare la configurazione dell'account secondario

1. Accedi alla console di AWS gestione e vai alla console dell' AWS DevOps agente
2. Seleziona il tuo Agent Space
3. Vai alla scheda Funzionalità
4. Nella sezione Cloud, individua la sottosezione Fonti secondarie
5. Fai clic su Aggiungi

Fase 2: Specificare il nome del ruolo

1. Nel campo Dai un nome al tuo ruolo, inserisci un nome per il ruolo che creerai nell'account secondario
2. Nota questo nome: lo utilizzerai nuovamente quando creerai il ruolo nell'account secondario
3. Copia la policy di attendibilità fornita nella console e salvala in uno spazio virtuale

Fase 3: Creare il ruolo nell'account secondario

1. Apri una nuova scheda del browser e accedi alla console IAM nell' AWS account secondario
2. Vai a IAM > Ruoli > Crea ruolo
3. Seleziona Politica di fiducia personalizzata
4. Incolla la politica di fiducia che hai copiato dal passaggio 2

5. Fai clic su Avanti

Fase 4: Allega la policy AWS gestita

1. Nella sezione Politiche di autorizzazione, cerca AIDevOpsAgentAccessPolicy
2. Seleziona la casella di controllo accanto alla politica gestita AIDevOpsAgentAccessPolicy
3. Fai clic su Avanti

Fase 5: Assegna un nome e crea il ruolo

1. Nel campo Nome ruolo, inserisci lo stesso nome di ruolo fornito nel passaggio 2
2. (Facoltativo) Aggiungi una descrizione per identificare lo scopo del ruolo
3. Rivedi la politica di attendibilità e le autorizzazioni allegate
4. Fai clic su Crea ruolo

Passaggio 6: allega la politica in linea

1. Nella console IAM, individua e seleziona il ruolo che hai appena creato
2. Vai alla scheda Autorizzazioni
3. Fai clic su Aggiungi autorizzazioni > Crea politica in linea
4. Passa alla scheda JSON
5. Incolla la politica che hai salvato nel passaggio 2
6. Incolla la policy nell'editor JSON nella console IAM
7. Fai clic su Avanti
8. Fornisci un nome per la politica in linea (ad esempio, "DevOpsAgentInlinePolicy«)
9. Fai clic su Crea politica

Fase 7: Completare la configurazione

1. Torna alla console AWS DevOps dell'agente nell'account principale
2. Fai clic su Avanti per completare la configurazione dell'account secondario
3. Verifica che lo stato della connessione sia impostato su Attivo

Comprensione delle politiche richieste

AWS DevOps L'agente richiede tre componenti di policy per accedere alle risorse in un account secondario:

- Politica di fiducia: consente all' AWS DevOps agente dell'account principale di assumere il ruolo nell'account secondario. Ciò stabilisce la relazione di fiducia tra gli account.
- AIDevOpsAgentAccessPolicy (policy AWS gestita): fornisce le autorizzazioni di base di sola lettura necessarie all' AWS DevOps agente per esaminare le risorse nell'account secondario. Questa politica viene gestita AWS e aggiornata man mano che vengono aggiunte nuove funzionalità.
- Policy in linea: fornisce autorizzazioni aggiuntive specifiche per la configurazione di Agent Space. Questa policy viene generata in base alle impostazioni di Agent Space e può includere autorizzazioni per integrazioni o funzionalità specifiche.

Nell'account primario, il ruolo AWS DevOps Agent IAM deve essere in grado di assumere il ruolo creato nell'account secondario.

Gestione degli account secondari

- Visualizzazione degli account connessi: nella scheda Funzionalità, la sottosezione Fonti secondarie elenca tutti gli account secondari collegati con il relativo stato di connessione.
- Aggiornamento del ruolo IAM: se devi modificare le autorizzazioni, aggiorna la policy in linea allegata al ruolo nell'account secondario. Le modifiche diventano effettive immediatamente.
- Rimuovere un account secondario: per disconnettere un account secondario, selezionalo nell'elenco Fonti secondarie e fai clic su Rimuovi. Ciò non elimina il ruolo IAM nell'account secondario.

Connessione delle fonti di telemetria

AWS DevOps Agent offre tre modi per connettersi alle fonti di telemetria.

Integrazione bidirezionale integrata

Attualmente, AWS DevOps Agent supporta gli utenti Dynatrace con un'integrazione bidirezionale integrata che consente quanto segue:

- Mappatura delle risorse topologiche: AWS DevOps Agent amplierà la topologia DevOps Agent Space con entità e relazioni disponibili tramite un server MCP Dynatrace ospitato da un agente. AWS DevOps
- Attivazione automatica delle indagini: i flussi di lavoro di Dynatrace possono essere configurati per attivare le indagini sulla risoluzione degli incidenti di Dynatrace Problems.
- Introspezione telemetrica: l' AWS DevOps agente può analizzare la telemetria di Dynatrace mentre indaga su un problema tramite il server MCP Dynatrace ospitato dall'agente. AWS DevOps
- Aggiornamenti sullo stato: l' AWS DevOps agente pubblicherà i risultati delle indagini chiave, le analisi delle cause principali e i piani di mitigazione generati sull'interfaccia utente di Dynatrace.

Per ulteriori informazioni sulle integrazioni bidirezionali, consulta

- the section called "Connessione di Dynatrace"

Integrazione unidirezionale integrata

Attualmente, AWS DevOps Agent supporta gli AWS CloudWatch utenti di Amazon S3, Datadog, Grafana, New Relic e Splunk con integrazioni unidirezionali integrate.

Best practice di sicurezza: quando si configurano le credenziali per le integrazioni unidirezionali integrate, consigliamo di utilizzare chiavi e token API per l'accesso in sola lettura. AWS DevOps L'agente utilizza queste credenziali solo per l'introspezione telemetrica e non richiede l'accesso in scrittura al provider di telemetria.

L'integrazione AWS CloudWatch unidirezionale integrata non richiede alcuna configurazione aggiuntiva e consente quanto segue:

- Mappatura delle risorse topologiche: AWS DevOps Agent amplierà la topologia di DevOps Agent Space con entità e relazioni a sua disposizione tramite gli account cloud primari e secondari configurati. AWS
- Introspezione telemetrica: l' AWS DevOps agente può eseguire l'introspezione della AWS CloudWatch telemetria mentre indaga su un problema tramite i ruoli IAM forniti durante la configurazione dell'account cloud primario e secondario. AWS

L'integrazione unidirezionale integrata di Amazon S3 consente quanto segue:

- Introspezione telemetrica: l' AWS DevOps agente può leggere gli oggetti dai bucket Amazon S3 mentre analizza un problema. Ciò è utile per accedere a log, file di configurazione e altri elementi archiviati in S3.

Per utilizzare l'integrazione con Amazon S3, aggiungi le `s3:ListBucket` autorizzazioni `s3:GetObject` and al ruolo IAM dell' DevOps agente. Seguendo il principio del privilegio minimo, assegna queste autorizzazioni solo ai bucket S3 specifici a cui l'agente deve accedere. Per ulteriori informazioni sulla configurazione delle autorizzazioni IAM, consulta. the section called "DevOps Autorizzazioni Agent IAM"

Le integrazioni unidirezionali integrate Datadog, Grafana, New Relic e Splunk richiedono la configurazione e l'abilitazione di quanto segue:

- Attivazione automatica delle indagini: gli eventi Datadog, Grafana, New Relic e Splunk possono essere configurati per attivare le indagini sulla risoluzione degli incidenti degli agenti tramite i webhook degli AWS DevOps agenti. AWS DevOps
- Introspezione telemetrica: l' AWS DevOps agente può analizzare la telemetria di Datadog, Grafana, New Relic e Splunk mentre indaga su un problema tramite il server MCP remoto di ciascun provider.

Per ulteriori informazioni sulle integrazioni unidirezionali, consulta quanto segue:

- the section called "Connessione DataDog"
- the section called "Collegamento a Grafana"
- the section called "Collegamento di New Relic"
- the section called "Connessione a Splunk"

Bring-your-own fonti di telemetria

Per qualsiasi altra fonte di telemetria, incluse le metriche di Prometheus, puoi sfruttare AWS DevOps il supporto di Agent per l'integrazione di webhook e server MCP.

bring-your-ownPer ulteriori informazioni sulle integrazioni, consulta quanto segue

- the section called "Richiamo DevOps dell'agente tramite Webhook"
- the section called "Connessione dei server MCP"

Connessione di Dynatrace

Built-in, integrazione bidirezionale

Attualmente, AWS DevOps Agent supporta gli utenti Dynatrace con un'integrazione bidirezionale integrata che consente quanto segue:

- Mappatura delle risorse topologiche: AWS DevOps Agent amplierà la topologia DevOps Agent Space con entità e relazioni disponibili dall'ambiente Dynatrace.
- Attivazione automatica delle indagini: i flussi di lavoro di Dynatrace possono essere configurati per attivare le indagini sulla risoluzione degli incidenti di Dynatrace Problems.
- Introspezione telemetrica: l' AWS DevOps agente può analizzare la telemetria di Dynatrace mentre indaga su un problema tramite il server MCP Dynatrace. AWS DevOps Agent-hosted
- Aggiornamenti di stato: l' AWS DevOps agente pubblicherà i risultati delle indagini chiave, le analisi delle cause principali e i piani di mitigazione generati sull'interfaccia utente di Dynatrace.

Prerequisiti

L'integrazione dell' AWS DevOps agente con Dynatrace richiede Dynatrace SaaS. L'integrazione dipende dalle funzionalità della piattaforma Dynatrace (flussi di lavoro, AppEngine app tra cui l'app SRE Agents e client OAuth) che sono disponibili solo negli ambienti SaaS Dynatrace.

Dynatrace Managed (locale) non è supportato e Dynatrace non ha intenzione di portare queste funzionalità della piattaforma in Managed. Se utilizzi Dynatrace Managed, dovrai eseguire l'aggiornamento a Dynatrace SaaS prima di collegarlo ad Agent. AWS DevOps Vedi Aggiornamento da Dynatrace Managed a SaaS.

Onboarding

Processo di onboarding

L'onboarding del sistema di osservabilità Dynatrace prevede tre fasi:

1. Connect - Stabilisci la connessione a Dynatrace configurando le credenziali di accesso all'account, con tutti gli ambienti di cui potresti aver bisogno
2. Abilita: attiva Dynatrace in spazi Agent specifici con ambienti Dynatrace specifici
3. Configura il tuo ambiente Dynatrace: usa l'app Dynatrace SRE Agents per completare la connessione in 2 clic

Fase 1: Connect

Stabilisci la connessione al tuo ambiente Dynatrace

Configurazione

1. Vai alla pagina dei Capability Provider (accessibile dalla barra di navigazione laterale)
2. Trova Dynatrace nella sezione Provider disponibili sotto Telemetria e fai clic su Registra
3. Crea un client OAuth in Dynatrace, con le autorizzazioni dettagliate.
   a. Vedi la documentazione di Dynatrace
   b. Quando sei pronto, premi Avanti
   c. Puoi connettere più ambienti Dynatrace e, successivamente, definire quelli specifici per ogni DevOps Agent Space di cui disponi.
4. Inserisci i tuoi dati Dynatrace dalla configurazione del client OAuth:
   - Nome del client
   - ID cliente
   - Segreto del cliente
   - URN dell'account
5. Fai clic su Avanti
6. Rivedi e aggiungi

Fase 2: Abilita

Attiva Dynatrace in uno spazio agente specifico e configura l'ambito appropriato

Configurazione

1. Dalla pagina degli spazi per gli agenti, seleziona uno spazio agente e premi Visualizza dettagli
2. Seleziona la scheda Funzionalità
3. Individua la sezione Telemetria, premi Aggiungi
4. Noterai che Dynatrace ha lo stato «Registrato». Fai clic su Aggiungi per aggiungerlo al tuo spazio agente
5. ID ambiente Dynatrace: fornisci l'ID dell'ambiente Dynatrace che desideri associare a questo spazio agente. DevOps

6. Inserisci uno o più Dynatrace Entity ID: questi aiutano l' DevOps agente a scoprire le tue risorse più importanti, ad esempio servizi o applicazioni. Se non sei sicuro puoi premere rimuovi.
7. Rivedi e premi Salva
8. Copia l'URL del Webhook e il Webhook Secret. Li utilizzerai nell'app Dynatrace SRE Agents per completare la connessione. Per i dettagli, consulta la sezione Guida introduttiva.

Fase 3: Configura il tuo ambiente Dynatrace

Per completare la configurazione di Dynatrace, utilizza l'app Dynatrace SRE Agents per configurare il lato Dynatrace dell'integrazione in 2 clic: non è necessaria alcuna configurazione manuale del flusso di lavoro. Per i dettagli, consulta la sezione Guida introduttiva.

Schemi di eventi supportati

AWS DevOps L'agente supporta due tipi di eventi di Dynatrace tramite webhook. Gli schemi di eventi supportati sono documentati di seguito:

Evento incidente

Gli eventi imprevisti vengono utilizzati per avviare un'indagine. Lo schema degli eventi è:

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

Evento di mitigazione

Gli eventi di mitigazione vengono utilizzati per attivare la generazione di un rapporto di mitigazione per l'indagine sulle fasi successive. Lo schema degli eventi è:

```
{
    "task_id": string;
```

```
    "task_version": number;
    "event.type": "mitigation_request";
}
```

Rimozione

La fonte di telemetria è connessa a due livelli a livello di spazio dell'agente e a livello di account. Per rimuoverla completamente, è necessario prima rimuoverla da tutti gli spazi dell'agente in cui viene utilizzata, dopodiché può essere annullata la registrazione.

Fase 1: Rimuovi dallo spazio dell'agente

1. Dalla pagina degli spazi per gli agenti, seleziona uno spazio agente e premi Visualizza dettagli
2. Seleziona la scheda Funzionalità
3. Scorri verso il basso fino alla sezione Telemetria
4. Seleziona Dynatrace
5. Premi rimuovi

Fase 2: Annullare la registrazione dall'account

1. Vai alla pagina dei Capability Provider (accessibile dalla barra di navigazione laterale)
2. Scorri fino alla sezione Attualmente registrato.
3. Verifica che il numero di spazi per gli agenti sia pari a zero (in caso contrario, ripeti il passaggio 1 precedente negli altri spazi riservati agli agenti)
4. Premi Annulla registrazione accanto a Dynatrace

Connessione DataDog

Built-in, integrazione unidirezionale

Attualmente, AWS DevOps Agent supporta gli utenti Datadog con un'integrazione unidirezionale integrata, che consente quanto segue:

- Attivazione automatica delle indagini: gli eventi Datadog possono essere configurati per attivare le indagini sulla risoluzione degli incidenti degli agenti tramite i AWS DevOps webhook degli agenti. AWS DevOps

- Introspezione telemetrica: l' AWS DevOps agente può analizzare la telemetria di Datadog mentre analizza un problema tramite il server MCP remoto di ciascun provider.

Onboarding

Fase 1: Connect

Stabilisci la connessione all'endpoint MCP remoto Datadog con le credenziali di accesso all'account

Configurazione

1. Vai alla pagina Capability Provider (accessibile dalla navigazione laterale)
2. Trova Datadog nella sezione Provider disponibili in Telemetria e fai clic su Registra
3. Inserisci i dettagli del tuo server Datadog MCP:
   - Nome server: identificatore univoco (ad es. my-datadog-server)
   - URL dell'endpoint: l'endpoint del server Datadog MCP. L'URL dell'endpoint varia a seconda del sito Datadog. Consulta la tabella degli endpoint del sito Datadog di seguito.
   - Descrizione: descrizione opzionale del server
4. Fai clic su Avanti
5. Verifica e invia

Endpoint del sito Datadog

L'URL dell'endpoint MCP varia a seconda del sito Datadog. Per identificare il tuo sito, controlla l'URL nel tuo browser quando accedi a Datadog, oppure consulta Accedi al sito Datadog.

Sito Datadog	Dominio del sito	URL dell'endpoint MCP
US1 (impostazione predefinita)	`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`

Sito Datadog	Dominio del sito	URL dell'endpoint MCP
NOI 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`
AP1	`ap1.datadoghq.com`	`https://mcp.ap1.datadoghq.com/api/unstable/mcp-server/mcp`
AP2	`ap2.datadoghq.com`	`https://mcp.ap2.datadoghq.com/api/unstable/mcp-server/mcp`

Autorizzazione

Autorizzazione OAuth completa tramite:

- Autorizzazione come utente nella pagina Datadog OAuth
- Se non hai effettuato l'accesso, fai clic su Consenti, accedi, quindi autorizza

Una volta configurato, Datadog diventa disponibile in tutti gli spazi dell'agente.

Fase 2: Abilita

DataDog Effettua l'attivazione in uno spazio specifico dell'agente e configura l'ambito appropriato

Configurazione

1. Dalla pagina degli spazi per agenti, seleziona uno spazio agente e premi Visualizza dettagli (se non hai ancora creato uno spazio agente, consultathe section called "Creazione di uno spazio per agenti")

2. Seleziona la scheda Funzionalità
3. Scorri verso il basso fino alla sezione Telemetria
4. Premi Aggiungi
5. Seleziona Datadog
6. Next
7. Rivedi e premi Salva
8. Copia l'URL del Webhook e la chiave API

Fase 3: Configurare i webhook

Utilizzando l'URL e la chiave API del Webhook, puoi configurare Datadog per inviare eventi per avviare un'indagine, ad esempio da un allarme.

I webhook Datadog utilizzano l'autenticazione con token Bearer. Per il formato completo di richiesta del webhook, lo schema di payload e il codice di esempio, consulta. the section called "Richiamo DevOps dell'agente tramite Webhook" Utilizza gli esempi della versione 2 (autenticazione con token Bearer), impostando l'`Authorization: Bearer <Token>`intestazione con la chiave API del passaggio 2.

Invia webhook con Datadog https://docs.datadoghq.com/integrations/webhooks/(nota: seleziona nessuna autorizzazione e utilizza invece l'opzione di intestazione personalizzata).

Per saperne di più: Datadog Remote MCP Server

Rimozione

La fonte di telemetria è connessa a due livelli a livello di spazio dell'agente e a livello di account. Per rimuoverla completamente, è necessario prima rimuoverla da tutti gli spazi dell'agente in cui viene utilizzata, dopodiché può essere annullata la registrazione.

Fase 1: Rimuovi dallo spazio dell'agente

1. Dalla pagina degli spazi per gli agenti, seleziona uno spazio agente e premi Visualizza dettagli
2. Seleziona la scheda Funzionalità
3. Scorri verso il basso fino alla sezione Telemetria
4. Seleziona Datadog
5. Premi rimuovi

Fase 2: Annullare la registrazione dall'account

1. Vai alla pagina dei Capability Provider (accessibile dalla barra di navigazione laterale)
2. Scorri fino alla sezione Attualmente registrato.
3. Verifica che il numero di spazi per gli agenti sia pari a zero (in caso contrario, ripeti il passaggio 1 precedente negli altri spazi riservati agli agenti)
4. Premi Annulla registrazione accanto a Datadog

Collegamento a Grafana

L'integrazione con Grafana consente all' AWS DevOps agente di interrogare metriche, dashboard e dati di avviso dall'istanza Grafana durante le indagini sugli incidenti. Questa integrazione segue un processo in due fasi: registrazione a livello di account di Grafana, seguita dal collegamento a singoli Agent Spaces.

Per migliorare la sicurezza, l'integrazione Grafana abilita solo strumenti di sola lettura. Gli strumenti di scrittura sono disabilitati e non possono essere abilitati. Ciò significa che l'agente può interrogare e leggere i dati dall'istanza Grafana ma non può creare, modificare o eliminare alcuna risorsa Grafana come dashboard, avvisi o annotazioni. Per ulteriori informazioni, consulta Security in Agent. AWS DevOps

Requisiti Grafana

Prima di collegare Grafana, assicurati di avere:

- Grafana versione 9.0 o successiva. Alcune funzionalità, in particolare le operazioni relative alle origini dati, potrebbero non funzionare correttamente con le versioni precedenti a causa della mancanza degli endpoint API.
- Un'istanza Grafana accessibile tramite HTTPS. Sono supportati sia gli endpoint di rete pubblici che quelli privati. Con la connettività di rete privata, l'istanza Grafana può essere ospitata all'interno di un VPC senza accesso pubblico a Internet. Per informazioni dettagliate, vedi the section called "Connessione a strumenti ospitati privatamente".
- Un account di servizio Grafana con un token di accesso con autorizzazioni di lettura appropriate

Registrazione di Grafana (a livello di account)

Grafana è registrata a livello di AWS account e condivisa tra tutti gli Agent Spaces di quell'account.

Fase 1: Configurare Grafana

1. Accedi alla console di AWS gestione
2. Passa alla console AWS DevOps dell'agente
3. Vai alla pagina Capability Provider (accessibile dalla barra di navigazione laterale)
4. Trova Grafana nella sezione Provider disponibili in Telemetria e fai clic su Registra
5. Nella pagina Configura Grafana, inserisci le seguenti informazioni:
   - Nome servizio (obbligatorio): inserisci un nome descrittivo per il tuo server Grafana utilizzando solo caratteri alfanumerici, trattini e caratteri di sottolineatura. Ad esempio, `my-grafana-server`.
   - URL Grafana (obbligatorio): inserisci l'URL HTTPS completo dell'istanza Grafana. Ad esempio, `https://myinstance.grafana.net`.
   - Token di accesso all'account di servizio (obbligatorio): immettere un token di accesso all'account di servizio Grafana. I token in genere iniziano con. `glsa_` Per creare un token di account di servizio, accedi alla tua istanza Grafana, vai su Amministrazione > Account di servizio, crea un account di servizio con il ruolo Viewer e genera un token.
   - Descrizione (opzionale): aggiungi una descrizione per identificare lo scopo del server. Ad esempio, `Production Grafana server for monitoring`.
6. (Facoltativo) Aggiungi AWS tag alla registrazione per scopi organizzativi.
7. Fai clic su Avanti

Fase 2: Rivedi e invia la registrazione Grafana

1. Rivedi tutti i dettagli della configurazione Grafana
2. Fai clic su Invia per completare la registrazione
3. Una volta completata la registrazione, Grafana appare nella sezione Attualmente registrato della pagina Provider di capacità

Aggiungere Grafana a uno spazio per agenti

Dopo aver registrato Grafana a livello di account, puoi collegarlo a singoli Agent Spaces:

1. Nella console dell' AWS DevOps agente, seleziona il tuo Agent Space
2. Vai alla scheda Funzionalità

3. Nella sezione Telemetria, fai clic su Aggiungi
4. Seleziona Grafana dall'elenco dei fornitori disponibili
5. Fai clic su Salva

Configurazione dei webhook di avvisi Grafana

È possibile configurare Grafana per attivare automaticamente le indagini degli AWS DevOps agenti quando vengono attivati gli avvisi inviando webhook tramite i punti di contatto Grafana. Per i dettagli sui metodi di autenticazione dei webhook e sulla gestione delle credenziali, consulta. the section called "Richiamo DevOps dell'agente tramite Webhook"

Fase 1: Creare un modello di notifica personalizzato

Nella tua istanza Grafana, vai su Avvisi > Punti di contatto > Modelli di notifica e crea un nuovo modello con il seguente contenuto:

```
{{ 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 }}
```

Questo modello formatta gli avvisi Grafana nella struttura di payload del webhook prevista dall'agente. AWS DevOps Mappa le etichette degli avvisi, le annotazioni e lo stato nei campi appropriati e include tutte le etichette di avviso come metadati.

> Nota: questo modello elabora solo il primo avviso di un gruppo. Grafana raggruppa più avvisi di attivazione in un'unica notifica per impostazione predefinita. Per garantire che ogni avviso venga inviato singolarmente, configura le politiche di notifica in modo che vengano raggruppate per. `alertname` Inoltre, questo modello non sfugge ai caratteri JSON speciali nei valori delle etichette o nelle annotazioni. Assicurati che le etichette degli avvisi e l'`summary`annotazione non contengano caratteri come virgolette o nuove righe, che produrrebbero un codice JSON non valido.

Fase 2: Creare un punto di contatto webhook

1. In Grafana, vai su Avvisi > Punti di contatto e fai clic su Aggiungi punto di contatto
2. Seleziona Webhook come tipo di integrazione
3. Imposta l'URL sull'endpoint AWS DevOps webhook dell'agente
4. In Impostazioni opzionali del Webhook, configura le intestazioni di autenticazione in base al tipo di webhook. Per i dettagli, consulta Metodi di autenticazione Webhook.
5. Imposta il campo Custom Payload per utilizzare il tuo modello personalizzato: `{{ template "devops-agent-payload" . }}`
6. Fai clic su Salva punto di contatto

Passaggio 3: assegna il punto di contatto a una politica di notifica

1. Passa a Avvisi > Politiche di notifica
2. Modifica una politica esistente o creane una nuova
3. Imposta il punto di contatto sul punto di contatto webhook che hai creato
4. Fai clic su Salva politica

Quando viene attivato un avviso corrispondente, Grafana invierà il payload formattato AWS DevOps all'agente, che avvierà automaticamente un'indagine.

Limitazioni

- ClickHouse strumenti per l'origine dei dati: gli strumenti per l'origine ClickHouse dei dati non sono attualmente supportati.
- Prevenzione proattiva degli incidenti: attualmente the section called "Prevenzione proattiva degli incidenti" non utilizza gli strumenti Grafana. Il supporto è previsto per le future release.

Considerazioni su Amazon Managed Grafana

Se utilizzi Amazon Managed Grafana (AMG), tieni presente le seguenti limitazioni:

- I punti di contatto Webhook non sono supportati: attualmente AMG non supporta i punti di contatto webhook nella sua configurazione di avviso. Non è possibile utilizzare AMG per inviare webhook di avviso direttamente all'agente. AWS DevOps Per i dettagli, consulta Avvisare i punti di contatto in Amazon Managed Grafana.
- Scadenza del token dell'account di servizio: i token dell'account di servizio AMG hanno una scadenza massima di 30 giorni. Dovrai ruotare i token e aggiornare la tua registrazione Grafana in AWS DevOps Agent prima che scadano. Vedi Gestione delle connessioni Grafana per sapere come aggiornare le credenziali. Per dettagli sui limiti dei token AMG, consulta Account di servizio in Amazon Managed Grafana.

Gestione delle connessioni Grafana

- Aggiornamento delle credenziali: se il token del tuo account di servizio scade o deve essere aggiornato, annulla la registrazione di Grafana dalla pagina Capability Provider e registrati nuovamente con il nuovo token.
- Visualizzazione delle istanze connesse: nella console dell' AWS DevOps agente, seleziona Agent Space e vai alla scheda Funzionalità per visualizzare le fonti di telemetria connesse.
- Rimuovere Grafana — Per disconnettere Grafana da un Agent Space, selezionalo nella sezione Telemetria e fai clic su Rimuovi. Per rimuovere completamente la registrazione, rimuovila prima da tutti gli Agent Spaces, quindi annulla la registrazione dalla pagina Capability Providers.

Collegamento di New Relic

Built-in, integrazione unidirezionale

Attualmente, AWS DevOps Agent supporta gli utenti di New Relic con un'integrazione unidirezionale integrata, che consente quanto segue:

- Attivazione automatica delle indagini: gli eventi New Relic possono essere configurati per attivare le indagini sulla risoluzione degli incidenti degli AWS DevOps agenti tramite i webhook degli agenti. AWS DevOps
- Introspezione telemetrica: l' AWS DevOps agente può analizzare la telemetria di New Relic mentre analizza un problema tramite il server MCP remoto di ciascun provider.

Onboarding

Fase 1: Connect

Stabilisci la connessione all'endpoint MCP remoto New Relic con le credenziali di accesso all'account

Utilizza un utente completo della piattaforma (non Basic/Core) in New Relic per abilitare gli strumenti MCP di New Relic.

Configurazione

1. Vai alla pagina dei Capability Provider (accessibile dalla barra di navigazione laterale)
2. Trovate New Relic nella sezione Provider disponibili in Telemetria e fate clic su Registra
3. Segui le istruzioni per ottenere la tua chiave API New Relic
4. Inserisci i dettagli della chiave API del server MCP New Relic:
   - ID account: inserisci l'ID del tuo account New Relic ottenuto sopra
   - Chiave API: inserisci la chiave API ottenuta sopra
   - Seleziona la regione degli Stati Uniti o dell'UE in base a dove si trova il tuo account New Relic.
5. Fai clic su Aggiungi

Fase 2: Abilita

Attiva New Relic in uno spazio agente specifico e configura l'ambito appropriato

Configurazione

1. Dalla pagina Agent Space, selezionate uno spazio agente e premete Visualizza dettagli (se non avete ancora creato uno spazio agente, consultate) the section called "Creazione di uno spazio per agenti"
2. Seleziona la scheda Funzionalità
3. Scorri verso il basso fino alla sezione Telemetria
4. Premi Aggiungi
5. Seleziona New Relic
6. Next
7. Rivedi e premi Salva
8. Copia l'URL del Webhook e la chiave API

Fase 3: Configurare i webhook

Utilizzando l'URL e la chiave API del Webhook, puoi configurare New Relic in modo che invii eventi per avviare un'indagine, ad esempio a seguito di un allarme. Per maggiori dettagli sulla configurazione dei webhook, consulta Change tracking webhook.

I nuovi webhook Relic utilizzano l'autenticazione con token Bearer. Per il formato completo di richiesta dei webhook, lo schema di payload e il codice di esempio, consulta. the section called "Richiamo DevOps dell'agente tramite Webhook" Utilizza gli esempi della versione 2 (autenticazione con token Bearer), impostando l'`Authorization: Bearer <Token>`intestazione con la chiave API del passaggio 2.

Inviate webhook con New Relic. https://newrelic.com/instant-observability/webhook-notifications Puoi selezionare il token Bearer per il tipo di autorizzazione oppure selezionare nessuna autorizzazione e aggiungerlo `Authorization: Bearer <Token>` come intestazione personalizzata.

Per saperne di più: https://docs.newrelic.com/docs/agentic-ai/mcp/overview/

Rimozione

La fonte di telemetria è connessa a due livelli a livello di spazio dell'agente e a livello di account. Per rimuoverla completamente, è necessario prima rimuoverla da tutti gli spazi dell'agente in cui viene utilizzata, dopodiché può essere annullata la registrazione.

Fase 1: Rimuovi dallo spazio dell'agente

1. Dalla pagina degli spazi per gli agenti, seleziona uno spazio agente e premi Visualizza dettagli
2. Seleziona la scheda Funzionalità
3. Scorri verso il basso fino alla sezione Telemetria
4. Seleziona New Relic
5. Premi rimuovi

Fase 2: Annullare la registrazione dall'account

1. Vai alla pagina dei Capability Provider (accessibile dalla barra di navigazione laterale)
2. Scorri fino alla sezione Attualmente registrato.
3. Verifica che il numero di spazi per gli agenti sia pari a zero (in caso contrario, ripeti il passaggio 1 precedente negli altri spazi riservati agli agenti)
4. Premi Annulla registrazione accanto a New Relic

Connessione a Splunk

Built-in, integrazione unidirezionale

Attualmente, AWS DevOps Agent supporta gli utenti Splunk con un'integrazione unidirezionale integrata, che consente quanto segue:

- Attivazione automatica delle indagini: gli eventi Splunk possono essere configurati per attivare le indagini sulla risoluzione degli incidenti degli AWS DevOps agenti tramite i webhook degli agenti. AWS DevOps
- Introspezione telemetrica: l' AWS DevOps agente può analizzare la telemetria di Splunk mentre analizza un problema tramite il server MCP remoto di ciascun provider.

Prerequisiti

Ottenere un token API Splunk

Avrai bisogno di un URL e di un token MCP per connettere Splunk.

Procedura dell'amministratore di Splunk

L'amministratore Splunk deve eseguire le seguenti operazioni:

- abilitare l'accesso all'API REST
- abilita l'autenticazione tramite token sulla distribuzione.
- crea un nuovo ruolo 'mcp_user', il nuovo ruolo non deve avere alcuna funzionalità.
- assegna il ruolo 'mcp_user' a tutti gli utenti della distribuzione autorizzati a utilizzare il server MCP.
- crea il token per gli utenti autorizzati con audience come 'mcp' e imposta la scadenza appropriata, se l'utente non ha l'autorizzazione per creare i token da solo.

Procedure per utenti Splunk

Un utente Splunk deve eseguire i seguenti passaggi:

- Ottieni un token appropriato dall'amministratore Splunk o creane uno lui stesso, se ne ha l'autorizzazione. Il pubblico per il token deve essere 'mcp'.

Onboarding

Fase 1: Connect

Stabilisci la connessione all'endpoint MCP remoto Splunk con le credenziali di accesso all'account

Configurazione

1. Vai alla pagina Capability Provider (accessibile dalla barra di navigazione laterale)
2. Trova Splunk nella sezione Provider disponibili sotto Telemetria e clicca su Registra
3. Inserisci i dettagli del tuo server MCP Splunk:
   - Nome del server: identificatore univoco (ad es. my-splunk-server)
   - URL dell'endpoint - L'endpoint del server Splunk MCP:

```
https://<YOUR_SPLUNK_DEPLOYMENT_NAME>.api.scs.splunk.com/
<YOUR_SPLUNK_DEPLOYMENT_NAME>/mcp/v1/
```

- Descrizione: descrizione opzionale del server
- Nome token: il nome del token portatore per l'autenticazione: `my-splunk-token`

- Valore del token - Il valore del token portatore per l'autenticazione

Fase 2: Abilita

Attiva Splunk in uno spazio specifico dell'agente e configura l'ambito appropriato

Configurazione

1. Dalla pagina Agent Space, seleziona uno spazio agente e premi Visualizza dettagli (se non hai ancora creato uno spazio agente, consulta) the section called "Creazione di uno spazio per agenti"
2. Seleziona la scheda Funzionalità
3. Scorri verso il basso fino alla sezione Telemetria
4. Premi Aggiungi
5. Seleziona Splunk
6. Next
7. Rivedi e premi Salva
8. Copia l'URL del Webhook e la chiave API

Fase 3: Configurare i webhook

Utilizzando l'URL e la chiave API del Webhook, puoi configurare Splunk per inviare eventi per avviare un'indagine, ad esempio a partire da un allarme.

I webhook Splunk utilizzano l'autenticazione con token bearer. Per il formato completo di richiesta dei webhook, lo schema di payload e il codice di esempio, consulta. the section called "Richiamo DevOps dell'agente tramite Webhook" Utilizza gli esempi della versione 2 (autenticazione con token Bearer), impostando l'`Authorization: Bearer <Token>`intestazione con la chiave API del passaggio 2.

Invia webhook con Splunk https://help.splunk.com/en/splunk-enterprise/alert-and-respond/alerting-manual/9.4/configure-alert-actions/use-a-webhook-alert-action(nota: seleziona nessuna autorizzazione e usa invece l'opzione di intestazione personalizzata)

Ulteriori informazioni:

- Documentazione del server MCP di Splunk: https://help.splunk.com/en/splunk-cloud-platform/mcp-server-for-splunk-platform/about-mcp-server-for-splunk-platform
- Requisiti e limitazioni di accesso per l'API REST di Splunk Cloud Platform: https://docs.splunk.com/Documentation/SplunkCloud/latest/RESTTUT/RESTandCloud

- Gestisci i token di autenticazione nella piattaforma Splunk Cloud: 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
- Crea e gestisci ruoli con Splunk Web: https://docs.splunk.com/Documentation/SplunkCloud/latest/Security/Addandeditroles

Rimozione

La fonte di telemetria è connessa a due livelli a livello di spazio dell'agente e a livello di account. Per rimuoverla completamente, è necessario prima rimuoverla da tutti gli spazi dell'agente in cui viene utilizzata, dopodiché può essere annullata la registrazione.

Fase 1: Rimuovi dallo spazio dell'agente

1. Dalla pagina degli spazi per gli agenti, seleziona uno spazio agente e premi Visualizza dettagli
2. Seleziona la scheda Funzionalità
3. Scorri verso il basso fino alla sezione Telemetria
4. Seleziona Splunk
5. Premi rimuovi

Fase 2: Annullare la registrazione dall'account

1. Vai alla pagina dei Capability Provider (accessibile dalla barra di navigazione laterale)
2. Scorri fino alla sezione Attualmente registrato.
3. Verifica che il numero di spazi per gli agenti sia pari a zero (in caso contrario, ripeti il passaggio 1 precedente negli altri spazi riservati agli agenti)
4. Premi Annulla registrazione accanto a Splunk

Connessione alla biglietteria e alla chat

AWS DevOps L'agente è progettato per agire come membro del team partecipando ai canali di comunicazione esistenti del team. Puoi collegare DevOps Agent ai tuoi sistemi di ticketing e allarme, ad esempio, per avviare automaticamente le indagini dai ticket relativi agli incidenti PagerDuty, accelerando la risposta agli incidenti all'interno dei flussi di lavoro esistenti ServiceNow e riducendo il tempo medio di ripristino (MTTR). Puoi anche collegare il tuo DevOps agente ai sistemi di

collaborazione del team come Slack per ricevere riepiloghi delle attività dal tuo agente in un canale di chat. DevOps

Per ulteriori informazioni su come collegare le integrazioni di ticketing e chat, consulta quanto segue:

- the section called "Connessione PagerDuty"
- the section called "Connessione ServiceNow"
- the section called "Connessione a Slack"

Connessione PagerDuty

PagerDuty l'integrazione consente all' AWS DevOps agente di accedere e aggiornare i dati sugli incidenti, gli orari delle chiamate e le informazioni di servizio dal tuo PagerDuty account durante le indagini sugli incidenti e la risposta automatica. Questa integrazione utilizza la OAuth versione 2.0 per l'autenticazione sicura.

> **Important**
>
> AWS DevOps L'agente supporta solo la versione PagerDuty OAuth 2.0 più recente (Scoped OAuth). La versione legacy PagerDuty OAuth con URI di reindirizzamento non è supportata.

PagerDuty requisiti

Prima di connetterti PagerDuty, assicurati di avere:

- Un PagerDuty account con il tuo ID OAuth cliente e il segreto del cliente
- Il sottodominio PagerDuty del tuo account (ad esempio, se il tuo PagerDuty URL è`https://your-company.pagerduty.com`, il sottodominio è) `your-company`

Registrazione PagerDuty

PagerDuty è registrato a livello di AWS account e condiviso tra tutti gli Agent Spaces di quell'account.

Fase 1: Configurare l'accesso in PagerDuty

1. Accedi alla console AWS di gestione

2. Passa alla console AWS DevOps dell'agente
3. Vai alla pagina Capability Provider (accessibile dalla barra di navigazione laterale)
4. Cerca PagerDutynella sezione Provider disponibili nella sezione Comunicazione e fai clic su Registra
5. Segui la configurazione guidata nella PagerDuty pagina Configura l'accesso in:

Controlla la regione e il sottodominio del servizio:

- Ambito dell'account: seleziona la tua PagerDuty regione (Stati Uniti o UE) e inserisci il PagerDuty sottodominio. Ad esempio, se il tuo PagerDuty URL è`https://your-company.pagerduty.com`, inserisci`your-company`.

Crea una nuova app in PagerDuty:

- In una scheda separata del browser, accedi PagerDuty e vai a Integrazioni > Registrazione app
- Crea una nuova app utilizzando OAuth 2.0 Scoped OAuth
- In Autorizzazioni, concedi i seguenti ambiti minimi richiesti:`incidents.read`,, e `incidents.write` `services.read`
- Abilita l'integrazione degli eventi per consentire la comunicazione bidirezionale tra Agente e AWS DevOps PagerDuty

Configura le credenziali: OAuth

- Ambito di autorizzazione: gli ambiti minimi richiesti sono:`incidents.read`,, `incidents.write` `services.read`
- Nome cliente: inserisci un nome descrittivo per il tuo cliente OAuth
- ID cliente: inserisci l'ID OAuth cliente riportato nella registrazione PagerDuty dell'app
- Segreto del cliente: inserisci il segreto OAuth del cliente ottenuto durante la registrazione PagerDuty dell'app

Fase 2: Rivedi e invia PagerDuty la registrazione

1. Rivedi tutti i dettagli PagerDuty di configurazione
2. Fai clic su Invia per completare la registrazione

3. Una volta completata la registrazione, PagerDuty viene visualizzato nella sezione Attualmente registrato della pagina Provider di capacità

Aggiunta PagerDuty a uno spazio per agenti

Dopo la registrazione PagerDuty a livello di account, puoi collegarlo a singoli Agent Spaces:

1. Nella console dell' AWS DevOps agente, seleziona il tuo Agent Space
2. Vai alla scheda Funzionalità
3. Nella sezione Comunicazioni, fai clic su Aggiungi
4. Seleziona PagerDutydall'elenco dei provider disponibili
5. Fai clic su Salva

Gestione delle PagerDuty connessioni

- Aggiornamento delle credenziali: se è necessario aggiornare OAuth le credenziali, annulla la registrazione PagerDuty dalla pagina Capability Providers e registrati nuovamente con le nuove credenziali.
- Visualizzazione delle connessioni: nella console dell' AWS DevOps agente, seleziona Agent Space e vai alla scheda Funzionalità per visualizzare le integrazioni delle comunicazioni connesse.
- Rimozione PagerDuty: per disconnetterti PagerDuty da un Agent Space, selezionalo nella sezione Comunicazioni e fai clic su Rimuovi. Per rimuovere completamente la registrazione, rimuovila prima da tutti gli Agent Spaces, quindi annulla la registrazione dalla pagina Capability Provider.

Supporto Webhook

AWS DevOps L'agente supporta solo i webhook PagerDuty V3. Le versioni precedenti dei webhook non sono supportate.

Per ulteriori informazioni sugli abbonamenti ai webhook PagerDuty V3, consulta Panoramica dei webhook nella documentazione per gli sviluppatori. PagerDuty

Connessione ServiceNow

Questo tutorial illustra come collegare un' ServiceNow istanza ad AWS DevOps Agent per consentirgli di avviare automaticamente le indagini sulla risposta agli incidenti quando viene creato

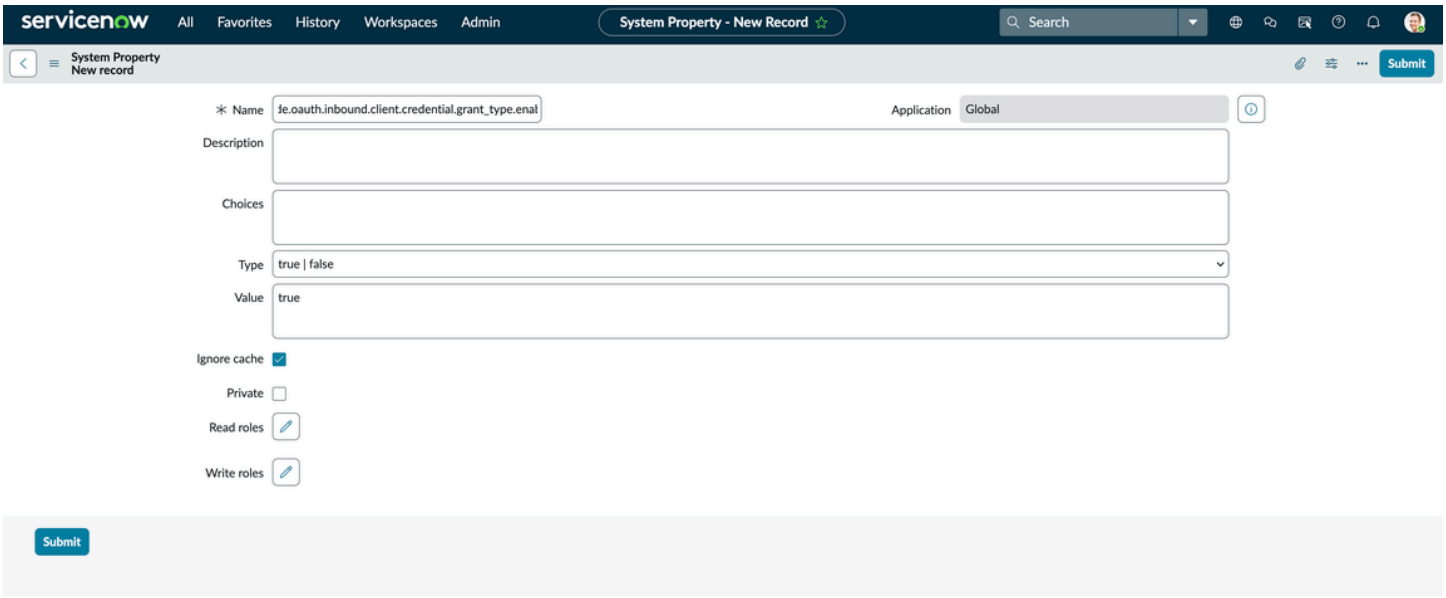
un ticket e di pubblicarne i risultati principali nel ticket di origine. Contiene anche esempi su come configurare l' ServiceNow istanza per inviare solo ticket specifici a un DevOps Agent Space e su come orchestrare il routing dei ticket su più Agent Spaces. DevOps

Configurazione iniziale

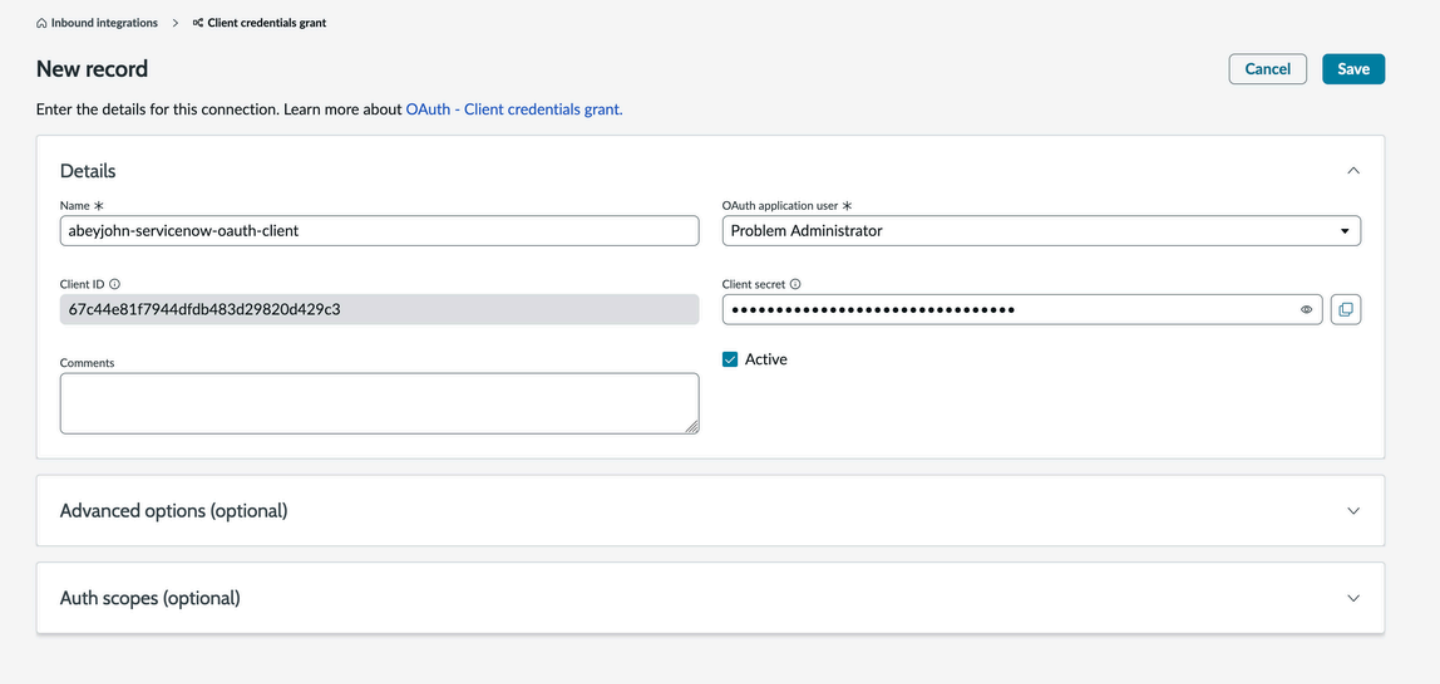
Il primo passaggio consiste nella creazione di ServiceNow un'applicazione client OAuth da utilizzare per accedere ServiceNow all'istanza. AWS DevOps

Crea un client applicativo ServiceNow OAuth

1. Abilita la proprietà del sistema di credenziali del client dell'istanza
   a. Cerca `sys_properties.list` nella casella di ricerca del filtro e poi premi invio (non mostrerà l'opzione ma premere invio funziona)
   b. Scegli Nuovo
   c. Aggiungi il nome as `glide.oauth.inbound.client.credential.grant_type.enabled` e il valore a true con type as true | false

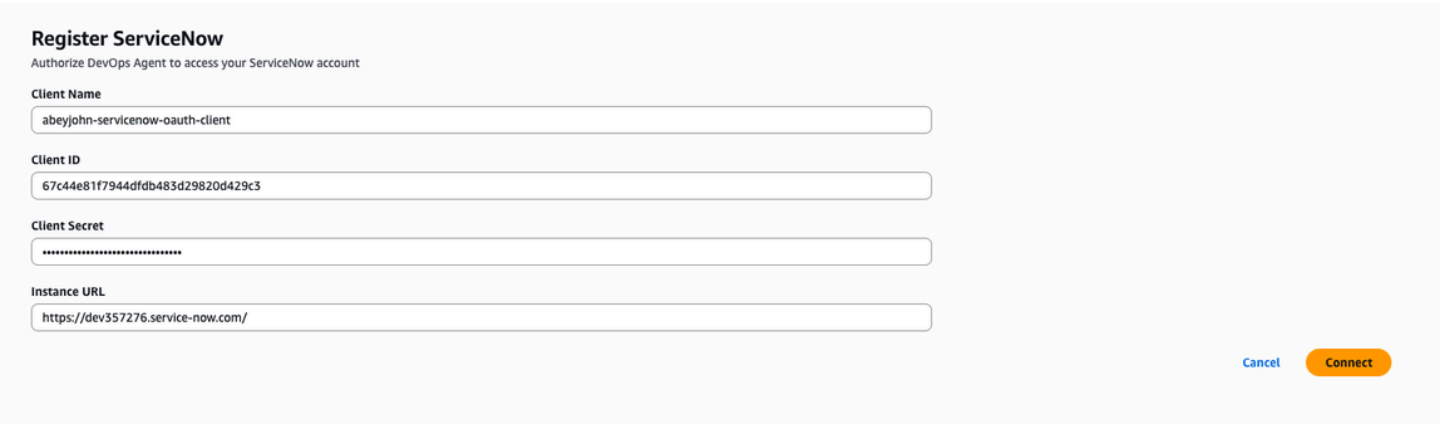


1. Passa a System OAuth > Application Registry dalla casella di ricerca del filtro
2. Scegli «Nuovo» > «Nuova esperienza di integrazione in entrata» > «Nuova integrazione» > «OAuth - Client Credentials Grant»
3. Scegli un nome e imposta l'utente dell'applicazione OAuth su «Problem Administrator», fai clic su «Salva»



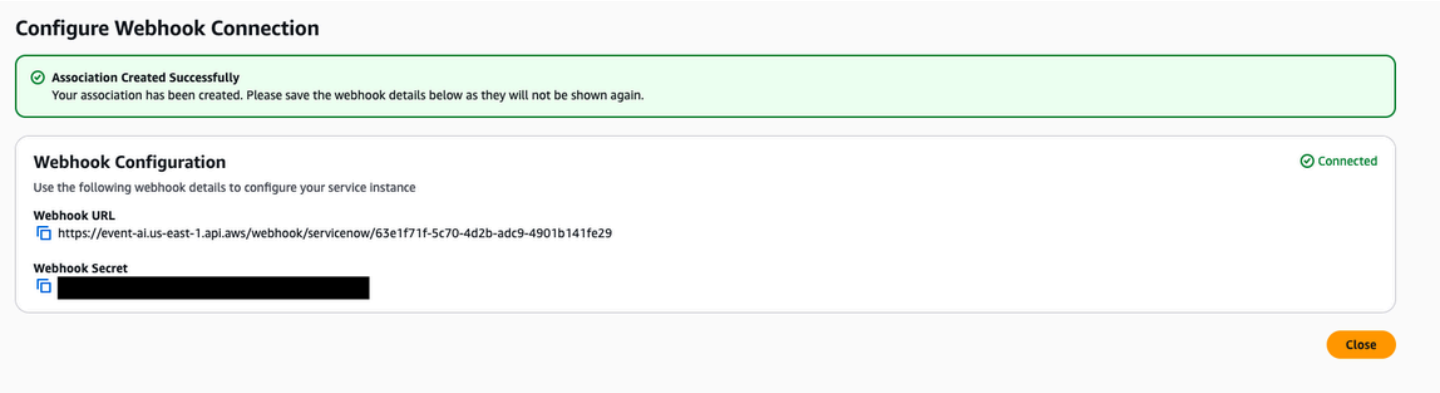
Connect il tuo ServiceNow client OAuth a AWS DevOps Agente

1. Puoi iniziare questo processo in due punti. Innanzitutto, accedendo alla pagina Capability Provider e trovandola nella ServiceNowsezione Comunicazione, quindi facendo clic su Registra. In alternativa, puoi selezionare qualsiasi DevOps Agent Space che potresti aver creato e accedere a Capacità → Comunicazioni → Aggiungi → ServiceNow e fare clic su Registra.
2. Successivamente, autorizza l' DevOps agente ad accedere alla tua ServiceNow istanza utilizzando il client applicativo OAuth che hai appena creato.



- Segui i passaggi successivi e salva le informazioni risultanti sul webhook

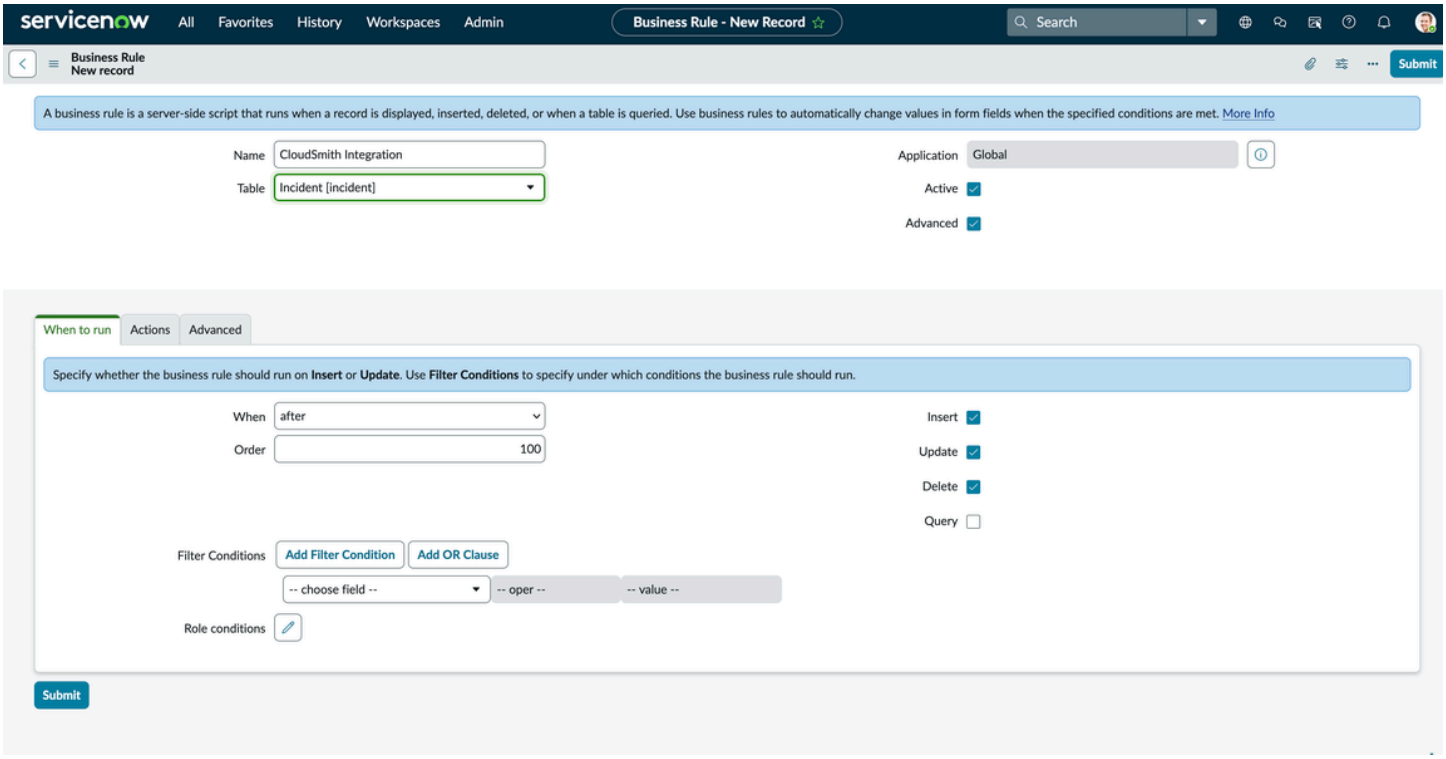
> **Important**
>
> Queste informazioni non verranno più visualizzate



Configura la tua regola aziendale ServiceNow

Una volta stabilita la connettività, dovrai configurare una regola aziendale per ServiceNow inviare i ticket al/i tuo/i DevOps Agent Space.

1. Vai su Activity Subscriptions → Administration → Business Rules e fai clic su Nuovo.
2. Imposta il campo «Tabella» su «Incidente [incidente]», seleziona la casella «Avanzate» e imposta la regola in modo che venga eseguita dopo l'inserimento, l'aggiornamento e l'eliminazione.



1. Vai alla scheda «Avanzate» e aggiungi il seguente script webhook, inserendo il segreto e l'URL del webhook dove indicato, e fai clic su Invia.

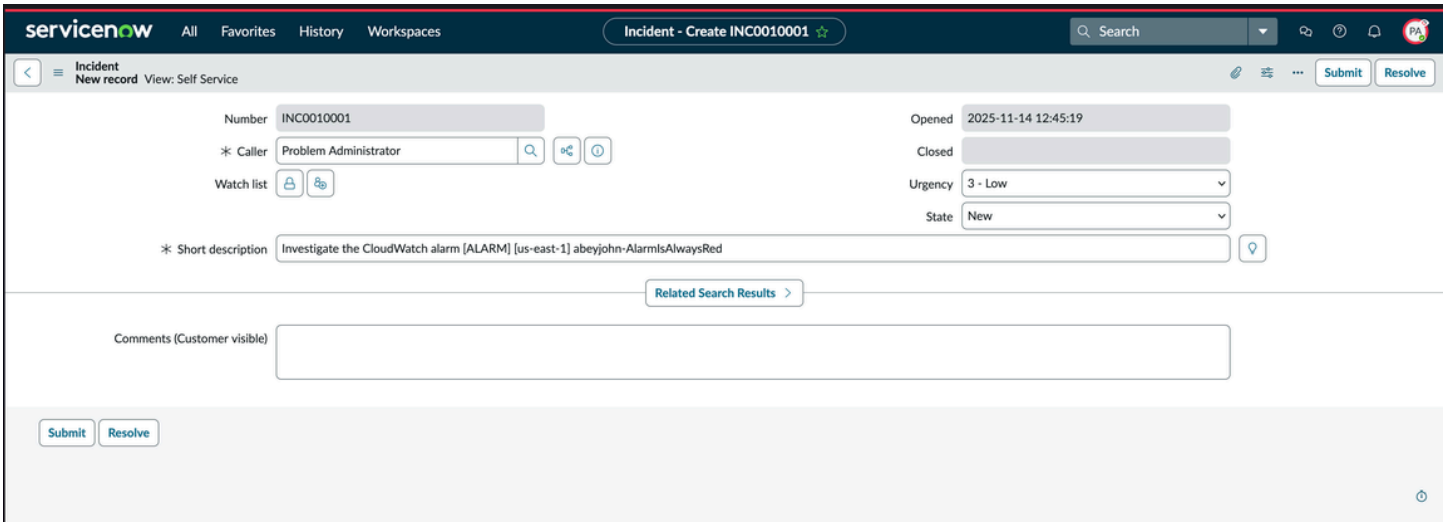
```
(function executeRule(current, previous /*null when async*/ ) {

    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);
```

Se hai scelto di registrare la ServiceNow connessione dalla pagina Provider di funzionalità, ora devi accedere all' DevOps Agent Space in cui desideri esaminare i ticket relativi agli ServiceNow incidenti, selezionare Capacità → Comunicazioni e quindi registrare l' ServiceNow istanza registrata nella pagina Provider di capacità. Ora, tutto dovrebbe essere configurato e tutti gli incidenti in cui il chiamante è impostato su «Problem Administrator» (per simulare le autorizzazioni concesse al client AWS DevOps OAuth) attiveranno un'indagine sulla risposta agli incidenti nello spazio DevOps agente configurato. Puoi verificarlo creando un nuovo incidente ServiceNow e impostando il campo Caller dell'incidente come «Problem Administrator».



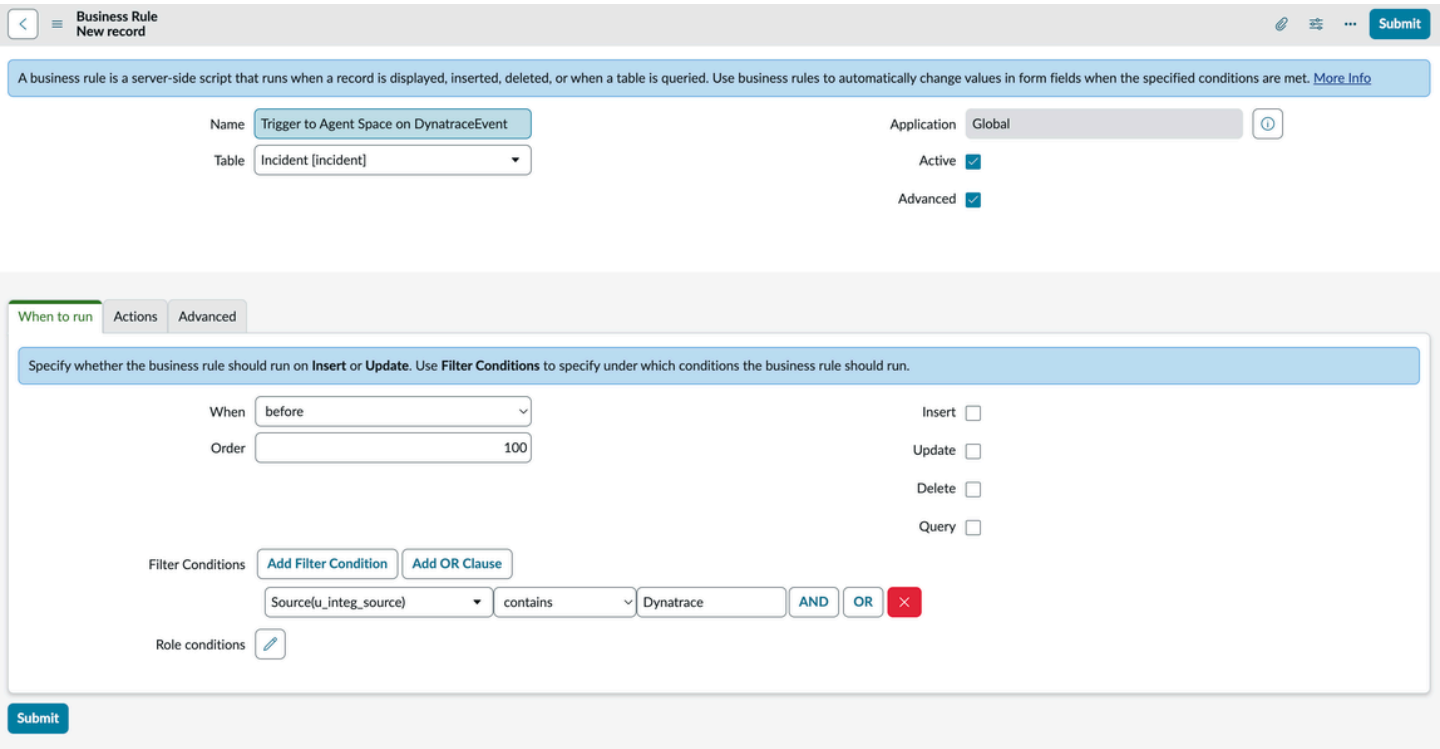
ServiceNow aggiornamenti dei ticket

Durante tutte le indagini relative alla risposta agli incidenti innescati, il tuo DevOps agente fornirà aggiornamenti sui risultati principali, sulle analisi delle cause principali e sui piani di mitigazione nel ticket di origine. I risultati dell'agente vengono pubblicati nei commenti relativi a un incidente e al momento pubblicheremo solo i dati relativi al tipo`finding`, `cause` `investigation_summary``mitigation_summary`, e agli aggiornamenti sullo stato delle indagini (ad esempio). `AWS DevOps Agent started/finished its investigation`

Esempi di routing e orchestrazione dei ticket

Scenario: filtraggio degli incidenti inviati a un Agent Space DevOps

Questo è uno scenario semplice ma richiede una configurazione ServiceNow per creare un campo in cui ServiceNow tracciare l'origine dell'incidente. Ai fini di questo esempio, crea un nuovo campo Source (u_source) utilizzando il generatore di moduli SNOW. Ciò consentirà di tracciare l'origine dell'incidente e di utilizzarla per indirizzare le richieste da una particolare fonte a un DevOps Agent Space. Il routing viene eseguito creando una regola aziendale di Service Now e nella scheda Quando eseguire impostando i trigger «When» e «Filter Conditions». In questo esempio le condizioni di filtro sono impostate come segue:

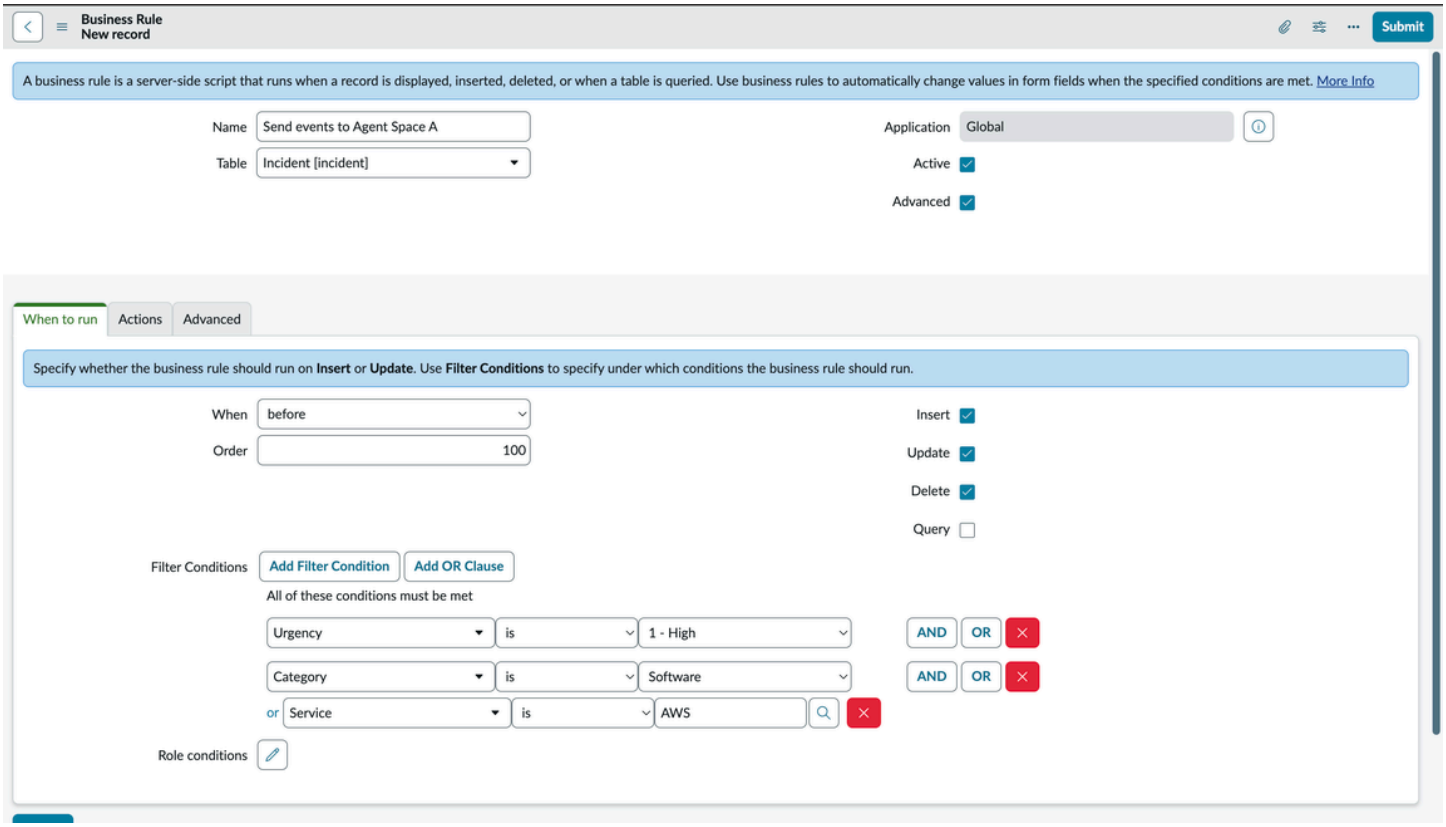


Scenario: instradamento degli incidenti su più Agent Spaces DevOps

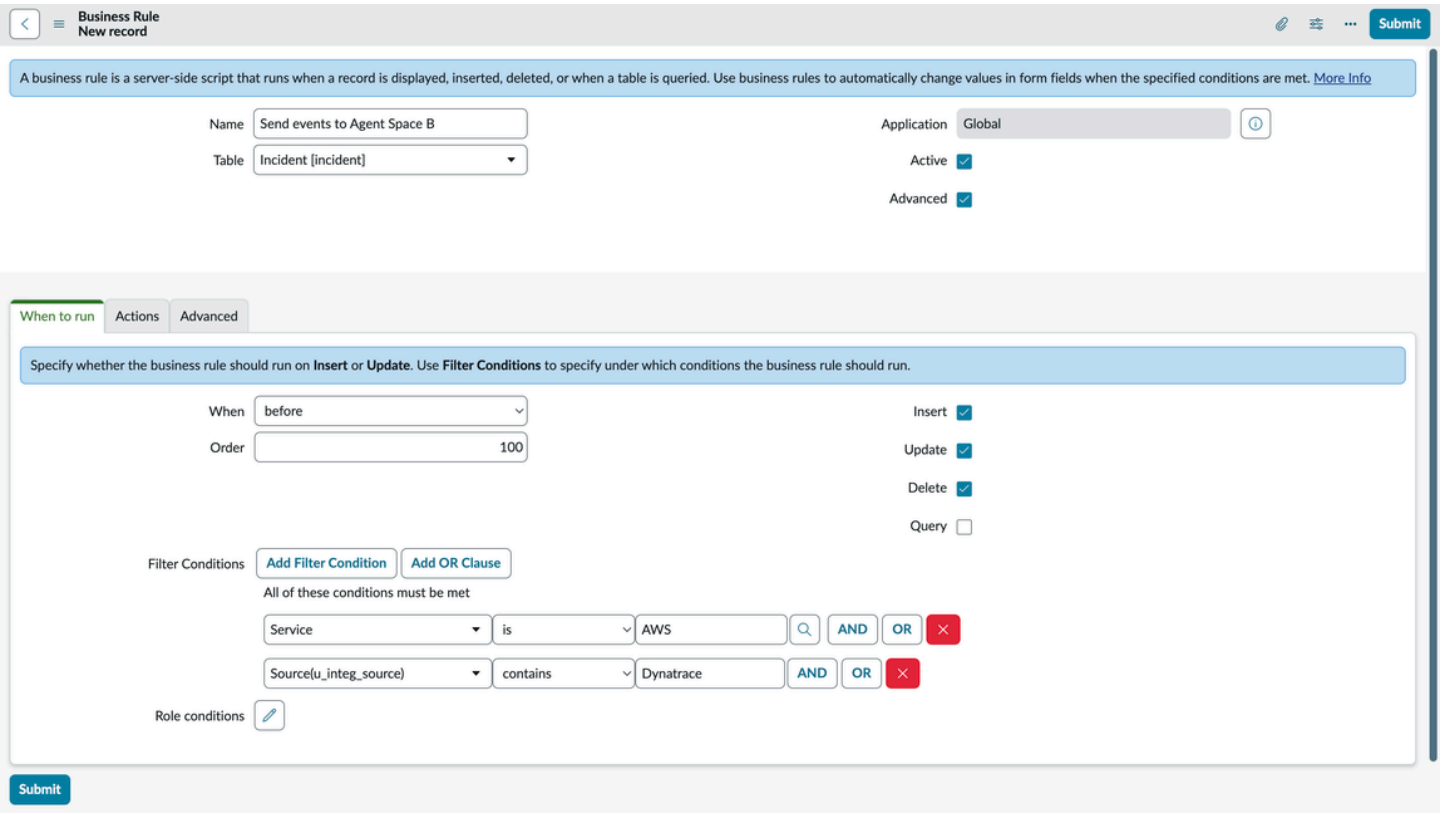
Questo esempio mostra come attivare un'indagine in DevOps Agent Space B quando l'urgenza è`1`, la categoria è `Software` o il servizio è `AWS` e avviare un'indagine in DevOps Agent Space A quando il servizio è e l'origine `AWS` sì. `Dynatrace`

Questo scenario può essere realizzato in due modi. Lo script webhook stesso può essere aggiornato per includere questa logica aziendale. In questo scenario mostreremo come realizzarlo con una ServiceNow Business Rule, per la trasparenza e la semplificazione del debug. Il routing viene eseguito creando due regole aziendali di Service Now.

- Crea una regola aziendale in ServiceNow DevOps Agent Space A e crea una condizione utilizzando il generatore di condizioni per inviare solo gli eventi in base alla nostra condizione specificata.



- Quindi, crea un'altra regola aziendale in ServiceNow per AgentSpace B per la quale la regola aziendale verrà attivata solo quando il servizio è AWS e l'origine è Dynatrace.



Ora, quando crei un nuovo Incidente che corrisponde alla condizione specificata, verrà avviata un'indagine su DevOps Agent Space A o DevOps Agent Space B, fornendoti un controllo granulare sul routing degli incidenti.

Connessione a Slack

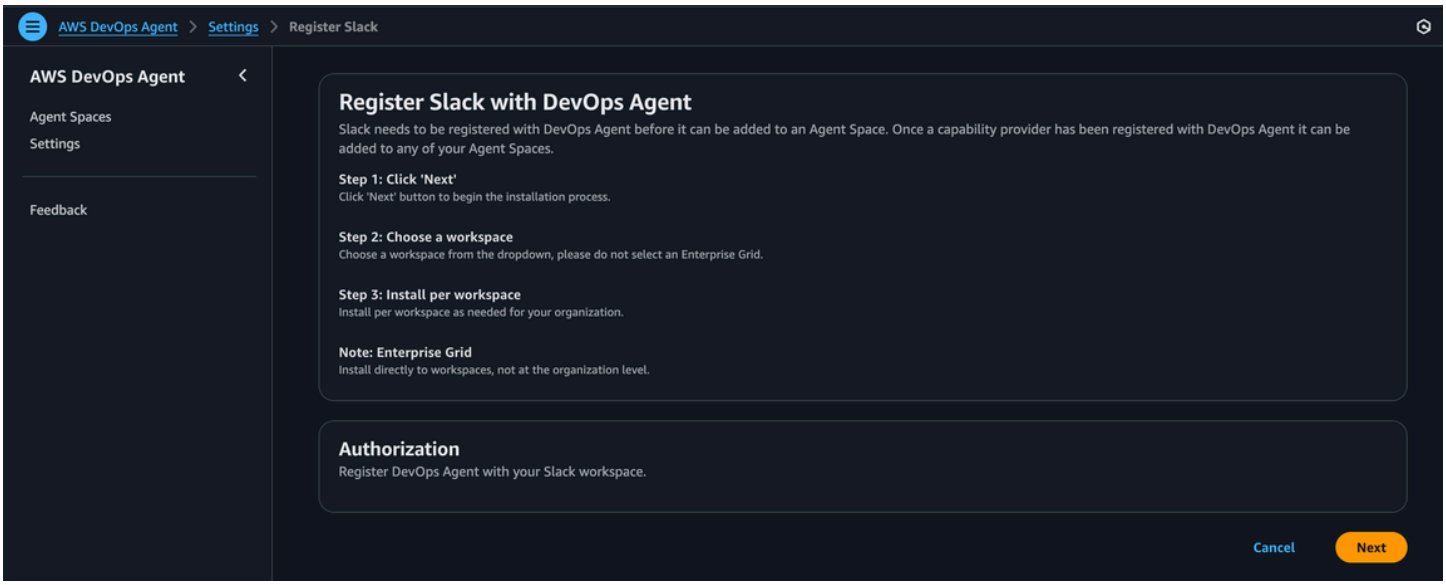
Puoi configurare AWS DevOps Agent in modo che aggiorni un canale Slack selezionato con indagini sulla risposta agli incidenti, risultati chiave, analisi delle cause principali e piani di mitigazione generati.

Prima di iniziare

Slack deve essere registrato con DevOps Agent prima di poter essere aggiunto a un Agent Space. Per integrare AWS DevOps Agent con Slack devi soddisfare questi requisiti:

- Accedi a uno spazio di lavoro Slack con la possibilità di installare e autorizzare applicazioni di terze parti
- Hai identificato i canali Slack a cui desideri che Agent invii notifiche AWS DevOps

Registra l'integrazione di Slack con Agent AWS DevOps



1. Dalla pagina Capability Provider della console dell' AWS DevOps agente, trova Slack nella sezione Provider disponibili in Comunicazione e fai clic su Registra.
2. Scegli il pulsante Registra.
3. Verrai reindirizzato a Slack per autorizzare l'applicazione AWS DevOps Agent per il tuo spazio di lavoro.
4. Nella pagina di autorizzazione di Slack, esegui l'installazione direttamente nelle aree di lavoro, non a livello di organizzazione.
5. Scegli uno spazio di lavoro dal menu a discesa. Non selezionate un'Enterprise Grid.
6. Eseguite l'installazione per area di lavoro in base alle esigenze dell'organizzazione.
7. Esamina gli ambiti richiesti e fai clic su Consenti per autorizzare l'integrazione.
8. Dopo l'autorizzazione, tornerai alla console dell' AWS DevOps agente.

Associa Slack ai tuoi spazi DevOps Agent

Dopo aver registrato Slack nel tuo DevOps Agent Space, puoi associarlo ai tuoi spazi DevOps Agent:

1. Dalla scheda Funzionalità all'interno della configurazione AgentSpace, accedi a Comunicazioni > Slack.
2. Seleziona Aggiungi Slack
3. Inserisci l'ID del canale

4. Scegli Crea per completare la configurazione di Slack.

> **Note**
>
> L'utente bot dell'agente deve essere aggiunto ai canali privati prima di poter pubblicare messaggi.

> **Important**
>
> La disinstallazione dell'app Slack potrebbe impedire la reinstallazione dell'app Slack. Evita di disinstallare l'app Slack.

Richiamo DevOps dell'agente tramite Webhook

I webhook consentono ai sistemi esterni di avviare automaticamente le indagini degli agenti. AWS DevOps Ciò consente l'integrazione con sistemi di ticketing, strumenti di monitoraggio e altre piattaforme in grado di inviare richieste HTTP in caso di incidenti.

Prerequisiti

Prima di configurare l'accesso ai webhook, assicurati di avere:

- Un Agent Space configurato in Agent AWS DevOps
- Accesso alla console dell' AWS DevOps agente
- Il sistema esterno che invierà le richieste di webhook

Tipi di webhook

AWS DevOps Agent supporta i seguenti tipi di webhook:

- Integration-specific webhook: generati automaticamente quando configuri integrazioni di terze parti come Dynatrace, Splunk, Datadog, New Relic o Slack. ServiceNow Questi webhook sono associati all'integrazione specifica e utilizzano metodi di autenticazione determinati dal tipo di integrazione

- Webhook generici: possono essere creati manualmente per avviare indagini da qualsiasi fonte non coperta da un'integrazione specifica. I webhook generici attualmente utilizzano l'autenticazione HMAC (il token bearer non è attualmente disponibile).
- Webhook di avviso Grafana: Grafana può inviare notifiche di avviso direttamente AWS DevOps all'agente tramite i punti di contatto webhook. Per istruzioni di configurazione che includono un modello di notifica personalizzato, vedi Connecting Grafana.

Metodi di autenticazione Webhook

Il metodo di autenticazione per il webhook dipende dall'integrazione a cui è associato:

Autenticazione HMAC: utilizzata da:

- Webhook di integrazione con Dynatrace
- Webhook generici (non collegati a una specifica integrazione di terze parti)

Autenticazione con token Bearer: utilizzata da:

- Webhook di integrazione Splunk
- Webhook di integrazione Datadog
- Nuovi webhook di integrazione con Relic
- ServiceNow webhook di integrazione
- webhook di integrazione con Slack
- Webhook di integrazione Grafana

Comprendere l'autenticazione HMAC

HMAC (Hash-based Message Authentication Code) è un meccanismo crittografico che verifica l'integrità e l'autenticità di una richiesta webhook. Quando si invia un webhook con autenticazione HMAC, si genera una firma eseguendo l'hashing del timestamp e del payload della richiesta utilizzando la chiave segreta con l'algoritmo. SHA-256 AWS DevOps L'agente calcola in modo indipendente lo stesso hash su un lato e confronta le due firme. Se corrispondono, la richiesta viene accettata.

Poiché il timestamp è incluso nella firma, HMAC fornisce anche una protezione dalla riproduzione: l' AWS DevOps agente può rifiutare le richieste con timestamp che risalgono troppo al passato, impedendo a un utente malintenzionato di acquisire e inviare nuovamente una richiesta valida.

Scelta tra HMAC e token Bearer

Considerazione	HMAC	Token Bearer
Complessità di configurazione	Più complesso: il cliente deve calcolare una firma per ogni richiesta utilizzando il timestamp e il payload	Più semplice: includi un token statico nell'intestazione `Authorization`
Integrità del carico utile	Verificato: qualsiasi modifica al payload dopo la firma invalida la firma	Non verificato: il token autentica il mittente ma non protegge il contenuto del payload
Protezione da replay	Built-in — il timestamp nella firma consente al server di rifiutare le richieste obsolete	Non integrato: un token acquisito può essere riutilizz ato finché non viene ruotato
Rischio di esposizione segreto	Inferiore: il segreto non viene mai trasmesso nella richiesta ; viene inviata solo la firma computerizzata	Più alto: il token viene inviato in ogni intestazione di richiesta , aumentando l'esposizione se il traffico viene intercettato
Quando utilizzare	Consigliato quando sono necessarie garanzie di sicurezza più solide, ad esempio per webhook generici o ambienti con requisiti di conformità rigorosi	Ideale quando la facilità di integrazione è una priorità e il trasporto di rete è affidabile, ad esempio per le integrazioni SaaS gestite tramite HTTPS

> Nota: il metodo di autenticazione è determinato dal tipo di integrazione. Integration-specific i webhook (Splunk, Datadog, New Relic, ServiceNow Slack, Grafana) utilizzano l'autenticazione con token al portatore. Dynatrace e i webhook generici utilizzano

l'autenticazione HMAC. Non è possibile modificare il metodo di autenticazione per un webhook specifico per l'integrazione.

Configurazione dell'accesso al webhook

Fase 1: Accedere alla configurazione del webhook

1. Accedi alla console di AWS gestione e vai alla console dell' AWS DevOps agente
2. Seleziona il tuo Agent Space
3. Vai alla scheda Funzionalità
4. Nella sezione Webhook, fai clic su Configura

Fase 2: Generazione delle credenziali del webhook

Per webhook specifici per l'integrazione:

I webhook vengono generati automaticamente quando si completa la configurazione di un'integrazione di terze parti. L'URL e le credenziali dell'endpoint webhook vengono forniti al termine del processo di configurazione dell'integrazione.

Per i webhook generici:

1. Fai clic su Genera webhook
2. Il sistema genererà una coppia di key pair HMAC
3. Archivia in modo sicuro la chiave e il segreto generati: non potrai più recuperarli
4. Copia l'URL dell'endpoint del webhook fornito

Fase 3: Configurazione del sistema esterno

Utilizza l'URL e le credenziali dell'endpoint webhook per configurare il tuo sistema esterno per l'invio di richieste all'agente. AWS DevOps I passaggi di configurazione specifici dipendono dal sistema esterno.

Gestione delle credenziali del webhook

Rimozione delle credenziali: per eliminare le credenziali del webhook, vai alla sezione di configurazione del webhook e fai clic su Rimuovi. Dopo aver rimosso le credenziali, l'endpoint webhook non accetterà più richieste finché non ne genererai di nuove.

Rigenerazione delle credenziali: per generare nuove credenziali, rimuovi prima le credenziali esistenti, quindi genera una nuova coppia di chiavi o token.

Utilizzo del webhook

Formato di richiesta Webhook

Per avviare un'indagine, il sistema esterno deve inviare una richiesta POST HTTP all'URL dell'endpoint del webhook.

Per la versione 1 (autenticazione HMAC):

Intestazioni:

- `Content-Type: application/json`
- `x-amzn-event-signature: <HMAC signature>`
- `x-amzn-event-timestamp: <+%Y-%m-%dT%H:%M:%S.000Z>`

La firma HMAC viene generata firmando il corpo della richiesta con la chiave segreta utilizzando. SHA-256

Per la versione 2 (autenticazione con token Bearer):

Intestazioni:

- `Content-Type: application/json`
- `Authorization: Bearer <your-token>`

Corpo della richiesta:

Il corpo della richiesta deve includere informazioni sull'incidente:

```
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"
    }
  }
}
```

Schema del payload:

```
{
    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;
}
```

Codice di esempio

Versione 1 (autenticazione HMAC) -: 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);
});
```

Versione 1 (autenticazione HMAC) - 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"
```

Versione 2 (autenticazione con token Bearer) -: 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),
  });
}
```

Versione 2 (autenticazione con token Bearer) - 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"
```

Risoluzione dei problemi relativi ai webhook

Se non ricevi un 200

Un 200 e un messaggio simile a un webhook ricevuto indicano che l'autenticazione è stata superata e il messaggio è stato messo in coda per essere verificato ed elaborato dal sistema. Se non ottieni un 200 ma un 4xx, molto probabilmente c'è qualcosa che non va nell'autenticazione o nelle intestazioni. Prova a inviare manualmente utilizzando le opzioni curl per aiutare a eseguire il debug dell'autenticazione.

Se ricevi un 200 ma non viene avviata un'indagine

La causa probabile è un payload non formattato.

1. Verifica che sia il timestamp che l'ID dell'incidente siano aggiornati e unici. I messaggi duplicati vengono deduplicati.
2. Verifica che il messaggio sia JSON valido
3. Verifica che il formato sia corretto

Se ricevi 200 dollari e l'indagine viene immediatamente annullata

Molto probabilmente hai raggiunto il limite mensile. Rivolgiti al tuo AWS contatto per chiedere una modifica del limite di tariffa, se del caso.

Argomenti correlati

- the section called “Creazione di uno spazio per agenti”
- the section called “Cos'è un'app Web per DevOps agenti?”
- the section called “DevOps Autorizzazioni Agent IAM”

Integrazione AWS DevOps Agente con Amazon EventBridge

Puoi integrare AWS DevOps Agent con le tue applicazioni basate sugli eventi utilizzando gli eventi che si verificano durante i cicli di vita di indagine e mitigazione. AWS DevOps L'agente invia eventi ad Amazon EventBridge quando lo stato di un'indagine o di mitigazione cambia. Puoi quindi creare EventBridge regole che agiscano in base a questi eventi.

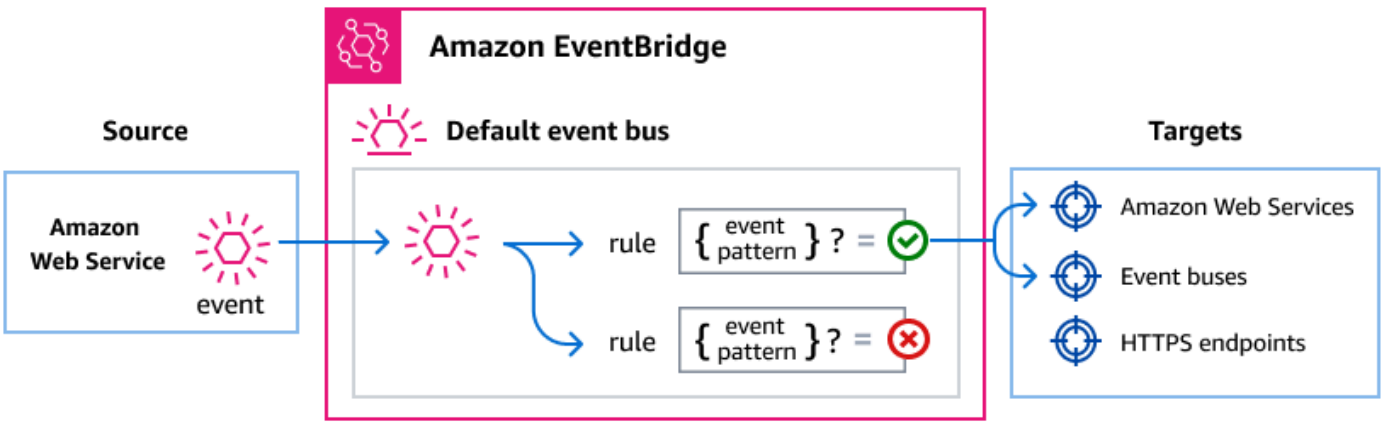
Ad esempio, è possibile creare regole che eseguono le seguenti azioni:

- Richiama una funzione AWS Lambda per elaborare i risultati dell'indagine al termine di un'indagine.
- Invia una notifica Amazon SNS quando un'indagine fallisce o scade.
- Aggiorna un sistema di ticketing quando viene creata una nuova indagine.
- Avvia un flusso di lavoro AWS Step Functions al termine di un'azione di mitigazione.

Come sono i percorsi EventBridge AWS DevOps Eventi per agenti

AWS DevOps L'agente invia gli eventi al bus degli eventi EventBridge predefinito. EventBridge quindi valuta gli eventi in base alle regole create dall'utente. Quando un evento corrisponde al modello di evento di una regola, EventBridge invia l'evento ai target specificati.

Il diagramma seguente mostra come EventBridge indirizza gli eventi AWS DevOps dell'agente.



1. AWS DevOps L'agente invia un evento al bus degli eventi EventBridge predefinito quando lo stato del ciclo di vita di un'indagine o di mitigazione cambia.
2. EventBridge valuta l'evento in base alle regole che hai creato.

3. Se l'evento corrisponde allo schema di eventi di una regola, EventBridge invia l'evento ai target specificati nella regola.

AWS DevOps Eventi dell'agente

AWS DevOps L'agente invia i seguenti eventi a EventBridge. Tutti gli eventi utilizzano la fonte`aws.aidevops`.

Eventi investigativi supportati

detail-type (tipo di dettaglio)	Description
`Investigation Created`	È stata creata un'indagine nello spazio dedicato agli agenti.
`Investigation Priority Updated`	La priorità di un'indagine è stata modificata.
`Investigation In Progress`	Un'indagine ha avviato un'analisi attiva.
`Investigation Completed`	Un'indagine si è conclusa con successo con i risultati.
`Investigation Failed`	Un'indagine ha rilevato un errore e non è stata completata.
`Investigation Timed Out`	Un'indagine ha superato la durata massima consentita.
`Investigation Cancelled`	Un'indagine è stata annullata prima del completamento.
`Investigation Pending Triage`	Un'indagine è in attesa di valutazione prima che inizi l'analisi attiva.
`Investigation Linked`	Un'indagine era collegata a un incidente o a un ticket correlato.
`Investigation Skipped`	Un'indagine è stata ignorata perché corrispon deva ai criteri di salto definiti in un'abilità.

Eventi di mitigazione supportati

detail-type (tipo di dettaglio)	Description
`Mitigation In Progress`	È iniziata un'azione di mitigazione.
`Mitigation Completed`	Un'azione di mitigazione è stata completata con successo.
`Mitigation Failed`	Un'azione di mitigazione ha rilevato un errore e non è stata completata.
`Mitigation Timed Out`	Un'azione di mitigazione ha superato la durata massima consentita.
`Mitigation Cancelled`	Un'azione di mitigazione è stata annullata prima del completamento.

Per descrizioni dettagliate dei campi ed eventi di esempio, vedere. the section called "AWS DevOps Riferimento dettagliato sugli eventi degli agenti"

Creazione di modelli di eventi che corrispondano AWS DevOps Eventi per agenti

EventBridge le regole utilizzano modelli di eventi per selezionare gli eventi e indirizzarli verso gli obiettivi. Un modello di eventi corrisponde alla struttura degli eventi che gestisce. Si creano modelli di eventi per filtrare gli eventi AWS DevOps dell'agente in base ai campi degli eventi.

Gli esempi seguenti mostrano modelli di eventi per casi d'uso comuni.

Abbina tutti gli eventi AWS DevOps dell'agente

Il seguente schema di eventi corrisponde a tutti gli eventi di AWS DevOps Agent.

```
{
  "source": ["aws.aidevops"]
}
```

Abbina solo gli eventi investigativi

Il seguente modello di eventi utilizza un prefisso match per selezionare solo gli eventi del ciclo di vita dell'indagine.

```
{
  "source": ["aws.aidevops"],
  "detail-type": [{"prefix": "Investigation"}]
}
```

Abbina solo gli eventi di completamento e di fallimento

Il seguente schema di eventi corrisponde agli eventi relativi a indagini e mitigazioni completate o non riuscite.

```
{
  "source": ["aws.aidevops"],
  "detail-type": [
    "Investigation Completed",
    "Investigation Failed",
    "Mitigation Completed",
    "Mitigation Failed"
  ]
}
```

Abbina gli eventi per uno spazio agente specifico

Il seguente schema di eventi corrisponde agli eventi di uno spazio agente specifico.

```
{
  "source": ["aws.aidevops"],
  "detail": {
    "metadata": {
      "agent_space_id": ["your-agent-space-id"]
    }
  }
}
```

Per ulteriori informazioni sui modelli di eventi, consulta i modelli di EventBridge eventi di Amazon nella Amazon EventBridge User Guide.

EventBridge Autorizzazioni Amazon

AWS DevOps L'agente non richiede autorizzazioni aggiuntive per fornire eventi. EventBridge Gli eventi vengono inviati automaticamente al bus eventi predefinito.

A seconda delle destinazioni configurate per le EventBridge regole, potrebbe essere necessario aggiungere autorizzazioni specifiche. Per ulteriori informazioni sulle autorizzazioni richieste per gli obiettivi, consulta Using resource-based policies for Amazon nella Amazon User EventBridge Guide. EventBridge

Risorse aggiuntive EventBridge

Per ulteriori informazioni su EventBridge concetti e configurazione, consulta i seguenti argomenti nella Amazon EventBridge User Guide:

- EventBridge autobus per eventi
- EventBridge eventi
- EventBridge modelli di eventi
- EventBridge regole
- EventBridge obiettivi

AWS DevOps Riferimento dettagliato sugli eventi degli agenti

Gli eventi dei AWS servizi hanno campi di metadati comuni, tra cui `source`detail-type,, `account`region, `time`. Questi eventi contengono anche un `detail` campo con dati specifici del servizio. Per gli eventi dell' AWS DevOps agente, `source` è sempre `aws.aidevops` e `detail-type` identifica l'evento specifico.

Eventi investigativi

I seguenti `detail-type` valori identificano gli eventi investigativi:

- `Investigation Created`
- `Investigation Priority Updated`
- `Investigation In Progress`
- `Investigation Completed`
- `Investigation Failed`

- `Investigation Timed Out`
- `Investigation Cancelled`
- `Investigation Pending Triage`
- `Investigation Linked`
- `Investigation Skipped`

I `detail-type` campi `source` e sono inclusi di seguito perché contengono valori specifici per gli eventi AWS DevOps dell'agente. Per le definizioni degli altri campi di metadati inclusi in tutti gli eventi, consulta la struttura degli eventi in Amazon EventBridge Events Reference.

Di seguito è riportata la struttura JSON per gli eventi investigativi.

```
{
  . . .,
  "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`Identifica il tipo di evento. Per gli eventi di indagine, questo è uno dei nomi degli eventi elencati in precedenza.

`source`Identifica il servizio che ha generato l'evento. Per gli eventi AWS DevOps dell'agente, questo valore è`aws.aidevops`.

`detail`Un oggetto JSON che contiene dati specifici dell'evento. L'`detail`oggetto include i seguenti campi:

- `version`(stringa) — La versione dello schema dei dettagli dell'evento. Attualmente`1.0.0`.
- `metadata.agent_space_id`(stringa) — L'identificatore univoco dello spazio agente in cui ha avuto origine l'evento.
- `metadata.task_id`(stringa) — L'identificatore univoco dell'attività.
- `metadata.execution_id`(stringa) — L'identificatore univoco dell'esecuzione. Presente quando un'esecuzione è stata assegnata all'indagine.
- `data.task_type`(stringa) — Il tipo di attività. Valore:`INVESTIGATION`.
- `data.priority`(stringa) — Il livello di priorità. Valori:`CRITICAL`,`HIGH`,`MEDIUM`,`LOW`,`MINIMAL`.
- `data.status`(stringa) — Lo stato corrente. Valori:`PENDING_START`,`IN_PROGRESS`, `COMPLETEDFAILED`,`TIMED_OUT`,`CANCELLED`,`PENDING_TRIAGE`,`LINKED`,`SKIPPED`.
- `data.created_at`(stringa) — Timestamp ISO 8601 al momento della creazione dell'attività.
- `data.updated_at`(stringa) — Data e ora ISO 8601 dell'ultimo aggiornamento dell'attività.
- `data.summary_record_id`(stringa) — L'identificatore della registrazione riassuntiva contenente i risultati dell'indagine. Incluso quando viene generato un riepilogo per l'indagine completata. È possibile recuperare il contenuto del riepilogo tramite l'API dell' AWS DevOps agente utilizzando questo identificatore per cercare il record del diario con un tipo di record di. `investigation_summary_md`

Esempio: evento Indagine completata

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

Esempio: Evento di indagine fallito

```
{
  "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"
```

```
    }
  }
}
```

Eventi di mitigazione

I seguenti `detail-type` valori identificano gli eventi di mitigazione:

- `Mitigation In Progress`
- `Mitigation Completed`
- `Mitigation Failed`
- `Mitigation Timed Out`
- `Mitigation Cancelled`

I `detail-type` campi `source` e sono inclusi di seguito perché contengono valori specifici per gli eventi AWS DevOps dell'agente. Per le definizioni degli altri campi di metadati inclusi in tutti gli eventi, consulta la struttura degli eventi in Amazon EventBridge Events Reference.

Di seguito è riportata la struttura JSON per gli eventi di mitigazione.

```
{
  . . .,
  "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`Identifica il tipo di evento. Per gli eventi di mitigazione, questo è uno dei nomi degli eventi elencati in precedenza.

`source`Identifica il servizio che ha generato l'evento. Per gli eventi AWS DevOps dell'agente, questo valore è`aws.aidevops`.

`detail`Un oggetto JSON che contiene dati specifici dell'evento. L'`detail`oggetto include i seguenti campi:

- `version`(stringa) — La versione dello schema dei dettagli dell'evento. Attualmente`1.0.0`.
- `metadata.agent_space_id`(stringa) — L'identificatore univoco dello spazio agente in cui ha avuto origine l'evento.
- `metadata.task_id`(stringa) — L'identificatore univoco dell'attività.
- `metadata.execution_id`(stringa) — L'identificatore univoco dell'esecuzione. Presente quando un'esecuzione è stata assegnata alla mitigazione.
- `data.task_type`(stringa) — Il tipo di attività. Valore:`INVESTIGATION`.
- `data.priority`(stringa) — Il livello di priorità. Valori:`CRITICAL`,`HIGH`,`MEDIUM`,`LOW`,`MINIMAL`.
- `data.status`(stringa) — Lo stato corrente. Valori:`IN_PROGRESS`,`COMPLETED`,`FAILED`,`TIMED_OUT`,`CANCELLED`.
- `data.created_at`(stringa) — Timestamp ISO 8601 al momento della creazione dell'attività.
- `data.updated_at`(stringa) — Data e ora ISO 8601 dell'ultimo aggiornamento dell'attività.
- `data.summary_record_id`(stringa) — L'identificatore del record di riepilogo contenente i risultati della mitigazione. Incluso quando viene generato un riepilogo per la mitigazione completata. È possibile recuperare il contenuto del riepilogo tramite l' AWS DevOps Agent API utilizzando questo identificatore per cercare il record del journal con un tipo di record di. `mitigation_summary_md`

Esempio: evento Mitigation Completed

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

Esempio: evento di mitigazione non riuscito

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

Registri e metriche venduti

Puoi monitorare gli spazi degli agenti e le operazioni di servizio utilizzando i CloudWatch parametri e i log di Amazon forniti. Questo argomento descrive le CloudWatch metriche che l' AWS DevOps agente pubblica automaticamente sul tuo account e i log forniti che puoi configurare per la consegna alle tue destinazioni preferite.

Metriche vendute CloudWatch

AWS DevOps L'agente pubblica automaticamente le metriche su Amazon CloudWatch nel tuo account. Queste metriche sono disponibili senza alcuna configurazione. È possibile utilizzarle per monitorare l'utilizzo, tenere traccia dell'attività operativa e creare allarmi.

Ruolo collegato ai servizi

Per far sì che i CloudWatch parametri Amazon vengano pubblicati nel tuo account per questo servizio, AWS DevOps Agent creerà automaticamente il ruolo collegato al servizio AWSServiceRoleForAIDevOps Service-Linked Role per te. Se il ruolo IAM che richiama l'API non dispone dell'autorizzazione appropriata, la creazione della risorsa avrà esito negativo con un. InvalidParameterException

> **Important**
>
> I clienti che hanno creato il proprio ruolo AgentSpace prima del 13 marzo 2026 dovranno creare manualmente il ruolo collegato al servizio AWSServiceRoleForAIDevOps per pubblicare le CloudWatch metriche relative all' AWS DevOps agente nel proprio account.

Crea manualmente un ruolo collegato al servizio (per i clienti esistenti)

Esegui una delle seguenti operazioni:

- Nella console IAM, crea il ruolo AWSServiceRoleForAIDevOps nel servizio AWS DevOps Agent.
- Dalla AWS CLI, esegui il seguente comando:

```
aws iam create-service-linked-role --aws-service-name aidevops.amazonaws.com
```

Namespace

Tutte le metriche sono pubblicate nel namespace. `AWS/AIDevOps`

Dimensioni

Tutte le metriche includono la seguente dimensione.

Dimensione	Description
`AgentSpaceUUID`	L'identificatore univoco dello spazio degli agenti. Per aggregare le metriche relative a tutti gli spazi degli agenti del tuo account, utilizza espressioni CloudWatch matematiche o ometti il filtro dimensionale.

Riferimento per le metriche

Nome parametro	Description	Unità	Frequenza di pubblicazione	Statistiche utili
ConsumedChatRequests	Il numero di richieste di chat consumate dallo spazio di un agente. Per ottenere	Conteggio	Ogni 5 minuti	Somma, media

Nome parametro	Description	Unità	Frequenza di pubblicazione	Statistiche utili
	il conteggio totale del tuo account, utilizza la `SUM` statistic a per tutte le `AgentSpaceUUID` dimensioni.			
ConsumedInvestigationTime	Il tempo impiegato a condurre indagini in uno spazio riservato agli agenti.	Secondi	Ogni 5 minuti	Somma, media, massimo
ConsumedEvaluationTime	Il tempo impiegato per eseguire le valutazioni in uno spazio dedicato agli agenti.	Secondi	Ogni 5 minuti	Somma, media, massimo

Nome parametro	Description	Unità	Frequenza di pubblicazione	Statistiche utili
TopologyCompletionCount	Il numero di completamenti dell'elaborazione della topologia. AWS DevOps L'agente emette questa metrica al termine dell'elaborazione di una topologia, che si tratti della creazione iniziale durante l'onboarding, di un aggiornamento manuale o di un aggiornamento giornaliero pianificato.	Conteggio	Basato sugli eventi (emesso a ogni completamento)	Somma, SampleCount

Visualizzazione delle metriche nella console CloudWatch

1. Apri la CloudWatch console.
2. Nel pannello di navigazione, scegli Parametri quindi scegli Tutti i parametri.
3. Scegli lo spazio dei nomi AWS/AIDevOps.
4. Scegli By AgentSpace per visualizzare le metriche relative agli spazi riservati agli agenti.

> **Note**
>
> Puoi creare CloudWatch allarmi in base a queste metriche per ricevere notifiche quando l'utilizzo supera una soglia. Ad esempio, crea un allarme per monitorare il consumo delle `ConsumedChatRequests` richieste di chat.

Prerequisiti

Prima di configurare la consegna dei log, assicurati di disporre di quanto segue:

- Un AWS account attivo con accesso alla console dell' AWS DevOps agente
- Un preside IAM con autorizzazioni per la consegna dei CloudWatch log APIs
- (Facoltativo) Un bucket Amazon S3 o un flusso di distribuzione Amazon Data Firehose, se prevedi di utilizzarli come destinazioni di log

Registri venduti

AWS DevOps L'agente supporta i log forniti che forniscono visibilità sugli eventi elaborati dagli spazi degli agenti e dalle registrazioni dei servizi. I registri venduti utilizzano l'infrastruttura Amazon CloudWatch Logs per consegnare i log alla destinazione preferita.

Per utilizzare i registri venduti, devi configurare una destinazione di consegna. Sono supportate le seguenti destinazioni:

- Amazon CloudWatch Logs: un gruppo di log nel tuo account
- Amazon S3: un bucket S3 nel tuo account
- Amazon Data Firehose: un flusso di distribuzione di Firehose nel tuo account

Tipi di log supportati

È supportato un solo tipo di registro:. `APPLICATION_LOGS` Questo tipo di registro copre tutti gli eventi operativi emessi dal servizio.

Registra i tipi di eventi

La tabella seguente riepiloga gli eventi registrati dall' AWS DevOps agente.

Event	Description	Livello di log
Evento in entrata dell'agente ricevuto	Un agente viene attivato da una fonte integrata e riceve un evento in entrata (ad esempio, un evento PagerDuty incidente).	INFO
L'evento in entrata dell'agente è stato interrotto	Un evento in entrata è stato eliminato prima che l'agente lo elaborasse. Il registro include il motivo (ad esempio, dati non validi).	TBD
Errore di comunicazione in uscita dell'agente	Una comunicazione in uscita con un'integrazione di terze parti non è riuscita. Il registro include l'ID dell'attività e l'identificatore di destinazi one (ad esempio, un errore di autenticazione).	TBD
Creazione della topologia in coda	Un processo di creazione della topologia era in coda per l'elaborazione.	INFO
La creazione della topologia è iniziata	È iniziata l'elaborazione di un processo di creazione della topologia.	INFO
Creazione della topologia terminata	Elaborazione completata di un processo di creazione della topologia. Questo evento si applica alla creazione iniziale, agli aggiornamenti e agli aggiornamenti giornalieri.	INFO

Event	Description	Livello di log
Individuazione delle risorse non riuscita	Si è verificato un errore nell'individuazione delle risorse durante la creazione della topologia.	ERRORE
Registrazione del servizio non riuscita	La registrazione del servizio rileva un errore irreversibile	ERRORE
La convalida del webhook fallisce	Quando il webhook ricevuto dall'agente Devops non corrisponde allo schema previsto	ERRORE
Aggiornamenti dello stato di convalida dell'associazione	Quando si verifica un'associazione nello spazio di un agente (primary/secondary account tipico), lo stato di convalida passa da valido a non valido e viceversa (ad esempio, a causa di un ruolo non valido, che non è ipotizzabile dal servizio).	ERRORE/INFORMAZIONI

Permissions

AWS DevOps L'agente utilizza i registri CloudWatch venduti (autorizzazioni V2) per fornire i log. Per configurare la consegna dei log, il ruolo IAM che configura la consegna deve disporre delle seguenti autorizzazioni:

- `aidevops:AllowVendedLogDeliveryForResource`— Necessario per consentire la consegna dei log per la risorsa dello spazio dell'agente.
- Autorizzazioni per la consegna CloudWatch dei log APIs (`logs:PutDeliverySource`, `logs:PutDeliveryDestinationlogs:CreateDelivery`, e operazioni correlate).
- Autorizzazioni specifiche per la destinazione di consegna scelta.

Per la policy IAM completa richiesta per ogni tipo di destinazione, consulta i seguenti argomenti nella Amazon CloudWatch Logs User Guide:

- Log inviati a Logs CloudWatch
- Registri inviati ad Amazon S3
- Log inviati a Firehose

Configura la consegna dei log (console)

AWS DevOps L'agente fornisce due posizioni nella console di AWS gestione per configurare la consegna dei log:

- Pagina delle impostazioni di registrazione del servizio: configura la consegna dei log per gli eventi a livello di servizio. Questi log utilizzano il servizio ARN `arn:aws:aidevops:<region>:<account-id>:service/<account-id>` () come risorsa.
- Pagina Agent Space: configura la consegna dei log per gli eventi specifici di un singolo spazio agente. Questi log utilizzano lo spazio agente ARN `arn:aws:aidevops:<region>:<account-id>:agentspace/<agent-space-id>` () come risorsa.

Per configurare la consegna dei log per la registrazione di un servizio

1. Aprire la console AWS DevOps dell'agente nella console AWS di gestione.
2. Nel pannello di navigazione scegli Impostazioni.
3. Nella scheda Provider di capacità > Registri, scegli Configura.
4. Per Tipo di destinazione, scegliete una delle seguenti opzioni:
5. CloudWatch Registri: seleziona o crea un gruppo di registri.
6. Amazon S3: inserisci l'ARN del bucket S3.
7. Amazon Data Firehose: seleziona o crea un flusso di distribuzione Firehose.
8. Per le impostazioni aggiuntive, facoltativo, puoi specificare le seguenti opzioni:
   a. Per Selezione del campo, seleziona i nomi dei campi di log che desideri consegnare alla destinazione. È possibile selezionare i campi del registro degli accessi e un sottoinsieme di campi del registro degli accessi in tempo reale.
   b. (Solo Amazon S3) Per Partizionamento, specifica il percorso per partizionare i dati del file di log.

c. (Solo Amazon S3) Per Formato file compatibile con Hive, puoi selezionare la casella di controllo per utilizzare percorsi S3 compatibili con Hive. Questo consente di semplificare il caricamento di nuovi dati negli strumenti compatibili con Hive.

d. Per Formato di output, specifica il formato preferito.

e. Per Delimitatore di campo, specifica come separare i campi di log.

9. Scegli Save (Salva).

10. Verifica che lo stato della spedizione sia Attivo.

Per configurare la consegna dei log per uno spazio agente

1. Aprire la console AWS DevOps dell'agente nella console AWS di gestione.
2. Scegli lo spazio dell'agente che desideri configurare.
3. Nella scheda Configurazione, scegli Configura.
4. Per Tipo di destinazione, scegli una delle seguenti opzioni:
5. CloudWatch Registri: seleziona o crea un gruppo di registri.
6. Amazon S3: inserisci l'ARN del bucket S3.
7. Amazon Data Firehose: seleziona o crea un flusso di distribuzione Firehose.
8. Per le impostazioni aggiuntive, *opzionale*, puoi specificare le seguenti opzioni:

a. Per Selezione del campo, seleziona i nomi dei campi di log che desideri consegnare alla destinazione. È possibile selezionare i campi del registro degli accessi e un sottoinsieme di campi del registro degli accessi in tempo reale.

b. (Solo Amazon S3) Per Partizionamento, specifica il percorso per partizionare i dati del file di log.

c. (Solo Amazon S3) Per Formato file compatibile con Hive, puoi selezionare la casella di controllo per utilizzare percorsi S3 compatibili con Hive. Questo consente di semplificare il caricamento di nuovi dati negli strumenti compatibili con Hive.

d. Per Formato di output, specifica il formato preferito.

e. Per Delimitatore di campo, specifica come separare i campi di log.

9. Scegli Save (Salva).

10. Verifica che lo stato della spedizione sia Attivo.

Configura la consegna dei log (CloudWatch API)

Puoi anche utilizzare l'API CloudWatch Logs per configurare la consegna dei log a livello di codice. Una consegna di log funzionante è composta da tre elementi:

- A DeliverySource— Rappresenta la risorsa spaziale AWS DevOps dell'agente che genera i log.
- A DeliveryDestination— Rappresenta la destinazione in cui vengono scritti i log.
- Una consegna: collega un'origine di consegna a una destinazione di consegna.

Fase 1: Creare una fonte di consegna

Usa l'PutDeliverySourceoperazione per creare una fonte di consegna. Passa l'ARN della tua risorsa di spazio AWS DevOps Agent e specifica `APPLICATION_LOGS` come tipo di registro.

L'esempio seguente crea una fonte di consegna per uno spazio agente:

```
{
    "name": "my-agent-space-delivery-source",
    "resourceArn": "arn:aws:aidevops:us-east-1:123456789012:agentspace/my-agent-space-
id",
    "logType": "APPLICATION_LOGS"
}
```

L'esempio seguente crea una fonte di consegna per il servizio:

```
{
    "name": "my-service-delivery-source",
    "resourceArn": "arn:aws:aidevops:us-east-1:123456789012:service",
    "logType": "APPLICATION_LOGS"
}
```

Fase 2: Creare una destinazione di consegna

Usa l'PutDeliveryDestinationoperazione per configurare dove vengono archiviati i log. Puoi scegliere Amazon CloudWatch Logs, Amazon S3 o Amazon Data Firehose.

L'esempio seguente crea una CloudWatch destinazione Logs:

```
{
```

```
    "name": "my-cwl-destination",
    "deliveryDestinationConfiguration": {
        "destinationResourceArn": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/
aidevops/my-agent-space"
    },
    "outputFormat": "json"
}
```

L'esempio seguente crea una destinazione Amazon S3:

```
{
    "name": "my-s3-destination",
    "deliveryDestinationConfiguration": {
        "destinationResourceArn": "arn:aws:s3:::my-aidevops-logs-bucket"
    },
    "outputFormat": "json"
}
```

L'esempio seguente crea una destinazione Amazon Data Firehose:

```
{
    "name": "my-firehose-destination",
    "deliveryDestinationConfiguration": {
        "destinationResourceArn": "arn:aws:firehose:us-
east-1:123456789012:deliverystream/my-aidevops-log-stream"
    },
    "outputFormat": "json"
}
```

> **Note**
>
> Se spedisci i log su più account, devi utilizzarli PutDeliveryDestinationPolicynell'account di destinazione per autorizzare la consegna.

Se desideri utilizzare CloudFormation, puoi utilizzare quanto segue:

- Delivery
- DeliveryDestination
- DeliverySource

`ResourceArn` è `AgentSpaceArn` e `LogType` deve essere `APPLICATION_LOGS` come tipo di log supportato.

Fase 3: Creare una consegna

Utilizza l'CreateDeliveryoperazione per collegare l'origine di consegna alla destinazione di consegna.

```
{
    "deliverySourceName": "my-agent-space-delivery-source",
    "deliveryDestinationArn": "arn:aws:logs:us-east-1:123456789012:delivery-
destination:my-cwl-destination"
}
```

AWS CloudFormation

È inoltre possibile configurare la consegna dei log utilizzando AWS CloudFormation le seguenti risorse:

- AWS: :Registri:: DeliverySource
- AWS: :Registri:: DeliveryDestination
- AWS: :Logs: :Consegna

`ResourceArn`Impostato sullo spazio AWS DevOps agente o sull'ARN del servizio e impostato su`LogType`. `APPLICATION_LOGS`

Riferimento allo schema di log

AWS DevOps L'agente utilizza uno schema di registro condiviso per tutti i tipi di eventi. Non tutti gli eventi di registro utilizzano tutti i campi.

La tabella seguente descrive i campi dello schema di registro.

Campo	Tipo	Description
event_timestamp	Long	Timestamp Unix di quando si è verificato l'evento
resource_arn	Stringa	ARN della risorsa che ha generato l'evento

Campo	Tipo	Description
optional_account_id	Stringa	AWS ID dell'account associato al registro.
livello_opzionale	Stringa	Livello di registro:, `INFO` `WARN` `ERROR`
opzionale_agent_space_id	Stringa	Identificatore dello spazio dell'agente.
id_associazione_opzionale	Stringa	Identificatore di associazione per il registro.
opzionale_status	Stringa	Stato dell'operazione di topologia.
opzionale_webhook_id	Stringa	Identificatore Webhook.
opzionale_mcp_endpoint_url	Stringa	URL dell'endpoint del server MCP
tipo_di_servizio opzionale	Stringa	Tipo di servizio:,,,,. `DYNATRACE` `DATADOG` `GITHUB` `SLACK` `SERVICENOW`
opzionale_service_endpoint_url	Stringa	URL dell'endpoint per integrazioni di terze parti.
id_servizio_opzionale	Stringa	Identificatore della fonte.
request_id	Stringa	Richiedi l'identificatore per la correlazione AWS CloudTrail o i ticket di assistenza.
operazione_opzionale	Stringa	Nome dell'operazione che è stata eseguita.

Campo	Tipo	Description
optional_task_type	Stringa	Tipo di attività Agent Backlog: o `INVESTIGATION` `EVALUATION`
optional_task_id	Stringa	Identificatore dell'attività di backlog di Agent Backlog Task. IDAgent
referenza_opzionale	Stringa	Riferimento tratto da un'attività di agente (ad esempio, un ticket Jira).
tipo_errore_opzionale	Stringa	Tipi di errore
messaggio_errore_opzionale	Stringa	Descrizione dell'errore quando un'operazione fallisce.
optional_details	Stringa (JSON)	Payload di eventi specifico del servizio che contiene i parametri e i risultati delle operazioni.

Gestisci e disabilita la consegna dei log

È possibile modificare o rimuovere la consegna dei log in qualsiasi momento dalla console dell' AWS DevOps agente nella console di AWS gestione o utilizzando l'API CloudWatch Logs.

Gestisci la consegna dei log (console)

1. Apri la console AWS DevOps dell'agente nella console AWS di gestione.
2. Passare alla pagina Impostazioni (per i registri a livello di servizio) o alla pagina specifica di Agent Space (per i registri a livello di Agent Space).
3. Nella scheda Configurazione (per i log a livello di Agent Space) o nella scheda Capability Provider > Logs (per i log a livello di servizio), scegli la consegna da modificare.
4. Aggiorna la configurazione secondo necessità e scegli Salva.

Nota: non puoi modificare il tipo di destinazione di una consegna esistente. Per modificare il tipo di destinazione, elimina la consegna corrente e creane una nuova.

Disabilita la consegna dei log (console)

1. Apri la console AWS DevOps dell'agente nella console AWS di gestione.
2. Passare alla pagina Impostazioni (per i registri a livello di servizio) o alla pagina specifica di Agent Space (per i registri a livello di Agent Space).
3. Nella scheda Configurazione (per i log a livello di Agent Space) o nella scheda Capability Provider > Logs (per i log a livello di servizio), seleziona la consegna da rimuovere.
4. Scegli Elimina e conferma.

Disabilita la consegna dei log (API)

Per rimuovere una consegna di log utilizzando l'API, elimina le risorse nel seguente ordine:

1. Eliminare la consegna utilizzando DeleteDelivery.
2. Elimina la fonte di consegna utilizzando DeleteDeliverySource.
3. (Facoltativo) Se la destinazione di consegna non è più necessaria, eliminala utilizzando DeleteDeliveryDestination.

> **Important**
>
> L'utente è responsabile della rimozione delle risorse di consegna dei log dopo aver eliminato la risorsa dello spazio agente che genera i log (ad esempio, dopo aver eliminato uno spazio agente). Se non rimuovi queste risorse, le configurazioni di consegna potrebbero rimanere orfane.

Prezzi

L' AWS DevOps agente non addebita alcun costo per l'abilitazione dei registri venduti. Tuttavia, potrebbero essere addebitati costi per la consegna, l'acquisizione, l'archiviazione o l'accesso, a seconda della destinazione di consegna dei log selezionata. Per i dettagli sui prezzi, consulta Vented Logs nella scheda Logs di Amazon Pricing. CloudWatch

Per i prezzi specifici della destinazione, consulta quanto segue:

- Prezzi di Amazon CloudWatch Logs
- Prezzi di Amazon S3
- Prezzi di Amazon Data Firehose

Connessione a strumenti ospitati privatamente

Panoramica delle connessioni private

AWS DevOps L'agente può essere esteso con strumenti personalizzati Model Context Protocol (MCP) e altre integrazioni che consentono all'agente di accedere a sistemi interni come registri di pacchetti privati, piattaforme di osservabilità ospitate autonomamente, API di documentazione interna e istanze di controllo del codice sorgente (vedi:). Configurazione delle funzionalità per AWS DevOps Agente Questi servizi vengono spesso eseguiti all'interno di un Amazon Virtual Private Cloud (Amazon VPC) con accesso pubblico a Internet limitato o nullo, il che significa che l' AWS DevOps agente non può raggiungerli per impostazione predefinita.

Le connessioni private per AWS DevOps Agent ti consentono di connettere in modo sicuro il tuo Agent Space ai servizi in esecuzione nel tuo VPC senza esporli alla rete Internet pubblica. Le connessioni private funzionano con qualsiasi integrazione che necessiti di raggiungere un endpoint privato, inclusi server MCP, istanze Grafana o Splunk ospitate autonomamente e sistemi di controllo del codice sorgente come Enterprise Server e. GitHub GitLab Self-Managed

> **Note**
>
> Se i tuoi strumenti ospitati privatamente inviano richieste in uscita all' AWS DevOps agente dall'interno del tuo VPC, questo traffico può essere protetto anche utilizzando un endpoint VPC in modo che rimanga all'interno della rete. AWS Ad esempio, può essere utilizzato con strumenti che attivano l' DevOps agente tramite eventi webhook (vedi:). the section called "Richiamo DevOps dell'agente tramite Webhook" Per ulteriori informazioni, consulta the section called "Endpoint VPC (AWS PrivateLink)".

Come funzionano le connessioni private

Una connessione privata crea un percorso di rete sicuro tra AWS DevOps l'agente e una risorsa di destinazione nel tuo VPC. Sotto il cofano, AWS DevOps Agent utilizza Amazon VPC Lattice per stabilire questo percorso di connettività privata sicuro. VPC Lattice è un servizio di rete di applicazioni

che consente di connettere, proteggere e monitorare la comunicazione tra le applicazioni tra VPC, account e tipi di elaborazione, senza gestire l'infrastruttura di rete sottostante.

Quando si crea una connessione privata, si verifica quanto segue:

- Fornisci il VPC, le sottoreti e (facoltativamente) i gruppi di sicurezza che dispongono di connettività di rete al servizio di destinazione.
- AWS DevOps L'agente crea un gateway di risorse gestito dai servizi e fornisce le sue interfacce di rete elastiche (ENI) nelle sottoreti specificate.
- L'agente utilizza il resource gateway per indirizzare il traffico verso l'indirizzo IP o il nome DNS del servizio di destinazione tramite il percorso di rete privata.

Il gateway di risorse è completamente gestito dall' AWS DevOps agente e viene visualizzato come risorsa di sola lettura nell'account (denominato). `aidevops-{your-private-connection-name}` Non è necessario configurarlo o gestirlo. Le uniche risorse create nel tuo VPC sono gli ENI nelle sottoreti che specifichi. Questi ENI fungono da punto di ingresso per il traffico privato e sono gestiti interamente dal servizio. Non accettano connessioni in entrata da Internet e tu mantieni il pieno controllo sul loro traffico tramite i tuoi gruppi di sicurezza.

Sicurezza

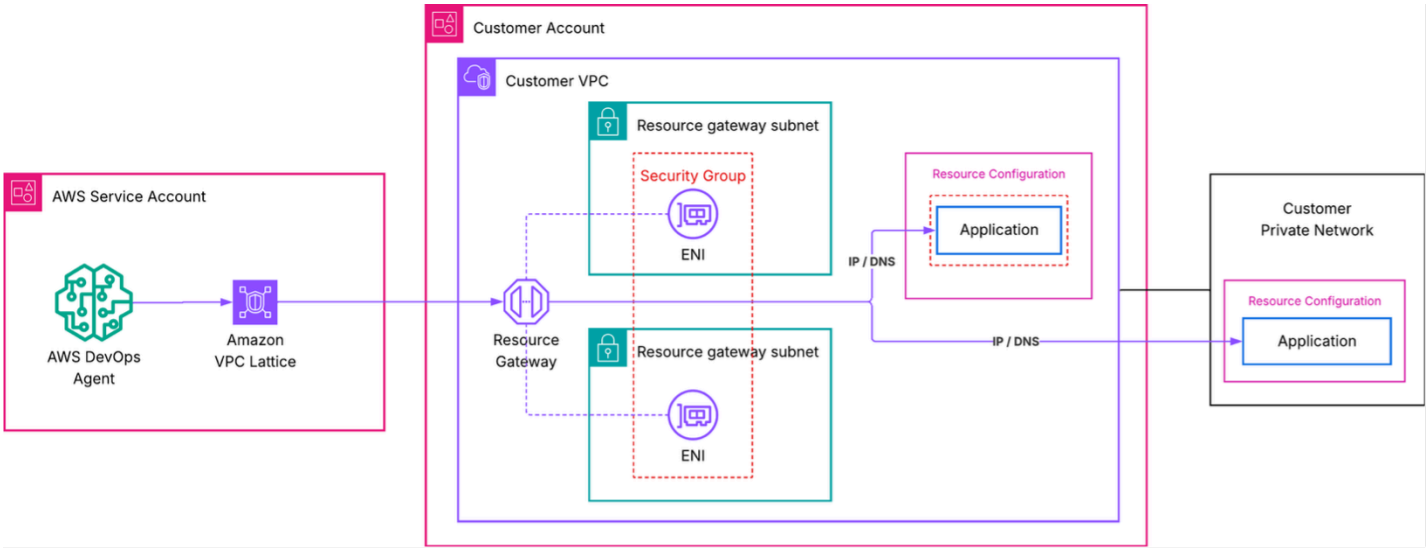
Le connessioni private sono progettate con più livelli di sicurezza:

- Nessuna esposizione pubblica a Internet: tutto il traffico tra AWS DevOps Agent e il servizio di destinazione rimane sulla AWS rete. Il tuo servizio non ha mai bisogno di un indirizzo IP pubblico o di un gateway Internet.
- Service-controlled gateway di risorse: il gateway di risorse gestito dal servizio è di sola lettura nel tuo account. Può essere utilizzato solo dall' AWS DevOps agente e nessun altro servizio o principale può instradare il traffico attraverso di esso. Puoi verificarlo nei AWS CloudTraillog, che registrano tutte le chiamate API VPC Lattice.
- I tuoi gruppi di sicurezza, le tue regole: puoi controllare il traffico in entrata e in uscita verso gli ENI tramite gruppi di sicurezza di tua proprietà e gestione. Se non specifichi gruppi di sicurezza, AWS DevOps Agent crea un gruppo di sicurezza predefinito con ambito alle porte che definisci.
- Service-linked ruoli con privilegi minimi: AWS DevOps l'agente utilizza un ruolo collegato al servizio per creare solo le risorse VPC Lattice e Amazon EC2 necessarie. Questo ruolo è limitato alle risorse contrassegnate `AWSAIDevOpsManaged` e non può accedere ad altre risorse del tuo account.

> **Note**
>
> Se la tua organizzazione dispone di policy di controllo dei servizi (SCP) che limitano le azioni dell'API VPC Lattice, il gateway di risorse gestito dai servizi viene creato tramite un ruolo collegato al servizio. Assicurati che i tuoi SCP consentano le azioni necessarie per il ruolo collegato al servizio.

Architecture

Il diagramma seguente mostra il percorso di rete per una connessione privata.



In questa architettura:

- AWS DevOps L'agente avvia una richiesta al servizio di destinazione.
- Amazon VPC Lattice indirizza la richiesta attraverso il gateway di risorse gestite dai servizi nel tuo VPC. Per configurazioni avanzate che utilizzano le tue risorse VPC Lattice, consulta Configurazione avanzata utilizzando le risorse VPC Lattice esistenti.
- Un ENI nel tuo VPC riceve il traffico e lo inoltra all'indirizzo IP o al nome DNS del servizio di destinazione.
- I tuoi gruppi di sicurezza regolano il traffico consentito attraverso gli ENI.
- Dal punto di vista del servizio di destinazione, la richiesta proviene da indirizzi IP privati di ENI all'interno del tuo VPC.

Crea una connessione privata

È possibile creare una connessione privata utilizzando la console di AWS gestione o la AWS CLI.

> **Note**
>
> Le seguenti zone di disponibilità non sono supportate da VPC Lattice:`use1-az3`,,,`usw1-az2`,`apne1-az3`,, `apne2-az2euc1-az2`,`euw1-az4`. `cac1-az3` `ilc1-az2`

Prerequisiti

Prima di creare una connessione privata, verifica di disporre di quanto segue:

- Uno spazio agente attivo: è necessario disporre di uno spazio agente esistente nel proprio account. Se non lo hai, consultare Guida introduttiva a AWS DevOps Agent.
- Un servizio di destinazione raggiungibile privatamente: il server MCP, la piattaforma di osservabilità o un altro servizio devono essere raggiungibili con un indirizzo IP privato o un nome DNS noto dal VPC in cui è distribuito il gateway di risorse. Il servizio può essere eseguito nello stesso VPC, in un VPC peer o in locale, purché sia instradabile dalle sottoreti del Resource Gateway. Il servizio deve servire il traffico HTTPS con una versione TLS minima di 1.2 su una porta specificata al momento della creazione della connessione.
- Sottoreti nel tuo VPC: identifica da 1 a 20 sottoreti in cui verranno create le ENI. Ti consigliamo di selezionare sottoreti in più zone di disponibilità per un'elevata disponibilità. Queste sottoreti devono disporre di connettività di rete al servizio di destinazione. Una sottorete per zona di disponibilità può essere utilizzata da VPC Lattice.
- (Facoltativo) Gruppi di sicurezza: se desideri controllare il traffico con regole specifiche, prepara fino a cinque ID di gruppi di sicurezza da collegare agli ENI. Se ometti i gruppi di sicurezza, AWS DevOps Agent crea un gruppo di sicurezza predefinito.

Le connessioni private sono risorse a livello di account. Dopo aver creato una connessione privata, puoi riutilizzarla su più integrazioni e Agent Spaces che devono raggiungere lo stesso host.

Crea una connessione privata utilizzando la console

1. Apri la console AWS DevOps dell'agente.
2. Nel riquadro di navigazione, scegli Provider di capacità, quindi scegli Connessioni private.

3. Scegli Crea una nuova connessione.
4. In Nome, inserisci un nome descrittivo per la connessione, ad esempio`my-mcp-tool-connection`.
5. Per VPC, seleziona il VPC in cui verrà distribuito il Resource Gateway ENI.
6. Per Subnet, seleziona una o più sottoreti (fino a 20). Consigliamo di scegliere sottoreti in almeno due zone di disponibilità.
7. Per il tipo di indirizzo IP, seleziona il tipo di indirizzo IP del servizio di destinazione (`IPv4`,`IPv6`, o`DualStack`).
8. (Facoltativo) Per Numero di indirizzi IPv4, se hai selezionato IPv4 o Dualstack per il tipo di indirizzo IP, puoi inserire il numero di indirizzi IPv4 per ENI per il tuo gateway di risorse. L'impostazione predefinita è 16 indirizzi IPv4 per ENI.
9. (Facoltativo) Per i gruppi di sicurezza, seleziona i gruppi di sicurezza esistenti (fino a 5) per limitare il traffico consentito per raggiungere il servizio di destinazione. Se non ne selezioni nessuno, viene creato un gruppo di sicurezza predefinito.
10. (Facoltativo) Per gli intervalli di porte, specificate le porte TCP su cui l'applicazione di destinazione ascolta (ad esempio `443` o`8080-8090`). È possibile specificare fino a 11 intervalli di porte.
11. Per Indirizzo host, inserisci l'indirizzo IP o il nome DNS del servizio di destinazione (ad esempio, `mcp.internal.example.com` o`10.0.1.50`). Il servizio deve essere raggiungibile dal VPC selezionato. Se scegli un nome DNS, deve essere risolvibile pubblicamente.
12. (Facoltativo) Per la chiave pubblica del certificato, se l'indirizzo host specificato utilizza certificati TLS emessi da un'autorità di certificazione privata, inserisci la chiave PEM-encoded pubblica del certificato. Ciò consente all' AWS DevOps agente di affidare la connessione TLS al servizio di destinazione.
13. Scegli Crea connessione.

Lo stato della connessione cambia in Creazione in corso. Questo processo può richiedere fino a 10 minuti. Quando lo stato diventa Attivo, il percorso di rete è pronto.

Se la modifica dello stato in Create non è riuscita, verifica quanto segue:

- Le sottoreti specificate hanno indirizzi IP disponibili.
- Il tuo account non ha raggiunto le quote del servizio VPC Lattice.
- Nessuna policy IAM restrittiva impedisce al ruolo collegato ai servizi di creare risorse.

> **Note**
>
> Questi passaggi possono essere eseguiti anche selezionando `Create a new private connection` durante la registrazione di un provider di capacità. Per ulteriori informazioni, consulta Utilizzare una connessione privata con un provider di funzionalità.

Creare una connessione privata utilizzando il AWS CLI

Esegui il comando seguente per creare una connessione privata. Sostituisci i valori segnaposto con i tuoi.

```
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"]
        }
    }'
```

La risposta include il nome della connessione e lo stato di: `CREATE_IN_PROGRESS`

```
{
    "name": "my-mcp-tool-connection",
    "status": "CREATE_IN_PROGRESS",
    "resourceGatewayId": "rgw-0123456789abcdef0",
    "hostAddress": "mcp.internal.example.com",
    "vpcId": "vpc-0123456789abcdef0"
}
```

Per verificare lo stato della connessione, utilizzare il `describe-private-connection` comando:

```
aws devops-agent describe-private-connection \
    --name my-mcp-tool-connection
```

Quando lo stato è`ACTIVE`, la tua connessione privata è pronta per l'uso.

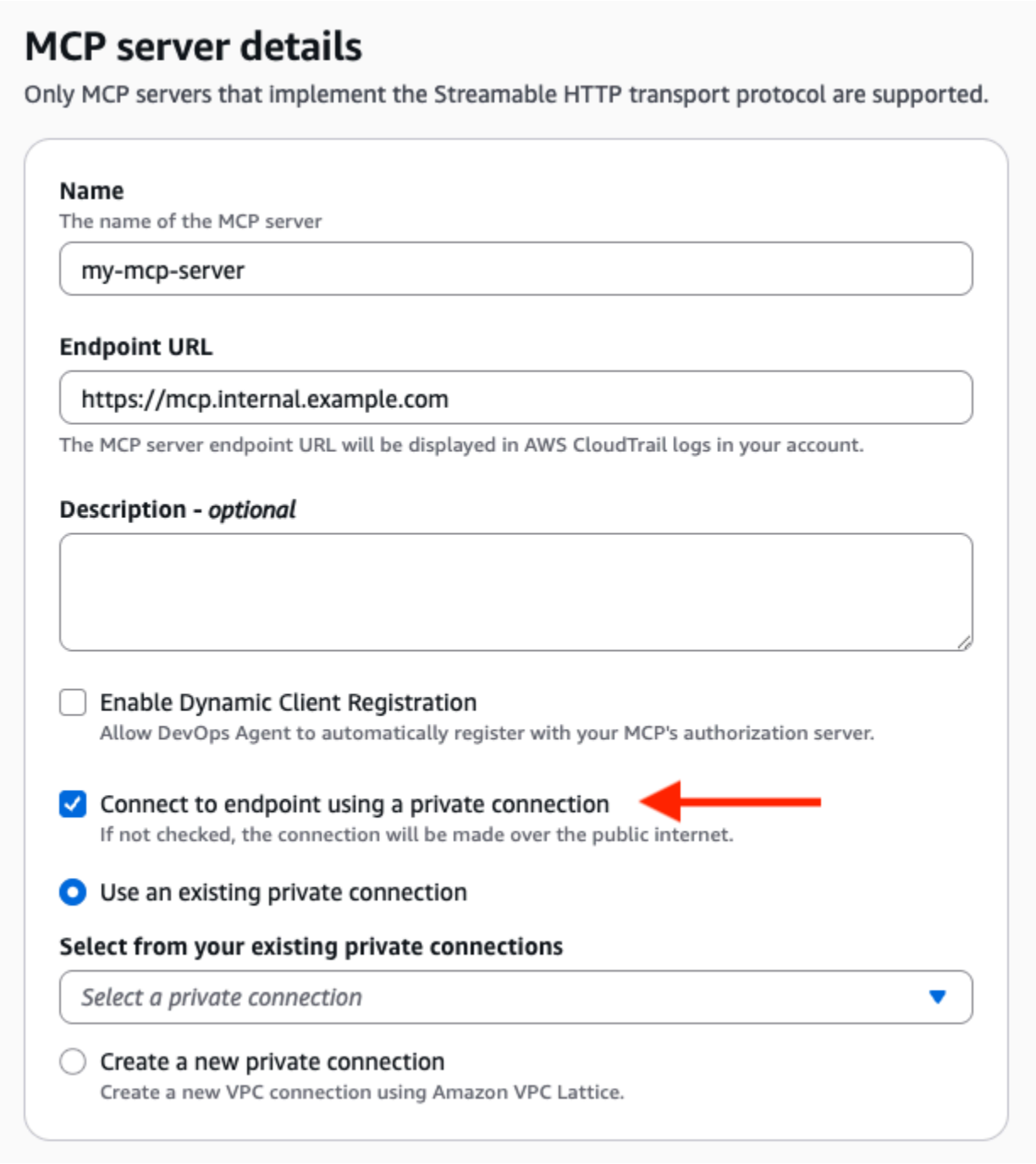
Utilizza una connessione privata con un provider di funzionalità

Per utilizzare una connessione privata, è possibile collegarsi ad essa durante la registrazione di un provider di capacità. Le funzionalità supportate che possono essere utilizzate con connessioni private includono: `GitHubGitLab`,`MCP Server`, e`Grafana`. È possibile eseguire questo passaggio utilizzando la console di AWS gestione o la AWS CLI.

> **Note**
>
> Al momento della registrazione di un provider di funzionalità, AWS DevOps Agent verifica che l'endpoint sia raggiungibile e risponda. Assicurati che il servizio di destinazione sia in esecuzione e accetti le connessioni prima di completare la registrazione.

Utilizza una connessione privata con un provider di funzionalità tramite la console

Nella console dell' AWS DevOps agente, le connessioni private possono essere collegate a una funzionalità durante la registrazione selezionando l'opzione «Connetti all'endpoint utilizzando una connessione privata».



1. Apri la console dell' AWS DevOps agente e accedi al tuo Agent Space.
2. Nella sezione Provider di capacità, scegli Registrazione.
3. Seleziona Registra per il tipo di funzionalità che desideri utilizzare con la connessione privata.
4. Nella visualizzazione dei dettagli della registrazione, inserisci l'URL dell'endpoint a cui desideri connetterti utilizzando la connessione privata (ad esempio,`https://mcp.internal.example.com`).

5. Seleziona Connetti all'endpoint usando una connessione privata.
6. Seleziona una connessione privata esistente che corrisponde all'URL dell'endpoint a cui desideri connetterti oppure seleziona Crea una nuova connessione privata per crearne una.
7. Completa il processo di registrazione per il provider di funzionalità.

> **Note**
>
> Quando si seleziona una connessione privata per un provider di funzionalità che utilizza l'autenticazione OAuth (credenziali client o 3LO), la connessione privata si applica sia all'endpoint del provider di capacità che all'endpoint di scambio di token. Assicurati che la connessione privata sia configurata con un indirizzo host in grado di indirizzare il traffico verso entrambi gli endpoint.

Utilizza una connessione privata con un provider di funzionalità utilizzando il AWS CLI

È possibile registrare le funzionalità con una connessione privata includendo l'`private-connection-name`argomento. Di seguito è riportato un esempio di registrazione di un server MCP con autorizzazione API Key utilizzando la connessione `my-mcp-tool-connection` privata. Sostituite i valori segnaposto con i vostri.

```
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
```

Verifica una connessione privata

Dopo che la connessione privata ha raggiunto lo stato Attivo ed è stata utilizzata da un provider di funzionalità, verifica che AWS DevOps Agent sia in grado di raggiungere il servizio di destinazione:

1. Apri la console dell' AWS DevOps agente e accedi al tuo Agent Space.
2. Inizia una nuova sessione di chat.
3. Invoca un comando che utilizza l'integrazione supportata dalla tua connessione privata. Ad esempio, se lo strumento MCP fornisce l'accesso a una knowledge base interna, ponete all'agente una domanda che richieda tale base di conoscenza.
4. Verificate che l'agente restituisca i risultati del servizio privato.

Se la connessione fallisce, controlla quanto segue:

- Limiti di VPC Lattice: verifica di non aver raggiunto alcun gateway di risorse o altri limiti di quota VPC Lattice
- Regole dei gruppi di sicurezza: verifica che i gruppi di sicurezza collegati agli ENI consentano il traffico in uscita sulla porta su cui il servizio è in ascolto. Verifica inoltre che il gruppo di sicurezza del servizio consenta il traffico in entrata sulla porta di destinazione. Il traffico proviene dagli IP del piano dati VPC Lattice all'interno dell'intervallo VPC CIDR. È possibile utilizzare il riferimento al gruppo di sicurezza (che consente il gruppo di sicurezza ENI come fonte) o consentire l'ingresso dal VPC CIDR.
- Connettività alla sottorete: verifica che le sottoreti selezionate siano in grado di indirizzare il traffico verso il servizio. Se il servizio viene eseguito in una sottorete diversa, verifica che le tabelle di routing consentano il traffico tra di esse.
- Disponibilità del servizio: verifica che il servizio sia in esecuzione e accetti connessioni sulla porta prevista.
- Zona di disponibilità non supportata: verifica che le sottoreti si trovino nelle zone di disponibilità supportate. Esegui `aws ec2 describe-subnets --subnet-ids <your-subnet-ids> --query 'Subnets[*].[SubnetId,AvailabilityZoneId]'` e verifica le zone di disponibilità non supportate elencate sopra.

Eliminare una connessione privata

È possibile eliminare le connessioni private non utilizzate utilizzando la console di AWS gestione o la AWS CLI.

Eliminare una connessione privata utilizzando la console

1. Apri la console AWS DevOps dell'agente.
2. Nel riquadro di navigazione, scegli Provider di capacità, quindi scegli Connessioni private.
3. Seleziona il menu Azioni per la connessione privata che desideri eliminare e seleziona Rimuovi.

La connessione privata verrà visualizzata con lo stato «Rimozione della connessione» mentre l' AWS DevOps agente rimuove il gateway di risorse gestite e gli ENI dal tuo VPC. Una volta completata l'eliminazione, la connessione non viene più visualizzata nell'elenco delle connessioni private.

Eliminare una connessione privata utilizzando il AWS CLI

```
aws devops-agent delete-private-connection \
    --name my-mcp-tool-connection
```

La risposta restituisce uno stato di`DELETE_IN_PROGRESS`. AWS DevOps L'agente rimuove il gateway di risorse gestite e gli ENI dal tuo VPC. Una volta completata l'eliminazione, la connessione non viene più visualizzata nell'elenco delle connessioni private.

Configurazione avanzata utilizzando le risorse VPC Lattice esistenti

Se la tua organizzazione utilizza già Amazon VPC Lattice e gestisce le configurazioni delle risorse, puoi creare una connessione privata in modalità autogestita. Invece di fare in modo che AWS DevOps Agent crei un gateway di risorse per te, fornisci l'Amazon Resource Name (ARN) di una configurazione di risorse esistente che punta al servizio di destinazione.

Questo approccio è utile quando:

- Desideri il pieno controllo sul Resource Gateway e sul ciclo di vita della configurazione delle risorse.
- Hai bisogno di condividere le configurazioni delle risorse tra più AWS account o servizi.
- Richiedi i log di accesso VPC Lattice per un monitoraggio dettagliato del traffico.

- Esegui un'architettura di rete hub-and-spoke.

Per creare una connessione privata autogestita con la 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"
        }
    }'
```

Per ulteriori dettagli sulla configurazione dei gateway di risorse VPC Lattice e sulle configurazioni delle risorse, consulta la Amazon VPC Lattice User Guide.

Argomenti correlati

- the section called "Endpoint VPC (AWS PrivateLink)"
- the section called "Connessione dei server MCP"
- Configurazione delle funzionalità per AWS DevOps Agente
- AWS DevOps Sicurezza degli agenti
- the section called "DevOps Autorizzazioni Agent IAM"

AWS DevOps Sicurezza degli agenti

Questo documento fornisce informazioni su considerazioni relative alla sicurezza, alla protezione dei dati, ai controlli degli accessi e alle funzionalità di conformità per AWS DevOps Agent. Utilizzate queste informazioni per comprendere in che modo AWS DevOps Agent è progettato per soddisfare i requisiti di sicurezza e conformità.

Multi-layered sicurezza

AWS DevOps L'agente implementa la sicurezza a più livelli. Anche se vengono concesse autorizzazioni più ampie al ruolo IAM dell'agente, l'agente applica i propri controlli di accesso interni per limitare l'ambito delle sue azioni.

Consigliamo di seguire il principio del privilegio minimo durante la configurazione delle autorizzazioni IAM per l' AWS DevOps agente e l'implementazione della sicurezza a più livelli. Una difesa approfondita garantisce che nessuna singola configurazione errata possa compromettere la sicurezza dell'ambiente.

Agent Spaces

Gli Agent Spaces fungono da limite di sicurezza principale in AWS DevOps Agent. Ogni spazio per agenti:

- Funziona in modo indipendente con le proprie configurazioni e autorizzazioni
- Definisce a quali AWS account e risorse può accedere l'agente
- Stabilisce connessioni a piattaforme di terze parti

Agent Spaces mantiene un rigoroso isolamento per garantire la sicurezza e prevenire accessi involontari tra diversi ambienti o team.

Elaborazione e flusso di dati a livello regionale

AWS DevOps L'agente opera a livello globale con funzionalità di elaborazione regionali. L'agente recupera i dati operativi dalle AWS regioni di tutti gli AWS account a cui è concesso l'accesso all'interno dell'Agent Space configurato. Questa raccolta di dati su più account in più regioni

garantisce un'analisi completa degli incidenti rispettando i confini geografici per l'elaborazione delle inferenze.

Utilizzo di Amazon Bedrock e inferenza tra regioni

AWS DevOps L'agente selezionerà automaticamente la regione ottimale all'interno della tua area geografica per elaborare le tue richieste di inferenza. Ciò ottimizza le risorse di elaborazione disponibili, la disponibilità dei modelli e offre la migliore esperienza al cliente. I dati rimarranno archiviati solo nella regione in cui viene creato Agent Space, tuttavia, le richieste di input e i risultati di output potrebbero essere elaborati al di fuori di tale regione, come descritto nell'elenco seguente. Tutti i dati verranno trasmessi crittografati attraverso la rete sicura di Amazon.

AWS DevOps L'agente indirizzerà in modo sicuro le richieste di inferenza alle risorse di calcolo disponibili all'interno dell'area geografica in cui ha avuto origine la richiesta, come segue:

- Le richieste di inferenza provenienti dall'Unione europea verranno elaborate all'interno dell'Unione europea.
- Le richieste di inferenza provenienti dagli Stati Uniti verranno elaborate all'interno degli Stati Uniti.
- Le richieste di inferenza provenienti dall'Australia verranno elaborate all'interno dell'Australia.
- Le richieste di inferenza provenienti dal Giappone verranno elaborate all'interno del Giappone.
- Se una richiesta di inferenza proviene da un'area non elencata, verrà elaborata per impostazione predefinita negli Stati Uniti d'America.
- DevOps Agent e Bedrock non sono influenzati dalle politiche dei clienti nelle Service Control Policies (SCP) o Control Tower che limitano i contenuti dei clienti a regioni specifiche
- Bedrock può utilizzare regioni diverse dalla regione di origine all'interno della vostra area geografica per eseguire inferenze senza stato e ottimizzare prestazioni e disponibilità

Gestione dell'identità e degli accessi

Metodi di autenticazione

AWS DevOps Agent fornisce due metodi di autenticazione per accedere all'app web Agent Space: AWS DevOps

- AWS Integrazione con Identity Center: il metodo di autenticazione principale utilizza OAuth 2.0 con autenticazione basata sulla sessione tramite cookie. HTTP-only AWS Identity Center può

federarsi con provider di identità esterni tramite protocolli OIDC e SAML standard, inclusi provider come Okta, Ping Identity e Microsoft Entra ID. Questo metodo supporta l'autenticazione a più fattori tramite il tuo provider di identità. AWS L'impostazione predefinita di Identity Center prevede una durata delle sessioni fino a 12 ore e può essere configurato sulla durata desiderata.

- Link di autenticazione IAM: un metodo alternativo fornisce l'accesso diretto all'app Web dalla console di AWS gestione utilizzando JWT-based token derivati da una sessione della console di gestione esistente AWS . Questa opzione è utile per valutare l' AWS DevOps agente prima di implementare l'integrazione completa di Identity Center e per ottenere l'accesso amministrativo se l'app Web dell' AWS DevOps agente diventa inaccessibile tramite l'autenticazione basata su Identity Center. Le sessioni sono limitate a 10 minuti.

Ruoli IAM

AWS DevOps L'agente utilizza i ruoli IAM per definire le autorizzazioni di accesso:

- Ruolo dell'account principale: concede all'agente l'accesso alle risorse dell' AWS account in cui viene creato l'Agent Space, nonché l'accesso ai ruoli dell'account secondario.
- Ruoli dell'account secondario: concede all'agente l'accesso alle risorse in AWS account aggiuntivi collegati all'Agent Space.
- Ruolo dell'app Web: consente agli utenti di accedere ai dati e ai risultati delle indagini dell' AWS DevOps agente nell'app Web.

Questi ruoli devono essere configurati secondo il principio del privilegio minimo, che concede solo le autorizzazioni di sola lettura necessarie per le indagini.

Protezione dei dati

Crittografia dei dati

AWS DevOps L'agente crittografa tutti i dati dei clienti:

- Crittografia a riposo: tutti i dati sono crittografati con chiavi AWS gestite.
- Crittografia in transito: tutti i log, le metriche, le informazioni, i metadati dei ticket e altri dati recuperati vengono crittografati in transito all'interno della rete privata dell'agente e verso reti esterne.

Archiviazione e conservazione dei dati

I dati vengono archiviati nella regione in cui viene creato il tuo Agent Space, mentre l'elaborazione delle inferenze può avvenire all'interno della tua area geografica, come descritto nella precedente sezione sull'utilizzo di Amazon Bedrock.

Informazioni personali identificabili (PII)

AWS DevOps L'agente non filtra le informazioni PII quando riepiloga i dati raccolti durante le indagini, le valutazioni dei consigli o le risposte alle chat. Si consiglia di oscurare i dati PII prima di archiviarli nei registri di osservabilità.

Registrazione del diario e degli audit degli agenti

Diario dell'agente

Entrambe le funzionalità di indagine e prevenzione degli incidenti mantengono diari dettagliati che:

- Registra ogni fase di ragionamento e le azioni intraprese
- Crea una trasparenza completa nei processi decisionali degli agenti
- Non può essere modificato dagli agenti una volta registrato, in modo da ridurre al minimo gli attacchi, come l'iniezione tempestiva, che nascondono azioni importanti
- Includi tutti i messaggi di chat dalla pagina Indagine

AWS CloudTrail integrazione

Tutte le chiamate AWS DevOps Agent API vengono acquisite automaticamente AWS CloudTrail dall' AWS account di hosting. Utilizzando le informazioni raccolte da CloudTrail, è possibile determinare:

- La richiesta che è stata fatta all'agente
- L'indirizzo IP dal quale è stata effettuata la richiesta
- L'utente che ha effettuato la richiesta
- Quando è stata effettuata

Protezione tempestiva per l'iniezione

Un attacco di pronta iniezione si verifica quando un aggressore incorpora istruzioni dannose in dati esterni, come una pagina Web o un documento, che un sistema di intelligenza artificiale generativa elaborerà successivamente. AWS DevOps L'agente utilizza nativamente molte fonti di dati nell'ambito delle sue normali operazioni, inclusi log, tag di risorse e altri dati operativi. AWS DevOps Agent protegge dagli attacchi di pronta iniezione attraverso le seguenti misure di sicurezza, ma è importante garantire che tutte le fonti di dati connesse e l'accesso degli utenti a tali fonti di dati siano affidabili. Per ulteriori informazioni, consulta la sezione Modello di responsabilità condivisa.

Protezioni per l'iniezione rapida:

- Capacità di scrittura limitate: gli strumenti a disposizione dell'agente non sono in grado di modificare le risorse, ad eccezione dell'apertura di ticket e richieste di assistenza. In questo modo si evita che istruzioni dannose modifichino l'infrastruttura o le applicazioni.
- Applicazione dei limiti dell'account: AWS DevOps l'agente opera solo entro i limiti consentiti dai ruoli assegnati all'agente negli account secondari primari e collegati. AWS L'agente non può accedere o modificare risorse al di fuori dell'ambito configurato.
- Protezioni di sicurezza AI: AWS DevOps l'agente utilizza modelli con protezioni AI Safety Level 3 (ASL-3). Queste protezioni includono classificatori che rilevano e prevengono gli attacchi di pronta iniezione prima che possano influire sul comportamento degli agenti.
- Audit trail immutabile: il diario dell'agente registra ogni fase di ragionamento e ogni azione intrapresa. Le voci del diario non possono essere modificate dall'agente una volta registrate, per evitare che gli attacchi di prompt injection nascondano azioni dannose.

Sebbene AWS DevOps Agent fornisca più livelli di protezione contro gli attacchi di prompt injection, alcune configurazioni possono aumentare il rischio:

- Strumenti server MCP personalizzati: la funzionalità MCP personalizzata consente di introdurre strumenti personalizzati per l'agente, che possono offrire ulteriori opportunità di iniezione tempestiva. Gli strumenti personalizzati potrebbero non avere gli stessi controlli di sicurezza degli strumenti nativi di AWS DevOps Agent e istruzioni dannose potrebbero potenzialmente sfruttare questi strumenti in modi non intenzionali. Per ulteriori informazioni, consulta la sezione Modello di responsabilità condivisa.
- Attacchi utente autorizzati: gli utenti autorizzati a operare entro i limiti dell' AWS account o degli strumenti connessi hanno maggiori probabilità di tentare un attacco contro l'agente. Questi utenti

possono avere la possibilità di modificare le fonti di dati utilizzate dall'agente, come i log o i tag delle risorse, facilitando l'incorporazione di istruzioni dannose che l'agente elaborerà.

Per mitigare questi rischi:

1. Esamina e testa attentamente i server MCP personalizzati prima di distribuirli in Agent Spaces.
   a. Assicurati che siano autorizzati a eseguire solo azioni di sola lettura
   b. Verificate che gli utenti degli strumenti esterni a cui accedono i server MCP siano entità attendibili, poiché AWS DevOps gli agenti che si interfacciano con MCP si basano sulla relazione di fiducia implicita stabilita tra questi utenti dello strumento e l'agente AWS DevOps
2. Applica il principio del privilegio minimo quando concedi agli utenti l'accesso ai sistemi che forniscono dati all'agente
3. Controlla regolarmente quali server MCP sono collegati ai tuoi Agent Spaces
4. Poiché qualsiasi contenuto recuperato dagli URL consentiti potrebbe tentare di manipolare il comportamento dell'agente, includi solo fonti attendibili nella tua lista degli indirizzi consentiti.

Sicurezza dell'integrazione

AWS DevOps Agent supporta diversi tipi di integrazione, ognuno con il proprio modello di sicurezza:

- Integrazioni bidirezionali native: Built-in integrazioni in grado di inviare dati all'agente e ricevere aggiornamenti dall'agente. Questo utilizza i metodi di autenticazione del fornitore
- Server MCP: server Remote Model Context Protocol che utilizzano flussi di autenticazione OAuth 2.0 e chiavi API per comunicare in modo sicuro con sistemi esterni.
- Trigger Webhook: trigger di indagine provenienti da servizi remoti come ticket o sistemi di osservabilità. I webhook utilizzano il Hash-based Message Authentication Code (HMAC) per motivi di sicurezza.
- Comunicazione in uscita: integrazioni come Slack e i sistemi di ticketing ricevono aggiornamenti dall'agente ma non supportano ancora la comunicazione bidirezionale.

Fornitori di registrazione

Alcuni strumenti esterni sono autenticati a livello di account e condivisi tra tutti gli Agent Spaces dell'account. Quando si registrano questi strumenti, ci si autentica una volta a livello di account,

quindi ogni Agent Space può connettersi a risorse specifiche all'interno di quella connessione registrata.

I seguenti strumenti utilizzano la registrazione a livello di account:

- GitHub— Utilizza il flusso OAuth per l'autenticazione. Dopo la registrazione GitHub a livello di account, ogni Agent Space può connettersi a repository specifici all'interno dell'organizzazione. GitHub
- Dynatrace: utilizza l'autenticazione tramite token OAuth. Dopo aver registrato Dynatrace a livello di account, ogni Agent Space può connettersi a specifici ambienti Dynatrace o configurazioni di monitoraggio.
- Slack: utilizza l'autenticazione tramite token OAuth. Dopo aver registrato Slack a livello di account, ogni Agent Space può connettersi a canali Slack specifici.
- Datadog: utilizza MCP con flusso OAuth per l'autenticazione. Dopo aver registrato Datadog a livello di account, ogni Agent Space può connettersi a risorse di monitoraggio Datadog specifiche.
- New Relic: utilizza l'autenticazione tramite chiave API. Dopo aver registrato New Relic a livello di account, ogni Agent Space può connettersi a specifiche configurazioni di monitoraggio New Relic.
- Splunk: utilizza l'autenticazione con token bearer. Dopo aver registrato Splunk a livello di account, ogni Agent Space può connettersi a fonti di dati Splunk specifiche.
- GitLab— Utilizza l'autenticazione tramite token di accesso. Dopo la registrazione GitLab a livello di account, ogni Agent Space può connettersi a GitLab repository specifici.
- ServiceNow— Utilizza l'autenticazione del client OAuth. key/token Dopo la registrazione ServiceNow a livello di account, ogni Agent Space può connettersi a ServiceNow istanze o code di ticket specifiche.
- Server MCP remoti accessibili al pubblico generico: utilizza il flusso OAuth per l'autenticazione. Dopo aver registrato un server MCP remoto a livello di account, ogni Agent Space può connettersi a risorse specifiche esposte da quel server.

La connettività di rete

AWS DevOps L'agente si connette ai sistemi di terze parti e ai server MCP remoti per eseguire indagini e altre operazioni.

Traffico in entrata da AWS DevOps Agente per i tuoi sistemi

AWS DevOps L'agente avvia connessioni in uscita verso i sistemi di terze parti e i server MCP remoti, che arrivano come traffico in entrata all'infrastruttura. Il modo in cui proteggi questo traffico dipende da come sono ospitati i tuoi strumenti:

- Strumenti ospitati privatamente: se i tuoi strumenti sono raggiungibili dall'interno di un AWS VPC, puoi utilizzare le connessioni private degli AWS DevOps agenti per mantenere il traffico isolato dalle AWS reti e lontano dalla rete Internet pubblica. Per ulteriori informazioni, consulta the section called "Connessione a strumenti ospitati privatamente".
- Strumenti ospitati pubblicamente: se gli strumenti sono raggiungibili sulla rete Internet pubblica e utilizzano l'elenco degli indirizzi IP consentiti o le regole del firewall, è necessario consentire il traffico in entrata dai seguenti indirizzi IP di origine dell'agente: AWS DevOps
  - Asia Pacifico (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`
  - Asia Pacifico (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 (Francoforte) (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 (Irlanda) (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
- Stati Uniti orientali (Virginia settentrionale) (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
- Stati Uniti occidentali (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

Traffico in uscita dal tuo VPC a AWS DevOps Agente

Per il traffico in uscita dal tuo AWS VPC AWS DevOps all'agente (ad esempio, the section called "Richiamo DevOps dell'agente tramite Webhook" utilizzando), puoi utilizzare gli endpoint VPC per mantenere questo traffico di rete isolato dalle reti. AWS Per ulteriori informazioni, consulta the section called "Endpoint VPC (AWS PrivateLink)".

Modello di responsabilità condivisa

AWS responsabilità

AWS è responsabile di:

- Mantenimento della sicurezza dei dati recuperati dall'agente
- Protezione degli strumenti nativi disponibili per l'uso da parte dell'agente
- Protezione dell'infrastruttura su cui è in esecuzione Agent AWS DevOps

Responsabilità del cliente

I clienti sono responsabili di:

- Gestione dell'accesso degli utenti allo spazio degli agenti
- Limitazione dell'accesso a utenti affidabili di sistemi esterni che forniscono input all'agente, ad esempio servizi e risorse che producono registri, CloudTrail eventi, ticket e altro, che possono essere utilizzati per tentare l'iniezione tempestiva di informazioni dannose.
- Assicurati che tutte le fonti di dati connesse dispongano di dati affidabili che è improbabile che vengano utilizzati per tentare attacchi di pronta iniezione
- Garantire che le integrazioni dei server MCP personalizzate funzionino in modo sicuro
- Garantire che i ruoli IAM assegnati all'agente abbiano un ambito adeguato
- Oscurare i dati PII prima di archiviarli nei registri di osservabilità e in altre fonti di dati degli agenti
- Seguendo la pratica consigliata di concedere solo autorizzazioni di sola lettura alle fonti di dati connesse, inclusi i server MCP personalizzati

Utilizzo dei dati

AWS non utilizza i dati degli agenti, i messaggi di chat o i dati provenienti da fonti di dati integrate per addestrare modelli o migliorare il prodotto. AWS DevOps Agent Space utilizza il feedback dei clienti integrato nel prodotto per migliorare le risposte e le indagini degli agenti, ma AWS non lo utilizza per migliorare il servizio stesso.

DevOps Autorizzazioni Agent IAM

AWS DevOps L'agente utilizza azioni AWS Identity and Access Management (IAM) specifiche del servizio per controllare l'accesso alle sue caratteristiche e funzionalità. Queste azioni determinano ciò che gli utenti possono fare all'interno della console dell' AWS DevOps agente e dell'app Web dell'operatore. Questo è separato dalle autorizzazioni dell'API di AWS servizio che l'agente stesso utilizza per esaminare le risorse.

Per ulteriori informazioni sulla limitazione dell'accesso degli agenti, consulta Limitazione dell'accesso degli agenti in un account. AWS

Azioni di gestione di Agent Space

Queste azioni controllano l'accesso alla configurazione e alla gestione di Agent Space:

- aidevops: GetAgentSpace — Consente agli utenti di visualizzare i dettagli su un Agent Space, inclusa la configurazione, lo stato e gli account associati. Gli utenti necessitano di questa autorizzazione per accedere a un Agent Space nella console di AWS gestione.
- aidevops: GetAssociation — Consente agli utenti di visualizzare i dettagli su una specifica associazione di account, inclusa la configurazione del ruolo IAM e lo stato della connessione.
- aidevops: ListAssociations — Consente agli utenti di elencare tutte le associazioni di AWS account configurate per un Agent Space, inclusi gli account primari e secondari.

Azioni di indagine ed esecuzione

Queste azioni controllano l'accesso alle funzionalità di indagine sugli incidenti:

- aidevops: ListExecutions — Consente agli utenti di visualizzare i metadati di esecuzione, tra cui ID, stato e altro, per indagini, mitigazioni, valutazioni e conversazioni in chat associate a un'attività.
- aidevops: ListJournalRecords — Consente agli utenti di accedere a log dettagliati che mostrano le fasi di ragionamento dell'agente, le azioni intraprese e le fonti di dati consultate durante un'indagine, una mitigazione, una valutazione e una conversazione in chat. Ciò è utile per capire in che modo l'agente è giunto alle sue conclusioni.

Azioni di gestione della chat

La chat richiede le seguenti autorizzazioni IAM per funzionare:

- aidevops: ListChats — Consente agli utenti di elencare e accedere alla cronologia delle conversazioni in chat.
- aidevops: CreateChat — Consente agli utenti di creare nuove conversazioni in chat.
- aidevops: SendMessage — Consente agli utenti di inviare domande e ricevere risposte in streaming.

Topologia e azioni di scoperta

Queste azioni controllano l'accesso alle funzionalità di mappatura delle risorse delle applicazioni:

- aidevops: DiscoverTopology — Consente agli utenti di attivare il rilevamento e la mappatura della topologia per un Agent Space. Questa azione avvia il processo di scansione degli AWS account e la creazione della topologia delle risorse dell'applicazione.

Azioni di prevenzione e raccomandazione

Queste azioni controllano l'accesso alla funzionalità di prevenzione:

- aidevops: ListGoals — Consente agli utenti di visualizzare gli scopi e gli obiettivi di prevenzione a cui l'agente sta lavorando sulla base dei modelli di incidenti recenti.
- aidevops: ListRecommendations — Consente agli utenti di visualizzare tutti i consigli generati dalla funzione Prevenzione, inclusa la priorità e la categoria.
- aidevops: GetRecommendation — Consente agli utenti di visualizzare informazioni dettagliate su una raccomandazione specifica, compresi gli incidenti che avrebbe potuto prevenire e le linee guida all'implementazione.

Azioni di gestione delle attività di backlog

Queste azioni controllano la capacità di gestire i consigli come attività arretrate:

- aidevops: CreateBacklogTask — Consente agli utenti di creare un'attività di indagine sugli incidenti o di valutazione della prevenzione.
- aidevops: UpdateBacklogTask — Consente agli utenti di approvare un piano di mitigazione o annullare un'indagine o una valutazione attiva.
- aidevops: GetBacklogTask — Consente agli utenti di recuperare i dettagli su un'attività specifica.

- aidevops: ListBacklogTasks — Consente agli utenti di elencare le attività per un Agent Space, filtrandole per tipo di attività, stato, priorità o ora di creazione.

Azioni di gestione della conoscenza

Queste azioni controllano la capacità di aggiungere e gestire conoscenze personalizzate che l'agente può utilizzare durante le indagini:

- aidevops: CreateKnowledgeItem — Consente agli utenti di aggiungere elementi di conoscenza personalizzati, come competenze, guide alla risoluzione dei problemi o informazioni specifiche sull'applicazione a cui l'agente dovrebbe fare riferimento.
- aidevops: ListKnowledgeItems — Consente agli utenti di visualizzare tutte le informazioni configurate per un Agent Space.
- aidevops: GetKnowledgeItem — Consente agli utenti di recuperare i dettagli di uno specifico elemento di conoscenza.
- aidevops: UpdateKnowledgeItem — Consente agli utenti di modificare gli elementi di conoscenza esistenti per mantenere le informazioni aggiornate.
- aidevops: DeleteKnowledgeItem — Consente agli utenti di rimuovere elementi di conoscenza che non sono più pertinenti.

AWS Supporta le azioni di integrazione

Queste azioni controllano l'integrazione con i casi di AWS Support:

- aidevops: InitiateChatForCase — Consente agli utenti di avviare una sessione di chat con AWS Support direttamente da un'indagine, fornendo automaticamente un contesto sull'incidente.
- aidevops: EndChatForCase — Consente agli utenti di terminare una sessione di chat attiva di AWS Support case.
- aidevops: DescribeSupportLevel — Consente agli utenti di controllare il livello del piano di AWS supporto per l'account per determinare le opzioni di supporto disponibili.

Azioni di utilizzo e monitoraggio

Queste azioni controllano l'accesso alle informazioni sull'utilizzo:

- aidevops: GetAccountUsage — Consente agli utenti di visualizzare la quota mensile dell' AWS DevOps agente per le ore di indagine, le ore di valutazione della prevenzione e le richieste di chat, nonché l'utilizzo del mese corrente.

Esempi comuni di policy IAM

Policy amministratore

Questa politica garantisce l'accesso completo a tutte le funzionalità AWS DevOps dell'agente:

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

Politica dell'operatore

Questa politica garantisce l'accesso alle funzionalità di indagine e prevenzione senza capacità amministrative:

```
{
  "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 politica

Questa politica garantisce l'accesso in sola visualizzazione alle indagini e ai consigli:

```
{
  "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": "*"
    }
  ]
}
```

Utilizzo di ruoli collegati ai servizi per AWS DevOps Agente

AWS DevOps L'agente utilizza AWS ruoli collegati al servizio Identity and Access Management (IAM). Un ruolo collegato al servizio è un tipo unico di ruolo IAM collegato direttamente all'agente. AWS DevOps Service-linked i ruoli sono predefiniti dall' AWS DevOps agente e includono tutte le autorizzazioni richieste dal servizio per chiamare altri AWS servizi per conto dell'utente.

Service-linked autorizzazioni di ruolo

Il ruolo collegato ai servizi `AWSServiceRoleForAIDevOps` considera attendibile il principale del servizio `aidevops.amazonaws.com` ai fini dell'assunzione del ruolo.

Il ruolo utilizza la politica gestita `AWSServiceRoleForAIDevOpsPolicy` con le seguenti autorizzazioni:

- `cloudwatch:PutMetricData`— Pubblica le metriche di utilizzo nel `AWS/AIDevOps` CloudWatch namespace. Riguardato da una `cloudwatch:namespace` condizione che consente solo lo spazio dei nomi. `AWS/AIDevOps`
- `vpc-lattice:CreateResourceGateway`— Crea gateway di risorse VPC Lattice per connessioni private. Riguardato da una `aws:RequestTag/AWSAIDevOpsManaged` condizione, in modo che il servizio possa creare solo gateway di risorse con il tag. `AWSAIDevOpsManaged`
- `vpc-lattice:TagResource`— Etichetta i gateway di risorse VPC Lattice. Ambito da qualsiasi condizione. `aws:RequestTag/AWSAIDevOpsManaged`
- `vpc-lattice:DeleteResourceGateway`— Eliminare i gateway di risorse VPC Lattice. Riguardato da una `aws:ResourceTag/AWSAIDevOpsManaged` condizione, in modo che il servizio possa eliminare solo i gateway di risorse da lui creati.
- `vpc-lattice:GetResourceGateway`— Recupera informazioni sui gateway di risorse VPC Lattice. Riguardato da una `aws:ResourceTag/AWSAIDevOpsManaged` condizione in modo che il servizio possa leggere solo i gateway di risorse che ha creato.
- `ec2:DescribeVpcs`,`ec2:DescribeSubnets`, `ec2:DescribeSecurityGroups` — Recupera informazioni sulle risorse di rete VPC necessarie per configurare i gateway di risorse. Queste azioni

di sola lettura si applicano a tutte le risorse VPC perché l'API EC2 non supporta le autorizzazioni a livello di risorsa per le chiamate Descrivi.

- `iam:CreateServiceLinkedRole`— Creare il ruolo collegato al servizio VPC Lattice necessario per le operazioni di Resource Gateway. Questa autorizzazione è limitata solo al responsabile del `vpc-lattice.amazonaws.com` servizio e non può essere utilizzata per creare ruoli collegati ai servizi per nessun altro servizio.

Creazione del ruolo collegato ai servizi

Non devi creare manualmente il ruolo collegato al servizio `AWSServiceRoleForAIDevOps`. Quando inizi a utilizzare AWS DevOps Agent, il servizio crea automaticamente il ruolo collegato al servizio.

Per consentire al servizio di creare il ruolo per conto dell'utente, è necessario disporre dell'`iam:CreateServiceLinkedRole`autorizzazione. Ti consigliamo di definire l'ambito di questa autorizzazione con una `iam:AWSServiceName` condizione `aidevops.amazonaws.com` per seguire il principio del privilegio minimo. Per ulteriori informazioni, consulta le autorizzazioni dei Service-linked ruoli.

Modifica del ruolo collegato ai servizi

Non puoi modificare il ruolo collegato ai servizi `AWSServiceRoleForAIDevOps`, Dopo aver creato il ruolo, non è possibile modificarne il nome, poiché diverse entità potrebbero fare riferimento al ruolo per nome. È possibile tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, vedere Modifica di un ruolo collegato al servizio.

Eliminazione del ruolo collegato ai servizi

Se non è più necessario utilizzare AWS DevOps Agent, si consiglia di eliminare il ruolo collegato al `AWSServiceRoleForAIDevOps` servizio. Prima di poter eliminare il ruolo, è necessario rimuovere tutte le connessioni private configurate nell'Agent Space. L'eliminazione del ruolo collegato al servizio non rimuove automaticamente i gateway di risorse VPC Lattice contrassegnati con `AWSAIDevOpsManaged` che erano stati precedentemente creati dal servizio. È necessario eliminare questi gateway di risorse manualmente se non sono più necessari. Per ulteriori informazioni, vedere Eliminazione di un ruolo collegato al servizio.

AWS Politiche gestite per AWS DevOps Agente

AWS affronta molti casi d'uso comuni fornendo policy IAM autonome create e amministrate da AWS. Queste policy AWS gestite concedono le autorizzazioni necessarie per i casi d'uso comuni in

modo da evitare di dover esaminare quali autorizzazioni sono necessarie. Per ulteriori informazioni, consulta le policy AWS gestite nella _IAM User Guide_.

Le seguenti politiche AWS gestite, che puoi allegare agli utenti del tuo account, sono specifiche di Agent. AWS DevOps

AIDevOpsAgentReadOnlyAccess

Fornisce accesso in sola lettura ad Amazon DevOps Agent tramite la console AWS di gestione

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "AIDevOpsAgentReadOnlyAccess",
   "Effect": "Allow",
   "Action": [
    "aidevops:Get*",
    "aidevops:List*",
    "aidevops:SearchServiceAccessibleResource"
   ],
   "Resource": "*"
  }
 ]
}
```

AIDevOpsAgentFullAccess

Fornisce accesso completo ad Amazon DevOps Agent tramite la console AWS di gestione

```
{
 "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

Fornisce l'accesso per utilizzare l'app Web AWS DevOps dell'operatore per un 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

Fornisce le autorizzazioni richieste dall' AWS DevOps agente per condurre indagini ed eseguire analisi sulle risorse dei clienti AWS .

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

Limitazione dell'accesso degli agenti in un AWS Account

AWS DevOps L'agente utilizza i ruoli IAM per scoprire e descrivere AWS le risorse durante le indagini sugli incidenti e le valutazioni preventive. Puoi controllare il livello di accesso dell'agente configurando le policy IAM associate a questi ruoli. La topologia dell'applicazione non mostra tutto ciò a cui l'agente ha accesso: le policy IAM sono l'unico modo per limitare realmente le API e le risorse di AWS servizio a cui l'agente può accedere.

Comprensione dei ruoli IAM per AWS DevOps Agente

AWS DevOps L'agente utilizza i ruoli IAM per accedere alle risorse in due tipi di account:

- Ruolo principale dell'account: consente all'agente di accedere alle risorse dell' AWS account in cui si crea l'Agent Space.
- Ruoli dell'account secondario: consente all'agente di accedere alle risorse di AWS account aggiuntivi collegati all'Agent Space.

Per entrambi i tipi di account, è possibile limitare AWS i servizi a cui l'agente può accedere, limitare l'accesso a risorse specifiche all'interno di tali servizi e controllare in quali aree l'agente può operare.

Comprensione delle barriere di autorizzazione

AWS DevOps L'agente applica un limite di autorizzazioni a ogni sessione che crea quando accede alle tue risorse. AWS Questa barriera funge da limite: definisce il set massimo di autorizzazioni che l'agente può mai utilizzare, indipendentemente dalle autorizzazioni concesse per il ruolo IAM.

Come funziona

Quando l'agente assume il tuo ruolo IAM, passa una policy di sessione che limita le autorizzazioni effettive per quella sessione. Le autorizzazioni effettive sono l'intersezione di:

1. Le tue politiche di ruolo IAM: la politica gestita e tutte le politiche in linea che alleghi al ruolo.
2. La barriera delle autorizzazioni: una politica di sessione applicata dall' AWS DevOps agente al momento dell'assunzione del ruolo.

Un'autorizzazione deve essere presente in entrambi i livelli per avere effetto. Se aggiungi un'autorizzazione al tuo ruolo che non è inclusa nel guardrail, l'agente non può utilizzarla.

Autorizzazioni predefinite

La policy `AIDevOpsAgentAccessPolicy` gestita fornisce il set predefinito di autorizzazioni di sola lettura che l'agente utilizza per le indagini. Queste autorizzazioni sono incluse nel guardrail, quindi funzionano senza configurazioni aggiuntive.

Estensione delle autorizzazioni oltre quelle predefinite

AWS DevOps L'agente supporta un set curato di autorizzazioni aggiuntive oltre alla politica gestita predefinita. Queste autorizzazioni sono incluse nel guardrail ma non sono abilitate per impostazione predefinita. Per utilizzarli, aggiungi le autorizzazioni specifiche al tuo ruolo come politica in linea.

Ad esempio, per consentire all'agente di leggere gli oggetti dai tuoi bucket S3 durante le indagini, aggiungi una policy in linea al tuo ruolo:

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

Poiché `s3:GetObject` `s3:ListBucket` sono inclusi nel guardrail, questa politica in linea ha effetto. È possibile assegnare l'ambito `Resource` a bucket specifici per seguire il principio del privilegio minimo.

Autorizzazioni aggiuntive supportate

Le seguenti autorizzazioni sono incluse nel guardrail e possono essere abilitate aggiungendole al proprio ruolo come policy in linea. Queste non sono concesse per impostazione predefinita: è necessario attivarle esplicitamente.

Servizio	Azioni	Caso d'uso
Simple Storage Service (Amazon S3)	`s3:GetObject` , `s3:ListBucket`	Leggi i dati, i log o la configura zione dell'applicazione archiviati in S3
AWS Connect diretto	`directconnect:Desc ribeConnections` , `directconnect:Desc ribeDirectConnectG atewayAssociations` , `directconnect:Desc ribeDirectConnectG ateways` , `directcon nect:DescribeLags` , `directconnect:Desc ribeVirtualInterfa ces`	Analizza i problemi di connettività di rete

> Nota: questo elenco potrebbe espandersi nel tempo man mano che nuove funzionalità vengono aggiunte ad AWS DevOps Agent. Le autorizzazioni non elencate qui o nella politica `AIDevOpsAgentAccessPolicy` gestita sono bloccate dal guardrail.

Autorizzazioni bloccate dal guardrail

Se aggiungi un'autorizzazione al tuo ruolo che non è nel guardrail, l'agente non può utilizzarla. Ciò è dovuto alla progettazione: il guardrail impedisce all'agente di eseguire azioni al di fuori dell'ambito previsto, anche se il ruolo le consentirebbe altrimenti.

Ad esempio, operazioni di scrittura come `s3:PutObjectec2:TerminateInstances`, o non `dynamodb:DeleteItem` sono incluse nel guardrail. Anche se il tuo ruolo concede queste autorizzazioni, l'agente non può eseguire queste azioni.

Riepilogo

Livello	Chi lo controlla	Scopo
Politiche di ruolo IAM	Utente corrente	Definisci cosa vuoi che l'agente sia in grado di fare
Guardrail di autorizzazione	AWS DevOps Agente	Definisce il massimo che l'agente può fare
Autorizzazioni valide	Intersezione di entrambi	Cosa può effettivamente fare l'agente

Questo modello garantisce che l'agente operi entro un limite di sicurezza ben definito, offrendo al contempo la flessibilità necessaria per estenderne le funzionalità per ogni caso d'uso specifico.

Scelta dei limiti delle risorse

Quando si limita l'accesso alle risorse, è necessario includere autorizzazioni sufficienti per consentire all'agente di indagare correttamente sugli incidenti delle applicazioni. Questo include:

- Tutte le risorse per le applicazioni pertinenti che l'agente deve monitorare e analizzare
- Tutta l'infrastruttura di supporto da cui dipendono tali applicazioni

L'infrastruttura di supporto può includere:

- Componenti di rete (VPC, sottoreti, sistemi di bilanciamento del carico, gateway API)
- Archivi dati (database, cache, storage di oggetti)
- Risorse di calcolo (istanze EC2, funzioni Lambda, contenitori)
- Servizi di monitoraggio e registrazione (,) CloudWatch CloudTrail
- Risorse per la gestione delle identità e degli accessi necessarie per comprendere le autorizzazioni

Se si limita l'accesso in modo troppo restrittivo, l'agente potrebbe non essere in grado di identificare le cause principali che hanno origine nel supporto dell'infrastruttura al di fuori dei confini definiti.

Limitazione dell'accesso al servizio

Puoi limitare AWS i servizi a cui l'agente può accedere modificando le policy IAM associate ai ruoli dell'agente. Quando crei policy personalizzate, segui queste best practice:

- Concedi solo autorizzazioni di sola lettura: l'agente deve leggere le configurazioni delle risorse, le metriche e i registri durante le indagini. Evita di concedere autorizzazioni che consentano all'agente di modificare o eliminare risorse.
- Limita ai servizi necessari: includi solo i AWS servizi che contengono risorse pertinenti alle tue applicazioni. Ad esempio, se la tua applicazione non utilizza Amazon RDS, non includere le autorizzazioni RDS nella policy.
- Usa azioni specifiche anziché caratteri jolly: invece di concedere `service:*` autorizzazioni, specifica azioni individuali come o. `cloudwatch:GetMetricData` `ec2:DescribeInstances`

Esempio di politica che si limita a servizi specifici:

```
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": "*"
    }
  ]
}
```

Limitazione dell'accesso alle risorse

Per limitare l'agente a risorse specifiche all'interno di un servizio, utilizza le autorizzazioni a livello di risorsa nelle tue policy IAM. Ciò consente di concedere l'accesso solo alle risorse che corrispondono a modelli specifici.

Utilizzo dei modelli ARN delle risorse:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lambda:GetFunction",
        "lambda:GetFunctionConfiguration"
      ],
      "Resource": "arn:aws:lambda:*:*:function:production-*"
    }
  ]
}
```

Questo esempio limita l'agente ad accedere solo alle funzioni Lambda con nomi che iniziano con «production-».

Utilizzo di restrizioni basate su tag:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeInstanceStatus"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Environment": "production"
        }
      }
```

```
    }
  ]
}
```

Questo esempio limita l'agente ad accedere solo alle istanze EC2 contrassegnate con. `Environment=production`

Limitazione dell'accesso regionale

Per limitare AWS le regioni a cui l'agente può accedere, utilizza la chiave di `aws:RequestedRegion` condizione nelle tue policy IAM:

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

Questo esempio limita l'agente all'accesso alle risorse solo nelle regioni us-east-1 e us-west-2.

Creazione di politiche IAM personalizzate

Quando crei un Agent Space o aggiungi account secondari, hai la possibilità di creare un ruolo IAM personalizzato utilizzando un modello di policy. Ciò consente di implementare il principio del privilegio minimo.

Quando si crea un Agent Space

Dalla console dell' DevOps agente nella console AWS di gestione...

- Scegli Crea un nuovo ruolo di DevOps agente utilizzando un documento di policy e segui le istruzioni

Quando modifichi un Agent Space

Dalla console dell' DevOps agente nella console AWS di gestione...

- Seleziona la scheda Funzionalità
- Seleziona l'account secondario che desideri modificare dalla sezione Cloud e fai clic su Modifica
- Scegli Crea una nuova policy per l' DevOps agente utilizzando un modello e segui le istruzioni

Best practice relative alle policy personalizzate

- Concedi autorizzazioni di sola lettura: evita le autorizzazioni che consentono la modifica o l'eliminazione delle risorse
- Usa le autorizzazioni a livello di risorsa quando possibile: limita l'accesso a risorse specifiche utilizzando modelli o tag ARN
- Esamina e verifica regolarmente le autorizzazioni: esamina periodicamente le politiche IAM dell'agente per assicurarti che siano ancora in linea con i tuoi requisiti di sicurezza

Configurazione dell'autenticazione IAM Identity Center

L'autenticazione IAM Identity Center offre un modo centralizzato per gestire l'accesso degli utenti all'applicazione web AWS DevOps Agent Space. Questa guida spiega come configurare l'autenticazione IAM Identity Center e gestire gli utenti.

Prerequisiti

Prima di configurare l'autenticazione IAM Identity Center, assicurati di avere:

- IAM Identity Center è abilitato nella tua organizzazione o nel tuo account
- Autorizzazioni di amministratore in Agent AWS DevOps
- Un Agent Space configurato o pronto per la creazione

Opzioni di autenticazione

AWS DevOps Agent offre due metodi di autenticazione per accedere all'app web Agent Space:

Autenticazione IAM Identity Center: consigliata per ambienti di produzione. Fornisce gestione centralizzata degli utenti, integrazione con provider di identità esterni e sessioni fino a 12 ore.

Accesso amministrativo (autenticazione IAM): fornisce un accesso rapido agli amministratori durante la configurazione e la configurazione iniziali. Le sessioni sono limitate a 30 minuti.

Configurazione di IAM Identity Center durante la creazione di Agent Space

Quando crei un Agent Space, puoi configurare l'autenticazione IAM Identity Center nella scheda Accesso:

Passaggio 1: vai alla configurazione dell'app Web

1. Dopo aver configurato i dettagli di Agent Space e l'accesso all' AWS account, procedi alla scheda Accesso
2. Vedrai due sezioni: «Connect IAM Identity Center» e «Accesso amministratore»

Fase 2: Configurazione dell'integrazione con IAM Identity Center

Nella sezione Connect [Agent Space] a IAM Identity Center:

1. Verifica l'istanza di IAM Identity Center: la console mostra quale istanza di Identity Center gestirà l'accesso degli utenti della Web App (ad esempio,`ssoins-7223a9580931edbe`). L'istanza IAM Identity Center più vicina verrà automaticamente precompilata.
2. Seleziona l'opzione IAM Identity Center Application Role Name: scegli una delle tre opzioni:

Crea automaticamente un nuovo ruolo di DevOps agente (consigliato):

- Il sistema crea automaticamente un nuovo ruolo di servizio con le autorizzazioni appropriate
- Questa è l'opzione più semplice e funziona per la maggior parte dei casi d'uso

Assegna un ruolo esistente:

- Usa un ruolo IAM esistente che hai già creato

- Il sistema verificherà che il ruolo disponga delle autorizzazioni richieste
- Scegli questa opzione se la tua organizzazione ha ruoli precreati per Agente AWS DevOps

Crea un nuovo ruolo di DevOps agente utilizzando un modello di policy:

- Utilizza i dettagli della policy forniti per creare il tuo ruolo personalizzato nella console IAM
- Scegli questa opzione se devi personalizzare le autorizzazioni dei ruoli

Dopo aver fatto clic su Connect, il sistema automaticamente:

- Crea o configura il ruolo IAM specificato
- Configura un'applicazione IAM Identity Center per il tuo Agent Space
- Stabilisce relazioni di fiducia tra IAM Identity Center e l'app web Agent Space
- Configura i flussi di autenticazione OAuth 2.0 per l'accesso sicuro degli utenti

Alternativa: utilizzo dell'accesso da amministratore

Se desideri accedere immediatamente all'app web Agent Space senza configurare IAM Identity Center:

1. Nella sezione Accesso amministratore, annota l'ARN del ruolo IAM che fornisce l'accesso all'amministratore (ad esempio,) `arn:aws:iam::440491339484:role/service-role/DevOpsAgentRole-WebappAdmin-15ppoc42`
2. Fai clic sul pulsante blu di accesso all'amministratore per avviare l'app web Agent Space con autenticazione IAM
3. Le sessioni che utilizzano questo metodo sono limitate a 30 minuti

> **Note**
>
> L'accesso da amministratore è destinato alla configurazione e alla configurazione iniziali. Per l'uso in produzione e le operazioni in corso, configura l'autenticazione IAM Identity Center.

Aggiungere utenti e gruppi

Dopo aver configurato l'autenticazione IAM Identity Center, devi concedere a utenti e gruppi specifici l'accesso all'app web Agent Space:

Fase 1: Accesso alla gestione degli utenti

1. Nella console dell' AWS DevOps agente, seleziona il tuo Agent Space
2. Vai alla scheda Accesso
3. In Accesso utente, fai clic su Gestisci utenti e gruppi

Passaggio 2: aggiungere utenti o gruppi

1. Scegli Aggiungi utenti o gruppi
2. Cerca utenti o gruppi nella tua directory IAM Identity Center
3. Seleziona le caselle di controllo accanto agli utenti o ai gruppi che desideri aggiungere
4. Fai clic su Aggiungi per concedere loro l'accesso

Gli utenti selezionati possono ora accedere all'app web Agent Space utilizzando le proprie credenziali IAM Identity Center.

Lavorare con provider di identità esterni

Se utilizzi un provider di identità esterno (come Okta, Microsoft Entra ID o Ping Identity) con IAM Identity Center:

- Gli utenti e i gruppi vengono sincronizzati dal tuo provider di identità esterno a IAM Identity Center
- Quando aggiungi utenti e gruppi all'app web Agent Space, effettui una selezione dalla directory sincronizzata
- Gli attributi utente e le appartenenze ai gruppi vengono gestiti dal provider di identità esterno
- Le modifiche al provider di identità si riflettono automaticamente in IAM Identity Center dopo la sincronizzazione

In che modo gli utenti accedono all'app web Agent Space

Dopo aver aggiunto utenti a Agent Space:

1. Condividi l'URL dell'app web Agent Space con gli utenti autorizzati
2. Quando gli utenti accedono all'URL, vengono reindirizzati alla pagina di accesso di IAM Identity Center
3. Dopo aver inserito le credenziali (e completato l'MFA, se configurato), vengono reindirizzati all'app web Agent Space
4. La loro sessione è valida per 8 ore per impostazione predefinita (configurabile dall'amministratore dell'Identity Center)

Gestione dell’accesso degli utenti

Puoi aggiornare l'accesso degli utenti in qualsiasi momento:

Aggiungere altri utenti o gruppi:

- Segui gli stessi passaggi descritti sopra per aggiungere altri utenti o gruppi

Rimuovere l'accesso:

1. Nella sezione Accesso utente, trova l'utente o il gruppo da rimuovere
2. Fai clic sul pulsante Rimuovi accanto al suo nome
3. Conferma la rimozione

Gli utenti rimossi perderanno immediatamente l'accesso, ma le sessioni attive potrebbero continuare fino alla scadenza.

Gestione della sessione

Le sessioni IAM Identity Center per l'app web Agent Space hanno le seguenti caratteristiche:

- Durata della sessione predefinita: 8 ore
- Sicurezza della sessione: cookie solo HTTP per una maggiore protezione
- Autenticazione a più fattori: supportata se configurata in IAM Identity Center
- Credenziali API: le credenziali SigV4 di breve durata (15 minuti) vengono emesse per le chiamate API e rinnovate automaticamente

Per configurare la durata della sessione:

1. Accedi alla console IAM Identity Center
2. Vai a Impostazioni > Autenticazione
3. In Durata della sessione, configura la durata preferita (da 1 ora a 12 ore)
4. Scegliere Salva modifiche.

Disconnessione di Identity Center

1. Nella console di Agent Space, fai clic su Azioni in alto a destra e seleziona Disconnetti da IAM Identity Center
2. Conferma nella finestra di dialogo di conferma

Configurazione dell'autenticazione tramite provider di identità esterno (IdP)

L'autenticazione con provider di identità esterno (IdP) consente all'organizzazione di utilizzare un provider di identità compatibile con OIDC esistente, come Okta o Microsoft Entra ID, per gestire l'accesso degli utenti all'applicazione Web Agent Space. AWS DevOps Gli utenti accedono con le proprie credenziali aziendali direttamente tramite il tuo IdP, senza AWS richiedere IAM Identity Center.

Prerequisiti

Prima di configurare l'autenticazione IdP esterna, assicurati di avere:

- Un provider di identità compatibile con OIDC (Okta o Microsoft Entra ID)
- Accesso amministrativo al tuo provider di identità
- Autorizzazioni di amministratore per accedere alla console AWS DevOps dell'agente
- Un Agent Space configurato o pronto per la creazione

Come funziona

Quando configuri l'autenticazione IdP esterna:

- Gli utenti accedono all'URL dell'app Web Agent Space

- Vengono reindirizzati alla pagina di accesso del tuo provider di identità
- Dopo l'autenticazione con le credenziali aziendali, vengono reindirizzati nuovamente all'app web
- L'app Web scambia il token di autenticazione con AWS credenziali di breve durata destinate all'Agent Space

Le sessioni sono valide per un massimo di 8 ore. Le credenziali vengono aggiornate automaticamente utilizzando i token di aggiornamento OIDC senza richiedere agli utenti di effettuare nuovamente l'autenticazione.

Configurazione dell'autenticazione IdP esterna

Passaggio 1: registra un'applicazione nel tuo provider di identità

Scegli il tuo provider di identità e segui le istruzioni di configurazione corrispondenti.

Opzione A: Okta

1. Nella console di amministrazione Okta, vai su Applicazioni > Applicazioni e scegli Crea integrazione tra app
2. Seleziona OIDC - OpenID Connect come metodo di accesso e Applicazione Web come tipo di applicazione. Seleziona Next (Successivo).
3. Imposta un nome descrittivo per l'applicazione (ad esempio,) `AWS DevOps Agent`
4. In Tipo di sovvenzione, assicurati che sia selezionato quanto segue:
   - Codice di autorizzazione (predefinito)
   - Token di aggiornamento: necessario per l'aggiornamento della sessione. Se non abilitato, gli utenti non saranno in grado di mantenere le sessioni.

> **Note**
>
> Per impostazione predefinita, Okta non abilita il tipo di concessione Refresh Token. È necessario abilitarlo esplicitamente.

1. Per ora, lascia il reindirizzamento di accesso URIs come valore predefinito: lo aggiornerai dopo aver configurato Agent Space

2. In Assegnazioni, assegna gli utenti o i gruppi che devono avere accesso
3. Seleziona Salva
4. Nella scheda Generale dell'applicazione, prendete nota dei seguenti valori:
   - ID cliente
   - Client secret: scegli Copy per salvare questo valore in modo sicuro
5. Prendi nota del tuo dominio Okta: questo è l'URL dell'emittente (ad esempio,). `https://dev-12345678.okta.com`

> **Note**
>
> Nella scheda Accedi, verifica che l'emittente sia impostato su Okta URL (non dinamico). Ciò garantisce un URL emittente stabile.

> **Note**
>
> Non aggiungete un'attestazione di gruppo al token ID nella scheda Reclami del server di autorizzazione. AWS DevOps L'agente non utilizza l'iscrizione al gruppo del tuo IdP.

Opzione B: Microsoft Entra ID

1. Nel portale di Azure, vai a Microsoft Entra ID > Registrazioni app > Nuova registrazione
2. Imposta un nome descrittivo (ad esempio,) `AWS DevOps Agent`
3. In Tipi di account supportati, seleziona l'opzione appropriata per la tua organizzazione (in genere solo account presenti in questa directory organizzativa)
4. Lascia vuoto l'URI di reindirizzamento per ora. Scegli Registrati
5. Nella pagina Panoramica dell'applicazione, tenete presente i seguenti valori:
   - ID dell'applicazione (client): utilizzato come ID client durante la configurazione di Agent Space
   - ID di directory (tenant): utilizzato per creare l'URL dell'emittente
6. Passa a Certificati e segreti > Nuovo segreto del client
   - Imposta una descrizione e un periodo di scadenza
   - Scegli Aggiungi e copia immediatamente il valore segreto: non verrà più mostrato

7. L'URL dell'emittente per Entra ID segue questo formato. `{tenant-id}`Sostituiscilo con il tuo ID di directory (tenant) riportato nella fase 5:
   - `https://login.microsoftonline.com/{tenant-id}/v2.0`

> **Note**
>
> Non abilitare l'attestazione opzionale del gruppo nella configurazione del token. AWS DevOps L'agente non utilizza l'iscrizione al gruppo del tuo IdP.

Passaggio 2: abilitare l'app Operator con l'autenticazione IdP

1. Nella console dell' AWS DevOps agente, seleziona il tuo Agent Space
2. Vai alla scheda Accesso
3. In Accesso utente, scegli Provider di identità esterno
4. Nel modulo di configurazione, configura quanto segue:
   - Provider di identità: seleziona il tuo provider di identità (Okta o Microsoft Entra ID)
   - URL dell'emittente: l'URL dell'emittente OIDC del tuo provider di identità
   - ID client: l'ID client dell'applicazione OIDC che hai creato
   - Client Secret: il client secret dell'applicazione OIDC
5. In Identity Provider Application Role Name, scegli una delle tre opzioni:
   - Crea automaticamente un nuovo ruolo di DevOps agente (consigliato): crea un nuovo ruolo di servizio con le autorizzazioni appropriate
   - Assegna un ruolo esistente: utilizza un ruolo IAM esistente che hai già creato
   - Crea un nuovo ruolo di DevOps agente utilizzando un modello di policy: utilizza i dettagli forniti per creare il tuo ruolo nella console IAM
6. Controlla l'avviso di avviso relativo all'URL di callback visualizzato nella parte inferiore del modulo. Copia questo URL: dovrai aggiungerlo al reindirizzamento consentito dal tuo provider di identità URIs prima che gli utenti possano accedere.
7. Scegli Connect

Dopo aver scelto Connect, la console visualizza la configurazione del provider di identità esterno con i seguenti dettagli:

- Provider: il provider di identità selezionato
- URL dell'emittente: l'URL dell'emittente OIDC configurato
- ID client: l'ID client configurato
- IAM Role ARN: il ruolo IAM utilizzato per l'accesso degli utenti
- URL di callback: configura questo URL nel tuo provider di identità come URI di reindirizzamento consentito
- URL di accesso: utilizza questo URL per accedere all'app Web tramite il tuo provider di identità

Passaggio 3: aggiungi l'URL di callback al tuo provider di identità

Okta

1. Nella console di amministrazione Okta, vai alla scheda Generale dell'applicazione
2. In Login, scegli Modifica
3. Aggiungi l'URL di callback come URI di reindirizzamento dell'accesso:
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/callback`
4. (Facoltativo) Imposta l'URI di accesso iniziale per abilitare l'accesso avviato dall'IdP dalla dashboard di Okta:
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/login`
5. (Consigliato) Aggiungi un URI di reindirizzamento alla disconnessione per reindirizzare gli utenti all'app Web dopo il logout. In caso contrario, gli utenti potrebbero visualizzare una pagina di errore durante la disconnessione:
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome`
6. Seleziona Salva

ID Microsoft Entra

1. Nel portale di Azure, accedi alla pagina di autenticazione dell'applicazione
2. In Configurazioni della piattaforma, scegli Aggiungi una piattaforma > Web
3. Inserisci l'URL di callback come URI di reindirizzamento:
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/callback`
4. (Facoltativo) Aggiungi un URI di reindirizzamento alla disconnessione per reindirizzare gli utenti all'app Web dopo il logout:

- `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome`

5. Scegli Configura

Fase 4: Verifica la configurazione

1. Vai all'URL di accesso mostrato nella console:
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/login`
2. Dovresti essere reindirizzato alla pagina di accesso del tuo provider di identità
3. Accedi con le tue credenziali aziendali
4. Una volta completata l'autenticazione, verrai reindirizzato all'app web Agent Space

Aggiornamento della configurazione IdP

Puoi ruotare il segreto del client senza disconnetterti:

1. Nella console dell' AWS DevOps agente, seleziona il tuo Agent Space
2. Vai alla scheda Accesso
3. In Configurazione del provider di identità esterno, scegli Ruota il segreto del client
4. Inserisci il nuovo Client Secret
5. Seleziona Salva

Per modificare qualsiasi altro campo di configurazione IdP (ad esempio URL dell'emittente, ID client o provider di identità), devi disconnettere l'IdP esistente e configurarne uno nuovo.

In che modo gli utenti accedono all'app web Agent Space

Dopo aver configurato l'autenticazione IdP esterna:

- Condividi l'URL dell'app Web Agent Space con gli utenti autorizzati
- Quando gli utenti accedono all'URL, vengono reindirizzati alla pagina di accesso del provider di identità
- Dopo aver inserito le loro credenziali (e completato l'MFA, se configurato dal tuo IdP), vengono reindirizzati all'app web di Agent Space
- Le sessioni si aggiornano automaticamente: consulta Gestione delle sessioni per i dettagli

Gestione della sessione

Le sessioni IdP esterne per l'app Web Agent Space hanno le seguenti caratteristiche:

- Durata della sessione: le sessioni del browser durano fino a 8 ore. Questo non è configurabile in AWS DevOps Agent. Se la durata della sessione del tuo IdP supera le 8 ore, gli utenti possono essere riautenticati automaticamente alla visita successiva senza inserire le credenziali. Configura la durata delle sessioni e dei token del tuo IdP in base ai requisiti di sicurezza della tua organizzazione.
- Aggiornamento delle credenziali: le sessioni vengono aggiornate automaticamente utilizzando i token di aggiornamento OIDC senza richiedere agli utenti di effettuare nuovamente l'autenticazione
- Autenticazione a più fattori: supportata se configurata nel tuo provider di identità. L'IdP gestisce l'MFA durante l'accesso: non è necessaria alcuna configurazione aggiuntiva in Agent AWS DevOps

Comportamento del logout

Quando un utente fa clic su Logout nell'app Web:

1. Tutti i cookie di sessione vengono cancellati immediatamente
2. L'utente viene reindirizzato all'endpoint di logout OIDC del provider di identità per terminare la sessione SSO
3. Se è configurato un URI di reindirizzamento della disconnessione, l'utente viene reindirizzato alla pagina di benvenuto dell'app Web

Revoca dell'accesso utente

Per revocare immediatamente l'accesso di un utente, puoi revocare le sue sessioni direttamente nel portale di amministrazione del tuo provider di identità:

- Okta — Nella Okta Admin Console, vai su Directory > Persone, seleziona l'utente, scegli Altre azioni > Cancella sessioni utente
- ID Microsoft Entra: nel portale di Azure, accedi a Utenti, seleziona l'utente e scegli Revoca sessioni

Considerazioni relative alla sicurezza

Archiviazione segreta del client: il segreto del client fornito durante la configurazione viene crittografato utilizzando la chiave KMS gestita dal cliente, se ne hai fornita una durante la creazione di Agent Space, o una chiave di proprietà del servizio in caso contrario. Non viene mai restituito nelle risposte API o visualizzato nella console dopo la configurazione iniziale.

Rotazione segreta dei client: i segreti dei client Entra hanno una scadenza configurabile. Imposta un promemoria per ruotare il segreto prima che scada utilizzando l'opzione Ruota client secret nella console dell'agente. AWS DevOps Se il segreto scade, gli utenti non potranno accedere finché non verrà ruotato.

Gestione permanente dei token: la durata dei token (token di accesso, token di aggiornamento) emessi dal tuo provider di identità è controllata dalla configurazione del tuo IdP. Ti consigliamo di configurare la durata appropriata dei token nel tuo IdP:

- Okta : configura la durata dei token in Sicurezza > API > Server di autorizzazione > Politiche di accesso
- Microsoft Entra ID: configura la durata dei token utilizzando i criteri di durata dei token

Dichiarazione di gruppo: non abilita l'attestazione di gruppo nella configurazione del token del tuo provider di identità. AWS DevOps Al momento l'agente non utilizza l'iscrizione al gruppo del tuo IdP.

Identificatore utente: l' AWS DevOps agente utilizza un'attestazione specifica del provider per identificare in modo univoco gli utenti:

- Okta: utilizza l'attestazione del token ID `sub`
- ID Microsoft Entra: utilizza l'attestazione `oid` (identificatore dell'oggetto) del token ID

Questi identificatori sono immutabili e vengono visualizzati nei CloudTrail registri a fini di controllo.

Disconnessione dell'IdP esterno

1. Nella console dell' AWS DevOps agente, seleziona il tuo Agent Space
2. Vai alla scheda Accesso
3. In Accesso utente, scegli Disconnetti
4. Esamina gli impatti elencati nella finestra di dialogo di conferma e conferma

La disconnessione comporterà:

- Rimuovere la configurazione IdP dall'Agent Space
- Impedisci agli utenti di accedere tramite il provider di identità esterno
- Rimuovi la cronologia delle chat individuali e degli artefatti associata agli account utente IdP

Le sessioni utente attive continueranno fino alla scadenza o al successivo aggiornamento delle credenziali fallisce.

Risoluzione dei problemi

- Il reindirizzamento a IdP non riesce: verifica che l'URL dell'emittente corrisponda all'endpoint di rilevamento OIDC del tuo IdP. Per Okta, assicurati che l'emittente sia impostato su Okta URL (non dinamico) nella scheda Accedi. Per Entra, usa il formato. `https://login.microsoftonline.com/{tenant-id}/v2.0`
- Accesso negato o errore di policy (Okta): verifica che l'utente o il relativo gruppo sia assegnato all'applicazione in Assegnazioni. Seleziona Sign On > Sign On Policy.
- Errore di configurazione IdP dopo l'accesso: il tuo provider di identità non ha restituito un token di aggiornamento. Assicurati che l'`offline_access`ambito e il tipo di concessione del token di aggiornamento siano abilitati:
  - Okta: vai alla scheda Generale dell'applicazione e abilita la casella di controllo Refresh Token in Tipo di concessione
  - Entra — Vai alle autorizzazioni API e assicurati che **`offline_access`** sia elencato tra le autorizzazioni delegate
- L'autenticazione ha esito positivo ma l'app Web mostra un errore: verifica che l'URI di reindirizzamento nel tuo IdP corrisponda esattamente all'URL di callback mostrato nella console dell'agente. AWS DevOps
- Errori di autenticazione: se l'attestazione opzionale del gruppo è abilitata nel tuo IdP, disabilitala. AWS DevOps L'agente non utilizza le attestazioni di gruppo.
- L'accesso non riesce dopo l'autenticazione IdP: per Entra, la verifica non **`requestedAccessTokenVersion`** è impostata **`null`** nel manifesto dell'applicazione. Per Okta, verifica che l'URL dell'emittente sia corretto.
- Pagina di errore dopo aver fatto clic su Logout (Okta): se visualizzi un **`post_logout_redirect_uri`** errore dopo la disconnessione, aggiungilo **`https://`**

`{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome` come URI di reindirizzamento per la disconnessione nella scheda Generale dell'applicazione Okta.

- Gli utenti rimangono sulla pagina del provider di identità dopo il logout (Entra): per reindirizzare gli utenti all'app Web dopo il logout, aggiungilo **`https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome`** come URI di reindirizzamento nella pagina di autenticazione dell'applicazione Entra.

Crittografia a riposo per AWS DevOps Agent

AWS DevOps L'agente crittografa tutti i dati dei clienti inattivi. Per impostazione predefinita, AWS DevOps Agent utilizza chiavi AWS proprietarie per crittografare automaticamente i dati senza costi aggiuntivi. Non è possibile visualizzare, gestire o controllare l'uso delle chiavi di AWS proprietà. Tuttavia, non è necessario intraprendere alcuna azione per proteggere queste chiavi. I tuoi dati vengono protetti automaticamente.

Puoi scegliere di crittografare i tuoi dati utilizzando una chiave simmetrica gestita dal cliente che crei, possiedi e gestisci in AWS Key Management Service (KMS).AWS Poiché hai il pieno controllo di questo livello di crittografia, puoi eseguire attività come le seguenti:

- Stabilire e mantenere le policy delle chiavi
- Abilitare e disabilitare le policy delle chiavi
- Ruotare i materiali crittografici delle chiavi
- Aggiungere tag
- Creare alias delle chiavi
- Pianificare l'eliminazione delle chiavi

Per ulteriori informazioni, consulta Customer managed keys nella AWS Key Management Service Developer Guide.

> **Note**
>
> AWS DevOps L'agente abilita automaticamente la crittografia dei dati AWS inattivi utilizzando chiavi proprietarie per proteggere gratuitamente i dati dei clienti. Le tariffe AWS KMS standard si applicano quando si utilizza una chiave gestita dal cliente. Per ulteriori informazioni sui prezzi, consulta i prezzi del servizio di gestione delle AWS chiavi.

Chiavi gestite dal cliente

Le chiavi gestite dal cliente sono chiavi KMS del tuo AWS account che crei, possiedi e gestisci. Hai il pieno controllo su queste chiavi KMS, inclusa la definizione e il mantenimento delle relative politiche chiave.

Quando configuri una chiave gestita dal cliente, AWS DevOps Agent la utilizza per proteggere i dati sensibili delle risorse. AWS DevOps Agent utilizza la crittografia a busta con il portachiavi gerarchico AWS Encryption SDK. La chiave KMS viene utilizzata per generare chiavi di filiale, che a loro volta proteggono i dati.

Puoi specificare una chiave gestita dal cliente quando crei le seguenti risorse:

- Agent Space: crittografa i dettagli e i contenuti di Agent Space creati dall' DevOps Agent Web App relativi a indagini, competenze e chat.
- Servizio: crittografa le credenziali di servizio di terze parti inutilizzate.

Per configurare una chiave gestita dal cliente in AWS DevOps Agent, segui questi passaggi.

Fase 1: creare una chiave gestita dal cliente

Puoi creare una chiave simmetrica gestita dal cliente utilizzando la console AWS KMS o l' AWS API KMS. La chiave deve soddisfare i seguenti requisiti:

Proprietà	Requisito
Tipo di chiavi	Simmetria
Specifica della chiave	`SYMMETRIC_DEFAULT`
Utilizzo delle chiavi	`ENCRYPT_DECRYPT`

> **Note**
>
> AWS DevOps L'agente supporta solo chiavi KMS con crittografia simmetrica con le specifiche della `SYMMETRIC_DEFAULT` chiave e l'utilizzo della chiave. `ENCRYPT_DECRYPT` Le chiavi multiregionali e le chiavi asimmetriche non sono attualmente supportate.

Per ulteriori informazioni, consulta Creazione di una chiave simmetrica gestita dal cliente nella Key Management Service Developer Guide.AWS

Fase 2: Impostare la politica chiave

Le policy della chiave controllano l'accesso alla chiave gestita dal cliente. Ogni chiave gestita dal cliente deve avere esattamente una policy della chiave, che contiene istruzioni che determinano chi può usare la chiave e come la possono usare.

La tua policy chiave deve concedere le autorizzazioni sia al principale chiamante (la tua identità IAM) che al servizio AWS DevOps Agent. AWS DevOps L'agente accede alla tua chiave utilizzando due set di credenziali:

1. Le credenziali del chiamante: utilizzate per tutte le operazioni sincrone, tra cui la convalida delle chiavi, la crittografia al momento della creazione delle risorse e qualsiasi chiamata API che restituisca una risposta diretta al chiamante.
2. AWS DevOps Agent Service Principal: utilizzato per operazioni asincrone eseguite in background, come indagini operative, analisi degli incidenti, correlazione degli eventi e generazione di analisi delle cause principali.

La tabella seguente elenca le azioni KMS richieste:

Azione KMS	Description
`kms:DescribeKey`	Convalida la configurazione delle chiavi al momento della creazione delle risorse
`kms:GenerateDataKey`	Genera chiavi di crittografia dei dati per la crittografia delle buste
`kms:Decrypt`	Decrittare i dati
`kms:Encrypt`	Crittografare i dati
`kms:ReEncrypt`	Crittografa nuovamente i dati con la stessa chiave o con una chiave diversa

AWS DevOps L'agente convalida tutte queste autorizzazioni al momento della configurazione utilizzando operazioni di esecuzione a secco. Se manca un'autorizzazione, la richiesta ha esito negativo con un'eccezione.

Di seguito è riportato un esempio di policy della chiave. Sostituisci i valori segnaposto con i tuoi.

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

```
}
```

La politica contiene le seguenti dichiarazioni:

- AllowKeyAdministration— Concede alla radice dell'account l'accesso amministrativo completo alla chiave. `111122223333`Sostituiscila con l'ID AWS del tuo account.
- AllowCallerAccessViaService— Concede ai responsabili IAM le autorizzazioni KMS necessarie per tutte le operazioni sincrone degli agenti. AWS DevOps Ciò include la convalida delle chiavi al momento della creazione delle risorse, nonché le operazioni di crittografia e decrittografia per qualsiasi chiamata API che restituisca una risposta diretta al chiamante. La `kms:ViaService` condizione garantisce che sia possibile utilizzare la chiave solo tramite il servizio Agent. AWS DevOps `111122223333`Sostituiscila con l'ID AWS del tuo account e `us-east-1` con la tua AWS regione.
- AllowDevOpsAgentServiceAccessForAgentSpace/AllowDevOpsAgentServiceAccessForService— Concede al responsabile del `aidevops.amazonaws.com` servizio le autorizzazioni KMS necessarie per le operazioni asincrone. AWS DevOps L'agente utilizza questo principio di servizio per crittografare e decrittografare i dati durante l'esecuzione di operazioni in background come indagini operative, analisi degli incidenti, correlazione di eventi tra i servizi e generazione di analisi delle cause principali. Senza questo accesso, l' AWS DevOps Agent non può leggere i dati crittografati necessari per svolgere indagini per conto dell'utente. La `aws:SourceArn` condizione limita l'accesso alle richieste provenienti dalle risorse AWS DevOps dell'agente e garantisce che il `kms:EncryptionContext` contesto di crittografia corrisponda alla risorsa. ARNs `111122223333`Sostituiscilo con l'ID AWS del tuo account e `us-east-1` con la tua AWS regione.

Per ulteriori informazioni sulle politiche chiave, consulta le politiche chiave in AWS KMS nella AWS Key Management Service Developer Guide.

Passaggio 3: Specificare la chiave durante la creazione di una risorsa

Dopo aver creato la chiave e configurato la politica chiave, è possibile specificare la chiave durante la creazione delle risorse AWS DevOps dell'agente.

Console

Per configurare una chiave gestita dal cliente durante la creazione di un Agent Space nella console:

1. Apri la console AWS DevOps dell'agente.
2. Scegli Create Agent Space o Register Service.

3. Inserisci i dettagli dello spazio dell'agente (nome, descrizione e ruolo IAM).
4. Espandi la sezione Configurazione avanzata.
5. In Tipo di chiave di crittografia, seleziona Chiave gestita dal cliente.
6. Scegli una chiave KMS dall'elenco a discesa o inserisci un ARN per la chiave KMS.
7. Rivedi la politica chiave visualizzata nella sezione Politica chiave espandibile. Assicurati di aver allegato questa politica alla tua chiave KMS. Puoi utilizzare il pulsante Copia per copiare la politica.
8. Completa la configurazione rimanente e scegli Crea.

> **Note**
>
> Se non vedi la tua chiave KMS nell'elenco a discesa, verifica che la chiave soddisfi i requisiti del passaggio 1 e di disporre `kms:ListKeys` delle autorizzazioni necessarie. `kms:DescribeKey`

"Hello, World!"

Creazione di un Agent Space con una chiave gestita dal cliente

Specificate il `kmsKeyArn` parametro quando create uno spazio agente. Il valore deve essere l'ARN completo della chiave KMS.

```
{
  "name": "my-agent-space",
  "description": "An encrypted agent space",
  "kmsKeyArn": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

Registrazione di un servizio con una chiave gestita dal cliente

Specificare il `kmsKeyArn` parametro al momento della registrazione di un servizio. Il valore deve essere l'ARN completo della chiave KMS. Questo parametro è supportato in tutti i tipi di servizio, inclusi i server Dynatrace,, ServiceNow PagerDuty GitLab GitHub, e MCP.

```
{
  "service": "dynatrace",
```

```
  "kmsKeyArn": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "serviceDetails": { ... }
}
```

> **Note**
>
> È necessario specificare la chiave gestita dal cliente al momento della creazione della risorsa. Non è possibile aggiungere o modificare la chiave gestita dal cliente per una risorsa esistente.

AWS DevOps Contesto di crittografia dell'agente

Un contesto di crittografia è un insieme di coppie chiave-valore non segrete che contengono informazioni contestuali aggiuntive sui dati. AWS KMS utilizza il contesto di crittografia come dati autenticati aggiuntivi per supportare la crittografia autenticata. Quando includi un contesto di crittografia in una richiesta di crittografia dei dati, AWS KMS associa il contesto di crittografia ai dati crittografati. Per decrittografare i dati, è necessario includere lo stesso contesto di crittografia nella richiesta.

AWS DevOps L'agente utilizza il seguente contesto di crittografia in tutte le operazioni crittografiche:

```
{
  "aws-crypto-ec:aws:aidevops:arn": "arn:aws:aidevops:{region}:{accountId}:
{resourceType}/{resourceId}"
}
```

Il valore del contesto di crittografia è l'ARN della risorsa AWS DevOps Agent da crittografare. È possibile utilizzare questo contesto di crittografia nelle condizioni della politica chiave e nei AWS CloudTrail registri per verificare come viene utilizzata la chiave.

Gestione delle chiavi

Se disabiliti o pianifichi l'eliminazione della tua chiave KMS, AWS DevOps Agent non può decrittografare i tuoi dati. Ciò comporta `AccessDeniedException` errori nelle operazioni di lettura dei dati crittografati.

> **Important**
>
> Se scegli di utilizzare una chiave gestita dal cliente, sei responsabile della gestione della chiave e delle relative autorizzazioni. Se la chiave viene disabilitata o eliminata, o se AWS DevOps Agent perde l'autorizzazione a utilizzare la chiave, si perde l'accesso ai dati crittografati.

La tabella seguente descrive gli scenari di errore più comuni:

Azione	Impatto
Autorizzazioni politiche chiave revocate	`AccessDeniedException` sulle operazioni di crittografia e decrittografia
La chiave KMS è disabilitata	`DisabledException` sulle operazioni di crittografia e decrittografia
La chiave KMS è pianificata per l'eliminazione	`KMSInvalidStateException` sulle operazioni di crittografia e decrittografia
La chiave KMS viene eliminata	Perdita permanente dei dati: i dati crittografati non possono essere recuperati

Prima di disabilitare o eliminare una chiave:

1. Verificate che nessuna risorsa attiva AWS DevOps dell'agente dipenda dalla chiave.
2. Valuta la possibilità di disabilitare prima la chiave per testarne l'impatto prima di pianificare l'eliminazione.
3. AWS KMS impone un periodo di attesa minimo prima dell'eliminazione della chiave, dandoti il tempo di annullare se necessario.

   Nota: AWS DevOps l'agente non cripta automaticamente i dati con una nuova chiave. Se è necessario passare a una nuova chiave gestita dal cliente, è necessario creare una nuova risorsa con la nuova chiave.

Monitoraggio delle chiavi di crittografia

Quando utilizzi una chiave gestita dal cliente con AWS DevOps Agent, puoi utilizzarla AWS CloudTrailper tenere traccia delle richieste che l' AWS DevOps agente invia a AWS KMS.

Puoi filtrare CloudTrail gli eventi per:

- Fonte dell'evento: `kms.amazonaws.com`
- Chiave contestuale di crittografia: `aws-crypto-ec:aws:aidevops:arn`
- Key ARN: l'ARN chiave gestito dal cliente nei parametri della richiesta

Per ulteriori informazioni, consulta la sezione Registrazione delle chiamate all'API AWS KMS AWS CloudTrail nella AWS Key Management Service Developer Guide.

Endpoint VPC (AWS PrivateLink)

Puoi usarlo AWS PrivateLink per creare una connessione privata tra il tuo VPC e AWS DevOps Agent. Puoi accedere ad AWS DevOps Agent come se fosse nel tuo VPC, senza l'uso di un gateway Internet, un dispositivo NAT, una connessione VPN o una connessione Direct Connect. Le istanze nel tuo VPC non necessitano di indirizzi IP pubblici per accedere all' AWS DevOps agente.

Questa connessione privata viene stabilita creando un endpoint di interfaccia, alimentato da. AWS PrivateLink In ciascuna sottorete viene creata un'interfaccia di rete endpoint da abilitare per l'endpoint di interfaccia. Si tratta di interfacce di rete gestite dai richiedenti che fungono da punto di ingresso per il traffico destinato all'agente. AWS DevOps

Per ulteriori informazioni, consulta Access AWS services through AWS PrivateLink nella _AWS Guide_. PrivateLink

Considerazioni sugli endpoint AWS DevOps Agent VPC

Prima di configurare un endpoint di interfaccia per AWS DevOps Agent, consulta le considerazioni nella _AWS Guide_. PrivateLink

AWS DevOps L'agente supporta l'esecuzione di chiamate API tramite i seguenti endpoint VPC.

Categoria	Suffisso dell'endpoint
AWS DevOps Azioni dell'API Agent Control Plane	`aidevops`
AWS DevOps Operazioni Agent Runtime	`aidevops-dataplane`
AWS DevOps Eventi Agent Webhook	`event-ai`

Crea un endpoint di interfaccia per Agent AWS DevOps

Puoi creare un endpoint di interfaccia per AWS DevOps Agent utilizzando la console Amazon VPC o l'interfaccia a riga di comando (AWS AWS CLI). Per ulteriori informazioni, consulta Creare un endpoint di interfaccia nella _AWS Guide_. PrivateLink

Crea un endpoint di interfaccia per AWS DevOps Agent utilizzando i seguenti nomi di servizio:

- com.amazonaws. {regione} .aidevops
- com.amazonaws. {regione} .aidevops-dataplane
- com.amazonaws. {regione} .event-ai

Dopo aver creato l'endpoint, puoi abilitare un nome host DNS privato. Abilita questo nome host selezionando Abilita nome DNS privato nella console VPC quando crei l'endpoint VPC.

Se abiliti il DNS privato per l'endpoint dell'interfaccia, puoi effettuare richieste API all' AWS DevOps agente utilizzando il nome DNS regionale predefinito. L'esempio seguente mostra il formato del nome DNS regionale predefinito.

- cp.aidevops. {regione} .api.aws
- dp.aidevops. {regione} .api.aws
- event-ai. {regione} .api.aws

Creazione di una policy dell' endpoint per l'endpoint dell'interfaccia

Una policy dell'endpoint è una risorsa IAM che è possibile allegare all'endpoint dell'interfaccia. La policy predefinita per gli endpoint consente l'accesso completo all' AWS DevOps agente tramite

l'endpoint dell'interfaccia. Per controllare l'accesso consentito all' AWS DevOps agente dal tuo VPC, collega una policy endpoint personalizzata all'endpoint di interfaccia.

Una policy di endpoint specifica le informazioni riportate di seguito:

- I principali che possono eseguire azioni (AWS account, utenti IAM e ruoli IAM).
- Le azioni che possono essere eseguite.
- Le risorse in cui è possibile eseguire le operazioni.

Per ulteriori informazioni, consulta Controllare l'accesso ai servizi utilizzando le policy degli endpoint nella PrivateLink _AWS Guide_.

Convalida della conformità per l'agente AWS DevOps

I revisori di terze parti valutano la sicurezza e la conformità dei AWS servizi nell'ambito di più programmi di AWS conformità. AWS DevOps L'agente rientra nell'ambito dei seguenti programmi di conformità: 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 e SOC. PiTuKri Inoltre, l' AWS DevOps agente è idoneo all'HIPAA. I nostri revisori esterni esamineranno e testeranno AWS DevOps Agent durante i prossimi cicli di audit per questi programmi di conformità.

Per un elenco dei AWS servizi che rientrano nell'ambito di specifici programmi di conformità, consulta la sezione AWS Servizi rientranti nell'ambito del programma di conformità. Per informazioni generali, consultare Programmi per la conformità di AWS.

Puoi scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Scaricamento dei report in AWS Artifact.

La tua responsabilità di conformità quando utilizzi AWS DevOps Agent è determinata dalla sensibilità dei tuoi dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. AWS fornisce le seguenti risorse per contribuire alla conformità:

- Guide introduttive su sicurezza e conformità: queste guide all'implementazione illustrano considerazioni sull'architettura e forniscono passaggi per implementare ambienti di base incentrati sulla sicurezza e la conformità. AWS
- AWS risorse per la conformità: una raccolta di cartelle di lavoro e guide che potrebbero riguardare il settore e la località in cui operi.

- AWS Config: questo AWS servizio valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida e alle normative del settore.
- AWS Security Hub: questo AWS servizio offre una visione completa dello stato di sicurezza interno AWS. Security Hub utilizza i controlli di sicurezza per valutare le AWS risorse e verificare la conformità rispetto agli standard e alle best practice del settore della sicurezza.

Quote

AWS DevOps Le quote degli agenti includono il numero di spazi per gli agenti, le indagini simultanee e altro ancora. È possibile richiedere aumenti per alcune quote, ma non tutte le quote possono essere aumentate. Questi aumenti non vengono concessi immediatamente, quindi potrebbero essere necessari un paio d'ore o giorni prima che l'aumento diventi effettivo. Salvo diversa indicazione, ogni quota si applica a una Regione specifica.

La tabella seguente descrive le quote per AWS DevOps Agent.

Nome	Predefinita	Adattabile	Description
Spazi per agente per account per regione	100	Sì	Il numero massimo di spazi per agenti che è possibile creare per account in ciascuna AWS regione.
Indagini simultanee per spazio agente	3	Sì	Il numero massimo di indagini sulla risoluzione degli incidenti che possono essere eseguite contemporaneamente in un unico spazio agente.
Valutazioni simultanee per spazio agente	1	No	Il numero massimo di valutazioni sulla prevenzione degli incidenti che possono essere eseguite contemporaneamente in un singolo spazio agente.
Richiamazioni simultanee su	10	Sì	Il numero massimo di chiamate su richiesta

Nome	Predefinita	Adattabile	Description
richiesta per spazio agente			che possono essere eseguite DevOps contemporaneamente in un singolo spazio agente.

Richiedere un aumento della quota

È possibile richiedere un aumento della quota utilizzando una delle seguenti opzioni:

- Dalla console AWS di gestione: aprire la console Service Quotas. Nel pannello di navigazione, scegliere servizi AWS . Seleziona DevOps Agente, seleziona una quota e segui le istruzioni per richiedere un aumento della quota. Per ulteriori informazioni, consulta Richiesta di un aumento delle quote nella Guida per l'utente di Service Quotas.
- Dalla AWS CLI: utilizzare il comando CLI request-service-quota-increase AWS . Per ulteriori informazioni, consulta Richiesta di un aumento di quota nella Guida per l'utente di Service Quotas.

Cronologia dei documenti

La tabella seguente descrive le modifiche importanti alla documentazione dall'ultima versione di AWS DevOps Agent. Per ricevere notifiche sugli aggiornamenti di questa documentazione, è possibile iscriversi a un feed RSS.

Modifica	Descrizione	Data
Istruzioni per l'agente	Sono state aggiunte istruzioni globali e specifiche per l'agente (AGENTS.md) che si applicano a ogni sessione.	21 maggio 2026
Skip skill e status SKIPPED	Sono stati aggiunti l'esempio di abilità di filtraggio degli incidenti (salto della finestra di manutenzione), la decisione di triage SKIPPED, le istruzioni per correggere le decisioni di triage e l'evento Investigation Skipped. EventBridge	20 maggio 2026
Invio di file allegati	È stata aggiunta documentazione per allegare immagini, documenti e file di codice ai messaggi di chat. Ciò include i tipi di file supportati, i limiti e i casi d'uso.	19 maggio 2026
Assegnazione di priorità alle raccomandazioni	Aggiunta la classificazione dei AI-powered backlog, inclusa la personalizzazione delle priorità tramite chat e la stabilità del ranking.	13 maggio 2026
Plugin Claude Code per MCP	È stato aggiunto un riferimento al plug-in Claude Code	12 maggio 2026

Modifica	Descrizione	Data
	di esempio nella sezione di integrazione MCP.	
Barriere di autorizzazione	È stato aggiunto il modello guardrail della politica di sessione, che copre le autorizzazioni predefinite, le autorizzazioni aggiuntive supportate e le autorizzazioni bloccate dal guardrail.	7 maggio 2026
Nuovi IP statici	Aggiunti nuovi indirizzi IP statici per le connessioni in uscita in tutte le regioni supportate.	7 maggio 2026
Storia del documento	È stata aggiunta la pagina della cronologia dei documenti per tenere traccia della nuova documentazione.	5 maggio 2026
Interfacciamento con l'agente DevOps	È stata aggiunta la documenta zione per cinque metodi di accesso: app web, MCP, ACP, webhook e API.	28 aprile 2026
Convalida della conformità	Aggiunta una pagina dedicata alla convalida della conformità.	15 aprile 2026
Guida introduttiva all'uso AWS CloudFormation	È stata aggiunta una guida CloudFormation introduttiva.	29 marzo 2026
Connessione a strumenti ospitati privatamente	Documentazione aggiunta per le connessioni private.	29 marzo 2026

Modifica	Descrizione	Data
Endpoint dell'interfaccia VPC	Aggiunta la documentazione VPC endpoint ()AWS PrivateLink.	29 marzo 2026
EventBridge Integrazione con Amazon	È stata aggiunta una guida all' EventBridge integrazione per applicazioni basate sugli eventi.	28 marzo 2026
EventBridge riferimento ai dettagli degli eventi	È stato aggiunto un riferimento ai dettagli dell'evento per EventBridge l'integrazione.	28 marzo 2026
Quote	Aggiunta la pagina delle quote di servizio.	28 marzo 2026
Collegamento a Grafana	È stata aggiunta la documentazione sull'integrazione della telemetria Grafana.	27 marzo 2026
Connessione ad Azure	È stata aggiunta la documentazione sull'integrazione di Azure.	27 marzo 2026
Connessione delle risorse di Azure	Aggiunta la guida alla connessione di Azure Resources.	27 marzo 2026
Connessione ad Azure DevOps	Aggiunta la guida alla DevOps connessione ad Azure.	27 marzo 2026
Connessione PagerDuty	Aggiunta documentazione sull'integrazione della PagerDuty comunicazione.	27 marzo 2026

Modifica	Descrizione	Data
Migrazione da Public Preview a GA	È stata aggiunta la guida alla migrazione da Public Preview a General Availability.	27 marzo 2026
Disponibilità generale	Questa è la versione iniziale di AWS DevOps Agent a disponibilità generale.	30 marzo 2026

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.