Guía del usuario de

AWS DevOps Agente

AWS DevOps Agente: Guía del usuario de

Table of Contents

Acerca del AWS DevOps agente

AWS DevOps Agent es un agente fronterizo que resuelve y previene los incidentes de forma proactiva, lo que mejora continuamente la confiabilidad y el rendimiento.

AWS DevOps El agente investiga los incidentes e identifica las mejoras operativas como ingeniero experimentado DevOps .

El agente trabaja de la siguiente manera:

- Aprender sus recursos y sus relaciones.
- Trabaje con sus herramientas de observabilidad, sus habilidades, sus repositorios de código y CI/CD sus flujos de procesamiento.
- Correlaciona los datos de telemetría, código e implementación para comprender las relaciones entre los recursos de la aplicación.
- Compatible con aplicaciones en entornos multinube e híbridos.

Características principales de

AWS DevOps Agent proporciona capacidades integrales de respuesta y prevención de incidentes a través de las siguientes funciones:

Respuesta a incidentes autónoma y permanente

AWS DevOps El agente investiga los problemas de forma autónoma en el momento en que se producen:

- Investigación automática de incidentes: comienza a investigar inmediatamente cuando llega una alerta o un ticket de soporte
- AWS DevOps Chat con agentes: consulte su infraestructura, analice el estado del sistema y guíe las investigaciones utilizando un lenguaje natural a través de la aplicación web DevOps Agent Space. El chat proporciona respuestas contextuales en función de la página que esté consultando, ya sea para preguntar sobre los recursos en Topología, dirigir una investigación o filtrar las recomendaciones en Prevención.
- Planes de mitigación detallados: proporcionan acciones específicas para resolver los incidentes, validar el éxito y revertir los cambios si es necesario

- Coordinación automatizada de incidentes: distribuye las observaciones, los hallazgos y las medidas de mitigación a través de sus canales de comunicación preferidos, como Slack y ServiceNow
- AWS Integración de Support: cree casos de AWS Support directamente a partir de una investigación con un contexto inmediato proporcionado a los expertos de AWS Support

Prevención de futuros incidentes

AWS DevOps Agent analiza los patrones de los incidentes históricos para ayudarlo a pasar de una lucha contra incendios reactiva a una mejora operativa proactiva:

- Recomendaciones específicas: ofrece mejoras específicas y prácticas que refuerzan cuatro áreas clave: la observabilidad (monitoreo, alertas y registro), la optimización de la infraestructura (escalado automático, ajuste de la capacidad) y mejora de la canalización de despliegue (pruebas y validación).
- Aprendizaje continuo: refina las recomendaciones en función de los comentarios de tu equipo

Saque más partido a sus herramientas DevOps

AWS DevOps El agente se integra con sus herramientas actuales sin cambiar sus flujos de trabajo:

- Mapeo de los recursos de la aplicación: crea un gráfico topológico de los recursos de la aplicación y sus relaciones
- Integraciones integradas: funciona con herramientas de observabilidad populares (Amazon CloudWatch, Dynatrace, Datadog, New Relic y Splunk), repositorios de código y CI/CD canalizaciones (acciones y repositorios, flujos de trabajo y repositorios) GitHub GitLab
- Integración de herramientas personalizadas: amplíe las capacidades conectándose a sus propios servidores del Model Context Protocol (MCP) para obtener herramientas adicionales
- Consultas de infraestructura conversacional: utilice un lenguaje natural para consultar AWS los recursos, las métricas del sistema y el estado de las alarmas sin tener que navegar por varias consolas. Chat entiende el contexto y mantiene el historial de conversaciones para las preguntas de seguimiento.

Cómo funciona AWS DevOps Agent

AWS DevOps El agente funciona mediante una arquitectura de doble consola. Los administradores utilizan la consola AWS de administración para crear y administrar espacios de agentes, configurar las integraciones y configurar los controles de acceso. Los equipos de operaciones utilizan la aplicación web AWS DevOps Agent para las actividades de investigación y respuesta a day-to-day incidentes. La aplicación web es el lugar donde los operadores pueden interactuar con las investigaciones de los agentes, explorar la topología de las aplicaciones multicuentas y obtener información sobre las mejoras preventivas en la observabilidad, el código, los procesos y las arquitecturas de infraestructura. Para obtener más información, consulte the section called "Prevención proactiva de incidentes".

El servicio se organiza en torno a los espacios de agente, que son contenedores lógicos que definen los elementos a los que el agente puede acceder e investigar. AWS DevOps Cada espacio de agente contiene las configuraciones de su AWS cuenta, las integraciones de herramientas de terceros y los permisos de acceso. Para obtener más información, consulte the section called "¿Qué son los espacios de DevOps agentes?".

AWS DevOps Agent crea automáticamente una topología de aplicaciones que mapea sus recursos y sus relaciones. Esta topología ayuda al servicio a comprender la arquitectura de la aplicación durante las investigaciones. Para obtener más información, consulte the section called "¿Qué es una topología de DevOps agentes?".

Ventajas

- Reduzca el tiempo medio de resolución (MTTR): la investigación autónoma comienza de inmediato, lo que acelera la resolución de incidentes de horas a minutos
- Evite los incidentes recurrentes: las recomendaciones específicas abordan las causas fundamentales y refuerzan la resiliencia del sistema
- Mejore la eficiencia operativa: libere a su equipo de tareas de investigación repetitivas para centrarse en la innovación
- Trabaje dentro de los flujos de trabajo existentes: se integra con sus herramientas y procesos actuales sin interrupciones

¿Qué es una aplicación web para DevOps agentes?

AWS DevOps El agente utiliza una arquitectura de doble consola que separa las funciones administrativas de las actividades day-to-day operativas. Este diseño permite a los administradores configurar el servicio mientras los equipos de operaciones se centran en la respuesta y la prevención de incidentes.

Consolas

AWS DevOps El agente proporciona dos interfaces distintas:

- AWS Consola de administración: los administradores usan la consola de AWS administración para configurar y administrar el AWS DevOps agente. En esta consola, puede the section called "Creación de un espacio de agentes" conectar AWS servicios y herramientas de terceros y administrar los permisos de acceso de su organización.
- DevOps Aplicación web para agentes: los equipos de operaciones utilizan las aplicaciones web de DevOps Agent Space para las actividades diarias de respuesta a incidentes. Esta aplicación independiente proporciona una interfaz en la que los ingenieros de guardia pueden iniciar investigaciones, interactuar con el agente a través de un chat en lenguaje natural, ver las topologías de las aplicaciones y revisar las recomendaciones de prevención de incidentes.

Capacidades de la aplicación web

La aplicación web DevOps Agent ofrece las siguientes funciones principales:

- Respuesta a incidentes: en esta página se crean investigaciones de incidentes y se hace un seguimiento de ellas, así como se generan planes de mitigación para resolverlos.
- Prevención de incidentes: en la página de prevención, encontrará recomendaciones para mejorar su postura de observabilidad, sus procesos de entrega y su arquitectura de infraestructura para evitar futuros incidentes.
- Topología: la página de topología proporciona una representación visual interactiva de los recursos de la cuenta y sus relaciones en todos los recursos de las cuentas conectadas. Puede ver la topología con distintos niveles de detalle mediante el menú desplegable «Mostrar» para cambiar entre las vistas de sistema, de contenedor y de recursos.
- Habilidades: conjuntos de instrucciones modulares que amplían las funciones de AWS DevOps Agent con funciones especializadas. Las habilidades incluyen conocimientos del campo, metodologías de investigación y configuraciones de herramientas adaptadas a su infraestructura.

Cada habilidad habilita herramientas específicas y proporciona una divulgación progresiva de las instrucciones solo cuando son relevantes para la investigación.

- Interfaz de chat en lenguaje natural: disponible en toda la aplicación web, Chat es un asistente de conversación basado en inteligencia artificial que le permite consultar su infraestructura, analizar el estado del sistema y trabajar con las investigaciones utilizando un lenguaje natural. El chat proporciona respuestas contextuales en función de la página que estés viendo.

Autenticación

AWS DevOps El agente admite métodos de autenticación flexibles para adaptarse a los diferentes requisitos organizativos:

- Integración del IAM Identity Center (acceso de usuario): las organizaciones pueden usar AWS Identity Center (IAM Identity Center) para administrar de forma centralizada el acceso de los usuarios a las aplicaciones web de DevOps Agent Space. IAM Identity Center puede federarse con proveedores de identidad externos a través de los protocolos OIDC y SAML estándar, incluidos proveedores como Okta, Ping Identity y Microsoft Entra ID. Este método admite la autenticación multifactorial de su proveedor de identidad.
- Autenticación de proveedor de identidad externo (IdP): las organizaciones pueden conectar un proveedor de identidad compatible con OIDC, como Okta o Microsoft Entra ID, directamente a la aplicación web Agent Space sin necesidad de IAM Identity Center. Los usuarios inician sesión con sus credenciales corporativas a través del IdP. Para obtener instrucciones de configuración, consultethe section called “Configuración de la autenticación de un proveedor de identidad externo (IdP)”.
- Enlace de autenticación de IAM (acceso de administrador): un método alternativo proporciona acceso directo a la aplicación web desde la consola de AWS administración mediante la sesión de consola existente. Esta opción resulta útil antes de implementar la integración completa de Identity Center, pero las sesiones se limitan a 10 minutos.

¿Qué son los espacios de DevOps agentes?

Un espacio de DevOps agente es un contenedor lógico que define las herramientas y la infraestructura a las que tiene acceso AWS DevOps un agente. Cada espacio de agente funciona de forma independiente con su propio acceso a la AWS cuenta, integraciones de terceros y permisos de usuario.

Un espacio de agente representa el límite a lo que el AWS DevOps agente puede acceder e investigar durante la respuesta a un incidente. Al crear un espacio de agentes, se definen a qué AWS cuentas puede acceder el agente, a qué herramientas externas se puede conectar y qué usuarios de la organización pueden interactuar con el agente.

Cada espacio de agente funciona como un despliegue independiente de AWS DevOps Agent. El Agent Space se configura a través de la consola de AWS administración, mientras que los equipos de operaciones utilizan la aplicación web del Agent Space para llevar a cabo investigaciones y revisar las recomendaciones dentro de ese espacio.

Cómo se aíslan los espacios de agentes

Los espacios de agentes mantienen el aislamiento para garantizar la seguridad y evitar el acceso no deseado entre diferentes entornos o equipos:

- AWS aislamiento de cuentas: cada espacio de agente utiliza funciones de IAM específicas que solo permiten el acceso a AWS cuentas y recursos específicos. El agente no puede acceder a AWS recursos distintos de los configurados explícitamente para el espacio de agentes.
- Aislamiento del acceso de los usuarios: usted controla qué usuarios o grupos pueden acceder a cada espacio de agente. Esto le permite alinear los permisos de acceso con su estructura organizativa, lo que garantiza que los equipos solo interactúen con los espacios de agente designados.
- Aislamiento de datos: los datos de investigación, el historial de incidentes y las recomendaciones se mantienen por separado en cada espacio de agentes. La información de un espacio de agente no es visible ni accesible desde otro espacio de agente.
- Aislamiento de los datos de chat: el historial de conversaciones de chat también está aislado dentro de cada espacio de agente. Las conversaciones y consultas de un espacio de agente no son visibles ni accesibles desde otro espacio de agente.

Aplicación web Agent Space

Cada Agent Space tiene una aplicación web dedicada a la que se puede acceder desde la consola de AWS administración. Consulte the section called “¿Qué es una aplicación web para DevOps agentes?” para obtener más información sobre la aplicación web.

Cuándo usar varios espacios de agente

Considere la posibilidad de crear varios espacios de agentes para satisfacer las diferentes necesidades de la organización:

- Separación de equipos: cree espacios de agentes exclusivos para diferentes equipos de aplicaciones o unidades de negocio a fin de mantener claros los límites de propiedad en el espacio de agentes.
- Aislamiento del entorno: separe los entornos de producción y los que no son de producción en diferentes espacios de agentes para evitar el acceso accidental entre entornos.
- Límites de servicio: alinee los espacios de los agentes con los límites de los servicios o aplicaciones específicos para mantener las investigaciones centradas y relevantes.
- Requisitos de conformidad: configure espacios de agentes independientes con diferentes controles de acceso o ajustes de residencia de datos para cumplir con los requisitos reglamentarios.

> **Note**
>
> Al crear varios espacios de agente, puede utilizar una AWS cuenta dedicada como cuenta principal para un espacio de agente y conectar distintas cuentas de aplicaciones como cuentas secundarias. Este enfoque le permite mantener controles de acceso detallados y, al mismo tiempo, garantizar que cada espacio de agente pueda acceder solo a los recursos específicos para su ámbito previsto, incluso cuando se utiliza la creación automática de roles.

¿Qué es una topología de DevOps agentes?

AWS DevOps Agent's descubre y visualiza automáticamente los recursos y las relaciones dentro de sus aplicaciones y utiliza la topología resultante para comprender su infraestructura durante la investigación de incidentes y al hacer recomendaciones preventivas.

Cómo se crean los gráficos de topología

AWS DevOps El agente crea gráficos de topología mediante varios procesos automatizados:

- Descubrimiento de recursos: el agente escanea automáticamente sus AWS cuentas para identificar los recursos, como las instancias de procesamiento, los servicios de almacenamiento, los componentes de red y las bases de datos, que forman parte de sus aplicaciones.

- Detección de relaciones: el agente analiza los datos de configuración, las CloudFormation pilas y las etiquetas de los recursos para determinar cómo se relacionan los recursos entre sí.
- Mapeo de código e implementación: cuando se conecta a CI/CD canalizaciones, el agente vincula los recursos de infraestructura a sus procesos de implementación y modifica el código de la aplicación y la infraestructura.
- Mapeo del comportamiento de observabilidad: los datos de los sistemas de observabilidad, como Amazon CloudWatch Application Signals y Dynatrace, se utilizan para identificar los comportamientos observados que indican relaciones entre los recursos.

Capacidades clave

El mapeo de recursos proporciona varias capacidades que mejoran la investigación y la prevención de incidentes:

- Visualización interactiva: explore la topología de su aplicación a través de un gráfico interactivo en la aplicación web Operator. Puede hacer zoom y navegar por la topología para comprender las complejas relaciones entre los recursos. También puede usar Chat para consultar información de topología mediante un lenguaje natural, como «Mostrar todas las funciones de Lambda conectadas a esta tabla de DynamoDB» o «¿Qué recursos se ven afectados por esta alarma?».
- Investigación contextual: durante la investigación de los incidentes, el AWS DevOps agente cuenta con la ayuda de la topología de los recursos para identificar los componentes afectados, comprender el radio de la explosión y trazar la trayectoria del impacto a través de sus sistemas.
- Análisis de la causa raíz: la comprensión detallada de las relaciones entre los recursos ayuda a determinar dónde se originan los problemas, incluso en sistemas distribuidos complejos con muchas interdependencias.
- Evaluación del impacto: al analizar los incidentes, el agente puede determinar mejor qué servicios descendentes podrían verse afectados identificando las cadenas de dependencia en la topología.
- Recomendaciones preventivas: el agente utiliza la información de la topología para hacer recomendaciones específicas para mejorar la resiliencia, sugiriendo los cambios que tendrán el impacto más significativo en la estabilidad del sistema.

Vistas de topología

La visualización de la topología de la página de topología de la aplicación web Operator ofrece varios niveles de detalle:

- Aprendido: la vista predeterminada, generada a partir de la habilidad Agent Space Understanding. Muestra un resumen estructurado de su infraestructura organizado por servicios lógicos y rutas de solicitud.
- Sistema: muestra los límites de las cuentas y regiones de alto nivel.
- Contenedor: muestra las pilas de implementación, como si CloudFormation fueran pilas que contienen recursos relacionados.
- Componentes: muestra los componentes individuales de los contenedores y sus relaciones.
- Todos los recursos: muestra una vista completa con todos los recursos descubiertos y sus relaciones.

Descubrimiento de recursos

Los recursos se descubren mediante dos métodos:

- CloudFormation pilas: el agente muestra todas las CloudFormation pilas y sus recursos en la AWS cuenta principal y en cualquier cuenta secundaria conectada. Esto es compatible con cualquier infrastructure-as-code herramienta que se utilice CloudFormation para la implementación, incluido el AWS Cloud Development Kit (AWS CDK).
- Explorador de recursos: en el caso de los recursos desde los que no se implementaron CloudFormation, los recursos etiquetados se detectan en el Explorador de AWS recursos. La AWS cuenta de destino debe tener activado el Explorador de recursos. Esto resulta útil para identificar los límites de las aplicaciones para los recursos desplegados a través de la consola de AWS administración APIs, el AWS servicio u otros infrastructure-as-code marcos.

El ámbito de investigación va más allá de la topología

Si bien la topología de la aplicación proporciona un contexto importante durante las investigaciones, AWS DevOps Agent no se limita a investigar únicamente los recursos que se muestran en la topología. El agente puede utilizar fuentes de datos adicionales, como herramientas de observabilidad conectadas APIs o de AWS servicios, para investigar los recursos que no se encuentran en la topología de la aplicación.

Para limitar los recursos a los que tiene acceso el agente, restrinja la política de acceso de la función asignada al agente a los recursos multicuentas. Para obtener más información, consulte the section called “Limitar el acceso de los agentes a una AWS cuenta”.

Habilidad para comprender la topología y el espacio de agentes

El gráfico de topología contribuye a la habilidad aprendida de Agent Space Understanding, que codifica un resumen estructurado de su infraestructura para utilizarlo durante las investigaciones. Cuando se completa el descubrimiento de la topología de un nuevo espacio de agentes, el sistema genera automáticamente la habilidad de comprensión del espacio de agentes. Para obtener más información sobre las habilidades adquiridas, consultethe section called “Habilidades aprendidas”.

DevOps Habilidades de agente

AWS DevOps Las habilidades de los agentes son conjuntos de instrucciones modulares que amplían las capacidades del agente con conocimientos de dominio especializados y metodologías de investigación adaptadas a su infraestructura y sus flujos de trabajo operativos.

¿Qué son las habilidades

Las habilidades son directorios independientes que contienen instrucciones de Markdown que proporcionan capacidades especializadas a AWS DevOps Agent. AWS DevOps Agent admite un subconjunto de la especificación Agent Skills, un estándar abierto para empaquetar instrucciones y recursos relacionados con los agentes, y solo admite documentos no ejecutables: instrucciones, imágenes y archivos de datos de Markdown. PDFs

Cada habilidad requiere un archivo Skill.md que contenga las instrucciones que desee proporcionar a su agente. AWS DevOps Además del archivo Skill.md requerido, las habilidades pueden incluir:

- Flujos de trabajo de investigación para escenarios o tipos de infraestructura específicos.
- Materiales de referencia que incluyen patrones de arquitectura y procedimientos operativos.
- Segmentación por tipo de agente: las habilidades se pueden orientar a tipos de agentes específicos (genéricos, bajo demanda, clasificación de incidentes, RCA de incidentes, mitigación de incidentes, evaluación) para reducir el consumo de contexto y mejorar la concentración de los agentes.

¿Por qué usar Skills

Las habilidades transforman a AWS DevOps Agent de un asistente de uso general en un especialista para sus flujos de trabajo operativos y de infraestructura. A diferencia de las instrucciones únicas que

se proporcionan en un mensaje de chat, las habilidades son capacidades reutilizables que se cargan automáticamente cuando son relevantes para las tareas realizadas por AWS DevOps el agente.

Ventajas clave:

- Especialice a su agente: personalice a AWS DevOps Agent con procedimientos de investigación, mejores prácticas y conocimientos organizativos específicos para su infraestructura y patrones operativos.
- Reduzca la repetición: cree flujos de trabajo de investigación de una sola vez y AWS DevOps Agent los utilizará automáticamente en todas las investigaciones relevantes, lo que eliminará la necesidad de proporcionar la misma orientación repetidamente.
- Capacidades de redacción: combine varias habilidades para crear flujos de trabajo de end-to-end investigación. AWS DevOps El agente aprende varias habilidades durante la ejecución, como una habilidad para recuperar las implementaciones de su CI/CD proceso personalizado y una habilidad para buscar en sus repositorios de código.
- Amplify las herramientas personalizadas: cree habilidades que guíen a AWS DevOps Agent a utilizar sus herramientas de servidor MCP personalizadas de manera eficaz. Las habilidades permiten documentar cuándo invocar herramientas específicas, qué parámetros usar en diferentes escenarios y cómo interpretar los resultados para lograr flujos de trabajo específicos para su infraestructura.

Cómo funcionan las habilidades

Cuando el AWS DevOps agente encuentra una tarea relevante, adquiere las habilidades adecuadas y sigue las instrucciones para guiar su investigación. Por ejemplo, una habilidad de «investigación del rendimiento de una base de datos» podría incluir step-by-step procedimientos para analizar los problemas de regulación del RDS, lo que permitiría al agente comprobar sistemáticamente el estado de las alarmas, analizar las métricas de conexión e identificar las consultas lentas.

Estructura de habilidades

Una habilidad se organiza como un directorio que contiene:

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

Skill.md

`SKILL.md`Es el único archivo obligatorio. Contiene las instrucciones básicas escritas en formato Markdown. Este archivo debe:

- Describa cuándo y cómo usar la habilidad.
- Proporcione los procedimientos de step-by-step investigación.
- Incluya árboles de decisión para diferentes escenarios.
- Documente los resultados esperados y los criterios de éxito.

Frontmatter

Frontmatter es el bloque de metadatos situado en la parte superior de un `SKILL.md` archivo, encerrado entre `---` delimitadores. Contiene los `description` campos `name` y campos que el AWS DevOps agente utiliza para determinar cuándo activar la habilidad durante una investigación o tarea.

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

nombre: un identificador único de la habilidad. Utilice únicamente letras minúsculas, números y guiones (64 caracteres como máximo). No debe empezar ni terminar con un guión.

descripción: una explicación detallada de cuándo y por qué el AWS DevOps agente debe usar esta habilidad. AWS DevOps El agente evalúa este campo para decidir si la habilidad es relevante para la tarea actual. Una descripción vaga o faltante puede provocar que el agente se salte la habilidad por completo, incluso si las instrucciones están bien redactadas.

Importante: escribe la descripción desde la perspectiva del agente. Incluya los escenarios, los servicios, los tipos de error o los síntomas específicos que deberían activar la habilidad. Por ejemplo, «Utilice esta habilidad para investigar la latencia de la base de datos, los errores de conexión o los tiempos de espera de las consultas para las instancias de Amazon RDS» es más eficaz que «habilidad RDS».

Al crear una habilidad en la interfaz de usuario, el sistema genera información preliminar automáticamente a partir del nombre y la descripción que usted proporciona. Las habilidades subidas como archivos zip deben incluir frontmatter en el `SKILL.md` archivo.

Ejemplo: habilidad completa

El siguiente ejemplo muestra una habilidad completa y bien formada para investigar los problemas de rendimiento del RDS. Muestra la estructura de directorios, la información preliminar de Skill.md, los procedimientos de investigación procesables y un archivo de referencias complementario.

Estructura de directorios:

```
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
```

referencias/ .mdrds-metrics-reference:

```
# 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% |
```

Creando habilidades

Antes de crear habilidades, debes tener un espacio de agente. Para obtener más información, consulte the section called “Creación de un espacio de agentes”.

Puede crear habilidades de dos maneras, en función de sus preferencias de flujo de trabajo y de la complejidad de las habilidades:

Crear una habilidad en la interfaz de usuario

Las habilidades creadas en la aplicación web AWS DevOps Agent Operator contienen un nombre, una descripción e instrucciones en un único archivo Skill.md.

Para crear una habilidad en la interfaz de usuario:

- Diríjase a la página de habilidades de su aplicación web Agent Space Operator.
- Haga clic en «Añadir habilidad».
- Selecciona «Crear habilidad» en el modal.
- Rellena el formulario de habilidades:
  - Nombre: solo letras minúsculas, números y guiones (máximo 64 caracteres). No debe empezar ni terminar con un guión. Ejemplo: `rds-throttling-investigation`
  - Descripción: breve explicación de cuándo usar esta habilidad (se recomienda un mínimo de 100 caracteres y un máximo de 1024 caracteres). Esto ayuda al agente a determinar cuándo activar la habilidad.
  - Estado: se establece en Activo (predeterminado) o Inactivo. El agente no utiliza las habilidades inactivas.
  - Tipo de agente: seleccione uno o más tipos de agentes que puedan utilizar esta habilidad. La opción Genérica está seleccionada de forma predeterminada y hace que la habilidad esté disponible para todos los tipos de agentes. Para dirigirse a agentes específicos, deseleccione la opción Genérico y elija entre: Bajo demanda, Triaje de incidentes, RCA de incidentes, Mitigación de incidentes o Evaluación.
  - Instrucciones: Step-by-step procedimientos en formato Markdown. Sea específico y práctico.
- Haz clic en «Crear» para guardar la habilidad.

El sistema genera automáticamente un archivo Skill.md con la estructura frontal adecuada.

Para editar una habilidad creada en la interfaz de usuario:

- Navegue hasta la habilidad en la lista de habilidades y haga clic en ella para abrirla.
- Haga clic en Edit.
- Modifique el nombre, la descripción o las instrucciones.

- Haga clic en Guardar para actualizar la habilidad.

Cargar una habilidad

Las habilidades subidas como archivos zip contienen un archivo Skill.md además de recursos adicionales, como materiales de referencia o activos.

Estructura de habilidades:

```
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
```

Requisitos de experiencia previa de Skill.md:

Las habilidades subidas como archivos zip deben incluir FrontMatter en Skill.md con los campos y. `name` `description` AWS DevOps El agente usa estos campos para determinar cuándo activar la habilidad. Para obtener más información sobre cómo escribir un material preliminar efectivo, consulte la sección Frontmatter que aparece al principio de este tema.

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

Para crear una habilidad subiéndola a un archivo zip, sigue estos pasos:

- Cree un directorio con sus archivos de habilidades siguiendo la estructura anterior.

- Asegúrese de que Skill.md incluya la información preliminar adecuada (nombre y descripción).
- Comprima el directorio en un archivo.zip.
- Diríjase a la página de habilidades de su aplicación web Agent Space Operator.
- Haga clic en «Añadir habilidad».
- Selecciona «Cargar habilidad» en el modal.
- Arrastra y suelta tu archivo.zip o haz clic para buscarlo (solo archivos ZIP, máximo 6 MB).
- Seleccione uno o más tipos de agentes que puedan utilizar esta habilidad (la opción Genérica está seleccionada de forma predeterminada y se aplica a todos los tipos de agentes; deseleccione esta opción para centrarse específicamente en la opción Bajo demanda, la clasificación de incidentes, la RCA de incidentes, la mitigación de incidentes o la evaluación).
- Revise los requisitos del archivo zip y los resultados de la validación.
- Haga clic en «Cargar» para añadir la habilidad a su espacio de agente.

Restricciones importantes para las habilidades que se suben como archivos zip:

- Actualmente no se admiten scripts: las habilidades que contengan scripts en el `scripts/` directorio se rechazarán durante la carga. La ejecución de scripts se habilitará en una versión futura una vez que los agentes tengan acceso a un entorno de codificación seguro.
- Límite de tamaño: el tamaño total del archivo zip no debe superar los 6 MB (incluidos todos los archivos).
- Se requiere Skill.md: el archivo zip debe contener un archivo Skill.md con una portada válida.

Mejores prácticas para las habilidades de nomenclatura:

Utilice nombres descriptivos y claros, como «rds-throttling-investigation», en lugar de nombres genéricos. Un buen nombre de habilidad refleja el escenario o servicio específico al que se dirige, lo que facilita la identificación de la habilidad adecuada de un vistazo.

Administración de Skills

AWS DevOps Agent ofrece capacidades integrales de gestión de habilidades a través de la aplicación web Operator:

Publica tus habilidades: consulta todas las habilidades en tu espacio de agente. La página de habilidades muestra el nombre de la habilidad, su estado activo o inactivo, la fecha de creación, la fecha de la última actualización y las acciones disponibles.

Visualización de habilidades: haga clic en cualquier habilidad para ver su vista detallada. Las habilidades creadas en la interfaz de usuario muestran contenido editable, donde puedes modificar el nombre, la descripción o las instrucciones directamente en la interfaz de usuario y hacer clic en «Guardar» para actualizarlas. Las habilidades subidas como archivos zip muestran un árbol de archivos que muestra Skill.md y cualquier directorio adicional, como references/ y assets/. Haga clic en los archivos del árbol para ver su contenido en modo de solo lectura.

Selección de agentes para una habilidad: configure qué tipos de agentes pueden usar cada habilidad al crearla o editarla. En el menú desplegable Tipo de agente, seleccione uno o más tipos de agentes mediante las casillas de verificación: Genérico (predeterminado, se aplica a todos los tipos de agentes), Bajo demanda (consultas conversacionales), Triaje de incidentes (evaluación inicial de incidentes), RCA de incidentes (análisis de causa raíz), Mitigación de incidentes (respuesta automática a incidentes) o Evaluación (recomendaciones proactivas). La opción Genérica está seleccionada de forma predeterminada y pone la habilidad a disposición de todos los tipos de agentes. Las habilidades dirigidas a agentes específicos reducen el consumo de contexto y mejoran la concentración de los agentes.

Activación y desactivación de habilidades: desactiva temporalmente las habilidades sin eliminarlas con el botón Active/Inactive . Abre la vista detallada de las habilidades y coloca el interruptor en «Inactivo» para evitar que el agente las cargue para nuevas investigaciones y, al mismo tiempo, conservar todo el contenido y las configuraciones. Las investigaciones en curso siguen utilizando la habilidad. Vuelve a «Activa» para que la habilidad vuelva a estar disponible inmediatamente.

Actualización de habilidades: modifique las habilidades existentes en función de cómo se crearon. En el caso de las habilidades creadas en la interfaz de usuario, haga clic en «Editar» en la vista de detalles de la habilidad, modifique el nombre, la descripción o las instrucciones y, a continuación, haga clic en «Guardar» para actualizarlas. En el caso de las habilidades subidas como archivos zip, modifique los archivos localmente, cree un nuevo archivo zip y cargue una nueva versión.

Eliminar habilidades: elimina permanentemente las habilidades de tu espacio de agente. Abre la vista de lista de habilidades, haz clic en el menú de más opciones (⊙) y selecciona «Eliminar», lee la advertencia sobre la eliminación permanente, escribe el nombre de la habilidad para confirmarla y haz clic en «Eliminar habilidad». La eliminación no se puede deshacer. Las investigaciones en curso pueden verse afectadas si intentan cargar la habilidad eliminada. Para las habilidades subidas como archivos zip, descarga el archivo zip antes de eliminarlas como copia de seguridad. Considera la posibilidad de desactivar la habilidad en lugar de eliminarla si es posible que la vuelvas a necesitar.

¿Migrar desde Runbooks

Los Runbooks existentes se migran automáticamente a Skills sin que el cliente tenga que hacer nada al respecto. Cuando tu espacio de agente pasa al modelo de habilidades, todos los cuadernos de runbooks se convierten en habilidades y aparecen en tu interfaz de usuario de habilidades. Tras la migración, podrás:

- Revisa las habilidades migradas: comprueba que la migración automática haya convertido correctamente tus Runbooks.
- Actualízalas según sea necesario: edita las habilidades directamente en la interfaz de usuario para refinar las instrucciones, actualizar las descripciones o configurar la segmentación por tipo de agente.
- Amplíe las habilidades con referencias: si necesita materiales de referencia o diagramas de arquitectura adicionales, vuelva a crearlas como habilidades para cargarlas en formato zip con un directorio references/ o assets/.
- Cree nuevas habilidades: añada nuevas habilidades para los flujos de trabajo de investigación que anteriormente no se trataban en Runbooks.

Póngase en contacto con AWS Support si tiene algún problema con las habilidades que se migran automáticamente o si necesita ayuda con las actualizaciones posteriores a la migración.

Habilidades aprendidas

¿Qué son las habilidades aprendidas?

Las habilidades aprendidas son archivos de conocimiento estructurados que el DevOps agente genera a partir de los datos de Agent Space. Cada habilidad aprendida codifica un tipo específico de conocimiento que el AWS DevOps agente utiliza al realizar las tareas. En el momento del lanzamiento, están disponibles dos habilidades aprendidas: la comprensión del espacio de agente y las mejores prácticas de uso de herramientas.

Comprensión de Agent Space

La habilidad Agent Space Understanding (`understanding-agent-space`) analiza las cuentas en la nube conectadas, los repositorios de código y las integraciones de telemetría para crear un mapa de los recursos y las relaciones de un Agent Space.

La habilidad produce un `SKILL.md` archivo principal y un conjunto de archivos de referencia. El archivo principal contiene una descripción general del sistema en un lenguaje sencillo con los conceptos clave del dominio, los entornos de implementación (pares de AWS cuentas y regiones, suscripciones y regiones de Azure, etc.), un diagrama de arquitectura a nivel de contenedor que muestra cómo se conectan los servicios lógicos, las rutas de solicitud que son fundamentales para la aplicación con los componentes que atraviesan y un mapeo de los repositorios de código a los contenedores.

Cada contenedor lógico recibe un archivo de referencia dedicado que describe sus componentes internos (procesamiento, datos, mensajería, red y otros) con tipos de recursos e identificadores físicos ARNs, como nombres de tablas y colas URLs. El archivo de referencia también captura la cobertura de observabilidad, incluidas las alarmas, los paneles y los monitores vinculados a cada componente. También mapea cada componente con sus repositorios de código, paquetes y infrastructure-as-code definiciones asociados, lo que proporciona una cadena de trazabilidad completa desde el código fuente hasta los recursos implementados.

Cada ruta de solicitud crítica recibe un archivo de referencia específico que describe el flujo completo de end-to-end solicitudes según la granularidad de los componentes, desde el punto de entrada hasta cada servicio intermedio, almacén de datos y dependencia externa. El archivo incluye un diagrama de flujo secuenciado que muestra el orden de las operaciones y los mecanismos de interacción entre los componentes, junto con la responsabilidad de cada participante. También cataloga las señales de observabilidad relevantes para la ruta: patrones de grupos de registros para cada salto, métricas clave (latencia, tasas de error, regulación, cuotas simbólicas) con sus nombres y dimensiones de alarma, y tramos de rastreo distribuidos que se pueden correlacionar entre servicios y cuentas.

Mejores prácticas de uso de herramientas

La habilidad sobre las mejores prácticas de uso de herramientas analiza los usos de las herramientas investigadas en el pasado para extraer patrones de uso efectivos, modos de falla comunes y orientación sobre parámetros. Esto ayuda al DevOps agente a evitar los errores conocidos y a llevar a cabo las investigaciones con menos pasos desperdiciados. La habilidad produce un archivo principal y un conjunto de archivos de referencia por herramienta. El archivo principal sirve como un índice de enrutamiento que enumera cada herramienta con los escenarios de investigación en los que se basa y enlaza con el archivo de referencia correspondiente.

Cada archivo de referencia por herramienta puede incluir hasta tres secciones:

- Mejores prácticas: técnicas basadas en la investigación extraídas del uso exitoso de la herramienta, como las plantillas de consultas de CloudWatch Logs Insights, los espacios de nombres y dimensiones de las métricas específicos del entorno y los filtros de fuentes de eventos. CloudTrail Cada entrada se organiza en torno a un escenario de investigación e incluye valores de parámetros concretos y ejemplos observados en investigaciones anteriores.
- Errores comunes: modos de falla recurrentes y sus correcciones. Cada entrada describe una condición de error específica, como consultar una cuenta inaccesible o crear una consulta de agregación con un formato incorrecto, y proporciona una acción correctiva para que el agente pueda evitar el error o recuperarse del mismo sin desperdiciar los pasos de investigación.
- Gestión de resultados: orientación para las llamadas a las herramientas que suelen arrojar grandes respuestas. Cada entrada describe un cambio de parámetro o una estrategia de procesamiento que reduce el tamaño de la salida y, al mismo tiempo, preserva el valor de diagnóstico.

Cuando está disponible el acceso a la infraestructura en tiempo real, la habilidad valida los patrones en función del entorno antes de incluirlos. Los patrones confirmados se expresan con confianza, los patrones no confirmados utilizan un lenguaje cauteloso y los patrones refutados se excluyen. Esto mantiene la habilidad alineada con el estado actual de su infraestructura.

Gestión de las habilidades aprendidas

Actualizaciones: el DevOps agente genera y actualiza automáticamente las habilidades aprendidas en función de la actividad en su espacio de agente. A continuación se describe cuándo se actualiza cada habilidad.

El DevOps agente genera una habilidad actualizada sobre las mejores prácticas de uso de herramientas cada 30 investigaciones.

El agente de aprendizaje genera la habilidad de comprensión de Agent Space y se ejecuta cada vez que añades, actualizas o eliminas una capacidad o integración de Agent Space.

Para regenerar manualmente las habilidades adquiridas, pulsa el botón Regenerar en la página de topología de la aplicación del operador o chatea con el agente y pídele que actualice las habilidades adquiridas.

Desactivación: las habilidades aprendidas están activas de forma predeterminada. Cuando están activas, el DevOps agente las carga al inicio de cada tarea del DevOps agente. Para evitar que se aplique una habilidad aprendida, desactívela en el visor de habilidades de la aplicación del operador. La desactivación de una habilidad no la elimina. La habilidad se conserva y se puede reactivar

en cualquier momento. Cuando se desactiva una habilidad, el DevOps agente actúa sin que esa habilidad lo sepa.

Vista de topología: la página de topología de la aplicación web de Agent Space utiliza la habilidad de comprensión de Agent Space para mostrar visualmente el entorno de Agent Space como contenedores y componentes lógicos. Haga clic en cualquier contenedor para ver sus componentes, identificadores de recursos y telemetría.

Regiones admitidas

En este tema se describen las AWS regiones en las que puede usar AWS DevOps Agent. Para obtener más información sobre AWS las regiones, consulte Especificar qué AWS regiones puede usar su cuenta en la Guía de referencia de administración de AWS cuentas.

Supervisión de recursos entre regiones

AWS DevOps El agente puede supervisar e investigar los recursos de AWS las cuentas ubicadas en cualquier AWS región, independientemente de la región compatible en la que cree su espacio de agente. Al asociar una AWS cuenta a un espacio de agentes, el agente descubre y mapea los recursos de todas las regiones de esa cuenta. Esto significa que no necesita un espacio de agente en cada región en la que se ejecuten sus cargas de trabajo.

Elija una región compatible en función de la residencia de datos que prefiera, la proximidad a su equipo de operaciones o los requisitos organizativos.

Regiones admitidas

AWS DevOps El agente está disponible en las siguientes AWS regiones.

Nombre de la región	Código de región	Enlace a la consola
Este de EE. UU. (Norte de Virginia)	us-east-1	Abra la consola
Oeste de EE. UU. (Oregón)	us-west-2	Abra la consola
Asia-Pacífico (Sídney)	ap-southeast-2	Abra la consola
Asia-Pacífico (Tokio)	ap-northeast-1	Abra la consola

Nombre de la región	Código de región	Enlace a la consola
Europa (Fráncfort)	eu-central-1	Abra la consola
Europa (Irlanda)	eu-west-1	Abra la consola

Puntos de conexión de servicio

Nombre de la región	Código de región	Punto de conexión	Protocolo
Este de EE. UU. (Norte de Virginia)	us-east-1	aidevops.us-east-1.amazonaws.com	HTTPS
Oeste de EE. UU. (Oregón)	us-west-2	aidevops.us-west-2.amazonaws.com	HTTPS
Asia-Pacífico (Sídney)	ap-southeast-2	aidevops.ap-southeast-2.amazonaws.com	HTTPS
Asia-Pacífico (Tokio)	ap-northeast-1	aidevops.ap-northeast-1.amazonaws.com	HTTPS
Europa (Fráncfort)	eu-central-1	aidevops.eu-central-1.amazonaws.com	HTTPS
Europa (Irlanda)	eu-west-1	aidevops.eu-west-1.amazonaws.com	HTTPS

Consideraciones

- Selección de una región de Agent Space: un Agent Space y sus datos (investigaciones,

la topología, las recomendaciones) se almacenan en la región en la que se creó. Elija una región que cumpla con sus requisitos de residencia de datos.

- Supervisión interregional: recursos en AWS las cuentas asociadas a un agente

El espacio se supervisa independientemente de la región en la que estén desplegados esos recursos. No es necesario crear espacios de agentes independientes en cada región en la que se ejecuten las cargas de trabajo.

- Integraciones de terceros: conexiones con CI/CD proveedores (GitHub,), GitLab

Las herramientas de observabilidad (Dynatrace, Datadog, New Relic, Splunk) y los servidores MCP se configuran por espacio de agente y no dependen de la región.

Cómo empezar con AWS DevOps Agent

En esta guía de introducción, crearás un espacio de agente básico, configurarás los permisos mínimos y realizarás tu primera investigación basada en la IA.

Temas:

Creación de un espacio de agentes

Un espacio de agentes define las herramientas y la infraestructura a las que tiene acceso el AWS DevOps agente. Esta guía explica cómo crear un espacio de agentes, configurar el acceso a la cuenta principal y habilitar la aplicación web del DevOps agente. Consulte «Qué es un espacio de agentes» para obtener más información sobre el concepto de espacio de agentes.

Creación de un espacio de agentes

Acceda a la consola AWS DevOps de agentes

1. Inicie sesión en la consola AWS de administración
2. Navegue hasta la consola del AWS DevOps agente

Asigne un nombre al espacio de agentes

1. Haga clic en Crear espacio de agente

En la sección de detalles del espacio de agentes, proporcione:

1. En el campo Nombre, introduce un nombre para tu espacio de agente
2. (Opcional) En el campo Descripción, añada detalles sobre el propósito del espacio de agentes
3. (Opcional) En el menú desplegable del idioma de respuesta del agente, seleccione el idioma que utiliza el agente al generar las respuestas, los hallazgos y los resultados de la investigación. Las opciones incluyen: bahasa indonesio, chino (Simplified/PRC), Chinese (Traditional/Taiwan), inglés (Reino Unido), francés (Francia), alemán (Alemania), italiano (Italia), japonés (Japón), coreano (Corea), portugués (Brasil), español (Latinoamérica), turco (Turquía), árabe (Arabia Saudita), tailandés (Tailandia) y vietnamita (Vietnam). Si no se selecciona ningún idioma, el agente responde en el idioma de la entrada.

Configurar el acceso a la cuenta principal

En la sección Conceder acceso a este espacio de agente a los AWS recursos, configurará una función de IAM para conceder al espacio de agente acceso a la AWS cuenta principal. La cuenta principal es la AWS cuenta en la que crea su espacio de agente. AWS DevOps El agente necesita una función de IAM para descubrir los AWS recursos de esta cuenta y acceder a ellos durante las investigaciones.

Elija un método de configuración de roles. Seleccione una de las siguientes opciones:

Opción 1: Crear automáticamente un nuevo rol de AWS DevOps agente (recomendado)

Esta opción crea automáticamente un rol con los permisos adecuados para que el AWS DevOps agente investigue los recursos de su cuenta.

> **Note**
>
> Para poder utilizar esta opción, debe tener permisos de IAM para crear nuevos roles.

1. Seleccione Crear automáticamente un nuevo rol de agente AWS DevOps
2. (Opcional) Actualice el nombre del rol de Agent Space que se va a crear

Opción 2: asignar un rol existente

Utilice esta opción cuando otro administrador haya creado previamente un rol específico para el AWS DevOps agente.

1. Seleccione Asignar un rol existente
2. En el menú desplegable, selecciona un rol existente que tenga los permisos adecuados

Opción 3: Crear un nuevo rol de AWS DevOps agente mediante una plantilla de políticas

Utilice esta opción cuando necesite limitar los servicios y recursos a los que el agente puede acceder en la cuenta principal.

1. Seleccione Crear un nuevo rol de AWS DevOps agente mediante una plantilla de políticas
2. Siga las instrucciones para crear la política de confianza y la política integrada del nuevo rol.

Habilitar la aplicación web Agent Space

La aplicación web es el lugar donde el personal interactúa con el AWS DevOps agente para investigar los incidentes y revisar las recomendaciones. Consulte AWS DevOps Agent Console Architecture [enlace] para obtener más información. Cuando está habilitada, los usuarios pueden acceder a la aplicación web Agent Space a través de un enlace de autenticación de IAM desde la consola de AWS administración.

Seleccione una de las siguientes opciones:

Opción 1: Crear automáticamente un nuevo rol de AWS DevOps agente (recomendado)

Esta opción crea automáticamente un rol con los permisos adecuados para acceder a la aplicación web del DevOps agente.

> **Note**
>
> Debe tener permisos de IAM para crear nuevos roles para usar esta opción.

1. Seleccione Crear automáticamente un nuevo rol de agente AWS DevOps
2. Revise los permisos que se otorgarán al rol

Opción 2: asignar un rol existente

Utilice esta opción cuando otro administrador haya creado previamente un rol de operador.

1. Seleccione Asignar un rol existente
2. En el menú desplegable, selecciona un rol existente que tenga los permisos adecuados

Opción 3: Crear un nuevo rol de AWS DevOps agente mediante una plantilla de políticas

Utilice esta opción cuando necesite personalizar los permisos de acceso a las aplicaciones web.

1. Seleccione Crear un nuevo rol de AWS DevOps agente mediante una plantilla de política
2. Siga las instrucciones para crear la política de confianza y la política integrada del nuevo rol.

Añadir etiquetas (opcional)

Puede añadir AWS etiquetas a su espacio de agente durante la creación. Las etiquetas son pares clave-valor que le ayudan a organizar e identificar sus recursos. Puede añadir hasta 50 etiquetas por espacio de agente. Para añadir etiquetas, expanda la sección Etiquetas de la página Crear espacio de agente y haga clic en Añadir nueva etiqueta.

Creación completa del espacio de agentes

Una vez rellenadas todas las secciones, haga clic en Crear

Verificando la configuración de Agent Space

Una vez configurado, el botón de acceso del operador aparecerá en la página de detalles del espacio de agentes. Al hacer clic en él, se abrirá la aplicación web en una nueva pestaña y se autenticará correctamente.

Siguientes pasos

Tras configurar su espacio de agente, tenga en cuenta los siguientes pasos:

- Agregue cuentas secundarias si sus aplicaciones abarcan varias AWS cuentas
- Configure integraciones de terceros, como herramientas de observación o sistemas de venta de entradas
- Configure la autenticación de AWS Identity Center para los entornos de producción
- Explore el mapeo de recursos de su aplicación para ayudar al AWS DevOps agente a entender su infraestructura

AWS DevOps Guía de incorporación de Agent CLI

Descripción general de

Con AWS DevOps Agent, puede monitorear y administrar su AWS infraestructura. Esta guía explica cómo configurar el AWS DevOps agente mediante la interfaz de línea de AWS comandos (AWS CLI). Puede crear funciones de IAM, configurar un espacio de agente y asociar su AWS cuenta. También habilitas la aplicación del operador y, de forma opcional, conectas integraciones de terceros. Esta guía tarda aproximadamente 20 minutos en completarse.

AWS DevOps El agente está disponible en seis AWS regiones: EE.UU. Este (Norte de Virginia), EE.UU. Oeste (Oregón), Asia Pacífico (Sídney), Asia Pacífico (Tokio), Europa (Fráncfort) y Europa (Irlanda). Para obtener más información sobre las regiones compatibles, consultethe section called "Regiones admitidas".

Requisitos previos

Antes de comenzar, asegúrese de que dispone de lo siguiente:

- AWS CLI versión 2 instalada y configurada
- Autenticación en su cuenta AWS de monitoreo
- Permisos para crear AWS funciones de Identity and Access Management (IAM) y adjuntar políticas
- Una AWS cuenta para usarla como cuenta de supervisión
- Familiaridad con la sintaxis AWS CLI y JSON

A lo largo de esta guía, sustituya los siguientes valores de marcador de posición por los suyos propios:

- `<MONITORING_ACCOUNT_ID>`— Su ID de AWS cuenta de 12 dígitos para la cuenta de monitoreo (principal)
- `<EXTERNAL_ACCOUNT_ID>`— El ID de AWS cuenta de 12 dígitos de la cuenta secundaria que se va a monitorear (usado en el paso 4)
- `<REGION>`— El código de AWS región de tu espacio de agente (por ejemplo, `us-east-1` o`eu-central-1`)
- `<AGENT_SPACE_ID>`— El identificador del espacio de agentes que devuelve el `create-agent-space` comando

Configuración de los roles de IAM

1. Cree el rol del espacio de DevOps agentes

Cree la política de confianza de IAM ejecutando el siguiente 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
```

Cree el rol de IAM:

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

Guarde el ARN del rol ejecutando el siguiente comando:

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

Adjunte la política AWS gestionada:

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

Cree y adjunte una política en línea para permitir la creación del rol vinculado al servicio 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. Cree el rol de IAM de la aplicación operadora

Cree la política de confianza de IAM ejecutando el siguiente 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
```

Cree el rol de IAM:

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

Guarde el ARN del rol ejecutando el siguiente comando:

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

Adjunte la política de la aplicación AWS gestionada por el operador:

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

Esta política gestionada concede a la aplicación del operador permisos para acceder a las funciones del espacio de agentes. Estas funciones incluyen investigaciones, recomendaciones, gestión del

conocimiento, chat e integración de AWS Support. La política limita el acceso al espacio de agentes específico mediante el uso de la `aws:PrincipalTag/AgentSpaceId` condición. Para obtener más información sobre la lista completa de acciones, consultethe section called “DevOps Permisos de IAM para agentes”.

Pasos de incorporación

1. Crea un espacio para agentes

Ejecute el siguiente comando para crear un espacio de agentes:

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

Si lo desea, especifique `--kms-key-arn` el uso de una clave AWS KMS administrada por el cliente para el cifrado. También puede utilizarla `--tags` para añadir etiquetas de recursos y `--locale` establecer el idioma de las respuestas de los agentes.

`agentSpaceId`Guarde el contenido de la respuesta (ubicado en`agentSpace.agentSpaceId`).

Para enumerar los espacios de sus agentes más adelante, ejecute el siguiente comando:

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

2. Asocia tu AWS cuenta

Asocie su AWS cuenta para activar la detección de topología. `accountType`Establézcalo en uno de los siguientes valores:

- `monitor`— La cuenta principal en la que se encuentra el espacio de agentes. Esta cuenta aloja el agente y se utiliza para el descubrimiento de la topología.
- `source`— Una cuenta adicional que el agente supervisa. Utilice este tipo cuando asocie cuentas externas en el paso 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. Habilita la aplicación del operador

Los flujos de autenticación pueden usar IAM, IAM Identity Center (IDC) o un proveedor de identidad externo (IdP). Ejecute el siguiente comando para habilitar la aplicación del operador en su espacio de 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>
```

Para la autenticación del Centro de Identidad de IAM, utilice `--auth-flow idc` y proporcione`--idc-instance-arn`. Para un proveedor de identidad externo, utilice `--auth-flow idp` y proporcione `--issuer-url``--idp-client-id`, y`--idp-client-secret`. Para obtener más información, consulte the section called “Configuración de la autenticación de IAM Identity Center” y the section called “Configuración de la autenticación de un proveedor de identidad externo (IdP)”.

> Nota: Si anteriormente creaste un rol de aplicación de operador para otro espacio de agente en tu cuenta, puedes reutilizar el ARN de ese rol.

4. (Opcional) Asocia cuentas de origen adicionales

Para supervisar cuentas adicionales con el AWS DevOps agente, cree un rol multicuenta de IAM.

Cree el rol multicuenta en la cuenta externa

Cambie a la cuenta externa y cree la política de confianza. `MONITORING_ACCOUNT_ID`Es la cuenta principal que aloja el espacio de agentes que configuró en el paso 2. Esta configuración permite que

el servicio de AWS DevOps agente asuma una función en las cuentas de origen secundarias en nombre de la cuenta de supervisión.

Ejecute el siguiente comando para crear la política de confianza:

```
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
```

Cree el rol de IAM multicuenta:

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

Guarde el ARN del rol ejecutando el siguiente comando:

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

Adjunte la política AWS gestionada:

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

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

Adjunte la política en línea para permitir la creación del rol vinculado al servicio Resource Explorer en la cuenta externa:

```
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
```

Asocie la cuenta externa

Vuelva a su cuenta de supervisión y, a continuación, ejecute el siguiente comando para asociar la cuenta externa:

```
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. (Opcional) Asociar GitHub

> Nota: Primero debe registrarse GitHub a través de la consola del AWS DevOps agente mediante el OAuth flujo antes de poder asociarlo a través de la CLI.

Para obtener instrucciones sobre cómo registrarse GitHub a través de la consola, consultethe section called “Conexión a CI/CD tuberías”.

Enumere los servicios registrados:

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

Guarde el <SERVICE_ID> para ServiceType:. `github`

Tras registrarse GitHub en la consola, asocie los GitHub repositorios ejecutando el siguiente 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. (Opcional) Registre y asocie ServiceNow

Primero, registre el ServiceNow servicio con OAuth las credenciales:

```
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>
```

Guarde lo devuelto y<SERVICE_ID>, a continuación, asocie 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. (Opcional) Registre y asocie Dynatrace

Primero, registre el servicio Dynatrace con las credenciales: 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>
```

Guarde las devueltas y, a continuación<SERVICE_ID>, asocie Dynatrace. Los recursos son opcionales. El entorno especifica el entorno de Dynatrace al que se debe asociar.

```
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 respuesta incluye información sobre webhooks para la integración. Puedes usar este webhook para iniciar una investigación por parte de Dynatrace. Para obtener más información, consulte the section called “Conectando Dynatrace”.

8. (Opcional) Registra y asocia Splunk

En primer lugar, registre el servicio de Splunk con BearerToken las credenciales.

El punto final utiliza el siguiente 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>
```

Guarde lo devuelto y<SERVICE_ID>, a continuación, asocie 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 respuesta incluye información sobre el webhook para la integración. Puedes usar este webhook para iniciar una investigación en Splunk. Para obtener más información, consulte the section called “Conexión de Splunk”.

9. (Opcional) Registra y asocia New Relic

En primer lugar, registre el servicio New Relic con las credenciales clave de la API.

Región: cualquiera de las dosUS. EU

Campos opcionales: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>
```

Guarda lo devuelto y<SERVICE_ID>, a continuación, asocia 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 respuesta incluye información sobre el webhook para la integración. Puedes usar este webhook para iniciar una investigación desde New Relic. Para obtener más información, consulte the section called “Conectando New Relic”.

10. (Opcional) Registre y asocie Datadog

Primero debe registrar Datadog a través de la consola del AWS DevOps agente mediante el OAuth flujo antes de poder asociarlo a través de la CLI. Para obtener más información, consulte the section called “Conectando DataDog”.

Enumere los servicios registrados:

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

Guarde el <SERVICE_ID> para ServiceType:. mcpserverdatadog

Luego asocie 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 respuesta incluye información sobre el webhook para la integración. Puedes usar este webhook para iniciar una investigación desde Datadog. Para obtener más información, consulte the section called “Conectando DataDog”.

11. (Opcional) Elimine un espacio de agente

Al eliminar un espacio de agentes, se eliminan todas las asociaciones, configuraciones y datos de investigación de ese espacio de agentes. Esta acción no se puede deshacer.

Para eliminar un espacio de agentes, ejecute el siguiente comando:

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

Verificación

Para comprobar la configuración, ejecute los siguientes comandos:

```
# 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>
```

Siguientes pasos

- Para conectar integraciones adicionales, consulteConfiguración de las capacidades del AWS DevOps agente.
- Para obtener información sobre las habilidades y capacidades de los agentes, consultethe section called “DevOps Habilidades de agente”.
- Para entender la aplicación web del operador, consultethe section called “¿Qué es una aplicación web para DevOps agentes?”.

Notas

- Sustituya `<AGENT_SPACE_ID>` `<MONITORING_ACCOUNT_ID><EXTERNAL_ACCOUNT_ID>,<REGION>`,, y así sucesivamente por sus valores reales.
- Para obtener una lista de las regiones admitidas, consulte the section called “Regiones admitidas”.

Creación de un entorno de pruebas

Esta guía proporciona pruebas prácticas para validar la funcionalidad de respuesta a incidentes del AWS DevOps agente mediante una arquitectura de muestra. Utilice este suplemento si quiere probar DevOps Agent antes de conectar sus sistemas de producción.

Requisitos previos

- AWS cuenta con acceso administrativo
- AWS DevOps Espacio de agente creado y configurado mediante el flujo de roles de DevOps agente de creación automática

Resumen de costes y seguridad

Protección de costes

- Prueba EC2: GRATUITA (nivel AWS gratuito) o aproximadamente 0,02$ durante 2 horas
- Prueba Lambda: GRATUITA (nivel gratuito de 1 millón requests/month)
- CloudWatch: GRATIS (10 alarmas, métricas básicas incluidas)

- Costo total estimado esperado: entre 0,00 y 0,05$ para completar las pruebas

Características de seguridad de estas pruebas

- Terminación automática: apagado automático incorporado
- Apto para el nivel gratuito: utiliza los tipos de instancias más pequeños
- Alcance limitado: recursos de prueba mínimos y aislados
- Limpieza sencilla: sencillos pasos de consola para quitarlo todo
- Sin impacto en la producción: entorno de prueba completamente independiente

Configure su AWS cuenta para realizar las pruebas

> **Important**
>
> Los recursos de infraestructura deben desplegarse en la AWS cuenta en la que creó su cuenta de nube principal de DevOps Agent Space. La región específica no importa.

1. Inicie sesión en AWS la consola: https://console.aws.amazon.com
2. Asegúrese de trabajar en la misma AWS cuenta en la que se encuentra su espacio de DevOps agente
3. Puedes usar cualquier región para tus recursos de prueba

> **Note**
>
> El mapeo individual entre la cuenta principal de su DevOps agente y los recursos del entorno de prueba que está creando simplifica la configuración de la prueba. Puede ampliar fácilmente su espacio de DevOps agente para incluir cuentas secundarias y permitir investigaciones entre cuentas.

Elige tu prueba

Puede realizar cualquiera de las dos pruebas de forma independiente o ambas a la vez:

Opción de prueba A: prueba de capacidad de la CPU EC2

Objetivo: validar la capacidad del AWS DevOps agente para detectar e investigar los problemas de rendimiento de EC2

Tiempo estimado: 5 minutos de configuración más 10 minutos de ejecución automática

Dificultad: Totalmente automatizado (no se requieren pasos manuales)

Opción de prueba B: prueba de tasa de error Lambda

Objetivo: Validar la capacidad del AWS DevOps agente para detectar e investigar los errores de la función Lambda

Tiempo estimado: 10 minutos de configuración más 2 minutos de activación

Dificultad: Muy fácil

Opción de prueba A: prueba de capacidad de la CPU EC2

Paso 1: Implemente la CloudFormation pila para la prueba de EC2

Los utilizaremos CloudFormation para crear nuestros recursos de prueba, lo que permitirá al AWS DevOps agente rastrearlos e investigarlos adecuadamente.

1. Navega hasta CloudFormation:
   a. En AWS la consola, busque CloudFormation "» y haga clic en CloudFormation
   b. Haga clic en Crear pila > Con recursos nuevos (estándar)
2. Cargar plantilla:
   a. Cree un nuevo archivo local llamado `AWS-DevOpsAgent-ec2-test.yaml`
   b. Copia y pega esta CloudFormation plantilla en el archivo:
      i.
```
AWSTemplateFormatVersion: '2010-09-09'
Description: 'AWS DevOps Agent EC2 CPU Test Stack'
Parameters:
  MyIP:
    Type: String
    Description: Your current IP address for SSH access (find at https://
whatismyipaddress.com)
    Default: '0.0.0.0/0'
Resources:
  # Security Group for SSH access
```

```
  TestSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupName: AWS-DevOpsAgent-test-sg
      GroupDescription: AWS DevOps Agent beta testing security group
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 22
          ToPort: 22
          CidrIp: !Ref MyIP
          Description: SSH access from your IP
      Tags:
        - Key: Name
          Value: AWS-DevOpsAgent-Test-SG
        - Key: Purpose
          Value: AWS-DevOpsAgent-Testing
  # Key Pair for SSH access
  TestKeyPair:
    Type: AWS::EC2::KeyPair
    Properties:
      KeyName: AWS-DevOpsAgent-test-key
      KeyType: rsa
      Tags:
        - Key: Name
          Value: AWS-DevOpsAgent-Test-Key
        - Key: Purpose
          Value: AWS-DevOpsAgent-Testing
  # EC2 Instance for CPU testing
  TestInstance:
    Type: AWS::EC2::Instance
    Properties:
      InstanceType: t3.micro
      ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-
kernel-6.1-x86_64}}'
      KeyName: !Ref TestKeyPair
      SecurityGroupIds:
        - !Ref TestSecurityGroup
      UserData:
        Fn::Base64: !Sub |
          #!/bin/bash
          yum update -y
          yum install -y htop

          # Create the CPU stress test script
```

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

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

          echo "Starting stress test (5 minutes)..."
          echo "This will generate >70% CPU usage to trigger CloudWatch alarm"
          echo ""

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

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

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

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

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

```
          sudo shutdown -h now
          SHUTDOWN_EOF

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

          echo "AWS DevOpsAgent test setup completed at $(date)" > /home/ec2-
user/setup-complete.txt
      Tags:
        - Key: Name
          Value: AWS-DevOpsAgent-Test-Instance
        - Key: Purpose
          Value: AWS-DevOpsAgent-Testing
  # CloudWatch Alarm for CPU utilization
  CPUAlarm:
    Type: AWS::CloudWatch::Alarm
    Properties:
      AlarmName: AWS-DevOpsAgent-EC2-CPU-Test
      AlarmDescription: AWS-DevOpsAgent beta test - EC2 CPU utilization alarm
      MetricName: CPUUtilization
      Namespace: AWS/EC2
      Statistic: Average
      Period: 60
      EvaluationPeriods: 1
      Threshold: 70
      ComparisonOperator: GreaterThanThreshold
      Dimensions:
        - Name: InstanceId
          Value: !Ref TestInstance
      TreatMissingData: notBreaching
Outputs:
  InstanceId:
    Description: EC2 Instance ID for testing
    Value: !Ref TestInstance

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

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

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

c. En la CloudFormation consola, selecciona Cargar un archivo de plantilla

d. Haz clic en Elegir archivo

e. Seleccione el `AWS-DevOpsAgent-ec2-test.yaml` archivo

f. Haga clic en Siguiente.

3. Configurar la pila:

a. Nombre de la pila: `AWS-DevOpsAgent-EC2-Test`

b. Parámetros:

i. MyIP: dejar como predeterminado `0.0.0.0/0` (puede asegurarlo más adelante si es necesario)

c. Haga clic en Siguiente.

4. Configura las opciones de pila:

a. Deje los valores predeterminados y haga clic en Siguiente

5. Revise y cree:

a. Marque Acepto que AWS CloudFormation podría crear recursos de IAM

b. Haga clic en Enviar

6. Espere a que finalice:

a. La creación de la pila tarda de 3 a 5 minutos

b. El estado cambiará de `CREATE_IN_PROGRESS` a `CREATE_COMPLETE`

c. Importante: ¡Su instancia EC2 ahora forma parte de una CloudFormation pila que AWS DevOpsAgent puede rastrear!

Opcional: acceso SSH seguro (solo si planea conectarse a la instancia)

Omite este paso si solo quieres ejecutar la prueba automática

1. Navegue hasta los grupos de seguridad de EC2:

a. En la AWS consola, vaya a EC2 → Grupos de seguridad

b. Buscar `AWS-DevOpsAgent-test-sg`

2. Actualizar la regla SSH:

a. Seleccione el grupo de seguridad → pestaña Reglas de entrada → Editar reglas de entrada
b. Busque la regla SSH (puerto 22)
c. Cambia la fuente de `0.0.0.0/0` a tu IP: `[YOUR_IP]/32`
d. Obtenga su IP de https://whatismyipaddress.com
e. Haz clic en Guardar reglas

Paso 2: Espere a que se ejecute automáticamente la prueba

1. Ejecución automática de la prueba:
   - La prueba de stress de la CPU se iniciará automáticamente 5 minutos después del lanzamiento de la instancia
   - No es necesaria ninguna intervención manual: espere, la prueba se ejecuta completamente en segundo plano
2. Supervise la prueba:
   - La instancia inicia y prepara la prueba automáticamente
   - El script se ejecutará durante 5 minutos y generará un uso de CPU superior al 70%
   - CloudWatch la alarma debería activarse en un plazo total de 8 a 10 minutos (5 minutos de retraso + 3-5 minutos para la alarma)
3. Opcional: repetición manual (para pruebas adicionales):
   - Conéctese a su instancia: consola EC2 → → Connect **`AWS-DevOpsAgent-Test-Instance`** → Session Manager
   - Vuelva a realizar la prueba de stress: `./cpu-stress-test.sh`
   - Perfecto para probar AWS DevOpsAgent la respuesta varias veces

Opción de prueba B: prueba de tasa de error Lambda

Paso 1: Implementar la CloudFormation pila para la prueba Lambda

1. Navegue hasta CloudFormation:
   a. En AWS la consola, vaya a CloudFormation
   b. Haga clic en Crear pila → Con nuevos recursos (estándar)
2. Cargar plantilla:

a. Cree un nuevo archivo local llamado `AWS-DevOpsAgent-lambda-test.yaml`

b. Copia y pega esta CloudFormation plantilla en el archivo:

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. En la CloudFormation consola, selecciona Cargar un archivo de plantilla

d. Haz clic en Elegir archivo

e. Seleccione el `AWS-DevOpsAgent-lambda-test.yaml` archivo

f. Haga clic en Siguiente.

3. Configurar la pila:
   a. Nombre de la pila: `AWS-DevOpsAgent-Lambda-Test`
   b. Haga clic en Siguiente.
4. Configura las opciones de pila:
   a. Deje los valores predeterminados, haga clic en Siguiente
5. Revise y cree:
   a. Marque Acepto que AWS CloudFormation podría crear recursos de IAM
   b. Haga clic en Enviar
6. Espere a que finalice:
   a. La creación de la pila tarda de 2 a 3 minutos
   b. El estado cambiará a `CREATE_COMPLETE`

Paso 2: Activar errores Lambda

1. Navegue a la consola Lambda:
   a. Ir a la AWS consola Lambda
   b. Encuentre su función `AWS-DevOpsAgent-test-lambda`
2. Pruebe la función:
   a. Haga clic en la pestaña Probar

b. Haz clic en Crear nuevo evento

c. Nombre del evento: `AWS-DevOpsAgent-test-event`

d. Usa esta carga útil de JSON:

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

e. Haz clic en Guardar

3. Generar errores:

a. Haga clic en el botón Probar 3 veces (espere 10 segundos entre cada una)

b. Cada prueba generará un error intencionado

c. CloudWatch la alarma debería activarse en 2-3 minutos

d. AWS DevOpsAgentahora debería poder detectar la alarma con una investigación en la aplicación Operator que configurarás a continuación.

Valide la detección AWS DevOps del agente

Paso 1: CloudWatch Alarmas de control de cordura (opcionales)

Este paso sirve para garantizar que las pruebas anteriores estén ahora en estado de alarma.

Para la prueba EC2:

- En la CloudWatch consola, vaya a Alarmas
- Espere de 3 a 5 minutos después de comenzar la prueba de stress
- La alarma debería mostrarse en estado de alarma
- Si sigue «bien»: espera otros 2 o 3 minutos (CloudWatch las métricas pueden retrasarse)

Para la prueba Lambda:

- Compruebe la alarma `AWS-DevOpsAgent-Lambda-Error-Test`
- Debería mostrar la alarma entre 2 y 3 minutos después de ejecutar las pruebas

Paso 2: Iniciar una investigación sobre un AWS DevOps agente

1. Abre tu AWS DevOps agente AgentSpace
2. Haz clic en Acceso de administrador. Esto abrirá la aplicación web DevOps Agent Space en una ventana nueva
3. Haga clic en el botón Iniciar investigación en la parte derecha de la pantalla
4. Complete el siguiente formulario:
   a. Detalles de la investigación: describe la investigación que te gustaría llevar a cabo. Incluye todos los detalles que puedas sobre los objetivos de la investigación, las áreas a explorar o la información relevante.
   b. Punto de partida de la investigación: describe la información con la que quieres iniciar la investigación. Puedes mencionar una alarma, una métrica, un fragmento de registro o cualquier otra cosa para que el DevOps agente tenga un punto de partida desde el que trabajar. En este caso, proporciona un resumen de las alarmas que acabas de crear.
   c. Fecha y hora del incidente (se prefiere la norma ISO 8601) ::MMZ YYYY-MM-DDTHH
   d. Ponle un nombre a tu investigación: ejemplo: `Oncall_investigation_1:2025-10-27`
   e. AWS ID de cuenta del incidente
   f. Región donde ocurrió el incidente
   g. Prioridad: AWS DevOpsAgent permite realizar dos investigaciones simultáneas. La prioridad le permite definir el orden de ejecución de sus investigaciones.
5. Haga clic en Investigar para iniciar la investigación.
6. Haz clic en la investigación que aparece en el panel de control. Accederás a la pantalla de detalles de la investigación, donde podrás ver las medidas detalladas que está tomando el DevOps agente.

Resultados esperados

Resultados de la prueba EC2:

- Detecta la alarma de la CPU EC2
- Identifica la causa raíz: «carga de trabajo de pruebas de stress de la CPU»
- Muestra la cronología: Prueba de esfuerzo → Pico de CPU → Alarma
- Proporciona recomendaciones para la supervisión y el escalado

Resultados de la prueba Lambda:

- Detecta un pico en la tasa de errores de Lambda
- Identifica la causa principal: «Excepciones de prueba intencionales»
- Muestra la cronología: Invocaciones de funciones → Errores → Alarma
- Proporciona recomendaciones para el manejo y monitoreo de errores

Instrucciones de limpieza

Prueba de limpieza A (prueba EC2)

Limpieza automática

- La instancia finalizará automáticamente después de 2 horas (integrada en la CloudFormation plantilla)

Limpieza manual (inmediata)

1. Eliminar CloudFormation pila:
   a. Ve a la CloudFormation consola
   b. Selecciona una `AWS-DevOpsAgent-EC2-Test` pila
   c. Haz clic en Eliminar
   d. Confirme la eliminación
   e. Esto eliminará automáticamente todos los recursos: instancia EC2, grupo de seguridad, key pair y alarma CloudWatch

Prueba de limpieza B (prueba Lambda)

1. Eliminar pila CloudFormation :
   a. Ve a la CloudFormation consola
   b. Selecciona una `AWS-DevOpsAgent-Lambda-Test` pila
   c. Haz clic en Eliminar
   d. Confirme la eliminación

e. Esto eliminará automáticamente todos los recursos: función Lambda, función de IAM y alarma CloudWatch

Resolución de problemas

Problemas comunes

«No se puede conectar a la instancia EC2»

- Compruebe el grupo de seguridad: asegúrese de que SSH (puerto 22) esté abierto a su IP
- Compruebe los permisos clave: ejecute `chmod 400 AWS-DevOpsAgent-test-key.pem`
- Verificar la IP pública: la instancia debe tener una IP pública asignada
- Espere una instancia: asegúrese de que la instancia esté en estado «En ejecución»

«La alarma no se activa»

- Espere a que aparezcan las métricas: CloudWatch las métricas pueden tardar entre 2 y 5 minutos en aparecer
- Comprueba la carga de la CPU: usa SSH a la instancia y ejecútala `top` para verificar que la CPU sea superior al 70%
- Verifica la prueba de esfuerzo: `ps aux | grep yes` ejecútala para ver si los procesos de carga se están ejecutando
- Espera prolongada: a veces se tarda entre 7 y 8 minutos en activarse la primera alarma

Validación de prueba

La prueba de su AWS DevOp agente se realiza correctamente cuando:

Validación técnica

- Precisión de la investigación: los resultados de la prueba EC2 deberían indicar correctamente que la alarma se activó debido a la carga de la CPU. El resultado de la prueba Lambda debería indicar que se trató de un fallo intencionado.
- Precisión del cronograma: se muestra la secuencia correcta de eventos
- Calidad de la recomendación: se proporcionan sugerencias prácticas

Cómo empezar a usar AWS DevOps Agent mediante AWS CDK

Descripción general de

Esta guía muestra cómo usar el AWS Cloud Development Kit (AWS CDK) para crear e implementar recursos de AWS DevOps agentes. La aplicación AWS CDK automatiza la creación de un espacio de agentes, funciones de AWS Identity and Access Management (IAM), una aplicación de operador y asociaciones de cuentas mediante. AWS AWS CloudFormation

El enfoque AWS CDK automatiza los pasos manuales descritos en la guía de incorporación de CLI al definir todos los recursos necesarios como infraestructura o código.

AWS DevOps El agente está disponible en las siguientes 6 AWS regiones: EE.UU. Este (Norte de Virginia), EE.UU. Oeste (Oregón), Asia Pacífico (Sídney), Asia Pacífico (Tokio), Europa (Fráncfort) y Europa (Irlanda). Para obtener más información sobre las regiones compatibles, consultethe section called “Regiones admitidas”.

Requisitos previos

Antes de comenzar, asegúrese de que dispone de lo siguiente:

- AWS Interfaz de línea de comandos (AWS CLI) instalada y configurada con las credenciales adecuadas
- La versión 18 o posterior de Node.js
- AWS Interfaz de línea de comandos (CLI) CDK instalada globalmente. Para instalar la CLI de AWS CDK, ejecute el siguiente comando:

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

- Una AWS cuenta para la cuenta de supervisión (principal)
- (Opcional) Una segunda AWS cuenta si desea configurar la supervisión multicuenta

¿Qué cubre esta guía

Esta guía se divide en dos partes:

- Parte 1: Implemente un espacio de agente con una aplicación de operador y una AWS asociación en su cuenta de monitoreo. Una vez completada esta parte, el agente puede supervisar los problemas en esa cuenta.
- Parte 2 (opcional): añadir una AWS asociación de origen para una cuenta de servicio e implementar una función de IAM multicuenta en esa cuenta. Esta configuración permite que el espacio de agentes supervise los recursos de todas las cuentas.

Recursos creados

Parte 1: DevOpsAgentStack (supervisión de la cuenta)

- Función de IAM (`DevOpsAgentRole-AgentSpace`): la asume el servicio de DevOps agente para supervisar la cuenta. Incluye la política `AIDevOpsAgentAccessPolicy` administrada y una política en línea que permite la creación del rol vinculado al servicio Resource Explorer.
- Función de IAM (**`DevOpsAgentRole-WebappAdmin`**): función de operador en la aplicación con la política `AIDevOpsOperatorAppAccessPolicy` gestionada para las operaciones de los agentes.
- Espacio de agente (`MyCDKAgentSpace`): el espacio de agente central, creado mediante el `AWS::DevOpsAgent::AgentSpace` CloudFormation recurso. Incluye la configuración de la aplicación del operador.
- Asociación (AWS monitor): vincula la cuenta de monitoreo al espacio del agente mediante el `AWS::DevOpsAgent::Association` CloudFormation recurso.
- Asociación (AWS fuente): (opcional) vincula la cuenta de servicio al espacio de agentes para la supervisión de varias cuentas.

Parte 2: ServiceStack (cuenta de servicio, opcional)

- Función de IAM (`DevOpsAgentRole-SecondaryAccount`): función multicuenta con un nombre fijo. El espacio de agente de la cuenta de supervisión confía en él. Incluye la política `AIDevOpsAgentAccessPolicy` administrada y una política en línea que permite la creación del rol vinculado al servicio Resource Explorer.
- Función Lambda (`echo-service`): un servicio de ejemplo sencillo que reproduce los eventos de entrada.

Configuración

Paso 1: clona el repositorio de muestras

Ejecute los siguientes comandos para clonar el repositorio y cambiarlo al directorio del proyecto:

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

Paso 2: Instalar las dependencias

Ejecute el siguiente comando para instalar las dependencias del proyecto:

```
npm install
```

Parte 1: Despliegue el espacio de agentes

En esta sección, creará el espacio de agentes, las funciones de IAM, la aplicación del operador y una AWS asociación en su cuenta de supervisión.

Paso 1: Configure el ID de la cuenta de monitoreo

Abra `lib/constants.ts` y configure el ID de su cuenta de monitoreo:

El siguiente ejemplo muestra la constante que se va a actualizar:

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

Paso 2: Inicie el entorno AWS CDK

Si no ha iniciado el AWS CDK en su cuenta de monitoreo, ejecute el siguiente comando:

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

Paso 3: Compila e implementa

Ejecute los siguientes comandos para crear el TypeScript código e implementar la pila:

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

Paso 4: Registra los resultados de la pila

Una vez completada la implementación, la AWS CDK imprime los resultados de la pila. Registre estos valores para usarlos más adelante.

El siguiente ejemplo muestra el resultado esperado:

```
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
```

Si planea completar la segunda parte, guarde el `AgentSpaceArn` valor. Lo necesita para configurar la pila de cuentas de servicio.

Paso 5: Verificar la implementación

Para comprobar que el espacio de agentes se creó correctamente, ejecute el siguiente comando AWS CLI:

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

En este punto, su espacio de agente se despliega con la aplicación del operador habilitada y su cuenta de supervisión asociada. El agente puede supervisar los problemas en esta cuenta.

Parte 2 (opcional): Añadir la supervisión entre cuentas

En esta sección, ampliará la configuración para que su espacio de agente pueda supervisar los recursos de una segunda AWS cuenta (la cuenta de servicio). Esto implica dos acciones:

1. Añadir una AWS asociación de origen en la DevOpsAgentStack que apunte a la cuenta de servicio.
2. Implementarla ServiceStack en la cuenta de servicio con una función de IAM que confíe en el espacio del agente.

> **Important**
>
> Debe completar la primera parte antes de continuar. ServiceStack Requiere el resultado `AgentSpaceArn` del DevOpsAgentStack despliegue.

Paso 1: Configurar el ID de la cuenta de servicio

Abra `lib/constants.ts` y configure el ID de su cuenta de servicio:

El siguiente ejemplo muestra la constante que se va a actualizar:

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

DevOpsAgentStack Crea una AWS asociación de fuentes con este identificador de cuenta. Si lo implementó DevOpsAgentStack antes de establecer este valor, vuelva a implementarlo para crear la asociación:

Ejecute los siguientes comandos para volver a desplegarla:

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

Paso 2: Configurar el ARN del espacio de agentes

Copie el `AgentSpaceArn` valor de la DevOpsAgentStack salida (parte 1, paso 4) y configúrelo en`lib/constants.ts`:

El siguiente ejemplo muestra la constante que se va a actualizar:

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

ServiceStack Utiliza este valor para determinar el alcance de la política de confianza en el rol de cuenta secundaria. El solo ServiceStack se sintetiza cuando se establece este valor.

Paso 3: Inicie la cuenta de servicio

Si no has iniciado la AWS CDK de tu cuenta de servicio, ejecuta el siguiente comando:

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

Paso 4: Implemente el ServiceStack

Ejecute los siguientes comandos para crear e implementar el ServiceStack mediante las credenciales de la cuenta de servicio:

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

Esto crea los siguientes recursos en la cuenta de servicio:

- Un rol de IAM (`DevOpsAgentRole-SecondaryAccount`) que confía en el espacio de agentes de la cuenta de supervisión
- Una función echo Lambda (`echo-service`) como servicio de ejemplo

Paso 5: Verificar el despliegue

Para confirmar que la función Lambda se implementó correctamente, ejecute los siguientes comandos para probar el servicio echo:

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

Resolución de problemas

En esta sección se describen los problemas más comunes y cómo resolverlos.

CloudFormation no se encontró el tipo de recurso

- Compruebe que está realizando el despliegue en unthe section called “Regiones admitidas”.
- Confirme que la AWS CLI esté configurada con los permisos adecuados.

Error al crear el rol de IAM

- Compruebe que su función de despliegue tenga permisos para crear funciones de IAM.
- Comprueba que las condiciones de la política de confianza coincidan con tu ID de cuenta.

El despliegue entre cuentas falla y muestra el mensaje «No se pudo asumir el rol en la cuenta de destino»

- Cada pila debe implementarse con las credenciales de la cuenta de destino. Utilice la `--profile` marca para especificar el perfil AWS CLI correcto.
- Compruebe que la AWS CDK se haya iniciado en la cuenta de destino.

Retrasos en la propagación de IAM

- Los cambios de rol de IAM pueden tardar unos minutos en propagarse. Si se produce un error al crear el espacio de agente inmediatamente después de crear el rol, espere unos minutos y vuelva a desplegarlo.

Limpieza

Para eliminar todos los recursos, destruye las pilas en orden inverso.

Ejecuta los siguientes comandos para destruir las pilas:

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

Advertencia: esta acción elimina permanentemente su espacio de agente y todos los datos asociados. Esta acción no se puede deshacer. Asegúrese de haber hecho una copia de seguridad de toda la información importante antes de continuar.

Consideraciones de seguridad

- La aplicación AWS CDK crea funciones de IAM con políticas de confianza que solo permiten que el director del `aidevops.amazonaws.com` servicio las asuma.
- Las políticas de confianza incluyen condiciones que restringen el acceso a su AWS cuenta específica y al ARN del espacio de agente.

- Todas las políticas siguen el principio del privilegio mínimo. Revise y personalice las políticas de IAM en función de los requisitos de seguridad de su organización.
- El rol multicuenta (`DevOpsAgentRole-SecondaryAccount`) usa un nombre fijo y se limita a un ARN de espacio de agentes específico.

Siguientes pasos

Una vez que haya desplegado su AWS DevOps agente mediante la CDK: AWS

1. Obtenga más información sobre la gama completa de funciones del DevOps agente en la Guía del usuario del AWS DevOps agente.
2. Considere la posibilidad de integrar la implementación de la AWS CDK en sus CI/CD procesos para automatizar la administración de la infraestructura.

Recursos adicionales

- AWS DevOps Guía del usuario del agente
- Ejemplo de repositorio de CDK en el sitio web GitHub
- Guía de incorporación de CLI

Cómo empezar a usar AWS DevOps Agent AWS CloudFormation

Descripción general de

En esta guía, se muestra cómo utilizar AWS CloudFormation plantillas para crear e implementar los recursos AWS DevOps del agente. Las plantillas automatizan la creación de un espacio de agentes, funciones de AWS Identity and Access Management (IAM), una aplicación de operador y asociaciones de AWS cuentas tanto de infraestructura como de código.

El CloudFormation enfoque automatiza los pasos manuales descritos en la guía de incorporación de la CLI al definir todos los recursos necesarios en plantillas YAML declarativas.

AWS DevOps El agente está disponible en las siguientes 6 AWS regiones: EE.UU. Este (Norte de Virginia), EE.UU. Oeste (Oregón), Asia Pacífico (Sídney), Asia Pacífico (Tokio), Europa (Fráncfort) y Europa (Irlanda). Para obtener más información sobre las regiones compatibles, consultethe section called “Regiones admitidas”.

Requisitos previos

Antes de comenzar, asegúrese de que dispone de lo siguiente:

- AWS Interfaz de línea de comandos (AWS CLI) instalada y configurada con las credenciales adecuadas
- Permisos para crear roles y CloudFormation pilas de IAM
- Una AWS cuenta para la cuenta de supervisión (principal)
- (Opcional) Una segunda AWS cuenta si desea configurar la supervisión multicuenta

¿Qué cubre esta guía

Esta guía se divide en dos partes:

- Parte 1: Implemente un espacio de agente con una aplicación de operador y una AWS asociación en su cuenta de monitoreo. Una vez completada esta parte, el agente puede supervisar los problemas en esa cuenta.
- Parte 2 (opcional): implementar una función de IAM multicuenta en una cuenta secundaria y añadir una asociación de origen AWS . Esta configuración permite que el espacio de agentes supervise los recursos de todas las cuentas.

Parte 1: Despliegue el espacio de agentes

En esta sección, debe crear una CloudFormation plantilla que aprovisione el espacio de agente, las funciones de IAM, la aplicación del operador y una AWS asociación en su cuenta de monitoreo.

Paso 1: Crea la plantilla CloudFormation

Guarde la siguiente plantilla como`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
```

Paso 2: Despliegue la pila

Ejecute el siguiente comando para implementar la pila. `<REGION>`Sustitúyalo por un the section called “Regiones admitidas” (por ejemplo,`us-east-1`).

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

Paso 3: Registre las salidas de la pila

Una vez completada la implementación, ejecute el siguiente comando para recuperar los resultados de la pila. Registre estos valores para usarlos más adelante.

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

El siguiente ejemplo muestra el resultado esperado:

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

Si planea completar la segunda parte, guarde el `AgentSpaceArn` valor. Lo necesita para configurar el rol multicuenta.

Paso 4: Verificar la implementación

Para comprobar que el espacio de agentes se creó correctamente, ejecute el siguiente comando AWS CLI:

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

En este punto, su espacio de agente se despliega con la aplicación del operador habilitada y su cuenta de supervisión asociada. El agente puede supervisar los problemas en esta cuenta.

Parte 2 (opcional): Añadir la supervisión entre cuentas

En esta sección, ampliará la configuración para que su espacio de agente pueda supervisar los recursos de una segunda AWS cuenta (la cuenta de servicio). Esto implica dos acciones:

1. Implementar una función de IAM en la cuenta de servicio que confía en el espacio de agentes.
2. Añadir una AWS asociación de origen a la cuenta de supervisión que apunte a la cuenta de servicio.

> Nota: Debe completar la parte 1 antes de continuar. La plantilla de la cuenta de servicio requiere los resultados `AgentSpaceArn` de la pila de la parte 1.

Paso 1: Crear la plantilla de cuenta de servicio

Guarde la siguiente plantilla como`devops-agent-service-account.yaml`. Esta plantilla crea un rol de IAM multicuenta en la cuenta secundaria.

```
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
```

Paso 2: Implemente la pila de cuentas de servicio

Con las credenciales de la cuenta de servicio, ejecute el siguiente comando:

```
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>
```

Paso 3: Añada la AWS asociación de origen

Vuelva a la cuenta de monitoreo y cree una AWS asociación de fuentes. Para ello, puede crear una pila independiente o actualizar la plantilla original. En el siguiente ejemplo, se utiliza una plantilla independiente.

Guarde la siguiente plantilla como`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
```

Implemente la pila de asociaciones mediante las credenciales de la cuenta de supervisión:

```
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>
```

Verificación

Verifique la configuración ejecutando los siguientes comandos de 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>
```

Resolución de problemas

En esta sección se describen los problemas más comunes y cómo resolverlos.

CloudFormation no se encontró el tipo de recurso

- Compruebe que está realizando el despliegue en unthe section called “Regiones admitidas”.
- Confirme que la AWS CLI esté configurada con los permisos adecuados.

No se pudo crear el rol de IAM

- Compruebe que sus credenciales de despliegue tengan permisos para crear roles de IAM con nombres personalizados ()`CAPABILITY_NAMED_IAM`.
- Comprueba que las condiciones de la política de confianza coincidan con tu ID de cuenta.

Se produce un error en el despliegue entre cuentas

- Cada pila debe implementarse con las credenciales de la cuenta de destino. Utilice la `--profile` marca para especificar el perfil AWS CLI correcto.
- Verifique que el `AgentSpaceArn` parámetro coincida con el ARN exacto de las salidas de la pila de la Parte 1.

Retrasos de propagación de IAM

- Los cambios de rol de IAM pueden tardar unos minutos en propagarse. Si se produce un error al crear el espacio de agente inmediatamente después de crear el rol, espere unos minutos y vuelva a desplegarlo.

Limpieza

Para eliminar todos los recursos, elimine las pilas en orden inverso.

Advertencia: esta acción elimina permanentemente su espacio de agente y todos los datos asociados. Esta acción no se puede deshacer. Asegúrese de haber hecho una copia de seguridad de toda la información importante antes de continuar.

Ejecute los siguientes comandos para eliminar las pilas:

```
# 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>
```

Siguientes pasos

Una vez que haya desplegado su AWS DevOps agente mediante AWS CloudFormation:

- Para conectar integraciones adicionales, consulteConfiguración de las capacidades del AWS DevOps agente.
- Para obtener más información sobre las habilidades y capacidades de los agentes, consultethe section called “DevOps Habilidades de agente”.

- Para entender la aplicación web del operador, consultethe section called “¿Qué es una aplicación web para DevOps agentes?”.

Cómo empezar a usar AWS DevOps Agent con Terraform

Descripción general de

En esta guía, se muestra cómo usar Terraform para crear y desplegar recursos de AWS DevOps agentes. La configuración de Terraform automatiza la creación de un espacio de agentes, funciones de IAM, una aplicación de operador y asociaciones de cuentas. AWS

El enfoque de Terraform automatiza los pasos manuales descritos en la guía de incorporación de CLI al definir todos los recursos necesarios como infraestructura o código.

AWS DevOps El agente está disponible en las siguientes 6 AWS regiones: EE.UU. Este (Norte de Virginia), EE.UU. Oeste (Oregón), Asia Pacífico (Sídney), Asia Pacífico (Tokio), Europa (Fráncfort) y Europa (Irlanda). Para obtener más información sobre las regiones compatibles, consultethe section called “Regiones admitidas”.

Requisitos previos

Antes de empezar, asegúrese de que tiene lo siguiente:

- Terraform >= 1.0 instalado
- AWS CLI instalada y configurada con las credenciales adecuadas
- Una AWS cuenta para la cuenta de supervisión (principal)
- (Opcional) Una segunda AWS cuenta si desea configurar la supervisión multicuenta

¿Qué cubre esta guía

Esta guía se divide en dos partes:

- Parte 1: Implemente un espacio de agente con una aplicación de operador y una AWS asociación en su cuenta de monitoreo. Tras completar esta parte, el agente puede supervisar los problemas en esa cuenta.

- Parte 2 (opcional): añadir una AWS asociación de origen para una cuenta de servicio e implementar un rol de IAM multicuenta más un echo Lambda en esa cuenta. Esto permite que el espacio de agentes supervise los recursos de todas las cuentas.

Recursos creados

Parte 1: Supervisión de la cuenta

- Función de IAM (`DevOpsAgentRole-AgentSpace-*`): la asume el servicio de DevOps agente para supervisar la cuenta. Incluye la política `AIDevOpsAgentAccessPolicy` administrada y una política en línea que permite la creación del rol vinculado al servicio Resource Explorer.
- Función de IAM (**`DevOpsAgentRole-WebappAdmin-*`**): función de operador en la aplicación con la política `AIDevOpsOperatorAppAccessPolicy` gestionada para las operaciones de los agentes.
- Espacio de agentes (nombre configurable): el espacio de agente central, creado con el `awscc_devopsagent_agent_space` recurso. Incluye la configuración de la aplicación del operador.
- Asociación (AWS monitor): vincula la cuenta de monitoreo al espacio del agente mediante el `awscc_devopsagent_association` recurso.
- Asociación (AWS fuente): (opcional) vincula la cuenta de servicio al espacio de agentes para la supervisión de varias cuentas.

Parte 2: Cuenta de servicio (opcional)

- Función de IAM (`DevOpsAgentRole-SecondaryAccount-TF`): función multicuenta con un nombre fijo. El espacio de agente de la cuenta de supervisión confía en él. Incluye la política `AIDevOpsAgentAccessPolicy` administrada y una política en línea que permite la creación del rol vinculado al servicio Resource Explorer.
- Función Lambda (`echo-service-tf`): un servicio de ejemplo sencillo que reproduce los eventos de entrada.

Configuración

Paso 1: clona el repositorio de muestras

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

Paso 2: Configurar las variables

Copie el archivo de variables de ejemplo y personalícelo para su entorno:

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

`terraform.tfvars`Edítelo con el nombre y la descripción de su espacio de agente:

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

Parte 1: Despliegue el espacio de agentes

En esta sección, creará el espacio de agentes, las funciones de IAM, la aplicación del operador y una AWS asociación en su cuenta de supervisión.

Paso 1: Implemente con automatización (recomendado)

Utilice el script de implementación proporcionado para una configuración simplificada:

```
./deploy.sh
```

Este script automáticamente:

- Comprueba los requisitos previos (Terraform, AWS CLI, credenciales)
- Crea a `terraform.tfvars` partir de un ejemplo si es necesario
- Inicializa, valida, planifica y aplica Terraform

Como alternativa, si prefiere el control manual:

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

Escriba `yes` cuando se le pida que confirme el despliegue.

Paso 2: Registre los resultados

Una vez completada la implementación, Terraform imprime los resultados. Registre estos valores para usarlos más adelante:

```
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"
```

Si planea completar la parte 2, guarde el `agent_space_arn` valor. Lo necesitará para configurar los recursos de la cuenta de servicio.

Paso 3: Verificar la implementación

Ejecute el script de verificación posterior al despliegue:

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

O utilice la AWS CLI para comprobar que el espacio de agentes se creó correctamente:

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

En este punto, su espacio de agente se despliega con la aplicación del operador habilitada y su cuenta de supervisión asociada. El agente puede supervisar los problemas en esta cuenta.

Parte 2 (opcional): Añadir la supervisión entre cuentas

En esta sección, ampliará la configuración para que el espacio de agentes pueda supervisar los recursos de una segunda AWS cuenta (la cuenta de servicio). Esto implica dos acciones:

1. Añadir una AWS asociación de origen que apunte a la cuenta de servicio.
2. Implementación de una función de IAM multicuenta y una función de echo Lambda en la cuenta de servicio.

> **Important**
>
> Debe completar la parte 1 antes de continuar. Los recursos de la cuenta de servicio requieren el resultado `agent_space_arn` del despliegue de la primera parte.

Paso 1: Configurar el ID de la cuenta de servicio

En`terraform.tfvars`, configura el ID de tu cuenta de servicio:

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

Paso 2: Configurar el ARN del espacio de agentes

Copie el `agent_space_arn` valor de la salida de la parte 1 (paso 2) y configúrelo `terraform.tfvars`:

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

Los recursos de la cuenta de servicio utilizan este valor para establecer el ámbito de la política de confianza en el rol de la cuenta secundaria. Estos recursos solo se crean cuando se establece este valor.

Paso 3: Configurar el proveedor `aws.service`

En`main.tf`, configure el alias del `aws.service` proveedor con las credenciales de la cuenta de servicio. Puede usar un perfil con nombre o un rol asumido:

Uso de un perfil:

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

O usando assume rol:

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

Paso 4: Implementar

Aplique la configuración actualizada:

```
terraform apply
```

Esto crea los siguientes recursos en la cuenta de servicio:

- Un rol de IAM (`DevOpsAgentRole-SecondaryAccount-TF`) que confía en el espacio de agentes de la cuenta de supervisión
- Una función echo Lambda (`echo-service-tf`) como servicio de ejemplo

También crea una AWS asociación de origen en la cuenta de monitoreo que vincula la cuenta de servicio.

Paso 5: Verificar el despliegue

Pruebe el servicio de eco para confirmar que la función Lambda se implementó correctamente:

```
aws lambda invoke \
  --function-name echo-service-tf \
  --payload '{"test": "hello world"}' \
```

```
  --profile <your-service-account-profile> \
  --region <REGION> \
  response.json
cat response.json
```

Resolución de problemas

Retrasos de propagación de IAM

- La configuración incluye un intervalo de 30 segundos `time_sleep` entre la creación del rol de IAM y la creación de Agent Space. El servicio de DevOps agente valida la política de confianza del rol de operador durante la creación de Agent Space, y esto puede fallar si IAM no se ha propagado por completo. Si sigue apareciendo errores en la política de confianza, espere un minuto y `terraform apply` vuelva a ejecutarlo: las funciones de IAM ya existen y la solicitud continuará donde la dejó.

Errores de permisos

- Compruebe que sus AWS credenciales tienen los permisos de IAM necesarios para crear funciones y políticas.
- Comprueba que las condiciones de la política de confianza coincidan con tu ID de cuenta.

Se produce un error en el despliegue entre cuentas

- El `aws.service` proveedor debe estar configurado con las credenciales de la cuenta de servicio. Utilice un perfil con nombre o un bloque de asunción de roles.
- Compruebe que el `agent_space_arn` valor coincide con el ARN de la salida de la parte 1.

No se encontró el tipo de recurso de Terraform

- Compruebe que tiene la versión del `awscc` proveedor `~> 1.0` o posterior. Los `awscc_devopsagent_association` recursos `awscc_devopsagent_agent_space` y los recursos requieren el proveedor de AWS Cloud Control.

Limpieza

Para eliminar todos los recursos, destrúyelos en orden inverso si has desplegado la segunda parte:

```
./cleanup.sh
```

O manualmente:

```
terraform destroy
```

Advertencia: esto elimina permanentemente tu espacio de agente y todos los datos asociados. Asegúrese de haber hecho una copia de seguridad de toda la información importante antes de continuar.

Consideraciones de seguridad

- La configuración de Terraform crea funciones de IAM con políticas de confianza que solo permiten que el director del `aidevops.amazonaws.com` servicio las asuma.
- Las políticas de confianza incluyen condiciones que restringen el acceso a su AWS cuenta específica y al ARN del espacio de agente.
- Todas las políticas siguen el principio del privilegio mínimo. Revise y personalice las políticas de IAM en función de los requisitos de seguridad de su organización.
- El rol multicuenta (`DevOpsAgentRole-SecondaryAccount-TF`) usa un nombre fijo y se limita a un ARN de espacio de agentes específico.

Siguientes pasos

Una vez que hayas desplegado tu AWS DevOps agente mediante Terraform:

1. Obtenga más información sobre la gama completa de funciones del DevOps agente en la Guía del usuario del AWS DevOps agente.
2. Considere la posibilidad de integrar la implementación de Terraform en sus CI/CD procesos para automatizar la administración de la infraestructura.

Recursos adicionales

- AWS DevOps Guía del usuario del agente
- Ejemplo de repositorio de Terraform
- Guía de incorporación de CLI

Trabajando con el DevOps agente

Trabajando con DevOps un agente

AWS DevOps El agente trabaja junto con su equipo de operaciones durante todo el ciclo de vida del incidente, desde la detección hasta la investigación, la recuperación y la prevención. En los siguientes temas se describe cómo utilizar DevOps Agent para gestionar cada fase de este ciclo de vida.

Respuesta autónoma a incidentes

Cuando se detecta un incidente, ya sea mediante una integración integrada con su sistema de emisión de entradas, un webhook de sus herramientas de supervisión o un disparador manual, el DevOps agente inicia automáticamente una investigación. El agente analiza las métricas, los registros, las trazas, los cambios de código y el historial de despliegues para determinar la causa raíz y proponer un plan de mitigación. Si necesita ayuda adicional, puede pasar directamente a AWS Support desde la aplicación web DevOps Agent Space, que comparte automáticamente el contexto de la investigación con los ingenieros de soporte para que no tenga que repetir lo que el agente ya ha encontrado. Para obtener más información, consulte the section called “Respuesta autónoma a incidentes”.

Tareas bajo demanda DevOps

En cualquier momento del ciclo de vida del incidente, puede interactuar con el DevOps agente a través de una interfaz de chat conversacional. Haga preguntas sobre sus AWS recursos, el estado del sistema, el estado de las alarmas y el historial de despliegue utilizando un lenguaje natural. El chat tiene en cuenta el contexto: cuando visualizas una investigación específica, puedes hacer que el agente explore determinadas hipótesis, se centre en registros específicos o actualice su análisis de la causa raíz. También puede consultar las configuraciones de los recursos, las tendencias de errores y la información de las investigaciones en todo su entorno sin tener que navegar de una consola a otra. Para obtener más información, consulte the section called “DevOps Tareas bajo demanda”.

Prevención proactiva de incidentes

Tras resolver los incidentes, DevOps Agent analiza los patrones de su historial de investigación para generar recomendaciones que eviten futuros incidentes y reduzcan el tiempo medio de detección. Las recomendaciones abarcan cuatro áreas: la postura de observabilidad, las brechas en las pruebas, los cambios en el código y la arquitectura de la infraestructura. El agente realiza evaluaciones semanales y actualiza las recomendaciones a medida que se producen nuevos incidentes. Puedes aceptar, rechazar o hacer un seguimiento de las recomendaciones, y el agente aprenderá de tus comentarios para refinar las sugerencias futuras. Para obtener más información, consulte the section called “Prevención proactiva de incidentes”.

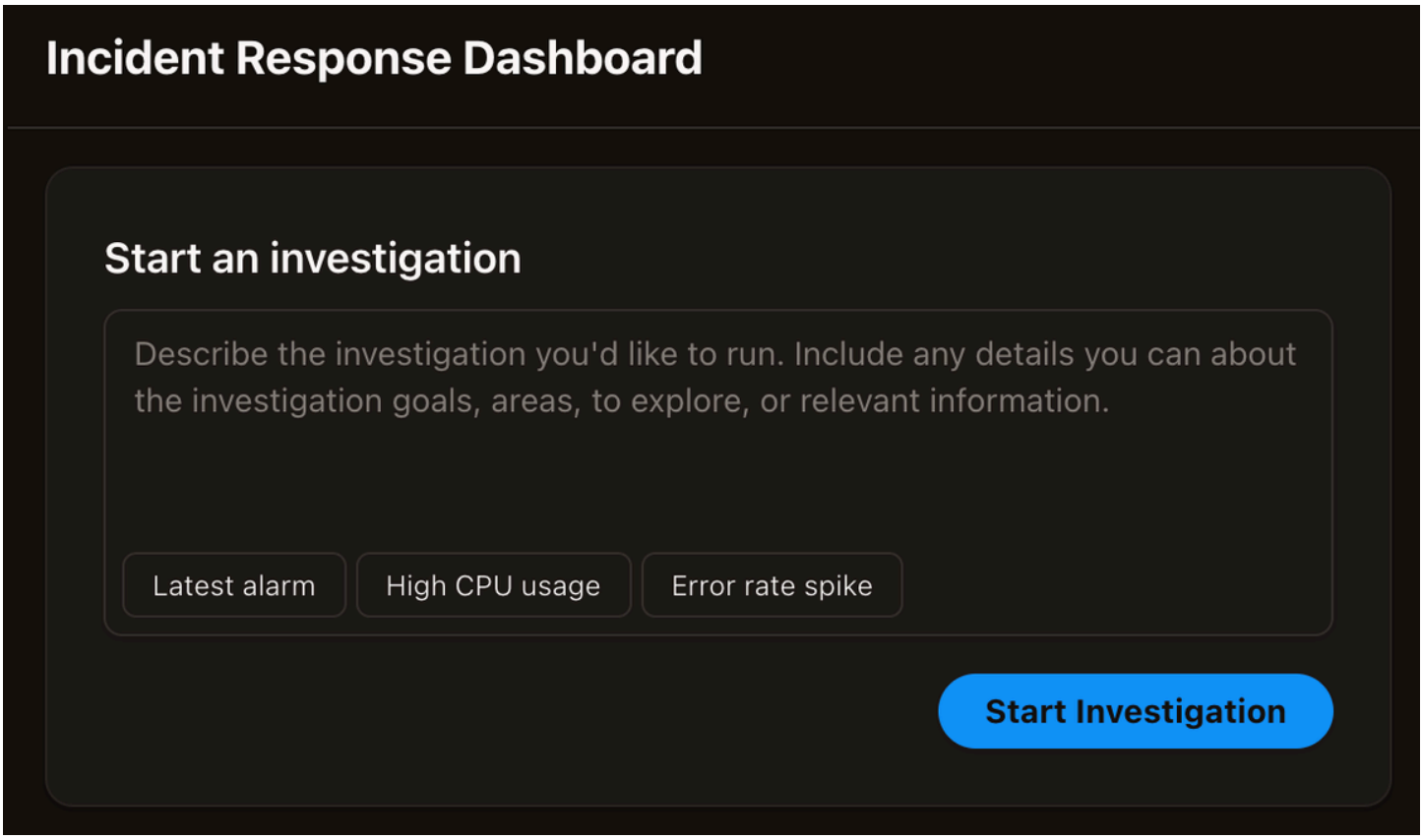
Respuesta autónoma a incidentes

Inicio de investigaciones

Las investigaciones sobre la respuesta a los incidentes se pueden iniciar de tres maneras.

- Integraciones integradas: puede conectar un DevOps Agent Space a los sistemas de venta de entradas, por ejemplo, ServiceNow mediante integraciones integradas. Una vez conectado, los DevOps agentes iniciarán automáticamente las investigaciones sobre la respuesta a los incidentes a partir de los tickets de soporte, y su DevOps agente proporcionará actualizaciones con sus principales hallazgos, análisis de las causas fundamentales y planes de mitigación en el ticket de origen.
- Webhooks: puedes usar webhooks para enviar eventos al AWS DevOps agente. Por ejemplo, puedes usar webhooks para iniciar investigaciones de respuesta a incidentes a partir de PagerDuty tickets o alarmas de Grafana.
- Manualmente: puedes iniciar manualmente las investigaciones de respuesta a incidentes desde la pestaña Respuesta a incidentes de cualquier aplicación web de DevOps Agent Space. Puedes introducir un texto libre que describa el incidente que quieres que investigue tu DevOps agente y este creará un plan de investigación, recopilará las conclusiones, determinará la causa raíz y ofrecerá la elaboración de un plan de mitigación. También puede elegir entre varios puntos de partida preconfigurados para iniciar rápidamente la investigación: la última alarma para investigar la última alarma activada y analizar las métricas y los registros subyacentes para determinar la causa principal, uso elevado de la CPU para investigar las métricas de uso elevado de la CPU en los recursos informáticos e identificar qué procesos o servicios consumen recursos excesivos, o el aumento de la tasa de errores para investigar el reciente aumento de las tasas de error

de las aplicaciones mediante el análisis de las métricas, los registros de las aplicaciones y la identificación del origen de los errores.



Cuando hagas clic en «Iniciar investigación», se te pedirá que proporciones algunos detalles adicionales para ayudar al agente a centrar su trabajo. El cuadro de diálogo de investigación incluye los siguientes campos:

- Detalles de la investigación: rellenados previamente con tu descripción. Puedes editarlo para afinar el alcance de la investigación.
- Punto de partida de la investigación: si lo desea, describa una alarma, una métrica, un fragmento de registro u otro punto de partida específico para el agente.
- Fecha y hora del incidente: se rellena automáticamente con la hora actual en formato UTC. Ajusta si el incidente ocurrió antes.
- Ponle un nombre a tu investigación: se genera automáticamente con una marca de tiempo. Puedes personalizarlo (máximo 400 caracteres).
- Prioridad: selecciona la prioridad de la investigación en el menú desplegable (la opción predeterminada es media).

Revisa y ajusta estos campos según sea necesario y, a continuación, haz clic en «Empezar a investigar...» para empezar. A continuación, accederás a la página de detalles de la investigación, ¡donde podrás ver a tu DevOps agente en acción!

Clasificación de incidentes

La fase de clasificación es la primera etapa del sistema de respuesta a incidentes del AWS DevOps agente. Cuando se desencadena un evento externo, como una alarma de Datadog, una notificación de incidente o un problema de Dynatrace ServiceNow, AWS DevOps Agent lo procesa automáticamente en cuestión de segundos para determinar si debe investigarse de forma independiente o estar vinculado a una investigación existente.

La función principal de la fase de clasificación es la correlación de incidentes: identificar los incidentes relacionados y consolidarlos en una sola investigación para evitar la duplicación de trabajo y el desperdicio de recursos. Cuando se produce un nuevo incidente, el AWS DevOps agente lo analiza junto con las investigaciones en curso en un período retrospectivo (normalmente 20 minutos). Mediante un análisis basado en la IA, examina factores como las similitudes de los componentes, la región geográfica y los patrones temporales para determinar las relaciones entre los incidentes.

AWS DevOps El agente toma una de estas dos decisiones:

- Vinculado: correlaciona el incidente con una investigación en curso y envía un mensaje orientativo a esa investigación con el contexto del nuevo incidente.
- Continuar: programa una nueva investigación independiente sobre el incidente.

Ver las decisiones de clasificación

Cuando los incidentes están relacionados, la investigación principal recibe un mensaje orientativo con los detalles del incidente vinculado y el razonamiento de correlación. En su aplicación web AWS DevOps Agent Space, verá el estado VINCULADO junto con un razonamiento de correlación que explica por qué se vincularon los incidentes. La investigación principal muestra una lista de todos los incidentes relacionados, lo que te permite ver todos los problemas relacionados que se están investigando juntos. Tu sistema de tickets externo (ServiceNow PagerDuty, etc.) y tu canal de comunicación (Slack) recibirán una notificación en la que se indicará que el incidente está relacionado, junto con un razonamiento de correlación.

Cómo desvincular los incidentes y las reglas de correlación personalizadas

Si AWS DevOps Agent correlaciona los incidentes de forma incorrecta, puede desvincularlos manualmente a través de la aplicación web AWS DevOps Agent Space. Esto reprogramará el incidente no vinculado como una investigación independiente. También puedes proporcionar reglas de correlación personalizadas para guiar al AWS DevOps agente creando una habilidad de AWS DevOps agente que contenga tu lógica de correlación y asociándola a la fase de clasificación.

Solicita apoyo humano

AWS DevOps El agente puede conectarse directamente con AWS Support para agilizar el proceso de respuesta a incidentes. Cuando necesite ayuda adicional de AWS Support, desde su aplicación web DevOps Agent Space puede crear casos de soporte que compartan automáticamente el contexto de la investigación con los ingenieros de AWS soporte, lo que reduce el tiempo necesario para explicar su problema.

Funcionamiento

Al investigar un incidente, AWS DevOps Agent crea un registro completo de sus análisis, que incluye:

- Resultados de la investigación de la causa raíz
- Se analizaron las métricas, los registros y las trazas
- Se revisaron los cambios de código y el historial de implementación
- Se recomiendan medidas correctivas
- Cronología de los eventos y del comportamiento del sistema

Puedes llevar tu investigación a AWS Support directamente desde la aplicación web AWS DevOps Agent Space. Cuando lo hace, AWS DevOps Agent pasa automáticamente su registro de investigación a AWS Support, lo que proporciona al ingeniero de soporte un contexto completo sobre su investigación sin necesidad de que usted recopile y explique los detalles manualmente.

Conversando con AWS Support

Una vez que haya creado un caso de soporte, podrá comunicarse con AWS Support en una ventana de chat independiente dentro de su aplicación web AWS DevOps Agent Space. Esto le permite:

- Hable de su problema con los ingenieros de AWS Support junto con el cronograma de investigación de su AWS DevOps agente

- Consulte el análisis automatizado del AWS DevOps agente y la orientación experta del equipo de AWS soporte en la misma interfaz
- Comparta fácilmente información o aclaraciones adicionales según sea necesario

La experiencia de chat permite que la investigación de su AWS DevOps agente y la conversación de AWS Support estén fácilmente accesibles, lo que permite una colaboración y una resolución más rápidas.

Requisitos del plan Support

Su capacidad para crear casos de soporte e interactuar con ellos a través de AWS DevOps Agent depende de su plan de AWS soporte. Consulte la guía del usuario de Support Plans para obtener más información sobre sus derechos.

Nota: Los clientes de Basic Support no pueden crear casos de soporte técnico y, por lo tanto, no pueden llevar las investigaciones de los AWS DevOps agentes a AWS Support Los clientes de Support Developer AWS Support pueden crear casos a través de AWS DevOps Agent, pero deben visitar el Centro de soporte para comunicarse con los ingenieros de soporte, ya que el soporte para desarrolladores no incluye soporte por chat. Todos los demás planes pueden utilizar la experiencia de chat integrada en Agent. AWS DevOps Para obtener detalles completos sobre los derechos del plan de soporte, incluidos los tiempos de respuesta y la gravedad de los casos disponibles, consulte la Guía del usuario de AWS Support Plans.

Qué información se comparte con AWS Support

Al crear un caso de soporte desde la aplicación web AWS DevOps Agent Space, la siguiente información se comparte automáticamente con AWS Support:

- Cronología de la investigación: registro cronológico del AWS DevOps análisis del agente
- Información sobre los recursos: Recursos afectados AWS
- Datos de observabilidad: métricas, registros y rastros relevantes de sus herramientas de monitoreo integradas
- Cambios recientes: despliegues de código, cambios en la infraestructura y actualizaciones de configuración
- Intentos de remediación: se recomiendan acciones AWS DevOps por parte del agente
- Evaluación del impacto: alcance y gravedad del incidente

Todos los datos compartidos con AWS Support siguen sus configuraciones de seguridad y residencia de AWS datos existentes. AWS DevOps El agente solo comparte información relacionada con su investigación específica y respeta las políticas de gobierno de datos de su organización.

Introducción

Para usar la integración de AWS DevOps Agent's AWS Support:

1. Asegúrese de tener un plan AWS Support activo.
2. Compruebe que los permisos de IAM de su AWS DevOps agente incluyan la creación de casos de soporte (support:CreateCase, support:DescribeCases).
3. Cuando AWS DevOps Agent esté investigando un problema y necesite asistencia de AWS soporte, elija Ask for human support en su aplicación web DevOps Agent Space.
4. Revisa el resumen de la investigación que se compartirá con AWS Support.
5. Seleccione la gravedad del caso adecuada en función de sus derechos en relación con el plan de soporte.
6. Envíe el caso: el AWS DevOps agente incluye automáticamente su registro de investigación.

La ventana de chat se abre automáticamente, lo que le permite empezar a colaborar con AWS Support de forma inmediata.

Prevención proactiva de incidentes

AWS DevOps El agente analiza los patrones de sus investigaciones de incidentes para ofrecer recomendaciones específicas que mejoren continuamente su postura operativa y eviten futuros incidentes. Acceda a la prevención proactiva de incidentes a través de la página Ops Backlog de la aplicación web Operator.

Cómo funciona la prevención proactiva de incidentes

AWS DevOps El agente evalúa las investigaciones de incidentes recientes para identificar mejoras duraderas a fin de prevenir futuros incidentes y acelerar el tiempo medio de detección (MTTD). El agente analiza varios incidentes para identificar recomendaciones que puedan prevenir toda clase de incidentes en el futuro, centrándose en las recomendaciones más impactantes para garantizar que sean procesables.

De forma predeterminada, el agente realiza evaluaciones automáticamente cada semana. Puede pausar la programación si prefiere ejecutar las evaluaciones solo a pedido. Las evaluaciones manuales están siempre disponibles, lo que resulta útil cuando una investigación reciente justifica una revisión rápida de las mejoras recomendadas.

El agente identifica las mejoras en cuatro categorías, que se muestran en el cuadro de categorización de recomendaciones de la página del registro de operaciones pendientes:

- Observabilidad: recomendaciones para mejorar la supervisión, las alertas, el registro y la visibilidad del sistema a fin de detectar problemas de forma más rápida y precisa.
- Infraestructura: recomendaciones para optimizar las configuraciones de los recursos, el ajuste de la capacidad y la resiliencia de la arquitectura.
- Gobernanza: recomendaciones para reforzar los procesos de implementación, las mejoras en los procesos, las prácticas de prueba y los controles operativos.
- Optimización del código: recomendaciones para mejorar la calidad del código de las aplicaciones, la gestión de errores y la resiliencia del código.

Esta categorización le ayuda a comprender dónde son más necesarias sus mejoras operativas y le permite priorizar las recomendaciones en función de las áreas de interés de su equipo.

Ventajas

- Evite los incidentes recurrentes: aborde las causas fundamentales de forma sistemática en lugar de responder repetidamente a los mismos tipos de problemas
- Reduzca el esfuerzo operativo: libere a su equipo de la repetitiva lucha contra incendios y céntrese en la innovación y las mejoras estratégicas
- Mejore la resiliencia del sistema: refuerce sus procesos de infraestructura, observabilidad e implementación basándose en datos de incidentes reales
- Aprenda de los patrones históricos: aproveche la información de los incidentes pasados para realizar mejoras específicas que tengan el mayor impacto

Resumen del agente

El resumen del agente en la página de registro de operaciones de la aplicación web proporciona una descripción de los resultados de la última evaluación de los incidentes recientes. El resumen explica

el número de investigaciones de incidentes analizadas, qué incidentes son similares a los anteriores y qué recomendaciones se crearon o actualizaron con nueva información.

El resumen le ayuda a comprender rápidamente lo que el agente descubrió durante su evaluación más reciente y destaca las recomendaciones más destacadas que podrían tener un mayor impacto en su postura operativa.

Controlar las evaluaciones

Puede controlar cuándo el AWS DevOps agente evalúa los incidentes y genera recomendaciones:

- Ejecutar las evaluaciones manualmente: haga clic en el botón Ejecutar ahora en la página del registro de operaciones pendientes para iniciar una evaluación de inmediato. Esto resulta útil cuando una investigación reciente justifica una revisión rápida de las mejoras recomendadas.
- Detener las evaluaciones activas: haga clic en el botón Detener la evaluación en la página del registro de operaciones pendientes para detener una evaluación que esté en curso actualmente.

Administrar las recomendaciones

AWS DevOps El agente proporciona recomendaciones en la página del registro de operaciones pendientes, donde puede revisarlas y gestionarlas:

- Ver los detalles de las recomendaciones: haga clic en una recomendación para abrir la página de detalles de la recomendación, donde podrá ver más información sobre la mejora sugerida, incluidos los incidentes en los que se basó la recomendación, los impactos esperados y los próximos pasos a seguir. Para obtener recomendaciones sobre cambios en el código, también puede consultar la especificación lista para el agente, que se puede entregar a un agente de codificación para su implementación.
- Conservar: haz clic en «Conservar» para conservar una recomendación en tu lista de pedidos pendientes y poder hacer un seguimiento de ella. Esto le permite controlar las mejoras que planea implementar y realizar un seguimiento de su progreso.
- Descartar: haz clic en «Descartar» para eliminar una recomendación de tu lista de pedidos pendientes. Cuando descartas una recomendación, puedes explicar en lenguaje natural por qué no se ajusta a tus necesidades. El agente aprende de estos comentarios y los utiliza para fundamentar sus futuras recomendaciones, asegurándose de que se adapten mejor a sus prioridades y requisitos operativos a lo largo del tiempo.

- Implementada: haga clic en «Implementada» para marcar una recomendación como completada. Esto le ayuda a realizar un seguimiento de las mejoras que se han aplicado y permite al agente medir la eficacia de sus recomendaciones a lo largo del tiempo.
- Eliminación automática: las recomendaciones que no se hayan marcado como conservadas o implementadas se pueden eliminar al cabo de aproximadamente 6 semanas si no se hubieran evitado nuevos incidentes con la aplicación de la recomendación. Esto garantiza que la página del registro de operaciones pendientes se centre en las mejoras más relevantes para sus desafíos operativos.
- Actualizaciones de las recomendaciones: las recomendaciones existentes se actualizan cuando se detectan nuevos incidentes que la recomendación habría evitado. Las actualizaciones pueden cambiar la prioridad de la recomendación o refinarla en función de los nuevos conocimientos.

Especificaciones listas para usar como agentes

Para las recomendaciones que impliquen cambios en el código o la configuración, AWS DevOps Agent puede generar una especificación lista para el agente. Esta especificación proporciona un documento estructurado que se puede entregar directamente a un agente de codificación para su implementación.

La especificación incluye:

- Enunciado del problema: un resumen del problema y su causa raíz
- Resumen de la solución: descripción detallada del enfoque recomendado
- Repositorios de destino: los repositorios específicos en los que es necesario realizar cambios
- Cambios de código: descripciones detalladas de lo que debe cambiar y por qué, con rutas de archivo específicas y consideraciones de implementación
- Requisitos de las pruebas: ¿Qué escenarios deben probarse
- Plan de implementación: un enfoque gradual para implementar los cambios

Las especificaciones listas para el agente aceleran la implementación al proporcionar a los agentes de codificación el contexto que necesitan para realizar cambios listos para la producción sin tener que recurrir a los ingenieros. back-and-forth

Implementación de recomendaciones

Para maximizar el valor de las recomendaciones proactivas de prevención de incidentes, considere las siguientes prácticas para ponerlas en práctica:

- Uso de especificaciones listas para el uso de agentes: para obtener recomendaciones con cambios en el código, utilice la especificación generada para acelerar la implementación entregándola a un agente de codificación o utilizándola como guía detallada para la implementación manual.
- Añadir recomendaciones a tu cartera de pedidos pendientes: copia las recomendaciones al sistema de gestión de proyectos o al sistema de gestión de proyectos de tu equipo para asegurarte de que se les dé prioridad junto con otros trabajos de ingeniería.
- Priorizar las recomendaciones en función del impacto: céntrese primero en las recomendaciones que aborden los tipos de incidentes más frecuentes o graves, o aquellos que afectan a los sistemas críticos.
- Seguimiento del progreso de la implementación: supervise qué recomendaciones se han implementado y mida su eficacia observando si los incidentes similares disminuyen con el tiempo.
- Coordinación con los equipos de desarrollo: comparta las recomendaciones con los equipos correspondientes que son propietarios de los sistemas afectados, asegurándose de que cuentan con el contexto y los recursos necesarios para implementar las mejoras.

DevOps Tareas bajo demanda

AWS DevOps Agent On Demand Tasks es un asistente conversacional generativo basado en inteligencia artificial (IA) que permite a los equipos de operaciones consultar la arquitectura de sus aplicaciones, analizar el estado del sistema y acceder a los conocimientos de las investigaciones utilizando un lenguaje natural. Puede hacer preguntas sobre sus AWS recursos, las métricas del sistema, el estado de las alarmas, el historial de implementación y los patrones de incidentes. El chat proporciona respuestas inmediatas basadas en los datos reales de su infraestructura y operaciones, lo que elimina la necesidad de navegar entre varias AWS consolas o herramientas de monitoreo.

El chat está integrado en toda la aplicación web DevOps Agent Space y proporciona respuestas contextuales en función de la página que esté viendo. La interfaz mantiene el historial de conversaciones, lo que le permite continuar con las discusiones anteriores y aprovechar las consultas anteriores.

Capacidades de tareas

AWS DevOps Agent On Demand Tasks ofrece funciones integrales que le ayudan a gestionar y comprender su infraestructura:

Consultas de recursos: pregunte acerca de AWS los recursos de su espacio de agente, incluidas las funciones de Lambda, las tablas de DynamoDB, las implementaciones de EKS, los certificados y las configuraciones de infraestructura. Chat puede filtrar y analizar los recursos en función de atributos como las versiones en tiempo de ejecución, la configuración de capacidad o el estado de la implementación. Por ejemplo, pregunte «¿Cuántos Lambdas utilizan Python 3.8?» o «¿Tengo algún certificado a punto de caducar?»

Análisis del estado del sistema: consulte las métricas actuales e históricas del estado del sistema, como el estado de las alarmas, las tasas de error, el uso de la CPU y la disponibilidad del servicio. El chat puede generar resúmenes de estado que abarquen períodos de tiempo específicos e identificar tendencias en el comportamiento del sistema. Haga preguntas como «¿Qué alarmas se activaron en las últimas 24 horas?» o «¿Ha habido cinco veces más errores en la última hora?»

Información sobre la investigación: acceda a la información de las investigaciones finalizadas y en curso, incluidos el análisis de la causa raíz, las hipótesis exploradas, los registros revisados y los patrones de resolución. El chat puede identificar las causas más comunes de los incidentes y ofrecer recomendaciones basadas en datos históricos. Consulta «¿Cuál fue la causa más común de los incidentes del mes pasado?» o «¿Cuál es el tiempo medio de resolución de las investigaciones finalizadas?»

Dirección de la investigación: al consultar la página de detalles de una investigación, guíe la investigación ordenando al agente que se centre en registros específicos, explore hipótesis concretas o actualice el análisis de la causa raíz. Proporcione información orientativa como «Céntrese en los registros del servicio de pago y actualice su RCA» o «explore la hipótesis de que la limitación de DynamoDB causó el problema».

Artificios de chat: genere informes y documentos estructurados, como resúmenes del estado operativo, informes de errores y análisis de incidentes. Los artefactos aparecen en un panel específico y permiten editar versiones durante la conversación.

Filtrado de recomendaciones: consulta las recomendaciones de prevención de incidentes con criterios específicos, como las recomendaciones relacionadas con servicios específicos o problemas operativos. Chat explica el impacto y las consideraciones de implementación de cada recomendación. Por ejemplo, «Muéstreme recomendaciones que eviten incidentes relacionados con

DynamoDB» o «¿Qué recomendaciones me ayudarían a detectar los problemas de latencia de las solicitudes con mayor rapidez?»

Acceder al chat

El chat está disponible como un panel persistente en la parte izquierda de la aplicación web DevOps Agent Space. La barra lateral izquierda incluye el botón + Nuevo chat, una sección de páginas para acceder a los incidentes, las operaciones pendientes y la topología, y una sección de chats en la que se muestran las conversaciones recientes. Selecciona Ver todo para ver el historial completo de conversaciones.

El chat proporciona respuestas contextuales en función del lugar al que accedas:

Topología: formule preguntas generales sobre los recursos, la arquitectura y el estado operativo de su Agent Space. Chat ofrece una visibilidad total de todas las cuentas y servicios conectados. Desde este contexto, puede consultar las configuraciones de los recursos, el historial de implementación, la información de topología y las integraciones de las herramientas de observabilidad.

Respuesta a incidentes: cuando consultes la página de respuesta a incidentes, formula preguntas sobre las tendencias de la investigación, los tiempos de resolución y los patrones de incidentes en tu espacio de agente. Chat puede analizar los datos históricos de las investigaciones para identificar las causas más comunes y las oportunidades de mejora.

Detalles de la investigación: al ver una investigación específica, Chat proporciona respuestas contextuales sobre esa investigación. Pregunte acerca de los registros revisados, las hipótesis exploradas, las conclusiones sobre las causas fundamentales y los planes de mitigación. También puede proporcionar información orientativa para orientar el enfoque de la investigación.

Prevención: en la página de prevención, consulte las recomendaciones con filtros, comprenda por qué se hicieron las recomendaciones y explore los enfoques de implementación. El chat le ayuda a priorizar y comprender el impacto de las recomendaciones de prevención de incidentes.

La interfaz de chat permanece disponible cuando cambias de página, pero el contexto cambia para proporcionar información relevante para tu vista actual. Cuando inicias una nueva conversación, comienza sin un contexto previo. Cuando continúas con una conversación existente, Chat conserva el historial completo de la conversación para las preguntas de seguimiento.

Respuestas sensibles al contexto

Chat adapta sus respuestas en función de la página que esté viendo en la aplicación web DevOps Agent Space. Este conocimiento del contexto garantiza que reciba información relevante sin necesidad de especificar el ámbito de la investigación o el recurso sobre el que está realizando la consulta.

Al ver la página de detalles de una investigación, Chat entiende automáticamente que estás preguntando acerca de esa investigación específica. Preguntas como «¿Qué registros consultaste?» o «¿Qué hipótesis exploraste?» consulte la investigación que se muestra actualmente. Cuando proporcionas información orientativa, Chat la aplica a la investigación en curso y crea una nueva versión sobre la causa raíz, si procede.

En la página de prevención, Chat entiende que le interesan las recomendaciones de prevención de incidentes. Las consultas filtran y analizan automáticamente las recomendaciones dentro del contexto de Agent Space. El sistema reconoce si se trata de recomendaciones generales o de detalles de recomendaciones específicas.

Al acceder a Chat desde la página de topología, Chat ofrece una amplia visibilidad de todos los recursos, métricas y datos históricos de su espacio de agente. Puede preguntar sobre cualquier recurso, servicio u problema operativo sin especificar el contexto de la investigación o la recomendación.

Este conocimiento del contexto elimina la necesidad de especificar repetidamente a qué investigación, recomendación o ámbito de recurso se hace referencia, lo que crea un flujo de conversación más natural.

Administración de conversaciones

El chat mantiene el historial de conversaciones para que puedas continuar con las conversaciones anteriores y hacer referencia a consultas anteriores.

Crear nuevas conversaciones: haz clic en el botón «Nueva sesión» del panel de chat para iniciar una nueva conversación sin contexto previo. Las nuevas conversaciones no transfieren información de chats anteriores, lo que te permite hacer preguntas no relacionadas sin confusión.

Acceder al historial de conversaciones: haz clic en «Historial» para ver todas las conversaciones anteriores en tu espacio de agente. Las conversaciones se organizan cronológicamente con marcas de tiempo y texto de vista previa. El historial de conversaciones se conserva durante 90 días y es privado para su cuenta de usuario en el espacio de agente.

Conversaciones continuas: seleccione cualquier conversación de su historial para reanudarla donde la dejó. El chat mantiene el contexto completo de los mensajes anteriores, lo que te permite hacer preguntas de seguimiento que hagan referencia a partes anteriores de la conversación. Cuando cambias de página mientras ves una conversación, el contexto de la conversación permanece, pero el contexto específico de la página se actualiza en función de tu ubicación actual.

Tenga en cuenta que el historial de conversaciones está aislado dentro de cada espacio de agente. Las conversaciones de un espacio de agente no son visibles ni accesibles desde otros espacios de agente. Este aislamiento garantiza que la información confidencial permanezca compartimentada de acuerdo con los límites de su organización.

Generando artefactos

AWS DevOps El agente admite los artefactos del chat, es decir, documentos estructurados y versionados que genera el agente durante una conversación. Los artefactos proporcionan un panel dedicado e interactivo en la interfaz de usuario del chat para revisar y editar el contenido generado por la IA, como los informes operativos, los resúmenes de errores y las evaluaciones de estado.

Puede solicitar artefactos desde cualquier página de la aplicación web DevOps Agent Space. Chat usa el contexto de la página actual para analizar el contenido del artefacto.

Cómo funcionan los artefactos

Cuando le pides a Chat que cree o actualice contenido, Chat genera un artefacto (normalmente un documento con formato) y lo muestra en el panel de artefactos junto a la conversación.

Generar: envía una solicitud en lenguaje natural para crear un informe o un documento. Por ejemplo, pregunta «Genera un informe de estado operativo semanal para mi Agent Space» o «Muéstrame un informe sobre mis cuatro veces de errores de la semana pasada».

Reseña: el artefacto aparece en un panel específico junto a la conversación. Puedes revisar todo el contenido sin dejar de interactuar con el chat.

Editar: solicita cambios en el artefacto a través del chat. Por ejemplo, pregunte «Añadir una sección sobre arranques en frío de Lambda» o «Actualizar el informe para incluir los datos del mes pasado». El chat crea una nueva versión del artefacto con los cambios solicitados.

Consultas de ejemplo

Los siguientes ejemplos muestran los tipos de preguntas que puedes hacerle a Chat. Estos ejemplos están organizados por caso de uso y contexto.

Consultas de generación de artefactos

Desde cualquier página de la aplicación web DevOps Agent Space:

- Genera un resumen semanal del estado operativo de mi Agent Space
- Crea un informe con todos los errores de 4xx de la semana pasada
- Crea un informe resumido de los incidentes de los últimos 30 días
- Crea un resumen de la actividad de las alarmas para el servicio de pago esta semana
- Genera un informe del historial de despliegues de los últimos 7 días
- Resuma todas las recomendaciones pendientes en un informe

Consultas de información sobre recursos

Desde cualquier página de la aplicación web DevOps Agent Space:

- ¿Cuántas funciones Lambda utilizan Python 3.8?
- ¿Tengo algún certificado a punto de caducar?
- Listar todas las tablas de DynamoDB con facturación bajo demanda
- Muéstreme los clústeres de EKS en producción
- ¿Qué funciones de Lambda no se han implementado en los últimos 90 días?
- Enumere los buckets de S3 sin activar el control de versiones
- ¿Qué instancias de RDS ejecutan la versión X de la base de datos?

Consultas sobre el estado del sistema

Desde las páginas de topología o respuesta a incidentes:

- ¿Qué alarmas se activaron en las últimas 24 horas?
- ¿Algún error de cinco veces en la última hora?
- Muéstreme las tendencias de errores de Lambda para el servicio de pago
- ¿Cuál es el uso de la CPU de mi clúster de ECS?
- ¿Hay algún objetivo defectuoso en mis balanceadores de carga?
- Muéstrame los eventos de limitación de API Gateway de ayer
- ¿Qué servicios tuvieron la tasa de error más alta la semana pasada?

- Deme un informe de salud general que abarque las últimas 24 horas

Consultas sobre herramientas de observabilidad

De la topología:

- Listar los grupos de registros de Splunk
- Muéstrame las métricas de Prometheus y sus umbrales de alarma
- ¿Qué monitores Datadog están configurados para este servicio?
- Enumere las políticas de alertas de New Relic
- Muéstrame las configuraciones del panel de control de Dynatrace

Consultas de información sobre la investigación

De la página de respuesta a incidentes:

- ¿Cuál fue la causa más común de los incidentes del mes pasado?
- ¿Cuál es el tiempo medio de resolución de las investigaciones finalizadas?
- Resuma las investigaciones de la semana pasada y su RCA
- ¿Cuántos incidentes se debieron a la ralentización de DynamoDB?
- Muéstreme las tendencias de investigación del último trimestre
- ¿Qué servicios tienen las incidencias más frecuentes?

Consultas detalladas de la investigación

De la página de detalles de la investigación:

- ¿Qué registros consultaste?
- ¿Qué hipótesis exploraste?
- ¿Qué tan arriesgada es la acción atenuante que propones?
- ¿Cuál fue la cronología de los acontecimientos ocurridos durante este incidente?
- ¿Por qué llegó a la conclusión de que esta era la causa principal?
- ¿Qué pruebas respaldan su análisis de la causa raíz?
- ¿Quién lo guió durante su investigación?

- Deme un resumen de la investigación de este incidente

Consultas de orientación de la investigación

De la página de detalles de la investigación:

- Céntrate en los registros del servicio de pago entre las 14:00 y las 15:00 UTC y actualiza tu RCA
- Explore la hipótesis de que la regulación de DynamoDB causó el problema
- Compruebe la configuración del clúster de ECS para ver si eso provocó la alarma
- Compruebe únicamente los registros de las últimas 2 horas, no de todo el día
- Investiga el aumento de errores a las 3 p.m.
- Mire los registros de API Gateway en lugar de los registros de Lambda

Consultas de recomendaciones de prevención

De la página de prevención:

- ¿Cuáles son mis tres recomendaciones principales para la prevención de incidentes?
- Mostrar recomendaciones que eviten incidentes relacionados con DynamoDB
- ¿Qué recomendaciones me ayudarían a detectar los problemas de latencia de las solicitudes con mayor rapidez?
- Enumere las mejoras de observabilidad que podrían evitar incidentes similares
- Muéstrame recomendaciones de infraestructura para el servicio de pago
- ¿Qué recomendaciones tienen el mayor impacto en la resiliencia del sistema?

Habilitar el chat en su espacio de agente

El chat está disponible en todas las aplicaciones web de DevOps Agent Space. El proceso de configuración depende de si tienes un Agent Space nuevo o existente.

Nuevos espacios para agentes

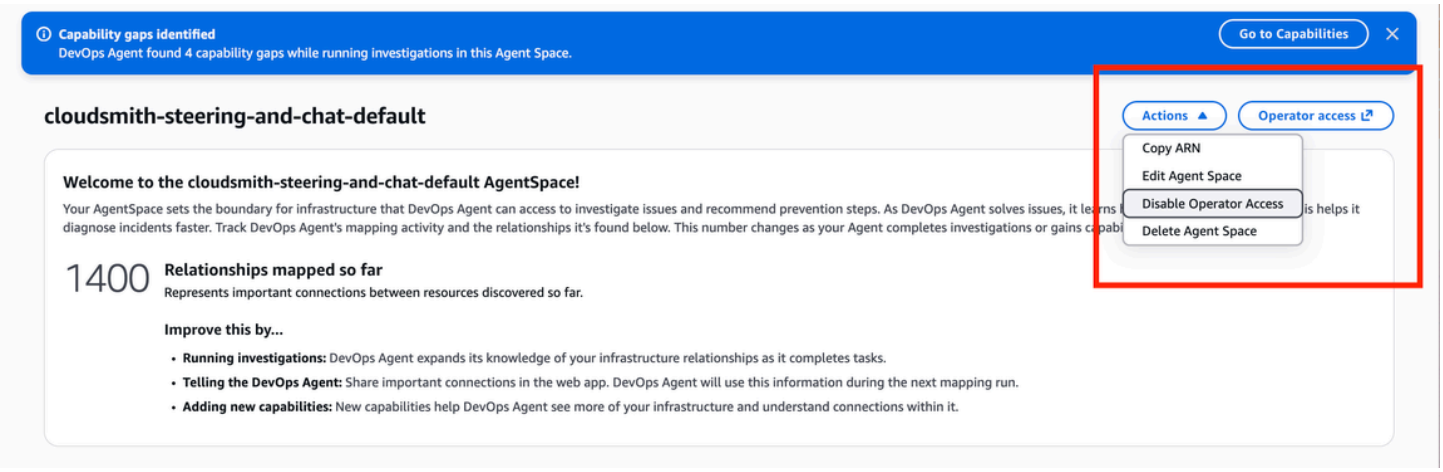
El chat se activa automáticamente al crear un nuevo espacio de agente. No se requiere ninguna configuración adicional ni configuración de permisos de IAM. Tras configurar la aplicación web DevOps Agent Space, Chat estará disponible inmediatamente como un panel persistente en la parte izquierda de cualquier página.

Espacios de agentes existentes

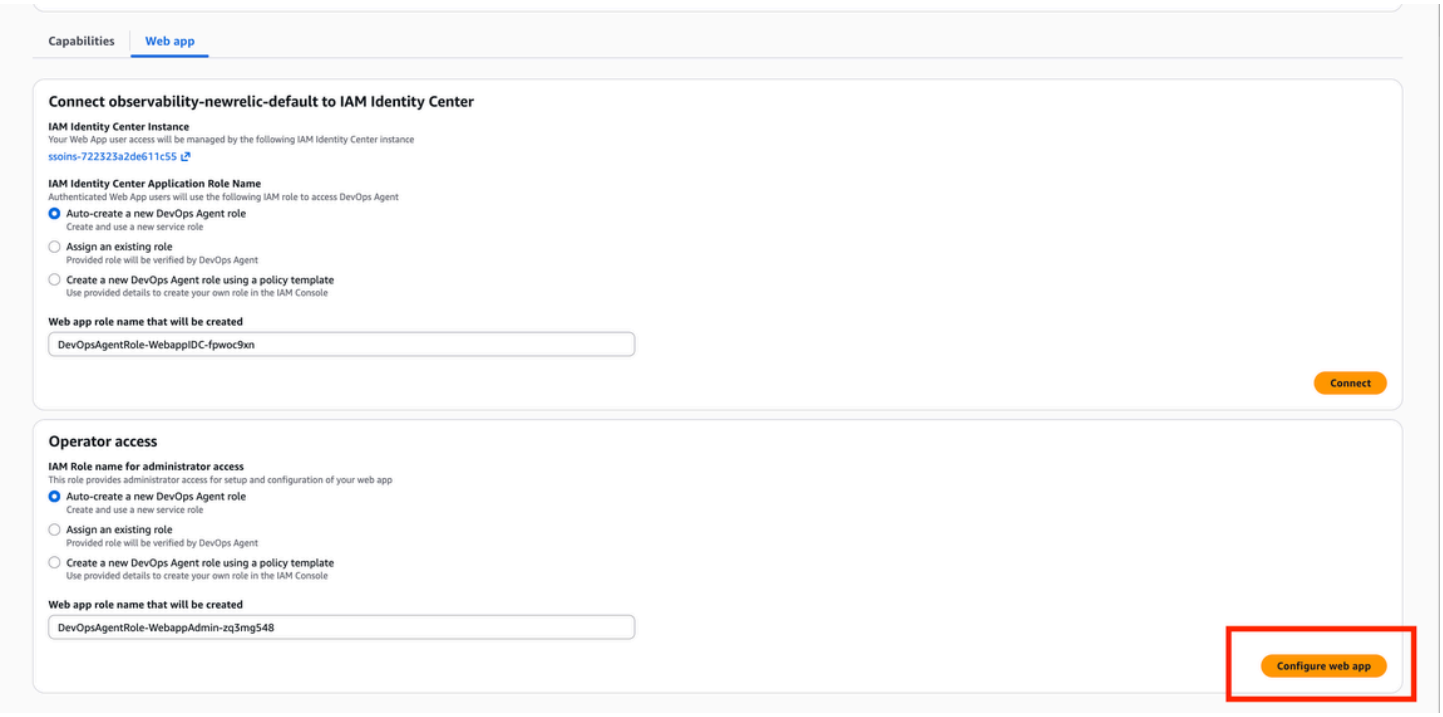
Si creó su espacio de agente antes del lanzamiento de Chat, debe habilitar los permisos de IAM necesarios. Tienes dos opciones:

Opción 1: revocar y volver a habilitar el acceso a la aplicación del operador

Ve a la consola de administración del AWS DevOps agente, localiza el menú desplegable Acción en la esquina superior derecha y desactiva la configuración actual de acceso del operador.



Luego habilite la opción de creación automática para el acceso del operador.



Esto aplica automáticamente los permisos de IAM necesarios para Chat junto con todos los demás permisos de operador actuales.

Opción 2: añadir los permisos de IAM manualmente

Añada los siguientes permisos de IAM a su función de acceso de operador actual:

- `aidevops:ListChats`— Ver el historial de conversaciones de chat
- `aidevops:CreateChat`— Crea nuevas conversaciones de chat
- `aidevops:SendMessage`— Enviar mensajes y recibir respuestas

Diríjase a la consola de AWS IAM, localice su rol de DevOps agente operador y añada estos permisos a la política de roles. El chat estará disponible inmediatamente después de añadir los permisos.

Tras completar cualquiera de las opciones, actualiza la aplicación web DevOps Agent Space y el panel de chat aparecerá en la parte izquierda de cualquier página.

Configuración de las capacidades del AWS DevOps agente

AWS DevOps Las capacidades del agente amplían la funcionalidad de su agente al conectarlo a sus herramientas e infraestructura existentes. Configure estas capacidades para permitir una investigación exhaustiva de incidentes, flujos de trabajo de respuesta automatizados y una integración perfecta con su DevOps ecosistema.

Las siguientes funciones le ayudan a maximizar la eficacia de su DevOps agente:

- AWS Configuración de acceso a EKS: permita la introspección de los clústeres, los registros de los pods y los eventos de los clústeres de Kubernetes para entornos EKS públicos y privados
- Integración con Azure: conecte las suscripciones de Azure y DevOps las organizaciones de Azure para investigar los recursos de Azure y correlacionar las DevOps implementaciones de Azure con los incidentes
- Integración de tuberías CI/CD: Connect GitHub y GitLab canalizaciones para correlacionar las implementaciones con los incidentes y rastrear los cambios de código durante las investigaciones
- Conexiones al servidor MCP: amplíe las capacidades de investigación conectando herramientas de observabilidad externas y sistemas de monitoreo personalizados mediante el protocolo Model Context
- AWS Acceso multicuenta: configure AWS cuentas secundarias para investigar los recursos de toda su organización durante la respuesta a los incidentes
- Integración de fuentes de telemetría: conecte plataformas de monitoreo como Datadog, Dynatrace, Grafana, New Relic y Splunk para un acceso integral a los datos de observabilidad
- Integración de tickets y chat: Connect ServiceNow y Slack para automatizar los flujos de trabajo de respuesta a incidentes y permitir la colaboración en equipo PagerDuty
- Configuración de webhook: permite que los sistemas externos activen automáticamente las investigaciones de los DevOps agentes mediante solicitudes HTTP
- EventBridge Integración con Amazon: incorpore el AWS DevOps agente a las aplicaciones basadas en eventos mediante el direccionamiento de los eventos del ciclo de vida de investigación y mitigación a los objetivos de Amazon EventBridge

Puede configurar cada capacidad de forma independiente en función de las necesidades específicas de su equipo y del conjunto de herramientas existente. Comience con las integraciones más importantes para su flujo de trabajo de respuesta a incidentes y, a continuación, amplíelas con capacidades adicionales según sea necesario.

Migración de la versión preliminar pública a la disponibilidad general

Si utilizó AWS DevOps Agent durante la versión preliminar pública, debe actualizar sus funciones de IAM antes del lanzamiento de la versión general. Esta guía explica cómo actualizar las funciones de supervisión y las funciones de operador en sus cuentas.

¿Qué está cambiando

1. Ya no se puede acceder a los historiales de chat bajo demanda durante la vista previa
2. Las nuevas políticas gestionadas sustituyen a las políticas disponibles durante la versión preliminar
3. Es posible que Agent Spaces tenga un ámbito de acceso a la aplicación IAM Identity Center desactualizado

Historial de chats bajo demanda obtenido de una vista previa pública

La versión de GA introduce medidas de seguridad adicionales para reforzar los controles de acceso a los historiales de chat. Como resultado de estos cambios, ya no se puede acceder a los historiales de chat bajo demanda del período de versión preliminar pública (antes del 30 de marzo de 2026). Las revistas de investigación y los hallazgos creados durante la vista previa pública no se ven afectados. Este cambio solo se aplica a las conversaciones de chat a pedido.

Nuevas políticas gestionadas

Para GA, AWS proporciona nuevas políticas administradas que sustituyen a las políticas de la era de la versión preliminar:

Tipo de rol	Quitar	Add (Suma)
Supervisión	Política administrada de `AIOpsAssistantPolicy`	Política administrada de `AIDevOpsAgentAccessPolicy`

Tipo de rol	Quitar	Add (Suma)
Operador (IAM e IDC)	Política insertada	Política administrada de `AIDevOpsOperatorAppAccessPolicy`

Además, las funciones de operador requieren políticas de confianza actualizadas y las funciones de operador de IDC requieren una nueva política en línea.

Requisitos previos

- Acceda a las AWS cuentas en las que están configuradas sus funciones de DevOps agente (cuentas principales y todas las secundarias)
- Permisos de IAM para modificar las funciones, las políticas y las relaciones de confianza
- Su ID de Agent Space, su ID de AWS cuenta y su región (visibles en la consola de DevOps Agent)

Paso 1: Actualizar las funciones de supervisión

Actualiza la función de supervisión en tu cuenta principal y en cada cuenta secundaria. Estos son los roles de Primary/Secondary origen configurados en la pestaña Capacidades de su espacio de agente (primary/secondary rol de ejemplo:`DevOpsAgentRole-AgentSpace-3xj2396z`).

1. En la consola de DevOps agentes, vaya a su espacio de agente y seleccione la pestaña Capacidades.
2. Busca la función de supervisión de tus Primary/Secondary fuentes (por ejemplo`DevOpsAgentRole-AgentSpace-3xj2396z`) y selecciona Editar.
3. En Políticas de permisos, elimina la política `AIOpsAssistantPolicy` AWS gestionada.
4. Seleccione Añadir permisos, Adjuntar políticas y adjuntar la política `AIDevOpsAgentAccessPolicy` gestionada.
5. Edita la política en línea y sustituye su contenido por lo siguiente, sustituyendo tu ID de cuenta:

```
{
    "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 política de confianza para la función de supervisión no requiere cambios. Compruebe que coincida con lo siguiente:

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

- Repita los pasos 2 a 6 para la función de supervisión en cada cuenta secundaria.

Paso 2: Actualice el rol de operador (IAM)

1. En la consola del DevOps agente, seleccione la pestaña Acceso y busque el rol de operador.
2. En la consola de IAM, elimine la política en línea existente del rol de operador.
3. Seleccione Añadir permisos, Adjuntar políticas y adjuntar la política `AIDevOpsOperatorAppAccessPolicy` gestionada.
4. Seleccione la pestaña Relaciones de confianza y elija Editar política de confianza. Sustituya la política de confianza por la siguiente, sustituyendo su ID de cuenta, región e ID de 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>"
                }
            }
        }
    ]
}
```

Paso 3: Actualizar las funciones de los operadores (IDC)

Si utiliza el Centro de identidad de IAM con DevOps un agente, actualice cada función de operador de IDC.

1. En la consola de IAM, vaya a Funciones y busque `WebappIDC` las funciones de su DevOps agente en IDC (por ejemplo,). `DevOpsAgentRole-WebappIDC-<id>`
2. Para cada función de IDC:

a. Elimine la política en línea existente.

b. Seleccione Añadir permisos, Adjuntar políticas y adjuntar la política `AIDevOpsOperatorAppAccessPolicy` gestionada.

c. Seleccione la pestaña Relaciones de confianza y elija Editar política de confianza. Sustituya la política de confianza por la siguiente, sustituyendo su ID de cuenta, región e ID de 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 nueva política en línea con los siguientes permisos, sustituyéndola por tu ID de cuenta:

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

Vuelva a conectar el centro de identidad de IAM (si corresponde)

Los espacios de agente creados durante la versión preliminar pública pueden tener una aplicación del Centro de Identidad de IAM configurada con un alcance de acceso obsoleto. En el caso de GA, el

ámbito correcto es **`aidevops:read_write`**. Si su aplicación del Centro de Identidad de IAM tiene el alcance anterior (**`awsaidevops:read_write`**), debe desconectar y volver a conectar el Centro de Identidad de IAM.

¿Cómo comprobar el alcance de su aplicación en el Centro de Identidad de IAM

Ejecute el siguiente comando AWS CLI para comprobar el alcance de la aplicación IAM Identity Center. Puede encontrar el ARN de la aplicación en la consola de IAM Identity Center, en Aplicaciones.

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

El resultado debe mostrar el alcance correcto: **`aidevops:read_write`**

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

Si se muestra el alcance **`awsaidevops:read_write`**, significa que está desactualizado. Siga los pasos que se indican a continuación para actualizarlo.

¿Cómo volver a conectar el Centro de Identidad de IAM

El alcance de acceso de una aplicación AWS gestionada del Centro de Identidad de IAM no se puede actualizar directamente. Debe desconectarse y volver a conectarse:

1. En la consola del AWS DevOps agente, vaya a su espacio de agente y seleccione la pestaña Acceso.
2. Seleccione Desconectar junto a la configuración del centro de identidad de IAM.
3. Confirme la desconexión.
4. Elija Connect para volver a configurar el Centro de identidad de IAM. El servicio crea una nueva aplicación del IAM Identity Center con el alcance correcto.

5. Reasigne los usuarios y grupos a la nueva aplicación en la consola del IAM Identity Center.

> **Important**
>
> Al desconectarse, se elimina el historial de chat y artefactos de los usuarios individuales asociado a las cuentas de usuario del IAM Identity Center. Los usuarios deberán volver a iniciar sesión después de volver a conectarse.

Verificación

Tras completar todos los pasos:

1. Regrese a la consola del DevOps agente y compruebe que no aparecen errores de permisos en la pestaña Agent Space Access.
2. Pruebe la aplicación web del operador para confirmar que se carga y funciona correctamente.
3. Si utilizas IDC, verifica que los usuarios puedan autenticarse y acceder a la experiencia del operador.

Resolución de problemas

Errores de permiso denegado tras la migración

- Compruebe que `AIOpsAssistantPolicy` se haya eliminado y que `AIDevOpsAgentAccessPolicy` esté asociado a las funciones de supervisión.
- Compruebe que se hayan eliminado las políticas integradas antiguas y `AIDevOpsOperatorAppAccessPolicy` que estén asociadas a las funciones de los operadores.
- Compruebe que las políticas de confianza de los operadores incluyan`sts:TagSession`.
- Confirme que ha sustituido todos los valores de marcador de posición (`<account-id><region>`,,`<agentspace-id>`) por valores reales.

Las cuentas secundarias no funcionan

- La función de supervisión de cada cuenta secundaria debe actualizarse de forma independiente. Inicie sesión en cada cuenta y repita el paso 1.

Fallos de autenticación de IDC

- Compruebe que la política de confianza de IDC incluya tanto la `sts:TagSession` sentencia`sts:AssumeRole`/como la `TrustedIdentityPropagation` sentencia.
- Confirme la política en línea con `sso:ListInstancessso:DescribeInstance`, y `identitystore:DescribeUser` se creó.

Falta el historial de chats bajo demanda tras la migración

- No se podrá acceder a los historiales de chat bajo demanda del período de vista previa pública tras su lanzamiento en GA. Este comportamiento es de esperar debido a las medidas de seguridad mejoradas introducidas en Georgia. Los diarios de investigación y los resultados de la versión preliminar pública no se ven afectados.

AWS Configuración de acceso a EKS

Puede permitir que AWS DevOps Agent investigue los problemas en sus clústeres de Amazon EKS ejecutando `kubectl` comandos de solo lectura en clústeres públicos y privados. Puede conectar cualquier número de clústeres de EKS al mismo espacio de agente.

Una vez conectado, el agente puede ayudar a diagnosticar problemas operativos en los clústeres: describe los recursos, recupera los registros de los módulos, inspecciona los eventos del clúster, comprueba el estado de los nodos y mucho más. El agente no puede crear, modificar ni eliminar ningún recurso del clúster.

Requisitos previos

Antes de configurar el acceso a EKS, asegúrese de que el modo de autenticación del clúster de EKS incluya la API de EKS. Puede comprobarlo en la pestaña Acceso de la consola de Amazon EKS. Si el modo no incluye la API EKS, seleccione un modo que sí la incluya antes de continuar.

Configuración

Estos pasos deben completarse desde la consola de Amazon EKS para cada clúster para el que desee crear una entrada de acceso. Puede encontrar el ARN de su rol de IAM en su espacio de agente (the section called “Creación de un espacio de agentes”consulte) en Capacidades > Nube > Fuente principal > Editar.

1. Ve a la pestaña Acceso. Si el modo de autenticación ya indica API EKS, puede añadir entradas de acceso. De lo contrario, seleccione un modo que incluya la API EKS.
2. En la pestaña Acceso, cree una nueva entrada de acceso de IAM. Copie el ARN del rol de IAM de la fuente principal en la nube e introdúzcalo como el principal de IAM para la entrada de acceso. Haga clic en Next (Siguiente).
3. Selecciona la política de AIOps AssistantPolicy acceso a Amazon AWS gestionado y selecciona Clúster para el ámbito de acceso. (Como alternativa, si quieres que el agente acceda solo a determinados espacios de nombres, selecciona los espacios de nombres de Kubernetes que desees). Haz clic en Añadir política y, a continuación, en Siguiente.
4. Revise los cambios y confirme que se eligieron la política de entrada de acceso y el rol de IAM correctos, y cree su entrada de acceso haciendo clic en «Crear».

Para comprobar que el acceso a EKS se configuró correctamente, navegue hasta la aplicación Operator e inicie una nueva investigación y formule al agente una pregunta sobre su clúster, como «incluir todos los pods en el espacio de nombres predeterminado» o «mostrarme los eventos recientes de mi clúster».

Resolución de problemas

Si el agente no puede acceder a tu clúster, comprueba que la entrada de acceso utiliza el ARN del rol de IAM correcto que se muestra en el cuadro de diálogo de configuración y que se adjunta la política de acceso de AIOpsAssistantPolicyAmazon.

Conexión de Azure

La integración con Azure permite a AWS DevOps Agent investigar los recursos de su entorno de Azure y correlacionar las implementaciones en DevOps canalización de Azure con los incidentes operativos. Al conectar Azure, el agente obtiene visibilidad de su infraestructura de Azure y puede realizar un análisis de la causa raíz tanto en los recursos de Azure como en AWS los de Azure.

La integración de Azure consta de dos capacidades independientes:

- Recursos de Azure: permite al agente descubrir e investigar los recursos de la nube de Azure, como máquinas virtuales, clústeres de Azure Kubernetes Service (AKS), bases de datos y componentes de red. El agente usa Azure Resource Graph para consultar sus recursos durante la investigación de incidentes.

- Azure DevOps: permite al agente acceder a los DevOps repositorios de Azure y al historial de ejecución de las canalizaciones. El agente puede correlacionar los cambios de código y las implementaciones con los incidentes para ayudar a identificar las posibles causas fundamentales.

Cada capacidad se registra a nivel de AWS cuenta y, a continuación, se puede asociar a espacios de agentes individuales.

Métodos de registro

AWS DevOps El agente admite dos métodos para conectarse a Azure:

- Consentimiento del administrador: un flujo simplificado basado en el consentimiento en el que se autoriza la aplicación AWS DevOps Agent Entra en su inquilino de Azure. En la consola, aparece como la opción de consentimiento del administrador. Este método requiere iniciar sesión con una cuenta que tenga permiso para otorgar el consentimiento de administrador en Microsoft Entra ID.
- Registro de aplicaciones: un enfoque autogestionado en el que puede crear su propia aplicación Entra con credenciales de identidad federadas mediante Outbound Identity Federation. En la consola, aparece como la opción de registro de aplicaciones. Este método es adecuado cuando se necesita un mayor control sobre la configuración de la aplicación o cuando los permisos de consentimiento del administrador no están disponibles.

Ambos métodos ofrecen las mismas capacidades. Puede usar uno o ambos métodos en la misma AWS cuenta.

Limitaciones conocidas

- Consentimiento del administrador: una AWS cuenta por inquilino de Azure: cada inquilino de Azure solo puede tener su aplicación AWS DevOps Agent Entra asociada a una AWS cuenta a la vez. Para asociar el mismo inquilino a una AWS cuenta diferente, primero debe anular el registro existente.
- Registro de aplicaciones: aplicación única por registro: cada registro de aplicación debe usar una aplicación diferente (ID de cliente). No puede registrar varias configuraciones con el mismo ID de cliente.
- Azure DevOps: acceso al código fuente: la DevOps integración de Azure proporciona acceso al historial de ejecución de la canalización, independientemente de dónde esté alojado el código fuente. Sin embargo, para acceder al código fuente real, el repositorio debe estar conectado por

separado a través de un proveedor de código fuente compatible (por ejemplo,the section called "Conectando GitHub"). No se puede acceder directamente al código fuente alojado en Bitbucket a través de la DevOps integración con Azure.

Temas

- the section called "Conexión de los recursos de Azure"
- the section called "Conexión de Azure DevOps"

Conexión de los recursos de Azure

La integración de Azure Resources permite al AWS DevOps agente descubrir e investigar los recursos de sus suscripciones de Azure durante la investigación de incidentes. El agente usa Azure Resource Graph para descubrir recursos y puede acceder a las métricas, los registros y los datos de configuración de todo el entorno de Azure.

Esta integración sigue un proceso de dos pasos: registre Azure a nivel de AWS cuenta y, a continuación, asocie suscripciones específicas de Azure a Agent Spaces individuales.

Requisitos previos

Antes de conectar Azure Resources, asegúrese de tener:

- Acceso a la consola del AWS DevOps agente
- Una cuenta de Azure con acceso a la suscripción de destino
- Para el método de consentimiento del administrador: una cuenta con permiso para otorgar el consentimiento del administrador en Microsoft Entra ID
- Para el método de registro de aplicaciones: una aplicación Entra con permisos para configurar las credenciales de identidad federadas y una federación de identidades salientes habilitada en su cuenta AWS

> Nota: También puede iniciar el registro desde un espacio de agente. Vaya a Fuentes secundarias, haga clic en Agregar y seleccione Azure. Si Azure Cloud aún no está registrado, la consola le guiará primero por el proceso de registro.

Cómo registrar los recursos de Azure mediante el consentimiento del administrador

El método de consentimiento del administrador utiliza un flujo basado en el consentimiento con la aplicación administrada por el AWS DevOps agente.

Paso 1: iniciar el registro

1. Inicie sesión en la consola AWS de administración y navegue hasta la consola del AWS DevOps agente
2. Vaya a la página de proveedores de capacidades
3. Busque la sección Azure Cloud y haga clic en Registrar
4. Seleccione el método de registro con el consentimiento del administrador

Paso 2: Completar el consentimiento del administrador

1. Revisa los permisos que se solicitan
2. Haga clic para continuar: se le redirigirá a la página de consentimiento del administrador de Microsoft Entra
3. Inicia sesión con una cuenta principal de usuario que tenga permiso para otorgar el consentimiento de administrador
4. Revisa la solicitud de AWS DevOps agente y otorga tu consentimiento

Paso 3: Completar la autorización del usuario

1. Tras el consentimiento del administrador, se le solicitará la autorización de usuario para verificar su identidad como miembro del arrendatario autorizado
2. Inicie sesión con una cuenta que pertenezca al mismo inquilino de Azure
3. Tras la autorización, se le redirigirá de nuevo a la consola del AWS DevOps agente con un estado correcto

Paso 4: Asignar funciones

Consulte Asignación de roles de Azure a continuación. Busque un AWS DevOps agente al seleccionar los miembros.

Registro de recursos de Azure mediante el registro de aplicaciones

El método de registro de aplicaciones utiliza su propia aplicación Entra con credenciales de identidad federadas.

Paso 1: Iniciar el registro

1. En la consola del AWS DevOps agente, vaya a la página de proveedores de capacidades
2. Busque la sección Azure Cloud y haga clic en Registrar
3. Seleccione el método de registro de la aplicación

Paso 2: Crea y configura tu aplicación Entra

Siga las instrucciones que aparecen en la consola para:

1. Habilite la federación de identidades salientes en su AWS cuenta (en la consola de IAM, vaya a Configuración de la cuenta → Federación de identidades salientes)
2. Cree una aplicación Entra en su ID de Microsoft Entra o utilice una existente
3. Configure las credenciales de identidad federadas en la aplicación

Paso 3: Proporcione los detalles de registro

Rellene el formulario de registro con:

- ID de inquilino: su identificador de inquilino de Azure
- Nombre del inquilino: nombre visible del inquilino
- ID de cliente: el ID de aplicación (cliente) de la aplicación Entra que creó
- Audiencia: el identificador de audiencia de la credencial federada

Paso 4: Crear el rol de IAM

Al enviar el registro a través de la consola, se creará automáticamente un rol de IAM. Permite al AWS DevOps agente asumir las credenciales e `sts:GetWebIdentityToken` invocarlas.

Paso 5: Asignar funciones

Consulte Asignación de roles de Azure a continuación. Busque la aplicación Entra que creó al seleccionar los miembros.

Paso 6: Complete el registro

1. Confirme la configuración en la consola del AWS DevOps agente
2. Haga clic en Enviar para completar el registro

Asignación de roles de Azure

Tras el registro, conceda a la aplicación acceso de lectura a su suscripción de Azure. Este paso es el mismo para los métodos de consentimiento del administrador y registro de la aplicación.

1. En el Portal de Azure, navegue hasta la suscripción de destino
2. Vaya a Control de acceso (IAM)
3. Haga clic en Añadir > Añadir asignación de funciones
4. Seleccione el rol de lector y haga clic en Siguiente
5. Haga clic en Seleccionar miembros y busque la aplicación (ya sea AWS DevOps un agente para obtener el consentimiento del administrador o su propia aplicación Entra para registrar la aplicación)
6. Seleccione la aplicación y haga clic en Revisar + asignar
7. (Opcional) Para permitir que el agente acceda a los clústeres de Azure Kubernetes Service (AKS), complete la siguiente configuración de acceso a AKS.

> Requisito de seguridad: al director del servicio solo se le debe asignar la función de lector (y, opcionalmente, las funciones de solo lectura de AKS que se indican a continuación). La función de lector sirve como límite de seguridad que restringe al agente a operaciones de solo lectura y limita el impacto de los ataques indirectos de inyección inmediata. La asignación de funciones con permisos de escritura o acción aumenta considerablemente el radio de acción de las inyecciones rápidas y puede comprometer los recursos de Azure. AWS DevOps El agente solo realiza operaciones de lectura. El agente no modifica, crea ni elimina los recursos de Azure.

Configuración de acceso a AKS (opcional)

Paso 1: acceso a nivel de Azure Resource Manager (ARM)

Asigne el rol de usuario del clúster de servicios de Azure Kubernetes a la aplicación.

En Azure Portal, vaya a Suscripciones → seleccione una suscripción → Control de acceso (IAM) → Agregar asignación de funciones → seleccione la función de usuario del clúster de servicios de Kubernetes de Azure → asígnela a la aplicación (ya sea AWS DevOps agente para obtener el consentimiento del administrador o su propia aplicación Entra para registrar la aplicación).

Esto cubre todos los clústeres de AKS de la suscripción. Para abarcar clústeres específicos, en su lugar, asígnelo a nivel de grupo de recursos o de clúster individual.

Paso 2: Acceso a la API de Kubernetes

Elige una opción en función de la configuración de autenticación de tu clúster:

Opción A: Control de acceso basado en roles (RBAC) de Azure para Kubernetes (recomendado)

1. Habilite Azure RBAC en el clúster si aún no lo ha hecho: Azure Portal → Clúster de AKS → Configuración → Configuración de seguridad → Autenticación y autorización → seleccione Azure RBAC
2. Asigne una función de solo lectura: Azure Portal → Suscripciones → seleccionar suscripción → Control de acceso (IAM) → Agregar asignación de funciones → seleccionar Azure Kubernetes Service RBAC Reader → asignar a la aplicación

Esto cubre todos los clústeres de AKS de la suscripción.

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

Úselo si su clúster ya usa la configuración de autenticación predeterminada de Azure AD y prefiere no habilitar el RBAC de Azure. Esto requiere una configuración por clúster. `kubectl`

1. Guarde el siguiente manifiesto como`devops-agent-reader.yaml`:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: devops-agent-reader
rules:
  - apiGroups: [""]
    resources: ["namespaces", "pods", "pods/log", "services", "events", "nodes"]
    verbs: ["get", "list"]
  - apiGroups: ["apps"]
```

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

1. <SERVICE_PRINCIPAL_OBJECT_ID>Sustitúyalo por el ID de objeto del director de servicio. Para encontrarlo: Azure Portal → Entra ID → Aplicaciones empresariales → busque el nombre de la aplicación (ya sea AWS DevOps agente para obtener el consentimiento del administrador o su propia aplicación Entra para registrar la aplicación).
2. Aplíquelo a cada clúster:

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

> Nota: No se admiten los clústeres que utilizan solo cuentas locales (sin Azure AD). Le recomendamos que habilite la integración de Azure AD en su clúster para usar esta función.

Función personalizada con menos privilegios (opcional)

Para un control de acceso más estricto, puede crear un rol de Azure personalizado que se limite únicamente a los proveedores de recursos que utiliza el AWS DevOps agente, en lugar del amplio rol de lector:

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

Asociar una suscripción a un espacio de agente

Tras registrar Azure a nivel de cuenta, asocie suscripciones específicas a sus Agent Spaces:

1. En la consola de AWS DevOps agentes, seleccione su espacio de agente
2. Ve a la pestaña Capacidades
3. En la sección Fuentes secundarias, haga clic en Agregar
4. Selecciona Azure
5. Proporcione el identificador de suscripción de la suscripción de Azure que desee asociar
6. Haga clic en Agregar para completar la asociación

Puede asociar varias suscripciones al mismo espacio de agente para que el agente tenga visibilidad en todo el entorno de Azure.

Administrar las conexiones de Azure Resources

- Visualización de las suscripciones conectadas: en la pestaña Capacidades, la sección Fuentes secundarias muestra todas las suscripciones de Azure conectadas.
- Eliminar una suscripción: para desconectar una suscripción de un Agent Space, selecciónela en la lista de fuentes secundarias y haga clic en Eliminar. Esto no afecta al registro a nivel de cuenta.
- Eliminar el registro: para eliminar por completo el registro en la nube de Azure, vaya a la página de proveedores de capacidades y elimine el registro. Primero se deben eliminar todas las asociaciones de Agent Space.

Conexión de Azure DevOps

DevOps La integración con Azure permite a AWS DevOps Agent acceder a los repositorios y al historial de ejecución de las canalizaciones de su DevOps organización de Azure. El agente puede correlacionar los cambios de código y las implementaciones con los incidentes operativos para ayudar a identificar las posibles causas fundamentales.

> Nota: DevOps Las canalizaciones de Azure pueden usar código fuente de Azure Repos o Bitbucket. GitHub La DevOps integración de Azure proporciona acceso al historial de ejecución de la canalización, independientemente del proveedor de origen. Sin embargo, para acceder al código fuente real durante las investigaciones, el repositorio debe estar conectado por separado mediante una integración compatible, por ejemplothe section called “Conectando GitHub”. No se puede acceder directamente al código fuente de Bitbucket a través de esta integración.

Esta integración sigue un proceso de dos pasos: registra Azure DevOps a nivel de AWS cuenta y, a continuación, asocia proyectos específicos a espacios de agente individuales.

Requisitos previos

Antes de conectar Azure DevOps, asegúrese de tener:

- Acceso a la consola del AWS DevOps agente
- Una DevOps organización de Azure con al menos un proyecto que contenga un repositorio y un historial de canalizaciones

- Permisos para agregar usuarios a su DevOps organización de Azure
- Para el método de consentimiento del administrador: una cuenta con permiso para otorgar el consentimiento del administrador en Microsoft Entra ID
- Para el método de registro de aplicaciones: una aplicación Entra con permisos para configurar las credenciales de identidad federadas y una federación de identidades salientes habilitada en su cuenta AWS

> Nota: También puede iniciar el registro desde un espacio de agente. Ve a la sección Pipelines, haz clic en Agregar y selecciona Azure DevOps. Si Azure aún no DevOps está registrado, la consola le guiará primero por el proceso de registro.

Registrar Azure con DevOps el consentimiento del administrador

El método de consentimiento del administrador utiliza un flujo basado en el consentimiento con la aplicación administrada por el AWS DevOps agente.

Paso 1: iniciar el registro

1. Inicie sesión en la consola AWS de administración y navegue hasta la consola del AWS DevOps agente
2. Vaya a la página de proveedores de capacidades
3. Busque la DevOps sección de Azure y haga clic en Registrar
4. Introduzca el nombre de su DevOps organización de Azure cuando se le solicite

Paso 2: Completar el consentimiento del administrador

1. Haga clic para continuar: se le redirigirá a la página de consentimiento del administrador de Microsoft Entra
2. Inicia sesión con una cuenta principal de usuario que tenga permiso para otorgar el consentimiento de administrador
3. Revisa la solicitud de AWS DevOps agente y otorga tu consentimiento

Paso 3: Completar la autorización del usuario

1. Tras el consentimiento del administrador, se le solicitará la autorización de usuario para verificar su identidad como miembro del arrendatario autorizado

2. Inicie sesión con una cuenta que pertenezca al mismo inquilino de Azure
3. Tras la autorización, se le redirigirá de nuevo a la consola del AWS DevOps agente con un estado correcto

Paso 4: Conceder el acceso en Azure DevOps

Consulte Otorgar acceso en Azure DevOps a continuación. Busque AWS DevOps Agent al agregar usuarios.

Registrar Azure DevOps mediante el registro de aplicaciones

El registro de aplicaciones se comparte entre Azure Resources y Azure DevOps. Si ya ha completado el registro de aplicaciones para los recursos de Azure, puede pasar a Conceder acceso a Azure DevOps.

Paso 1: Inicie el registro de la aplicación ADO

1. En la consola del AWS DevOps agente, vaya a la página de proveedores de capacidades
2. Busque la sección Azure Cloud y haga clic en Registrar
3. Seleccione el método de registro de la aplicación

Paso 2: Crea y configura tu aplicación Entra

Siga las instrucciones que aparecen en la consola para:

1. Habilite la federación de identidades salientes en su AWS cuenta (en la consola de IAM, vaya a Configuración de la cuenta → Federación de identidades salientes)
2. Cree una aplicación Entra en su ID de Microsoft Entra o utilice una existente
3. Configure las credenciales de identidad federadas en la aplicación

Paso 3: Proporcione los detalles de registro

Rellene el formulario de registro con:

- ID de inquilino: su identificador de inquilino de Azure
- Nombre del inquilino: nombre visible del inquilino
- ID de cliente: el ID de aplicación (cliente) de la aplicación Entra

- Audiencia: el identificador de audiencia de la credencial federada

Paso 4: Crear el rol de IAM

Al enviar el registro a través de la consola, se creará automáticamente un rol de IAM. Permite al AWS DevOps agente asumir las credenciales e `sts:GetWebIdentityToken` invocarlas.

Paso 5: Complete el registro

1. Confirme la configuración en la consola del AWS DevOps agente
2. Haga clic en Enviar para completar el registro

Paso 6: Conceder el acceso en Azure DevOps

Consulte Otorgar acceso en Azure DevOps a continuación. Busque la aplicación Entra que creó durante el registro de la aplicación al agregar usuarios.

Otorgar acceso en Azure DevOps

Tras el registro, conceda acceso a la aplicación a su DevOps organización de Azure. Este paso es el mismo para los métodos de consentimiento del administrador y registro de la aplicación.

1. En Azure DevOps, vaya a Configuración de la organización > Usuarios > Agregar usuarios
2. Busque la aplicación (ya sea AWS DevOps un agente para obtener el consentimiento del administrador o su propia aplicación Entra para el registro de la aplicación)
3. Establece el nivel de acceso en Básico
4. En Añadir a proyectos, seleccione los proyectos a los que desee que acceda el agente
5. En DevOps Grupos de Azure, seleccione Project Readers
6. Haga clic en Agregar para completar

> Requisito de seguridad: asigne solo el grupo de lectores del proyecto. El acceso de solo lectura sirve como límite de seguridad que restringe al agente a operaciones de solo lectura y limita el impacto de los ataques indirectos de inyección inmediata. La asignación de permisos de escritura o acción a grupos aumenta considerablemente el radio de acción de las inyecciones rápidas y puede comprometer los recursos de Azure. DevOps

Asociar un proyecto a un espacio de agente

Tras registrar Azure DevOps a nivel de cuenta, asocie proyectos específicos a sus Agent Spaces:

1. En la consola de AWS DevOps Agent, selecciona tu Agent Space
2. Ve a la pestaña Capacidades
3. En la sección Canalizaciones, haga clic en Añadir
4. Seleccione Azure DevOps de la lista de proveedores disponibles
5. Seleccione el proyecto en el menú desplegable de proyectos disponibles
6. Haga clic en Añadir para completar la asociación

Administrar DevOps las conexiones de Azure

- Visualización de los proyectos conectados: en la pestaña Capacidades, la sección Pipelines muestra todos los DevOps proyectos de Azure conectados.
- Eliminar un proyecto: para desconectar un proyecto de un Agent Space, selecciónelo en la sección Pipelines y haga clic en Eliminar.
- Eliminar el registro: para eliminar el DevOps registro de Azure por completo, vaya a la página de proveedores de capacidades y elimine el registro. Primero se deben eliminar todas las asociaciones de Agent Space.

Conexión a CI/CD tuberías

La integración de la canalización de CI/CD permite a AWS DevOps Agent monitorear las implementaciones y correlacionar los cambios de código con los incidentes operativos durante las investigaciones. Al conectar a sus CI/CD proveedores, el agente puede realizar un seguimiento de los eventos de despliegue y asociarlos con AWS recursos para ayudar a identificar las posibles causas fundamentales durante la respuesta a los incidentes.

AWS DevOps El agente permite la integración con CI/CD las plataformas más populares mediante un proceso de dos pasos:

1. Registro a nivel de cuenta: registre a su CI/CD proveedor una vez a nivel de cuenta AWS
2. Conexión con Agent Space: conecte proyectos o repositorios específicos a Agent Spaces individuales en función de las necesidades de su organización

Este enfoque le permite compartir los registros de CI/CD proveedores en varios espacios de agentes y, al mismo tiempo, mantener un control pormenorizado sobre los proyectos que supervisa cada espacio.

Proveedores compatibles CI/CD

AWS DevOps El agente es compatible con las siguientes CI/CD plataformas:

- GitHub— Conecta los repositorios desde GitHub.com mediante la GitHub aplicación AWS DevOps Agent.
- GitLab— Conecta proyectos desde GitLab.com, GitLab instancias gestionadas o GitLab despliegues autohospedados de acceso público.

Temas

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

Conectando GitHub

GitHub La integración permite al AWS DevOps agente acceder a los repositorios de códigos y recibir los eventos de despliegue durante la investigación de incidentes. Esta integración sigue un proceso de dos pasos: el registro a nivel de cuenta y, a continuación GitHub, la conexión de repositorios específicos a los espacios de agente individuales.

AWS DevOps El agente es compatible con GitHub instancias .com (SaaS) y GitHub Enterprise Server (autohospedadas).

Requisitos previos

Antes de conectarse GitHub, asegúrese de tener:

- Acceso a la consola de administración del AWS DevOps agente
- Una cuenta GitHub de usuario o una organización con permisos de administrador
- Autorización para instalar GitHub aplicaciones en tu cuenta u organización

Para GitHub Enterprise Server, también necesitas:

- Una instancia de GitHub Enterprise Server (versión 3.x o posterior) accesible a través de HTTPS
- La URL HTTPS de su instancia de GitHub Enterprise Server (por ejemplo,`https://github.example.com`)
- (Opcional) Una conexión privada, si su instancia de GitHub Enterprise Server no es de acceso público

Registrarse GitHub (a nivel de cuenta)

GitHub se registra a nivel de AWS cuenta y se comparte entre todos los espacios de agentes de esa cuenta. Solo necesita registrarse GitHub una vez por AWS cuenta.

Paso 1: Dirígete a los proveedores de gasoductos

1. Inicie sesión en la consola AWS de administración
2. Navegue hasta la consola del AWS DevOps agente
3. Vaya a la pestaña Capacidades
4. En la sección Pipeline, haga clic en Agregar
5. Seleccione GitHubde la lista de proveedores disponibles

Si GitHub aún no se ha registrado, se le pedirá que lo registre primero.

Paso 2: elige el tipo de conexión

En la pantalla « GitHub Registrar cuenta/organización», selecciona si te conectas como usuario u organización:

- Usuario: su GitHub cuenta personal con un nombre de usuario y un perfil
- Organización: una GitHub cuenta compartida en la que varias personas pueden colaborar en varios proyectos a la vez

Si te estás conectando a una instancia de GitHub Enterprise Server, marca la casilla Usar GitHub Enterprise Server e introduce la URL HTTPS de la instancia (por ejemplo,`https://github.example.com`).

Si su instancia de GitHub Enterprise Server no es de acceso público, si lo desea, puede configurar una conexión privada para que el AWS DevOps agente pueda acceder a su instancia de forma

segura. Para obtener más información, consulte the section called “Conexión a herramientas alojadas de forma privada”.

> **Note**
>
> No incluyas `/api/v3` ni ninguna ruta final en la URL; introduce solo la URL base.

Paso 3: Configura la aplicación GitHub

Haga clic en Enviar para iniciar el proceso de configuración de la aplicación. Los siguientes pasos varían en función de si se conecta a GitHub .com o GitHub Enterprise Server.

Para GitHub .com

1. Se te redirigirá a GitHub para instalar la GitHub aplicación AWS DevOps Agent.
2. Seleccione en qué cuenta u organización desea instalar la aplicación.
3. La aplicación permite al AWS DevOps agente recibir eventos de los repositorios conectados, incluidos los eventos de despliegue.

Para GitHub Enterprise Server

GitHub Enterprise Server usa un flujo de manifiesto de GitHub aplicaciones, que configura automáticamente una nueva GitHub aplicación en la instancia. Esto implica dos redireccionamientos a tu instancia de GitHub Enterprise Server.

1. El navegador se redirigirá a la página «Crear GitHub aplicación» de la instancia de GitHub Enterprise Server.
2. Verás el nombre de la aplicación rellenado previamente. No dudes en cambiar el nombre según sea necesario. Haz clic en Crear GitHub aplicación.
3. Se te redirigirá de nuevo a AWS DevOps Agent, que intercambia el código del manifiesto por las credenciales de la aplicación.

Paso 4: Selecciona los repositorios y completa la instalación

1. Verás la página de instalación y autorización de la GitHub aplicación.
2. Selecciona a qué repositorios quieres permitir el acceso de la aplicación:

- Todos los repositorios: otorga acceso a todos los repositorios actuales y futuros
- Selecciona solo repositorios: elige repositorios específicos de tu cuenta u organización

3. Haz clic en Instalar y autorizar.
4. Se le redirigirá de nuevo a la consola del AWS DevOps agente, donde GitHub aparecerá como registrado a nivel de cuenta.

Conectar los repositorios a un espacio de agentes

Tras registrarse GitHub a nivel de cuenta, puede conectar repositorios específicos a espacios de agente individuales:

1. En la consola de AWS DevOps agentes, selecciona tu espacio de agente
2. Ve a la pestaña Capacidades
3. En la sección Pipeline, haga clic en Agregar
4. Seleccione GitHubde la lista de proveedores disponibles
5. Seleccione el subconjunto de repositorios correspondiente a este espacio de agentes
6. Haga clic en Agregar para completar la conexión

Puede conectar diferentes conjuntos de repositorios a diferentes espacios de agentes en función de las necesidades de su organización.

Entender la aplicación GitHub

La GitHub aplicación AWS DevOps Agent:

- Solicita acceso de solo lectura a tus repositorios
- Recibe eventos de despliegue y otros eventos del repositorio
- Permite al AWS DevOps agente correlacionar los cambios de código con los incidentes operativos
- Se puede desinstalar en cualquier momento a través de su configuración GitHub

En el caso de GitHub Enterprise Server, la GitHub aplicación se crea automáticamente en la instancia durante el registro. Puede administrar el acceso al repositorio de la aplicación o desinstalarla desde Configuración > Aplicaciones > GitHub Aplicaciones instaladas. Para eliminar por completo la definición de la aplicación, ve a Configuración > Configuración del desarrollador > GitHub Aplicaciones.

Administrar GitHub las conexiones

- Actualización del acceso a los repositorios: para cambiar a qué repositorios puede acceder la GitHub aplicación, vaya a la configuración de su GitHub cuenta u organización (o a la configuración de la instancia de GitHub Enterprise Server), vaya a GitHub las aplicaciones instaladas y modifique la configuración de la aplicación AWS DevOps Agent.
- Visualización de los repositorios conectados: en la consola del AWS DevOps agente, seleccione su espacio de agente y vaya a la pestaña Capacidades para ver los repositorios conectados en la sección Pipeline.
- Eliminar una GitHub conexión: para desconectarse GitHub de un espacio de agentes, seleccione la conexión en la sección Pipeline y haga clic en Eliminar. Para desinstalar la GitHub aplicación por completo, desinstálela de la configuración de su GitHub cuenta u organización. En el caso de GitHub Enterprise Server, dado que la GitHub aplicación se crea directamente en la instancia durante el registro, si lo desea, puede limpiarla por completo realizando las dos acciones siguientes:
  - Desinstale la aplicación: vaya a Configuración **>** Aplicaciones **>** GitHub Aplicaciones instaladas, haga clic en Configurar en la aplicación y, a continuación, desinstálela.
  - Elimine la aplicación: vaya a Configuración **>** Configuración del desarrollador **>** GitHub Aplicaciones, seleccione la aplicación, vaya a la pestaña Opciones avanzadas y elija Eliminar GitHub aplicación. Advertencia: la eliminación de la GitHub aplicación es permanente y no se puede deshacer. Si la elimina, tendrá que volver a registrar GitHub Enterprise Server desde el principio en la consola del AWS DevOps agente para crear una nueva aplicación.

Conectando GitLab

GitLab La integración permite a AWS DevOps Agent monitorear los despliegues desde GitLab Pipelines para fundamentar las investigaciones causales durante la respuesta a los incidentes. Esta integración sigue un proceso de dos pasos: el registro a nivel de cuenta y, a continuación GitLab, la conexión de proyectos específicos con los espacios de agente individuales.

Registro GitLab (a nivel de cuenta)

GitLab se registra a nivel de AWS cuenta y se comparte entre todos los espacios de agentes de esa cuenta. Los espacios de agente individuales pueden entonces elegir qué proyectos específicos se van a aplicar a su espacio de agente.

Paso 1: Dirígete a los proveedores en proceso

1. Inicie sesión en la consola AWS de administración
2. Navegue hasta la consola del AWS DevOps agente
3. Vaya a la página de proveedores de capacidades (a la que se puede acceder desde el panel de navegación lateral)
4. Busque GitLaben la sección Proveedores disponibles en Pipeline y haga clic en Registrarse

Paso 2: Configurar la GitLab conexión

En la página GitLab de registro, configure lo siguiente:

Tipo de conexión: seleccione si se va a conectar como persona o como grupo:

- Personal (predeterminada): tu cuenta de GitLab usuario individual con un nombre de usuario y un perfil
- Grupo: en GitLab, los grupos se utilizan para gestionar uno o más proyectos relacionados al mismo tiempo

GitLab tipo de instancia: elige el tipo de GitLab instancia al que te vas a conectar:

- GitLab.com (predeterminado): el GitLab servicio público
- Autohospedado y de acceso público GitLab: marca la casilla Usar punto de conexión GitLab autohospedado y proporciona la URL de tu instancia GitLab

> **Note**
>
> Actualmente, solo se admiten GitLab las instancias de acceso público.

Token de acceso: proporciona un token de acceso GitLab personal:

1. En otra pestaña del navegador, inicia sesión en tu GitLab cuenta
2. Ve a la configuración de usuario y selecciona Tokens de acceso
3. Cree un nuevo token de acceso personal con los siguientes permisos:
   - `read_repository`— Necesario para acceder al contenido del repositorio

- `read_virtual_registry`— Necesario para acceder a la información del registro virtual
- `read_registry`— Necesario para acceder a la información del registro
- `api`— Necesario para el acceso a la API de lectura y escritura
- `self_rotate`- Necesario para la rotación de fichas. El AWS DevOps agente no admite esta función actualmente, pero la admitirá más adelante. Añadir ahora evita la necesidad de crear un nuevo token en el futuro.

4. Establezca la caducidad del token en un máximo de 365 días a partir de la fecha actual
5. Copia el token generado
6. Regrese a la consola del AWS DevOps agente
7. Pegue el token en el campo «Token de acceso»

Paso 3: Completar el registro

Etiquetas (opcionales): añada AWS etiquetas al GitLab registro con fines organizativos.

Haga clic en Siguiente para revisar la configuración y, a continuación, en Enviar para completar el proceso GitLab de registro. El sistema validará su token de acceso y establecerá la conexión.

Conectar proyectos a un espacio de agentes

Tras registrarte GitLab a nivel de cuenta, puedes conectar proyectos específicos a espacios de agentes individuales:

1. En la consola de AWS DevOps agentes, selecciona tu espacio de agente
2. Ve a la pestaña Capacidades
3. En la sección Pipeline, haga clic en Agregar
4. Seleccione GitLabde la lista de proveedores disponibles
5. Seleccione los GitLab proyectos relevantes para su espacio de agente
6. Haga clic en Guardar

AWS DevOps El agente supervisará estos proyectos en busca de despliegues desde GitLab Pipelines para fundamentar las investigaciones causales.

Administrar las conexiones GitLab

- Actualización del token de acceso: si su token de acceso caduca o necesita actualizarse, puede actualizarlo en la consola del AWS DevOps agente modificando el GitLab registro a nivel de cuenta.
- Visualización de los proyectos conectados: en la consola del AWS DevOps agente, seleccione su espacio de agente y vaya a la pestaña Capacidades para ver los proyectos conectados en la sección Pipeline.
- Eliminar la GitLab conexión: para desconectar GitLab los proyectos de un espacio de agentes, seleccione la conexión en la sección Pipeline y haga clic en Eliminar. Para eliminar el GitLab registro por completo, elimínelo primero de todos los Agent Spaces y, a continuación, elimine el registro a nivel de cuenta.

Conexión de servidores MCP

Los servidores Model Context Protocol (MCP) amplían las capacidades de investigación del AWS DevOps agente al proporcionar acceso a los datos de sus herramientas de observabilidad externas, sistemas de monitoreo personalizados y fuentes de datos operativos. Esta guía explica cómo conectar un servidor MCP a Agent. AWS DevOps

Requisitos

Antes de conectar un servidor MCP, asegúrese de que el servidor cumpla estos requisitos:

- Protocolo de transporte HTTP con capacidad de transmisión: solo se admiten los servidores MCP que implementan el protocolo de transporte HTTP con capacidad de transmisión.
- Soporte de autenticación: su servidor MCP debe admitir los flujos de autenticación OAuth 2.0 o la autenticación basada en claves o tokens de API.

Consideraciones de seguridad

Al conectar los servidores MCP al AWS DevOps agente, tenga en cuenta estos aspectos de seguridad:

- Lista de herramientas permitidas: debe incluir en la lista solo las herramientas específicas que su espacio de agente necesita, en lugar de exponer todas las herramientas de su servidor MCP.

Consulte Configurar las herramientas de MCP en un espacio de agente para ver cómo permitir la creación de listas de herramientas por espacio de agente.

Tenga en cuenta que la longitud máxima de cualquier herramienta MCP es de 64.

- Riesgos de inyección inmediata: los servidores MCP personalizados pueden suponer un riesgo adicional de ataques de inyección inmediata. Para obtener más información, consulte Protección contra inyecciones rápidas: AWS DevOps Agent Security.
- Acceso y herramientas de solo lectura: solo permita incluir en la lista las herramientas MCP de solo lectura y asegúrese de que las credenciales de autenticación solo permitan el acceso de solo lectura.

Consulte AWS DevOps Seguridad del agente para obtener más información sobre la inyección rápida y el modelo de responsabilidad compartida.

> **Note**
>
> Si su servidor MCP está en una red privada, consulte the section called “Conexión a herramientas alojadas de forma privada”

Registrar un servidor MCP (a nivel de cuenta)

Los servidores MCP se registran a nivel de AWS cuenta y se comparten entre todos los espacios de agentes de esa cuenta. A continuación, los espacios de agentes individuales pueden elegir qué herramientas específicas necesitan de cada servidor MCP.

Paso 1: Detalles del servidor MCP

1. Inicie sesión en la consola de AWS administración
2. Navegue hasta la consola del AWS DevOps agente
3. Vaya a la página de proveedores de capacidades (a la que se puede acceder desde el panel de navegación lateral)
4. Busque el servidor MCP en la sección de proveedores disponibles y haga clic en Registrarse
5. En la página de detalles del servidor MCP, introduzca la siguiente información:
   - Nombre: introduzca un nombre descriptivo para su servidor MCP

- URL del punto final: introduzca la URL HTTPS completa del punto final de su servidor MCP
- Descripción (opcional): añada una descripción para ayudar a identificar el propósito del servidor
- Habilitar el registro dinámico de clientes: seleccione esta casilla de verificación si desea permitir que el AWS DevOps agente se registre automáticamente en el servidor de autorización de su servidor MCP

6. Haga clic en Siguiente.

> **Note**
>
> La URL del punto final del servidor MCP se mostrará en AWS CloudTrail los registros de su cuenta.

Paso 2: Flujo de autorización

Seleccione el método de autenticación para su servidor MCP:

OAuth Credenciales de cliente: si su servidor MCP usa el flujo de credenciales de OAuth cliente:

1. Seleccione las credenciales OAuth del cliente
2. Haga clic en Siguiente.

OAuth 3LO (de tres patas OAuth): si su servidor MCP utiliza OAuth 3LO para la autenticación:

1. Seleccione 3LO OAuth
2. Haga clic en Siguiente.

Clave API: si su servidor MCP utiliza la autenticación mediante clave API:

1. Seleccione la clave de API
2. Haga clic en Siguiente.

Paso 3: Configuración de la autorización

Configure parámetros de autorización adicionales en función del método de autenticación seleccionado:

Para las credenciales OAuth del cliente:

1. ID de cliente: introduzca el ID de OAuth cliente del cliente
2. Secreto de cliente: introduzca el secreto de OAuth cliente del cliente
3. URL de Exchange: introduzca la URL del punto de conexión de intercambio del OAuth token
4. Parámetros de intercambio: introduzca los parámetros de intercambio de OAuth tokens para autenticarse con el servicio
5. Añadir ámbito: añadir OAuth ámbitos para la autenticación
6. Haga clic en Siguiente.

Para OAuth 3LO:

1. ID de cliente: introduzca el ID de cliente del OAuth cliente
2. Secreto de cliente: introduzca el secreto de OAuth cliente del cliente si su OAuth cliente lo requiere
3. URL de Exchange: introduzca la URL del punto de conexión de intercambio del OAuth token
4. URL de autorización: introduzca la URL del punto final de OAuth autorización
5. Code Challenge Support: seleccione esta casilla de verificación si su OAuth cliente admite el desafío de código
6. Añadir ámbito: añada OAuth ámbitos para la autenticación
7. Haga clic en Siguiente.

Para la clave de API:

1. Introduzca un nombre de clave de API
2. Introduce el nombre del encabezado que contendrá la clave de API en la solicitud
3. Introduce el valor de tu clave de API
4. Haga clic en Siguiente.

Paso 4: Revisa y envía

1. Revise todos los detalles de configuración del servidor MCP
2. Haga clic en Enviar para completar el registro

3. AWS DevOps El agente validará la conexión con su servidor MCP
4. Tras la validación correcta, su servidor MCP se registrará a nivel de cuenta

Configuración de las herramientas de MCP en un espacio de agentes

Tras registrar un servidor MCP a nivel de cuenta, puede configurar qué herramientas de ese servidor están disponibles en espacios de agente específicos:

1. En la consola de AWS DevOps agentes, seleccione su espacio de agente
2. Ve a la pestaña Capacidades
3. En la sección Servidores MCP, haga clic en Agregar
4. Seleccione el servidor MCP registrado que desee conectar a este espacio de agentes
5. Configure qué herramientas de este servidor MCP deberían estar disponibles en el espacio de agentes:
   - Permitir todas las herramientas: hace que todas las herramientas del servidor MCP estén disponibles
   - Seleccionar herramientas específicas: le permite elegir qué herramientas permitir en la lista
6. Haga clic en Agregar para conectar el servidor MCP a su espacio de agente

AWS DevOps El agente ahora podrá utilizar las herramientas permitidas de su servidor MCP durante las investigaciones en este espacio de agentes.

Administrar las conexiones del servidor MCP

Actualización de las credenciales de autenticación: si es necesario actualizar sus credenciales de autenticación, tendrá que volver a registrar el servidor MCP. Vaya a la página de proveedores de capacidades de la consola del AWS DevOps agente, localice su servidor MCP, elimine cualquier asociación activa y haga clic en Anular el registro. A continuación, registre su servidor MCP con las nuevas credenciales de autenticación y vuelva a crear las asociaciones necesarias con su espacio de agente.

Visualización de los servidores MCP conectados: para ver todos los servidores MCP conectados a su espacio de agente, seleccione su espacio de agente, vaya a la pestaña Capacidades y consulte la sección Servidores MCP. También puede actualizar las herramientas seleccionadas aquí.

Eliminar las conexiones del servidor MCP: para desconectar un servidor MCP de un espacio de agentes, seleccione el servidor en la sección Servidores MCP y haga clic en Eliminar. Para eliminar por completo el registro de un servidor MCP, elimínelo primero de todos los Agent Spaces y, a continuación, elimine el registro a nivel de cuenta.

Temas relacionados

- Seguridad en el agente AWS DevOps
- Configuración de un espacio de agentes
- Protección de inyección rápida

Conexión de varias AWS cuentas

AWS Las cuentas secundarias permiten al AWS DevOps agente investigar los recursos de varias AWS cuentas de la organización. Cuando sus aplicaciones abarcan varias cuentas, añadir cuentas secundarias garantiza que el agente tenga visibilidad de todos los recursos relevantes durante la investigación de los incidentes. Un mayor acceso a las cuentas y los recursos que componen una aplicación garantiza una mayor precisión en la investigación.

Requisitos previos

Antes de añadir una AWS cuenta secundaria, asegúrese de tener:

- Acceso a la consola del AWS DevOps agente desde la cuenta principal
- Acceso administrativo a la AWS cuenta secundaria
- Permisos de IAM para crear funciones en la cuenta secundaria

Añadir una cuenta secundaria AWS

Además de los pasos que se indican a continuación, puede utilizarlos the section called “AWS DevOps Guía de incorporación de Agent CLI” para añadir cuentas secundarias mediante programación.

Paso 1: Inicie la configuración de la cuenta secundaria

1. Inicie sesión en la consola AWS de administración y navegue hasta la consola del AWS DevOps agente

2. Seleccione su espacio de agente
3. Ve a la pestaña Capacidades
4. En la sección Nube, localice la subsección de fuentes secundarias
5. Haz clic en Añadir

Paso 2: especifique el nombre del rol

1. En el campo Asigne un nombre a su función, introduzca un nombre para la función que va a crear en la cuenta secundaria
2. Anota este nombre: lo volverás a usar al crear el rol en la cuenta secundaria
3. Copia la política de confianza proporcionada en la consola y guárdala en un espacio temporal

Paso 3: Crea el rol en la cuenta secundaria

1. Abre una nueva pestaña del navegador e inicia sesión en la consola de IAM en la cuenta secundaria AWS
2. Vaya a IAM > Funciones > Crear función
3. Seleccione Política de confianza personalizada
4. Pegue la política de confianza que copió del paso 2
5. Haga clic en Siguiente.

Paso 4: Adjunte la política AWS gestionada

1. En la sección Políticas de permisos, busque AIOpsAssistantPolicy
2. Seleccione la casilla de verificación situada junto a la política AIOpsAssistantPolicygestionada
3. Haga clic en Siguiente.

Paso 5: Asigne un nombre al rol y créelo

1. En el campo Nombre del rol, ingresa el mismo nombre del rol que proporcionaste en el paso 2
2. (Opcional) Agregue una descripción para ayudar a identificar el propósito del rol
3. Revise la política de confianza y los permisos adjuntos

4. Haz clic en Crear rol

Paso 6: Adjunte la política en línea

1. En la consola de IAM, busque y seleccione el rol que acaba de crear
2. Vaya a la pestaña Permisos
3. Haga clic en Añadir permisos > Crear política en línea
4. Cambie a la pestaña JSON
5. Pegue la política que guardó en el paso 2
6. Pegue la política en el editor JSON de la consola de IAM
7. Haga clic en Siguiente.
8. Proporcione un nombre para la política en línea (por ejemplo, "«DevOpsAgentInlinePolicy)
9. Haga clic en Crear política

Paso 7: Complete la configuración

1. Regrese a la consola del AWS DevOps agente en la cuenta principal
2. Haga clic en Siguiente para completar la configuración de la cuenta secundaria
3. Compruebe que el estado de la conexión se muestre como Activo

Comprenda las políticas requeridas

AWS DevOps El agente necesita tres componentes de política para acceder a los recursos de una cuenta secundaria:

- Política de confianza: permite que el AWS DevOps agente de la cuenta principal asuma el rol en la cuenta secundaria. Esto establece la relación de confianza entre las cuentas.
- AIOpsAssistantPolicy (política AWS gestionada): proporciona los permisos básicos de solo lectura que el AWS DevOps agente necesita para investigar los recursos de la cuenta secundaria. Esta política se mantiene AWS y actualiza a medida que se añaden nuevas capacidades.
- Política integrada: proporciona permisos adicionales específicos para la configuración de Agent Space. Esta política se genera en función de la configuración del espacio de agente y puede incluir permisos para integraciones o funciones específicas.

En la cuenta principal, el rol de AWS DevOps agente de IAM debe poder asumir el rol creado en la cuenta secundaria.

Administrar cuentas secundarias

- Visualización de las cuentas conectadas: en la pestaña Capacidades, la subsección Fuentes secundarias muestra todas las cuentas secundarias conectadas con su estado de conexión.
- Actualización del rol de IAM: si necesita modificar los permisos, actualice la política interna asociada al rol en la cuenta secundaria. Los cambios surten efecto inmediatamente.
- Eliminar una cuenta secundaria: para desconectar una cuenta secundaria, selecciónela en la lista de fuentes secundarias y haga clic en Eliminar. Esto no elimina la función de IAM en la cuenta secundaria.

Conexión de fuentes de telemetría

AWS DevOps El agente ofrece tres formas de conectarse a las fuentes de telemetría.

Integración bidireccional integrada

Actualmente, AWS DevOps Agent apoya a los usuarios de Dynatrace con una integración bidireccional integrada que permite lo siguiente:

- Mapeo de recursos topológicos: AWS DevOps Agent ampliará la topología de su espacio de DevOps agentes con entidades y relaciones disponibles a través de un servidor MCP de Dynatrace alojado en un agente. AWS DevOps
- Activación automática de investigaciones: los flujos de trabajo de Dynatrace se pueden configurar para activar la resolución de incidentes a partir de problemas de Dynatrace.
- Introspección telemétrica: el AWS DevOps agente puede realizar una introspección de la telemetría de Dynatrace mientras investiga un problema a través del servidor MCP de Dynatrace alojado en el agente. AWS DevOps
- Actualizaciones de estado: el AWS DevOps agente publicará en la interfaz de usuario de Dynatrace los principales resultados de la investigación, los análisis de las causas fundamentales y los planes de mitigación generados.

Para obtener más información sobre las integraciones bidireccionales, consulte

- the section called “Conectando Dynatrace”

Integración unidireccional integrada

Actualmente, AWS DevOps Agent admite a los usuarios de Datadog AWS CloudWatch, Grafana, New Relic y Splunk con integraciones unidireccionales integradas.

Mejores prácticas de seguridad: al configurar las credenciales para las integraciones unidireccionales integradas, recomendamos limitar las claves y los tokens de la API al acceso de solo lectura. AWS DevOps El agente utiliza estas credenciales únicamente para la introspección telemétrica y no requiere acceso de escritura a su proveedor de telemetría.

La integración AWS CloudWatch unidireccional integrada no requiere ninguna configuración adicional y permite lo siguiente:

- Mapeo de recursos topológicos: AWS DevOps Agent ampliará la topología de su espacio de DevOps agentes con entidades y relaciones disponibles a través de sus cuentas de nube principales y secundarias configuradas. AWS
- Introspección telemétrica: el AWS DevOps agente puede realizar una introspección de la AWS CloudWatch telemetría mientras investiga un problema mediante las funciones de IAM asignadas durante la configuración de la cuenta de nube principal y secundaria. AWS

Las integraciones unidireccionales integradas de Datadog, Grafana, New Relic y Splunk requieren configuración y permiten lo siguiente:

- Activación automática de investigaciones: los eventos de Datadog, Grafana, New Relic y Splunk se pueden configurar para activar las investigaciones de resolución de incidentes de los agentes a través de los webhooks de los AWS DevOps agentes. AWS DevOps
- Introspección telemétrica: el AWS DevOps agente puede realizar una introspección de la telemetría de Datadog, Grafana, New Relic y Splunk mientras investiga un problema a través del servidor MCP remoto de cada proveedor.

Para obtener información sobre las integraciones unidireccionales, consulte lo siguiente:

- the section called “Conectando DataDog”
- the section called “Conectando Grafana”
- the section called “Conectando New Relic”
- the section called “Conexión de Splunk”

Bring-your-own fuentes de telemetría

Para cualquier otra fuente de telemetría, incluidas las métricas de Prometheus, puede aprovechar el soporte de AWS DevOps Agent para la integración de servidores webhook y MCP.

Para obtener más información sobre las integraciones, consulte lo siguiente bring-your-own

- the section called “Invocar al DevOps agente a través de Webhook”
- the section called “Conexión de servidores MCP”

Conectando Dynatrace

Integración bidireccional integrada

Actualmente, AWS DevOps Agent apoya a los usuarios de Dynatrace con una integración bidireccional integrada que permite lo siguiente:

- Mapeo de recursos topológicos: AWS DevOps Agent ampliará la topología del espacio de DevOps agentes con las entidades y relaciones disponibles en su entorno de Dynatrace.
- Activación automática de investigaciones: los flujos de trabajo de Dynatrace se pueden configurar para iniciar investigaciones de resolución de incidentes a partir de problemas de Dynatrace.
- Introspección telemétrica: el AWS DevOps agente puede realizar una introspección de la telemetría de Dynatrace mientras investiga un problema a través del servidor MCP de Dynatrace alojado en el agente. AWS DevOps
- Actualizaciones de estado: el AWS DevOps agente publicará en la interfaz de usuario de Dynatrace los principales resultados de la investigación, los análisis de las causas fundamentales y los planes de mitigación generados.

Incorporación

Proceso de incorporación

La incorporación del sistema de observabilidad de Dynatrace consta de tres etapas:

1. Connect: establezca una conexión con Dynatrace configurando las credenciales de acceso a la cuenta, con todos los entornos que pueda necesitar
2. Habilitar: active Dynatrace en espacios de agentes específicos con entornos de Dynatrace específicos

3. Configure su entorno de Dynatrace: descargue los flujos de trabajo y el panel de control e impórtelos a Dynatrace y anote los detalles del webhook para iniciar las investigaciones en los espacios de agentes designados

Paso 1: Conectar

Establezca una conexión con su entorno de Dynatrace

Configuración

1. Vaya a la página de proveedores de capacidades (a la que se puede acceder desde la barra de navegación lateral)
2. Busque Dynatrace en la sección de proveedores disponibles, en Telemetría, y haga clic en Registrarse
3. Cree un OAuth cliente en Dynatrace, con los permisos detallados.
   a. Consulte la documentación de Dynatrace
   b. Cuando esté listo, presione Siguiente
   c. Puedes conectar varios entornos de Dynatrace y, posteriormente, conectarlos a otros específicos para cada espacio de DevOps agente del que dispongas.
4. Introduce los detalles de Dynatrace desde la configuración del cliente: OAuth
   - Nombre del cliente
   - ID de cliente
   - Secreto de cliente
   - URN de la cuenta
5. Haga clic en Siguiente.
6. Revisa y agrega

Paso 2: Habilitar

Active Dynatrace en un espacio de agente específico y configure el alcance adecuado

Configuración

1. En la página de espacios de agentes, seleccione un espacio de agentes y pulse ver detalles
2. Seleccione la pestaña Capacidades

3. Localice la sección de telemetría y pulse Añadir
4. Verás que Dynatrace tiene el estado «Registrado». Haga clic en añadir para añadirlo a su espacio de agente
5. ID de entorno de Dynatrace: proporcione el ID de entorno de Dynatrace que le gustaría asociar a este espacio de agentes. DevOps
6. Introduzca una o más entidades de Dynatrace. Estas entidades ayudarán a los DevOps agentes a descubrir sus recursos más importantes IDs ; por ejemplo, servicios o aplicaciones. Si no está seguro, puede pulsar quitar.
7. Revise y pulse Guardar
8. Copia la URL del Webhook y el secreto del Webhook. Consulte la documentación de Dynatrace para añadir estas credenciales a Dynatrace.

Paso 3: Configure su entorno de Dynatrace

Para completar la configuración de Dynatrace, deberá realizar ciertos pasos de configuración en su entorno de Dynatrace. Siga las instrucciones de la documentación de Dynatrace.

Esquemas de eventos compatibles

AWS DevOps El agente admite dos tipos de eventos de Dynatrace mediante webhooks. Los esquemas de eventos compatibles se documentan a continuación:

Incidente o evento

Los incidentes se utilizan para iniciar una investigación. El esquema de eventos es:

```
{
    "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 de mitigación

Los eventos de mitigación se utilizan para activar la generación de un informe de mitigación para la investigación sobre los próximos pasos. El esquema del evento es:

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

Eliminación

La fuente de telemetría está conectada en dos niveles: a nivel de espacio de agente y a nivel de cuenta. Para eliminarla por completo, primero debe eliminarla de todos los espacios de agentes en los que se utilice y, a continuación, podrá anular su registro.

Paso 1: Eliminar del espacio de agentes

1. En la página de espacios de agentes, selecciona un espacio de agentes y pulsa ver detalles
2. Seleccione la pestaña Capacidades
3. Desplázate hacia abajo hasta la sección de telemetría
4. Selecciona Dynatrace
5. Presiona quitar

Paso 2: Anular el registro de la cuenta

1. Vaya a la página de proveedores de capacidades (a la que se puede acceder desde la barra de navegación lateral)
2. Desplázate hasta la sección Registrados actualmente.
3. Comprueba que el número de espacios para agentes sea cero (si no, repite el paso 1 anterior en los demás espacios para agentes)
4. Presiona Cancelar registro junto a Dynatrace

Conectando DataDog

Integración unidireccional integrada

Actualmente, AWS DevOps Agent admite a los usuarios de Datadog con una integración unidireccional integrada, lo que permite lo siguiente:

- Activación automática de la investigación: los eventos de Datadog se pueden configurar para activar las investigaciones de resolución de incidentes del AWS DevOps agente mediante los webhooks de los agentes. AWS DevOps
- Introspección telemétrica: el AWS DevOps agente puede realizar una introspección de la telemetría de Datadog mientras investiga un problema a través del servidor MCP remoto de cada proveedor.

Incorporación

Paso 1: Conectar

Establezca la conexión con su terminal MCP remoto de Datadog con las credenciales de acceso a la cuenta

Configuración

1. Vaya a la página de proveedores de capacidades (accesible desde el panel de navegación lateral)
2. Busque Datadog en la sección Proveedores disponibles en Telemetría y haga clic en Registrar
3. Introduzca los detalles de su servidor MCP de Datadog:
   - Nombre del servidor: identificador único (por ejemplo,) my-datadog-server
   - URL del punto final: el punto final de su servidor MCP de Datadog. La URL del punto final varía en función del sitio de Datadog. Consulte la tabla de puntos finales del sitio de Datadog que aparece a continuación.
   - Descripción: descripción del servidor opcional
4. Haga clic en Siguiente.
5. Revisión y envío

Puntos finales del sitio Datadog

La URL del punto final del MCP varía según el sitio de Datadog. Para identificar su sitio, compruebe la URL en su navegador cuando haya iniciado sesión en Datadog o consulte Acceder al sitio de Datadog.

Sitio de Datadog	Dominio del sitio	URL del punto final de MCP
US1 (predeterminado)	`datadoghq.com`	`https://mcp.datadoghq.com/api/unstable/mcp-server/mcp`
US3	`us3.datadoghq.com`	`https://mcp.us3.datadoghq.com/api/unstable/mcp-server/mcp`
US5	`us5.datadoghq.com`	`https://mcp.us5.datadoghq.com/api/unstable/mcp-server/mcp`
EU1	`datadoghq.eu`	`https://mcp.datadoghq.eu/api/unstable/mcp-server/mcp`
AP1	`ap1.datadoghq.com`	`https://mcp.ap1.datadoghq.com/api/unstable/mcp-server/mcp`
AP2	`ap2.datadoghq.com`	`https://mcp.ap2.datadoghq.com/api/unstable/mcp-server/mcp`

Autorización

OAuth Autorización completa por parte de:

- Autorizar como usuario en la página de Datadog OAuth
- Si no ha iniciado sesión, haga clic en Permitir, iniciar sesión y, a continuación, autorizar

Una vez configurado, Datadog estará disponible en todos los espacios de agentes.

Paso 2: Habilitar

Actívelo DataDog en un espacio de agente específico y configure el alcance adecuado

Configuración

1. En la página de espacios de agentes, seleccione un espacio de agentes y pulse ver detalles (si aún no ha creado un espacio de agentes, consultethe section called “Creación de un espacio de agentes”)
2. Seleccione la pestaña Capacidades
3. Desplázate hacia abajo hasta la sección de telemetría
4. Presiona Agregar
5. Seleccione Datadog
6. Next
7. Revise y pulse Guardar
8. Copia la URL y la clave de API de Webhook

Paso 3: Configurar los webhooks

Con la URL y la clave de API de Webhook, puede configurar Datadog para que envíe eventos que inicien una investigación, por ejemplo, a partir de una alarma.

Para garantizar que el DevOps agente pueda utilizar los eventos enviados, asegúrese de que los datos transmitidos al webhook coincidan con el esquema de datos que se especifica a continuación. El DevOps agente puede ignorar los eventos que no coincidan con este esquema.

Establezca el método y los encabezados

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

Envía el cuerpo como una cadena JSON.

```
{
    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;
}
```

Envía webhooks con Datadog https://docs.datadoghq.com/integrations/webhooks/ (ten en cuenta que no seleccionas ninguna autorización y, en su lugar, usa la opción de encabezado personalizado).

Más información: Datadog Remote MCP Server

Eliminación

La fuente de telemetría está conectada en dos niveles: a nivel de espacio de agente y a nivel de cuenta. Para eliminarla por completo, primero debe eliminarla de todos los espacios de agentes en los que se utilice y, a continuación, podrá anular su registro.

Paso 1: Eliminar del espacio de agentes

1. En la página de espacios de agentes, selecciona un espacio de agentes y pulsa ver detalles
2. Seleccione la pestaña Capacidades
3. Desplázate hacia abajo hasta la sección de telemetría
4. Selecciona Datadog
5. Presiona quitar

Paso 2: Anular el registro de la cuenta

1. Vaya a la página de proveedores de capacidades (a la que se puede acceder desde la barra de navegación lateral)
2. Desplázate hasta la sección Registrados actualmente.
3. Comprueba que el número de espacios para agentes sea cero (si no, repite el paso 1 anterior en los demás espacios para agentes)
4. Presiona Cancelar registro junto a Datadog

Conectando Grafana

La integración de Grafana permite al AWS DevOps agente consultar métricas, paneles y datos de alertas de su instancia de Grafana durante la investigación de incidentes. Esta integración sigue un proceso de dos pasos: el registro de Grafana a nivel de cuenta y, a continuación, su conexión a los espacios de agente individuales.

Para mejorar la seguridad, la integración de Grafana solo permite herramientas de solo lectura. Las herramientas de escritura están deshabilitadas y no se pueden activar. Esto significa que el agente puede consultar y leer los datos de tu instancia de Grafana, pero no puede crear, modificar ni eliminar ningún recurso de Grafana, como paneles, alertas o anotaciones. Para obtener más información, consulta Seguridad en el agente. AWS DevOps

Requisitos de Grafana

Antes de conectar Grafana, asegúrese de tener:

- Grafana versión 9.0 o posterior. Es posible que algunas funciones, en particular las operaciones relacionadas con la fuente de datos, no funcionen correctamente con versiones anteriores debido a la falta de puntos finales de la API.
- Una instancia de Grafana accesible a través de HTTPS. Se admiten puntos de conexión de red públicos y privados. Con la conectividad de red privada, su instancia de Grafana se puede alojar en una VPC sin acceso público a Internet. Para obtener más información, consulte the section called “Conexión a herramientas alojadas de forma privada”.
- Una cuenta de servicio de Grafana con un token de acceso que tenga los permisos de lectura adecuados

Registro de Grafana (a nivel de cuenta)

Grafana está registrada a nivel de AWS cuenta y se comparte entre todos los Agent Spaces de esa cuenta.

Paso 1: Configurar Grafana

1. Inicie sesión en la consola de AWS administración
2. Navegue hasta la consola del AWS DevOps agente
3. Vaya a la página de proveedores de capacidades (a la que se puede acceder desde el panel de navegación lateral)
4. Busque Grafana en la sección Proveedores disponibles en Telemetría y haga clic en Registrar
5. En la página Configurar Grafana, introduzca la siguiente información:
   - Nombre del servicio (obligatorio): introduzca un nombre descriptivo para su servidor Grafana utilizando únicamente caracteres alfanuméricos, guiones y guiones bajos. Por ejemplo, `my-grafana-server`.
   - URL de Grafana (obligatorio): introduce la URL HTTPS completa de tu instancia de Grafana. Por ejemplo, `https://myinstance.grafana.net`.
   - Token de acceso a la cuenta de servicio (obligatorio): introduce un token de acceso a la cuenta de servicio de Grafana. Los tokens suelen empezar `glsa_` por. Para crear un token de cuenta de servicio, dirígete a tu instancia de Grafana, ve a Administración > Cuentas de servicio, crea una cuenta de servicio con el rol de Visor y genera un token.
   - Descripción (opcional): agrega una descripción para ayudar a identificar el propósito del servidor. Por ejemplo, `Production Grafana server for monitoring`.
6. (Opcional) Añada AWS etiquetas al registro con fines organizativos.
7. Haga clic en Siguiente.

Paso 2: Revise y envíe el registro de Grafana

1. Revise todos los detalles de configuración de Grafana
2. Haga clic en Enviar para completar el registro
3. Tras el registro exitoso, Grafana aparece en la sección Actualmente registrada de la página de proveedores de capacidades

Añadir Grafana a un espacio de agentes

Tras registrar Grafana a nivel de cuenta, puedes conectarla a espacios de agente individuales:

1. En la consola de AWS DevOps agentes, selecciona tu espacio de agente
2. Ve a la pestaña Capacidades
3. En la sección Telemetría, haga clic en Agregar
4. Seleccione Grafana de la lista de proveedores disponibles
5. Haga clic en Guardar

Configuración de webhooks de alertas de Grafana

Puedes configurar Grafana para que active automáticamente las investigaciones de los AWS DevOps agentes cuando se activen alertas mediante el envío de webhooks a través de los puntos de contacto de Grafana. Para obtener más información sobre los métodos de autenticación de webhooks y la administración de credenciales, consulte. the section called “Invocar al DevOps agente a través de Webhook”

Paso 1: Cree una plantilla de notificación personalizada

En tu instancia de Grafana, ve a Alertas > Puntos de contacto > Plantillas de notificaciones y crea una nueva plantilla con el siguiente contenido:

```
{{ 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 }}
```

Esta plantilla formatea las alertas de Grafana en la estructura de carga útil de webhook que espera el agente. AWS DevOps Asigna las etiquetas, las anotaciones y el estado de las alertas a los campos correspondientes e incluye todas las etiquetas de alerta como metadatos.

> Nota: Esta plantilla procesa solo la primera alerta de un grupo. Grafana agrupa varias alertas de disparo en una sola notificación de forma predeterminada. Para asegurarte de que cada alerta se envíe de forma individual, configura tus políticas de notificación para agruparlas por`alertname`. Además, esta plantilla no escapa a los caracteres JSON especiales en los valores o anotaciones de las etiquetas. Asegúrese de que las etiquetas de alerta y la `summary` anotación no contengan caracteres como comillas dobles o líneas nuevas, ya que generarían un JSON no válido.

Paso 2: Crea un punto de contacto de webhook

1. En Grafana, vaya a Alertas > Puntos de contacto y haga clic en Añadir punto de contacto
2. Selecciona Webhook como tipo de integración
3. Establezca la URL del punto final del webhook de su AWS DevOps agente
4. En la sección Configuración opcional del webhook, configura los encabezados de autenticación en función del tipo de webhook. Consulta los métodos de autenticación de Webhook para obtener más información.
5. Configura el campo Mensaje para usar tu plantilla personalizada: `{{ template "devops-agent-payload" . }}`
6. Haz clic en Guardar punto de contacto

Paso 3: Asigne el punto de contacto a una política de notificaciones

1. Vaya a Alertas > Políticas de notificación
2. Edite una política existente o cree una nueva
3. Configura el punto de contacto en el punto de contacto del webhook que creaste
4. Haz clic en Guardar política

Cuando se active una alerta coincidente, Grafana enviará la carga útil formateada al AWS DevOps Agente, que iniciará una investigación automáticamente.

Limitaciones

- ClickHouse herramientas de fuentes de ClickHouse datos: actualmente no se admiten las herramientas de fuentes de datos.
- Prevención proactiva de incidentes: actualmente the section called “Prevención proactiva de incidentes” no utiliza las herramientas de Grafana. Support está previsto para una versión futura.

Consideraciones sobre Amazon Managed Grafana

Si utilizas Amazon Managed Grafana (AMG), ten en cuenta las siguientes limitaciones:

- No se admiten los puntos de contacto de Webhook. Actualmente, AMG no admite los puntos de contacto de Webhook en su configuración de alertas. No puedes usar AMG para enviar webhooks de alertas directamente al agente. AWS DevOps Para obtener más información, consulta Cómo alertar a los puntos de contacto en Amazon Managed Grafana.
- Caducidad de los tokens de las cuentas de servicio de AMG: los tokens de las cuentas de servicio de AMG tienen una caducidad máxima de 30 días. Deberás rotar las fichas y actualizar tu registro de Grafana en AWS DevOps Agent antes de que caduquen. Consulte Administrar las conexiones de Grafana para saber cómo actualizar las credenciales. Para obtener más información sobre los límites de los tokens AMG, consulte Cuentas de servicio en Amazon Managed Grafana.

Gestión de las conexiones de Grafana

- Actualización de credenciales: si el token de su cuenta de servicio caduca o necesita actualizarse, anule el registro de Grafana en la página de proveedores de capacidades y vuelva a registrarse con el nuevo token.
- Visualización de las instancias conectadas: en la consola del AWS DevOps agente, selecciona tu espacio de agente y ve a la pestaña Capacidades para ver las fuentes de telemetría conectadas.
- Eliminar Grafana: para desconectar Grafana de un espacio de agentes, selecciónelo en la sección Telemetría y haga clic en Eliminar. Para eliminar por completo el registro, elimínelo primero de todos los Agent Spaces y, a continuación, anule el registro en la página de proveedores de capacidades.

Conectando New Relic

Integración unidireccional integrada

Actualmente, AWS DevOps Agent apoya a los usuarios de New Relic con una integración unidireccional integrada, que permite lo siguiente:

- Activación automática de las investigaciones: los eventos de New Relic se pueden configurar para activar las investigaciones de resolución de incidentes por parte de los AWS DevOps agentes mediante AWS DevOps los webhooks de los agentes.
- Introspección telemétrica: el AWS DevOps agente puede realizar una introspección de la telemetría de New Relic mientras investiga un problema a través del servidor MCP remoto de cada proveedor.

Incorporación

Paso 1: Conectar

Establezca la conexión con su terminal MCP remoto de New Relic con las credenciales de acceso a la cuenta

Utilice un usuario de plataforma completa (no básico o básico) en New Relic para habilitar las herramientas MCP de New Relic.

Configuración

1. Vaya a la página de proveedores de capacidades (a la que se puede acceder desde la barra de navegación lateral)
2. Busque New Relic en la sección de proveedores disponibles, en Telemetría, y haga clic en Registrarse
3. Sigue las instrucciones para obtener tu clave de API de New Relic
4. Introduce los detalles de la clave de API de tu servidor MCP de New Relic:
   - ID de cuenta: introduce tu ID de cuenta de New Relic obtenido anteriormente
   - Clave de API: Introduzca la clave de API obtenida anteriormente
   - Selecciona la región de EE. UU. o la UE según la ubicación de tu cuenta de New Relic.
5. Haz clic en Añadir

Paso 2: Habilitar

Active New Relic en un espacio de agente específico y configure el alcance adecuado

Configuración

1. En la página de espacios de agentes, seleccione un espacio de agentes y pulse ver los detalles (si aún no ha creado un espacio de agentes, consulte) the section called “Creación de un espacio de agentes”
2. Seleccione la pestaña Capacidades
3. Desplázate hacia abajo hasta la sección de telemetría
4. Presiona Agregar
5. Selecciona New Relic
6. Next
7. Revisa y presiona Guardar
8. Copia la URL y la clave de API de Webhook

Paso 3: Configurar los webhooks

Con la URL y la clave de API de Webhook, puedes configurar New Relic para que envíe eventos que activen una investigación, por ejemplo, a partir de una alarma. Para obtener más información sobre la configuración de los webhooks, consulta el artículo sobre el seguimiento de cambios en los webhooks.

Para garantizar que el DevOps agente pueda utilizar los eventos enviados, asegúrese de que los datos transmitidos al webhook coincidan con el esquema de datos que se especifica a continuación. El DevOps agente puede ignorar los eventos que no coincidan con este esquema.

Establezca el método y los encabezados

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

Envía el cuerpo como una cadena JSON.

```
{
    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;
}
```

Envía webhooks con las notificaciones de webhooks de New Relichttps://newrelic.com/instant-observability/. Puedes seleccionar el token de portador como tipo de autorización o no seleccionar ninguna autorización y, en su lugar, añadirlo como un encabezado personalizado. `Authorization: Bearer <Token>`

Más información: https://docs.newrelic.com/docs/agentic-/ai/mcp/overview

Eliminación

La fuente de telemetría está conectada en dos niveles: a nivel de espacio de agente y a nivel de cuenta. Para eliminarla por completo, primero debe eliminarla de todos los espacios de agentes en los que se utilice y, a continuación, podrá anular su registro.

Paso 1: Eliminar del espacio de agentes

1. En la página de espacios de agentes, selecciona un espacio de agentes y pulsa ver detalles
2. Seleccione la pestaña Capacidades
3. Desplázate hacia abajo hasta la sección de telemetría
4. Selecciona New Relic
5. Presiona quitar

Paso 2: Anular el registro de la cuenta

1. Vaya a la página de proveedores de capacidades (a la que se puede acceder desde la barra de navegación lateral)

2. Desplázate hasta la sección Registrados actualmente.
3. Comprueba que el número de espacios para agentes sea cero (si no, repite el paso 1 anterior en los demás espacios para agentes)
4. Pulsa Cancelar registro junto a New Relic

Conexión de Splunk

Integración unidireccional integrada

Actualmente, AWS DevOps Agent apoya a los usuarios de Splunk con una integración unidireccional integrada, que permite lo siguiente:

- Activación automática de las investigaciones: los eventos de Splunk se pueden configurar para activar las investigaciones de resolución de incidentes por parte del AWS DevOps agente mediante AWS DevOps los webhooks de los agentes.
- Introspección telemétrica: el AWS DevOps agente puede realizar una introspección de la telemetría de Splunk mientras investiga un problema a través del servidor MCP remoto de cada proveedor.

Requisitos previos

Obtener un token de API de Splunk

Necesitará una URL y un token del MCP para conectarse a Splunk.

Pasos para el administrador de Splunk

Su administrador de Splunk debe realizar los siguientes pasos:

- habilitar el acceso a la API REST
- habilite la autenticación mediante token en la implementación.
- cree un nuevo rol 'mcp_user', el nuevo rol no necesita tener ninguna capacidad.
- asigne el rol «mcp_user» a todos los usuarios de la implementación que estén autorizados a usar el servidor MCP.
- cree el token para los usuarios autorizados con el nombre de «mcp» y establezca la caducidad adecuada si el usuario no tiene permiso para crear los tokens por sí mismo.

Pasos para usar Splunk

Un usuario de Splunk debe realizar los siguientes pasos:

- Obtenga un token apropiado del administrador de Splunk o cree uno por sí mismo, si tiene el permiso. La audiencia del token debe ser «mcp».

Incorporación

Paso 1: Conectar

Establezca la conexión con su terminal MCP remoto de Splunk con las credenciales de acceso a la cuenta

Configuración

1. Vaya a la página de proveedores de capacidades (a la que se puede acceder desde la barra de navegación lateral)
2. Busque Splunk en la sección Proveedores disponibles, en Telemetría, y haga clic en Registrar
3. Introduzca los detalles de su servidor MCP de Splunk:
   - Nombre del servidor: identificador único (por ejemplo,) my-splunk-server
   - URL del punto final: el punto final de su servidor MCP de Splunk:

```
https://<YOUR_SPLUNK_DEPLOYMENT_NAME>.api.scs.splunk.com/
<YOUR_SPLUNK_DEPLOYMENT_NAME>/mcp/v1/
```

- Descripción: descripción del servidor opcional
- Nombre del token: el nombre del token portador para la autenticación: `my-splunk-token`
- Valor del token: el valor del token del portador para la autenticación

Paso 2: Habilitar

Active Splunk en un espacio de agente específico y configure el alcance adecuado

Configuración

1. En la página de espacios de agentes, seleccione un espacio de agentes y pulse ver los detalles (si aún no ha creado un espacio de agentes, consulte) the section called “Creación de un espacio de agentes”
2. Seleccione la pestaña Capacidades
3. Desplázate hacia abajo hasta la sección de telemetría
4. Presiona Agregar
5. Selecciona Splunk
6. Next
7. Revise y pulse Guardar
8. Copia la URL y la clave de API de Webhook

Paso 3: Configurar los webhooks

Con la URL y la clave de API de Webhook, puede configurar Splunk para que envíe eventos que inicien una investigación, por ejemplo, a partir de una alarma.

Para garantizar que el DevOps agente pueda utilizar los eventos enviados, asegúrese de que los datos transmitidos al webhook coincidan con el esquema de datos que se especifica a continuación. El DevOps agente puede ignorar los eventos que no coincidan con este esquema.

Establezca el método y los encabezados

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

Envía el cuerpo como una cadena JSON.

```
{
    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;
}
```

Envía webhooks con Splunk https://help.splunk.com/en/splunk- enterprise/alert-and-respond/alerting-manual/9.4/configure-alert-actions/use - a-webhook-alert-action (ten en cuenta que no seleccionas autorización y, en su lugar, usa la opción de encabezado personalizado)

Más información:

- Documentación del servidor MCP de Splunk:/-platform/ -splunk-platform https://help.splunk.com/en/splunk-cloud-platform mcp-server-for-splunk about-mcp-server-for
- Requisitos y limitaciones de acceso a la API REST de la plataforma Splunk Cloud: https://docs.splunk.com/Documentation/SplunkCloud/latest/RESTTUT/RESTandCloud
- Gestione los tokens de autenticación en Splunk Cloud Platform:/- 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
- Cree y gestione funciones con Splunk Web: https://docs.splunk.com/Documentation/SplunkCloud/latest/Security/Addandeditroles

Eliminación

La fuente de telemetría está conectada en dos niveles: a nivel de espacio de agente y a nivel de cuenta. Para eliminarla por completo, primero debe eliminarla de todos los espacios de agentes en los que se utilice y, a continuación, podrá anular su registro.

Paso 1: Eliminar del espacio de agentes

1. En la página de espacios de agentes, selecciona un espacio de agentes y pulsa ver detalles
2. Seleccione la pestaña Capacidades
3. Desplázate hacia abajo hasta la sección de telemetría
4. Selecciona Splunk
5. Presiona eliminar

Paso 2: Anular el registro de la cuenta

1. Vaya a la página de proveedores de capacidades (a la que se puede acceder desde la barra de navegación lateral)
2. Desplázate hasta la sección Registrados actualmente.
3. Comprueba que el número de espacios para agentes sea cero (si no, repite el paso 1 anterior en los demás espacios para agentes)
4. Presiona Cancelar registro junto a Splunk

Conexión a la venta de entradas y al chat

AWS DevOps El agente está diseñado para actuar como miembro de tu equipo al participar en los canales de comunicación existentes de tu equipo. Puedes conectar DevOps Agent a tus sistemas de emisión de tickets y alarmas y, por ejemplo ServiceNow , iniciar automáticamente investigaciones a partir de los tickets de incidentes, acelerando la respuesta a los incidentes dentro de tus flujos de trabajo actuales para reducir el tiempo de recuperación previsto (MTTR). PagerDuty También puedes conectar a tu DevOps agente a los sistemas de colaboración de tu equipo, como Slack, para recibir resúmenes de las actividades de tu DevOps agente en un canal de chat.

Para obtener información sobre cómo conectar las integraciones de venta de entradas y chat, consulta lo siguiente:

- the section called “Conectando PagerDuty”
- the section called “Conectando ServiceNow”
- the section called “Conectar Slack”

Conectando PagerDuty

PagerDuty La integración permite al AWS DevOps agente acceder a los datos de los incidentes, los horarios de guardia y la información de servicio de su PagerDuty cuenta y actualizarlos durante la investigación de los incidentes y la respuesta automática. Esta integración utiliza la OAuth versión 2.0 para una autenticación segura.

> **Important**
>
> AWS DevOps El agente solo es compatible con la versión PagerDuty OAuth 2.0 más reciente (con alcance OAuth). No se admite la versión antigua PagerDuty OAuth con URI de redireccionamiento.

PagerDuty requisitos

Antes de realizar PagerDuty la conexión, asegúrese de tener:

- Una PagerDuty cuenta con su ID de OAuth cliente y su secreto
- El subdominio de tu PagerDuty cuenta (por ejemplo, si tu PagerDuty URL es`https://your-company.pagerduty.com`, el subdominio es) `your-company`

Registrarse PagerDuty

PagerDuty se registra a nivel de AWS cuenta y se comparte entre todos los espacios de agentes de esa cuenta.

Paso 1: Configurar el acceso en PagerDuty

1. Inicie sesión en la consola AWS de administración
2. Navegue hasta la consola del AWS DevOps agente
3. Vaya a la página de proveedores de capacidades (a la que se puede acceder desde el panel de navegación lateral)
4. Busque PagerDutyen la sección Proveedores disponibles en Comunicación y haga clic en Registrarse
5. Siga la configuración guiada de la PagerDuty página Configurar el acceso en:

Compruebe la región y el subdominio de su servicio:

- Ámbito de la cuenta: selecciona tu PagerDuty región (EE. UU. o UE) e introduce tu PagerDuty subdominio. Por ejemplo, si tu PagerDuty URL es`https://your-company.pagerduty.com`, `your-company` introdúcela.

Crea una nueva aplicación en PagerDuty:

- En otra pestaña del navegador, inicia sesión PagerDuty y ve a Integraciones > Registro de aplicaciones
- Crea una nueva aplicación con OAuth 2.0 Scope OAuth
- En Permisos, concede los siguientes ámbitos mínimos obligatorios:`incidents.read`, y `incidents.write` `services.read`
- Habilite la integración de eventos para permitir la comunicación bidireccional entre AWS DevOps el agente y PagerDuty

Configure OAuth las credenciales:

- Alcance del permiso: los ámbitos mínimos requeridos son:`incidents.read`,, `incidents.write` `services.read`
- Nombre del cliente: introduzca un nombre descriptivo para su OAuth cliente
- ID de cliente: introduzca el ID de OAuth cliente que aparece en el registro de PagerDuty la aplicación
- Secreto de cliente: introduce el secreto de OAuth cliente que aparece en el registro de PagerDuty la aplicación

Paso 2: Revisa y envía PagerDuty el registro

1. Revise todos los detalles PagerDuty de configuración
2. Haga clic en Enviar para completar el registro
3. Si el registro se ha realizado correctamente, PagerDuty aparece en la sección actualmente registrados de la página de proveedores de capacidades

Añadir PagerDuty a un espacio de agente

Tras registrarse PagerDuty a nivel de cuenta, puede conectarla a espacios de agente individuales:

1. En la consola de AWS DevOps agentes, selecciona tu espacio de agente
2. Ve a la pestaña Capacidades
3. En la sección Comunicaciones, haga clic en Agregar
4. Seleccione PagerDutyde la lista de proveedores disponibles
5. Haga clic en Guardar

Administrar PagerDuty las conexiones

- Actualización de credenciales: si es necesario actualizar sus OAuth credenciales, cancele el registro en la página PagerDuty de proveedores de capacidades y vuelva a registrarse con las nuevas credenciales.
- Visualización de las conexiones: en la consola del AWS DevOps agente, seleccione su espacio de agente y vaya a la pestaña Capacidades para ver las integraciones de comunicación conectadas.
- Eliminar PagerDuty: para desconectarse PagerDuty de un espacio de agentes, selecciónelo en la sección de comunicaciones y haga clic en Eliminar. Para eliminar por completo el registro, elimínelo primero de todos los espacios de agentes y, a continuación, anule el registro en la página de proveedores de capacidades.

Soporte para Webhook

AWS DevOps El agente solo admite webhooks PagerDuty V3. No se admiten las versiones anteriores de webhook.

Para obtener más información sobre las suscripciones a los webhooks de la versión PagerDuty 3, consulta la descripción general de los webhooks en la documentación para desarrolladores. PagerDuty

Conectando ServiceNow

En este tutorial, se explica cómo conectar una ServiceNow instancia con el AWS DevOps agente para que pueda iniciar automáticamente investigaciones de respuesta a incidentes cuando se crea un ticket y publicar sus principales hallazgos en el ticket de origen. También contiene ejemplos sobre cómo configurar la ServiceNow instancia para enviar solo tickets específicos a un espacio de DevOps agente y cómo organizar el enrutamiento de los tickets entre varios espacios de DevOps agente.

Configuración inicial

El primer paso es crear ServiceNow un cliente de OAuth aplicación que AWS DevOps puedas usar para acceder a tu ServiceNow instancia.

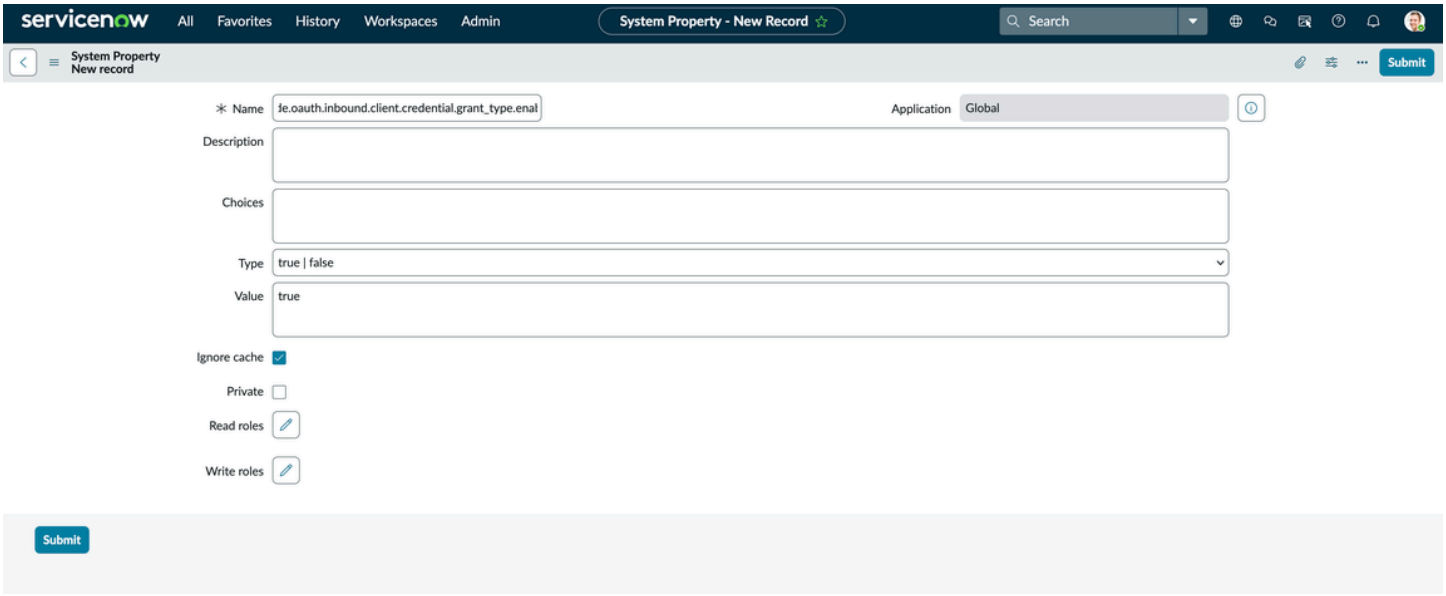
Cree un cliente ServiceNow OAuth de aplicación

1. Habilite la propiedad del sistema de credenciales de cliente de su instancia

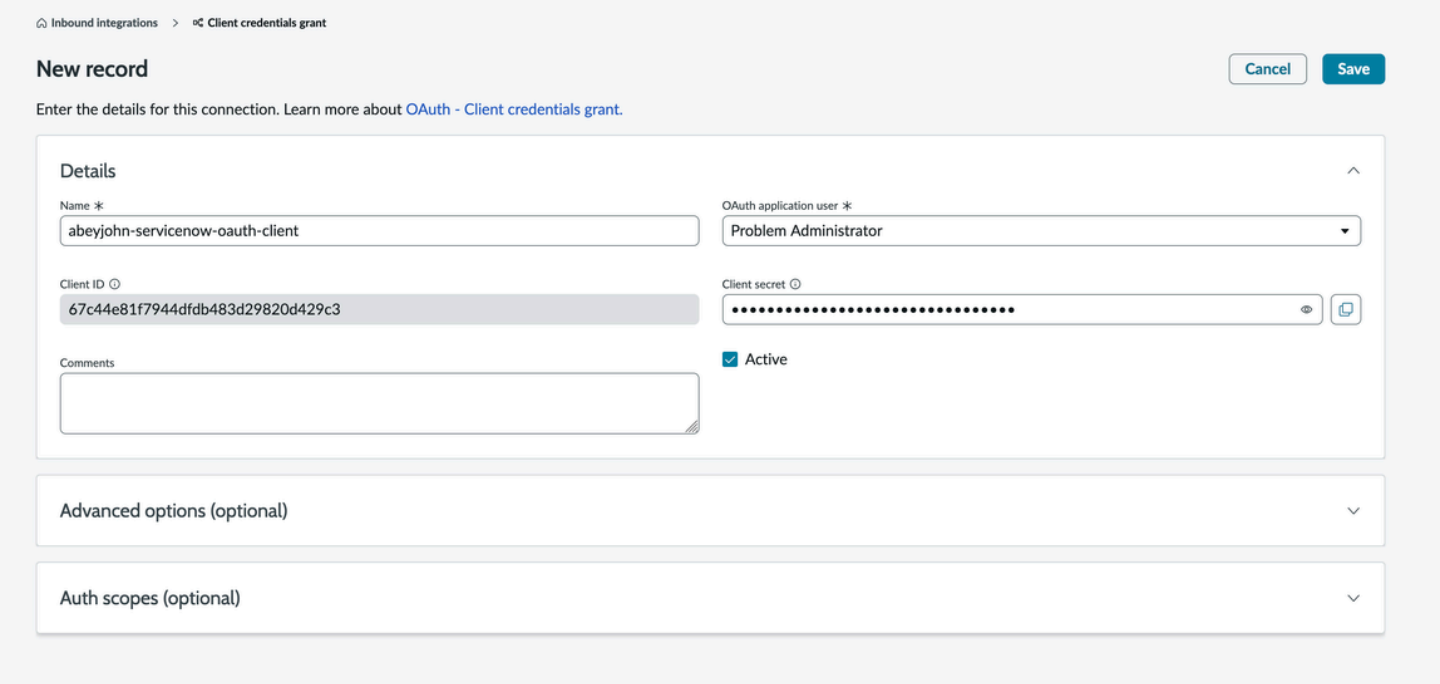
a. Busca `sys_properties.list` en el cuadro de búsqueda del filtro y, a continuación, pulsa enter (no mostrará la opción, pero pulsar enter funciona)

b. Elige Nuevo

c. Añada el nombre como `glide.oauth.inbound.client.credential.grant_type.enabled` y el valor a true y escriba true | false

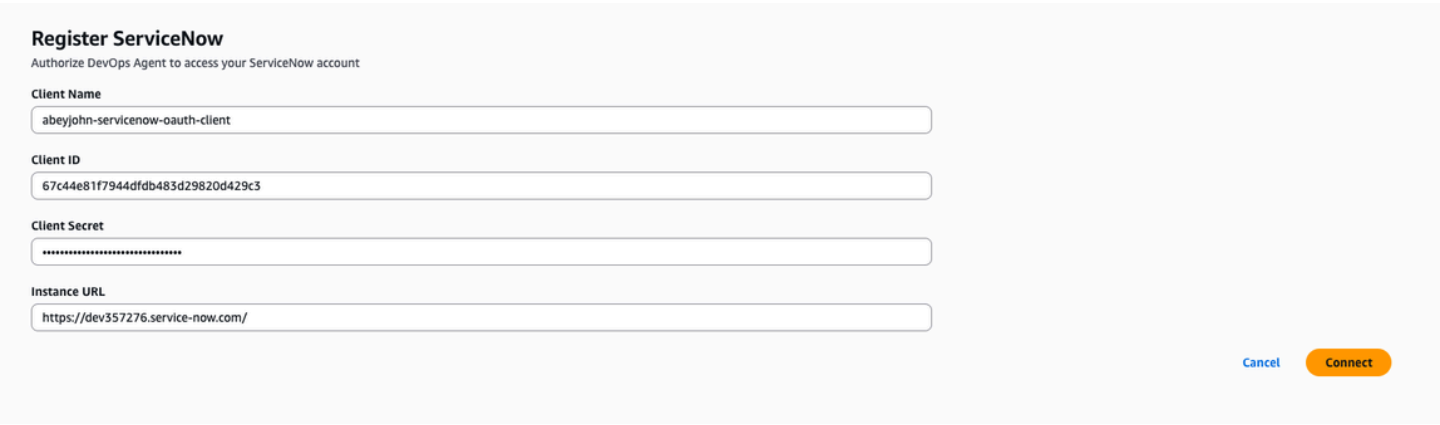


1. Vaya a Sistema OAuth > Registro de aplicaciones desde el cuadro de búsqueda del filtro
2. Seleccione «Nueva» > «Nueva experiencia de integración entrante» > «Nueva integración» > «OAuth - Concesión de credenciales de cliente»
3. Elija un nombre y establezca el usuario de la OAuth aplicación como «Administrador de problemas» y haga clic en «Guardar»



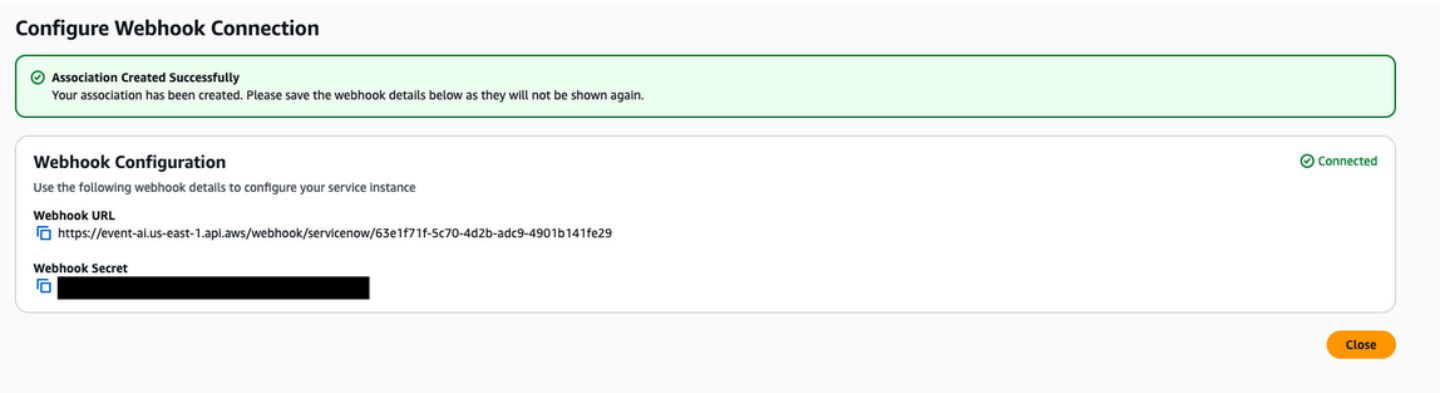
Conecta a tu ServiceNow OAuth cliente con el AWS DevOps agente

1. Puede iniciar este proceso en dos lugares. En primer lugar, vaya a la página de proveedores de capacidades y busque en Comunicación y, ServiceNowa continuación, haga clic en Registrar. También puede seleccionar cualquier espacio de DevOps agente que haya creado y navegar hasta Capacidades → Comunicaciones → Añadir → ServiceNow y hacer clic en Registrar.
2. A continuación, autorice al DevOps agente a acceder a su ServiceNow instancia mediante el cliente de OAuth aplicación que acaba de crear.



- Siga los pasos siguientes y guarde la información resultante sobre el webhook

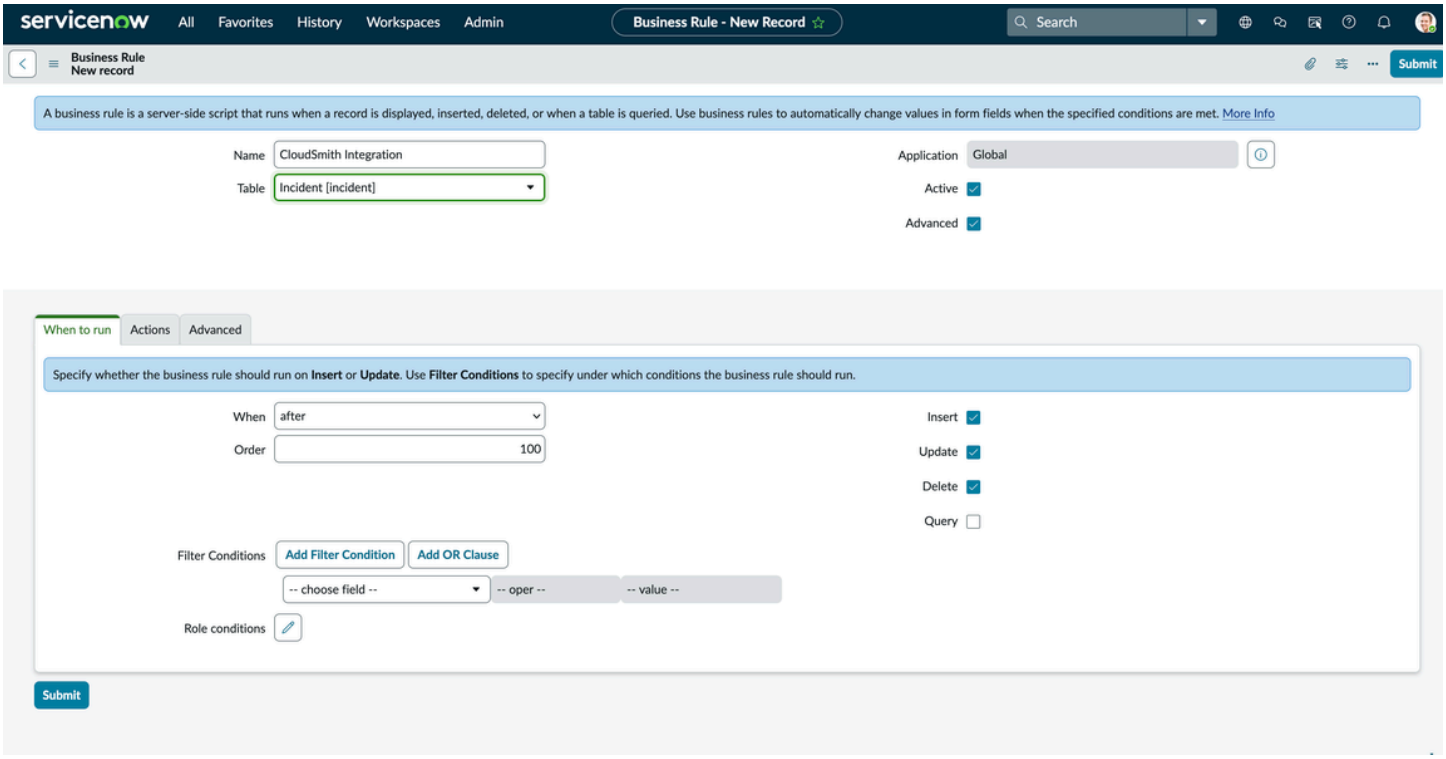
> **Important**
>
> No volverás a ver esta información



Configure su regla ServiceNow de negocio

Una vez que haya establecido la conectividad, tendrá que configurar una regla empresarial ServiceNow para enviar los tickets a sus espacios de DevOps agente.

1. Vaya a Suscripciones de actividades → Administración → Reglas empresariales y haga clic en Nuevo.
2. Defina el campo «Tabla» como «Incidente [incidente]», marque la casilla «Avanzado» y configure la regla para que se ejecute después de Insertar, Actualizar y Eliminar.



1. Ve a la pestaña «Avanzado» y añade el siguiente script de webhook, inserta el secreto y la URL del webhook donde se indica y pulsa Enviar.

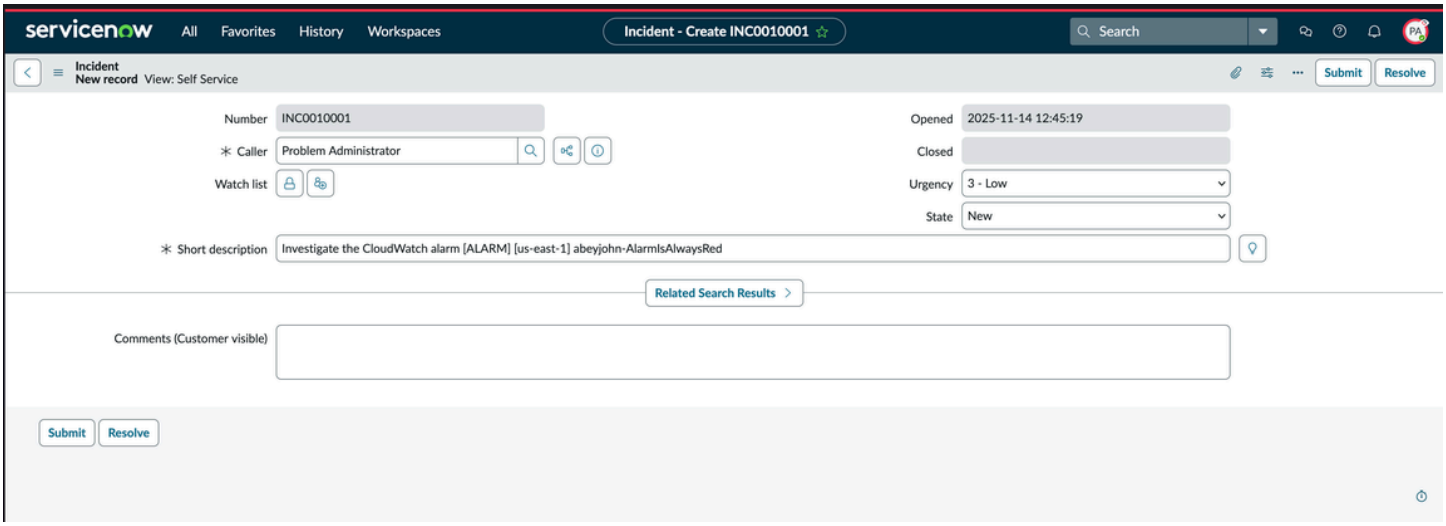
```
(function executeRule(current, previous /*null when async*/ ) {

    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);
```

Si ha decidido registrar su ServiceNow conexión desde la página de proveedores de capacidades, ahora tiene que ir al espacio de DevOps agentes en el que desea investigar las denuncias de ServiceNow incidentes, seleccionar Capacidades → Comunicaciones y, a continuación, registrar la ServiceNow instancia que registró en la página de proveedores de capacidades. Ahora, todo está listo y todos los incidentes en los que la persona que llama tenga el nombre de «administrador de problemas» (para imitar los permisos que usted otorgó al AWS DevOps OAuth cliente) darán lugar a una investigación de respuesta al incidente en el espacio de DevOps agente configurado. Para comprobarlo, crea un nuevo incidente ServiceNow y configura el campo de llamada del incidente como «Administrador de problemas».



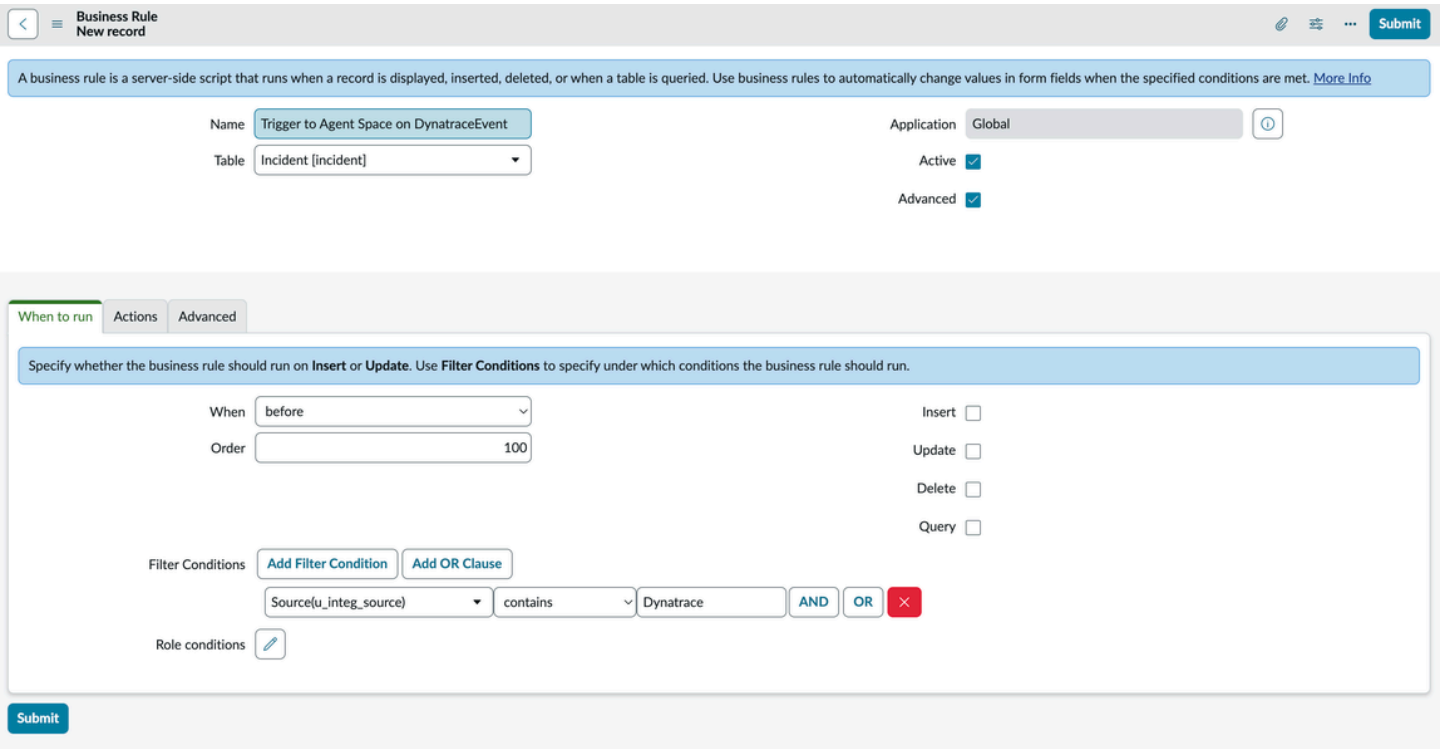
ServiceNow actualizaciones de tickets

Durante todas las investigaciones de respuesta a incidentes que se inicien, su DevOps agente actualizará sus principales hallazgos, sus análisis de las causas fundamentales y los planes de mitigación en la solicitud de origen. Las conclusiones del agente se publican junto con los comentarios de un incidente y, por el momento, solo publicaremos los registros de los agentes relacionados con las actualizaciones del tipo `finding` `causeinvestigation_summary`,`mitigation_summary`, y del estado de la investigación (p. ej.`AWS DevOps Agent started/finished its investigation`).

Ejemplos de enrutamiento y organización de tickets

Escenario: filtrar qué incidentes se envían a un espacio de DevOps agentes

Se trata de un escenario sencillo, pero requiere cierta configuración ServiceNow para crear un campo ServiceNow que permita rastrear el origen del incidente. Para este ejemplo, cree un nuevo campo Source (u_source) con el generador de formularios SNOW. Esto permitirá rastrear la fuente del incidente y usarla para enrutar las solicitudes de una fuente en particular a un espacio de DevOps agente. El enrutamiento se logra creando una regla de negocio de Service Now y, en la pestaña Cuándo ejecutar, configurando «Cuándo» se activan y «Condiciones de filtrado». En este ejemplo, las condiciones del filtro se establecen de la siguiente manera:

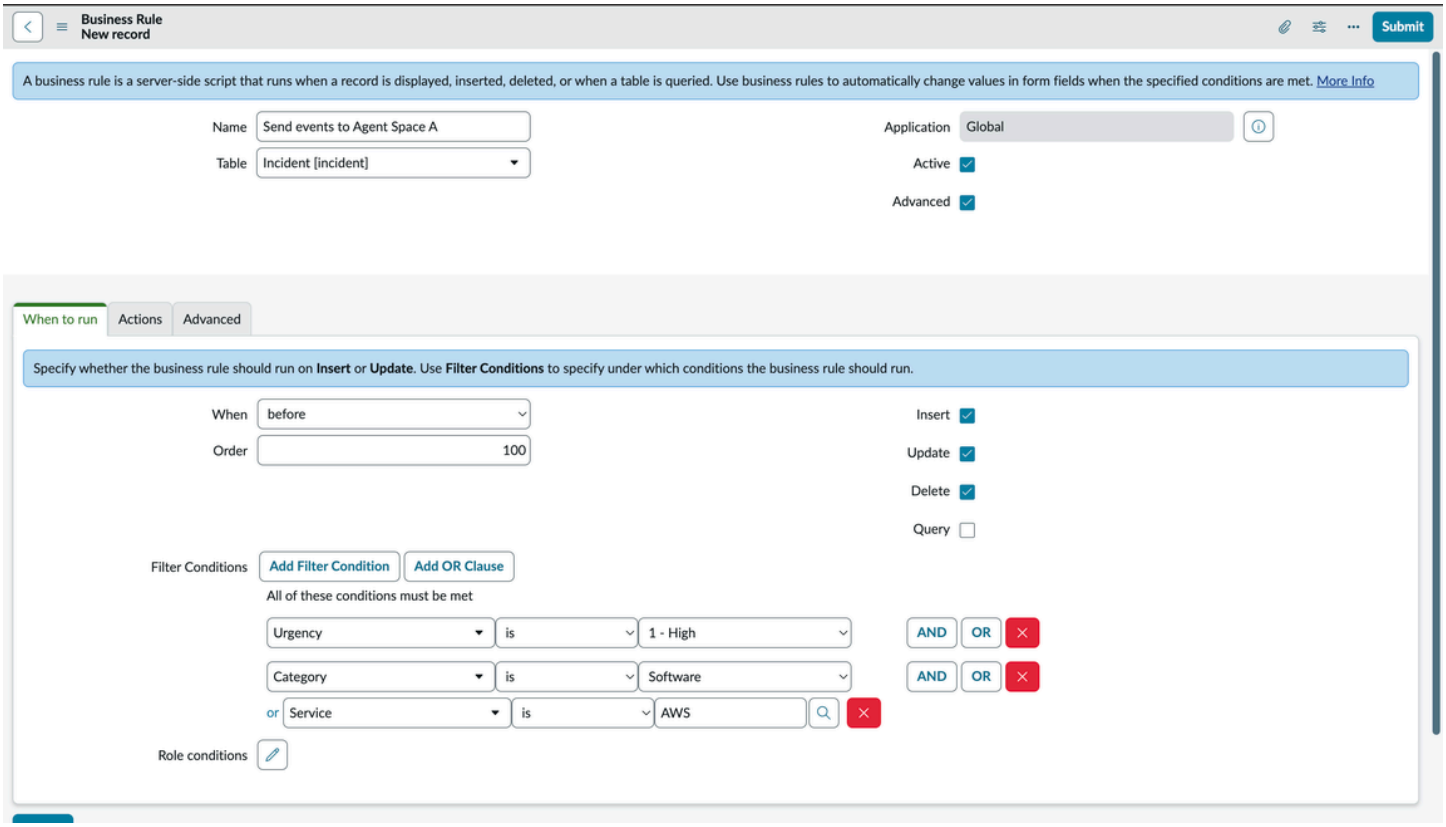


Escenario: enrutar los incidentes a través de varios espacios de DevOps agentes

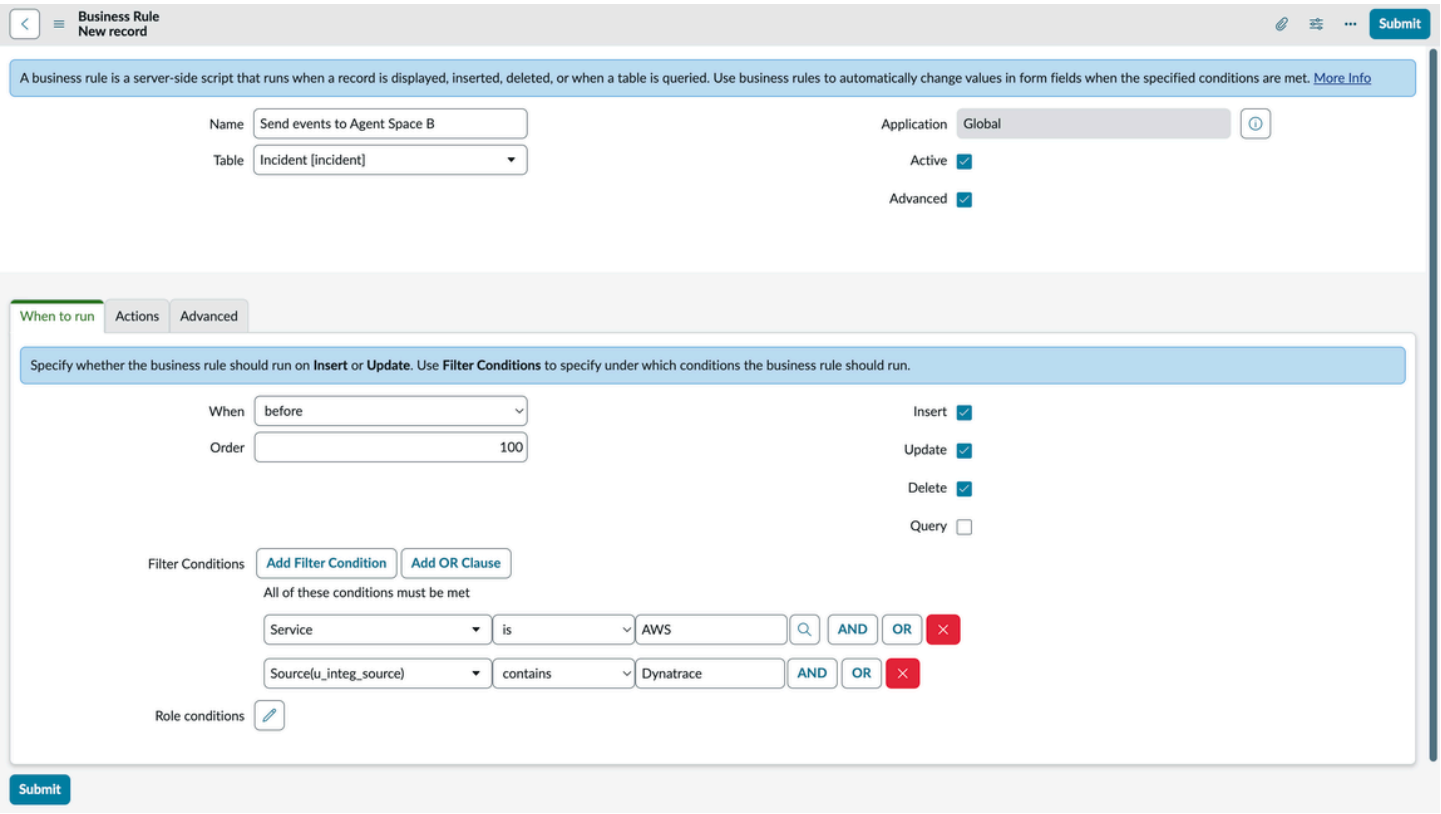
En este ejemplo, se muestra cómo iniciar una investigación en el espacio de DevOps agentes B cuando la urgencia es1, la categoría o el servicioAWS, y cómo iniciar una investigación en el espacio de DevOps agentes A cuando el servicio lo es AWS y la fuente esDynatrace. Software

Este escenario se puede llevar a cabo de dos maneras. El propio script del webhook se puede actualizar para incluir esta lógica empresarial. En este escenario, mostraremos cómo lograrlo con una regla de ServiceNow negocio, para aumentar la transparencia y simplificar la depuración. El enrutamiento se logra mediante la creación de dos reglas comerciales de Service Now.

- Cree una regla de negocio ServiceNow para el espacio de DevOps agentes A y cree una condición utilizando el generador de condiciones para enviar únicamente los eventos en función de nuestra condición especificada.



- A continuación, cree otra regla de negocio ServiceNow para AgentSpace B, cuya regla de negocio solo se active cuando el servicio sea AWS y la fuente sea Dynatrace.



Ahora, cuando cree un nuevo incidente que cumpla con la condición especificada, se iniciará una investigación en el Agent Space A o DevOps en el DevOps Agent Space B, lo que le proporcionará un control pormenorizado sobre la distribución de los incidentes.

Conectar Slack

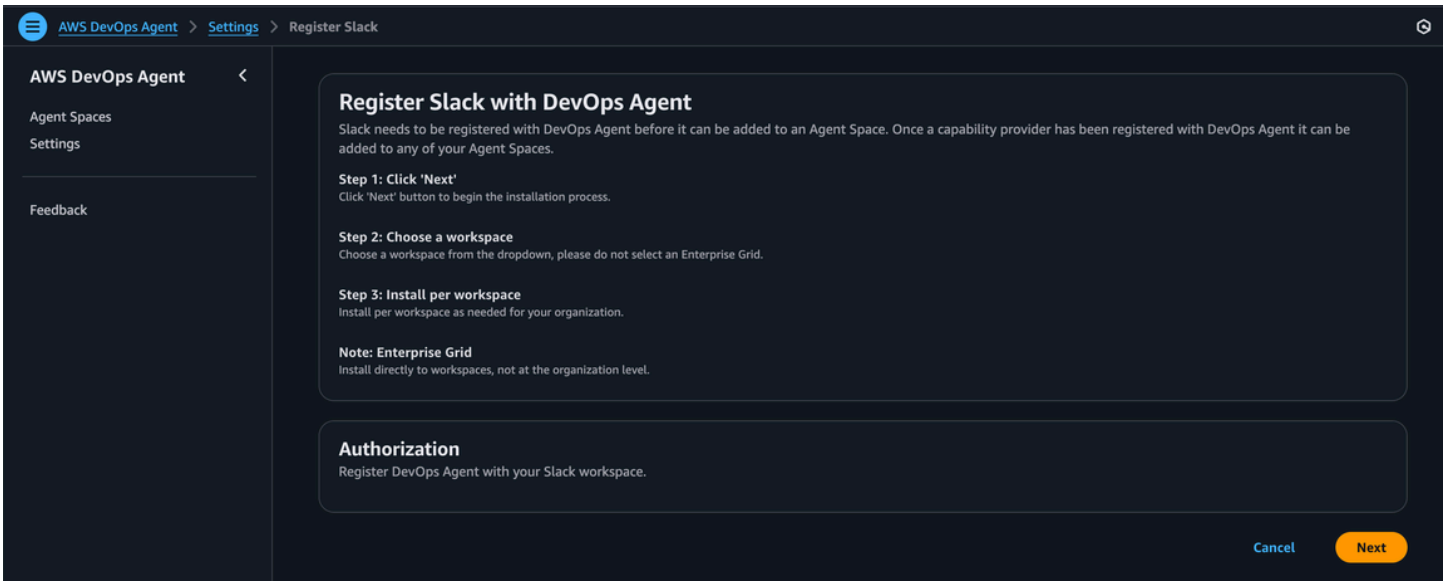
Puedes configurar AWS DevOps Agent para que actualice el canal de Slack que selecciones con las principales conclusiones de la investigación sobre la respuesta a incidentes, los análisis de las causas principales y los planes de mitigación generados.

Antes de empezar

Slack debe estar registrado en DevOps Agent para poder añadirlo a un espacio de agentes. Para integrar AWS DevOps Agent con Slack, debes cumplir los siguientes requisitos:

- Ten acceso a un espacio de trabajo de Slack con la posibilidad de instalar y autorizar aplicaciones de terceros
- Has identificado los canales de Slack a los que quieres que el AWS DevOps agente envíe las notificaciones

Registra la integración de Slack con el agente AWS DevOps



1. En la página Proveedores de capacidades de la consola del AWS DevOps agente, busca Slack en la sección Proveedores disponibles, en la sección Comunicación, y haz clic en Registrar.
2. Pulsa el botón de registro.
3. Se te redirigirá a Slack para que autorices la solicitud de AWS DevOps agente en tu espacio de trabajo.
4. En la página de autorización de Slack, instálala directamente en los espacios de trabajo, no a nivel de la organización.
5. Elige un espacio de trabajo en el menú desplegable. No selecciones un Enterprise Grid.
6. Instálalo por espacio de trabajo según sea necesario para tu organización.
7. Revisa los ámbitos solicitados y haz clic en Permitir para autorizar la integración.
8. Tras la autorización, volverá a la consola del AWS DevOps agente.

Asocia Slack a tus espacios de DevOps agente

Tras registrar Slack en tu espacio de DevOps agente, puedes asociarlo a tus espacios de DevOps agente:

1. En la pestaña Capacidades de tu configuración AgentSpace, ve a Comunicaciones > Slack.
2. Selecciona Añadir Slack
3. Ingresa el ID del canal

4. Selecciona Crear para completar la configuración de Slack.

> **Note**
>
> El usuario bot del agente debe añadirse a los canales privados para que pueda publicar mensajes.

> **Important**
>
> Si desinstalas la aplicación de Slack, es posible que no se pueda volver a instalar. Evita desinstalar la aplicación de Slack.

Invocar al DevOps agente a través de Webhook

Los webhooks permiten que los sistemas externos AWS DevOps activen automáticamente las investigaciones de los agentes. Esto permite la integración con sistemas de emisión de tickets, herramientas de monitoreo y otras plataformas que pueden enviar solicitudes HTTP cuando se producen incidentes.

Requisitos previos

Antes de configurar el acceso a los webhooks, asegúrate de tener:

- Un espacio de agente configurado en AWS DevOps Agent
- Acceso a la consola del AWS DevOps agente
- El sistema externo que enviará las solicitudes de webhook

Tipos de webhook

AWS DevOps El agente admite los siguientes tipos de webhooks:

- Webhooks específicos para la integración: se generan automáticamente al configurar integraciones de terceros, como Dynatrace, Splunk, Datadog, New Relic o Slack. ServiceNow Estos webhooks están asociados a la integración específica y utilizan métodos de autenticación determinados por el tipo de integración

- Webhooks genéricos: se pueden crear manualmente para iniciar investigaciones desde cualquier fuente que no esté incluida en una integración específica. Los webhooks genéricos utilizan actualmente la autenticación HMAC (el token portador no está disponible actualmente).
- Webhooks de alertas de Grafana: Grafana puede enviar notificaciones de alerta directamente al AWS DevOps agente a través de los puntos de contacto de webhook. Para obtener instrucciones de configuración que incluyen una plantilla de notificación personalizada, consulte Conectar Grafana.

Métodos de autenticación de Webhook

El método de autenticación de tu webhook depende de la integración a la que esté asociado:

Autenticación HMAC: utilizada por:

- Webhooks de integración con Dynatrace
- Webhooks genéricos (no vinculados a una integración específica de terceros)

Autenticación mediante token de portador: utilizada por:

- Webhooks de integración con Splunk
- Webhooks de integración de Datadog
- Webhooks de integración de New Relic
- ServiceNow webhooks de integración
- Webhooks de integración de Slack

Configurar el acceso a los webhooks

Paso 1: Navega hasta la configuración del webhook

1. Inicie sesión en la consola AWS de administración y navegue hasta la consola del AWS DevOps agente
2. Seleccione su espacio de agente
3. Ve a la pestaña Capacidades
4. En la sección Webhook, haga clic en Configurar

Paso 2: Generar las credenciales del webhook

Para webhooks específicos de la integración:

Los webhooks se generan automáticamente al completar la configuración de una integración de terceros. La URL y las credenciales del punto final del webhook se proporcionan al final del proceso de configuración de la integración.

Para los webhooks genéricos:

1. Haz clic en Generar webhook
2. El sistema generará un key pair HMAC
3. Guarde de forma segura la clave y el secreto generados; no podrá volver a recuperarlos
4. Copia la URL del punto de conexión del webhook proporcionada

Paso 3: Configura tu sistema externo

Utilice la URL y las credenciales del punto de conexión del webhook para configurar su sistema externo y enviar solicitudes al AWS DevOps agente. Los pasos de configuración específicos dependen del sistema externo.

Administrar las credenciales de webhook

Eliminar credenciales: para eliminar las credenciales del webhook, vaya a la sección de configuración del webhook y haga clic en Eliminar. Tras eliminar las credenciales, el punto final del webhook ya no aceptará solicitudes hasta que generes nuevas credenciales.

Regeneración de credenciales: para generar nuevas credenciales, elimine primero las credenciales existentes y, a continuación, genere un nuevo key pair o token.

Uso del webhook

Formato de solicitud de webhook

Para iniciar una investigación, tu sistema externo debe enviar una solicitud HTTP POST a la URL del punto final del webhook.

Para la versión 1 (autenticación HMAC):

Encabezados:

- `Content-Type: application/json`
- `x-amzn-event-signature: <HMAC signature>`
- `x-amzn-event-timestamp: <+%Y-%m-%dT%H:%M:%S.000Z>`

La firma HMAC se genera al firmar el cuerpo de la solicitud con tu clave secreta mediante el SHA-256.

Para la versión 2 (autenticación con token de portador):

Encabezados:

- `Content-Type: application/json`
- `Authorization: Bearer <your-token>`

Cuerpo de la solicitud:

El organismo solicitante debe incluir información sobre el 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"
    }
  }
}
```

Código de ejemplo

Versión 1 (autenticación 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);
});
```

Versión 1 (autenticación 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"
```

Versión 2 (autenticación con token de portador) -: 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),
  });
}
```

Versión 2 (autenticación por token de portador) - 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"
```

Solución de problemas de webhooks

Si no recibes un 200

Un 200 y un mensaje como el webhook recibido indican que la autenticación se ha aprobado y que el mensaje se ha puesto en cola para que el sistema lo verifique y procese. Si no obtienes un 200 sino un 4xx, lo más probable es que haya algún problema con la autenticación o los encabezados. Intenta enviarlo manualmente usando las opciones curl para ayudar a depurar la autenticación.

Si recibes un 200 pero no se inicia una investigación

La causa probable es una carga mal formateada.

1. Compruebe que tanto la marca de tiempo como el identificador del incidente estén actualizados y sean únicos. Los mensajes duplicados se deduplican.
2. Comprueba que el mensaje es un JSON válido
3. Comprueba que el formato es correcto

Si recibes un 200$ y la investigación se cancela inmediatamente

Lo más probable es que hayas alcanzado el límite del mes. Hable con su persona de AWS contacto para solicitar un cambio en el límite de la tarifa, si es necesario.

Temas relacionados

- the section called “Creación de un espacio de agentes”
- the section called “¿Qué es una aplicación web para DevOps agentes?”
- the section called “DevOps Permisos de IAM para agentes”

Integración de AWS DevOps Agent con Amazon EventBridge

Puede integrar AWS DevOps Agent con sus aplicaciones basadas en eventos utilizando los eventos que se producen durante los ciclos de vida de la investigación y la mitigación. AWS DevOps El agente envía eventos a Amazon EventBridge cuando cambia el estado de una investigación o mitigación. A continuación, puede crear EventBridge reglas que actúen en función de estos eventos.

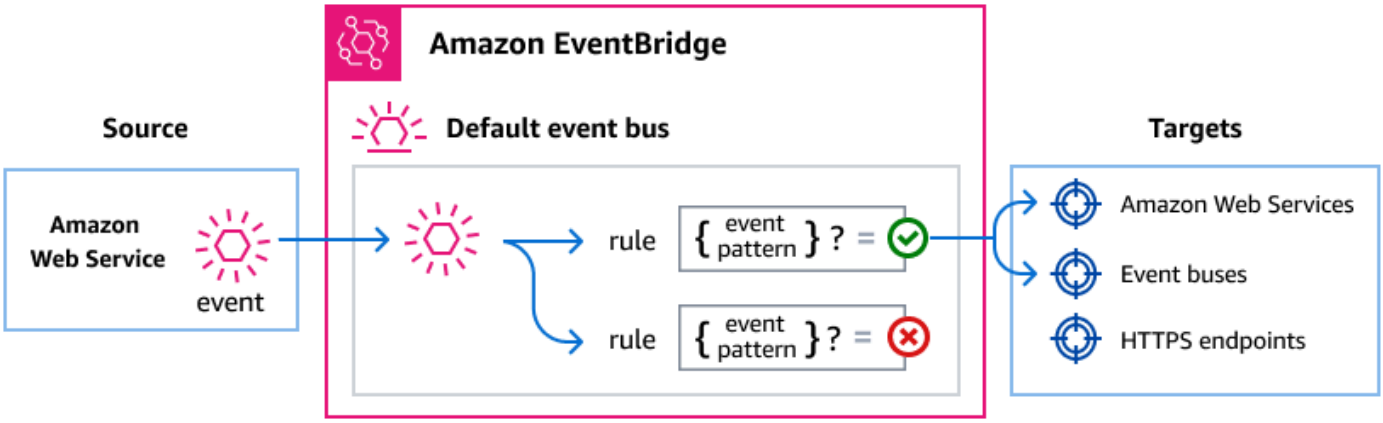
Por ejemplo, puede crear reglas que realicen las siguientes acciones:

- Invoque una función AWS Lambda para procesar los resultados de la investigación cuando se complete una investigación.
- Envía una notificación a Amazon SNS cuando una investigación falle o se agote el tiempo de espera.
- Actualice un sistema de emisión de entradas cuando se cree una nueva investigación.
- Inicie un flujo de trabajo de AWS Step Functions cuando se complete una acción de mitigación.

¿Cómo EventBridge dirige los eventos AWS DevOps del agente?

AWS DevOps El agente envía los eventos al bus de eventos EventBridge predeterminado. EventBridge a continuación, evalúa los eventos según las reglas que usted cree. Cuando un evento coincide con el patrón de eventos de una regla, EventBridge envía el evento a los destinos especificados.

El siguiente diagrama muestra cómo se distribuyen EventBridge los eventos AWS DevOps del agente.



1. AWS DevOps El agente envía un evento al bus de eventos EventBridge predeterminado cuando cambia el estado del ciclo de vida de una investigación o mitigación.
2. EventBridge evalúa el evento según las reglas que usted creó.
3. Si el evento coincide con el patrón de eventos de una regla, EventBridge envía el evento a los destinos especificados en la regla.

AWS DevOps Eventos del agente

AWS DevOps El agente envía los siguientes eventos a EventBridge. Todos los eventos utilizan la fuente`aws.aidevops`.

Eventos de investigación respaldados

detail-type	Description (Descripción)
`Investigation Created`	Se creó una investigación en el espacio de agentes.
`Investigation Priority Updated`	Se cambió la prioridad de una investigación.
`Investigation In Progress`	Una investigación inició un análisis activo.
`Investigation Completed`	Una investigación terminó satisfactoriamente con los hallazgos.

detail-type	Description (Descripción)
`Investigation Failed`	Se detectó un error en la investigación y no se pudo completar.
`Investigation Timed Out`	Una investigación superó la duración máxima permitida.
`Investigation Cancelled`	Se canceló una investigación antes de su finalización.
`Investigation Pending Triage`	Hay una investigación pendiente de clasificación antes de que comience el análisis activo.
`Investigation Linked`	Una investigación estaba relacionada con un incidente o una multa relacionados.

Eventos de mitigación compatibles

detail-type	Description (Descripción)
`Mitigation In Progress`	Se inició una acción de mitigación.
`Mitigation Completed`	Una acción de mitigación finalizó satisfactoriamente.
`Mitigation Failed`	Una acción de mitigación detectó un error y no se pudo completar.
`Mitigation Timed Out`	Una acción de mitigación ha superado la duración máxima permitida.
`Mitigation Cancelled`	Se canceló una acción de mitigación antes de completarse.

Para obtener descripciones de campo detalladas y ejemplos de eventos, consultethe section called “AWS DevOps Referencia detallada de eventos de agentes”.

Crear patrones de eventos que coincidan con los eventos del AWS DevOps agente

EventBridge las reglas utilizan patrones de eventos para seleccionar eventos y enviarlos a los objetivos. Un patrón de eventos coincide con la estructura de los eventos que gestiona. Los patrones de eventos se crean para filtrar los eventos del AWS DevOps agente en función de los campos de eventos.

Los siguientes ejemplos muestran patrones de eventos para casos de uso comunes.

Haga coincidir todos los eventos AWS DevOps del agente

El siguiente patrón de eventos coincide con todos los eventos del AWS DevOps agente.

```
{
  "source": ["aws.aidevops"]
}
```

Coincide solo con los eventos de investigación

El siguiente patrón de eventos utiliza una coincidencia de prefijos para seleccionar solo los eventos del ciclo de vida de la investigación.

```
{
  "source": ["aws.aidevops"],
  "detail-type": [{"prefix": "Investigation"}]
}
```

Haga coincidir solo los eventos de finalización y error

El siguiente patrón de eventos coincide con los eventos de las investigaciones y mitigaciones finalizadas o fallidas.

```
{
  "source": ["aws.aidevops"],
  "detail-type": [
    "Investigation Completed",
    "Investigation Failed",
    "Mitigation Completed",
    "Mitigation Failed"
```

```
  ]
}
```

Haga coincidir los eventos de un espacio de agentes específico

El siguiente patrón de eventos coincide con los eventos de un espacio de agentes específico.

```
{
  "source": ["aws.aidevops"],
  "detail": {
    "metadata": {
      "agent_space_id": ["your-agent-space-id"]
    }
  }
}
```

Para obtener más información sobre los patrones de eventos, consulta los patrones de EventBridge eventos de Amazon en la Guía del EventBridge usuario de Amazon.

EventBridge Permisos de Amazon

AWS DevOps El agente no necesita permisos adicionales para enviar eventos a EventBridge. Los eventos se envían automáticamente al bus de eventos predeterminado.

En función de los destinos que configure para sus EventBridge reglas, es posible que necesite añadir permisos específicos. Para obtener más información sobre los permisos necesarios para los objetivos, consulta Uso de políticas basadas en recursos para Amazon EventBridge en la Guía del EventBridge usuario de Amazon.

Recursos adicionales EventBridge

Para obtener más información sobre EventBridge los conceptos y la configuración, consulta los siguientes temas de la Guía del EventBridge usuario de Amazon:

- EventBridge autobuses de eventos
- EventBridge eventos
- EventBridge patrones de eventos
- EventBridge reglas
- EventBridge objetivos

AWS DevOps Referencia detallada de eventos de agentes

Los eventos de AWS los servicios tienen campos de metadatos comunes `source`detail-type, como`account`,`region`, ytime. Estos eventos también contienen un `detail` campo con datos específicos del servicio. En el caso de los eventos del AWS DevOps agente, el símbolo `source` es siempre `aws.aidevops` y el `detail-type` identifica el evento específico.

Eventos de investigación

Los siguientes `detail-type` valores identifican los eventos de investigación:

- `Investigation Created`
- `Investigation Priority Updated`
- `Investigation In Progress`
- `Investigation Completed`
- `Investigation Failed`
- `Investigation Timed Out`
- `Investigation Cancelled`
- `Investigation Pending Triage`
- `Investigation Linked`

Los `detail-type` campos `source` y se incluyen a continuación porque contienen valores específicos para los eventos AWS DevOps del agente. Para ver las definiciones de los demás campos de metadatos que se incluyen en todos los eventos, consulte Estructura de eventos en la referencia de Amazon EventBridge Events.

La siguiente es la estructura JSON para los eventos de investigación.

```
{
  . . .,
  "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 el tipo de evento. En el caso de los eventos de investigación, este es uno de los nombres de eventos enumerados anteriormente.

`source`Identifica el servicio que generó el evento. Para los eventos del AWS DevOps agente, este valor es`aws.aidevops`.

`detail`Un objeto JSON que contiene datos específicos del evento. El `detail` objeto incluye los siguientes campos:

- `version`(cadena): la versión esquemática del detalle del evento. Actualmente`1.0.0`.
- `metadata.agent_space_id`(cadena): el identificador único del espacio de agentes donde se originó el evento.
- `metadata.task_id`(cadena): el identificador único de la tarea.
- `metadata.execution_id`(cadena): el identificador único de la ejecución. Está presente cuando se ha asignado una ejecución a la investigación.
- `data.task_type`(cadena): el tipo de tarea. Valor:`INVESTIGATION`.
- `data.priority`(cadena): el nivel de prioridad. Valores:`CRITICAL`,`HIGH`,`MEDIUM`,`LOW`,`MINIMAL`.
- `data.status`(cadena): el estado actual. Valores:`PENDING_START`, `IN_PROGRESSCOMPLETED`,`FAILED`,`TIMED_OUT`,`CANCELLED`,`PENDING_TRIAGE`,`LINKED`.
- `data.created_at`(cadena): marca de tiempo ISO 8601 en la que se creó la tarea.
- `data.updated_at`(cadena): fecha y hora ISO 8601 de la última actualización de la tarea.
- `data.summary_record_id`(cadena): identificador del acta resumida que contiene los resultados de la investigación. Se incluye cuando se genera un resumen de la investigación finalizada. Puede recuperar el contenido del resumen a través de la API AWS DevOps

Agent utilizando este identificador para buscar el registro del diario con un tipo de registro de`investigation_summary_md`.

Ejemplo: evento «Investigación completada»

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

Ejemplo: evento fallido en la investigación

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

Eventos de mitigación

Los siguientes `detail-type` valores identifican los eventos de mitigación:

- `Mitigation In Progress`
- `Mitigation Completed`
- `Mitigation Failed`
- `Mitigation Timed Out`
- `Mitigation Cancelled`

Los `detail-type` campos `source` y se incluyen a continuación porque contienen valores específicos para los eventos AWS DevOps del agente. Para ver las definiciones de los demás campos de metadatos que se incluyen en todos los eventos, consulte Estructura de eventos en la referencia de Amazon EventBridge Events.

La siguiente es la estructura de JSON para los eventos de mitigación.

```
{
  . . .,
  "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 el tipo de evento. En el caso de los eventos de mitigación, este es uno de los nombres de eventos enumerados anteriormente.

`source`Identifica el servicio que generó el evento. Para los eventos del AWS DevOps agente, este valor es`aws.aidevops`.

`detail`Un objeto JSON que contiene datos específicos del evento. El `detail` objeto incluye los siguientes campos:

- `version`(cadena): la versión esquemática del detalle del evento. Actualmente`1.0.0`.
- `metadata.agent_space_id`(cadena): el identificador único del espacio de agentes donde se originó el evento.
- `metadata.task_id`(cadena): el identificador único de la tarea.
- `metadata.execution_id`(cadena): el identificador único de la ejecución. Está presente cuando se ha asignado una ejecución a la mitigación.
- `data.task_type`(cadena): el tipo de tarea. Valor:`INVESTIGATION`.
- `data.priority`(cadena): el nivel de prioridad. Valores:`CRITICAL`,`HIGH`,`MEDIUM`,`LOW`,`MINIMAL`.

- `data.status`(cadena): el estado actual. Valores:`IN_PROGRESS`,`COMPLETED`,`FAILED`,`TIMED_OUT`,`CANCELLED`.
- `data.created_at`(cadena): marca de tiempo ISO 8601 en la que se creó la tarea.
- `data.updated_at`(cadena): fecha y hora ISO 8601 de la última actualización de la tarea.
- `data.summary_record_id`(cadena): el identificador del registro resumido que contiene los resultados de la mitigación. Se incluye cuando se genera un resumen de la mitigación completada. Puede recuperar el contenido del resumen a través de la API del AWS DevOps agente utilizando este identificador para buscar el registro del diario con un tipo de registro de`mitigation_summary_md`.

Ejemplo: evento Mitigación completada

```
{
  "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"
    }
  }
```

```
}
```

Ejemplo: Evento fallido de mitigación

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

Registros y métricas de Vended

Puedes monitorear tus espacios de agente y tus operaciones de servicio mediante CloudWatch métricas y registros vendidos de Amazon. En este tema se describen las CloudWatch métricas que el AWS DevOps agente publica automáticamente en tu cuenta y los registros de ventas que puedes configurar para realizar entregas en los destinos que prefieras.

Métricas vendidas CloudWatch

AWS DevOps El agente publica automáticamente las métricas CloudWatch en Amazon en tu cuenta. Estas métricas están disponibles sin ninguna configuración. Puede utilizarlos para supervisar el uso, realizar un seguimiento de la actividad operativa y crear alarmas.

Rol vinculado a servicio

Para que CloudWatch las métricas de Amazon se publiquen en su cuenta para este servicio, el AWS DevOps agente creará automáticamente el rol vinculado al servicio AWSServiceRoleForAIDevOps Service-Linked Role para usted. Si el rol de IAM que invoca la API no tiene el permiso adecuado, la creación del recurso fallará con un. InvalidParameterException

> **Important**
>
> Los clientes que hayan creado su rol AgentSpace antes del 13 de marzo de 2026 deberán crear manualmente el rol vinculado al servicio de AWSServiceRoleForAIDevoperaciones para que CloudWatch las métricas del AWS DevOps agente se publiquen en su cuenta.

Cree manualmente un rol vinculado al servicio (para clientes existentes)

Realice una de las siguientes acciones:

- En la consola de IAM, cree el rol AWSServiceRoleForAIDevOps en el servicio de agente.AWS DevOps
- Desde la AWS CLI, ejecute el siguiente comando:

```
aws iam create-service-linked-role --aws-service-name aidevops.amazonaws.com
```

Namespace

Todas las métricas se publican en el espacio de `AWS/AIDevOps` nombres.

Dimensiones

Todas las métricas incluyen la siguiente dimensión.

Dimensión	Description (Descripción)
`AgentSpaceUUID`	El identificador único del espacio de agentes. Para agregar métricas en todos los espacios de agentes de su cuenta, utilice expresiones CloudWatch matemáticas u omita el filtro de dimensiones.

Referencia de métricas

Nombre de métrica	Description (Descripción)	Unidad	Frecuencia de publicación	Estadísticas útiles
ConsumedChatRequests	El número de solicitudes de chat que ha consumido el espacio de un agente. Para obtener el recuento total de tu cuenta, usa la `SUM` estadística en todas las `AgentSpaceUUID` dimensiones.	Recuento	Cada 5 minutos	Suma, media
ConsumedInvestigationTime	El tiempo dedicado a realizar investigaciones en un espacio de agentes.	Segundos	Cada 5 minutos	Suma, media, máxima

Nombre de métrica	Description (Descripción)	Unidad	Frecuencia de publicación	Estadísticas útiles
ConsumedEvaluationTime	El tiempo dedicado a ejecutar las evaluaciones en un espacio de agentes.	Segundos	Cada 5 minutos	Suma, media y máxima
TopologyCompletionCount	El número de procesamientos topológicos finalizados. AWS DevOps El agente emite esta métrica cuando una topología termina de procesarse, ya sea desde la creación inicial durante la incorporación, una actualización manual o una actualización diaria programada.	Recuento	Basado en eventos (se emite cada vez que se completa)	Suma, SampleCount

Visualización de las métricas en la CloudWatch consola

1. Abra la consola de CloudWatch .
2. En el panel de navegación, elija Metrics (Métricas) y, a continuación, All metrics (Todas las métricas).

3. Elija el espacio de nombres AWS AIDev/Ops.
4. Seleccione Por AgentSpace para ver las métricas de sus espacios de agentes.

> **Note**
>
> Puede crear CloudWatch alarmas en estas métricas para recibir notificaciones cuando el uso supere un umbral. Por ejemplo, crea una alarma `ConsumedChatRequests` para controlar el consumo de solicitudes de chat.

Requisitos previos

Antes de configurar la entrega de registros, asegúrese de tener lo siguiente:

- Una AWS cuenta activa con acceso a la consola del AWS DevOps agente
- Un director de IAM con permisos para la entrega de CloudWatch registros APIs
- (Opcional) Un bucket de Amazon S3 o una transmisión de entrega de Amazon Data Firehose, si piensa utilizarlos como destinos de registro

Registros proporcionados

AWS DevOps El agente admite registros vendidos que proporcionan visibilidad de los eventos que procesan los espacios de agente y los registros de servicios. Los registros vendidos utilizan la infraestructura de Amazon CloudWatch Logs para entregar los registros a su destino preferido.

Para utilizar los registros vendidos, debe configurar un destino de entrega. Se admiten los siguientes destinos:

- Amazon CloudWatch Logs: un grupo de registros en tu cuenta
- Amazon S3: un bucket de S3 en su cuenta
- Amazon Data Firehose: un flujo de entrega de Firehose en tu cuenta

Tipos de registro admitidos

Se admite un solo tipo de registro:. `APPLICATION_LOGS` Este tipo de registro cubre todos los eventos operativos que emite el servicio.

Registra los tipos de eventos

En la siguiente tabla se resumen los eventos que el AWS DevOps agente registra.

Event	Description (Descripción)	Nivel de registro
El agente recibió un evento entrante	Un agente es activado por una fuente integrada y recibe un evento entrante (por ejemplo, un evento de PagerDuty incidente).	INFO
Se descarta el evento entrante del agente	Se descartó un evento entrante antes de que el agente lo procesara. El registro incluye el motivo (por ejemplo, datos con formato incorrecto).	POR DETERMINAR
Fallo en la comunicación saliente del agente	Se produjo un error en la comunicación saliente con una integración de terceros. El registro incluye el identificador de la tarea y el identificador de destino (por ejemplo, un error de autenticación).	POR DETERMINAR
Creación de topología en cola	Se puso en cola un trabajo de creación de topología para su procesamiento.	INFO
Se inició la creación de la topología	Se inició el procesamiento de un trabajo de creación de topología.	INFO
Finalizó la creación de la topología	Se completó el procesami ento de un trabajo de creación de topología. Este evento se	INFO

Event	Description (Descripción)	Nivel de registro
	aplica a la creación inicial, a las actualizaciones y a las actualizaciones diarias.	
Falló el descubrimiento de recursos	Se produjo un error en la detección de recursos durante la creación de la topología.	ERROR
Falló el registro del servicio	El registro del servicio presenta un error irrecuper able	ERROR
La validación de Webhook falla	Cuando el webhook recibido por el agente de Devops no coincide con el esquema esperado	ERROR
Se actualiza el estado de validación de la asociación	Cuando se asocia un espacio de agentes (una primary/ secondary cuenta típica), el estado de la validación cambia de válido a no válido y viceversa (por ejemplo, debido a un rol mal diseñado, que el servicio no puede asumir).	ERROR/INFORMACIÓN

Permisos

AWS DevOps El agente usa registros CloudWatch vendidos (permisos V2) para entregar los registros. Para configurar la entrega de registros, la función de IAM que configura la entrega debe tener los siguientes permisos:

- `aidevops:AllowVendedLogDeliveryForResource`— Necesario para permitir la entrega de registros para el recurso de espacio del agente.

- Permisos para la entrega de CloudWatch registros APIs (`logs:PutDeliverySourcelogs:PutDeliveryDestination`,`logs:CreateDelivery`, y operaciones relacionadas).
- Permisos específicos para el destino de entrega elegido.

Para ver la política de IAM completa que se requiere para cada tipo de destino, consulte los siguientes temas de la Guía del usuario de Amazon CloudWatch Logs:

- Registros enviados a CloudWatch Logs
- Registros enviados a Amazon S3
- Registros enviados a Firehose

Configurar la entrega de registros (consola)

AWS DevOps El agente proporciona dos ubicaciones en la consola AWS de administración para configurar la entrega de registros:

- Página de configuración de registro del servicio: configure la entrega de registros para los eventos de nivel de servicio. Estos registros utilizan el ARN del servicio (`arn:aws:aidevops:<region>:<account-id>:service/<account-id>`) como recurso.
- Página Agent Space: configure la entrega de registros para los eventos específicos de un espacio de agente individual. Estos registros utilizan el ARN (`arn:aws:aidevops:<region>:<account-id>:agentspace/<agent-space-id>`) del espacio de agentes como recurso.

Para configurar la entrega de registros para el registro de un servicio

1. Abra la consola del AWS DevOps agente en la consola AWS de administración.
2. En el panel de navegación, seleccione Configuración.
3. En la pestaña Proveedores de capacidades > Registros, seleccione Configurar.
4. En Tipo de destino, elija una de las siguientes opciones:
5. CloudWatch Registros: seleccione o cree un grupo de registros.
6. Amazon S3: introduzca el ARN del bucket S3.
7. Amazon Data Firehose: selecciona o crea una transmisión de entrega de Firehose.
8. Para la configuración adicional (opcional), puede especificar las siguientes opciones:

a. En Selección de campos, seleccione los nombres de los campos de registro que desea entregar en su destino. Puede seleccionar campos de registro de acceso y un subconjunto de campos de registro de acceso en tiempo real.
b. (Solo Amazon S3) En Partición, especifique la ruta para particionar los datos del archivo de registro.
c. (Solo Amazon S3) En Formato de archivo compatible con Hive, puede seleccionar la casilla de verificación para utilizar rutas de S3 compatibles con Hive. Esto ayuda a simplificar la carga de nuevos datos en las herramientas compatibles con Hive.
d. En Formato de salida, especifique el formato que prefiera.
e. En Delimitador de campo, especifique cómo separar los campos de registro.

9. Seleccione Save.
10. Compruebe que el estado de la entrega sea Activo.

Para configurar la entrega de registros para un espacio de agentes

1. Abra la consola del AWS DevOps agente en la consola AWS de administración.
2. Elija el espacio de agentes que desee configurar.
3. En la pestaña Configuración, elija Configurar.
4. En Tipo de destino, elija una de las siguientes opciones:
5. CloudWatch Registros: seleccione o cree un grupo de registros.
6. Amazon S3: introduzca el ARN del bucket S3.
7. Amazon Data Firehose: selecciona o crea una transmisión de entrega de Firehose.
8. Para ajustes adicionales (*opcional), puede especificar las siguientes opciones:
   a. En Selección de campos, seleccione los nombres de los campos de registro que desea entregar en su destino. Puede seleccionar campos de registro de acceso y un subconjunto de campos de registro de acceso en tiempo real.
   b. (Solo Amazon S3) En Partición, especifique la ruta para particionar los datos del archivo de registro.
   c. (Solo Amazon S3) En Formato de archivo compatible con Hive, puede seleccionar la casilla de verificación para utilizar rutas de S3 compatibles con Hive. Esto ayuda a simplificar la carga de nuevos datos en las herramientas compatibles con Hive.
   d. En Formato de salida, especifique el formato que prefiera.
   e. En Delimitador de campo, especifique cómo separar los campos de registro.

9. Seleccione Save.
10. Comprueba que el estado de la entrega sea Activo.

Configure la entrega de registros (CloudWatch API)

También puede usar la API de CloudWatch registros para configurar la entrega de registros mediante programación. La entrega de un registro funcional consta de tres elementos:

- A DeliverySource: representa el recurso de espacio de AWS DevOps agentes que genera los registros.
- A DeliveryDestination: Representa el destino donde se escriben los registros.
- Una entrega: conecta una fuente de entrega con un destino de entrega.

Paso 1: Crear una fuente de entrega

Utilice la PutDeliverySourceoperación para crear una fuente de entrega. Pase el ARN de su recurso de espacio de AWS DevOps agente y especifíquelo `APPLICATION_LOGS` como tipo de registro.

El siguiente ejemplo crea una fuente de entrega para un espacio de agentes:

```
{
    "name": "my-agent-space-delivery-source",
    "resourceArn": "arn:aws:aidevops:us-east-1:123456789012:agentspace/my-agent-space-
id",
    "logType": "APPLICATION_LOGS"
}
```

El siguiente ejemplo crea una fuente de entrega para el servicio:

```
{
    "name": "my-service-delivery-source",
    "resourceArn": "arn:aws:aidevops:us-east-1:123456789012:service",
    "logType": "APPLICATION_LOGS"
}
```

Paso 2: Crear un destino de entrega

Utilice la PutDeliveryDestinationoperación para configurar dónde se almacenan los registros. Puede elegir Amazon CloudWatch Logs, Amazon S3 o Amazon Data Firehose.

En el siguiente ejemplo, se crea un destino de CloudWatch Logs:

```
{
    "name": "my-cwl-destination",
    "deliveryDestinationConfiguration": {
        "destinationResourceArn": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/
aidevops/my-agent-space"
    },
    "outputFormat": "json"
}
```

El siguiente ejemplo crea un destino de Amazon S3:

```
{
    "name": "my-s3-destination",
    "deliveryDestinationConfiguration": {
        "destinationResourceArn": "arn:aws:s3:::my-aidevops-logs-bucket"
    },
    "outputFormat": "json"
}
```

El siguiente ejemplo crea un destino Amazon Data Firehose:

```
{
    "name": "my-firehose-destination",
    "deliveryDestinationConfiguration": {
        "destinationResourceArn": "arn:aws:firehose:us-
east-1:123456789012:deliverystream/my-aidevops-log-stream"
    },
    "outputFormat": "json"
}
```

> **Note**
>
> Si entrega registros entre cuentas, debe utilizarlos PutDeliveryDestinationPolicyen la cuenta de destino para autorizar la entrega.

Si quieres usar CloudFormation, puedes usar lo siguiente:

- Delivery

- DeliveryDestination
- DeliverySource

El `ResourceArn` es el `AgentSpaceArn` y el `LogType` admitido debe ser `APPLICATION_LOGS`.

Paso 3: Crea una entrega

Utilice la CreateDeliveryoperación para vincular la fuente de entrega con el destino de entrega.

```
{
    "deliverySourceName": "my-agent-space-delivery-source",
    "deliveryDestinationArn": "arn:aws:logs:us-east-1:123456789012:delivery-
destination:my-cwl-destination"
}
```

AWS CloudFormation

También puede configurar la entrega de registros AWS CloudFormation mediante los siguientes recursos:

- AWS: :Registros:: DeliverySource
- AWS: :Registros:: DeliveryDestination
- AWS:: Registros:: Entrega

`ResourceArn`Configúrelo en el espacio de AWS DevOps agente o en el ARN del servicio y configúrelo en. `LogType` `APPLICATION_LOGS`

Referencia del esquema de registro

AWS DevOps El agente usa un esquema de registro compartido en todos los tipos de eventos. No todos los eventos de registro utilizan todos los campos.

En la siguiente tabla se describen los campos del esquema de registro.

Campo	Tipo	Description (Descripción)
event_timestamp	Largo	Marca de tiempo de Unix del momento en que ocurrió el evento

Campo	Tipo	Description (Descripción)
resource_arn	Cadena	ARN del recurso que generó el evento
optional_account_id	Cadena	AWS ID de cuenta asociada al registro.
nivel_opcional	Cadena	Nivel de registro:,, `INFO` `WARN` `ERROR`
optional_agent_space_id	Cadena	Identificador del espacio de agentes.
optional_association_id	Cadena	Identificador de asociación para el registro.
estado_opcional	Cadena	Estado de la operación de topología.
optional_webhook_id	Cadena	Identificador de webhook.
optional_mcp_endpoint_url	Cadena	URL del punto final del servidor MCP
tipo_de_servicio_opcional	Cadena	Tipo de servicio:`DYNATRACE` ,,,,. `DATADOG` `GITHUB` `SLACK` `SERVICENOW`
optional_service_endpoint_url	Cadena	URL de punto final para integraciones de terceros.
optional_service_id	Cadena	Identificador de la fuente.
request_id	Cadena	Identificador de solicitud para correlacionarlo con los tickets de soporte AWS CloudTrail o de soporte.

Campo	Tipo	Description (Descripción)
operación_opcional	Cadena	Nombre de la operación que se realizó.
tipo_de_tarea opcional	Cadena	Tipo de tarea pendiente del agente: o `INVESTIGATION` `EVALUATION`
optional_task_id	Cadena	Identificador de tareas pendientes de Agent Backlog Task. IDAgent
referencia_opcional	Cadena	Referencia de una tarea de un agente (por ejemplo, un ticket de Jira).
Tipo_de_error opcional	Cadena	Tipo de error
mensaje_de_error opcional	Cadena	Descripción del error cuando se produce un error en una operación.
detalles_opcionales	Cadena (JSON)	Carga útil de eventos específica del servicio que contiene los parámetros y resultados de la operación.

Administre e inhabilite la entrega de registros

Puede modificar o eliminar la entrega de registros en cualquier momento desde la consola de AWS DevOps agente de la consola AWS de administración o mediante la API de CloudWatch registros.

Gestione la entrega de registros (consola)

1. Abra la consola del AWS DevOps agente en la consola AWS de administración.
2. Diríjase a la página de configuración (para los registros de nivel de servicio) o a la página específica de Agent Space (para los registros de nivel de Agent Space).

3. En la pestaña Configuración (para los registros a nivel de espacio de agente) o en la pestaña Proveedores de capacidades > Registros (para los registros a nivel de servicio), seleccione la entrega que desee modificar.
4. Actualice la configuración según sea necesario y seleccione Guardar.

> Nota: No puedes cambiar el tipo de destino de un envío existente. Para cambiar el tipo de destino, elimina la entrega actual y crea una nueva.

Inhabilite la entrega de registros (consola)

1. Abra la consola del AWS DevOps agente en la consola AWS de administración.
2. Diríjase a la página de configuración (para los registros de nivel de servicio) o a la página específica de Agent Space (para los registros de nivel de Agent Space).
3. En la pestaña Configuración (para los registros a nivel de espacio de agente) o en la pestaña Proveedores de capacidades > Registros (para los registros a nivel de servicio), seleccione la entrega que desee eliminar.
4. Seleccione Eliminar y confirme.

Inhabilite la entrega de registros (API)

Para eliminar una entrega de registros mediante la API, elimine los recursos en el siguiente orden:

1. Elimine la entrega mediante DeleteDelivery.
2. Elimine la fuente de entrega mediante DeleteDeliverySource.
3. (Opcional) Si el destino de entrega ya no es necesario, elimínelo utilizando DeleteDeliveryDestination.

> **Important**
>
> Usted es responsable de eliminar los recursos de entrega de registros después de eliminar el recurso de espacio de agente que genera los registros (por ejemplo, después de eliminar un espacio de agente). Si no eliminas estos recursos, es posible que queden configuraciones de entrega huérfanas.

Precios

El AWS DevOps agente no cobra por habilitar los registros vendidos. Sin embargo, puede incurrir en cargos por la entrega, la ingesta, el almacenamiento o el acceso, según el destino de entrega de los registros que seleccione. Para obtener más información sobre los precios, consulta Vended Logs en la pestaña Logs de Amazon CloudWatch Pricing.

Para conocer los precios específicos de cada destino, consulta lo siguiente:

- Precios de Amazon CloudWatch Logs
- Precios de Amazon S3
- Precios de Amazon Data Firehose

Conexión a herramientas alojadas de forma privada

Descripción general de las conexiones privadas

AWS DevOps El agente se puede ampliar con herramientas personalizadas del Model Context Protocol (MCP) y otras integraciones que permiten al agente acceder a los sistemas internos, como los registros de paquetes privados, las plataformas de observabilidad autohospedadas, la documentación APIs interna y las instancias de control de código fuente (consulte:). Configuración de las capacidades del AWS DevOps agente Estos servicios suelen ejecutarse dentro de una Amazon Virtual Private Cloud (Amazon VPC) con acceso a Internet público o restringido, lo que significa que el AWS DevOps agente no puede acceder a ellos de forma predeterminada.

Las conexiones privadas para AWS DevOps Agent le permiten conectar de forma segura su Agent Space a los servicios que se ejecutan en su VPC sin exponerlos a la Internet pública. Las conexiones privadas funcionan con cualquier integración que necesite llegar a un punto final privado, incluidos los servidores MCP, las instancias de Grafana o Splunk autohospedadas y los sistemas de control de código fuente como GitHub Enterprise Server y Self-Managed. GitLab

> **Note**
>
> Si sus herramientas alojadas de forma privada realizan solicitudes salientes al AWS DevOps agente desde su VPC, este tráfico también se puede proteger mediante un punto de enlace de la VPC para que permanezca dentro de la red. AWS Por ejemplo, esto se puede utilizar con herramientas que activan el DevOps agente mediante eventos de webhook (consulte:).

the section called “Invocar al DevOps agente a través de Webhook” Para obtener más información, consulte the section called “Puntos de enlace de la VPC (AWS PrivateLink)”.

Cómo funcionan las conexiones privadas

Una conexión privada crea una ruta de red segura entre el AWS DevOps agente y un recurso de destino en la VPC. De manera clandestina, AWS DevOps Agent utiliza Amazon VPC Lattice para establecer esta ruta de conectividad privada segura. VPC Lattice es un servicio de redes de aplicaciones que le permite conectar, proteger y supervisar la comunicación entre aplicaciones VPCs, cuentas y tipos de procesamiento, sin administrar la infraestructura de red subyacente.

Al crear una conexión privada, ocurre lo siguiente:

- Usted proporciona la VPC, las subredes y (opcionalmente) los grupos de seguridad que tienen conectividad de red con el servicio de destino.
- AWS DevOps El agente crea una puerta de enlace de recursos gestionada por un servicio y aprovisiona sus interfaces de red elásticas (ENIs) en las subredes que especificó.
- El agente usa la puerta de enlace de recursos para enrutar el tráfico a la dirección IP o el nombre DNS del servicio de destino a través de la ruta de red privada.

El AWS DevOps agente administra completamente la puerta de enlace de recursos y aparece como un recurso de solo lectura en su cuenta (nombre`aidevops-{your-private-connection-name}`). No necesita configurarlo ni mantenerlo. Los únicos recursos que se crean en la VPC se encuentran ENIs en las subredes que especifique. ENIs Sirven como punto de entrada para el tráfico privado y el servicio los administra en su totalidad. No aceptan conexiones entrantes de Internet y tú conservas el control total sobre su tráfico a través de tus propios grupos de seguridad.

Seguridad

Las conexiones privadas están diseñadas con varios niveles de seguridad:

- Sin exposición a Internet pública: todo el tráfico entre el AWS DevOps agente y el servicio de destino permanece en la AWS red. Su servicio nunca necesita una dirección IP pública ni una puerta de enlace de Internet.
- Puerta de enlace de recursos controlada por el servicio: la puerta de enlace de recursos gestionados por el servicio es de solo lectura en su cuenta. Solo la puede usar el AWS DevOps

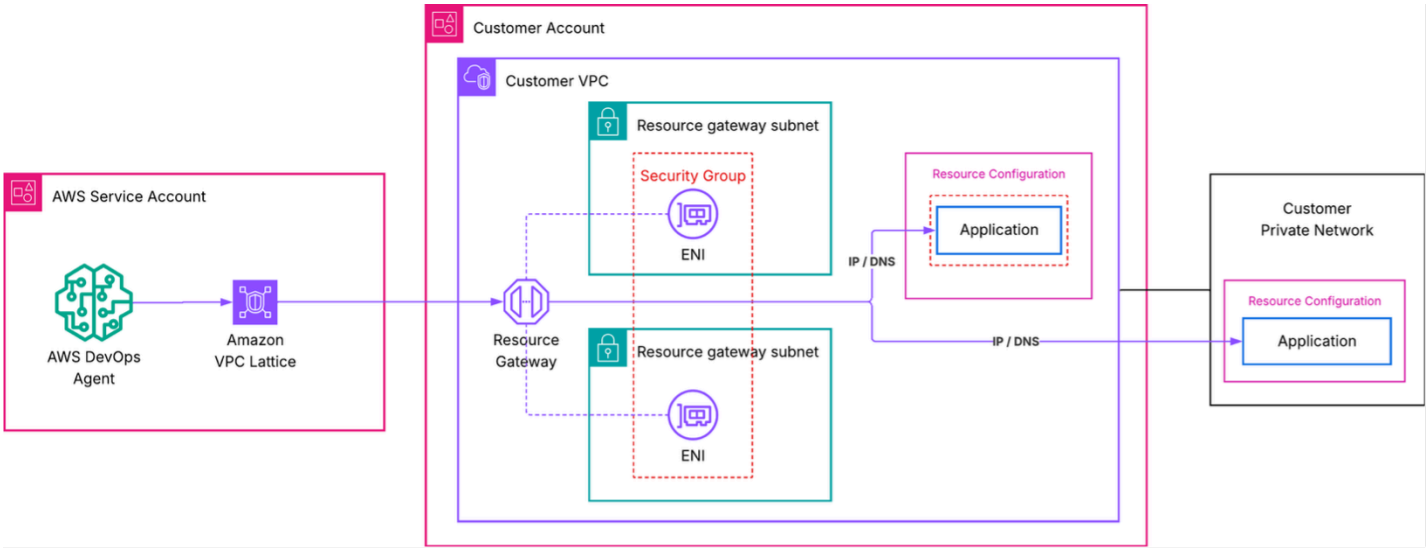
agente y ningún otro servicio o entidad principal puede dirigir el tráfico a través de ella. Puede verificarlo en AWS CloudTraillos registros, que registran todas las llamadas a la API de VPC Lattice.

- Sus grupos de seguridad, sus reglas: usted controla el tráfico entrante y saliente que llega ENIs a los grupos de seguridad de los que es propietario y que administra. Si no especifica los grupos de seguridad, el AWS DevOps agente crea un grupo de seguridad predeterminado con el alcance de los puertos que defina.
- Funciones vinculadas a servicios con privilegios mínimos: el AWS DevOps agente utiliza una función vinculada a servicios para crear solo los recursos necesarios de VPC Lattice y Amazon EC2. Este rol se limita a los recursos etiquetados con `AWSAIDevOpsManaged` y no puede acceder a ningún otro recurso de su cuenta.

> **Note**
>
> Si su organización tiene políticas de control de servicios (SCPs) que restringen las acciones de la API VPC Lattice, la puerta de enlace de recursos gestionados por el servicio se crea mediante un rol vinculado al servicio. Asegúrese de SCPs permitir las acciones necesarias para el rol vinculado al servicio.

Arquitectura

El siguiente diagrama muestra la ruta de red de una conexión privada.



En esta arquitectura:

- AWS DevOps El agente inicia una solicitud a su servicio de destino.
- Amazon VPC Lattice enruta la solicitud a través de la puerta de enlace de recursos gestionados por el servicio de su VPC. Para obtener información sobre las configuraciones avanzadas que utilizan sus propios recursos de VPC Lattice, consulte Configuración avanzada con los recursos de VPC Lattice existentes.
- Un ENI de su VPC recibe el tráfico y lo reenvía a la dirección IP o al nombre DNS del servicio de destino.
- Sus grupos de seguridad determinan qué tráfico se permite a través de. ENIs
- Desde la perspectiva del servicio de destino, la solicitud se origina en direcciones IP privadas de su ENIs VPC.

Cree una conexión privada

Puede crear una conexión privada mediante la consola AWS de administración o la AWS CLI.

> **Note**
>
> VPC Lattice no admite las siguientes zonas de disponibilidad:`use1-az3`,,,`usw1-az2`,`apne1-az3`, `apne2-az2euc1-az2`,`euw1-az4`. `cac1-az3` `ilc1-az2`

Requisitos previos

Antes de crear una conexión privada, compruebe que dispone de lo siguiente:

- Un espacio de agente activo: necesita un espacio de agente existente en su cuenta. Si no dispone de una, consulte Cómo empezar con AWS DevOps Agent.
- Un servicio de destino al que se pueda acceder de forma privada: se debe poder acceder a su servidor MCP, plataforma de observabilidad u otro servicio en una dirección IP privada conocida o un nombre de DNS de la VPC en la que se implementa la puerta de enlace de recursos. El servicio se puede ejecutar en la misma VPC, en una VPC interconectada o de forma local, siempre que se pueda enrutar desde las subredes de la puerta de enlace de recursos. El servicio debe ofrecer tráfico HTTPS con una versión TLS mínima de 1.2 en un puerto que especifiques al crear la conexión.
- Subredes de la VPC: identifique de 1 a 20 subredes en las que se ENIs crearán. Recomendamos seleccionar subredes en varias zonas de disponibilidad para obtener una alta disponibilidad. Estas

subredes deben tener conectividad de red con el servicio de destino. VPC Lattice puede usar una subred por zona de disponibilidad.

- Grupos de seguridad (opcionales): si desea controlar el tráfico con reglas específicas, prepare hasta cinco grupos de seguridad para adjuntarlos IDs a ellos. ENIs Si omite los grupos de seguridad, el AWS DevOps agente crea un grupo de seguridad predeterminado.

Las conexiones privadas son recursos a nivel de cuenta. Después de crear una conexión privada, puede reutilizarla en varias integraciones y espacios de agentes que necesiten llegar al mismo anfitrión.

Cree una conexión privada mediante la consola

1. Abre la consola del AWS DevOps agente.
2. En el panel de navegación, elija Proveedores de capacidades y, a continuación, elija Conexiones privadas.
3. Seleccione Crear nuevo perfil de conexión.
4. En Nombre, introduzca un nombre descriptivo para la conexión, por ejemplo`my-mcp-tool-connection`.
5. Para la VPC, seleccione la VPC en la que se implementará la puerta de enlace ENIs de recursos.
6. Para las subredes, seleccione una o más subredes (hasta 20). Recomendamos elegir subredes en al menos dos zonas de disponibilidad.
7. Para el tipo de dirección IP, seleccione el tipo de dirección IP del servicio de destino (`IPv4`IPv6, o`DualStack`).
8. (Opcional) En Número de IPv4 direcciones, si IPv4 seleccionó Dualstack como tipo de dirección IP, puede introducir el número de IPv4 direcciones por ENI para su puerta de enlace de recursos. El valor predeterminado es 16 IPv4 direcciones por ENI.
9. (Opcional) Para los grupos de seguridad, seleccione los grupos de seguridad existentes (hasta 5) para restringir el tráfico que puede llegar al servicio de destino. Si no selecciona ninguno, se crea un grupo de seguridad predeterminado.
10. (Opcional) Para los rangos de puertos, especifique los puertos TCP en los que escucha la aplicación de destino (por ejemplo, `443` o`8080-8090`). Puede especificar hasta 11 rangos de puertos.
11. En Dirección de host, introduzca la dirección IP o el nombre DNS del servicio de destino (por ejemplo, `mcp.internal.example.com` o`10.0.1.50`). Se debe poder acceder al servicio desde la VPC seleccionada. Si elige un nombre DNS, debe poder resolverse desde la VPC seleccionada.

12. (Opcional) En el caso de la clave pública del certificado, si la dirección de host que especificó utiliza certificados TLS emitidos por una entidad de certificación privada, introduzca la clave pública del certificado codificada en PEM. Esto permite al AWS DevOps agente confiar en la conexión TLS con el servicio de destino.
13. Elija Crear conexión.

El estado de la conexión cambia a Crear en curso. Este proceso puede tardar hasta 10 minutos. Cuando el estado cambia a Activo, la ruta de red está lista.

Si el estado cambia a Error al crear, compruebe lo siguiente:

- Las subredes que especificó tienen direcciones IP disponibles.
- Su cuenta no ha alcanzado las cuotas de servicio de VPC Lattice.
- No hay políticas de IAM restrictivas que impidan que el rol vinculado al servicio cree recursos.

> **Note**
>
> Estos pasos también se pueden realizar seleccionando un proveedor de capacidades `Create a new private connection` durante el registro. Para obtener más información, consulte Usar una conexión privada con un proveedor de capacidades.

Cree una conexión privada mediante la AWS CLI

Ejecute el siguiente comando para crear una conexión privada. Sustituya los valores de los marcadores de posición por los suyos propios.

```
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 respuesta incluye el nombre de la conexión y un estado de`CREATE_IN_PROGRESS`:

```
{
    "name": "my-mcp-tool-connection",
    "status": "CREATE_IN_PROGRESS",
    "resourceGatewayId": "rgw-0123456789abcdef0",
    "hostAddress": "mcp.internal.example.com",
    "vpcId": "vpc-0123456789abcdef0"
}
```

Para comprobar el estado de la conexión, utilice el `describe-private-connection` comando:

```
aws devops-agent describe-private-connection \
    --name my-mcp-tool-connection
```

Cuando el estado es`ACTIVE`, tu conexión privada está lista para usarse.

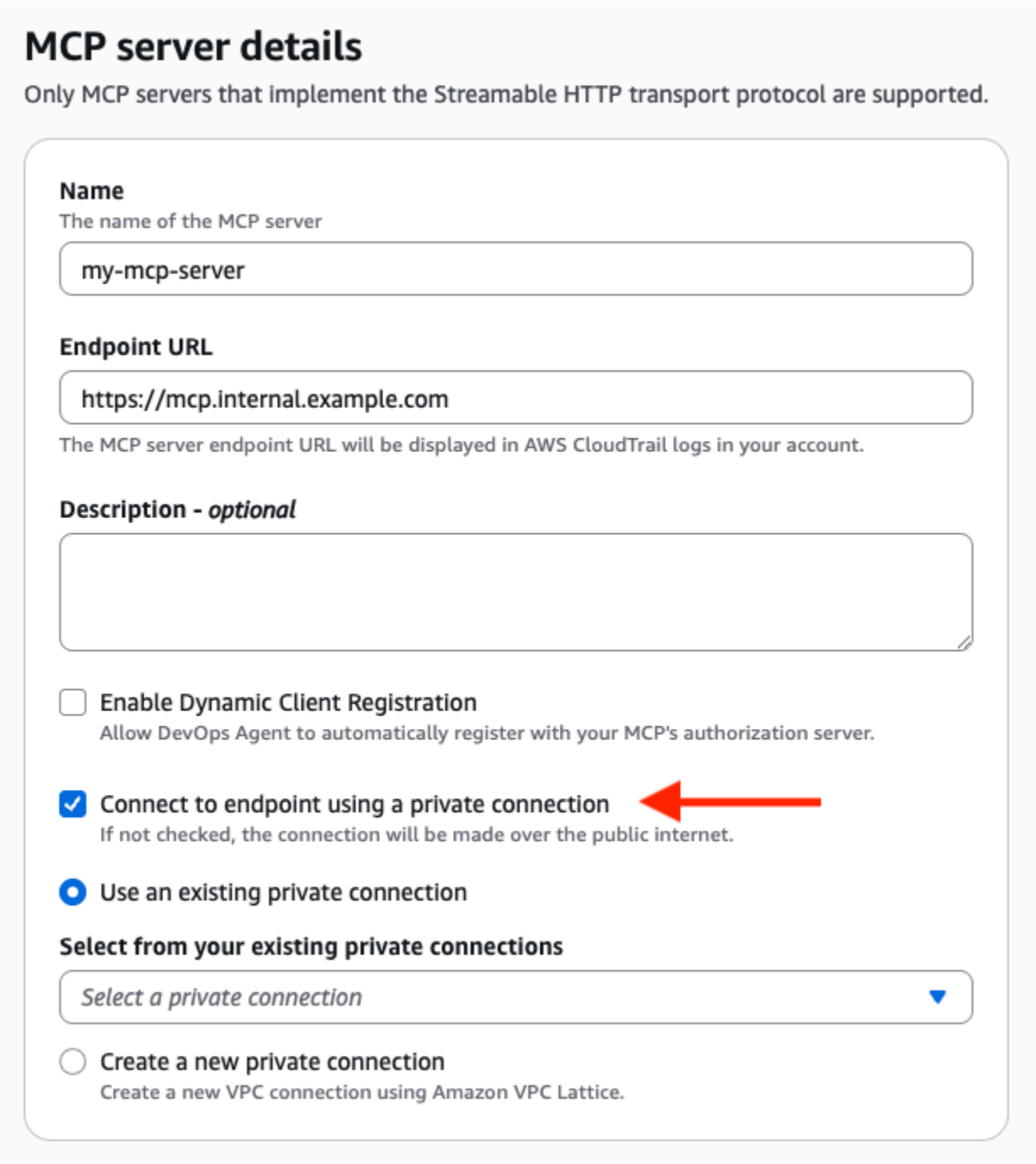
Use una conexión privada con un proveedor de capacidades

Para usar una conexión privada, puede vincularse a ella durante el registro de un proveedor de capacidades. Las capacidades compatibles que se pueden usar con conexiones privadas incluyen: `GitHubGitLab`,`MCP Server`, y`Grafana`. Puede realizar este paso mediante la consola AWS de administración o la AWS CLI.

> **Note**
>
> Al registrar un proveedor de capacidades, el AWS DevOps agente valida que el punto final es accesible y responde. Asegúrese de que el servicio de destino esté funcionando y aceptando conexiones antes de completar el registro.

Utilice una conexión privada con un proveedor de servicios mediante la consola

En la consola del AWS DevOps agente, las conexiones privadas se pueden vincular a una capacidad durante el registro seleccionando la opción «Conectar al punto final mediante una conexión privada».



1. Abre la consola del AWS DevOps agente y navega hasta tu espacio de agente.
2. En la sección Proveedores de capacidades, seleccione Registro.
3. Seleccione Registrar para el tipo de capacidad que desee utilizar con la conexión privada.
4. En la vista de detalles de registro, introduzca la URL del punto final al que desee conectarse mediante la conexión privada (por ejemplo,`https://mcp.internal.example.com`).
5. Seleccione Conectarse al punto final mediante una conexión privada.

6. Seleccione una conexión privada existente que corresponda a la URL del punto final al que desea conectarse o seleccione Crear una nueva conexión privada para crear una.
7. Complete el proceso de registro del proveedor de capacidades.

Utilice una conexión privada con un proveedor de capacidades mediante la AWS CLI

Puede registrar las capacidades con una conexión privada si incluye el `private-connection-name` argumento. A continuación, se muestra un ejemplo de cómo registrar un servidor MCP con la autorización de clave API mediante la conexión `my-mcp-tool-connection` privada. Sustituya los valores de los marcadores de posición por los suyos propios.

```
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 conexión privada

Una vez que la conexión privada alcance el estado activo y la haya utilizado un proveedor de capacidades, compruebe que el AWS DevOps agente pueda acceder al servicio de destino:

1. Abra la consola del AWS DevOps agente y diríjase a su espacio de agente.
2. Inicie una nueva sesión de chat.
3. Invoca un comando que utilice la integración respaldada por tu conexión privada. Por ejemplo, si su herramienta MCP proporciona acceso a una base de conocimientos interna, hágale al agente una pregunta que requiera esa base de conocimientos.

4. Confirme que el agente devuelva los resultados del servicio privado.

Si se produce un error en la conexión, compruebe lo siguiente:

- Límites de VPC Lattice: compruebe que no ha alcanzado ninguna puerta de enlace de recursos u otros límites de cuota de VPC Lattice
- Reglas de los grupos de seguridad: compruebe que los grupos de seguridad adjuntos ENIs permiten el tráfico saliente en el puerto al que atiende su servicio. Compruebe también que el grupo de seguridad de su servicio permita el tráfico entrante en el puerto de destino. El tráfico proviene del plano de datos de VPC Lattice dentro de IPs su rango CIDR de VPC. Puede utilizar la referencia a grupos de seguridad (permitiendo el grupo de seguridad ENI como fuente) o permitir la entrada desde el CIDR de la VPC.
- Conectividad de subred: compruebe que las subredes que ha seleccionado pueden enrutar el tráfico a su servicio. Si el servicio se ejecuta en una subred diferente, confirme que las tablas de enrutamiento permiten el tráfico entre ellas.
- Disponibilidad del servicio: confirme que su servicio se esté ejecutando y aceptando conexiones en el puerto esperado.
- Zona de disponibilidad no compatible: compruebe que las subredes estén en las zonas de disponibilidad compatibles. Ejecute `aws ec2 describe-subnets --subnet-ids <your-subnet-ids> --query 'Subnets[*].[SubnetId,AvailabilityZoneId]'` las zonas de disponibilidad no compatibles enumeradas anteriormente y compruébelas.

Elimine una conexión privada

Puede eliminar las conexiones privadas no utilizadas mediante la consola AWS de administración o la AWS CLI.

Elimine una conexión privada mediante la consola

1. Abre la consola del AWS DevOps agente.
2. En el panel de navegación, elija Proveedores de capacidades y, a continuación, elija Conexiones privadas.
3. Seleccione el menú Acciones de la conexión privada que desee eliminar y, a continuación, seleccione Eliminar.

La conexión privada se mostrará con el estado «Eliminando conexión» mientras el AWS DevOps agente elimina la puerta de enlace de recursos gestionados y la elimina ENIs de su VPC. Una vez completada la eliminación, la conexión dejará de aparecer en la lista de conexiones privadas.

Eliminar una conexión privada mediante la AWS CLI

```
aws devops-agent delete-private-connection \
    --name my-mcp-tool-connection
```

La respuesta devuelve un estado de`DELETE_IN_PROGRESS`. AWS DevOps El agente elimina la puerta de enlace de recursos gestionados y ENIs la elimina de la VPC. Una vez completada la eliminación, la conexión ya no aparece en la lista de conexiones privadas.

Configuración avanzada con los recursos de VPC Lattice existentes

Si su organización ya utiliza Amazon VPC Lattice y administra sus propias configuraciones de recursos, puede crear una conexión privada en modo autogestionado. En lugar de que el AWS DevOps agente cree una puerta de enlace de recursos para usted, usted proporciona el nombre de recurso de Amazon (ARN) de una configuración de recursos existente que apunta a su servicio de destino.

Este enfoque resulta útil cuando:

- Desea tener un control total sobre la pasarela de recursos y el ciclo de vida de la configuración de los recursos.
- Necesita compartir las configuraciones de recursos entre varias AWS cuentas o servicios.
- Requiera los registros de acceso de VPC Lattice para una supervisión detallada del tráfico.
- Ejecute una arquitectura de hub-and-spoke red.

Para crear una conexión privada autogestionada 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"
        }
```

```
    }'
```

Para obtener más información sobre la configuración de las pasarelas de recursos y las configuraciones de recursos de VPC Lattice, consulte la Guía del usuario de Amazon VPC Lattice.

Temas relacionados

- the section called “Puntos de enlace de la VPC (AWS PrivateLink)”
- the section called “Conexión de servidores MCP”
- Configuración de las capacidades del AWS DevOps agente
- AWS DevOps Seguridad del agente
- the section called “DevOps Permisos de IAM para agentes”

AWS DevOps Seguridad del agente

Este documento proporciona información sobre las consideraciones de seguridad, la protección de datos, los controles de acceso y las capacidades de conformidad de AWS DevOps Agent. Utilice esta información para comprender cómo se ha diseñado AWS DevOps Agent para cumplir sus requisitos de seguridad y conformidad.

Seguridad multicapa

AWS DevOps El agente implementa la seguridad en varios niveles. Incluso si se conceden permisos más amplios a la función de IAM del agente, el agente aplica sus propios controles de acceso internos para limitar el alcance de sus acciones. Por ejemplo, si un cliente añade una política completa de IAM de acceso a Amazon S3 a la función de IAM del agente, el AWS DevOps agente se asegurará de que solo se lean los registros posteriores al `AWSLogs` prefijo para solucionar problemas.

Recomendamos seguir el principio de privilegios mínimos al configurar los permisos de IAM para el AWS DevOps agente e implementar la seguridad en varios niveles. Una defensa exhaustiva garantiza que ningún error de configuración pueda comprometer la seguridad de su entorno.

Espacios para agentes

Los espacios de agente sirven como límite de seguridad principal en AWS DevOps Agent. Cada espacio de agente:

- Funciona de forma independiente con sus propias configuraciones y permisos
- Define a qué AWS cuentas y recursos puede acceder el agente
- Establece conexiones con plataformas de terceros

Los Agent Spaces mantienen un aislamiento estricto para garantizar la seguridad y evitar el acceso no deseado a través de diferentes entornos o equipos.

Procesamiento regional y flujo de datos

AWS DevOps El agente opera a nivel mundial con capacidades de procesamiento regionales. El agente recupera los datos operativos de AWS las regiones de todas las AWS cuentas a las que se

ha concedido acceso en el espacio de agente configurado. Esta recopilación de datos entre cuentas multirregionales garantiza un análisis exhaustivo de los incidentes y, al mismo tiempo, respeta los límites geográficos para el procesamiento de las inferencias.

Uso de Amazon Bedrock e inferencia entre regiones

AWS DevOps El agente seleccionará automáticamente la región óptima dentro de su zona geográfica para procesar sus solicitudes de inferencia. Esto maximiza los recursos informáticos disponibles, la disponibilidad del modelo y ofrece la mejor experiencia al cliente. Sus datos permanecerán almacenados únicamente en la región en la que se creó su espacio de agente; sin embargo, es posible que las solicitudes de entrada y los resultados de salida se procesen fuera de esa región, como se describe en la siguiente lista. Todos los datos se transmitirán cifrados a través de la red segura de Amazon.

AWS DevOps El agente dirigirá sus solicitudes de inferencia de forma segura a los recursos informáticos disponibles en el área geográfica en la que se originó la solicitud, de la siguiente manera:

- Las solicitudes de inferencia que se originen en la Unión Europea se procesarán dentro de la Unión Europea.
- Las solicitudes de inferencia que se originen en los Estados Unidos se procesarán en los Estados Unidos.
- Las solicitudes de inferencia que se originen en Australia se procesarán en Australia.
- Las solicitudes de inferencia que se originen en Japón se procesarán en Japón.
- Si una solicitud de inferencia se origina en un área que no figura en la lista, se procesará de forma predeterminada en los Estados Unidos.
- DevOps Agent y Bedrock no se ven afectados por las políticas de cliente de las Políticas de Control de Servicios (SCPs) o de la Torre de Control, que restringen el contenido de los clientes a regiones específicas
- Bedrock puede utilizar regiones distintas de la región de origen dentro de su geografía para realizar inferencias sin estado a fin de optimizar el rendimiento y la disponibilidad

Identity and Access Management

Métodos de autenticación

AWS DevOps El agente proporciona dos métodos de autenticación para iniciar sesión en la aplicación web AWS DevOps Agent Space:

- AWS Integración con Identity Center: el método de autenticación principal utiliza la OAuth versión 2.0 y la autenticación basada en sesiones mediante cookies exclusivas de HTTP. AWS Identity Center puede federarse con proveedores de identidad externos a través de los protocolos OIDC y SAML estándar, incluidos proveedores como Okta, Ping Identity y Microsoft Entra ID. Este método admite la autenticación multifactorial a través de su proveedor de identidad. AWS Identity Center establece de forma predeterminada una duración de sesión de hasta 12 horas y se puede configurar según la duración deseada.
- Enlace de autenticación de IAM: un método alternativo proporciona acceso directo a la aplicación web desde la consola de AWS administración mediante tokens basados en JWT derivados de una sesión de la consola de administración existente. AWS Esta opción es útil para evaluar el AWS DevOps agente antes de implementar la integración completa de Identity Center, así como para obtener acceso administrativo si la aplicación web del AWS DevOps agente deja de ser accesible mediante la autenticación basada en Identity Center. Las sesiones están limitadas a 10 minutos.

Roles de IAM

AWS DevOps El agente usa las funciones de IAM para definir los permisos de acceso:

- Función de cuenta principal: otorga al agente acceso a los recursos de la AWS cuenta en la que se creó el espacio de agentes, así como acceso a las funciones de la cuenta secundaria.
- Funciones de cuenta secundarias: otorga al agente acceso a los recursos de AWS cuentas adicionales conectadas al espacio de agentes.
- Función de aplicación web: otorga a los usuarios acceso a los datos y hallazgos de la investigación del AWS DevOps agente en la aplicación web.

Estas funciones deben configurarse siguiendo el principio de privilegios mínimos y concediendo solo los permisos de solo lectura necesarios para las investigaciones.

Protección de datos

Cifrado de datos

AWS DevOps El agente cifra todos los datos de los clientes:

- Cifrado en reposo: todos los datos se cifran con claves AWS administradas.
- Cifrado en tránsito: todos los registros, métricas, elementos de conocimiento, metadatos de los tickets y otros datos recuperados se cifran en tránsito dentro de la red privada del agente y hacia redes externas.

Almacenamiento y retención de datos

Los datos se almacenan en la región en la que se creó su espacio de agente, mientras que el procesamiento de inferencias puede realizarse dentro de su zona geográfica, tal y como se describe en la sección anterior sobre el uso de Amazon Bedrock.

Información de identificación personal (PII)

AWS DevOps El agente no filtra la información de identificación personal al resumir los datos recopilados durante las investigaciones, las evaluaciones de las recomendaciones o las respuestas al chat. Se recomienda redactar los datos de PII antes de almacenarlos en los registros de observabilidad.

Registro de auditoría y diario del agente

Diario del agente

Tanto las funciones de investigación como de prevención de incidentes mantienen diarios detallados que:

- Registre cada paso de razonamiento y cada acción realizada
- Cree una transparencia total en los procesos de toma de decisiones de los agentes
- Los agentes no pueden modificarlos una vez registrados, lo que minimiza los ataques, como la inyección rápida, al ocultar acciones importantes
- Incluye todos los mensajes de chat de la página de investigación

AWS CloudTrail integración

Todas las llamadas a la API del AWS DevOps agente se capturan automáticamente AWS CloudTrail en la AWS cuenta de alojamiento. Con la información recopilada por CloudTrail, puede determinar:

- La solicitud que se hizo al agente
- La dirección IP desde la que se realizó la solicitud
- Quién realizó la solicitud
- Cuando se realizó

Protección de inyección rápida

Un ataque de inyección rápida se produce cuando un atacante inserta instrucciones maliciosas en datos externos, como una página web o un documento, que posteriormente procesará un sistema de IA generativa. AWS DevOps El agente consume muchas fuentes de datos de forma nativa como parte de sus operaciones normales, incluidos los registros, las etiquetas de recursos y otros datos operativos. AWS DevOps Agent protege contra los ataques de inyección inmediata mediante las siguientes medidas de seguridad, pero es importante garantizar que todas las fuentes de datos conectadas y el acceso de los usuarios a esas fuentes de datos sean confiables. Consulte la sección sobre el modelo de responsabilidad compartida para obtener más información.

Garantías de inyección rápida:

- Capacidades de escritura limitadas: las herramientas de las que dispone el agente no pueden modificar los recursos, con la excepción de abrir tickets y casos de soporte. Esto evita que instrucciones malintencionadas modifiquen la infraestructura o las aplicaciones.
- Control de los límites de la cuenta: el AWS DevOps agente solo opera dentro de los límites permitidos por las funciones asignadas al agente en la cuenta principal y en la AWS cuenta secundaria conectada. El agente no puede acceder a los recursos ni modificarlos fuera del ámbito configurado.
- Protecciones de seguridad mediante IA: el AWS DevOps agente utiliza modelos con protecciones de nivel 3 de seguridad mediante IA (ASL-3). Estas protecciones incluyen clasificadores que detectan y previenen los ataques de inyección inmediata antes de que puedan afectar al comportamiento del agente.

- Registro de auditoría inmutable: el diario del agente registra cada paso de razonamiento y cada acción realizada. El agente no puede modificar las entradas del diario una vez registradas, lo que evita que los ataques de inyección inmediata oculten las acciones maliciosas.

Si bien AWS DevOps Agent ofrece varios niveles de protección contra los ataques de inyección inmediata, algunas configuraciones pueden aumentar el riesgo:

- Herramientas de servidor MCP personalizadas: la función bring-your-own MCP permite introducir herramientas personalizadas en el agente, lo que puede ofrecer oportunidades adicionales para una rápida inyección. Es posible que las herramientas personalizadas no tengan los mismos controles de seguridad que las herramientas de AWS DevOps agente nativas, y las instrucciones malintencionadas podrían aprovechar estas herramientas de forma no deseada. Consulte la sección sobre el modelo de responsabilidad compartida para obtener más información.
- Ataques de usuarios autorizados: los usuarios que están autorizados a operar dentro del límite de la AWS cuenta o de las herramientas conectadas tienen más probabilidades de intentar atacar al agente. Es posible que estos usuarios puedan modificar las fuentes de datos que consume el agente, como registros o etiquetas de recursos, lo que facilita la incrustación de instrucciones maliciosas que el agente procesará.

Para mitigar estos riesgos:

1. Revise y pruebe detenidamente los servidores MCP personalizados antes de implementarlos en Agent Spaces.
   a. Asegúrese de que solo se les permita realizar acciones de solo lectura
   b. Compruebe que los usuarios de las herramientas externas a las que acceden los servidores MCP sean entidades de confianza, ya que AWS DevOps los agentes que interactúan con el MCP se basan en la relación de confianza implícita que se establece entre estos usuarios de la herramienta y el agente AWS DevOps
2. Aplique el principio del privilegio mínimo al conceder a los usuarios el acceso a los sistemas que proporcionan datos al agente
3. Audite periódicamente qué servidores MCP están conectados a sus Agent Spaces
4. Dado que cualquier contenido recuperado de una lista de permitidos URLs podría intentar manipular el comportamiento del agente, incluya únicamente fuentes confiables en su lista de permitidos.

Seguridad de integración

AWS DevOps El agente admite varios tipos de integración, cada uno con su propio modelo de seguridad:

- Integraciones bidireccionales nativas: integraciones integradas que pueden enviar datos al agente y recibir actualizaciones del agente. Utiliza los métodos de autenticación del proveedor
- Servidores MCP: servidores del protocolo de contexto de modelo remoto que utilizan flujos de autenticación OAuth 2.0 y claves de API para comunicarse de forma segura con sistemas externos.
- Activadores de webhook: activadores de investigación desde servicios remotos, como tickets o sistemas de observabilidad. Los webhooks utilizan un código de autenticación de mensajes basado en hash (HMAC) por motivos de seguridad.
- Comunicación saliente: las integraciones como Slack y los sistemas de venta de entradas reciben actualizaciones del agente, pero aún no admiten la comunicación bidireccional.

Proveedores de registro

Algunas herramientas externas se autentican a nivel de cuenta y se comparten entre todos los espacios de agentes de la cuenta. Al registrar estas herramientas, se autentica una vez a nivel de cuenta y, a continuación, cada Agent Space puede conectarse a recursos específicos dentro de esa conexión registrada.

Las siguientes herramientas utilizan el registro a nivel de cuenta:

- GitHub— Utiliza el OAuth flujo para la autenticación. Tras registrarse GitHub a nivel de cuenta, cada espacio de agente puede conectarse a repositorios específicos de su GitHub organización.
- Dynatrace: utiliza OAuth la autenticación mediante token. Tras registrar Dynatrace a nivel de cuenta, cada espacio de agente puede conectarse a entornos o configuraciones de monitoreo específicos de Dynatrace.
- Slack: utiliza la autenticación mediante token. OAuth Tras registrar Slack a nivel de cuenta, cada espacio de agente puede conectarse a canales y canales específicos de Slack.
- Datadog: usa MCP con flow para la autenticación. OAuth Tras registrar Datadog a nivel de cuenta, cada espacio de agente puede conectarse a recursos de monitoreo específicos de Datadog.

- New Relic: utiliza la autenticación mediante clave de API. Tras registrar New Relic a nivel de cuenta, cada espacio de agente puede conectarse a configuraciones de monitoreo específicas de New Relic.
- Splunk: utiliza la autenticación mediante token portador. Tras registrar Splunk a nivel de cuenta, cada espacio de agente puede conectarse a fuentes de datos específicas de Splunk.
- GitLab— Utiliza la autenticación mediante token de acceso. Tras registrarse GitLab a nivel de cuenta, cada espacio de agente puede conectarse a GitLab repositorios específicos.
- ServiceNow— Utiliza la key/token autenticación OAuth del cliente. Tras registrarse ServiceNow a nivel de cuenta, cada espacio de agente puede conectarse a ServiceNow instancias o colas de tickets específicas.
- Servidores MCP remotos accesibles al público en general: utilice el OAuth flujo para la autenticación. Tras registrar un servidor MCP remoto a nivel de cuenta, cada espacio de agente puede conectarse a los recursos específicos expuestos por ese servidor.

Conectividad de red

AWS DevOps El agente se conecta a los sistemas de terceros y a los servidores MCP remotos para realizar investigaciones y otras operaciones.

Tráfico entrante del AWS DevOps agente a sus sistemas

AWS DevOps El agente inicia las conexiones salientes a los sistemas de terceros y a los servidores MCP remotos, que llegan como tráfico entrante a su infraestructura. La forma de proteger este tráfico depende de cómo estén alojadas las herramientas:

- Herramientas alojadas de forma privada: si se puede acceder a sus herramientas desde una AWS VPC, puede AWS DevOps usar las conexiones privadas del agente para mantener el tráfico aislado AWS de las redes y fuera de la Internet pública. Para obtener más información, consulte the section called “Conexión a herramientas alojadas de forma privada”.
- Herramientas alojadas públicamente: si se puede acceder a sus herramientas a través de la Internet pública y utilizan reglas de firewall o listas de IP permitidas, debe permitir el tráfico entrante desde las siguientes AWS DevOps direcciones IP de origen del agente:
  - Asia-Pacífico (Sídney) (ap-southeast-2)
    - `13.237.95.197`
    - `13.238.84.102`

- Asia-Pacífico (Tokio) (ap-northeast-1)
  - `13.192.12.233`
  - `35.74.181.230`
  - `57.183.50.158`
- Europa (Fráncfort) (eu-central-1)
  - `18.158.110.140`
  - `52.57.96.160`
  - `52.59.55.56`
- Europa (Irlanda) (eu-west-1)
  - `34.251.85.24`
  - `52.30.157.157`
  - `52.51.192.222`
- Este de EE. UU. (Norte de Virginia) (us-east-1)
  - `34.228.181.128`
  - `44.219.176.187`
  - `54.226.244.221`
- Oeste de EE. UU. (Oregón) (us-west-2)
  - `34.212.16.133`
  - `52.89.67.212`
  - `54.187.135.61`

Tráfico saliente de su AWS DevOps VPC al agente

Para el tráfico saliente de la AWS VPC AWS DevOps al agente (por ejemplo, the section called “Invocar al DevOps agente a través de Webhook” mediante), puede utilizar los puntos de enlace de la VPC para mantener este tráfico de red aislado de las redes. AWS Para obtener más información, consulte the section called “Puntos de enlace de la VPC (AWS PrivateLink)”.

Modelo de responsabilidad compartida

AWS responsabilidades

AWS es responsable de:

- Mantener la seguridad de los datos recuperados por el agente
- Proteger las herramientas nativas disponibles para que las utilice el agente
- Proteger la infraestructura en la que se ejecuta el AWS DevOps agente

Responsabilidades del cliente

Los clientes son responsables de:

- Administrar el acceso de los usuarios al espacio de agentes
- Limitar el acceso a los usuarios de confianza a los sistemas externos que proporcionan información al agente, como los servicios y recursos que generan registros, CloudTrail eventos, tickets y más, que pueden utilizarse para intentar realizar una inyección rápida maliciosa.
- Asegúrese de que todas las fuentes de datos conectadas tengan datos confiables que probablemente no se utilicen para intentar ataques de inyección rápida
- Garantizar que las integraciones de servidores bring-your-own MCP funcionen de forma segura
- Garantizar que las funciones de IAM asignadas al agente tengan el alcance adecuado
- Redactar los datos de PII antes de almacenarlos en los registros de observabilidad y otras fuentes de datos de los agentes
- Seguir la práctica recomendada de conceder únicamente permisos de solo lectura a las fuentes de datos conectadas, incluidos los servidores MCP bring-your-own

Uso de datos

AWS no utiliza los datos de los agentes, los mensajes de chat ni los datos de fuentes de datos integradas para entrenar modelos o mejorar el producto. El espacio de AWS DevOps agentes utiliza los comentarios de los clientes sobre el producto para mejorar las respuestas y las investigaciones del agente, pero AWS no los utiliza para mejorar el servicio en sí.

Conformidad

En la versión preliminar, AWS DevOps Agent no cumple con estándares como SOC 2, PCI-DSS, ISO 27001 o FedRAMP. AWS anunciará qué certificaciones de conformidad estarán disponibles más adelante.

DevOps Permisos de IAM para agentes

AWS DevOps El agente utiliza acciones de AWS Identity and Access Management (IAM) específicas del servicio para controlar el acceso a sus funciones y capacidades. Estas acciones determinan lo que los usuarios pueden hacer en la consola del AWS DevOps agente y en la aplicación web Operator. Esto es independiente de los permisos de la API de AWS servicio que el propio agente utiliza para investigar sus recursos.

Para obtener más información sobre cómo limitar el acceso de los agentes, consulte Limitar el acceso de los agentes a una AWS cuenta.

Acciones de administración del espacio de agentes

Estas acciones controlan el acceso a la configuración y la administración de Agent Space:

- aidevops: GetAgentSpace — Permite a los usuarios ver los detalles de un Agent Space, incluida su configuración, estado y cuentas asociadas. Los usuarios necesitan este permiso para acceder a un espacio de agentes en la consola de AWS administración.
- aidevops: GetAssociation — Permite a los usuarios ver los detalles sobre una asociación de cuentas específica, incluida la configuración del rol de IAM y el estado de la conexión.
- aidevops: ListAssociations — Permite a los usuarios enumerar todas las asociaciones de AWS cuentas configuradas para un Agent Space, incluidas las cuentas principales y secundarias.

Acciones de investigación y ejecución

Estas acciones controlan el acceso a las funciones de investigación de incidentes:

- aidevops: ListExecutions — Permite a los usuarios ver los metadatos de ejecución, incluidos el ID, el estado y más, para realizar investigaciones, mitigaciones, evaluaciones y conversaciones de chat asociadas a una tarea.
- aidevops: ListJournalRecords — Permite a los usuarios acceder a registros detallados que muestran el razonamiento del agente, las medidas adoptadas y las fuentes de datos consultadas

durante una investigación, una mitigación, una evaluación y una conversación de chat. Esto es útil para entender cómo el agente llegó a sus conclusiones.

Acciones de administración del chat

El chat requiere los siguientes permisos de IAM para funcionar:

- aidevops: ListChats — Permite a los usuarios enumerar el historial de conversaciones de chat y acceder a él.
- aidevops: CreateChat — Permite a los usuarios crear nuevas conversaciones de chat.
- aidevops: SendMessage — Permite a los usuarios enviar consultas y recibir respuestas en streaming.

Acciones de topología y descubrimiento

Estas acciones controlan el acceso a las funciones de mapeo de recursos de la aplicación:

- aidevops: DiscoverTopology — Permite a los usuarios activar el descubrimiento y el mapeo de la topología de un espacio de agentes. Esta acción inicia el proceso de escaneo de AWS cuentas y creación de la topología de recursos de la aplicación.

Acciones de prevención y recomendación

Estas acciones controlan el acceso a la función de prevención:

- aidevops: ListGoals — Permite a los usuarios ver las metas y los objetivos de prevención por los que el agente está trabajando en función de los patrones de incidentes recientes.
- aidevops: ListRecommendations — Permite a los usuarios ver todas las recomendaciones generadas por la función de prevención, incluidas su prioridad y categoría.
- aidevops: GetRecommendation — Permite a los usuarios ver información detallada sobre una recomendación específica, incluidos los incidentes que se habrían evitado y una guía de implementación.

Acciones de gestión de tareas pendientes

Estas acciones controlan la capacidad de gestionar las recomendaciones como tareas pendientes:

- aidevops: CreateBacklogTask — Permite a los usuarios crear una tarea de investigación de incidentes o de evaluación de la prevención.
- aidevops: UpdateBacklogTask — Permite a los usuarios aprobar un plan de mitigación o cancelar una investigación o evaluación en curso.
- aidevops: GetBacklogTask — Permite a los usuarios recuperar detalles sobre una tarea específica.
- aidevops: ListBacklogTasks — Permite a los usuarios enumerar las tareas de un Agent Space, filtrándolas por tipo de tarea, estado, prioridad o hora de creación.

Acciones de gestión del conocimiento

Estas acciones controlan la capacidad de añadir y gestionar conocimientos personalizados que el agente puede utilizar durante las investigaciones:

- aidevops: CreateKnowledgeItem — Permite a los usuarios añadir elementos de conocimiento personalizados, como habilidades, guías de solución de problemas o información específica de la aplicación, a los que el agente debería consultar.
- aidevops: ListKnowledgeItems — Permite a los usuarios ver todos los elementos de conocimiento configurados para un espacio de agente.
- aidevops: GetKnowledgeItem — Permite a los usuarios recuperar los detalles de un elemento de conocimiento específico.
- aidevops: UpdateKnowledgeItem — Permite a los usuarios modificar los elementos de conocimiento existentes para mantener la información actualizada.
- aidevops: DeleteKnowledgeItem — Permite a los usuarios eliminar los elementos de conocimiento que ya no son relevantes.

AWS Support integration actions

Estas acciones controlan la integración con los casos AWS de Support:

- aidevops: InitiateChatForCase — Permite a los usuarios iniciar una sesión de chat con AWS Support directamente desde una investigación, proporcionando automáticamente el contexto del incidente.
- aidevops: EndChatForCase — Permite a los usuarios finalizar una sesión de chat activa sobre un caso de AWS Support.

- aidevops: DescribeSupportLevel — Permite a los usuarios comprobar el nivel del plan de AWS Support de la cuenta para determinar las opciones de soporte disponibles.

Acciones de uso y monitoreo

Estas acciones controlan el acceso a la información de uso:

- aidevops: GetAccountUsage — Permite a los usuarios ver la cuota mensual del AWS DevOps agente en cuanto a horas de investigación, horas de evaluación preventiva y solicitudes de chat, así como el uso del mes en curso.

Ejemplos comunes de políticas de IAM

Política de administrador

Esta política otorga acceso completo a todas las funciones AWS DevOps del agente:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "aidevops:*",
      "Resource": "*"
    }
  ]
}
```

Política del operador

Esta política otorga acceso a las funciones de investigación y prevención sin capacidades administrativas:

```
{
  "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": "*"
    }
  ]
}
```

Política de solo lectura

Esta política otorga acceso de solo lectura a las investigaciones y recomendaciones:

```
{
  "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": "*"
    }
  ]
}
```

Uso de funciones vinculadas al servicio para el agente AWS DevOps

AWS DevOps El agente utiliza AWS funciones vinculadas al servicio Identity and Access Management (IAM). Un rol vinculado a un servicio es un tipo único de rol de IAM que está vinculado directamente al agente. AWS DevOps El AWS DevOps agente predefine las funciones vinculadas al servicio e incluyen todos los permisos que el servicio requiere para llamar a otros AWS servicios en su nombre.

Permisos de roles vinculados a servicios

El rol vinculado al servicio `AWSServiceRoleForAIDevOps` confía en el principal del servicio `aidevops.amazonaws.com` para asumir el rol.

El rol usa la política administrada `AWSServiceRoleForAIDevOpsPolicy` con los siguientes permisos:

- `cloudwatch:PutMetricData`— Publica las métricas de uso en el espacio de `AWS/AIDevOps` CloudWatch nombres. Se rige por una `cloudwatch:namespace` condición que permite solo el espacio de nombres. `AWS/AIDevOps`
- `vpc-lattice:CreateResourceGateway`— Cree pasarelas de recursos de VPC Lattice para conexiones privadas. Establece una `aws:RequestTag/AWSAIDevOpsManaged` condición para que el servicio solo pueda crear pasarelas de recursos que contengan la etiqueta. `AWSAIDevOpsManaged`
- `vpc-lattice:TagResource`— Etiquete las pasarelas de recursos de VPC Lattice. Limitado por una condición. `aws:RequestTag/AWSAIDevOpsManaged`

- `vpc-lattice:DeleteResourceGateway`— Eliminar las pasarelas de recursos de VPC Lattice. Definido por una `aws:ResourceTag/AWSAIDevOpsManaged` condición, por lo que el servicio solo puede eliminar las pasarelas de recursos que haya creado.
- `vpc-lattice:GetResourceGateway`— Recuperar información sobre las pasarelas de recursos de VPC Lattice. Se basa en una `aws:ResourceTag/AWSAIDevOpsManaged` condición para que el servicio solo pueda leer las pasarelas de recursos que haya creado.
- `ec2:DescribeVpcs`,`ec2:DescribeSubnets`, `ec2:DescribeSecurityGroups` — Recupera información sobre los recursos de red de VPC necesarios para configurar las puertas de enlace de recursos. Estas acciones de solo lectura se aplican a todos los recursos de la VPC porque la API de EC2 no admite permisos a nivel de recursos para las llamadas de Describe.
- `iam:CreateServiceLinkedRole`— Cree la función vinculada al servicio VPC Lattice necesaria para las operaciones de la puerta de enlace de recursos. Este permiso está limitado únicamente al director del `vpc-lattice.amazonaws.com` servicio y no se puede utilizar para crear funciones vinculadas al servicio para ningún otro servicio.

Creación del rol vinculado al servicio

No necesita crear manualmente el rol vinculado al servicio `AWSServiceRoleForAIDevOps`. Cuando empiece a utilizar AWS DevOps Agent, el servicio le creará el rol vinculado al servicio.

Para permitir que el servicio cree el rol en su nombre, debe tener el `iam:CreateServiceLinkedRole` permiso. Recomendamos limitar el alcance de este permiso con una `iam:AWSServiceName` condición `aidevops.amazonaws.com` para seguir el principio del privilegio mínimo. Para obtener más información, consulte Permisos de rol vinculado al servicio.

Edición del rol vinculado al servicio

No puede editar el rol vinculado a servicio `AWSServiceRoleForAIDevOps`. Una vez creado el rol, no puede cambiarle el nombre, ya que varias entidades pueden hacer referencia al rol por su nombre. Sin embargo, puede editar la descripción del rol mediante IAM. Para obtener más información, consulte Edición de un rol vinculado a un servicio.

Eliminación del rol vinculado a un servicio

Si ya no necesita usar el AWS DevOps agente, le recomendamos que elimine el rol vinculado al `AWSServiceRoleForAIDevOps` servicio. Antes de poder eliminar el rol, primero debe eliminar todas las conexiones privadas configuradas en su espacio de agente. Al eliminar el rol vinculado

al servicio, no se eliminan automáticamente las puertas de enlace de recursos de VPC Lattice etiquetadas con las `AWSAIDevOpsManaged` que el servicio creó anteriormente. Debe eliminar estas pasarelas de recursos manualmente si ya no las necesita. Para obtener más información, consulte Eliminar un rol vinculado a un servicio.

AWS Políticas administradas para el agente AWS DevOps

AWS aborda muchos casos de uso comunes al proporcionar políticas de IAM independientes que son creadas y administradas por. AWS Estas políticas AWS gestionadas conceden los permisos necesarios para casos de uso comunes, de modo que no tenga que investigar qué permisos son necesarios. Para obtener más información, consulte las políticas AWS administradas en la _Guía del usuario de IAM_.

Las siguientes políticas AWS gestionadas, que puede adjuntar a los usuarios de su cuenta, son específicas del agente. AWS DevOps

AIDevOpsAgentReadOnlyAccess

Proporciona acceso de solo lectura a Amazon DevOps Agent a través de la consola AWS de administración

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "AIDevOpsAgentReadOnlyAccess",
   "Effect": "Allow",
   "Action": [
    "aidevops:Get*",
    "aidevops:List*",
    "aidevops:SearchServiceAccessibleResource"
   ],
   "Resource": "*"
  }
 ]
}
```

AIDevOpsAgentFullAccess

Proporciona acceso completo a Amazon DevOps Agent a través de la consola AWS de administración

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "AIDevOpsAgentSpaceAccess",
   "Effect": "Allow",
   "Action": [
    "aidevops:CreateAgentSpace",
    "aidevops:DeleteAgentSpace",
    "aidevops:GetAgentSpace",
    "aidevops:ListAgentSpaces",
    "aidevops:UpdateAgentSpace"
   ],
   "Resource": "*"
  },
  {
   "Sid": "AIDevOpsServiceAccess",
   "Effect": "Allow",
   "Action": [
    "aidevops:DeregisterService",
    "aidevops:GetService",
    "aidevops:ListServices",
    "aidevops:RegisterService",
    "aidevops:SearchServiceAccessibleResource"
   ],
   "Resource": "*"
  },
  {
   "Sid": "AIDevOpsAssociationAccess",
   "Effect": "Allow",
   "Action": [
    "aidevops:AssociateService",
    "aidevops:DisassociateService",
    "aidevops:GetAssociation",
    "aidevops:ListAssociations",
    "aidevops:UpdateAssociation",
    "aidevops:ValidateAwsAssociations"
   ],
   "Resource": "*"
  },
  {
   "Sid": "AIDevOpsWebhookAccess",
   "Effect": "Allow",
```

```
    "Action": [
     "aidevops:ListWebhooks"
    ],
    "Resource": "*"
   },
   {
    "Sid": "AIDevOpsOperatorAppAccess",
    "Effect": "Allow",
    "Action": [
     "aidevops:DisableOperatorApp",
     "aidevops:EnableOperatorApp",
     "aidevops:GetOperatorApp",
     "aidevops:UpdateOperatorAppIdpConfig"
    ],
    "Resource": "*"
   },
   {
    "Sid": "AIDevOpsKnowledgeAccess",
    "Effect": "Allow",
    "Action": [
     "aidevops:CreateKnowledgeItem",
     "aidevops:DeleteKnowledgeItem",
     "aidevops:GetKnowledgeItem",
     "aidevops:ListKnowledgeItems",
     "aidevops:ListKnowledgeItemVersions",
     "aidevops:UpdateKnowledgeItem"
    ],
    "Resource": "*"
   },
   {
    "Sid": "AIDevOpsBacklogAccess",
    "Effect": "Allow",
    "Action": [
     "aidevops:CreateBacklogTask",
     "aidevops:GetBacklogTask",
     "aidevops:ListBacklogTasks",
     "aidevops:ListGoals",
     "aidevops:UpdateBacklogTask",
     "aidevops:UpdateGoal"
    ],
    "Resource": "*"
   },
   {
    "Sid": "AIDevOpsRecommendationAccess",
```

```
    "Effect": "Allow",
    "Action": [
     "aidevops:GetRecommendation",
     "aidevops:ListRecommendations",
     "aidevops:UpdateRecommendation"
    ],
    "Resource": "*"
   },
   {
    "Sid": "AIDevOpsAgentChatAccess",
    "Effect": "Allow",
    "Action": [
     "aidevops:CreateChat",
     "aidevops:ListChats",
     "aidevops:ListPendingMessages",
     "aidevops:SendMessage"
    ],
    "Resource": "*"
   },
   {
    "Sid": "AIDevOpsJournalAccess",
    "Effect": "Allow",
    "Action": [
     "aidevops:ListExecutions",
     "aidevops:ListJournalRecords"
    ],
    "Resource": "*"
   },
   {
    "Sid": "AIDevOpsTopologyAccess",
    "Effect": "Allow",
    "Action": [
     "aidevops:DiscoverTopology"
    ],
    "Resource": "*"
   },
   {
    "Sid": "AIDevOpsSupportAccess",
    "Effect": "Allow",
    "Action": [
     "aidevops:DescribeSupportLevel",
     "aidevops:EndChatForCase",
     "aidevops:InitiateChatForCase"
    ],
```

```
    "Resource": "*"
   },
   {
    "Sid": "AIDevOpsUsageAccess",
    "Effect": "Allow",
    "Action": [
     "aidevops:GetAccountUsage"
    ],
    "Resource": "*"
   },
   {
    "Sid": "AIDevOpsTaggingAccess",
    "Effect": "Allow",
    "Action": [
     "aidevops:ListTagsForResource",
     "aidevops:TagResource",
     "aidevops:UntagResource"
    ],
    "Resource": "*"
   },
   {
    "Sid": "AIDevOpsVendedLogs",
    "Effect": "Allow",
    "Action": [
     "aidevops:AllowVendedLogDeliveryForResource"
    ],
    "Resource": "*"
   }
  ]
}
```

AIDevOpsOperatorAppAccessPolicy

Proporciona acceso para usar la aplicación web AWS DevOps del operador como espacio de agente.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "AllowOperatorAgentSpaceActions",
   "Effect": "Allow",
   "Action": [
    "aidevops:GetAgentSpace",
```

```
    "aidevops:GetAssociation",
    "aidevops:ListAssociations",
    "aidevops:CreateBacklogTask",
    "aidevops:GetBacklogTask",
    "aidevops:UpdateBacklogTask",
    "aidevops:ListBacklogTasks",
    "aidevops:ListJournalRecords",
    "aidevops:DiscoverTopology",
    "aidevops:ListGoals",
    "aidevops:ListRecommendations",
    "aidevops:ListExecutions",
    "aidevops:GetRecommendation",
    "aidevops:UpdateRecommendation",
    "aidevops:CreateKnowledgeItem",
    "aidevops:ListKnowledgeItems",
    "aidevops:ListKnowledgeItemVersions",
    "aidevops:GetKnowledgeItem",
    "aidevops:UpdateKnowledgeItem",
    "aidevops:DeleteKnowledgeItem",
    "aidevops:ListPendingMessages",
    "aidevops:InitiateChatForCase",
    "aidevops:EndChatForCase",
    "aidevops:DescribeSupportLevel",
    "aidevops:ListChats",
    "aidevops:CreateChat",
    "aidevops:SendMessage"
   ],
   "Resource": "arn:aws:aidevops:*:*:agentspace/${aws:PrincipalTag/AgentSpaceId}",
   "Condition": {
    "StringEquals": {
     "aws:ResourceAccount": "${aws:PrincipalAccount}"
    }
   }
  },
  {
   "Sid": "AllowOperatorAccountActions",
   "Effect": "Allow",
   "Action": [
    "aidevops:GetAccountUsage"
   ],
   "Resource": "*",
   "Condition": {
    "StringEquals": {
     "aws:ResourceAccount": "${aws:PrincipalAccount}"
```

```
    }
   }
  },
  {
   "Sid": "AllowSupportOperatorActions",
   "Effect": "Allow",
   "Action": [
    "support:DescribeCases",
    "support:InitiateChatForCase",
    "support:DescribeSupportLevel"
   ],
   "Resource": "*",
   "Condition": {
    "StringEquals": {
     "aws:ResourceAccount": "${aws:PrincipalAccount}"
    }
   }
  }
 ]
}
```

AIDevOpsAgentAccessPolicy

Proporciona los permisos necesarios para que el AWS DevOps agente lleve a cabo investigaciones y analice AWS los recursos de los clientes.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AIOPSServiceAccess",
            "Effect": "Allow",
            "Action": [
                "access-analyzer:GetAnalyzer",
                "access-analyzer:List*",
                "acm-pca:Describe*",
                "acm-pca:GetCertificate",
                "acm-pca:GetCertificateAuthorityCertificate",
                "acm-pca:GetCertificateAuthorityCsr",
                "acm-pca:List*",
                "acm:DescribeCertificate",
                "acm:GetAccountConfiguration",
                "aidevops:GetKnowledgeItem",
```

```
                "aidevops:ListKnowledgeItems",
                "airflow:List*",
                "amplify:GetApp",
                "amplify:GetBranch",
                "amplify:GetDomainAssociation",
                "amplify:List*",
                "aoss:BatchGetCollection",
                "aoss:BatchGetLifecyclePolicy",
                "aoss:BatchGetVpcEndpoint",
                "aoss:GetAccessPolicy",
                "aoss:GetSecurityConfig",
                "aoss:GetSecurityPolicy",
                "aoss:List*",
                "appconfig:GetApplication",
                "appconfig:GetConfigurationProfile",
                "appconfig:GetEnvironment",
                "appconfig:GetHostedConfigurationVersion",
                "appconfig:List*",
                "appflow:Describe*",
                "appflow:List*",
                "application-autoscaling:Describe*",
                "application-signals:BatchGetServiceLevelObjectiveBudgetReport",
                "application-signals:GetService",
                "application-signals:GetServiceLevelObjective",
                "application-signals:List*",
                "applicationinsights:Describe*",
                "applicationinsights:List*",
                "apprunner:Describe*",
                "apprunner:List*",
                "appstream:Describe*",
                "appstream:List*",
                "appsync:GetApiAssociation",
                "appsync:GetDataSource",
                "appsync:GetDomainName",
                "appsync:GetFunction",
                "appsync:GetGraphqlApi",
                "appsync:GetGraphqlApiEnvironmentVariables",
                "appsync:GetIntrospectionSchema",
                "appsync:GetResolver",
                "appsync:GetSourceApiAssociation",
                "appsync:List*",
                "aps:Describe*",
                "aps:List*",
                "arc-zonal-shift:GetManagedResource",
```

```
                "arc-zonal-shift:List*",
                "athena:GetCapacityAssignmentConfiguration",
                "athena:GetCapacityReservation",
                "athena:GetDataCatalog",
                "athena:GetNamedQuery",
                "athena:GetPreparedStatement",
                "athena:GetWorkGroup",
                "athena:List*",
                "auditmanager:GetAssessment",
                "auditmanager:List*",
                "autoscaling:Describe*",
                "backup-gateway:GetHypervisor",
                "backup-gateway:List*",
                "backup:Describe*",
                "backup:GetBackupPlan",
                "backup:GetBackupSelection",
                "backup:GetBackupVaultAccessPolicy",
                "backup:GetBackupVaultNotifications",
                "backup:GetRestoreTestingPlan",
                "backup:GetRestoreTestingSelection",
                "backup:List*",
                "batch:DescribeComputeEnvironments",
                "batch:DescribeJobQueues",
                "batch:DescribeSchedulingPolicies",
                "batch:List*",
                "bedrock:GetAgent",
                "bedrock:GetAgentActionGroup",
                "bedrock:GetAgentAlias",
                "bedrock:GetAgentKnowledgeBase",
                "bedrock:GetDataSource",
                "bedrock:GetGuardrail",
                "bedrock:GetKnowledgeBase",
                "bedrock:List*",
                "budgets:Describe*",
                "budgets:List*",
                "ce:Describe*",
                "ce:GetAnomalyMonitors",
                "ce:GetAnomalySubscriptions",
                "ce:List*",
                "chatbot:Describe*",
                "chatbot:GetMicrosoftTeamsChannelConfiguration",
                "chatbot:List*",
                "cleanrooms-ml:GetTrainingDataset",
                "cleanrooms-ml:List*",
```

```
                "cleanrooms:GetAnalysisTemplate",
                "cleanrooms:GetCollaboration",
                "cleanrooms:GetConfiguredTable",
                "cleanrooms:GetConfiguredTableAnalysisRule",
                "cleanrooms:GetConfiguredTableAssociation",
                "cleanrooms:GetMembership",
                "cleanrooms:List*",
                "cloudformation:Describe*",
                "cloudformation:GetResource",
                "cloudformation:GetStackPolicy",
                "cloudformation:GetTemplate",
                "cloudformation:List*",
                "cloudfront:Describe*",
                "cloudfront:GetCachePolicy",
                "cloudfront:GetCloudFrontOriginAccessIdentity",
                "cloudfront:GetContinuousDeploymentPolicy",
                "cloudfront:GetDistribution",
                "cloudfront:GetDistributionConfig",
                "cloudfront:GetFunction",
                "cloudfront:GetKeyGroup",
                "cloudfront:GetMonitoringSubscription",
                "cloudfront:GetOriginAccessControl",
                "cloudfront:GetOriginRequestPolicy",
                "cloudfront:GetPublicKey",
                "cloudfront:GetRealtimeLogConfig",
                "cloudfront:GetResponseHeadersPolicy",
                "cloudfront:List*",
                "cloudtrail:Describe*",
                "cloudtrail:GetChannel",
                "cloudtrail:GetEventConfiguration",
                "cloudtrail:GetEventDataStore",
                "cloudtrail:GetEventSelectors",
                "cloudtrail:GetInsightSelectors",
                "cloudtrail:GetQueryResults",
                "cloudtrail:GetResourcePolicy",
                "cloudtrail:GetTrail",
                "cloudtrail:GetTrailStatus",
                "cloudtrail:List*",
                "cloudtrail:LookupEvents",
                "cloudtrail:StartQuery",
                "cloudwatch:Describe*",
                "cloudwatch:GenerateQuery",
                "cloudwatch:GetDashboard",
                "cloudwatch:GetInsightRuleReport",
```

```
                "cloudwatch:GetMetricData",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:GetMetricStream",
                "cloudwatch:GetService",
                "cloudwatch:GetServiceLevelObjective",
                "cloudwatch:List*",
                "codeartifact:Describe*",
                "codeartifact:GetDomainPermissionsPolicy",
                "codeartifact:GetRepositoryPermissionsPolicy",
                "codeartifact:List*",
                "codebuild:BatchGetFleets",
                "codebuild:List*",
                "codecommit:GetRepository",
                "codecommit:GetRepositoryTriggers",
                "codedeploy:BatchGetDeployments",
                "codedeploy:BatchGetDeploymentTargets",
                "codedeploy:GetApplication",
                "codedeploy:GetDeploymentConfig",
                "codedeploy:GetDeploymentTarget",
                "codedeploy:List*",
                "codeguru-profiler:Describe*",
                "codeguru-profiler:GetNotificationConfiguration",
                "codeguru-profiler:GetPolicy",
                "codeguru-profiler:List*",
                "codeguru-reviewer:Describe*",
                "codeguru-reviewer:List*",
                "codepipeline:GetPipeline",
                "codepipeline:GetPipelineState",
                "codepipeline:List*",
                "codestar-connections:GetConnection",
                "codestar-connections:GetRepositoryLink",
                "codestar-connections:GetSyncConfiguration",
                "codestar-connections:List*",
                "codestar-notifications:Describe*",
                "codestar-notifications:List*",
                "cognito-identity:DescribeIdentityPool",
                "cognito-identity:GetIdentityPoolRoles",
                "cognito-identity:ListIdentityPools",
                "cognito-identity:ListTagsForResource",
                "cognito-idp:AdminListGroupsForUser",
                "cognito-idp:DescribeIdentityProvider",
                "cognito-idp:DescribeResourceServer",
                "cognito-idp:DescribeRiskConfiguration",
                "cognito-idp:DescribeUserImportJob",
```

```
                "cognito-idp:DescribeUserPool",
                "cognito-idp:DescribeUserPoolDomain",
                "cognito-idp:GetGroup",
                "cognito-idp:GetLogDeliveryConfiguration",
                "cognito-idp:GetUICustomization",
                "cognito-idp:GetUserPoolMfaConfig",
                "cognito-idp:GetWebACLForResource",
                "cognito-idp:ListGroups",
                "cognito-idp:ListIdentityProviders",
                "cognito-idp:ListResourceServers",
                "cognito-idp:ListUserPoolClients",
                "cognito-idp:ListUserPools",
                "cognito-idp:ListTagsForResource",
                "comprehend:Describe*",
                "comprehend:List*",
                "config:Describe*",
                "config:GetStoredQuery",
                "config:List*",
                "connect:Describe*",
                "connect:GetTaskTemplate",
                "connect:List*",
                "databrew:Describe*",
                "databrew:List*",
                "datapipeline:Describe*",
                "datapipeline:GetPipelineDefinition",
                "datapipeline:List*",
                "datasync:Describe*",
                "datasync:List*",
                "deadline:GetFarm",
                "deadline:GetFleet",
                "deadline:GetLicenseEndpoint",
                "deadline:GetMonitor",
                "deadline:GetQueue",
                "deadline:GetQueueEnvironment",
                "deadline:GetQueueFleetAssociation",
                "deadline:GetStorageProfile",
                "deadline:List*",
                "detective:GetMembers",
                "detective:List*",
                "devicefarm:GetDevicePool",
                "devicefarm:GetInstanceProfile",
                "devicefarm:GetNetworkProfile",
                "devicefarm:GetProject",
                "devicefarm:GetTestGridProject",
```

```
                "devicefarm:GetVPCEConfiguration",
                "devicefarm:List*",
                "devops-guru:Describe*",
                "devops-guru:GetResourceCollection",
                "devops-guru:List*",
                "dms:Describe*",
                "dms:List*",
                "ds:Describe*",
                "dynamodb:Describe*",
                "dynamodb:GetResourcePolicy",
                "dynamodb:List*",
                "ec2:Describe*",
                "ec2:GetAssociatedEnclaveCertificateIamRoles",
                "ec2:GetIpamPoolAllocations",
                "ec2:GetIpamPoolCidrs",
                "ec2:GetManagedPrefixListEntries",
                "ec2:GetNetworkInsightsAccessScopeContent",
                "ec2:GetSnapshotBlockPublicAccessState",
                "ec2:GetTransitGatewayMulticastDomainAssociations",
                "ec2:GetTransitGatewayRouteTableAssociations",
                "ec2:GetTransitGatewayRouteTablePropagations",
                "ec2:GetVerifiedAccessEndpointPolicy",
                "ec2:GetVerifiedAccessGroupPolicy",
                "ec2:GetVerifiedAccessInstanceWebAcl",
                "ec2:SearchLocalGatewayRoutes",
                "ec2:SearchTransitGatewayRoutes",
                "ecr:Describe*",
                "ecr:GetLifecyclePolicy",
                "ecr:GetRegistryPolicy",
                "ecr:GetRepositoryPolicy",
                "ecr:List*",
                "ecs:Describe*",
                "ecs:List*",
                "eks:AccessKubernetesApi",
                "eks:Describe*",
                "eks:List*",
                "elasticache:Describe*",
                "elasticache:List*",
                "elasticbeanstalk:Describe*",
                "elasticbeanstalk:List*",
                "elasticfilesystem:Describe*",
                "elasticloadbalancing:GetResourcePolicy",
                "elasticloadbalancing:GetTrustStoreCaCertificatesBundle",
                "elasticloadbalancing:GetTrustStoreRevocationContent",
```

```
                "elasticloadbalancing:Describe*",
                "elasticmapreduce:Describe*",
                "elasticmapreduce:List*",
                "emr-containers:Describe*",
                "emr-containers:List*",
                "emr-serverless:GetApplication",
                "emr-serverless:List*",
                "es:Describe*",
                "es:List*",
                "events:Describe*",
                "events:List*",
                "evidently:GetExperiment",
                "evidently:GetFeature",
                "evidently:GetLaunch",
                "evidently:GetProject",
                "evidently:GetSegment",
                "evidently:List*",
                "firehose:Describe*",
                "firehose:List*",
                "fis:GetExperimentTemplate",
                "fis:GetTargetAccountConfiguration",
                "fis:List*",
                "fms:GetNotificationChannel",
                "fms:GetPolicy",
                "fms:List*",
                "forecast:Describe*",
                "forecast:List*",
                "frauddetector:BatchGetVariable",
                "frauddetector:Describe*",
                "frauddetector:GetDetectors",
                "frauddetector:GetDetectorVersion",
                "frauddetector:GetEntityTypes",
                "frauddetector:GetEventTypes",
                "frauddetector:GetExternalModels",
                "frauddetector:GetLabels",
                "frauddetector:GetListElements",
                "frauddetector:GetListsMetadata",
                "frauddetector:GetModelVersion",
                "frauddetector:GetOutcomes",
                "frauddetector:GetRules",
                "frauddetector:GetVariables",
                "frauddetector:List*",
                "fsx:Describe*",
                "gamelift:Describe*",
```

```
                "gamelift:List*",
                "globalaccelerator:Describe*",
                "globalaccelerator:List*",
                "glue:GetDatabase",
                "glue:GetDatabases",
                "glue:GetJob",
                "glue:GetRegistry",
                "glue:GetSchema",
                "glue:GetSchemaVersion",
                "glue:GetTable",
                "glue:GetTags",
                "glue:GetTrigger",
                "glue:List*",
                "glue:querySchemaVersionMetadata",
                "grafana:Describe*",
                "grafana:List*",
                "greengrass:Describe*",
                "greengrass:GetDeployment",
                "greengrass:List*",
                "groundstation:GetConfig",
                "groundstation:GetDataflowEndpointGroup",
                "groundstation:GetMissionProfile",
                "groundstation:List*",
                "guardduty:GetDetector",
                "guardduty:GetFilter",
                "guardduty:GetIPSet",
                "guardduty:GetMalwareProtectionPlan",
                "guardduty:GetMasterAccount",
                "guardduty:GetMembers",
                "guardduty:GetThreatIntelSet",
                "guardduty:List*",
                "health:DescribeEvents",
                "health:DescribeEventDetails",
                "healthlake:Describe*",
                "healthlake:List*",
                "iam:GetGroup",
                "iam:GetGroupPolicy",
                "iam:GetInstanceProfile",
                "iam:GetLoginProfile",
                "iam:GetOpenIDConnectProvider",
                "iam:GetPolicy",
                "iam:GetPolicyVersion",
                "iam:GetRole",
                "iam:GetRolePolicy",
```

```
                "iam:GetSAMLProvider",
                "iam:GetServerCertificate",
                "iam:GetServiceLinkedRoleDeletionStatus",
                "iam:GetUser",
                "iam:GetUserPolicy",
                "iam:ListAttachedRolePolicies",
                "iam:ListOpenIDConnectProviders",
                "iam:ListRolePolicies",
                "iam:ListRoles",
                "iam:ListServerCertificates",
                "iam:ListVirtualMFADevices",
                "identitystore:DescribeGroup",
                "identitystore:DescribeGroupMembership",
                "identitystore:ListGroupMemberships",
                "identitystore:ListGroups",
                "imagebuilder:GetComponent",
                "imagebuilder:GetContainerRecipe",
                "imagebuilder:GetDistributionConfiguration",
                "imagebuilder:GetImage",
                "imagebuilder:GetImagePipeline",
                "imagebuilder:GetImageRecipe",
                "imagebuilder:GetInfrastructureConfiguration",
                "imagebuilder:GetLifecyclePolicy",
                "imagebuilder:GetWorkflow",
                "imagebuilder:List*",
                "inspector2:List*",
                "inspector:Describe*",
                "inspector:List*",
                "internetmonitor:GetMonitor",
                "internetmonitor:List*",
                "iot:Describe*",
                "iot:GetPackage",
                "iot:GetPackageVersion",
                "iot:GetPolicy",
                "iot:GetThingShadow",
                "iot:GetTopicRule",
                "iot:GetTopicRuleDestination",
                "iot:GetV2LoggingOptions",
                "iot:List*",
                "iotanalytics:Describe*",
                "iotanalytics:List*",
                "iotevents:Describe*",
                "iotevents:List*",
                "iotsitewise:Describe*",
```

```
                "iotsitewise:List*",
                "iotwireless:GetDestination",
                "iotwireless:GetDeviceProfile",
                "iotwireless:GetFuotaTask",
                "iotwireless:GetMulticastGroup",
                "iotwireless:GetNetworkAnalyzerConfiguration",
                "iotwireless:GetServiceProfile",
                "iotwireless:GetWirelessDevice",
                "iotwireless:GetWirelessGateway",
                "iotwireless:GetWirelessGatewayTaskDefinition",
                "iotwireless:List*",
                "ivs:GetChannel",
                "ivs:GetEncoderConfiguration",
                "ivs:GetPlaybackRestrictionPolicy",
                "ivs:GetRecordingConfiguration",
                "ivs:GetStage",
                "ivs:List*",
                "ivschat:GetLoggingConfiguration",
                "ivschat:GetRoom",
                "ivschat:List*",
                "kafka:Describe*",
                "kafka:GetClusterPolicy",
                "kafka:List*",
                "kafkaconnect:Describe*",
                "kafkaconnect:List*",
                "kendra:Describe*",
                "kendra:List*",
                "kinesis:Describe*",
                "kinesis:GetResourcePolicy",
                "kinesis:List*",
                "kinesisanalytics:Describe*",
                "kinesisanalytics:List*",
                "kinesisvideo:Describe*",
                "kms:DescribeKey",
                "kms:ListResourceTags",
                "kms:ListKeys",
                "kms:GetKeyPolicy",
                "kms:GetKeyRotationStatus",
                "kms:ListAliases",
                "kms:ListKeyRotations",
                "lakeformation:Describe*",
                "lakeformation:GetLFTag",
                "lakeformation:GetResourceLFTags",
                "lakeformation:List*",
```

```
                "lambda:GetAlias",
                "lambda:GetCodeSigningConfig",
                "lambda:GetEventSourceMapping",
                "lambda:GetFunctionCodeSigningConfig",
                "lambda:GetFunctionConfiguration",
                "lambda:GetFunctionEventInvokeConfig",
                "lambda:GetFunctionRecursionConfig",
                "lambda:GetFunctionUrlConfig",
                "lambda:GetLayerVersion",
                "lambda:GetLayerVersionPolicy",
                "lambda:GetPolicy",
                "lambda:GetProvisionedConcurrencyConfig",
                "lambda:GetRuntimeManagementConfig",
                "lambda:List*",
                "launchwizard:GetDeployment",
                "launchwizard:List*",
                "license-manager:GetLicense",
                "license-manager:List*",
                "lightsail:GetAlarms",
                "lightsail:GetBuckets",
                "lightsail:GetCertificates",
                "lightsail:GetContainerServices",
                "lightsail:GetDisk",
                "lightsail:GetDisks",
                "lightsail:GetInstance",
                "lightsail:GetInstances",
                "lightsail:GetLoadBalancer",
                "lightsail:GetLoadBalancers",
                "lightsail:GetLoadBalancerTlsCertificates",
                "lightsail:GetStaticIp",
                "lightsail:GetStaticIps",
                "logs:Describe*",
                "logs:FilterLogEvents",
                "logs:GetDataProtectionPolicy",
                "logs:GetDelivery",
                "logs:GetDeliveryDestination",
                "logs:GetDeliveryDestinationPolicy",
                "logs:GetDeliverySource",
                "logs:GetLogAnomalyDetector",
                "logs:GetLogDelivery",
                "logs:GetLogGroupFields",
                "logs:GetQueryResults",
                "logs:List*",
                "logs:StartQuery",
```

```
                "logs:StopLiveTail",
                "logs:StopQuery",
                "logs:TestMetricFilter",
                "m2:GetApplication",
                "m2:GetEnvironment",
                "m2:List*",
                "macie2:GetAllowList",
                "macie2:GetCustomDataIdentifier",
                "macie2:GetFindingsFilter",
                "macie2:GetMacieSession",
                "macie2:List*",
                "mediaconnect:Describe*",
                "mediaconnect:List*",
                "medialive:Describe*",
                "medialive:GetCloudWatchAlarmTemplate",
                "medialive:GetCloudWatchAlarmTemplateGroup",
                "medialive:GetEventBridgeRuleTemplate",
                "medialive:GetEventBridgeRuleTemplateGroup",
                "medialive:GetSignalMap",
                "medialive:List*",
                "mediapackage-vod:Describe*",
                "mediapackage-vod:List*",
                "mediapackage:Describe*",
                "mediapackage:List*",
                "mediapackagev2:GetChannel",
                "mediapackagev2:GetChannelGroup",
                "mediapackagev2:GetChannelPolicy",
                "mediapackagev2:GetOriginEndpoint",
                "mediapackagev2:GetOriginEndpointPolicy",
                "mediapackagev2:List*",
                "memorydb:Describe*",
                "memorydb:List*",
                "mobiletargeting:GetInAppTemplate",
                "mobiletargeting:List*",
                "mq:Describe*",
                "mq:List*",
                "network-firewall:Describe*",
                "network-firewall:List*",
                "networkmanager:Describe*",
                "networkmanager:GetConnectAttachment",
                "networkmanager:GetConnectPeer",
                "networkmanager:GetCoreNetwork",
                "networkmanager:GetCoreNetworkPolicy",
                "networkmanager:GetCustomerGatewayAssociations",
```

```
                "networkmanager:GetDevices",
                "networkmanager:GetLinkAssociations",
                "networkmanager:GetLinks",
                "networkmanager:GetSites",
                "networkmanager:GetSiteToSiteVpnAttachment",
                "networkmanager:GetTransitGatewayPeering",
                "networkmanager:GetTransitGatewayRegistrations",
                "networkmanager:GetTransitGatewayRouteTableAttachment",
                "networkmanager:GetVpcAttachment",
                "networkmanager:List*",
                "oam:GetLink",
                "oam:GetSink",
                "oam:GetSinkPolicy",
                "oam:List*",
                "omics:GetAnnotationStore",
                "omics:GetReferenceStore",
                "omics:GetRunGroup",
                "omics:GetSequenceStore",
                "omics:GetVariantStore",
                "omics:GetWorkflow",
                "omics:List*",
                "organizations:Describe*",
                "organizations:List*",
                "osis:GetPipeline",
                "osis:List*",
                "payment-cryptography:GetAlias",
                "payment-cryptography:GetKey",
                "payment-cryptography:List*",
                "pca-connector-ad:GetConnector",
                "pca-connector-ad:GetDirectoryRegistration",
                "pca-connector-ad:GetServicePrincipalName",
                "pca-connector-ad:GetTemplate",
                "pca-connector-ad:GetTemplateGroupAccessControlEntry",
                "pca-connector-ad:List*",
                "pca-connector-scep:GetChallengeMetadata",
                "pca-connector-scep:GetConnector",
                "pca-connector-scep:List*",
                "personalize:Describe*",
                "personalize:List*",
                "pi:DescribeDimensionKeys",
                "pi:GetResourceMetadata",
                "pi:GetResourceMetrics",
                "pi:ListAvailableResourceDimensions",
                "pi:ListAvailableResourceMetrics",
```

```
                "pipes:Describe*",
                "pipes:List*",
                "proton:GetEnvironmentTemplate",
                "proton:GetServiceTemplate",
                "proton:List*",
                "qbusiness:GetApplication",
                "qbusiness:GetDataSource",
                "qbusiness:GetIndex",
                "qbusiness:GetPlugin",
                "qbusiness:GetRetriever",
                "qbusiness:GetWebExperience",
                "qbusiness:List*",
                "ram:GetPermission",
                "ram:GetResourceShares",
                "ram:List*",
                "rds:Describe*",
                "rds:List*",
                "redshift-serverless:GetNamespace",
                "redshift-serverless:GetWorkgroup",
                "redshift-serverless:List*",
                "redshift:Describe*",
                "refactor-spaces:GetApplication",
                "refactor-spaces:GetEnvironment",
                "refactor-spaces:GetRoute",
                "refactor-spaces:List*",
                "rekognition:Describe*",
                "rekognition:List*",
                "resiliencehub:Describe*",
                "resiliencehub:List*",
                "resource-explorer-2:GetDefaultView",
                "resource-explorer-2:GetIndex",
                "resource-explorer-2:GetView",
                "resource-explorer-2:List*",
                "resource-explorer-2:Search",
                "resource-groups:GetGroup",
                "resource-groups:GetGroupConfiguration",
                "resource-groups:GetGroupQuery",
                "resource-groups:GetTags",
                "resource-groups:List*",
                "route53-recovery-control-config:Describe*",
                "route53-recovery-control-config:List*",
                "route53-recovery-readiness:GetCell",
                "route53-recovery-readiness:GetReadinessCheck",
                "route53-recovery-readiness:GetRecoveryGroup",
```

```
                "route53-recovery-readiness:GetResourceSet",
                "route53-recovery-readiness:List*",
                "route53:GetDNSSEC",
                "route53:GetHealthCheck",
                "route53:GetHealthCheckStatus",
                "route53:GetHostedZone",
                "route53:List*",
                "route53profiles:GetProfile",
                "route53profiles:GetProfileAssociation",
                "route53profiles:GetProfileResourceAssociation",
                "route53profiles:List*",
                "route53resolver:GetFirewallDomainList",
                "route53resolver:GetFirewallRuleGroup",
                "route53resolver:GetFirewallRuleGroupAssociation",
                "route53resolver:GetOutpostResolver",
                "route53resolver:GetResolverConfig",
                "route53resolver:GetResolverQueryLogConfig",
                "route53resolver:GetResolverQueryLogConfigAssociation",
                "route53resolver:GetResolverRule",
                "route53resolver:GetResolverRuleAssociation",
                "route53resolver:List*",
                "rum:GetAppMonitor",
                "rum:List*",
                "s3-outposts:ListEndpoints",
                "s3-outposts:ListOutpostsWithS3",
                "s3:GetAccessGrant",
                "s3:GetAccessGrantsInstance",
                "s3:GetAccessGrantsLocation",
                "s3:GetAccessPoint",
                "s3:GetAccessPointConfigurationForObjectLambda",
                "s3:GetAccessPointForObjectLambda",
                "s3:GetAccessPointPolicy",
                "s3:GetAccessPointPolicyForObjectLambda",
                "s3:GetAccessPointPolicyStatusForObjectLambda",
                "s3:GetBucketAbac",
                "s3:GetBucketAcl",
                "s3:GetBucketCORS",
                "s3:GetBucketLocation",
                "s3:GetBucketLogging",
                "s3:GetBucketMetadataTableConfiguration",
                "s3:GetBucketNotification",
                "s3:GetBucketObjectLockConfiguration",
                "s3:GetBucketOwnershipControls",
                "s3:GetBucketPolicy",
```

```
                "s3:GetBucketPublicAccessBlock",
                "s3:GetBucketTagging",
                "s3:GetBucketVersioning",
                "s3:GetEncryptionConfiguration",
                "s3:GetLifecycleConfiguration",
                "s3:GetMultiRegionAccessPoint",
                "s3:GetMultiRegionAccessPointPolicy",
                "s3:GetMultiRegionAccessPointPolicyStatus",
                "s3:GetReplicationConfiguration",
                "s3:GetStorageLensConfiguration",
                "s3:GetStorageLensConfigurationTagging",
                "s3:GetStorageLensGroup",
                "s3:ListAllMyBuckets",
                "sagemaker:Describe*",
                "sagemaker:List*",
                "scheduler:GetSchedule",
                "scheduler:GetScheduleGroup",
                "scheduler:List*",
                "schemas:Describe*",
                "schemas:GetResourcePolicy",
                "schemas:List*",
                "secretsmanager:Describe*",
                "secretsmanager:GetResourcePolicy",
                "secretsmanager:List*",
                "securityhub:BatchGetAutomationRules",
                "securityhub:BatchGetSecurityControls",
                "securityhub:Describe*",
                "securityhub:GetConfigurationPolicy",
                "securityhub:GetConfigurationPolicyAssociation",
                "securityhub:GetEnabledStandards",
                "securityhub:GetFindingAggregator",
                "securityhub:GetInsights",
                "securityhub:List*",
                "securitylake:GetSubscriber",
                "securitylake:List*",
                "servicecatalog:Describe*",
                "servicecatalog:GetApplication",
                "servicecatalog:GetAttributeGroup",
                "servicecatalog:List*",
                "servicequotas:GetServiceQuota",
                "ses:Describe*",
                "ses:GetAccount",
                "ses:GetAddonInstance",
                "ses:GetAddonSubscription",
```

```
                "ses:GetArchive",
                "ses:GetConfigurationSet",
                "ses:GetConfigurationSetEventDestinations",
                "ses:GetContactList",
                "ses:GetDedicatedIpPool",
                "ses:GetDedicatedIps",
                "ses:GetEmailIdentity",
                "ses:GetEmailTemplate",
                "ses:GetIngressPoint",
                "ses:GetRelay",
                "ses:GetRuleSet",
                "ses:GetTemplate",
                "ses:GetTrafficPolicy",
                "ses:List*",
                "shield:Describe*",
                "shield:List*",
                "signer:GetSigningProfile",
                "signer:List*",
                "sns:GetDataProtectionPolicy",
                "sns:GetSubscriptionAttributes",
                "sns:GetTopicAttributes",
                "sns:List*",
                "sqs:GetQueueAttributes",
                "sqs:GetQueueUrl",
                "sqs:List*",
                "ssm-contacts:GetContact",
                "ssm-contacts:GetContactChannel",
                "ssm-contacts:List*",
                "ssm-incidents:GetReplicationSet",
                "ssm-incidents:GetResponsePlan",
                "ssm-incidents:List*",
                "ssm-sap:GetApplication",
                "ssm-sap:List*",
                "ssm:Describe*",
                "ssm:GetDefaultPatchBaseline",
                "ssm:GetDocument",
                "ssm:GetParameters",
                "ssm:GetPatchBaseline",
                "ssm:GetResourcePolicies",
                "ssm:List*",
                "sso:GetInlinePolicyForPermissionSet",
                "sso:GetManagedApplicationInstance",
                "sso:GetPermissionsBoundaryForPermissionSet",
                "sso:GetSharedSsoConfiguration",
```

```
                "sso:ListAccountAssignments",
                "sso:ListApplicationAssignments",
                "sso:ListApplications",
                "sso:ListCustomerManagedPolicyReferencesInPermissionSet",
                "sso:ListInstances",
                "sso:ListManagedPoliciesInPermissionSet",
                "sso:ListTagsForResource",
                "states:GetExecutionHistory",
                "states:Describe*",
                "states:List*",
                "support:CreateCase",
                "support:DescribeCases",
                "synthetics:Describe*",
                "synthetics:GetCanary",
                "synthetics:GetCanaryRuns",
                "synthetics:GetGroup",
                "synthetics:List*",
                "tag:GetResources",
                "timestream:Describe*",
                "timestream:List*",
                "transfer:Describe*",
                "transfer:List*",
                "verifiedpermissions:GetIdentitySource",
                "verifiedpermissions:GetPolicy",
                "verifiedpermissions:GetPolicyStore",
                "verifiedpermissions:GetPolicyTemplate",
                "verifiedpermissions:GetSchema",
                "verifiedpermissions:List*",
                "vpc-lattice:GetAccessLogSubscription",
                "vpc-lattice:GetAuthPolicy",
                "vpc-lattice:GetListener",
                "vpc-lattice:GetResourcePolicy",
                "vpc-lattice:GetRule",
                "vpc-lattice:GetService",
                "vpc-lattice:GetServiceNetwork",
                "vpc-lattice:GetServiceNetworkServiceAssociation",
                "vpc-lattice:GetServiceNetworkVpcAssociation",
                "vpc-lattice:GetTargetGroup",
                "vpc-lattice:List*",
                "wafv2:GetIPSet",
                "wafv2:GetLoggingConfiguration",
                "wafv2:GetRegexPatternSet",
                "wafv2:GetRuleGroup",
                "wafv2:GetWebACL",
```

```
                "wafv2:GetWebACLForResource",
                "wafv2:List*",
                "workspaces-web:GetBrowserSettings",
                "workspaces-web:GetIdentityProvider",
                "workspaces-web:GetNetworkSettings",
                "workspaces-web:GetPortal",
                "workspaces-web:GetPortalServiceProviderMetadata",
                "workspaces-web:GetTrustStore",
                "workspaces-web:GetUserAccessLoggingSettings",
                "workspaces-web:GetUserSettings",
                "workspaces-web:List*",
                "workspaces:Describe*",
                "xray:BatchGetTraces",
                "xray:GetGroup",
                "xray:GetGroups",
                "xray:GetSamplingRules",
                "xray:GetServiceGraph",
                "xray:GetTraceSummaries",
                "xray:List*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AIOPSAPIGatewayAccess",
            "Effect": "Allow",
            "Action": [
                "apigateway:GET"
            ],
            "Resource": [
                "arn:aws:apigateway:*::/restapis",
                "arn:aws:apigateway:*::/restapis/*",
                "arn:aws:apigateway:*::/restapis/*/deployments",
                "arn:aws:apigateway:*::/restapis/*/deployments/*",
                "arn:aws:apigateway:*::/restapis/*/resources/*/methods/*/integrations",
                "arn:aws:apigateway:*::/restapis/*/resources/*/methods/*/integrations/
*",
                "arn:aws:apigateway:*::/restapis/*/stages",
                "arn:aws:apigateway:*::/restapis/*/stages/*",
                "arn:aws:apigateway:*::/apis",
                "arn:aws:apigateway:*::/apis/*",
                "arn:aws:apigateway:*::/apis/*/deployments",
                "arn:aws:apigateway:*::/apis/*/deployments/*",
                "arn:aws:apigateway:*::/apis/*/integrations",
                "arn:aws:apigateway:*::/apis/*/integrations/*",
```

```
                "arn:aws:apigateway:*::/apis/*/stages",
                "arn:aws:apigateway:*::/apis/*/stages/*",
                "arn:aws:apigateway:*::/domainnames/*"
            ]
        }
    ]
}
```

Limitar el acceso de los agentes a una AWS cuenta

AWS DevOps El agente utiliza las funciones de IAM para descubrir y describir AWS los recursos durante las investigaciones de incidentes y las evaluaciones preventivas. Puede controlar el nivel de acceso del agente configurando las políticas de IAM asociadas a estas funciones. La topología de la aplicación no muestra todo a lo que tiene acceso el agente; las políticas de IAM son la única forma de limitar realmente a qué AWS servicios APIs y recursos puede acceder el agente.

Comprender las funciones de IAM para el agente AWS DevOps

AWS DevOps El agente usa las funciones de IAM para acceder a los recursos en dos tipos de cuentas:

- Función de cuenta principal: otorga al agente acceso a los recursos de la AWS cuenta en la que se crea el espacio de agentes.
- Funciones de cuenta secundarias: otorga al agente acceso a los recursos de AWS las cuentas adicionales que usted conecte al espacio de agentes.

Para cualquier tipo de cuenta, puede restringir AWS los servicios a los que puede acceder el agente, limitar el acceso a recursos específicos dentro de esos servicios y controlar en qué regiones puede operar el agente.

Elegir los límites de los recursos

Al limitar el acceso a los recursos, debe incluir permisos suficientes para que el agente investigue correctamente los incidentes de las aplicaciones. Esto incluye:

- Todos los recursos para las aplicaciones incluidas en el ámbito de aplicación que el agente debe supervisar e investigar
- Toda la infraestructura de soporte de la que dependen esas aplicaciones

La infraestructura de soporte puede incluir:

- Componentes de red (subredesVPCs, balanceadores de carga, pasarelas de API)
- Almacenes de datos (bases de datos, cachés, almacenamiento de objetos)
- Recursos informáticos (instancias EC2, funciones Lambda, contenedores)
- Servicios de supervisión y registro (CloudWatch,) CloudTrail
- Recursos de administración de identidad y acceso necesarios para comprender los permisos

Si restringe el acceso de forma demasiado restringida, es posible que el agente no pueda identificar las causas fundamentales que se originan en la infraestructura de soporte que está fuera de los límites definidos.

Restricción del acceso al servicio

Puede limitar AWS los servicios a los que puede acceder el agente modificando las políticas de IAM asociadas a las funciones del agente. Al crear políticas personalizadas, siga estas prácticas recomendadas:

- Otorgue únicamente permisos de solo lectura: el agente debe leer las configuraciones, las métricas y los registros de los recursos durante las investigaciones. Evite conceder permisos que permitan al agente modificar o eliminar recursos.
- Limite a los servicios necesarios: incluya solo los AWS servicios que contienen los recursos relevantes para sus aplicaciones. Por ejemplo, si su aplicación no usa Amazon RDS, no incluya los permisos de RDS en la política.
- Utilice acciones específicas en lugar de caracteres comodín: en lugar de conceder `service:*` permisos, especifique acciones individuales como o. `cloudwatch:GetMetricData` `ec2:DescribeInstances`

Ejemplo de política que se restringe a servicios específicos:

```
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": "*"
    }
  ]
}
```

Limitar el acceso a los recursos

Para limitar el agente a recursos específicos dentro de un servicio, utilice los permisos a nivel de recurso en sus políticas de IAM. Esto le permite conceder acceso únicamente a los recursos que coincidan con patrones específicos.

Uso de patrones de ARN de recursos:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lambda:GetFunction",
        "lambda:GetFunctionConfiguration"
      ],
      "Resource": "arn:aws:lambda:*:*:function:production-*"
    }
  ]
}
```

Este ejemplo limita al agente a acceder únicamente a las funciones de Lambda con nombres que comiencen por «production-».

Uso de restricciones basadas en etiquetas:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeInstanceStatus"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Environment": "production"
        }
      }
    }
  ]
}
```

Este ejemplo limita al agente a acceder únicamente a las instancias de EC2 etiquetadas con. `Environment=production`

Restringir el acceso regional

Para limitar AWS las regiones a las que puede acceder el agente, utilice la clave de `aws:RequestedRegion` condición de sus políticas de 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"
          ]
        }
      }
    }
  ]
}
```

Este ejemplo limita al agente a acceder a los recursos únicamente en las regiones us-east-1 y us-west-2.

Creación de políticas de IAM personalizadas

Al crear un espacio de agente o añadir cuentas secundarias, tiene la opción de crear un rol de IAM personalizado mediante una plantilla de políticas. Esto le permite implementar el principio de privilegios mínimos.

Al crear un espacio de agente

Desde la consola del DevOps agente en la consola AWS de administración...

- Elija Crear un nuevo rol de DevOps agente mediante un documento de política y siga las instrucciones

Al editar un espacio de agente

Desde la consola del DevOps agente a la consola AWS de administración...

- Seleccione la pestaña Capacidades
- Seleccione la cuenta secundaria que desee editar en la sección Cloud y haga clic en Editar
- Elige Crear una nueva política de DevOps agente mediante una plantilla y sigue las instrucciones

Mejores prácticas en materia de políticas personalizadas

- Otorgue únicamente permisos de solo lectura: evite los permisos que permiten la modificación o eliminación de recursos
- Utilice permisos a nivel de recursos siempre que sea posible: restrinja el acceso a recursos específicos mediante patrones o etiquetas ARN

- Revise y audite los permisos con regularidad: revise periódicamente las políticas de IAM del agente para asegurarse de que siguen ajustándose a sus requisitos de seguridad

Configuración de la autenticación de IAM Identity Center

La autenticación del IAM Identity Center proporciona una forma centralizada de gestionar el acceso de los usuarios a la aplicación web AWS DevOps Agent Space. En esta guía se explica cómo configurar la autenticación del IAM Identity Center y cómo gestionar los usuarios.

Requisitos previos

Antes de configurar la autenticación del IAM Identity Center, asegúrese de:

- El Centro de identidad de IAM está activado en su organización o cuenta
- Permisos de administrador en el agente AWS DevOps
- Un espacio de agente configurado o listo para ser creado

Opciones de autenticación

AWS DevOps El agente ofrece dos métodos de autenticación para acceder a la aplicación web Agent Space:

Autenticación de IAM Identity Center: recomendada para entornos de producción. Proporciona una administración centralizada de usuarios, integración con proveedores de identidad externos y sesiones de hasta 12 horas.

Acceso de administrador (autenticación de IAM): proporciona un acceso rápido a los administradores durante la instalación y la configuración iniciales. Las sesiones están limitadas a 30 minutos.

Configuración del centro de identidad de IAM durante la creación de Agent Space

Al crear un espacio de agente, puede configurar la autenticación del Centro de Identidad de IAM en la pestaña Acceso:

Paso 1: Navegue hasta la configuración de la aplicación web

1. Tras configurar los detalles del espacio de agente y el acceso a la AWS cuenta, vaya a la pestaña Acceso
2. Verás dos secciones: «Connect IAM Identity Center» y «Acceso de administrador»

Paso 2: Configurar la integración del centro de identidad de IAM

En la sección Connect [Agent Space] al centro de identidad de IAM:

1. Compruebe la instancia del IAM Identity Center: la consola muestra qué instancia de Identity Center gestionará el acceso de los usuarios de la aplicación web (por ejemplo,`ssoins-7223a9580931edbe`). La instancia de IAM Identity Center más cercana se rellenará automáticamente.
2. Seleccione la opción Nombre del rol de la aplicación de IAM Identity Center y elija una de estas tres opciones:

Cree automáticamente un nuevo rol de DevOps agente (recomendado):

- El sistema crea automáticamente un nuevo rol de servicio con los permisos adecuados
- Esta es la opción más sencilla y funciona en la mayoría de los casos de uso

Asigne un rol existente:

- Utilice un rol de IAM existente que ya haya creado
- El sistema verificará que el rol tiene los permisos necesarios
- Elija esta opción si su organización tiene funciones precreadas para el agente AWS DevOps

Cree un nuevo rol de DevOps agente mediante una plantilla de políticas:

- Utilice los detalles de la política proporcionados para crear su propio rol personalizado en la consola de IAM
- Elija esta opción si necesita personalizar los permisos del rol

Tras hacer clic en Conectar, el sistema automáticamente:

- Crea o configura el rol de IAM especificado
- Configura una aplicación del centro de identidad de IAM para su espacio de agente
- Establece relaciones de confianza entre el Centro de Identidad de IAM y la aplicación web Agent Space
- Configura los flujos de autenticación OAuth 2.0 para un acceso seguro de los usuarios

Alternativa: usar el acceso de administrador

Si quiere acceder inmediatamente a la aplicación web Agent Space sin configurar el IAM Identity Center:

1. En la sección Acceso de administrador, anote el ARN del rol de IAM que proporciona acceso de administrador (por ejemplo,) `arn:aws:iam::440491339484:role/service-role/DevOpsAgentRole-WebappAdmin-15ppoc42`
2. Haga clic en el botón azul de acceso de administrador para iniciar la aplicación web Agent Space con autenticación de IAM
3. Las sesiones que utilizan este método están limitadas a 30 minutos

> **Note**
>
> El acceso de administrador está destinado a la instalación y configuración iniciales. Para uso en producción y operaciones continuas, configure la autenticación del IAM Identity Center.

Agregación de usuarios y grupos

Tras configurar la autenticación del IAM Identity Center, debe conceder a usuarios y grupos específicos acceso a la aplicación web Agent Space:

Paso 1: Acceder a la gestión de usuarios

1. En la consola del AWS DevOps agente, seleccione su espacio de agente
2. Ve a la pestaña Acceso
3. En Acceso de usuarios, haga clic en Administrar usuarios y grupos

Paso 2: Añadir usuarios o grupos

1. Elija Agregar usuarios o grupos
2. Busque usuarios o grupos en el directorio del centro de identidad de IAM
3. Seleccione las casillas de verificación situadas junto a los usuarios o grupos que desee añadir
4. Haga clic en Agregar para concederles acceso

Los usuarios seleccionados ahora pueden acceder a la aplicación web Agent Space con sus credenciales del IAM Identity Center.

Trabajar con proveedores de identidad externos

Si utilizas un proveedor de identidad externo (como Okta, Microsoft Entra ID o Ping Identity) con IAM Identity Center:

- Los usuarios y los grupos se sincronizan desde su proveedor de identidad externo con el Centro de identidades de IAM
- Al añadir usuarios y grupos a la aplicación web Agent Space, los selecciona del directorio sincronizado
- Su proveedor de identidad externo mantiene los atributos de usuario y las pertenencias a grupos
- Los cambios en su proveedor de identidad se reflejan automáticamente en el Centro de identidades de IAM tras la sincronización

Cómo acceden los usuarios a la aplicación web Agent Space

Una vez que haya agregado usuarios a su espacio de agente:

1. Comparta la URL de la aplicación web Agent Space con los usuarios autorizados
2. Cuando los usuarios acceden a la URL, se les redirige a la página de inicio de sesión del IAM Identity Center
3. Tras introducir sus credenciales (y completar el MFA si está configurado), se les redirige de nuevo a la aplicación web Agent Space
4. Su sesión es válida durante 8 horas de forma predeterminada (la puede configurar el administrador del Centro de Identidad)

Administración del acceso de los usuarios

Puede actualizar el acceso de los usuarios en cualquier momento:

Añadir más usuarios o grupos:

- Siga los mismos pasos descritos anteriormente para añadir usuarios o grupos adicionales

Eliminar el acceso:

1. En la sección Acceso de usuario, busque el usuario o grupo que desee eliminar
2. Haga clic en el botón Eliminar situado junto a su nombre
3. Confirme la eliminación

Los usuarios eliminados perderán el acceso inmediatamente, pero las sesiones activas pueden continuar hasta que caduquen.

Administración de sesiones

Las sesiones del IAM Identity Center para la aplicación web Agent Space tienen las siguientes características:

- Duración predeterminada de la sesión: 8 horas
- Seguridad de sesión: cookies solo HTTP para una protección mejorada
- Autenticación multifactorial: se admite cuando se configura en el IAM Identity Center
- Credenciales de API: las credenciales SigV4 de corta duración (15 minutos) se emiten para las llamadas a la API y se renuevan automáticamente

Para configurar la duración de la sesión:

1. Navegue hasta la consola del IAM Identity Center
2. Vaya a Configuración > Autenticación
3. En Duración de la sesión, configura la duración que prefieras (de 1 hora a 12 horas)
4. Elija Guardar cambios.

Desconectar Identity Center

1. En la consola de su Agent Space, haga clic en Acciones en la esquina superior derecha y seleccione Desconectarse del centro de identidades de IAM
2. Confirme en el cuadro de diálogo de confirmación

Configuración de la autenticación de un proveedor de identidad externo (IdP)

La autenticación con un proveedor de identidad externo (IdP) permite a su organización utilizar un proveedor de identidad existente compatible con OIDC, como Okta o Microsoft Entra ID, para administrar el acceso de los usuarios a la aplicación web Agent Space. AWS DevOps Los usuarios inician sesión con sus credenciales corporativas directamente a través de su IdP, sin necesidad de AWS IAM Identity Center.

Requisitos previos

Antes de configurar la autenticación de IdP externa, asegúrese de tener:

- Un proveedor de identidad compatible con OIDC (Okta o Microsoft Entra ID)
- Acceso de administrador a su proveedor de identidad
- Permisos de administrador para acceder a la consola AWS DevOps del agente
- Un espacio de agente configurado o listo para ser creado

Funcionamiento

Al configurar la autenticación de IdP externa:

- Los usuarios navegan hasta la URL de la aplicación web Agent Space
- Se les redirige a la página de inicio de sesión de su proveedor de identidad
- Tras autenticarse con sus credenciales corporativas, se les redirige de nuevo a la aplicación web
- La aplicación web intercambia el token de autenticación por AWS credenciales de corta duración destinadas al Agent Space

Las sesiones son válidas durante un máximo de 8 horas. Las credenciales se actualizan automáticamente mediante los tokens de actualización del OIDC sin que los usuarios tengan que volver a autenticarse.

Configuración de la autenticación de IdP externa

Paso 1: registre una aplicación en su proveedor de identidad

Elija su proveedor de identidad y siga las instrucciones de configuración correspondientes.

Opción A: Okta

1. En la consola de administración de Okta, vaya a Aplicaciones > Aplicaciones y seleccione Crear integración de aplicaciones
2. Seleccione OIDC - OpenID Connect como método de inicio de sesión y Aplicación web como tipo de aplicación. Elija Siguiente.
3. Establezca un nombre descriptivo para la aplicación (por ejemplo,) `AWS DevOps Agent`
4. En Tipo de subvención, asegúrese de que esté marcada la siguiente casilla:
   - Código de autorización (predeterminado)
   - Token de actualización: es necesario para actualizar la sesión. Si no está activado, los usuarios no podrán mantener las sesiones.

> **Note**
>
> Okta no habilita el tipo de concesión Refresh Token de forma predeterminada. Debe habilitarlo de forma explícita.

1. Deje la redirección de inicio de sesión URIs como valor predeterminado por ahora; la actualizará después de configurar el espacio de agentes
2. En Asignaciones, asigne los usuarios o grupos a los que deberían tener acceso
3. Seleccione Save.
4. En la pestaña General de la aplicación, anote los siguientes valores:
   - ID de cliente
   - Secreto de cliente: seleccione Copiar para guardar este valor de forma segura

5. Anote su dominio de Okta: esta es la URL de su emisor (por ejemplo,`https://dev-12345678.okta.com`).

> **Note**
>
> En la pestaña Iniciar sesión, comprueba que el emisor esté configurado como URL de Okta (no dinámica). Esto garantiza una URL de emisor estable.

> **Note**
>
> No añadas la reclamación de un grupo al token de identificación de la pestaña Reclamaciones del servidor de autorización. AWS DevOps El agente no utiliza la pertenencia a un grupo de su IdP.

Opción B: Microsoft Entra ID

1. En el portal de Azure, vaya a Microsoft Entra ID > Registros de aplicaciones > Nuevo registro
2. Establezca un nombre descriptivo (por ejemplo,`AWS DevOps Agent`)
3. En Tipos de cuentas compatibles, selecciona la opción adecuada para tu organización (normalmente, solo las cuentas de este directorio organizativo)
4. Deja el URI de redireccionamiento en blanco por ahora. Selecciona Registrar
5. En la página de descripción general de la aplicación, anote los siguientes valores:
   - ID de aplicación (cliente): se utiliza como ID de cliente al configurar el espacio de agentes
   - ID de directorio (inquilino): se utiliza para construir la URL del emisor
6. Vaya a Certificados y secretos > Nuevo secreto de cliente
   - Establezca una descripción y un período de caducidad
   - Selecciona Añadir y copia el valor secreto inmediatamente; no se volverá a mostrar
7. La URL del emisor de Entra ID sigue este formato. `{tenant-id}`Sustitúyala por tu ID de directorio (inquilino) del paso 5:
   - `https://login.microsoftonline.com/{tenant-id}/v2.0`

> **Note**
>
> No habilites la afirmación opcional del grupo en la configuración del token. AWS DevOps El agente no utiliza la pertenencia a un grupo de su IdP.

Paso 2: Habilite la aplicación Operator con la autenticación de IdP

1. En la consola del AWS DevOps agente, seleccione su espacio de agente
2. Ve a la pestaña Acceso
3. En Acceso de usuario, selecciona Proveedor de identidad externo
4. En el formulario de configuración, configure lo siguiente:
   - Proveedor de identidad: seleccione su proveedor de identidad (Okta o Microsoft Entra ID)
   - URL del emisor: la URL del emisor del OIDC de su proveedor de identidad
   - ID de cliente: el ID de cliente de la aplicación OIDC que creó
   - Secreto de cliente: el secreto de cliente de su aplicación OIDC
5. En Nombre del rol de la aplicación del proveedor de identidad, elija una de estas tres opciones:
   - Crear automáticamente un nuevo rol de DevOps agente (recomendado): crea un nuevo rol de servicio con los permisos adecuados
   - Asigne un rol existente: utilice un rol de IAM existente que ya haya creado
   - Cree un nuevo rol de DevOps agente mediante una plantilla de políticas: utilice los detalles proporcionados para crear su propio rol en la consola de IAM
6. Revise la alerta de advertencia de URL de llamada que aparece en la parte inferior del formulario. Copia esta URL: tendrás que añadirla a la redirección permitida por tu proveedor de identidad para URIs que los usuarios puedan iniciar sesión.
7. Elige Connect

Tras seleccionar Connect, la consola muestra la configuración del proveedor de identidad externo con los siguientes detalles:

- Proveedor: el proveedor de identidad que seleccionó
- URL del emisor: la URL del emisor del OIDC configurada
- ID de cliente: el ID de cliente configurado

- Rol de IAM (ARN): el rol de IAM utilizado para el acceso de los usuarios
- URL de devolución de llamada: configura esta URL en tu proveedor de identidad como una URI de redireccionamiento permitida
- URL de inicio de sesión: utilice esta URL para acceder a la aplicación web a través de su proveedor de identidad

Paso 3: Agrega la URL de devolución de llamada a tu proveedor de identidad

Okta

1. En la consola de administración de Okta, dirígete a la pestaña General de tu aplicación
2. En Iniciar sesión, selecciona Editar
3. Agrega la URL de devolución de llamada como URI de redireccionamiento de inicio de sesión:
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/callback`
4. (Opcional) Defina el URI de inicio de sesión para habilitar el inicio de sesión iniciado por el IdP desde el panel de Okta:
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/login`
5. (Recomendado) Agregue un URI de redireccionamiento de cierre de sesión para redirigir a los usuarios a la aplicación web después de cerrar sesión. De lo contrario, los usuarios podrían ver una página de error al cerrar sesión:
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome`
6. Seleccione Save.

ID de Microsoft Entra

1. En el portal de Azure, vaya a la página de autenticación de la aplicación
2. En Configuraciones de plataforma, elija Agregar una plataforma **>** Web
3. Introduzca la URL de devolución de llamada como URI de redireccionamiento:
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/callback`
4. (Opcional) Agrega un URI de redireccionamiento de cierre de sesión para redirigir a los usuarios a la aplicación web después de cerrar sesión:
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome`
5. Seleccione Configurar

Paso 4: Verificar la configuración

1. Navegue hasta la URL de inicio de sesión que aparece en la consola:
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/login`
2. Deberías ser redirigido a la página de inicio de sesión de tu proveedor de identidad
3. Inicie sesión con sus credenciales corporativas
4. Tras la autenticación correcta, se le redirigirá de nuevo a la aplicación web Agent Space

Actualización de la configuración de IdP

Puede rotar el secreto del cliente sin desconectarlo:

1. En la consola del AWS DevOps agente, selecciona tu espacio de agente
2. Ve a la pestaña Acceso
3. En Configuración del proveedor de identidad externo, elija Rotar el secreto del cliente
4. Introduzca el nuevo secreto de cliente
5. Seleccione Save.

Para cambiar cualquier otro campo de configuración del IdP (como la URL del emisor, el ID de cliente o el proveedor de identidad), debe desconectar el IdP existente y configurar uno nuevo.

Cómo acceden los usuarios a la aplicación web Agent Space

Después de configurar la autenticación de IdP externa:

- Comparta la URL de la aplicación web Agent Space con los usuarios autorizados
- Cuando los usuarios acceden a la URL, se les redirige a la página de inicio de sesión de su proveedor de identidad
- Tras introducir sus credenciales (y completar el MFA si lo ha configurado su IdP), se les redirige de nuevo a la aplicación web Agent Space
- Las sesiones se actualizan automáticamente; consulte Administración de sesiones para obtener más información

Administración de sesiones

Las sesiones de IdP externas para la aplicación web Agent Space tienen las siguientes características:

- Duración de la sesión: las sesiones del navegador duran hasta 8 horas. Esto no se puede configurar en el AWS DevOps agente. Si la duración de la sesión de su IdP supera las 8 horas, los usuarios pueden volver a autenticarse automáticamente en su próxima visita sin necesidad de introducir sus credenciales. Configure la duración de la sesión y el token de su IdP de acuerdo con los requisitos de seguridad de su organización.
- Actualización de credenciales: las sesiones se actualizan automáticamente mediante los tokens de actualización del OIDC sin que los usuarios tengan que volver a autenticarse
- Autenticación multifactorial: se admite cuando se configura en su proveedor de identidad. El IdP gestiona el MFA durante el inicio de sesión; no se necesita ninguna configuración adicional en el agente AWS DevOps

Comportamiento de cierre de sesión

Cuando un usuario hace clic en Cerrar sesión en la aplicación web:

1. Todas las cookies de sesión se borran inmediatamente
2. Se redirige al usuario al punto de cierre de sesión OIDC del proveedor de identidad para finalizar la sesión de SSO
3. Si se configura un URI de redireccionamiento de cierre de sesión, se redirige al usuario a la página de bienvenida de la aplicación web

Revocar el acceso de los usuarios

Para revocar inmediatamente el acceso de un usuario, puedes revocar sus sesiones directamente en el portal de administración de tu proveedor de identidad:

- Okta: en la consola de administración de Okta, navega hasta Directorio > Personas, selecciona el usuario y selecciona Más acciones > Borrar sesiones de usuario
- Microsoft Entra ID: en el portal de Azure, vaya a Usuarios, seleccione el usuario y elija Revocar sesiones

Consideraciones de seguridad

Almacenamiento del secreto del cliente: el secreto del cliente que proporciona durante la configuración se cifra con la clave de KMS administrada por el cliente, si la proporcionó al crear el espacio de agente, o con una clave propiedad del servicio en caso contrario. Después de la configuración inicial, nunca se devuelve en las respuestas de la API ni se muestra en la consola.

Rotación de los secretos de los clientes: los secretos de los clientes de Entra tienen una caducidad configurable. Configura un recordatorio para rotar el secreto antes de que caduque mediante la opción Rotar el secreto del cliente de la consola del AWS DevOps agente. Si el secreto caduca, los usuarios no podrán iniciar sesión hasta que se rote.

Administración de la vida útil de los tokens: la duración de los tokens (tokens de acceso, tokens de actualización) emitidos por su proveedor de identidad depende de la configuración de su IdP. Recomendamos configurar los tiempos de vida de los tokens adecuados en su IdP:

- Okta: Configura la vida útil de los tokens en Seguridad > API > Servidores de autorización > Políticas de acceso
- Microsoft Entra ID: configure la vida útil de los tokens mediante políticas de vida útil de los tokens

Notificación de grupos: no habilite la reclamación de grupos en la configuración de token de su proveedor de identidad. AWS DevOps Actualmente, el agente no utiliza la pertenencia a un grupo de su IdP.

Identificador de usuario: el AWS DevOps agente utiliza una afirmación específica del proveedor para identificar a los usuarios de forma exclusiva:

- Okta: utiliza la `sub` afirmación del token de identificación
- Microsoft Entra ID: utiliza la afirmación `oid` (identificador de objeto) del token de ID

Estos identificadores son inmutables y aparecen en los CloudTrail registros con fines de auditoría.

Desconectar el IdP externo

1. En la consola del AWS DevOps agente, seleccione su espacio de agente
2. Ve a la pestaña Acceso
3. En Acceso de usuario, selecciona Desconectar

4. Revise los impactos que aparecen en el cuadro de diálogo de confirmación y confirme

La desconexión permitirá:

- Elimine la configuración de IdP del espacio de agentes
- Impida que los usuarios inicien sesión a través del proveedor de identidad externo
- Eliminar el historial individual de chat y artefactos asociado a las cuentas de usuario de IdP

Las sesiones de usuario activas continuarán hasta que caduquen o se produzca un error en la próxima actualización de credenciales.

Resolución de problemas

- La redirección al IdP falla: compruebe que la URL del emisor coincida con el punto final de detección de OIDC de su IdP. En el caso de Okta, asegúrate de que el emisor esté configurado como URL de Okta (no dinámica) en la pestaña de inicio de sesión. Para Entra, usa el formato. `https://login.microsoftonline.com/{tenant-id}/v2.0`
- Acceso denegado o error de política (Okta): compruebe que el usuario o su grupo estén asignados a la aplicación en Asignaciones. Seleccione Inicio de sesión **>** Reglas de la política de inicio de sesión.
- Error de configuración del IdP después del inicio de sesión: su proveedor de identidad no devolvió un token de actualización. Asegúrese de que el `offline_access` alcance y el tipo de concesión del token de actualización estén habilitados:
  - Okta: Ve a la pestaña General de tu aplicación y activa la casilla Actualizar el token en Tipo de concesión
  - Entra: ve a los permisos de la API y asegúrate de que aparezca en `offline_access` la lista de permisos delegados
- La autenticación se realiza correctamente, pero la aplicación web muestra un error: compruebe que el URI de redireccionamiento de su IdP coincida exactamente con la URL de devolución de llamada que se muestra en AWS DevOps la consola del agente.
- Fallos de autenticación: si la notificación opcional del grupo está habilitada en su IdP, deshabilítela. AWS DevOps El agente no usa notificaciones grupales.
- El inicio de sesión falla después de la autenticación del IdP: en el caso de Entra, la verificación no **`requestedAccessTokenVersion`** está configurada **`null`** en el manifiesto de la aplicación. En el caso de Okta, compruebe que la URL del emisor sea correcta.

- Página de error tras hacer clic en Cerrar sesión (Okta): si ves un **`post_logout_redirect_uri`** error después de cerrar sesión, agrégala **`https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome`** como URI de redirección de cierre de sesión en la pestaña General de la aplicación de Okta.
- Los usuarios permanecen en la página del proveedor de identidad después de cerrar sesión (Entra): para redirigir a los usuarios a la aplicación web después de cerrar sesión, añada **`https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome`** un URI de redireccionamiento en la página de autenticación de la aplicación Entra.

Cifrado en reposo para AWS DevOps Agent

AWS DevOps El agente cifra todos los datos de los clientes en reposo. De forma predeterminada, el AWS DevOps agente utiliza las claves AWS propias para cifrar automáticamente los datos sin coste adicional. No puede ver, administrar ni auditar el uso de claves AWS propias. Sin embargo, no es necesario que tome ninguna medida para proteger estas claves. Sus datos se protegen automáticamente.

Puede optar por cifrar sus datos mediante una clave simétrica gestionada por el cliente que cree, posea y gestione en el Servicio de administración de AWS claves (AWS KMS). Como tiene el control total de esta capa de cifrado, puede realizar tareas como las siguientes:

- Establecer y mantener políticas de claves
- Habilitar y deshabilitar políticas de claves
- Rotar el material criptográfico
- Adición de etiquetas de
- Crear alias de clave
- Programar la eliminación de claves

Para obtener más información, consulte las claves administradas por el cliente en la Guía AWS para desarrolladores del Servicio de administración de claves.

> **Note**
>
> AWS DevOps El agente habilita automáticamente el cifrado en reposo mediante claves AWS propias para proteger los datos de los clientes sin coste alguno. Se aplican cargos de

AWS KMS estándar cuando se utiliza una clave gestionada por el cliente. Para obtener más información sobre los precios, consulte los precios del servicio de administración de AWS claves.

Claves administradas por el cliente

Las claves administradas por el cliente son claves de KMS de su AWS cuenta que usted crea, posee y administra. Usted tiene el control total sobre estas claves de KMS, incluido el establecimiento y el mantenimiento de sus políticas clave.

Al configurar una clave gestionada por el cliente, AWS DevOps Agent la utiliza para proteger los datos confidenciales de los recursos. AWS DevOps El agente utiliza el cifrado de sobres con el conjunto de claves jerárquicas del SDK de AWS cifrado. Su clave KMS se utiliza para generar claves de sucursal, que a su vez protegen sus datos.

Puede especificar una clave administrada por el cliente al crear los siguientes recursos:

- Agent Space: cifra los detalles y el contenido de Agent Space creados a partir de la aplicación web DevOps Agent relacionados con las investigaciones, las habilidades y el chat.
- Servicio: cifra las credenciales de servicio de terceros en reposo.

Para configurar una clave gestionada por el cliente en AWS DevOps Agent, siga estos pasos.

Paso 1: Crear una clave administrada por el cliente

Puede crear una clave simétrica gestionada por el cliente mediante la consola de AWS KMS o la API de AWS KMS. La clave debe cumplir los siguientes requisitos:

Propiedad	Requisito
Tipo de clave	Simétrica
Especificación de clave	`SYMMETRIC_DEFAULT`
Uso de clave	`ENCRYPT_DECRYPT`

> **Note**
>
> AWS DevOps El agente solo admite claves KMS de cifrado simétrico con la especificación y el uso de la `SYMMETRIC_DEFAULT ENCRYPT_DECRYPT` clave. Actualmente, no se admiten las claves multirregionales ni las claves asimétricas.

Para obtener más información, consulte Creación de una clave simétrica gestionada por el cliente en la Guía para desarrolladores de AWS Key Management Service.

Paso 2: Defina la política clave

Las políticas de clave controlan el acceso a la clave administrada por el cliente. Cada clave administrada por el cliente debe tener exactamente una política de clave, que contiene instrucciones que determinan quién puede usar la clave y cómo puede utilizarla.

Su política clave debe conceder permisos tanto a la persona principal que realiza la llamada (su identidad de IAM) como al servicio de AWS DevOps agente. AWS DevOps El agente accede a su clave mediante dos conjuntos de credenciales:

1. Sus credenciales de la persona que llama: se utilizan para todas las operaciones sincrónicas, incluida la validación de claves, el cifrado al momento de la creación del recurso y cualquier llamada a la API que devuelva una respuesta directa a la persona que llama.
2. AWS DevOps Agente principal: se utiliza para operaciones asíncronas que se ejecutan en segundo plano, como las investigaciones operativas, el análisis de incidentes, la correlación de eventos y la generación de análisis de la causa raíz.

En la siguiente tabla se enumeran las acciones de KMS necesarias:

Acción de KMS	Description (Descripción)
`kms:DescribeKey`	Valide la configuración clave en el momento de la creación del recurso
`kms:GenerateDataKey`	Genere claves de cifrado de datos para el cifrado de sobres
`kms:Decrypt`	Descifrado de datos

Acción de KMS	Description (Descripción)
`kms:Encrypt`	Cifrar datos
`kms:ReEncrypt`	Vuelva a cifrar los datos con la misma clave o con una clave diferente

AWS DevOps El agente valida todos estos permisos en el momento de la configuración mediante operaciones de ejecución en seco. Si falta algún permiso, se produce un error en la solicitud, salvo una excepción.

A continuación, se muestra una política de claves de ejemplo. Sustituya los valores de los marcadores de posición por los suyos propios.

```
{
  "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 política contiene las siguientes declaraciones:

- AllowKeyAdministration— Otorga al usuario raíz de la cuenta acceso administrativo completo a la clave. `111122223333`Sustitúyala por tu ID de AWS cuenta.
- AllowCallerAccessViaService— Otorga a sus directores de IAM los permisos de KMS necesarios para todas las operaciones sincrónicas AWS DevOps del agente. Esto incluye la validación de claves en el momento de la creación del recurso, así como las operaciones de cifrado y descifrado para cualquier llamada a la API que devuelva una respuesta directa a la persona que llama. Esta `kms:ViaService` condición garantiza que solo se pueda utilizar la clave a través del AWS DevOps servicio de agente. `111122223333`Sustitúyala por tu ID de AWS cuenta y `us-east-1` por tu AWS región.
- AllowDevOpsAgentServiceAccessForAgentSpace/AllowDevOpsAgentServiceAccessForService— Otorga al director del `aidevops.amazonaws.com` servicio los permisos de KMS necesarios para las operaciones asincrónicas. AWS DevOps El agente utiliza este principio de servicio para cifrar y descifrar los datos al realizar operaciones en segundo plano, como investigaciones operativas, analizar incidentes, correlacionar eventos entre servicios y generar análisis de la causa raíz. Sin este acceso, AWS DevOps Agent no puede leer los datos cifrados necesarios para llevar a cabo las investigaciones en su nombre. La `aws:SourceArn` condición restringe el acceso a las solicitudes que se originan en los recursos de su AWS DevOps agente y garantiza que el contexto de cifrado coincida con su recurso ARNs. `kms:EncryptionContext` `111122223333`Sustitúyala por tu ID de AWS cuenta y `us-east-1` por tu AWS región.

Para obtener más información sobre las políticas clave, consulte Políticas clave de AWS KMS en la Guía AWS para desarrolladores del Servicio de administración de claves.

Paso 3: Especifique la clave al crear un recurso

Tras crear la clave y configurar la política de claves, puede especificarla al crear los recursos del AWS DevOps agente.

Consola

Para configurar una clave gestionada por el cliente al crear un espacio de agente en la consola:

1. Abra la consola del AWS DevOps agente.
2. Seleccione Crear espacio de agente o Registrar servicio.
3. Introduzca los detalles del espacio de agentes (nombre, descripción y función de IAM).
4. Amplíe la sección Configuración avanzada.
5. En Tipo de clave de cifrado, selecciona Clave gestionada por el cliente.
6. Elija una clave de KMS de la lista desplegable o introduzca un ARN de clave de KMS.
7. Revise la política clave que se muestra en la sección Política clave ampliable. Asegúrese de haber adjuntado esta política a su clave de KMS. Puede usar el botón de copiar para copiar la política.
8. Complete el resto de la configuración y elija Crear.

> **Note**
>
> Si no ve la clave de KMS en la lista desplegable, compruebe que la clave cumple los requisitos del paso 1 y que dispone `kms:ListKeys` de `kms:DescribeKey` los permisos necesarios.

API

Crear un espacio de agente con una clave gestionada por el cliente

Especifique el `kmsKeyArn` parámetro al crear un espacio de agentes. El valor debe ser el ARN completo de la clave KMS.

```
{
  "name": "my-agent-space",
  "description": "An encrypted agent space",
  "kmsKeyArn": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
```

```
}
```

Registrar un servicio con una clave gestionada por el cliente

Especifique el `kmsKeyArn` parámetro al registrar un servicio. El valor debe ser el ARN completo de la clave KMS. Este parámetro es compatible con todos los tipos de servicios, incluidos los servidores Dynatrace, ServiceNow, PagerDuty GitLab GitHub, y MCP.

```
{
  "service": "dynatrace",
  "kmsKeyArn": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "serviceDetails": { ... }
}
```

> **Note**
>
> Debe especificar la clave administrada por el cliente en el momento de crear el recurso. No puede añadir ni cambiar la clave gestionada por el cliente para un recurso existente.

AWS DevOps Contexto de cifrado del agente

Un contexto de cifrado es un conjunto de pares clave-valor no secretos que contienen información contextual adicional sobre los datos. AWS El KMS usa el contexto de cifrado como datos autenticados adicionales para admitir el cifrado autenticado. Al incluir un contexto de cifrado en una solicitud de cifrado de datos, AWS KMS vincula el contexto de cifrado a los datos cifrados. Para descifrar los datos, debe incluir el mismo contexto de cifrado en la solicitud.

AWS DevOps El agente utiliza el siguiente contexto de cifrado en todas las operaciones criptográficas:

```
{
  "aws-crypto-ec:aws:aidevops:arn": "arn:aws:aidevops:{region}:{accountId}:
{resourceType}/{resourceId}"
}
```

El valor del contexto de cifrado es el ARN del recurso de AWS DevOps agente que se está cifrando. Puede utilizar este contexto de cifrado en las condiciones de su política de claves y en AWS CloudTrail los registros para auditar cómo se utiliza su clave.

Administración de claves

Si deshabilita o programa la eliminación de su clave KMS, el AWS DevOps agente no podrá descifrar sus datos. Esto provoca `AccessDeniedException` errores en las operaciones que leen datos cifrados.

> **Important**
>
> Si decide usar una clave administrada por el cliente, usted es responsable de administrar la clave y sus permisos. Si la clave se deshabilita o se elimina, o si el AWS DevOps agente pierde el permiso para usarla, usted pierde el acceso a los datos cifrados.

En la siguiente tabla se describen los escenarios de error más comunes:

Action	Impact
Permisos de política clave revocados	`AccessDeniedException` sobre las operaciones de cifrado y descifrado
La clave KMS está deshabilitada	`DisabledException` en las operaciones de cifrado y descifrado
La eliminación de la clave KMS está programada	`KMSInvalidStateException` sobre las operaciones de cifrado y descifrado
Se elimina la clave KMS	Pérdida permanente de datos: los datos cifrados no se pueden recuperar

Antes de deshabilitar o eliminar una clave:

1. Compruebe que ningún recurso del AWS DevOps agente activo dependa de la clave.
2. Considere la posibilidad de deshabilitar primero la clave para comprobar el impacto antes de programar su eliminación.
3. AWS KMS impone un período de espera mínimo antes de eliminar la clave, lo que le da tiempo para cancelarla si es necesario.

Nota: El AWS DevOps agente no vuelve a cifrar automáticamente los datos con una clave nueva. Si necesita cambiar a una nueva clave gestionada por el cliente, debe crear un nuevo recurso con la nueva clave.

Supervisión de sus claves de cifrado

Cuando utiliza una clave gestionada por el cliente con AWS DevOps Agent, puede utilizarla AWS CloudTrailpara realizar un seguimiento de las solicitudes que el AWS DevOps agente envía a AWS KMS.

Puede filtrar CloudTrail los eventos por:

- Fuente del evento: `kms.amazonaws.com`
- Clave de contexto de cifrado — `aws-crypto-ec:aws:aidevops:arn`
- ARN clave: el ARN clave gestionado por el cliente en los parámetros de la solicitud

Para obtener más información, consulte Registrar las llamadas a la API de AWS KMS AWS CloudTrail en la Guía para desarrolladores del servicio de administración de AWS claves.

Puntos de enlace de la VPC (AWS PrivateLink)

Se puede utilizar AWS PrivateLink para crear una conexión privada entre la VPC y AWS DevOps el agente. Puede acceder al AWS DevOps agente como si estuviera en su VPC, sin utilizar una puerta de enlace a Internet, un dispositivo NAT, una conexión VPN o una conexión Direct Connect. Las instancias de su VPC no necesitan direcciones IP públicas para acceder AWS DevOps al agente.

Esta conexión privada se establece mediante la creación de un punto final de interfaz, con la tecnología de AWS PrivateLink. Creamos una interfaz de red de punto de conexión en cada subred habilitada para el punto de conexión de interfaz. Se trata de interfaces de red administradas por el solicitante que sirven como punto de entrada para el tráfico destinado al agente. AWS DevOps

Para obtener más información, consulte Acceder a AWS los servicios AWS PrivateLink en la Guía.AWS PrivateLink

Consideraciones sobre los puntos AWS DevOps finales de Agent VPC

Antes de configurar un punto final de interfaz para AWS DevOps Agent, consulte las consideraciones de la AWS PrivateLink guía.

AWS DevOps El agente admite la realización de llamadas a la API a través de los siguientes puntos finales de VPC.

Categoría	Sufijo de punto de conexión
AWS DevOps Acciones de la API del plano de control del agente	`aidevops`
AWS DevOps Operaciones en tiempo de ejecución del agente	`aidevops-dataplane`
AWS DevOps Eventos de Agent Webhook	`event-ai`

Cree un punto final de interfaz para el agente AWS DevOps

Puede crear un punto final de interfaz para el AWS DevOps agente mediante la consola de Amazon VPC o la interfaz de línea de AWS comandos (AWS CLI). Para obtener más información, consulte Creación de un punto de conexión de interfaz en la Guía de AWS PrivateLink .

Cree un punto final de interfaz para el AWS DevOps agente con los siguientes nombres de servicio:

- com.amazonaws. {región} .aidevops
- com.amazonaws. {región} .aidevops-dataplane
- com.amazonaws. {región} .event-ai

Cuando se crea el punto de conexión, tiene la opción de habilitar un nombre de host de DNS privado. Habilite esta configuración seleccionando Enable Private DNS Name (Habilitar nombre de DNS privado) en la consola de VPC al crear el punto de conexión de la VPC.

Si habilita el DNS privado para el punto final de la interfaz, puede realizar solicitudes de API al AWS DevOps agente utilizando su nombre de DNS regional predeterminado. En el siguiente ejemplo, se muestra el formato del nombre DNS regional predeterminado.

- aidevops. {región} .api.aws
- aidevops-dataplane. {region} .amazonaws.com
- event-ai. {región} .api.aws

Creación de una política de puntos de conexión para el punto de conexión de interfaz

Una política de punto de conexión es un recurso de IAM que puede adjuntar a un punto de conexión de interfaz. La política de puntos finales predeterminada permite el acceso total al AWS DevOps agente a través del punto final de la interfaz. Para controlar el acceso permitido al AWS DevOps agente desde su VPC, adjunte una política de punto final personalizada al punto final de la interfaz.

Una política de punto de conexión especifica la siguiente información:

- Los principales que pueden realizar acciones (AWS cuentas, usuarios de IAM y funciones de IAM).
- Las acciones que se pueden realizar.
- El recurso en el que se pueden realizar las acciones.

Para obtener más información, consulte Control del acceso a los servicios con políticas de punto de conexión en la Guía del usuario de AWS PrivateLink .

Cuotas

AWS DevOps Las cuotas de agentes incluyen la cantidad de espacios de agentes, investigaciones simultáneas y más. Puede solicitar los aumentos de algunas cuotas, pero no todas se pueden aumentar. Estos aumentos no se conceden de forma inmediata, por lo que el aumento puede tardar unas horas o días en hacerse efectivo. A menos que se indique lo contrario, cada cuota es específica de la región.

En la siguiente tabla se describen las cuotas de AWS DevOps Agent.

Name	Predeterminado	Ajustable	Description (Descripción)
Espacios de agentes por cuenta y región	10	Sí	El número máximo de espacios de agentes que puede crear por cuenta en cada AWS región.
Investigaciones simultáneas por espacio de agentes	3	Sí	El número máximo de investigaciones de resolución de incidentes que se pueden ejecutar simultáneamente en un único espacio de agentes.
Evaluaciones simultáneas por espacio de agentes	1	No	El número máximo de evaluaciones de prevención de incidentes que se pueden ejecutar simultáneamente en un único espacio de agentes.

Name	Predeterminado	Ajustable	Description (Descripción)
Invocaciones simultáneas bajo demanda por espacio de agente	10	Sí	El número máximo de DevOps invocaciones bajo demanda que se pueden ejecutar simultáneamente en un único espacio de agente.

Solicitud de aumento de cuota

Puede solicitar un aumento de cuota mediante una de las siguientes opciones:

- Desde la consola AWS de administración: abra la consola Service Quotas. En el panel de navegación, elija Servicios de AWS . Seleccione un DevOps agente, seleccione una cuota y siga las instrucciones para solicitar un aumento de cuota. Para obtener más información, consulte Solicitud de aumento de cuota en la Guía del usuario de Service Quotas.
- Desde la AWS CLI: utilice el comando request-service-quota-increase AWS CLI. Para obtener más información, consulte Solicitud de aumento de cuota en la Guía del usuario de Service Quotas.

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.