Manual do usuário

AWS DevOps Agente

AWS DevOps Agente: Manual do usuário

Table of Contents

Sobre AWS DevOps Agente

AWS DevOps O agente é um agente de fronteira que resolve e previne incidentes de forma proativa, melhorando continuamente a confiabilidade e o desempenho.

AWS DevOps O agente investiga incidentes e identifica melhorias operacionais como engenheiro experiente. DevOps

O agente trabalha por:

- Aprendendo seus recursos e seus relacionamentos.
- Trabalhando com suas ferramentas de observabilidade, habilidades, repositórios de código e CI/CD pipelines.
- Correlacionando dados de telemetria, código e implantação para entender as relações entre os recursos do seu aplicativo.
- Suporte a aplicativos em ambientes multicloud e híbridos.

Recursos principais

AWS DevOps O Agent fornece recursos abrangentes de prevenção e resposta a incidentes por meio dos seguintes recursos:

Always-on, resposta autônoma a incidentes

AWS DevOps O agente investiga problemas de forma autônoma no momento em que eles ocorrem:

- Investigação automatizada de incidentes — começa a investigar imediatamente quando um alerta ou ticket de suporte chega
- AWS DevOps Chat do agente - consulte sua infraestrutura, analise a integridade do sistema e oriente as investigações usando linguagem natural em todo o aplicativo web do DevOps Agent Space. O Chat fornece respostas contextuais com base na página que você está visualizando, seja perguntando sobre recursos em Topologia, conduzindo uma investigação ou filtrando recomendações na Prevenção.
- Planos de mitigação detalhados — Fornecem ações específicas para resolver incidentes, validar o sucesso e reverter alterações, se necessário

- Coordenação automatizada de incidentes — direciona observações, descobertas e etapas de mitigação por meio de seus canais de comunicação preferidos, como Slack e ServiceNow
- AWS Integração de AWS suporte — Crie casos de suporte diretamente de uma investigação com contexto imediato fornecido aos especialistas do AWS Support

Previna futuros incidentes

AWS DevOps O agente analisa padrões em incidentes históricos para ajudá-lo a passar do combate a incêndios reativo para a melhoria operacional proativa:

- Recomendações direcionadas — Oferece melhorias específicas e acionáveis que fortalecem quatro áreas principais: observabilidade (monitoramento, alerta, registro), otimização da infraestrutura (escalonamento automático, ajuste de capacidade) e aprimoramento do pipeline de implantação (teste, validação).
- Aprendizado contínuo — refina as recomendações com base no feedback da sua equipe

Obtenha mais de suas DevOps ferramentas

AWS DevOps O agente se integra às suas ferramentas existentes sem alterar seus fluxos de trabalho:

- Mapeamento de recursos do aplicativo — cria um gráfico de topologia dos recursos do seu aplicativo e seus relacionamentos
- Built-in integrações — Funciona com ferramentas populares de observabilidade (Amazon CloudWatch, Dynatrace, Datadog, New Relic e Splunk), repositórios de código e CI/CD pipelines (ações e repositórios, fluxos de trabalho e repositórios) GitHub GitLab
- Integração de ferramentas personalizadas — amplie os recursos conectando-se aos seus próprios servidores do Model Context Protocol (MCP) para obter ferramentas adicionais
- Consultas de infraestrutura conversacional — Use linguagem natural para consultar AWS recursos, métricas do sistema e status do alarme sem navegar em vários consoles. O Chat entende o contexto e mantém o histórico de conversas para perguntas complementares.

Como AWS DevOps Agente trabalha

AWS DevOps O agente opera por meio de uma arquitetura de console duplo. Os administradores usam o AWS Management Console para criar e gerenciar Agent Spaces, configurar integrações e configurar controles de acesso. As equipes de operações usam o aplicativo web AWS DevOps Agent para atividades diárias de resposta a incidentes e investigação. O aplicativo web é onde os operadores podem interagir com as investigações de agentes, pesquisar a topologia de aplicativos entre contas e aprender sobre melhorias preventivas na observabilidade, no código, nos pipelines e nas arquiteturas de infraestrutura. Para saber mais, consulte the section called “Prevenção proativa de incidentes”.

O serviço é organizado em torno de Agent Spaces, que são contêineres lógicos que definem o que o AWS DevOps Agente pode acessar e investigar. Cada Espaço do Agente contém as configurações AWS da sua conta, integrações de ferramentas de terceiros e permissões de acesso. Para saber mais, consulte the section called “O que são DevOps Agent Spaces?”.

AWS DevOps O agente cria automaticamente uma topologia de aplicativo que mapeia seus recursos e seus relacionamentos. Essa topologia ajuda o serviço a entender a arquitetura do seu aplicativo durante as investigações. Para saber mais, consulte the section called “O que é uma topologia de DevOps agente?”.

Benefícios

- Reduza o tempo médio de resolução (MTTR) — A investigação autônoma começa imediatamente, acelerando a resolução de incidentes de horas para minutos
- Evite incidentes recorrentes — recomendações direcionadas abordam as causas principais e fortalecem a resiliência do sistema
- Melhore a eficiência operacional — liberte sua equipe de tarefas repetitivas de investigação para se concentrar na inovação
- Trabalhe dentro dos fluxos de trabalho existentes — Integra-se às ferramentas e processos existentes sem interrupções

O que é um DevOps Agent Web App?

AWS DevOps O agente usa uma arquitetura de console duplo que separa as funções administrativas das day-to-day atividades operacionais. Esse design permite que os administradores configurem o serviço enquanto as equipes de operações se concentram na resposta e prevenção de incidentes.

Consoles

AWS DevOps O agente fornece duas interfaces distintas:

- AWS Console de gerenciamento — Os administradores usam o console AWS de gerenciamento para configurar e gerenciar o AWS DevOps agente. Nesse console, você pode the section called "Criação de um espaço de agente" conectar AWS serviços e ferramentas de terceiros e gerenciar permissões de acesso para sua organização.
- DevOps Aplicativo web do agente - As equipes de operações usam os aplicativos web do DevOps Agent Space para atividades diárias de resposta a incidentes. Esse aplicativo independente fornece uma interface na qual engenheiros de plantão podem iniciar investigações, interagir com o agente por meio de bate-papo em linguagem natural, visualizar topologias de aplicativos e revisar recomendações de prevenção de incidentes.

Capacidades do aplicativo Web

O aplicativo web DevOps Agent fornece os seguintes recursos principais:

- Resposta a incidentes — A página é onde você cria e rastreia investigações de incidentes, bem como gera planos de mitigação para resolver incidentes.
- Prevenção de incidentes — Na página Prevenção, é aqui que você encontrará recomendações para melhorar sua postura de observabilidade, processos de entrega e arquitetura de infraestrutura para evitar futuros incidentes.
- Topologia — A página Topologia fornece uma representação visual interativa dos recursos da conta e seus relacionamentos em todos os recursos nas contas conectadas. Você pode visualizar a topologia com diferentes níveis de detalhes usando o menu suspenso "Mostrar" para alternar entre as visualizações de sistema, contêiner e recurso.
- Habilidades — Conjuntos de instruções modulares que ampliam o AWS DevOps Agent com recursos especializados. As habilidades contêm conhecimento de domínio, metodologias de investigação e configurações de ferramentas personalizadas para sua infraestrutura. Cada

habilidade habilita ferramentas específicas e fornece divulgação progressiva de instruções somente quando relevante para a investigação.

- Interface de bate-papo em linguagem natural — Disponível em todo o aplicativo web, o Chat é um assistente de conversação com inteligência artificial que permite consultar sua infraestrutura, analisar a integridade do sistema e trabalhar com investigações usando linguagem natural. O Chat fornece respostas contextuais com base na página que você está visualizando.

Autenticação

AWS DevOps O agente oferece suporte a métodos flexíveis de autenticação para acomodar diferentes requisitos organizacionais:

- Integração do IAM Identity Center (acesso do usuário) — As organizações podem usar o AWS Identity Center (IAM Identity Center) para gerenciar centralmente o acesso dos usuários aos aplicativos web do DevOps Agent Space. O IAM Identity Center pode se federar com provedores de identidade externos por meio de protocolos OIDC e SAML padrão, incluindo provedores como Okta, Ping Identity e Microsoft Entra ID. Esse método oferece suporte à autenticação multifatorial do seu provedor de identidade.
- Autenticação de provedor de identidade externo (IdP) — As organizações podem conectar um provedor de identidade compatível com OIDC, como Okta ou Microsoft Entra ID, diretamente ao aplicativo web Agent Space sem precisar do IAM Identity Center. Os usuários fazem login com suas credenciais corporativas por meio do IdP. Para obter instruções de configuração, consultethe section called “Configurando a autenticação do provedor de identidade externo (IdP)”.
- Link de autenticação do IAM (acesso de administrador) — Um método alternativo fornece acesso direto ao aplicativo web a partir do AWS Management Console usando sua sessão de console existente. Essa opção é útil antes de implementar a integração completa do Identity Center, mas as sessões são limitadas a 10 minutos.

O que são DevOps Agent Spaces?

Um Espaço do DevOps Agente é um contêiner lógico que define as ferramentas e a infraestrutura às quais o AWS DevOps Agente tem acesso. Cada Espaço do Agente opera de forma independente com seu próprio acesso à AWS conta, integrações de terceiros e permissões de usuário.

Um Espaço do Agente representa o limite do que o AWS DevOps Agente pode acessar e investigar durante a resposta a incidentes. Ao criar um Espaço do Agente, você define quais AWS contas o

agente pode acessar, a quais ferramentas externas ele pode se conectar e quais usuários da sua organização podem interagir com o agente.

Cada Espaço do Agente funciona como uma implantação independente do AWS DevOps Agente. Você configura o Agent Space por meio do AWS Management Console, enquanto suas equipes de operações usam o aplicativo web do Agent Space para conduzir investigações e revisar recomendações dentro desse espaço.

Como os Agent Spaces são isolados

Os Agent Spaces mantêm o isolamento para garantir a segurança e evitar o acesso não intencional em diferentes ambientes ou equipes:

- AWS isolamento da conta — cada Agent Space usa funções dedicadas do IAM que concedem acesso somente a AWS contas e recursos específicos. O agente não pode acessar AWS recursos fora daqueles explicitamente configurados para o Espaço do Agente.
- Isolamento do acesso do usuário — Você controla quais usuários ou grupos podem acessar cada Espaço do Agente. Isso permite que você alinhe as permissões de acesso à sua estrutura organizacional, garantindo que as equipes interajam apenas com os Agent Spaces designados.
- Isolamento de dados — Os dados da investigação, o histórico de incidentes e as recomendações são mantidos separadamente em cada Espaço do Agente. As informações de um Espaço do Agente não são visíveis nem acessíveis a partir de outro Espaço do Agente.
- Isolamento de dados do chat - O histórico de conversas do chat também é isolado dentro de cada espaço do agente. Conversas e consultas em um Espaço do Agente não são visíveis nem acessíveis a partir de outro Espaço do Agente.

Aplicativo Web Agent Space

Cada Espaço do Agente tem um aplicativo web dedicado que pode ser acessado fora do AWS Management Console. Consulte the section called “O que é um DevOps Agent Web App?” para saber mais sobre o aplicativo web.

Quando usar vários Agent Spaces

Considere criar vários Agent Spaces para atender às diferentes necessidades organizacionais:

- Separação de equipes — Crie Agent Spaces dedicados para diferentes equipes de aplicativos ou unidades de negócios para manter limites claros de propriedade no Agent Space.

- Isolamento do ambiente — Separe os ambientes de produção e não produção em diferentes Agent Spaces para evitar o acesso acidental entre ambientes.
- Limites do serviço — Alinhe os Agent Spaces aos limites de serviços ou aplicativos específicos para manter as investigações focadas e relevantes.
- Requisitos de conformidade — configure Agent Spaces separados com diferentes controles de acesso ou configurações de residência de dados para atender aos requisitos regulamentares.

> **Note**
>
> Ao criar vários Agent Spaces, você pode usar uma AWS conta dedicada como conta principal para um Agent Space e conectar contas de aplicativos distintas como contas secundárias. Essa abordagem permite que você mantenha controles de acesso granulares e, ao mesmo tempo, garanta que cada Espaço do Agente possa acessar somente os recursos específicos do escopo pretendido, mesmo usando a criação automática de funções.

O que é uma topologia de DevOps agente?

AWS DevOps O agente descobre e visualiza automaticamente os recursos e relacionamentos em seus aplicativos e usa a topologia resultante para entender sua infraestrutura durante investigações de incidentes e ao fazer recomendações preventivas.

Como os gráficos de topologia são criados

AWS DevOps O agente cria gráficos de topologia por meio de vários processos automatizados:

- Descoberta de recursos — O agente verifica automaticamente suas AWS contas para identificar recursos como instâncias de computação, serviços de armazenamento, componentes de rede e bancos de dados que fazem parte de seus aplicativos.
- Detecção de relacionamento — o agente analisa dados de configuração, CloudFormation pilhas e tags de recursos para determinar como os recursos se relacionam entre si.
- Mapeamento de código e implantação — Quando conectado a CI/CD pipelines, o agente vincula os recursos de infraestrutura aos seus processos de implantação e ao código alterado do aplicativo e da infraestrutura.

- Mapeamento do comportamento de observabilidade — Dados de sistemas de observabilidade, como Amazon CloudWatch Application Signals e Dynatrace, são usados para identificar comportamentos observados que indicam relações entre recursos.

Capacidades gerais

O mapeamento de recursos fornece vários recursos que aprimoram a investigação e a prevenção de incidentes:

- Visualização interativa — Explore a topologia do seu aplicativo por meio de um gráfico interativo no Operator Web App. Você pode ampliar e navegar pela topologia para entender as relações complexas entre os recursos. Você também pode usar o Chat para consultar informações de topologia usando linguagem natural, como “Mostrar todas as funções do Lambda conectadas a esta tabela do DynamoDB” ou “Quais recursos são afetados por esse alarme?”.
- Investigação contextual — Durante as investigações de incidentes, o AWS DevOps agente é auxiliado pela topologia de recursos para identificar os componentes afetados, entender o raio de explosão e rastrear o caminho do impacto em seus sistemas.
- Análise da causa raiz — A compreensão detalhada das relações de recursos ajuda a identificar a origem dos problemas, mesmo em sistemas distribuídos complexos com muitas interdependências.
- Avaliação de impacto — Ao analisar incidentes, o agente pode determinar melhor quais serviços posteriores podem ser afetados identificando cadeias de dependência na topologia.
- Recomendações preventivas — O agente usa insights de topologia para fazer recomendações direcionadas para melhorias de resiliência, sugerindo mudanças que terão o impacto mais significativo na estabilidade do sistema.

Visualizações de topologia

A visualização da topologia na página Topologia do Operator Web App oferece vários níveis de detalhes:

- Aprendido — A visualização padrão, gerada a partir da habilidade Agent Space Understanding. Exibe um resumo estruturado de sua infraestrutura organizado por serviços lógicos e caminhos de solicitação.
- Sistema — Mostra os limites de alto nível da conta e da região.

- Contêiner — Exibe pilhas de implantação como CloudFormation pilhas que contêm recursos relacionados.
- Componentes — Mostra os componentes individuais dentro dos contêineres e seus relacionamentos.
- Todos os recursos — Mostra a visão completa de todos os recursos descobertos e seus relacionamentos.

Descoberta de recursos

Os recursos são descobertos por meio de dois métodos:

- CloudFormation pilhas — O agente lista todas as CloudFormation pilhas e seus recursos na AWS conta principal e em todas as contas secundárias conectadas. Isso é compatível com qualquer infrastructure-as-code ferramenta usada CloudFormation para implantação, incluindo o AWS Cloud Development Kit (AWS CDK).
- Resource Explorer — Para recursos não implantados CloudFormation, os recursos marcados são descobertos no AWS Resource Explorer. A AWS conta de destino deve ter o Resource Explorer ativado. Isso é útil para identificar limites de aplicativos para recursos implantados por meio do AWS Management Console, do AWS serviço APIs ou de outras infrastructure-as-code estruturas.

Escopo da investigação além da topologia

Embora a topologia do aplicativo forneça um contexto importante durante as investigações, o AWS DevOps Agente não se limita a investigar somente os recursos mostrados na topologia. O agente pode usar fontes de dados adicionais, como AWS serviços APIs ou ferramentas de observabilidade conectadas, para investigar recursos que não estão na topologia do aplicativo.

Para limitar os recursos aos quais o agente tem acesso, restrinja a política da função atribuída ao agente para acessar recursos entre contas. Para obter mais informações, consulte the section called “Limitando o acesso do agente em um AWS Conta”.

Topologia e a habilidade de compreensão do espaço do agente

O gráfico de topologia alimenta a habilidade aprendida do Agent Space Understanding, que codifica um resumo estruturado de sua infraestrutura para uso durante investigações. Quando a descoberta da topologia é concluída para um novo espaço de agente, o sistema gera automaticamente a

habilidade de compreensão do espaço do agente. Para obter mais informações sobre as habilidades aprendidas, consultethe section called “Habilidades aprendidas”.

DevOps Habilidades do agente

AWS DevOps As habilidades do agente são conjuntos de instruções modulares que ampliam as capacidades do agente com conhecimento especializado do domínio e metodologias de investigação adaptadas à sua infraestrutura e fluxos de trabalho operacionais.

O que são habilidades

As habilidades são diretórios independentes contendo instruções do Markdown que fornecem recursos especializados ao Agente. AWS DevOps AWS DevOps O Agent suporta um subconjunto da especificação Agent Skills — um padrão aberto para empacotar instruções e recursos do agente — suportando somente documentos não executáveis: instruções Markdown, PDFs, imagens e arquivos de dados.

Cada habilidade requer um SKILL.md arquivo contendo as instruções que você deseja fornecer ao seu AWS DevOps agente. Além do SKILL.md arquivo necessário, as habilidades podem incluir:

- Fluxos de trabalho de investigação para cenários ou tipos de infraestrutura específicos.
- Materiais de referência, incluindo padrões de arquitetura e procedimentos operacionais.
- Segmentação por tipo de agente — As habilidades podem ser direcionadas a tipos específicos de agentes (genéricos, triagem de incidentes On-demand, RCA de incidentes, mitigação de incidentes, avaliação) para reduzir o consumo de contexto e melhorar o foco do agente.

Por que usar Skills

As habilidades transformam o AWS DevOps agente de um assistente de uso geral em um especialista para sua infraestrutura e fluxos de trabalho operacionais. Diferentemente das instruções únicas fornecidas em uma mensagem de bate-papo, as habilidades são recursos reutilizáveis que são carregados automaticamente quando relevantes às tarefas executadas pelo Agente. AWS DevOps

Principais benefícios:

- Especialize seu agente — Personalize o AWS DevOps agente com procedimentos de investigação, melhores práticas e conhecimento organizacional específicos para sua infraestrutura e padrões operacionais.
- Reduza a repetição — Crie fluxos de trabalho de investigação uma vez e o AWS DevOps agente os usará automaticamente em todas as investigações relevantes, eliminando a necessidade de fornecer a mesma orientação repetidamente.
- Recursos do Compose — Combine várias habilidades para criar fluxos de trabalho de investigação de ponta a ponta. AWS DevOps O agente lê várias habilidades durante a execução, como uma habilidade para recuperar implantações do seu CI/CD pipeline personalizado e uma habilidade para pesquisar seus repositórios de código.
- Amplifique as ferramentas personalizadas — Crie habilidades que orientem o AWS DevOps Agente a usar suas ferramentas personalizadas de servidor MCP de forma eficaz. As habilidades podem documentar quando invocar ferramentas específicas, quais parâmetros usar em diferentes cenários e como interpretar os resultados para realizar fluxos de trabalho específicos para sua infraestrutura.

Como as habilidades funcionam

Quando o AWS DevOps agente encontra uma tarefa relevante, ele carrega as habilidades apropriadas e segue as instruções para orientar sua investigação. Por exemplo, uma habilidade de “Investigação de desempenho de banco de dados” pode incluir procedimentos passo a passo para analisar problemas de limitação do RDS, permitindo que o agente verifique sistematicamente o status do alarme, analise métricas de conexão e identifique consultas lentas.

Estrutura de habilidades

Uma habilidade é organizada como um diretório contendo:

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

SKILL.md

Esse `SKILL.md` é o único arquivo obrigatório. Ele contém as principais instruções escritas no formato Markdown. Esse arquivo deve:

- Descreva quando e como usar a habilidade.
- Forneça procedimentos de investigação passo a passo.
- Inclua árvores de decisão para diferentes cenários.
- Documente os resultados esperados e os critérios de sucesso.

Matéria frontal

Frontmatter é o bloco de metadados na parte superior de um `SKILL.md` arquivo, entre delimitadores. `---` Ele contém os `description` campos `name` e que o AWS DevOps Agente usa para determinar quando ativar a habilidade durante uma investigação ou tarefa.

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

nome — Um identificador exclusivo para a habilidade. Use somente letras minúsculas, números e hífens (máximo de 64 caracteres). Não deve começar nem terminar com um hífen.

descrição — Uma explicação detalhada de quando e por que o AWS DevOps agente deve usar essa habilidade. AWS DevOps O agente avalia esse campo para decidir se a habilidade é relevante para a tarefa atual. Uma descrição vaga ou ausente pode fazer com que o agente ignore completamente a habilidade, mesmo que as instruções estejam bem escritas.

Importante — Escreva a descrição do ponto de vista do agente. Inclua os cenários, serviços, tipos de erros ou sintomas específicos que devem acionar a habilidade. Por exemplo, “Use essa habilidade ao investigar a latência do banco de dados, erros de conexão ou tempos limite de consulta para instâncias do Amazon RDS” é mais eficaz do que “habilidade RDS”.

Quando você cria uma habilidade na interface do usuário, o sistema gera o frontmatter automaticamente a partir do nome e da descrição que você fornece. As habilidades enviadas como arquivos zip devem incluir o frontmatter no `SKILL.md` arquivo.

Exemplo: habilidade completa

O exemplo a seguir mostra uma habilidade completa e bem formada para investigar problemas de desempenho do RDS. Ele demonstra a estrutura do diretório, o SKILL.md front-matter, os procedimentos de investigação acionáveis e um arquivo de referências suplementar.

Estrutura de diretórios:

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

SKILL.md:

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

# RDS Performance Investigation

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


## Step 1: Check alarm status

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


## Step 2: Analyze connection metrics
```

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


## Step 3: Identify slow queries

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


## Step 4: Summarize findings

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

references/rds-metrics-reference.md:

```
# RDS CloudWatch Metrics Reference


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

Exemplo: habilidade de filtragem de incidentes

As habilidades direcionadas ao tipo de agente de triagem de incidentes podem definir critérios para ignorar incidentes automaticamente. Use isso para filtrar incidentes que não exigem investigação. Quando um novo incidente corresponde aos critérios de ignorar, o AWS DevOps agente o marca como Ignorado. O sistema fornece um motivo que explica por que ele foi filtrado.

O exemplo a seguir mostra uma habilidade que ignora incidentes de baixa prioridade durante a manutenção programada:

SKILL.md:

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

# Skip Scheduled Maintenance

Skip all incidents that meet BOTH of the following criteria:

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

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

Ao criar essa habilidade, selecione Triagem de incidentes como o tipo de agente. Isso garante que a habilidade seja avaliada somente durante a fase de triagem.

Criando habilidades

Antes de criar habilidades, você deve ter um Espaço do Agente. Para obter mais informações, consulte the section called “Criação de um espaço de agente”.

Você pode criar habilidades de duas maneiras, dependendo das preferências do fluxo de trabalho e da complexidade das habilidades:

Criando uma habilidade na interface

As habilidades criadas no AWS DevOps Agent Operator Web App contêm nome, descrição e instruções em um único SKILL.md arquivo.

Para criar uma habilidade na interface do usuário:

- Navegue até a página de habilidades em seu aplicativo web Agent Space Operator.
- Clique em “Adicionar habilidade”.
- Selecione “Criar habilidade” no modal.

- Preencha o formulário de habilidades:
  - Nome — Somente letras minúsculas, números e hífens (máximo de 64 caracteres). Não deve começar nem terminar com um hífen. Exemplo: `rds-throttling-investigation`
  - Descrição — Breve explicação de quando usar essa habilidade (mínimo de 100 caracteres recomendados, máximo de 1.024 caracteres). Isso ajuda o agente a determinar quando ativar a habilidade.
  - Status — Defina como Ativo (padrão) ou Inativo. Habilidades inativas não são usadas pelo agente.
  - Tipo de agente — Selecione um ou mais tipos de agentes que podem usar essa habilidade. Genérico é selecionado por padrão e disponibiliza a habilidade para todos os tipos de agentes. Para atingir agentes específicos, desmarque Genérico e escolha entre: Triagem de incidentes On-demand, RCA de incidentes, Mitigação de incidentes ou Avaliação.
  - Instruções — Step-by-step procedimentos no formato Markdown. Seja específico e acionável.
- Clique em “Criar” para salvar a habilidade.

O sistema gera automaticamente um SKILL.md arquivo com a estrutura de frontmatter adequada.

Para editar uma habilidade criada na interface do usuário:

- Navegue até a habilidade na lista de habilidades e clique na habilidade para abri-la.
- Clique em Edit.
- Modifique o nome, a descrição ou as instruções.
- Clique em Salvar para atualizar a habilidade.

Fazendo o upload de uma habilidade

As habilidades enviadas como arquivos zip contêm um SKILL.md arquivo e recursos adicionais, como materiais de referência ou ativos.

Estrutura 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
```

SKILL.md requisitos de frontmatter:

As habilidades enviadas como arquivos zip devem incluir o frontmatter em SKILL.md with `name` e `description` os campos. AWS DevOps O agente usa esses campos para determinar quando ativar a habilidade. Para obter detalhes sobre como escrever um frontmatter eficaz, consulte a seção Frontmatter anteriormente neste tópico.

```
---
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 criar uma habilidade via upload zip:

- Crie um diretório com seus arquivos de habilidades seguindo a estrutura acima.
- Certifique-se de SKILL.md incluir o conteúdo inicial adequado (nome e descrição).
- Comprima o diretório em um arquivo.zip.
- Navegue até a página de habilidades em seu aplicativo web Agent Space Operator.
- Clique em “Adicionar habilidade”.
- Selecione “Habilidade de upload” no modal.
- Arraste e solte seu arquivo.zip ou clique para navegar (somente arquivos ZIP, máximo de 6 MB).
- Selecione um ou mais tipos de agentes que possam usar essa habilidade (Genérico é selecionado por padrão e se aplica a todos os tipos de agentes; desmarque especificamente para alvo On-demand, triagem de incidentes, RCA de incidentes, mitigação de incidentes ou avaliação).
- Analise os requisitos do arquivo zip e os resultados da validação.

- Clique em “Carregar” para adicionar a habilidade ao seu Espaço do Agente.

Restrições importantes para habilidades enviadas como arquivos zip:

- Atualmente, não há suporte para scripts — as habilidades que contêm scripts no `scripts/` diretório serão rejeitadas durante o upload. A execução de scripts será habilitada em uma versão futura quando os agentes tiverem acesso a um ambiente de codificação seguro.
- Limite de tamanho — O tamanho total do arquivo zip não deve exceder 6 MB (incluindo todos os arquivos).
- SKILL.md obrigatório — O arquivo zip deve conter um SKILL.md arquivo com frontmatter válido.

Melhores práticas para habilidades de nomenclatura:

Use nomes claros e descritivos, como “rds-throttling-investigation”, em vez de nomes genéricos. Um bom nome de habilidade reflete o cenário ou serviço específico abordado, facilitando a identificação rápida da habilidade certa.

Gerenciar habilidades

AWS DevOps O Agent fornece recursos abrangentes de gerenciamento de habilidades por meio do Operator Web App:

Listando habilidades — Veja todas as habilidades em seu Espaço do Agente. A página Habilidades exibe o nome da habilidade, o status ativo ou inativo, a data de criação, a data da última atualização e as ações disponíveis.

Habilidades de visualização — Clique em qualquer habilidade para ver sua visualização detalhada. As habilidades criadas na interface exibem conteúdo editável em que você pode modificar o nome, a descrição ou as instruções diretamente na interface do usuário e clicar em “Salvar” para atualizar. As habilidades enviadas como arquivos zip exibem uma árvore de arquivos SKILL.md e quaisquer diretórios adicionais, como references/ e assets/. Clique nos arquivos na árvore para ver seu conteúdo no modo somente leitura.

Seleção de agentes para uma habilidade — Configure quais tipos de agentes podem usar cada habilidade ao criá-la ou editá-la. No menu suspenso Tipo de agente, selecione um ou mais tipos de agentes usando as caixas de seleção: Genérico (padrão — aplica-se a todos os tipos de agentes), On-demand(consultas conversacionais), Triagem de incidentes (avaliação inicial do incidente), RCA

do incidente (análise da causa raiz), Mitigação de incidentes (resposta automática a incidentes) ou Avaliação (recomendações proativas). Genérico é selecionado por padrão e disponibiliza a habilidade para todos os tipos de agentes. As habilidades direcionadas a agentes específicos reduzem o consumo de contexto e melhoram o foco do agente.

Ativando e desativando habilidades — Desative temporariamente as habilidades sem excluí-las usando o botão de alternância. Active/Inactive Abra a visualização de detalhes da habilidade e alterne a opção para “Inativa” para evitar que o agente a carregue para novas investigações, preservando todo o conteúdo e as configurações. In-progress as investigações continuam usando a habilidade. Volte para “Ativo” para tornar a habilidade imediatamente disponível novamente.

Atualização de habilidades — modifique as habilidades existentes com base em como elas foram criadas. Para habilidades criadas na interface do usuário, clique em “Editar” na visualização de detalhes da habilidade, modifique o nome, a descrição ou as instruções e clique em “Salvar” para atualizar. Para habilidades enviadas como arquivos zip, modifique os arquivos localmente, crie um novo arquivo zip e faça upload de uma nova versão.

Excluindo habilidades — Remova permanentemente as habilidades do seu Espaço do Agente. Abra a visualização da lista de habilidades, clique no menu de mais opções (✓) e selecione “Excluir”, revise o aviso sobre exclusão permanente, digite o nome da habilidade para confirmar e clique em “Excluir habilidade”. A exclusão não pode ser desfeita. In-progress as investigações podem ser afetadas se tentarem carregar a habilidade excluída. Para habilidades enviadas como arquivos zip, baixe o arquivo zip antes de excluí-lo como backup. Considere desativar a habilidade em vez de excluí-la se precisar dela novamente.

Migrando dos Runbooks

Os Runbooks existentes são migrados automaticamente para o Skills sem a necessidade de ação do cliente. Quando seu Agent Space faz a transição para o modelo Skills, todos os Runbooks são convertidos em Skills e aparecem na sua interface de Skills. Após a migração, você pode:

- Analise as habilidades migradas — Verifique se a migração automática converteu corretamente seus Runbooks.
- Atualize conforme necessário — edite as habilidades diretamente na interface do usuário para refinar as instruções, atualizar as descrições ou configurar a segmentação por tipo de agente.
- Expanda com referências — Para habilidades que se beneficiariam de materiais de referência adicionais ou diagramas de arquitetura, recrie-as como habilidades de upload de zip com um diretório references/ ou assets/.

- Crie novas habilidades — Adicione novas habilidades para fluxos de trabalho de investigação não cobertos anteriormente pelos Runbooks.

Entre em contato com o AWS Support se você encontrar algum problema com as habilidades migradas automaticamente ou precisar de ajuda com atualizações pós-migração.

Habilidades aprendidas

O que são habilidades aprendidas?

As habilidades aprendidas são arquivos de conhecimento estruturados que o DevOps Agente gera a partir dos dados do seu Espaço do Agente. Cada habilidade aprendida codifica um tipo específico de conhecimento que o AWS DevOps agente usa ao realizar tarefas. No lançamento, duas habilidades aprendidas estão disponíveis: compreensão do espaço do agente e melhores práticas de uso de ferramentas.

Compreensão do Agent Space

A habilidade Agent Space Understanding (`understanding-agent-space`) analisa suas contas de nuvem conectadas, repositórios de código e integrações de telemetria para criar um mapa dos recursos e relacionamentos em um Agent Space.

A habilidade produz um `SKILL.md` arquivo principal e um conjunto de arquivos de referência. O arquivo principal contém uma visão geral do sistema em linguagem simples com os principais conceitos de domínio, os ambientes de implantação (pares de AWS contas e regiões, assinaturas e regiões do Azure etc.), um diagrama de arquitetura em nível de contêiner que mostra como os serviços lógicos se conectam, os caminhos de solicitação que são centrais para seu aplicativo com os componentes que eles atravessam e um mapeamento de repositórios de código para contêineres.

Cada contêiner lógico recebe um arquivo de referência dedicado descrevendo seus componentes internos (computação, dados, mensagens, rede e outros) com tipos de recursos e identificadores físicos ARNs, como nomes de tabelas e filas. URLs O arquivo de referência também captura a cobertura de observabilidade, incluindo os alarmes, painéis e monitores vinculados a cada componente. Ele também mapeia cada componente para seus repositórios de código, pacotes e infrastructure-as-code definições associados, fornecendo uma cadeia de rastreabilidade completa, do código-fonte aos recursos implantados.

Cada caminho crítico de solicitação recebe um arquivo de referência dedicado que descreve o fluxo completo de end-to-end solicitações na granularidade do componente, desde o ponto de entrada até

cada serviço intermediário, armazenamento de dados e dependência externa. O arquivo inclui um diagrama de fluxo sequenciado que mostra a ordem das operações e os mecanismos de interação entre os componentes, junto com a responsabilidade de cada participante. Ele também cataloga os sinais de observabilidade relevantes para o caminho: padrões de grupos de registros para cada salto, métricas principais (latência, taxas de erro, limitação, cotas de tokens) com seus nomes e dimensões de alarme e extensões de rastreamento distribuídas que podem ser correlacionadas entre serviços e contas.

Práticas recomendadas de uso de ferramentas

A habilidade Tool Use Best Practices analisa os usos anteriores de ferramentas de investigação para extrair padrões de uso efetivos, modos de falha comuns e orientação de parâmetros. Isso ajuda o DevOps agente a evitar armadilhas conhecidas e a realizar investigações com menos etapas desperdiçadas. A habilidade produz um arquivo principal e um conjunto de arquivos de referência por ferramenta. O arquivo principal serve como um índice de roteamento que lista cada ferramenta com os cenários de investigação que ela suporta e vincula ao arquivo de referência correspondente.

Cada arquivo de referência por ferramenta pode incluir até três seções:

- Práticas recomendadas — técnicas orientadas por investigação extraídas do uso bem-sucedido da ferramenta, como modelos de consulta do CloudWatch Logs Insights, namespaces e dimensões de métricas específicas do ambiente e filtros de origem de eventos. CloudTrail Cada entrada é organizada em torno de um cenário de investigação e inclui valores de parâmetros concretos e exemplos observados em investigações anteriores.
- Erros comuns — Modos de falha recorrentes e suas correções. Cada entrada descreve uma condição de erro específica, como consultar uma conta inacessível ou criar uma consulta de agregação malformada, e fornece uma ação corretiva para que o agente possa evitar ou se recuperar do erro sem desperdiçar as etapas de investigação.
- Gerenciamento de resultados — Orientação para chamadas de ferramentas que tendem a retornar respostas grandes. Cada entrada descreve uma mudança de parâmetro ou estratégia de processamento que reduz o tamanho da saída e preserva o valor do diagnóstico.

Quando o acesso ativo à infraestrutura está disponível, a habilidade valida os padrões em relação ao seu ambiente antes de incluí-los. Os padrões confirmados são declarados com confiança, os padrões não confirmados usam uma linguagem cautelosa e os padrões refutados são excluídos. Isso mantém a habilidade alinhada com o estado atual da sua infraestrutura.

Gerenciando habilidades aprendidas

Atualizações — O DevOps Agente gera e atualiza automaticamente as habilidades aprendidas com base na atividade em seu Espaço do Agente. A seguir, descrevemos quando cada habilidade é atualizada.

O DevOps agente gera uma habilidade atualizada de melhores práticas de uso de ferramentas a cada 30 investigações.

A habilidade de Compreensão do Espaço do Agente é gerada pelo agente de aprendizado, que é executada sempre que você adiciona, atualiza ou remove um recurso ou integração do Espaço do Agente.

Para regenerar as habilidades aprendidas manualmente, escolha o botão Regenerar na página Topologia no aplicativo do operador ou converse com o agente e peça que ele atualize as habilidades aprendidas.

Desativação — As habilidades aprendidas estão ativas por padrão. Quando ativos, o DevOps Agente os carrega no início de cada tarefa do DevOps Agente. Para impedir que uma habilidade aprendida seja aplicada, desative-a no visualizador de habilidades no aplicativo do operador. Desativar uma habilidade não a exclui. A habilidade é mantida e pode ser reativada a qualquer momento. Quando uma habilidade é desativada, o DevOps Agente opera sem o conhecimento dessa habilidade.

Visualização de topologia — A página de topologia no aplicativo web do seu Agent Space usa a habilidade de compreensão do Agent Space para exibir visualmente seu ambiente do Agent Space como contêineres e componentes lógicos. Clique em qualquer contêiner para ver seus componentes, identificadores de recursos e telemetria.

Instruções do agente

Use as instruções do agente para fornecer orientações sempre ativas que o AWS DevOps Agente aplica a todas as sessões. Uma sessão é uma única conversa ou investigação com um agente. Na página Agentes do seu Agent Space Operator Web App, você pode definir instruções globais que se aplicam a todos os agentes ou definir instruções para um agente gerenciado específico, como Chat ou triagem de incidentes. Essas instruções são armazenadas como um AGENTS.md arquivo. the section called “DevOps Habilidades do agente”Diferentemente das que são carregadas sob demanda quando o agente combina uma descrição de habilidade com a tarefa atual, as instruções do agente estão sempre presentes desde o início de cada sessão, independentemente do que o agente esteja trabalhando.

Quais são as instruções do agente

As instruções do agente fornecem orientação incondicional e sempre ativa para seus agentes. No início de cada sessão, o serviço do agente recupera as instruções configuradas para o seu Espaço do Agente e injeta o conteúdo diretamente no prompt do sistema do agente. O agente não decide se quer carregá-los; eles estão sempre presentes.

Cada sessão do agente recebe instruções das instruções globais e das instruções relevantes específicas do agente, por exemplo, o Chat.

As instruções do agente são armazenadas como AGENTS.md arquivos e the section called “DevOps Habilidades do agente” diferem de várias maneiras importantes:

Aspecto	Habilidade	Instruções do agente (AGENTS.md)
Nome e descrição	Obrigatório	Não aplicável
Formato de conteúdo	Pacote Markdown ou ZIP	Somente Markdown
Arquivos de recursos	Compatível	Não compatível
Injeção de contexto	Sob demanda (o agente decide por meio da correspondência da descrição da habilidade)	Sempre (incondicional, em todas as sessões)
Exclusividade	Vários por espaço de agente	Um por agente (um para instruções globais, um por agente gerenciado)

As instruções do agente não têm nome nem campo de descrição. O AGENTS.md arquivo subjacente contém apenas markdown sem frontmatter, sem suporte a pacotes ZIP e sem arquivos de recursos.

Por que usar as instruções do agente

As instruções do agente oferecem uma maneira confiável de garantir que determinadas orientações estejam sempre contextualizadas, sem depender das decisões de carga de habilidades do agente.

Principais benefícios:

- Previsibilidade: as instruções estão sempre presentes, independentemente da tarefa em que o agente está trabalhando. Nenhuma correspondência de descrição é necessária e o agente não pode ignorar o conteúdo.
- Cobertura garantida: ao contrário do Skills, que o agente pode ou não carregar dependendo da relevância da tarefa, as instruções do agente são sempre injetadas no início de cada sessão.
- Políticas permanentes: use as instruções do agente para políticas operacionais permanentes, diretrizes de segurança, padrões de codificação ou qualquer orientação que deva ser aplicada a todas as sessões, sem exceção.
- Escopo direcionado: você pode aplicar instruções a todos os tipos de agentes ao mesmo tempo usando instruções globais ou restringir as instruções a um tipo específico de agente quando a orientação for relevante somente para o trabalho desse agente.

Como as instruções do agente funcionam

Quando uma sessão é iniciada, o serviço do agente recupera as instruções configuradas para o seu Espaço do Agente e injeta seu conteúdo no prompt do sistema do agente antes do início da sessão. Isso acontece automaticamente em cada sessão. O agente não avalia se deve carregá-los; ele sempre injeta o conteúdo.

Cada nova sessão carrega as instruções atualizadas na inicialização. Se você atualizar suas instruções, a alteração entrará em vigor imediatamente para as sessões iniciadas após o salvamento. As sessões que já estão em andamento continuam usando o conteúdo que foi carregado quando foram iniciadas.

O escopo determina quais instruções uma sessão recebe. As instruções globais se aplicam a todos os tipos de agentes em seu Espaço do Agente, portanto, todas as sessões as recebem. Agent-specific as instruções se aplicam somente às sessões desse tipo específico de agente. Uma sessão recebe instruções das instruções globais e das instruções relevantes específicas do agente.

Escopo do tipo de agente

O escopo controla quais sessões do agente recebem um determinado conjunto de instruções. Há duas opções de escopo:

- Instruções globais: se aplicam a todos os tipos de agentes em seu Espaço do Agente. Cada sessão do agente recebe esse conteúdo.

- Agent-specific: aplica-se somente às sessões do tipo de agente selecionado.

Os agentes gerenciados disponíveis para instruções específicas do agente são:

- Chat - Ad-hoc perguntas e solicitações durante as sessões de bate-papo.
- Triagem de incidentes — filtragem de alarmes, classificação de gravidade e escopo inicial.
- Incidente RCA - Análise da causa raiz com coleta e validação de evidências.
- Mitigação de incidentes — recomendações de Short-term remediação e correção de longo prazo.
- Avaliação - Pontuação de desempenho do agente e verificações de conformidade com as políticas.

Orientação sobre o tamanho do conteúdo

Toda vez que você inicia uma conversa ou investigação, o agente lê todas as instruções antes de fazer qualquer outra coisa. O agente tem uma quantidade fixa de memória de trabalho por sessão, e suas instruções usam parte dela. Quanto maior o arquivo, menos espaço resta para suas perguntas, investigações, os registros que o agente lê e seu próprio raciocínio. Instruções mais curtas e focadas dão ao agente mais capacidade de resolver seu problema.

- Limite rígido: 25 KB
- Tamanho recomendado: 120 linhas (recomendado para a maioria das configurações)

Concentre suas instruções nas orientações que devem estar presentes em todas as sessões. Para procedimentos de investigação especializados que se aplicam somente a tarefas específicas, considere usar the section called “DevOps Habilidades do agente” em vez disso.

Exemplo

O exemplo a seguir mostra instruções bem elaboradas do agente com orientações de investigação, padrões de formatação de respostas e requisitos de segurança que se aplicam a todas as sessões do agente.

```
# Agent Instructions

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

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

## Security
- Never log, display, or suggest storing credentials or secrets in plaintext.
- When recommending IAM changes, follow least-privilege principles.
```

Definindo as instruções do agente

Antes de definir as instruções do agente, você deve ter um Espaço do Agente. Para obter mais informações, consulte the section called “Criação de um espaço de agente”.

Cada agente tem exatamente um conjunto de instruções. Quando você salva um novo conteúdo, ele sobrescreve o conteúdo existente para esse agente.

Para definir instruções globais (aplica-se a todos os agentes):

1. Navegue até a página Agentes em seu aplicativo Web Agent Space Operator.
2. Escolha Exibir ao lado de Instruções globais.
3. Insira suas instruções de marcação no editor.
4. Escolha Salvar.

Para definir instruções para um agente específico:

1. Navegue até a página Agentes em seu aplicativo Web Agent Space Operator.
2. Em Agentes gerenciados, escolha Exibir ao lado do agente que você deseja configurar: Chat, Triagem de incidentes, RCA de incidentes, Mitigação de incidentes ou Avaliação.
3. Insira suas instruções de marcação no editor.
4. Escolha Salvar.

Gerenciando as instruções do agente

AWS DevOps O agente fornece recursos de gerenciamento para as instruções do agente por meio do Operator Web App.

Instruções de visualização: navegue até a página Agentes e escolha Exibir ao lado de Instruções globais ou do agente gerenciado específico. O editor mostra o conteúdo atual. Use a guia Visualizar para ver a redução renderizada ou a guia Código para ver a redução bruta.

Instruções de edição: abra o agente conforme descrito acima, modifique o conteúdo no editor e escolha Salvar.

Carregar instruções de um arquivo: abra o agente e escolha o botão Carregar no editor para carregar um arquivo markdown do seu computador.

Instruções de download: abra o agente e escolha o botão Download no editor para baixar o conteúdo atual como um arquivo.

Instruções de exclusão: abra o agente, escolha o botão Excluir no editor e confirme a exclusão. Esta ação não pode ser desfeita. Considere baixar o conteúdo primeiro se precisar dele novamente.

Regiões aceitas

Este tópico descreve as AWS regiões em que você pode usar o AWS DevOps Agent. Para obter mais informações sobre AWS regiões, consulte Especificar quais AWS regiões sua conta pode usar no Guia de referência de gerenciamento de AWS contas.

Monitoramento de recursos entre regiões

AWS DevOps O agente pode monitorar e investigar recursos em AWS contas localizadas em qualquer AWS região, independentemente da região suportada na qual você cria seu Espaço do Agente. Quando você associa uma AWS conta a um Espaço do Agente, o agente descobre e mapeia recursos em todas as regiões dessa conta. Isso significa que você não precisa de um espaço de agente em todas as regiões em que suas cargas de trabalho são executadas.

Escolha uma região com suporte com base na residência de dados de sua preferência, na proximidade com sua equipe de operações ou nos requisitos organizacionais.

Regiões aceitas

AWS DevOps O agente está disponível nas seguintes AWS regiões.

Nome da região	Código da região	Link do console
Leste dos EUA (Norte da Virgínia)	us-east-1	Abra o console
Oeste dos EUA (Oregon)	us-west-2	Abra o console
Ásia-Pacífico (Sydney)	ap-southeast-2	Abra o console
Ásia-Pacífico (Tóquio)	ap-northeast-1	Abra o console
Europa (Frankfurt)	eu-central-1	Abra o console
Europa (Irlanda)	eu-west-1	Abra o console

Service endpoints

Nome da região	Código da região	Endpoint	Protocolo
Leste dos EUA (Norte da Virgínia)	us-east-1	aidevops.us-east-1.amazonaws.com	HTTPS
Oeste dos EUA (Oregon)	us-west-2	aidevops.us-west-2.amazonaws.com	HTTPS
Ásia-Pacífico (Sydney)	ap-southeast-2	aidevops.ap-southeast-2.amazonaws.com	HTTPS
Ásia-Pacífico (Tóquio)	ap-northeast-1	aidevops.ap-northeast-1.amazonaws.com	HTTPS
Europa (Frankfurt)	eu-central-1	aidevops.eu-central-1.amazonaws.com	HTTPS
Europa (Irlanda)	eu-west-1	aidevops.eu-west-1.amazonaws.com	HTTPS

Considerações

- Seleção da região do espaço do agente — Um espaço do agente e seus dados (investigações,

topologia, recomendações) são armazenadas na região em que você as criou. Escolha uma região que atenda aos seus requisitos de residência de dados.

- Monitoramento entre regiões — Recursos em AWS contas associadas a um agente

O espaço é monitorado independentemente da região em que esses recursos estão implantados. Você não precisa criar espaços de agente separados em cada região em que suas cargas de trabalho são executadas.

- Integrações de terceiros — Conexões com CI/CD provedores (GitHub, GitLab),

as ferramentas de observabilidade (Dynatrace, Datadog, New Relic, Splunk) e servidores MCP são configurados por espaço de agente e não dependem da região.

Começando com o AWS DevOps Agent

Neste guia de introdução, você criará um Agent Space básico, configurará permissões mínimas e conduzirá sua primeira investigação baseada em IA.

Tópicos:

Criação de um espaço de agente

Um Espaço do Agente define as ferramentas e a infraestrutura às AWS DevOps quais o Agente tem acesso. Este guia orienta você na criação de um Espaço do Agente, na configuração do acesso à conta principal e na ativação do DevOps Agent Web App. Consulte “O que é um espaço do agente” para saber mais sobre o conceito do espaço do agente.

Criação de um espaço de agente

Acesse o console do AWS DevOps agente

1. Faça login no console AWS de gerenciamento
2. Navegue até o console do AWS DevOps agente

Nomeie o Agent Space

1. Clique em Criar espaço do agente

Na seção Detalhes do Espaço do Agente, forneça:

1. No campo Nome, insira um nome para o seu Espaço do Agente

2. (Opcional) No campo Descrição, adicione detalhes sobre a finalidade do Espaço do Agente
3. (Opcional) No menu suspenso Idioma de resposta do agente, selecione o idioma que o agente usa ao gerar respostas, descobertas e resultados da investigação. As opções incluem: indonésio, chinês (Simplified/PRC), Chinese (Traditional/Taiwan), inglês (Reino Unido), francês (França), alemão (Alemanha), italiano (Itália), japonês (Japão), coreano (Coreia), português (Brasil), espanhol (América Latina), turco (Turquia), árabe (Arábia Saudita), tailandês (Tailândia) e vietnamita (Vietnã) Vietnã). Se nenhum idioma for selecionado, o agente responderá no idioma da entrada. Essa configuração também é usada para determinar o idioma dos casos de AWS Support criados por meio do recurso Ask for human support.

Configurando o acesso à conta principal

Na seção Conceder acesso aos AWS recursos do Agent Space, você configurará uma função do IAM para conceder ao Agent Space acesso à AWS conta principal. A conta principal é a AWS conta na qual você cria seu Espaço do Agente. AWS DevOps O agente exige uma função do IAM para descobrir e acessar AWS recursos nessa conta durante as investigações.

Escolha um método de configuração de função. Selecione uma das opções a seguir:

Opção 1: criar automaticamente uma nova função de AWS DevOps agente (recomendado)

Essa opção cria automaticamente uma função com as permissões apropriadas para que o AWS DevOps Agente investigue recursos em sua conta.

> **Note**
>
> Você precisa ter permissões do IAM para criar novas funções para usar essa opção.

1. Selecione Criar automaticamente uma nova função de AWS DevOps agente
2. (Opcional) Atualize o nome da função do Agent Space a ser criada

Opção 2: atribuir uma função existente

Use essa opção quando outro administrador tiver criado anteriormente uma função específica para o AWS DevOps Agente.

1. Selecione Atribuir uma função existente

2. No menu suspenso, selecione uma função existente que tenha as permissões apropriadas

Opção 3: criar uma nova função de AWS DevOps agente usando um modelo de política

Use essa opção quando precisar limitar os serviços e recursos que o agente pode acessar na conta principal.

1. Selecione Criar uma nova função de AWS DevOps agente usando um modelo de política
2. Siga as instruções para criar a política de confiança e a política embutida da nova função.

Ativando o Agent Space Web App

O aplicativo Web é onde a equipe interage com o AWS DevOps Agente para investigações de incidentes e revisão de recomendações. Consulte Arquitetura do console do AWS DevOps agente [link] para saber mais. Quando ativado, os usuários podem acessar o Agent Space Web App por meio de um link de autenticação do IAM no AWS Management Console.

Selecione uma das opções a seguir:

Opção 1: criar automaticamente uma nova função de AWS DevOps agente (recomendado)

Essa opção cria automaticamente uma função com as permissões apropriadas para acessar o DevOps Agent Web App.

> **Note**
>
> Você precisa ter permissões do IAM para criar novas funções para usar essa opção.

1. Selecione Criar automaticamente uma nova função de AWS DevOps agente
2. Revise as permissões que serão concedidas à função

Opção 2: atribuir uma função existente

Use essa opção quando outro administrador tiver criado anteriormente uma função de operador.

1. Selecione Atribuir uma função existente
2. No menu suspenso, selecione uma função existente que tenha as permissões apropriadas

Opção 3: criar uma nova função de AWS DevOps agente usando um modelo de política

Use essa opção quando precisar personalizar as permissões para acesso ao aplicativo web.

1. Selecione Criar uma nova função de AWS DevOps agente usando um modelo de política
2. Siga as instruções para criar a política de confiança e a política embutida da nova função.

Adicionar tags (opcional)

Você pode adicionar AWS tags ao seu Espaço do Agente durante a criação. As tags são pares de valores-chave que ajudam você a organizar e identificar seus recursos. Você pode adicionar até 50 tags por espaço do agente. Para adicionar tags, expanda a seção Tags na página Criar espaço do agente e clique em Adicionar nova tag.

Criação completa do espaço do agente

Depois que todas as seções estiverem preenchidas, clique em Criar

Verificando sua configuração do Agent Space

Depois de configurado, o botão de acesso do operador aparecerá na página de detalhes do Espaço do agente. Clicar nele abrirá o aplicativo Web em uma nova guia e será autenticado com êxito.

Próximas etapas

Depois de configurar seu Espaço do Agente, considere estas próximas etapas:

- Adicione contas secundárias se seus aplicativos abrangerem várias AWS contas
- Configure integrações de terceiros, como ferramentas de observabilidade ou sistemas de emissão de bilhetes
- Configurar a autenticação do AWS Identity Center para ambientes de produção
- Explore o mapeamento de recursos do seu aplicativo para ajudar o AWS DevOps Agente a entender sua infraestrutura

AWS DevOps Guia de integração do Agent CLI

Visão geral do

Com o AWS DevOps Agent, você pode monitorar e gerenciar sua AWS infraestrutura. Este guia explica como configurar o AWS DevOps Agente usando a Interface de Linha de AWS Comando (AWS CLI). Você cria funções do IAM, configura um espaço de agente e associa sua AWS conta. Você também ativa o aplicativo do operador e, opcionalmente, conecta integrações de terceiros. Este guia leva aproximadamente 20 minutos para ser concluído.

AWS DevOps O agente está disponível em seis AWS regiões: Leste dos EUA (Norte da Virgínia), Oeste dos EUA (Oregon), Ásia-Pacífico (Sydney), Ásia-Pacífico (Tóquio), Europa (Frankfurt) e Europa (Irlanda). Para obter mais informações sobre as regiões suportadas, consultethe section called “Regiões aceitas”.

Pré-requisitos

Antes de começar, verifique se você tem o seguinte:

- AWS CLI versão 2 instalada e configurada
- Autenticação em sua conta AWS de monitoramento
- Permissões para criar funções de AWS Identity and Access Management (IAM) e anexar políticas
- Uma AWS conta para usar como conta de monitoramento
- Familiaridade com a AWS CLI e a sintaxe JSON

Ao longo deste guia, substitua os seguintes valores de espaço reservado pelos seus próprios:

- `<MONITORING_ACCOUNT_ID>`— Seu ID de AWS conta de 12 dígitos para a conta de monitoramento (primária)
- `<EXTERNAL_ACCOUNT_ID>`— O ID da AWS conta de 12 dígitos da conta secundária a ser monitorada (usado na etapa 4)
- `<REGION>`— O código AWS da região do seu espaço de agente (por exemplo, `us-east-1` ou`eu-central-1`)
- `<AGENT_SPACE_ID>`— O identificador do espaço do agente que é retornado pelo `create-agent-space` comando

Configuração de funções do IAM

1. Crie a função de espaço do DevOps agente

Crie a política de confiança do IAM executando o seguinte 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
```

Crie o perfil do IAM:

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

Salve o ARN da função executando o seguinte comando:

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

Anexe a política AWS gerenciada:

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

Crie e anexe uma política em linha para permitir a criação da função vinculada ao serviço 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. Crie a função IAM do aplicativo do operador

Crie a política de confiança do IAM executando o seguinte 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
```

Crie o perfil do IAM:

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

Salve o ARN da função executando o seguinte comando:

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

Anexe a política do aplicativo AWS gerenciado do operador:

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

Essa política gerenciada concede ao aplicativo do operador permissões para acessar os recursos do espaço do agente. Esses recursos incluem investigações, recomendações, gerenciamento de

conhecimento, bate-papo e integração com o AWS Support. A política define o escopo do acesso ao espaço específico do agente usando a `aws:PrincipalTag/AgentSpaceId` condição. Para obter mais informações sobre a lista completa de ações, consultethe section called “DevOps Permissões do Agent IAM”.

Etapas de integração

1. Crie um espaço para agentes

Execute o comando a seguir para criar um espaço de agente:

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

Opcionalmente, especifique `--kms-key-arn` o uso de uma chave AWS KMS gerenciada pelo cliente para criptografia. Você também pode usar `--tags` para adicionar tags de recursos e `--locale` definir o idioma das respostas do agente.

Salve o `agentSpaceId` da resposta (localizada em`agentSpace.agentSpaceId`).

Para listar seus espaços de agente posteriormente, execute o seguinte comando:

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

2. Associe sua AWS conta

Associe sua AWS conta para ativar a descoberta de topologia. Defina o `accountType` para um dos seguintes valores:

- `monitor`— A conta principal em que o espaço do agente existe. Essa conta hospeda o agente e é usada para descoberta de topologia.
- `source`— Uma conta adicional que o agente monitora. Use esse tipo ao associar contas externas na etapa 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. Ativar o aplicativo do operador

Os fluxos de autenticação podem usar o IAM, o IAM Identity Center (IDC) ou um provedor de identidade externo (IdP). Execute o comando a seguir para habilitar o aplicativo do operador para seu espaço 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 autenticação do IAM Identity Center, use `--auth-flow idc` e forneça`--idc-instance-arn`. Para um provedor de identidade externo, use `--auth-flow idp` e forneça `--issuer-url--idp-client-id`, `--idp-client-secret` e. Para obter mais informações, consulte the section called “Configurando a autenticação do IAM Identity Center” e the section called “Configurando a autenticação do provedor de identidade externo (IdP)”.

> Observação: se você criou anteriormente uma função de aplicativo de operador para outro espaço de agente em sua conta, você pode reutilizar o ARN dessa função.

4. (Opcional) Associar contas de origem adicionais

Para monitorar contas adicionais com o AWS DevOps Agent, crie uma função entre contas do IAM.

Crie a função entre contas na conta externa

Mude para a conta externa e crie a política de confiança. `MONITORING_ACCOUNT_ID`É a conta principal que hospeda o espaço do agente que você configurou na etapa 2. Essa configuração

permite que o serviço do AWS DevOps Agente assuma uma função nas contas de origem secundária em nome da conta de monitoramento.

Execute o comando a seguir para criar a política de confiança:

```
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
```

Crie a função do IAM entre contas:

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

Salve o ARN da função executando o seguinte comando:

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

Anexe a política AWS gerenciada:

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

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

Anexe a política em linha para permitir a criação da função vinculada ao serviço Resource Explorer na conta 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
```

Associar a conta externa

Volte para sua conta de monitoramento e execute o seguinte comando para associar a conta 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) Associado GitHub

> Observação: primeiro, você deve se registrar GitHub por meio do console do AWS DevOps agente usando o OAuth fluxo antes de poder associá-lo por meio da CLI.

Para obter instruções sobre como se registrar GitHub por meio do console, consultethe section called "Conexão a CI/CD tubulações".

Liste os serviços registrados:

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

Salve o `<SERVICE_ID>` para ServiceType:. `github`

Depois de se registrar GitHub no console, associe GitHub repositórios executando o seguinte 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-se e associe-se ServiceNow

Primeiro, registre o ServiceNow serviço com OAuth as credenciais:

```
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>
```

Salve o retornado <SERVICE_ID> e associe 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 e associe o Dynatrace

Primeiro, registre o serviço Dynatrace com OAuth as credenciais:

```
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>
```

Salve o retornado `<SERVICE_ID>` e associe o Dynatrace. Os recursos são opcionais. O ambiente especifica a qual ambiente Dynatrace se associar.

```
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>
```

A resposta inclui informações de webhook para integração. Você pode usar esse webhook para acionar uma investigação da Dynatrace. Para obter mais informações, consulte the section called “Conectando o Dynatrace”.

8. (Opcional) Registre e associe o Splunk

Primeiro, registre o serviço Splunk com BearerToken as credenciais.

O endpoint usa o seguinte 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>
```

Salve o retornado `<SERVICE_ID>` e associe o 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>
```

A resposta inclui informações de webhook para integração. Você pode usar esse webhook para acionar uma investigação da Splunk. Para obter mais informações, consulte the section called “Conectando o Splunk”.

9. (Opcional) Registre e associe a New Relic

Primeiro, registre o serviço New Relic com as credenciais da chave de API.

Região: US OuEU.

Campos opcionais:`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>
```

Salve o retornado `<SERVICE_ID>` e, em seguida, associe o 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>
```

A resposta inclui informações de webhook para integração. Você pode usar esse webhook para iniciar uma investigação da New Relic. Para obter mais informações, consulte the section called “Conectando a New Relic”.

10. (Opcional) Registre e associe o Datadog

Primeiro, você deve registrar o Datadog por meio do console do AWS DevOps agente usando o OAuth fluxo antes de poder associá-lo por meio da CLI. Para obter mais informações, consulte the section called “Conectando DataDog”.

Liste os serviços registrados:

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

Salve o `<SERVICE_ID>` para ServiceType:. `mcpserverdatadog`

Em seguida, associe o 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>
```

A resposta inclui informações de webhook para integração. Você pode usar esse webhook para acionar uma investigação do Datadog. Para obter mais informações, consulte the section called "Conectando DataDog".

11. (Opcional) Excluir um espaço de agente

A exclusão de um espaço de agente remove todas as associações, configurações e dados de investigação desse espaço de agente. Esta ação não pode ser desfeita.

Para excluir um espaço de agente, execute o seguinte comando:

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

Verificação

Para verificar sua configuração, execute os seguintes 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>
```

Próximas etapas

- Para conectar integrações adicionais, consulteConfigurando recursos para AWS DevOps Agente.
- Para saber mais sobre as habilidades e capacidades dos agentes, consultethe section called "DevOps Habilidades do agente".
- Para entender o aplicativo web do operador, consultethe section called "O que é um DevOps Agent Web App?".

Observações

- Substitua `<AGENT_SPACE_ID><MONITORING_ACCOUNT_ID>,<EXTERNAL_ACCOUNT_ID>,<REGION>`, e assim por diante por seus valores reais.
- Para obter uma lista de regiões compatíveis, consulte the section called "Regiões aceitas".

Criação de um ambiente de teste

Este guia fornece testes práticos para validar a funcionalidade de resposta a incidentes do AWS DevOps Agente usando uma arquitetura de amostra. Use este suplemento se quiser testar o DevOps Agent antes de conectar seus sistemas de produção.

Pré-requisitos

- AWS conta com acesso administrativo
- AWS DevOps Espaço do agente criado e configurado usando o fluxo de função de criação automática do DevOps agente
- Para o teste do EC2: uma VPC existente com pelo menos uma sub-rede na região em que você implantará.

Visão geral de custos e segurança

Proteção de custos

- Teste EC2: GRATUITO (nível AWS gratuito) ou ~ $0,02 por 2 horas
- Teste Lambda: GRATUITO (nível gratuito de 1 milhão requests/month)

- CloudWatch: GRÁTIS (10 alarmes, métricas básicas incluídas)
- Custo total estimado esperado: $0,00 - $0,05 para o teste completo

Características de segurança nesses testes

- Auto-termination: desligamento Built-in automático
- Nível gratuito qualificado: usa os menores tipos de instância
- Escopo limitado: recursos de teste mínimos e isolados
- Limpeza fácil: etapas simples do console para remover tudo
- Sem impacto na produção: ambiente de teste completamente separado

Configure seu AWS conta para teste

> **Important**
>
> Os recursos de infraestrutura precisam ser implantados na AWS conta em que você criou a conta de nuvem primária do seu DevOps Agent Space. A região específica não importa.

1. Faça login no AWS console: https://console.aws.amazon.com
2. Verifique se você está trabalhando na mesma AWS conta em que seu Espaço DevOps do Agente está localizado
3. Você pode usar qualquer região para seus recursos de teste

> **Note**
>
> O mapeamento 1:1 entre a conta principal do seu DevOps agente e os recursos do ambiente de teste que você está criando simplifica a configuração do teste. Você pode estender facilmente seu Espaço do DevOps Agente para incluir contas secundárias e permitir investigações entre contas.

Escolha seu teste

Você pode executar qualquer um dos testes de forma independente ou os dois juntos:

Opção de teste A: teste de capacidade da CPU EC2

Objetivo: validar a capacidade do AWS DevOps agente de detectar e investigar problemas de desempenho do EC2

Tempo estimado: 5 minutos de configuração + 10 minutos de execução automática

Dificuldade: Totalmente automatizado (sem necessidade de etapas manuais)

Opção de teste B: teste de taxa de erro Lambda

Objetivo: validar a capacidade do AWS DevOps agente de detectar e investigar erros da função Lambda

Tempo estimado: 10 minutos de configuração +2 minutos para acionar

Dificuldade: Muito fácil

Opção de teste A: teste de capacidade da CPU EC2

Etapa 1: implantar a CloudFormation pilha para teste do EC2

Usaremos CloudFormation para criar nossos recursos de teste, o que permite ao AWS DevOps Agente rastreá-los e investigá-los adequadamente.

1. Navegue até CloudFormation:
   a. No AWS Console, pesquise por "CloudFormation" e clique em CloudFormation
   b. Clique em Criar pilha > Com novos recursos (padrão)
2. Carregar modelo:
   a. Crie um novo arquivo local chamado `AWS-DevOpsAgent-ec2-test.yaml`
   b. Copie e cole esse CloudFormation modelo no arquivo:
      i.
```
AWSTemplateFormatVersion: '2010-09-09'
Description: 'AWS DevOps Agent EC2 CPU Test Stack'
Parameters:
  VpcId:
    Type: AWS::EC2::VPC::Id
    Description: ID of an existing VPC where the test instance will be launched.
  SubnetId:
    Type: AWS::EC2::Subnet::Id
    Description: ID of an existing subnet within the selected VPC. Choose a
 subnet that routes to an internet gateway if you plan to connect via SSH.
```

```
  MyIP:
    Type: String
    Description: Your current IP address for SSH access (find at https://
whatismyipaddress.com)
    Default: '0.0.0.0/0'
Resources:
  # Security Group for SSH access
  TestSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: AWS DevOps Agent beta testing security group
      VpcId: !Ref VpcId
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 22
          ToPort: 22
          CidrIp: !Ref MyIP
          Description: SSH access from your IP
      Tags:
        - Key: Name
          Value: AWS-DevOpsAgent-Test-SG
        - Key: Purpose
          Value: AWS-DevOpsAgent-Testing
  # Key Pair for SSH access
  TestKeyPair:
    Type: AWS::EC2::KeyPair
    Properties:
      KeyName: AWS-DevOpsAgent-test-key
      KeyType: rsa
      Tags:
        - Key: Name
          Value: AWS-DevOpsAgent-Test-Key
        - Key: Purpose
          Value: AWS-DevOpsAgent-Testing
  # IAM Role for Session Manager access
  SSMInstanceRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              Service: ec2.amazonaws.com
```

```
            Action: sts:AssumeRole
      ManagedPolicyArns:
        - arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
      Tags:
        - Key: Name
          Value: AWS-DevOpsAgent-Test-SSMRole
        - Key: Purpose
          Value: AWS-DevOpsAgent-Testing
  # Instance profile wrapping the SSM role
  SSMInstanceProfile:
    Type: AWS::IAM::InstanceProfile
    Properties:
      Roles:
        - !Ref SSMInstanceRole
  # EC2 Instance for CPU testing
  TestInstance:
    Type: AWS::EC2::Instance
    Properties:
      InstanceType: t3.micro
      ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-
kernel-6.1-x86_64}}'
      KeyName: !Ref TestKeyPair
      SubnetId: !Ref SubnetId
      SecurityGroupIds:
        - !GetAtt TestSecurityGroup.GroupId
      IamInstanceProfile: !Ref SSMInstanceProfile
      InstanceInitiatedShutdownBehavior: terminate
      UserData:
        Fn::Base64: !Sub |
          #!/bin/bash
          yum update -y
          yum install -y htop

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

          # Get number of CPU cores
          CORES=$(nproc)
```

```
          echo "CPU Cores: $CORES"
          echo ""

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

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

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

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

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

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

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

          echo "AWS DevOpsAgent test setup completed at $(date)" > /home/ec2-
user/setup-complete.txt
      Tags:
```

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

  SecurityGroupId:
    Description: Security Group ID
    Value: !GetAtt TestSecurityGroup.GroupId

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

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

c. No CloudFormation console, selecione Carregar um arquivo de modelo

d. Clique em Escolher arquivo

e. Selecione o `AWS-DevOpsAgent-ec2-test.yaml` arquivo

f. Clique em Avançar.

3. Configurar pilha:
   a. Nome da pilha: `AWS-DevOpsAgent-EC2-Test`
   b. Parâmetros:
      i. VpcId: selecione uma VPC existente no menu suspenso.
      ii. SubnetId: selecione uma sub-rede dentro da VPC que você escolheu. Para acesso SSH, a sub-rede deve ser roteada para um gateway da Internet e a instância deve ter um endereço IPv4 público associado. Caso contrário, a `SSHCommand` saída ficará vazia e as conexões SSH não serão bem-sucedidas.
      iii. MyIP: deixe como padrão `0.0.0.0/0` (você pode protegê-lo mais tarde, se necessário)
   c. Clique em Avançar.
4. Configure as opções de pilha:
   a. Deixe os padrões, clique em Avançar
5. Revisar e criar:
   a. Verifique, reconheço que isso AWS CloudFormation pode criar recursos do IAM
   b. Clique em Enviar
6. Aguarde a conclusão:
   a. A criação da pilha leva de 3 a 5 minutos
   b. O status mudará de `CREATE_IN_PROGRESS` para `CREATE_COMPLETE`
   c. Importante: Sua instância do EC2 agora faz parte de uma CloudFormation pilha que AWS DevOpsAgent pode ser rastreada!

Opcional: acesso SSH seguro (somente se você planeja se conectar à instância)

Ignore esta etapa se você quiser apenas executar o teste automatizado

1. Localize o grupo de segurança:
   a. No AWS Console, acesse CloudFormatione selecione a `AWS-DevOpsAgent-EC2-Test` pilha
   b. Abra a guia Saídas e copie o valor de `SecurityGroupId` (começa com`sg-`)
   c. Vá para EC2 → Security Groups e cole o ID na barra de pesquisa para abrir o grupo de segurança
2. Atualize a regra SSH:
   a. Selecione o grupo de segurança → guia Regras de entrada → Editar regras de entrada
   b. Encontre a regra SSH (porta 22)

c. Altere a fonte de `0.0.0.0/0` para seu IP: `[YOUR_IP]/32`

d. Obtenha seu IP de https://whatismyipaddress.com

e. Clique em Salvar regras

Etapa 2: Aguarde a execução automática do teste

1. Execução automática de testes:
   - O teste de estresse da CPU será iniciado automaticamente 5 minutos após a inicialização da instância
   - Nenhuma intervenção manual é necessária - basta esperar, o teste é executado completamente em segundo plano
2. Monitore o teste:
   - A instância inicializa e prepara o teste automaticamente
   - O script será executado por 5 minutos e gerará > 70% de uso da CPU
   - CloudWatch o alarme deve ser acionado dentro de 8 a 10 minutos no total (5 min de atraso + 3-5 min para alarme)
3. Opcional: nova execução manual (para testes adicionais):
   - Conecte-se à sua instância: console EC2 → → Connect **`AWS-DevOpsAgent-Test-Instance`** → Session Manager
   - Execute o teste de estresse novamente: `./cpu-stress-test.sh`
   - Perfeito para testar AWS DevOpsAgent a resposta várias vezes

Opção de teste B: teste de taxa de erro Lambda

Etapa 1: implantar a CloudFormation pilha para o teste Lambda

1. Navegue até CloudFormation:
   a. No AWS Console, acesse CloudFormation
   b. Clique em Criar pilha → Com novos recursos (padrão)
2. Carregar modelo:
   a. Crie um novo arquivo local chamado `AWS-DevOpsAgent-lambda-test.yaml`
   b. Copie e cole esse CloudFormation modelo no arquivo:

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. No CloudFormation console, selecione Carregar um arquivo de modelo

d. Clique em Escolher arquivo

e. Selecione o `AWS-DevOpsAgent-lambda-test.yaml` arquivo

f. Clique em Avançar.

3. Configurar pilha:

a. Nome da pilha: `AWS-DevOpsAgent-Lambda-Test`

b. Clique em Avançar.

4. Configure as opções de pilha:

a. Deixe os padrões, clique em Avançar

5. Revisar e criar:

a. Verifique, reconheço que isso AWS CloudFormation pode criar recursos do IAM

b. Clique em Enviar

6. Aguarde a conclusão:

a. A criação da pilha leva de 2 a 3 minutos

b. O status mudará para `CREATE_COMPLETE`

Etapa 2: acionar erros do Lambda

1. Navegue até o console Lambda:

a. Vá para o console AWS Lambda

b. Encontre sua função `AWS-DevOpsAgent-test-lambda`

2. Teste a função:

a. Clique na guia Teste

b. Clique em Criar novo evento

c. Nome do evento: `AWS-DevOpsAgent-test-event`

d. Use essa carga JSON:

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

e. Clique em Salvar

3. Gere erros:

a. Clique no botão Testar 3 vezes (aguarde 10 segundos entre cada uma)

b. Cada teste gerará um erro intencional

c. CloudWatch o alarme deve ser acionado dentro de 2-3 minutos

d. AWS DevOpsAgentagora deve ser capaz de detectar o alarme com uma investigação no aplicativo Operator, que você configurará a seguir.

Validar AWS DevOps Detecção de agentes

Etapa 1: CloudWatch alarmes de verificação de sanidade (opcional)

Esta etapa é para garantir que os testes acima estejam agora em um estado de alarme.

Para o teste EC2:

- No CloudWatch console, acesse Alarmes
- Aguarde de 3 a 5 minutos após iniciar o teste de estresse
- Seu alarme deve aparecer Em estado de alarme
- Se ainda estiver “OK”: aguarde mais 2 a 3 minutos (CloudWatch as métricas podem ser atrasadas)

Para o Teste Lambda:

- Verifique o `AWS-DevOpsAgent-Lambda-Error-Test` alarme
- Deve mostrar o alarme dentro de 2-3 minutos após a execução dos testes

Etapa 2: iniciar um AWS DevOps Investigação do agente

1. Abra seu AWS DevOps agente AgentSpace

2. Clique em Acesso de administrador. Isso abrirá o aplicativo web DevOps Agent Space em uma nova janela
3. Clique no botão Iniciar investigação no lado direito da tela
4. Preencha o seguinte formulário:
   a. Detalhes da investigação: descreva a investigação que você gostaria de realizar. Inclua todos os detalhes que puder sobre os objetivos da investigação, áreas a serem exploradas ou informações relevantes.
   b. Ponto de partida da investigação: descreva as informações a partir das quais você gostaria de iniciar a investigação. Você pode mencionar um alarme, uma métrica, um trecho de registro ou qualquer outra coisa para dar ao DevOps Agente um ponto de partida para trabalhar. Nesse caso, forneça um resumo dos alarmes que você acabou de criar.
   c. Data e hora do incidente (de preferência ISO 8601): YYYY-MM-DDTHH:MMZ
   d. Dê um nome à sua investigação: exemplo: `Oncall_investigation_1:2025-10-27`
   e. AWS ID da conta do incidente
   f. Região onde o incidente ocorreu
   g. Prioridade - AWS DevOpsAgent permite duas investigações simultâneas. A Prioridade permite que você defina a ordem de execução de suas investigações.
5. Clique em Investigar para iniciar a investigação.
6. Clique na sua investigação listada no painel. Você será direcionado para a tela Detalhes da Investigação, onde poderá ver as etapas granulares que o DevOps Agente está realizando.

Resultados esperados

Resultados do teste EC2:

- Detecta o alarme da CPU EC2
- Identifica a causa raiz: “carga de trabalho do teste de estresse da CPU”
- Mostra o cronograma: Teste de estresse → pico da CPU → Alarme
- Fornece recomendações para monitoramento e escalabilidade

Resultados do teste Lambda:

- Detecta o pico da taxa de erro do Lambda
- Identifica a causa raiz: “Exceções de testes intencionais”

- Mostra a linha do tempo: Invocações de funções → Erros → Alarme
- Fornece recomendações para tratamento e monitoramento de erros

Instruções de limpeza

Teste de limpeza A (teste EC2)

Limpeza automática

- A instância será encerrada automaticamente após 2 horas (CloudFormation incorporada ao modelo)

Limpeza manual (imediata)

1. Excluir CloudFormation pilha:
   a. Vá para o CloudFormation console
   b. Selecione a `AWS-DevOpsAgent-EC2-Test` pilha
   c. Clique em Excluir
   d. Confirme a exclusão
   e. Isso excluirá automaticamente todos os recursos: instância do EC2, grupo de segurança, key pair e alarme CloudWatch

Teste de limpeza B (teste Lambda)

1. Excluir CloudFormation pilha:
   a. Vá para o CloudFormation console
   b. Selecione a `AWS-DevOpsAgent-Lambda-Test` pilha
   c. Clique em Excluir
   d. Confirme a exclusão
   e. Isso excluirá automaticamente todos os recursos: função Lambda, função do IAM e alarme CloudWatch

Solução de problemas

Problemas comuns

“Não é possível se conectar à instância do EC2”

- Verifique o grupo de segurança: verifique se o SSH (porta 22) está aberto para o seu IP
- Verifique as principais permissões: Executar `chmod 400 AWS-DevOpsAgent-test-key.pem`
- Verifique o IP público: a instância deve ter um IP público atribuído
- Aguarde a instância: verifique se a instância está no estado “Executando”

“O alarme não está sendo acionado”

- Aguarde as métricas: CloudWatch as métricas podem levar de 2 a 5 minutos para aparecer
- Verifique a carga da CPU: SSH para a instância e execute `top` para verificar a CPU > 70%
- Verifique o teste de estresse: execute `ps aux | grep yes` para ver se os processos de carregamento estão em execução
- Espera prolongada: às vezes, leva de 7 a 8 minutos para o primeiro acionamento do alarme

Validação de testes

Seu teste de AWS DevOp agente é bem-sucedido quando:

Validação técnica

- Precisão da investigação: Os resultados do teste EC2 devem indicar corretamente que o alarme foi acionado devido à carga da CPU. O resultado do teste Lambda deve indicar que se tratou de uma falha intencional.
- Precisão do cronograma: sequência correta de eventos mostrados
- Qualidade da recomendação: sugestões práticas fornecidas

Introdução ao AWS DevOps Agent usando o AWS CDK

Visão geral do

Este guia mostra como usar o AWS Cloud Development Kit (AWS CDK) para criar e implantar recursos do AWS DevOps Agente. O aplicativo AWS CDK automatiza a criação de um espaço de agente, funções de AWS Identity and Access Management (IAM), um aplicativo de operador e associações de contas por meio de. AWS AWS CloudFormation

A abordagem AWS CDK automatiza as etapas manuais descritas no guia de integração da CLI, definindo todos os recursos necessários como infraestrutura como código.

AWS DevOps O agente está disponível nas seguintes 6 AWS regiões: Leste dos EUA (Norte da Virgínia), Oeste dos EUA (Oregon), Ásia-Pacífico (Sydney), Ásia-Pacífico (Tóquio), Europa (Frankfurt) e Europa (Irlanda). Para obter mais informações sobre as regiões suportadas, consultethe section called “Regiões aceitas”.

Pré-requisitos

Antes de começar, verifique se você tem o seguinte:

- AWS Interface de linha de comando (AWS CLI) instalada e configurada com as credenciais apropriadas
- Node.js, versão 18 ou versões posteriores
- AWS Interface de linha de comando (CLI) do CDK instalada globalmente. Para instalar a CLI do AWS CDK, execute o seguinte comando:

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

- Uma AWS conta para a conta de monitoramento (primária)
- (Opcional) Uma segunda AWS conta se você quiser configurar o monitoramento entre contas

O que este guia aborda

Este guia está dividido em duas partes:

- Parte 1 — Implante um espaço de agente com um aplicativo de operador e uma AWS associação em sua conta de monitoramento. Depois de concluir essa parte, o agente pode monitorar problemas nessa conta.
- Parte 2 (opcional) — Adicione uma AWS associação de origem para uma conta de serviço e implante uma função do IAM entre contas nessa conta. Essa configuração permite que o espaço do agente monitore recursos em todas as contas.

Recursos criados

Parte 1: DevOpsAgentStack (conta de monitoramento)

- Função do IAM (`DevOpsAgentRole-AgentSpace`) — assumida pelo serviço do DevOps agente para monitorar a conta. Inclui a política `AIDevOpsAgentAccessPolicy` gerenciada e uma política em linha que permite a criação da função vinculada ao serviço Resource Explorer.
- Função do IAM (`DevOpsAgentRole-WebappAdmin`) — Função do aplicativo operador com a política `AIDevOpsOperatorAppAccessPolicy` gerenciada para operações do agente.
- Espaço do agente (`MyCDKAgentSpace`) — O espaço do agente central, criado usando o `AWS::DevOpsAgent::AgentSpace` CloudFormation recurso. Inclui a configuração do aplicativo do operador.
- Associação (AWS monitor) — vincula a conta de monitoramento ao espaço do agente usando o `AWS::DevOpsAgent::Association` CloudFormation recurso.
- Associação (AWS fonte) — (Opcional) Vincula a conta de serviço ao espaço do agente para monitoramento entre contas.

Parte 2: ServiceStack (conta de serviço, opcional)

- Função do IAM (`DevOpsAgentRole-SecondaryAccount`) — Função entre contas com um nome fixo. Confiável pelo espaço do agente na conta de monitoramento. Inclui a política `AIDevOpsAgentAccessPolicy` gerenciada e uma política em linha que permite a criação da função vinculada ao serviço Resource Explorer.
- Função Lambda (`echo-service`) — Um exemplo simples de serviço que reflete eventos de entrada.

Configuração

Etapa 1: clonar o repositório de amostra

Execute os comandos a seguir para clonar o repositório e mudar para o diretório do projeto:

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

Etapa 2: instalar dependências

Execute o comando a seguir para instalar as dependências do projeto:

```
npm install
```

Parte 1: Implantar o espaço do agente

Nesta seção, você cria o espaço do agente, as funções do IAM, o aplicativo do operador e uma AWS associação na sua conta de monitoramento.

Etapa 1: configurar o ID da conta de monitoramento

Abra `lib/constants.ts` e defina o ID da sua conta de monitoramento:

O exemplo a seguir mostra a constante a ser atualizada:

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

Etapa 2: Inicializar o ambiente AWS CDK

Se você não inicializou o AWS CDK em sua conta de monitoramento, execute o seguinte comando:

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

Etapa 3: criar e implantar

Execute os comandos a seguir para criar o TypeScript código e implantar a pilha:

```
npm run build
```

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

Etapa 4: Grave as saídas da pilha

Após a conclusão da implantação, o AWS CDK imprime as saídas da pilha. Registre esses valores para uso posterior.

O exemplo a seguir mostra a saída esperada:

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

Se você planeja concluir a Parte 2, salve o `AgentSpaceArn` valor. Você precisa dele para configurar a pilha de contas de serviço.

Etapa 5: verificar a implantação

Para verificar se o espaço do agente foi criado com êxito, execute o seguinte comando da AWS CLI:

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

Nesse ponto, seu espaço de agente é implantado com o aplicativo do operador ativado e sua conta de monitoramento associada. O agente pode monitorar problemas nessa conta.

Parte 2 (opcional): adicionar monitoramento entre contas

Nesta seção, você estende a configuração para que seu espaço de agente possa monitorar recursos em uma segunda AWS conta (a conta de serviço). Isso envolve duas ações:

1. Adicionar uma AWS associação de origem na DevOpsAgentStack que aponta para a conta de serviço.
2. Implantando o ServiceStack na conta de serviço com uma função do IAM que confia no espaço do agente.

> **Important**
>
> Você deve concluir a Parte 1 antes de continuar. O ServiceStack requer o `AgentSpaceArn` da saída DevOpsAgentStack de implantação.

Etapa 1: configurar o ID da conta de serviço

Abra `lib/constants.ts` e defina o ID da sua conta de serviço:

O exemplo a seguir mostra a constante a ser atualizada:

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

O DevOpsAgentStack cria uma AWS associação de origem usando esse ID de conta. Se você implantou o DevOpsAgentStack antes de definir esse valor, reimplante para criar a associação:

Execute os seguintes comandos para reimplantar:

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

Etapa 2: definir o ARN do espaço do agente

Copie o `AgentSpaceArn` valor da DevOpsAgentStack saída (Parte 1, Etapa 4) e defina-o em`lib/constants.ts`:

O exemplo a seguir mostra a constante a ser atualizada:

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

O ServiceStack usa esse valor para definir o escopo da política de confiança na função de conta secundária. O só ServiceStack é sintetizado quando esse valor é definido.

Etapa 3: inicializar a conta de serviço

Se você não inicializou o AWS CDK na sua conta de serviço, execute o seguinte comando:

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

Etapa 4: implantar o ServiceStack

Execute os comandos a seguir para criar e implantar o ServiceStack usando credenciais para a conta de serviço:

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

Isso cria os seguintes recursos na conta de serviço:

- Uma função do IAM (`DevOpsAgentRole-SecondaryAccount`) que confia no espaço do agente na conta de monitoramento
- Uma função echo Lambda `echo-service` () como um serviço de exemplo

Etapa 5: verificar a implantação

Para confirmar que a função Lambda foi implantada com sucesso, execute os seguintes comandos para testar o serviço echo:

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

Solução de problemas

Esta seção descreve problemas comuns e como resolvê-los.

CloudFormation tipo de recurso não encontrado

- Verifique se você está implantando em umthe section called “Regiões aceitas”.
- Confirme se sua AWS CLI está configurada com as permissões apropriadas.

Falha na criação da função do IAM

- Verifique se sua função de implantação tem permissões para criar funções do IAM.
- Verifique se as condições da política de confiança correspondem ao ID da sua conta.

A implantação entre contas falha com “Não foi possível assumir a função na conta de destino”

- Cada pilha deve ser implantada com credenciais para a conta de destino. Use o `--profile` sinalizador para especificar o perfil AWS CLI correto.
- Verifique se o AWS CDK foi inicializado na conta de destino.

Atrasos na propagação do IAM

- As mudanças de função do IAM podem levar alguns minutos para se propagar. Se a criação do espaço do agente falhar imediatamente após a criação da função, aguarde alguns minutos e reimplante.

Limpeza

Para remover todos os recursos, destrua as pilhas na ordem inversa.

Execute os seguintes comandos para destruir as pilhas:

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

Aviso: essa ação exclui permanentemente o espaço do agente e todos os dados associados. Esta ação não pode ser desfeita. Certifique-se de ter feito backup de todas as informações importantes antes de continuar.

Considerações sobre segurança

- O aplicativo AWS CDK cria funções do IAM com políticas de confiança que só permitem que o responsável pelo `aidevops.amazonaws.com` serviço as assuma.
- As políticas de confiança incluem condições que restringem o acesso à sua AWS conta específica e ao ARN do espaço do agente.
- Todas as políticas seguem o princípio do menor privilégio. Analise e personalize as políticas do IAM com base nos requisitos de segurança da sua organização.
- A função entre contas (`DevOpsAgentRole-SecondaryAccount`) usa um nome fixo e tem como escopo um ARN de espaço de agente específico.

Próximas etapas

Depois de implantar seu AWS DevOps agente usando o AWS CDK:

1. Saiba mais sobre a gama completa de recursos do DevOps Agente no Guia do Usuário do AWS DevOps Agente.
2. Considere integrar a implantação do AWS CDK em seus CI/CD pipelines para gerenciamento automatizado da infraestrutura.

Recursos adicionais do

- AWS DevOps Guia do usuário do agente
- Exemplo de repositório CDK no site GitHub
- Guia de integração da CLI

Começando a usar o AWS DevOps Agent usando AWS CloudFormation

Visão geral do

Este guia mostra como usar AWS CloudFormation modelos para criar e implantar recursos do AWS DevOps Agente. Os modelos automatizam a criação de um espaço para agentes, funções de AWS Identity and Access Management (IAM), um aplicativo de operador e associações de AWS contas como infraestrutura como código.

A CloudFormation abordagem automatiza as etapas manuais descritas no guia de integração da CLI definindo todos os recursos necessários em modelos YAML declarativos.

AWS DevOps O agente está disponível nas seguintes 6 AWS regiões: Leste dos EUA (Norte da Virgínia), Oeste dos EUA (Oregon), Ásia-Pacífico (Sydney), Ásia-Pacífico (Tóquio), Europa (Frankfurt) e Europa (Irlanda). Para obter mais informações sobre as regiões suportadas, consultethe section called “Regiões aceitas”.

Pré-requisitos

Antes de começar, verifique se você tem o seguinte:

- AWS Interface de linha de comando (AWS CLI) instalada e configurada com as credenciais apropriadas
- Permissões para criar funções e CloudFormation pilhas do IAM
- Uma AWS conta para a conta de monitoramento (primária)
- (Opcional) Uma segunda AWS conta se você quiser configurar o monitoramento entre contas

O que este guia aborda

Este guia está dividido em duas partes:

- Parte 1 — Implante um espaço de agente com um aplicativo de operador e uma AWS associação em sua conta de monitoramento. Depois de concluir essa parte, o agente pode monitorar problemas nessa conta.
- Parte 2 (opcional) — Implante uma função do IAM entre contas em uma conta secundária e adicione uma AWS associação de origem. Essa configuração permite que o espaço do agente monitore recursos em todas as contas.

Parte 1: Implantar o espaço do agente

Nesta seção, você cria um CloudFormation modelo que provisiona o espaço do agente, as funções do IAM, o aplicativo do operador e uma AWS associação na sua conta de monitoramento.

Etapa 1: criar o CloudFormation modelo

Salve o modelo a seguir 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
```

Etapa 2: implantar a pilha

Execute o comando a seguir para implantar a pilha. <REGION>Substitua por um the section called "Regiões aceitas" (por exemplo,us-east-1).

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

Etapa 3: Grave as saídas da pilha

Após a conclusão da implantação, execute o comando a seguir para recuperar as saídas da pilha. Registre esses valores para uso posterior.

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

O exemplo a seguir mostra a saída esperada:

```
[
  {
    "OutputKey": "AgentSpaceId",
    "OutputValue": "abc123def456"
  },
  {
    "OutputKey": "AgentSpaceArn",
    "OutputValue": "arn:aws:aidevops:<REGION>:<ACCOUNT_ID>:agentspace/abc123def456"
  },
  {
```

```
    "OutputKey": "AgentSpaceRoleArn",
    "OutputValue": "arn:aws:iam::<ACCOUNT_ID>:role/DevOpsAgentRole-AgentSpace"
  },
  {
    "OutputKey": "OperatorRoleArn",
    "OutputValue": "arn:aws:iam::<ACCOUNT_ID>:role/DevOpsAgentRole-WebappAdmin"
  }
]
```

Se você planeja concluir a Parte 2, salve o `AgentSpaceArn` valor. Você precisa dele para configurar a função entre contas.

Etapa 4: Verificar a implantação

Para verificar se o espaço do agente foi criado com êxito, execute o seguinte comando da AWS CLI:

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

Nesse ponto, seu espaço de agente é implantado com o aplicativo do operador ativado e sua conta de monitoramento associada. O agente pode monitorar problemas nessa conta.

Parte 2 (opcional): adicionar monitoramento entre contas

Nesta seção, você estende a configuração para que seu espaço de agente possa monitorar recursos em uma segunda AWS conta (a conta de serviço). Isso envolve duas ações:

1. Implantação de uma função do IAM na conta de serviço que confia no espaço do agente.
2. Adicionar uma AWS associação de origem na conta de monitoramento que aponta para a conta de serviço.

> Nota: Você deve concluir a Parte 1 antes de continuar. O modelo de conta de serviço exige as saídas `AgentSpaceArn` da pilha da Parte 1.

Etapa 1: criar o modelo de conta de serviço

Salve o modelo a seguir como`devops-agent-service-account.yaml`. Esse modelo cria uma função do IAM entre contas na conta secundária.

```
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
```

Etapa 2: implantar a pilha de contas de serviço

Usando as credenciais da conta de serviço, execute o seguinte 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>
```

Etapa 3: Adicionar a AWS associação de origem

Volte para a conta de monitoramento e crie uma AWS associação de origem. Você pode fazer isso criando uma pilha separada ou atualizando o modelo original. O exemplo a seguir usa um modelo independente.

Salve o modelo a seguir 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
```

Implante a pilha de associações usando as credenciais da conta de monitoramento:

```
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>
```

Verificação

Verifique sua configuração executando os seguintes comandos da 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>
```

Solução de problemas

Esta seção descreve problemas comuns e como resolvê-los.

CloudFormation tipo de recurso não encontrado

- Verifique se você está implantando em umthe section called “Regiões aceitas”.
- Confirme se sua AWS CLI está configurada com as permissões apropriadas.

Falha na criação da função do IAM

- Verifique se suas credenciais de implantação têm permissões para criar funções do IAM com nomes personalizados (`CAPABILITY_NAMED_IAM`).
- Verifique se as condições da política de confiança correspondem ao ID da sua conta.

Falha na implantação entre contas

- Cada pilha deve ser implantada com credenciais para a conta de destino. Use o `--profile` sinalizador para especificar o perfil AWS CLI correto.
- Verifique se o `AgentSpaceArn` parâmetro corresponde ao ARN exato das saídas da pilha da Parte 1.

Atrasos na propagação do IAM

- As mudanças de função do IAM podem levar alguns minutos para se propagar. Se a criação do espaço do agente falhar imediatamente após a criação da função, aguarde alguns minutos e reimplante.

Limpeza

Para remover todos os recursos, exclua as pilhas na ordem inversa.

Aviso: essa ação exclui permanentemente o espaço do seu agente e todos os dados associados. Esta ação não pode ser desfeita. Certifique-se de ter feito backup de todas as informações importantes antes de continuar.

Execute os seguintes comandos para excluir as pilhas:

```
# 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>
```

Próximas etapas

Depois de implantar seu AWS DevOps agente usando AWS CloudFormation:

- Para conectar integrações adicionais, consulteConfigurando recursos para AWS DevOps Agente.
- Para saber mais sobre as habilidades e capacidades dos agentes, consultethe section called “DevOps Habilidades do agente”.
- Para entender o aplicativo web do operador, consultethe section called “O que é um DevOps Agent Web App?”.

Introdução ao AWS DevOps Agent usando o Terraform

Visão geral do

Este guia mostra como usar o Terraform para criar e implantar recursos do AWS DevOps Agente. A configuração do Terraform automatiza a criação de um espaço de agente, funções do IAM, um aplicativo de operador e associações de AWS contas.

A abordagem do Terraform automatiza as etapas manuais descritas no guia de integração da CLI, definindo todos os recursos necessários como infraestrutura como código.

AWS DevOps O agente está disponível nas seguintes 6 AWS regiões: Leste dos EUA (Norte da Virgínia), Oeste dos EUA (Oregon), Ásia-Pacífico (Sydney), Ásia-Pacífico (Tóquio), Europa (Frankfurt) e Europa (Irlanda). Para obter mais informações sobre as regiões suportadas, consultethe section called “Regiões aceitas”.

Pré-requisitos

Antes de começar, você deve ter o seguinte:

- Terraform >= 1.0 instalado
- AWS CLI instalada e configurada com as credenciais apropriadas
- Uma AWS conta para a conta de monitoramento (primária)
- (Opcional) Uma segunda AWS conta se você quiser configurar o monitoramento entre contas

O que este guia aborda

Este guia está dividido em duas partes:

- Parte 1 — Implante um espaço de agente com um aplicativo de operador e uma AWS associação em sua conta de monitoramento. Depois de concluir essa parte, o agente pode monitorar problemas nessa conta.
- Parte 2 (opcional) — Adicione uma AWS associação de origem para uma conta de serviço e implante uma função do IAM entre contas, além de um echo Lambda nessa conta. Isso permite que o espaço do agente monitore os recursos em todas as contas.

Recursos criados

Parte 1: Conta de monitoramento

- Função do IAM (`DevOpsAgentRole-AgentSpace-*`) — assumida pelo serviço do DevOps agente para monitorar a conta. Inclui a política `AIDevOpsAgentAccessPolicy` gerenciada e uma política em linha que permite a criação da função vinculada ao serviço Resource Explorer.
- Função do IAM (`DevOpsAgentRole-WebappAdmin-*`) — Função do aplicativo operador com a política `AIDevOpsOperatorAppAccessPolicy` gerenciada para operações do agente.
- Espaço do agente (nome configurável) — O espaço do agente central, criado usando o `awscc_devopsagent_agent_space` recurso. Inclui a configuração do aplicativo do operador.
- Associação (AWS monitor) — vincula a conta de monitoramento ao espaço do agente usando o `awscc_devopsagent_association` recurso.
- Associação (AWS fonte) — (Opcional) Vincula a conta de serviço ao espaço do agente para monitoramento entre contas.

Parte 2: Conta de serviço (opcional)

- Função do IAM (`DevOpsAgentRole-SecondaryAccount-TF`) — Função entre contas com um nome fixo. Confiável pelo espaço do agente na conta de monitoramento. Inclui a política `AIDevOpsAgentAccessPolicy` gerenciada e uma política em linha que permite a criação da função vinculada ao serviço Resource Explorer.
- Função Lambda (`echo-service-tf`) — Um exemplo simples de serviço que reflete eventos de entrada.

Configuração

Etapa 1: clonar o repositório de amostra

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

Etapa 2: configurar variáveis

Copie o arquivo de variáveis de exemplo e personalize-o para seu ambiente:

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

Edite `terraform.tfvars` com o nome e a descrição do seu espaço de agente:

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

Parte 1: Implantar o espaço do agente

Nesta seção, você cria o espaço do agente, as funções do IAM, o aplicativo do operador e uma AWS associação na sua conta de monitoramento.

Etapa 1: implantar com automação (recomendado)

Use o script de implantação fornecido para uma configuração simplificada:

```
./deploy.sh
```

Esse script automaticamente:

- Verifica os pré-requisitos (Terraform, AWS CLI, credenciais)
- Cria `terraform.tfvars` a partir do exemplo, se necessário
- Inicializa, valida, planeja e aplica o Terraform

Como alternativa, se você preferir o controle manual:

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

Digite `yes` quando solicitado para confirmar a implantação.

Etapa 2: Grave as saídas

Após a conclusão da implantação, o Terraform imprime as saídas. Registre esses valores para uso posterior:

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

Se você planeja concluir a Parte 2, salve o `agent_space_arn` valor. Você precisará dele para configurar os recursos da conta de serviço.

Etapa 3: verificar a implantação

Execute o script de verificação pós-implantação:

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

Ou use a AWS CLI para verificar se o espaço do agente foi criado com êxito:

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

Nesse ponto, seu espaço de agente é implantado com o aplicativo do operador ativado e sua conta de monitoramento associada. O agente pode monitorar problemas nessa conta.

Parte 2 (opcional): adicionar monitoramento entre contas

Nesta seção, você estende a configuração para que o espaço do agente possa monitorar recursos em uma segunda AWS conta (a conta de serviço). Isso envolve duas ações:

1. Adicionar uma AWS associação de origem que aponta para a conta de serviço.
2. Implantação de uma função IAM entre contas e uma função echo Lambda na conta de serviço.

> **Important**
>
> Você deve concluir a Parte 1 antes de continuar. Os recursos da conta de serviço exigem o resultado `agent_space_arn` da implantação da Parte 1.

Etapa 1: configurar o ID da conta de serviço

Em`terraform.tfvars`, defina o ID da sua conta de serviço:

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

Etapa 2: definir o ARN do espaço do agente

Copie o `agent_space_arn` valor da saída da Parte 1 (Etapa 2) e defina-o em`terraform.tfvars`:

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

Os recursos da conta de serviço usam esse valor para definir o escopo da política de confiança na função secundária da conta. Esses recursos são criados somente quando esse valor é definido.

Etapa 3: configurar o provedor `aws.service`

Em`main.tf`, configure o alias do `aws.service` provedor com as credenciais da conta de serviço. Você pode usar um perfil nomeado ou assumir uma função:

Usando um perfil:

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

Ou usando assume a função:

```
provider "aws" {
  alias  = "service"
  region = var.aws_region
  assume_role {
```

```
    role_arn = "arn:aws:iam::<SERVICE_ACCOUNT_ID>:role/OrganizationAccountAccessRole"
  }
}
```

Etapa 4: implantar

Aplique a configuração atualizada:

```
terraform apply
```

Isso cria os seguintes recursos na conta de serviço:

- Uma função do IAM (`DevOpsAgentRole-SecondaryAccount-TF`) que confia no espaço do agente na conta de monitoramento
- Uma função echo Lambda `echo-service-tf` () como um serviço de exemplo

Ele também cria uma AWS associação de origem na conta de monitoramento que vincula a conta de serviço.

Etapa 5: verificar a implantação

Teste o serviço echo para confirmar se a função Lambda foi implantada com sucesso:

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

Solução de problemas

Atrasos na propagação do IAM

- A configuração inclui 30 segundos `time_sleep` entre a criação da função do IAM e a criação do Agent Space. O serviço de DevOps agente valida a política de confiança da função de operador durante a criação do Agent Space, e isso pode falhar se o IAM não for totalmente propagado. Se você ainda encontrar erros na política de confiança, espere um minuto e execute `terraform apply` novamente. As funções do IAM já existirão e a aplicação continuará de onde parou.

Erros de permissão

- Verifique se suas AWS credenciais têm as permissões necessárias do IAM para criar funções e políticas.
- Verifique se as condições da política de confiança correspondem ao ID da sua conta.

Falha na implantação entre contas

- O `aws.service` provedor deve ser configurado com as credenciais da conta de serviço. Use um perfil nomeado ou um bloco de assumir funções.
- Verifique se o `agent_space_arn` valor corresponde ao ARN da saída da Parte 1.

Tipo de recurso do Terraform não encontrado

- Verifique se você tem a versão do `awscc` provedor `~> 1.0` ou posterior. Os `awscc_devopsagent_association` recursos `awscc_devopsagent_agent_space` e exigem o provedor do AWS Cloud Control.

Limpeza

Para remover todos os recursos, destrua na ordem inversa se você implantou a Parte 2:

```
./cleanup.sh
```

Ou manualmente:

```
terraform destroy
```

Aviso: Isso exclui permanentemente seu espaço de agente e todos os dados associados. Certifique-se de ter feito backup de todas as informações importantes antes de continuar.

Considerações sobre segurança

- A configuração do Terraform cria funções do IAM com políticas de confiança que só permitem que o responsável pelo `aidevops.amazonaws.com` serviço as assuma.
- As políticas de confiança incluem condições que restringem o acesso à sua AWS conta específica e ao ARN do espaço do agente.

- Todas as políticas seguem o princípio do menor privilégio. Analise e personalize as políticas do IAM com base nos requisitos de segurança da sua organização.
- A função entre contas (`DevOpsAgentRole-SecondaryAccount-TF`) usa um nome fixo e tem como escopo um ARN de espaço de agente específico.

Próximas etapas

Depois de implantar seu AWS DevOps agente usando o Terraform:

1. Saiba mais sobre a gama completa de recursos do DevOps Agente no Guia do Usuário do AWS DevOps Agente.
2. Considere integrar a implantação do Terraform em seus CI/CD pipelines para gerenciamento automatizado da infraestrutura.

Recursos adicionais do

- AWS DevOps Guia do usuário do agente
- Exemplo de repositório Terraform
- Guia de integração da CLI

Trabalhando com o DevOps agente

Trabalhando com o DevOps agente

AWS DevOps O agente trabalha junto com sua equipe de operações em todo o ciclo de vida do incidente, desde a detecção até a investigação, recuperação e prevenção. Os tópicos a seguir descrevem como usar o DevOps Agent para gerenciar cada fase desse ciclo de vida.

Resposta autônoma a incidentes

Quando um incidente é detectado, seja por meio de uma integração integrada com seu sistema de tíquetes, um webhook de suas ferramentas de monitoramento ou um gatilho manual, o DevOps agente inicia automaticamente uma investigação. O agente analisa métricas, registros, rastreamentos, alterações de código e histórico de implantação para determinar a causa raiz e propor um plano de mitigação. Se precisar de ajuda adicional, você pode encaminhar diretamente para o AWS Support a partir do aplicativo web DevOps Agent Space, que compartilha automaticamente o contexto da investigação com os engenheiros de suporte para que você não precise repetir o que o agente já encontrou. Para obter mais informações, consulte the section called “Resposta autônoma a incidentes”.

Tarefas sob demanda DevOps

A qualquer momento durante o ciclo de vida do incidente, você pode interagir com o DevOps Agente por meio de uma interface de bate-papo conversacional. Faça perguntas sobre seus AWS recursos, integridade do sistema, status do alarme e histórico de implantação usando linguagem natural. O chat reconhece o contexto — ao visualizar uma investigação específica, você pode orientar o agente a explorar hipóteses específicas, focar em registros específicos ou atualizar a análise da causa raiz. Você também pode consultar configurações de recursos, tendências de erros e insights de investigação em todo o seu ambiente sem navegar entre os consoles. Para obter mais informações, consulte the section called “DevOps Tarefas sob demanda”.

Prevenção proativa de incidentes

Depois de resolver os incidentes, o DevOps agente analisa os padrões em seu histórico de investigação para gerar recomendações que evitem futuros incidentes e reduzam o tempo médio de detecção. As recomendações abrangem quatro áreas: postura de observabilidade, lacunas nos

testes, alterações no código e arquitetura da infraestrutura. O agente realiza avaliações semanais e atualiza as recomendações à medida que novos incidentes ocorrem. Você pode aceitar, rejeitar ou monitorar recomendações, e o agente aprende com seus comentários a refinar sugestões futuras. Para obter mais informações, consulte the section called “Prevenção proativa de incidentes”.

Interface com o agente DevOps

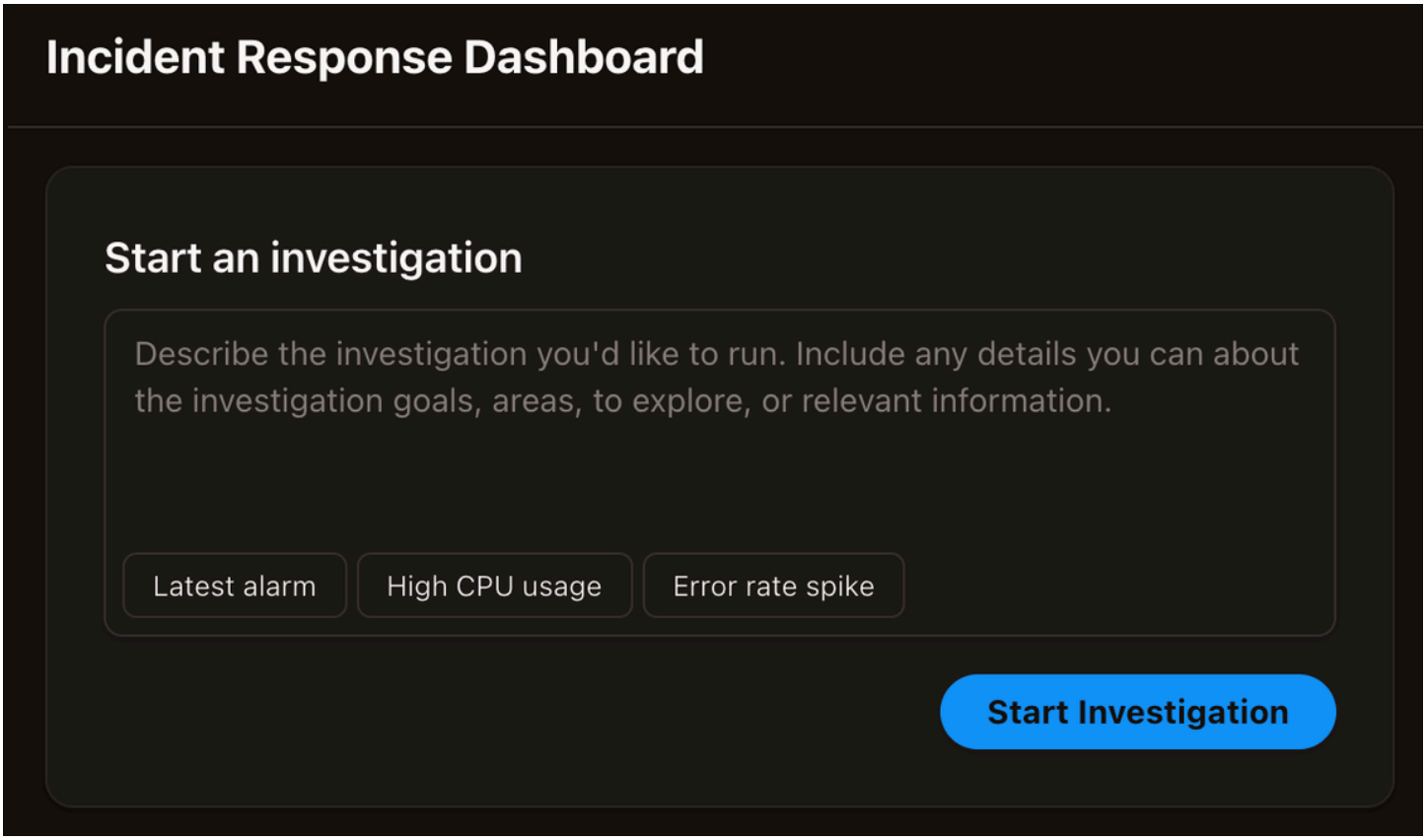
AWS DevOps O Agent suporta vários métodos de acesso, incluindo o console de aplicativos web, integração MCP para IDEs, Agent Client Protocol (ACP), webhooks para automação orientada por eventos e acesso direto à API. Para obter mais informações, consulte the section called “Interface com o agente DevOps”.

Resposta autônoma a incidentes

Iniciando investigações

As investigações de resposta a incidentes podem ser iniciadas de três maneiras.

- Built-in integrações - Você pode conectar um DevOps Agent Space a sistemas de emissão de tíquetes, ServiceNow usando integrações integradas. Depois de conectadas, as investigações de resposta a incidentes do DevOps agente serão acionadas automaticamente a partir dos tíquetes de suporte, e seu DevOps agente fornecerá atualizações das principais descobertas, análises da causa raiz e planos de mitigação no ticket de origem.
- Webhooks - Você pode usar webhooks para enviar eventos ao Agente. AWS DevOps Por exemplo, você pode usar webhooks para acionar investigações de resposta a incidentes a partir de PagerDuty tickets ou alarmes da Grafana.
- Manualmente - Você pode iniciar manualmente as investigações de resposta a incidentes na guia Resposta a incidentes de qualquer aplicativo web do DevOps Agent Space. Você pode inserir um texto de formato livre que descreva o incidente que você deseja que seu DevOps agente investigue, e ele criará um plano de investigação, coletará descobertas, determinará a causa raiz e oferecerá a geração de um plano de mitigação. Você também pode escolher entre vários pontos de partida pré-configurados para iniciar rapidamente sua investigação: alarme mais recente para investigar o alarme acionado mais recente e analisar as métricas e registros subjacentes para determinar a causa raiz, Alto uso da CPU para investigar métricas de alta utilização da CPU em seus recursos computacionais e identificar quais processos ou serviços estão consumindo recursos excessivos ou pico na taxa de erro para investigar o aumento recente nas taxas de erro do aplicativo analisando métricas, registros de aplicativos e identificando a origem das falhas.



Depois de clicar em “Iniciar investigação”, você deverá fornecer alguns detalhes adicionais para ajudar o agente a concentrar seu trabalho. A caixa de diálogo de investigação inclui os seguintes campos:

- Detalhes da investigação — Pre-filled com sua descrição. Você pode editar isso para refinar o escopo da investigação.
- Ponto de partida da investigação — opcionalmente, descreva um alarme, métrica, trecho de registro ou outro ponto de partida específico para o agente.
- Data e hora do incidente — Auto-filled com a hora atual no formato UTC. Ajuste se o incidente ocorreu mais cedo.
- Dê um nome à sua investigação — Auto-generated com um registro de data e hora. Você pode personalizar isso (máximo de 400 caracteres).
- Prioridade — Selecione a prioridade da investigação na lista suspensa (Média é o padrão).

Revise e ajuste esses campos conforme necessário e clique em “Começar a investigar...” para começar. Em seguida, você será direcionado para a página de detalhes da investigação, onde poderá ver seu DevOps agente em ação!

Triagem de incidentes

A fase de triagem é a primeira etapa do sistema de resposta a incidentes do AWS DevOps agente. Quando um evento externo é acionado, como um alarme do Datadog, um ticket de incidente ou um problema do Dynatrace ServiceNow, o AWS DevOps Agent o processa automaticamente em segundos para determinar se ele deve ser investigado de forma independente ou vinculado a uma investigação existente.

A função principal do estágio de triagem é a correlação de incidentes — identificar incidentes relacionados e consolidá-los em uma única investigação para evitar trabalho duplicado e desperdício de recursos. Quando chega um novo incidente, o AWS DevOps agente o analisa junto com as investigações ativas em uma janela retrospectiva (normalmente 20 minutos). Usando a AI-powered análise, ele examina fatores como semelhanças de componentes, região geográfica e padrões de tempo para determinar as relações entre incidentes.

AWS DevOps O agente toma uma das três decisões:

- Vinculado — Correlaciona o incidente a uma investigação existente e envia uma mensagem orientadora para essa investigação com o contexto do novo incidente.
- Ignorado — O incidente corresponde aos critérios de ignorar definidos em uma habilidade e é automaticamente descartado sem investigação. Para obter mais informações, consulte the section called “DevOps Habilidades do agente”.
- Prosseguir — Agenda uma nova investigação independente para o incidente.

Visualizando decisões de triagem

Quando os incidentes são vinculados, a investigação primária recebe uma mensagem de orientação contendo os detalhes do incidente vinculado e o raciocínio da correlação. Em seu aplicativo web do AWS DevOps Agent Space, você verá o status de VINCULADO junto com o raciocínio de correlação explicando por que os incidentes foram vinculados. A investigação primária exibe uma lista de todos os incidentes vinculados, permitindo que você veja o escopo completo dos problemas relacionados que estão sendo investigados em conjunto. Seu sistema externo de tickets (ServiceNow, PagerDuty, etc.) e seu canal de comunicação (Slack) receberão uma notificação de que o incidente foi vinculado ao raciocínio da correlação.

Quando os incidentes são ignorados, o aplicativo web AWS DevOps Agent Space exibe o status IGNORADO junto com o motivo que explica por que o incidente foi filtrado. Seu sistema de tickets

externo e seu canal de comunicação também recebem uma notificação de que o incidente foi ignorado junto com o motivo do cancelamento.

Corrigindo decisões de triagem

Se o AWS DevOps Agent vincular incorretamente um incidente, você poderá desvinculá-lo manualmente por meio do aplicativo web AWS DevOps Agent Space. Isso reprograma o incidente não vinculado como uma investigação independente. Você também pode fornecer regras de correlação personalizadas criando uma habilidade de AWS DevOps agente contendo sua lógica de correlação e associando-a ao estágio de triagem.

Se o AWS DevOps Agent ignorar incorretamente um incidente, você poderá cancelá-lo manualmente por meio do aplicativo web AWS DevOps Agent Space. Isso reprograma o incidente para investigação. Para ajustar quais incidentes são ignorados, modifique ou desative a habilidade que define os critérios de ignorar.

Peça apoio humano

AWS DevOps O agente pode se conectar diretamente ao AWS Support para agilizar seu processo de resposta a incidentes. Quando precisar de ajuda adicional do AWS Support, a partir do seu aplicativo web DevOps Agent Space, você pode criar casos de suporte que compartilham automaticamente o contexto da investigação com os engenheiros do AWS Support, reduzindo o tempo necessário para explicar seu problema.

Como funciona

Ao investigar um incidente, o AWS DevOps Agent cria um registro abrangente de sua análise, incluindo:

- Descobertas da investigação da causa raiz
- Métricas, registros e traços analisados
- Alterações de código e histórico de implantação revisados
- Ações de remediação recomendadas
- Cronograma de eventos e comportamento do sistema

Você pode encaminhar sua investigação para o AWS Support diretamente do aplicativo web AWS DevOps Agent Space. Ao fazer isso, o AWS DevOps Agent passa automaticamente seu registro

de investigação para o AWS Support, fornecendo ao engenheiro de suporte um contexto completo sobre sua investigação, sem exigir que você reúna e explique manualmente os detalhes.

Conversando com AWS Suporte

Depois de criar um caso de suporte, você pode se comunicar com o AWS Support em uma janela de bate-papo separada no seu aplicativo web AWS DevOps Agent Space. Isso permite a você:

- Discuta seu problema com os engenheiros do AWS Support junto com o cronograma de investigação do seu AWS DevOps agente
- Veja a análise automatizada do AWS DevOps agente e a orientação especializada do AWS suporte na mesma interface
- Compartilhe facilmente informações ou esclarecimentos adicionais conforme necessário

A experiência de bate-papo mantém a investigação do AWS DevOps agente e a conversa com o AWS Support prontamente acessíveis, permitindo colaboração e resolução mais rápidas.

Idioma do caso de suporte

Quando você cria um caso de suporte por meio do AWS DevOps Agent, o caso é criado automaticamente no idioma configurado na configuração de idioma de resposta do agente do Agent Space. Isso garante que seu caso de suporte seja encaminhado para um engenheiro de suporte que fale seu idioma preferido.

Por exemplo, se o idioma do seu Agent Space estiver definido como japonês, seu caso de suporte será encaminhado para um engenheiro de Japanese-speaking suporte. Se nenhum idioma estiver configurado ou se o idioma configurado não for suportado pelo AWS Support para a categoria de caso selecionada, o padrão do caso será inglês.

AWS Atualmente, o Support suporta os seguintes idiomas para roteamento de casos: chinês, inglês, francês, japonês, coreano, português e espanhol. Para alterar o idioma usado nos casos de suporte, atualize a configuração do idioma de resposta do agente na sua configuração do Espaço do agente. Para obter mais informações, consulte the section called “Criação de um espaço de agente”.

Requisitos do plano de suporte

Sua capacidade de criar e interagir com casos de suporte por meio do AWS DevOps Agent depende do seu plano de AWS Support. Consulte o guia do usuário do Support Plans para saber mais sobre seus direitos.

Observação Os clientes do Suporte Básico não podem criar casos de suporte técnico e, portanto, não podem encaminhar as investigações do AWS DevOps Agente para AWS o Support. Os clientes do Suporte ao Desenvolvedor podem criar casos por meio do AWS DevOps Agent, mas devem visitar a AWS Central de Suporte para se comunicar com os engenheiros de suporte, pois o Suporte ao Desenvolvedor não inclui suporte baseado em bate-papo. Todos os outros planos podem usar a experiência de bate-papo integrada no Agent. AWS DevOps Para obter detalhes completos sobre os direitos do plano de suporte, incluindo tempos de resposta e gravidade de casos disponíveis, consulte o Guia do usuário do AWS Support Plans.

Com quais informações são compartilhadas AWS Suporte

Quando você cria um caso de suporte a partir do aplicativo web AWS DevOps Agent Space, as seguintes informações são compartilhadas automaticamente com o AWS Support:

- Cronograma da investigação: Registro cronológico da análise do agente AWS DevOps
- Informações sobre o recurso: AWS Recursos afetados
- Dados de observabilidade: métricas, registros e traços relevantes de suas ferramentas de monitoramento integradas
- Mudanças recentes: implantações de código, mudanças na infraestrutura e atualizações de configuração
- Tentativas de remediação: Ações recomendadas pelo AWS DevOps Agente
- Avaliação de impacto: escopo e gravidade do incidente

Todos os dados compartilhados com o AWS Support seguem suas configurações existentes de residência e segurança de AWS dados. AWS DevOps O agente compartilha somente informações relacionadas à sua investigação específica e respeita as políticas de governança de dados da sua organização.

Introdução

Para usar a integração do AWS DevOps Agent's AWS Support:

1. Certifique-se de ter um plano de AWS Support ativo.
2. Verifique se as permissões do IAM do seu AWS DevOps agente incluem a criação de casos de suporte (support:CreateCase, support:DescribeCases).

3. Quando o AWS DevOps Agente estiver investigando um problema e você precisar de assistência do AWS Support, escolha Solicitar suporte humano no seu aplicativo web do DevOps Agent Space.
4. Analise o resumo da investigação que será compartilhado com o AWS Support.
5. Selecione a gravidade apropriada do caso com base nos direitos do seu plano de suporte.
6. Envie o caso - O AWS DevOps agente inclui automaticamente seu registro de investigação.

A janela de bate-papo é aberta automaticamente, permitindo que você comece a colaborar com o AWS Support imediatamente.

Prevenção proativa de incidentes

AWS DevOps O agente analisa padrões em suas investigações de incidentes para fornecer recomendações direcionadas que melhoram continuamente sua postura operacional e evitam futuros incidentes. Acesse a prevenção proativa de incidentes por meio da página de melhorias no Operator Web App.

Como funciona a prevenção proativa de incidentes

AWS DevOps O agente avalia investigações recentes de incidentes para identificar melhorias duradouras para evitar futuros incidentes e acelerar o tempo médio de detecção (MTTD). O agente analisa vários incidentes para identificar recomendações que possam evitar classes inteiras de incidentes no futuro, concentrando-se nas recomendações mais impactantes para garantir que sejam acionáveis.

Por padrão, o agente executa automaticamente as avaliações semanalmente. Você pode pausar o cronograma se preferir executar avaliações somente sob demanda. As avaliações manuais estão sempre disponíveis, o que é útil quando uma investigação recente garante uma rápida resposta às melhorias recomendadas.

O agente identifica melhorias em quatro categorias, mostradas no gráfico Categorização de Recomendações na página Melhorias:

- Observabilidade — Recomendações para aprimorar o monitoramento, os alertas, o registro e a visibilidade do sistema para detectar problemas com mais rapidez e precisão.
- Infraestrutura — recomendações para otimizar as configurações de recursos, o ajuste de capacidade e a resiliência arquitetônica.

- Governança — Recomendações para fortalecer os processos de implantação, melhorias no pipeline, práticas de teste e controles operacionais.
- Otimização de código — Recomendações para melhorar a qualidade do código do aplicativo, o tratamento de erros e a resiliência do código.

Essa categorização ajuda você a entender onde suas melhorias operacionais são mais necessárias e permite que você priorize as recomendações com base nas áreas de foco da sua equipe.

Benefícios

- Evite incidentes recorrentes — Aborde as causas básicas de forma sistemática, em vez de responder repetidamente aos mesmos tipos de problemas
- Reduza o trabalho operacional — liberte sua equipe do combate repetitivo a incêndios para se concentrar na inovação e nas melhorias estratégicas
- Melhore a resiliência do sistema — Fortaleça sua infraestrutura, observabilidade e processos de implantação com base em dados reais de incidentes
- Aprenda com os padrões históricos — Aproveite os insights de incidentes anteriores para fazer melhorias direcionadas que tenham o maior impacto

Resumo do agente

O resumo do agente na página de melhorias do aplicativo Web fornece uma descrição dos resultados da última avaliação de incidentes recentes. O resumo explica o número de investigações de incidentes analisadas, quais incidentes são semelhantes aos anteriores e quais recomendações foram criadas ou atualizadas com novas informações.

O resumo ajuda você a entender rapidamente o que o agente descobriu durante sua avaliação mais recente e destaca as recomendações mais notáveis que podem ter o maior impacto em sua postura operacional.

Controlando as avaliações

Você pode controlar quando o AWS DevOps Agente avalia incidentes e gera recomendações:

- Executando avaliações manualmente — Clique no botão Executar agora na página Melhorias para iniciar uma avaliação imediatamente. Isso é útil quando uma investigação recente garante uma rápida reviravolta nas melhorias recomendadas.

- Interromper avaliações ativas — Clique no botão Interromper avaliação na página Melhorias para interromper uma avaliação que está em andamento no momento.

Gerenciando recomendações

AWS DevOps O agente fornece recomendações na página de melhorias, onde você pode analisá-las e gerenciá-las:

- Visualizando detalhes da recomendação — Clique em uma recomendação para abrir a página de detalhes da recomendação, onde você pode ver mais informações sobre a melhoria sugerida, incluindo os incidentes que informaram a recomendação, os impactos esperados e as próximas etapas. Para recomendações com alterações de código, você também pode visualizar a especificação pronta para agente que pode ser entregue a um agente de codificação para implementação.
- Manter — Clique em “Manter” para manter uma recomendação em sua lista de pendências para rastreamento. Isso permite monitorar quais melhorias você planeja implementar e acompanhar seu progresso.
- Descartar — Clique em “Descartar” para remover uma recomendação da sua lista de pendências. Ao descartar uma recomendação, você pode fornecer uma explicação em linguagem natural do motivo pelo qual ela não atende às suas necessidades. O agente aprende com esse feedback e o usa para embasar recomendações futuras, garantindo que elas se tornem mais alinhadas com suas prioridades e requisitos operacionais ao longo do tempo.
- Implementada — Clique em “Implementada” para marcar uma recomendação como concluída. Isso ajuda você a monitorar quais melhorias foram aplicadas e permite que o agente meça a eficácia de suas recomendações ao longo do tempo.
- Remoção automática — As recomendações que não foram marcadas como Manter ou Implementadas podem ser removidas após aproximadamente 6 semanas se nenhum novo incidente tivesse sido evitado com a implementação da recomendação. Isso garante que a página de melhorias se concentre nas melhorias mais relevantes para seus desafios operacionais.
- Atualizações de recomendações — As recomendações existentes são atualizadas quando são encontrados novos incidentes que teriam sido evitados pela recomendação. As atualizações podem alterar a prioridade da recomendação ou refinar a recomendação com base em novos insights.

Priorização de recomendações

AWS DevOps O agente classifica automaticamente suas recomendações por prioridade para ajudar você a se concentrar primeiro nas melhorias mais impactantes. A classificação considera o contexto específico da sua equipe, os padrões operacionais e a gravidade dos problemas abordados por cada recomendação.

Como funciona a priorização

Em cada ciclo de avaliação, o agente classifica suas recomendações ativas (aquelas em um estado proposto ou mantido) usando uma combinação de:

- AI-powered classificação — O agente avalia a importância relativa de suas principais recomendações com base na relevância da categoria, gravidade do incidente e impacto operacional.
- Pontuação determinística — Para atrasos maiores, o agente aplica uma pontuação de prioridade com base na frequência de incidentes, nos padrões de gravidade e na atualidade para garantir pedidos consistentes além dos itens mais bem classificados.

A lista classificada aparece na página Melhorias com uma posição de classificação numérica (1 sendo a prioridade mais alta). As recomendações que foram descartadas ou implementadas não são classificadas.

Personalizando prioridades

Você pode influenciar a forma como o agente classifica as recomendações comunicando as prioridades da sua equipe por meio da interface de bate-papo:

- Definindo preferências de categoria — informe ao agente quais categorias de recomendação são mais importantes para sua equipe (por exemplo, “Priorizamos melhorias de observabilidade em vez de mudanças na infraestrutura”). O agente armazena essas preferências e as usa em futuras avaliações de classificação.
- Fornecendo contexto — compartilhe informações sobre projetos futuros, requisitos de conformidade ou áreas de foco da equipe. O agente incorpora esse contexto ao determinar quais recomendações devem ser priorizadas.

Para atualizar suas preferências, use a interface de bate-papo e descreva as prioridades da sua equipe em linguagem natural. O agente confirmará que entendeu e aplicará suas preferências no próximo ciclo de avaliação.

Estabilidade de classificação

As classificações de recomendação podem mudar entre os ciclos de avaliação quando:

- Novas recomendações são adicionadas com maior prioridade do que as existentes
- As preferências declaradas da sua equipe mudam
- Novos dados de incidentes fortalecem ou enfraquecem a necessidade de uma recomendação

As recomendações que você já marcou como Keep mantêm sua posição em sua lista de pendências, independentemente das mudanças de classificação, garantindo que seu fluxo de trabalho não seja interrompido.

Agent-ready especificações

Para recomendações que envolvam alterações no código ou na configuração, o AWS DevOps Agente pode gerar uma especificação pronta para o agente. Essa especificação fornece um documento estruturado que pode ser entregue diretamente a um agente de codificação para implementação.

A especificação inclui:

- Declaração do problema — Um resumo do problema e sua causa raiz
- Resumo da solução — Uma descrição de alto nível da abordagem recomendada
- Repositórios de destino — Os repositórios específicos em que as alterações precisam ser feitas
- Alterações no código — descrições detalhadas do que precisa ser mudado e por quê, com caminhos de arquivo específicos e considerações de implementação
- Requisitos de teste — Quais cenários precisam ser testados
- Plano de implementação — Uma abordagem em fases para implementar as mudanças

Agent-ready as especificações aceleram a implementação fornecendo aos agentes de codificação o contexto de que precisam para fazer alterações prontas para a produção sem exigir muitas idas e vindas com engenheiros.

Implementando recomendações

Para maximizar o valor das recomendações proativas de prevenção de incidentes, considere as seguintes práticas para agir de acordo com elas:

- Usando especificações prontas para agentes — Para recomendações com alterações de código, use a especificação gerada para acelerar a implementação, entregando-a a um agente de codificação ou usando-a como um guia detalhado para a implementação manual.
- Adicionar recomendações à sua lista de pedidos — Copie as recomendações para o sistema de emissão de tíquetes ou para a ferramenta de gerenciamento de projetos da sua equipe para garantir que elas sejam priorizadas junto com outros trabalhos de engenharia.
- Priorizando recomendações com base no impacto — Concentre-se primeiro nas recomendações que abordam os tipos de incidentes mais frequentes ou graves, ou aqueles que afetam sistemas críticos.
- Acompanhamento do progresso da implementação — Monitore quais recomendações foram implementadas e meça sua eficácia observando se incidentes semelhantes diminuem com o tempo.
- Coordenação com as equipes de desenvolvimento — Compartilhe recomendações com as equipes apropriadas que são proprietárias dos sistemas afetados, garantindo que elas tenham o contexto e os recursos necessários para implementar melhorias.

DevOps Tarefas sob demanda

AWS DevOps O Agent On Demand Tasks é um assistente de conversação com inteligência artificial generativa (IA) que permite que as equipes de operações consultem sua arquitetura de aplicativos, analisem a integridade do sistema e acessem insights de investigação usando linguagem natural. Você pode fazer perguntas sobre seus AWS recursos, métricas do sistema, status do alarme, histórico de implantação e padrões de incidentes. O Chat fornece respostas imediatas com base em seus dados reais de infraestrutura e operações, eliminando a necessidade de navegar entre vários AWS consoles ou ferramentas de monitoramento.

O chat é integrado em todo o aplicativo web do DevOps Agent Space e fornece respostas contextuais com base na página que você está visualizando. A interface mantém o histórico de conversas, permitindo que você continue as discussões anteriores e se baseie nas consultas anteriores.

Capacidades de tarefas

AWS DevOps O Agent On Demand Tasks fornece recursos abrangentes para ajudá-lo a gerenciar e entender sua infraestrutura:

Consultas de recursos — pergunte sobre AWS os recursos em seu espaço do agente, incluindo funções Lambda, tabelas do DynamoDB, implantações do EKS, certificados e configurações de infraestrutura. O Chat pode filtrar e analisar recursos com base em atributos como versões de tempo de execução, configurações de capacidade ou status de implantação. Por exemplo, pergunte “Quantos Lambdas estão usando o Python 3.8?” ou “Tenho algum certificado prestes a expirar?”

Análise da integridade do sistema — consulte as métricas atuais e históricas da integridade do sistema, incluindo status do alarme, taxas de erro, utilização da CPU e disponibilidade do serviço. O Chat pode gerar resumos de saúde cobrindo períodos específicos e identificar tendências no comportamento do sistema. Faça perguntas como “Quais alarmes dispararam nas últimas 24 horas?” ou “Algum erro 5xx na última hora?”

Insights de investigação — acesse informações de investigações concluídas e em andamento, incluindo análise da causa raiz, hipóteses exploradas, registros revisados e padrões de resolução. O Chat pode identificar causas comuns de incidentes e fornecer recomendações com base em dados históricos. Consulta “Qual é a causa mais comum de incidentes no mês passado?” ou “Qual é o tempo médio de resolução de investigações concluídas?”

Orientação da investigação — Ao visualizar uma página de detalhes da investigação, oriente a investigação orientando o agente a se concentrar em registros específicos, explorar hipóteses específicas ou atualizar a análise da causa raiz. Forneça informações orientadoras, como “Concentre-se nos registros do serviço de pagamento e atualize seu RCA” ou “Explore a hipótese de que a limitação do DynamoDB causou o problema”.

Artefatos de bate-papo — Gere relatórios e documentos estruturados, como resumos de integridade operacional, relatórios de erros e análises de incidentes. Os artefatos aparecem em um painel dedicado e oferecem suporte à edição de versões na conversa.

Anexos de arquivo — Anexe imagens, documentos e arquivos de código às suas mensagens para que o Chat possa analisá-las em contexto. Por exemplo, anexe uma captura de tela de um painel de alarme, um arquivo de configuração YAML ou um PDF do runbook e pergunte ao Chat o que fazer a seguir. Consulte Envio de anexos de arquivo para obter detalhes.

Filtragem de recomendações — consulte recomendações de prevenção de incidentes com critérios específicos, como recomendações relacionadas a serviços específicos ou questões operacionais.

O Chat explica o impacto e as considerações de implementação de cada recomendação. Por exemplo, “Mostre-me recomendações que evitarão incidentes envolvendo o DynamoDB” ou “Quais recomendações me ajudariam a detectar problemas de latência de solicitações mais rapidamente?”

Acessando o Chat

O bate-papo está disponível como um painel persistente no lado esquerdo do aplicativo web DevOps Agent Space. A barra lateral esquerda inclui um botão + Novo bate-papo, uma seção de páginas para navegar até Incidentes, melhorias e topologia e uma seção de bate-papos que exibe suas conversas recentes. Escolha Exibir tudo para ver seu histórico completo de conversas.

O Chat fornece respostas contextuais com base em onde você o acessa:

Topologia — Faça perguntas gerais sobre seus recursos, arquitetura e integridade operacional do Agent Space. O Chat tem visibilidade total de todas as contas e serviços conectados. Nesse contexto, você pode consultar configurações de recursos, histórico de implantação, informações de topologia e integrações de ferramentas de observabilidade.

Resposta a incidentes — Ao visualizar a página de resposta a incidentes, faça perguntas sobre tendências de investigação, tempos de resolução e padrões de incidentes em seu Espaço do Agente. O Chat pode analisar dados históricos da investigação para identificar causas comuns e oportunidades de melhoria.

Detalhe da investigação — Ao visualizar uma investigação específica, o Chat fornece respostas contextuais sobre essa investigação. Pergunte sobre registros revisados, hipóteses exploradas, conclusões sobre a causa raiz e planos de mitigação. Você também pode fornecer informações de orientação para orientar o foco da investigação.

Prevenção — na página de prevenção, consulte recomendações com filtros, entenda por que as recomendações foram feitas e explore as abordagens de implementação. O Chat ajuda você a priorizar e entender o impacto das recomendações de prevenção de incidentes.

A interface de bate-papo permanece disponível à medida que você alterna entre as páginas, mas o contexto muda para fornecer informações relevantes para sua visualização atual. Quando você inicia uma nova conversa, ela começa sem contexto prévio. Quando você continua uma conversa existente, o Chat mantém o histórico completo da conversa para perguntas complementares.

Context-aware respostas

O Chat adapta suas respostas com base na página que você está visualizando no aplicativo web do DevOps Agent Space. Essa percepção do contexto garante que você receba informações relevantes sem precisar especificar sobre qual investigação ou escopo de recurso você está perguntando.

Ao visualizar a página de detalhes de uma investigação, o Chat automaticamente entende que você está perguntando sobre aquela investigação específica. Perguntas como “Quais registros você viu?” ou “Quais hipóteses você explorou?” consulte a investigação exibida atualmente. Quando você fornece informações de orientação, o Chat as aplica à investigação ativa e cria uma nova versão da causa raiz, se apropriado.

Na página de prevenção, o Chat entende que você está interessado nas recomendações de prevenção de incidentes. As consultas filtram e analisam automaticamente as recomendações dentro do contexto do seu Espaço do Agente. O sistema reconhece se você está perguntando sobre recomendações gerais ou detalhes específicos da recomendação.

Ao acessar o Chat a partir da página de Topologia, o Chat fornece ampla visibilidade de todos os recursos, métricas e dados históricos em seu Espaço do Agente. Você pode perguntar sobre qualquer recurso, serviço ou preocupação operacional sem especificar o contexto da investigação ou da recomendação.

Essa percepção do contexto elimina a necessidade de especificar repetidamente qual investigação, recomendação ou escopo de recurso você está referenciando, criando um fluxo de conversação mais natural.

Gerenciar conversas

O Chat mantém o histórico de conversas para permitir que você continue as discussões anteriores e faça referência a consultas anteriores.

Criação de novas conversas — Clique no botão “Nova sessão” no painel de bate-papo para iniciar uma nova conversa sem contexto prévio. Novas conversas não transferem informações de bate-papos anteriores, permitindo que você faça perguntas não relacionadas sem confusão.

Acessando o histórico de conversas — Clique em “Histórico” para ver todas as conversas anteriores no seu Espaço do Agente. As conversas são organizadas cronologicamente com registros de data e hora e texto de pré-visualização. O histórico de conversas é retido por 90 dias e é privado para sua conta de usuário no Espaço do Agente.

Conversas contínuas — Selecione qualquer conversa do seu histórico para continuar de onde você parou. O Chat mantém o contexto completo das mensagens anteriores, permitindo que você faça perguntas complementares que façam referência a partes anteriores da conversa. Quando você troca de página enquanto visualiza uma conversa, o contexto da conversa permanece, mas o contexto específico da página é atualizado com base na sua localização atual.

Observe que o histórico de conversas é isolado em cada Espaço do Agente. As conversas em um Espaço do Agente não são visíveis nem acessíveis a partir de outros Espaços do Agente. Esse isolamento garante que as informações confidenciais permaneçam compartimentadas de acordo com seus limites organizacionais.

Gerando artefatos

AWS DevOps O agente suporta artefatos de bate-papo — documentos estruturados e versionados gerados pelo agente durante uma conversa. Os artefatos fornecem um painel dedicado e interativo na interface do usuário do chat para revisar e editar AI-generated conteúdo, como relatórios operacionais, resumos de erros e avaliações de saúde.

Você pode solicitar artefatos de qualquer página no aplicativo web do DevOps Agent Space. O Chat usa o contexto da página atual para definir o escopo do conteúdo do artefato.

Como os artefatos funcionam

Quando você pede que o Chat crie ou atualize conteúdo, o Chat gera um artefato — normalmente um documento formatado — e o exibe no painel de artefatos ao lado da conversa.

Gerar — Envie uma solicitação de linguagem natural para criar um relatório ou documento. Por exemplo, pergunte “Gere um relatório de integridade operacional semanal para meu Agent Space” ou “Mostre-me um relatório dos meus 4xx erros da semana passada”.

Análise — O artefato aparece em um painel dedicado ao lado da conversa. Você pode revisar o conteúdo completo enquanto continua interagindo com o Chat.

Editar — Solicite alterações no artefato por meio do Chat. Por exemplo, pergunte “Adicionar uma seção sobre partidas a frio do Lambda” ou “Atualizar o relatório para incluir os dados do mês passado”. O Chat cria uma nova versão do artefato com as alterações solicitadas.

Enviando anexos de arquivo

Você pode anexar arquivos às suas mensagens de bate-papo para que o Chat possa lê-las junto com sua pergunta. Use anexos para compartilhar o que você está vendo — uma captura de tela de

um painel ou alarme, um arquivo de configuração, código-fonte, um caderno operacional — e peça ao agente que raciocine diretamente sobre isso.

Os arquivos têm como escopo o seu Espaço do Agente — eles não são visíveis em outros Espaços do Agente e o acesso é limitado pelas mesmas permissões do IAM que bloqueiam o resto do Chat. Os arquivos são enviados para o armazenamento gerenciado do Agent Space assim que você os anexa.

Como anexar arquivos

Você pode adicionar arquivos a uma mensagem de três maneiras:

- Escolha o ícone de upload na barra de ferramentas de entrada de bate-papo e selecione um ou mais arquivos do seu dispositivo.
- Arraste e solte um ou mais arquivos na área de entrada do bate-papo.
- Cole uma imagem diretamente da sua área de transferência, por exemplo, depois de fazer uma captura de tela.

Cada arquivo anexado aparece como um chip na entrada do bate-papo com um indicador de progresso do upload. Para visualizar um arquivo, escolha seu chip. Para remover um arquivo, escolha o X no chip. O botão Enviar permanece desativado enquanto qualquer arquivo anexado ainda está sendo carregado.

Tipos de arquivos compatíveis

O Chat aceita as três categorias de arquivos a seguir:

- Imagens —`png`,`jpeg`,`jpg`,`gif`, `webp`
- Documentos — `pdf`,`csv`,`doc`,`docx`,`xls`,`xlsx`,`html`,`txt`, `md`
- Arquivos de texto e código — `json` `yaml`,`yml`,`xml`,`js`,`ts`,`py`,`java`,`rb`, `gor`,`s`,`h`,`sh`,`bash`,`log`,`cfg`,`ini`, `toml`

Arquivos fora dessas categorias são rejeitados antes do upload.

Limites

Os seguintes limites se aplicam a cada mensagem:

Limite	Valor
Tamanho máximo do arquivo	3,75 MB
Anexos por mensagem (qualquer combinação de tipos)	20
Desses, documentos binários (PDF, DOC, DOCX, XLS, XLSX)	até 5

Além disso, o texto da mensagem e o conteúdo do anexo devem caber na janela de contexto por mensagem do modelo. Se uma mensagem e seus anexos forem muito grandes, o Chat rejeitará a mensagem e solicitará que você reduza o tamanho ou o número de anexos antes de enviá-la.

Casos de uso

Maneiras comuns de usar anexos de arquivo com o DevOps Agente:

- Anexe uma captura de tela de um painel de alarme ou erro e peça ao Chat que interprete o que está falhando e onde procurar em seguida.
- Anexe o código-fonte do serviço e peça ao Chat que analise a alteração, sugira correções ou explique seu comportamento.
- Anexe um arquivo de configuração (por exemplo, uma configuração YAML, JSON ou TOML) e peça ao Chat que solucione por que uma implantação, alarme ou integração está se comportando mal.
- Anexe um caderno operacional ou um relatório pós-incidente em PDF e peça ao Chat que o converta em uma habilidade. O agente extrai o procedimento e o salva no seu Espaço do Agente para que futuras investigações possam aplicá-lo automaticamente.

Exemplos de consultas

Os exemplos a seguir demonstram os tipos de perguntas que você pode fazer ao Chat. Esses exemplos são organizados por caso de uso e contexto.

Consultas de geração de artefatos

De qualquer página no aplicativo web do DevOps Agent Space:

- Gere um resumo semanal da saúde operacional para meu Agent Space
- Crie um relatório de todos os erros 4xx da semana passada
- Crie um relatório resumido de incidentes dos últimos 30 dias
- Crie um resumo da atividade de alarme para o serviço de pagamento nesta semana
- Gere um relatório do histórico de implantação dos últimos 7 dias
- Resuma todas as recomendações abertas em um relatório

Consultas de informações sobre recursos

De qualquer página no aplicativo web do DevOps Agent Space:

- Quantas funções do Lambda estão usando o Python 3.8?
- Tenho algum certificado prestes a expirar?
- Listar todas as tabelas do DynamoDB com faturamento sob demanda
- Mostre-me clusters EKS em produção
- Quais funções do Lambda não foram implantadas nos últimos 90 dias?
- Listar buckets S3 sem o controle de versão ativado
- Quais instâncias do RDS estão executando a versão X do banco de dados?

Consultas de integridade do sistema

Nas páginas de topologia ou resposta a incidentes:

- Quais alarmes dispararam nas últimas 24 horas?
- Algum erro 5xx na última hora?
- Mostre-me as tendências de erro do Lambda para o serviço de pagamento
- Qual é a utilização da CPU para meu cluster ECS?
- Há algum alvo insalubre em meus balanceadores de carga?
- Mostre-me os eventos de limitação do API Gateway de ontem
- Quais serviços tiveram a maior taxa de erro na semana passada?
- Me dê um relatório geral de saúde cobrindo as últimas 24 horas

Consultas de ferramentas de observabilidade

Da topologia:

- Listar grupos de registros do Splunk
- Mostre-me as métricas do Prometheus e seus limites de alarme
- Quais monitores do Datadog estão configurados para esse serviço?
- Listar políticas de alerta da New Relic
- Mostre-me as configurações do painel do Dynatrace

Consultas sobre insights de investigação

Da página de resposta a incidentes:

- Qual é a causa mais comum de incidentes no mês passado?
- Qual é o tempo médio de resolução das investigações concluídas?
- Resuma as investigações da semana passada e seu RCA
- Quantos incidentes foram causados pela limitação do DynamoDB?
- Mostre-me as tendências da investigação no último trimestre
- Quais serviços têm os incidentes mais frequentes?

Consultas detalhadas da investigação

Da página de detalhes da investigação:

- Quais registros você viu?
- Quais hipóteses você explorou?
- Quão arriscada é a ação mitigadora que você propõe?
- Qual foi o cronograma dos eventos durante esse incidente?
- Por que você concluiu que essa era a causa raiz?
- Quais evidências apoiam sua análise de causa raiz?
- Quem orientou durante sua investigação?
- Me dê um resumo desta investigação do incidente

Consultas de orientação da investigação

Da página de detalhes da investigação:

- Concentre-se nos registros do serviço de pagamento entre 14:00-15:00 UTC e atualize seu RCA
- Explore a hipótese de que a limitação do DynamoDB causou o problema
- Verifique a configuração do cluster ECS para ver se isso causou o alarme
- Verifique somente os registros das últimas 2 horas, não do dia inteiro
- Investigue o aumento de erros às 15h
- Veja os registros do API Gateway em vez dos registros do Lambda

Consultas de recomendação de prevenção

Da página de Prevenção:

- Quais são minhas três principais recomendações de prevenção de incidentes?
- Mostre-me recomendações que evitarão incidentes envolvendo o DynamoDB
- Quais recomendações me ajudariam a detectar problemas de latência de solicitações com mais rapidez?
- Listar melhorias de observabilidade que poderiam evitar incidentes semelhantes
- Mostre-me recomendações de infraestrutura para o serviço de pagamento
- Quais recomendações têm o maior impacto na resiliência do sistema?

Ativando o Chat no seu Espaço do Agente

O chat está disponível em todos os aplicativos web do DevOps Agent Space. O processo de configuração depende se você tem um Espaço do Agente novo ou existente.

Novos espaços para agentes

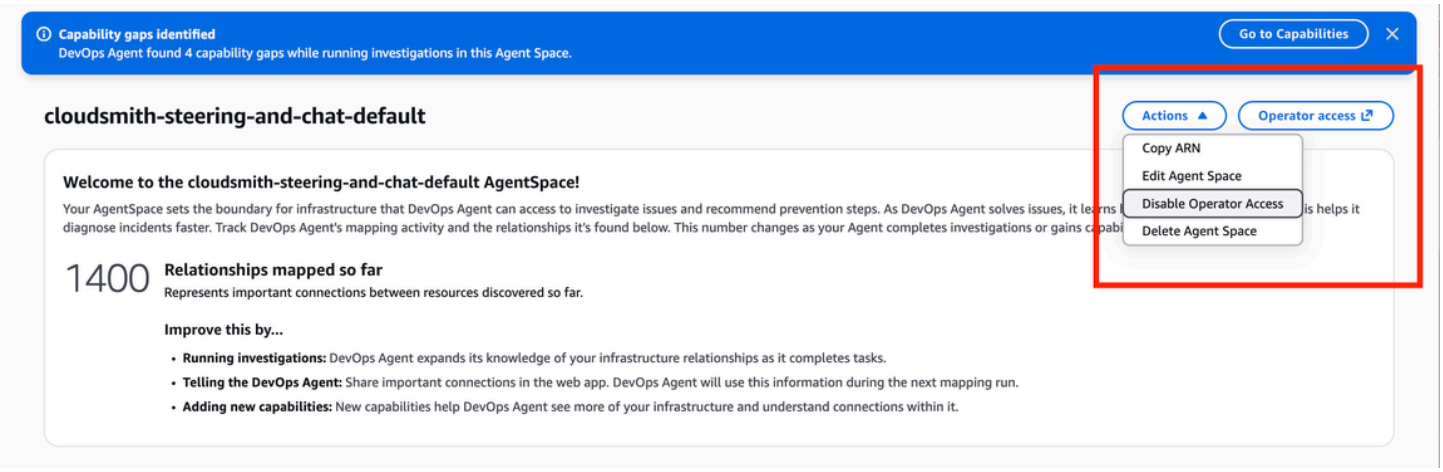
O chat é ativado automaticamente quando você cria um novo Espaço do Agente. Nenhuma configuração adicional ou configuração de permissões do IAM é necessária. Depois de configurar seu aplicativo web DevOps Agent Space, o Chat fica imediatamente disponível como um painel persistente no lado esquerdo de qualquer página.

Espaços de agentes existentes

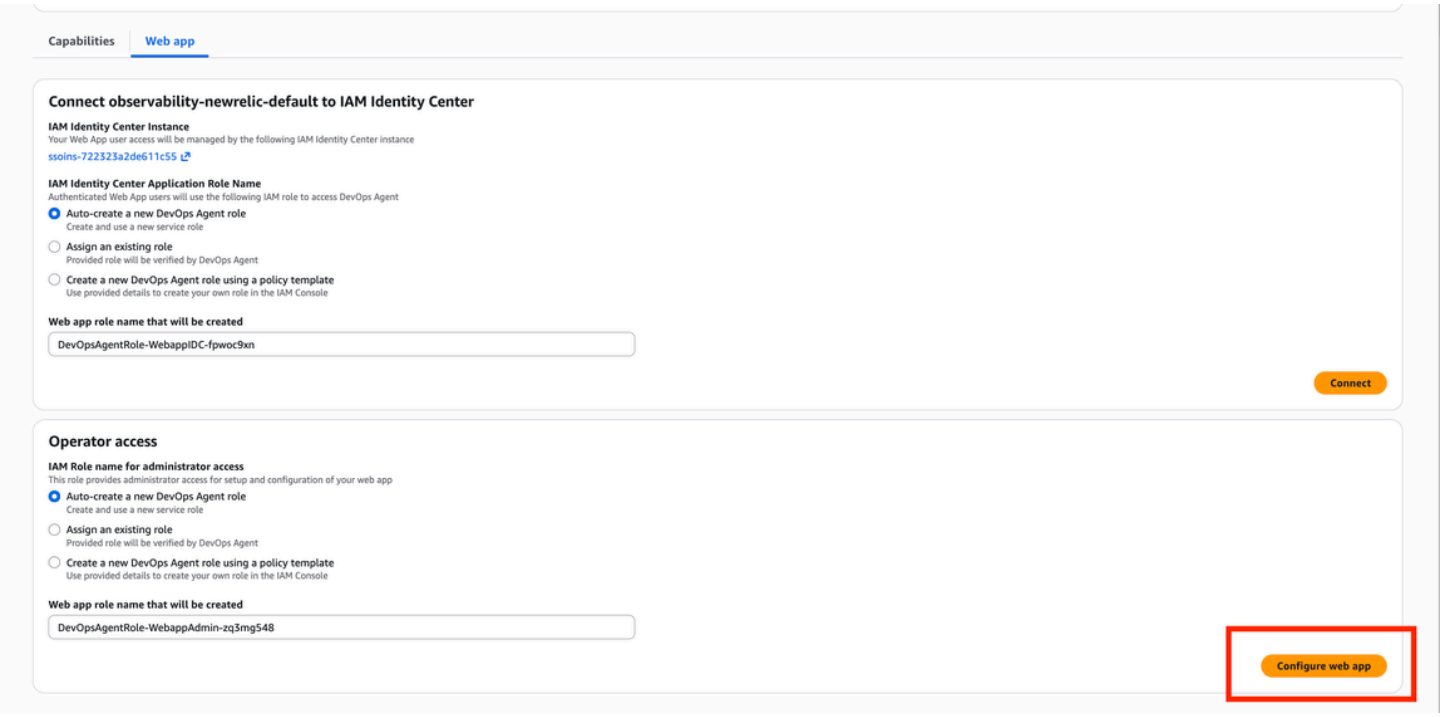
Se você criou seu Espaço do Agente antes do lançamento do Chat, você deve habilitar as permissões necessárias do IAM. Você tem duas opções:

Opção 1: revogar e reativar o acesso ao aplicativo do operador

Navegue até o AWS DevOps Agent Admin Console, localize o menu suspenso Ação no canto superior direito e desative a configuração atual de acesso do operador.



Em seguida, ative a opção de criação automática para acesso do operador.



Isso aplica automaticamente as permissões necessárias do IAM para o Chat junto com todas as outras permissões atuais do operador.

Opção 2: adicionar permissões do IAM manualmente

Adicione as seguintes permissões do IAM à sua função de acesso de operador existente:

- `aidevops:ListChats`— Exibir histórico de conversas de bate-papo
- `aidevops:CreateChat`— Crie novas conversas de bate-papo
- `aidevops:SendMessage`— Envie mensagens e receba respostas

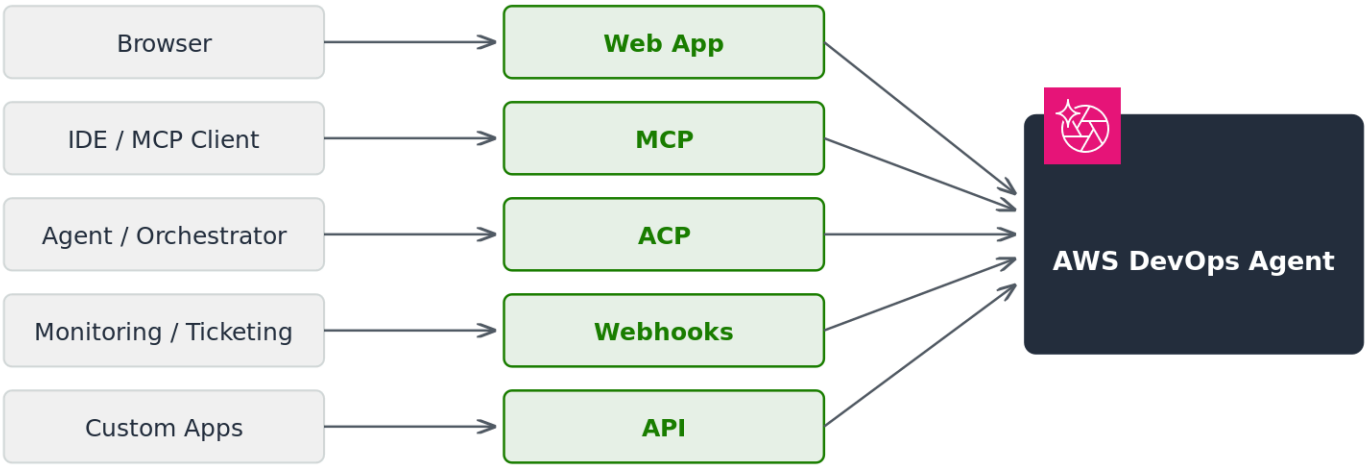
Navegue até o console AWS do IAM, localize sua função de operador de DevOps agente e adicione essas permissões à política de função. O bate-papo fica disponível imediatamente após a adição das permissões.

Depois de concluir qualquer uma das opções, atualize seu aplicativo web do DevOps Agent Space e o painel de bate-papo aparecerá no lado esquerdo de qualquer página.

Interface com o agente DevOps

AWS DevOps O Agent oferece suporte a cinco métodos de acesso: console do aplicativo web, integração do Model Context Protocol (MCP), integração do Agent Client Protocol (ACP), webhooks para automação orientada por eventos e acesso direto à API. Escolha o método que melhor se adapta ao seu fluxo de trabalho e aos requisitos técnicos.

O diagrama a seguir ilustra esses métodos de acesso e como eles se conectam ao serviço de DevOps agente.



DevOps Aplicativo web do agente

O aplicativo web é a interface principal do DevOps Agent. Use o bate-papo conversacional para investigar incidentes, consultar sua infraestrutura e gerenciar recomendações. Para obter mais informações, consulte the section called “O que é um DevOps Agent Web App?”.

Integração do Model Context Protocol (MCP)

Você pode acessar os recursos do AWS DevOps Agente diretamente de MCP-compatible clientes e IDEs. Use o servidor AWS MCP para se conectar. Você pode investigar incidentes, otimizar custos, revisar a arquitetura e mapear a topologia sem sair do seu ambiente de desenvolvimento.

Para usuários do Kiro, um poder dedicado do aws-devops-agent está disponível no repositório Kiro powers. Esse poder conecta Kiro ao AWS DevOps Agente por meio do Servidor AWS MCP. Ele fornece inteligência AI-powered operacional diretamente em seu IDE.

Para usuários do Claude Code, o sample-aws-devops-agent-claude-plugin fornece um plug-in pré-configurado que conecta o Claude Code ao Agent por meio do MCP Server. AWS DevOps AWS

Integração do Agent Client Protocol (ACP)

Você pode invocar o AWS DevOps Agente programaticamente usando o Agent Client Protocol (ACP). Para ver um exemplo de implementação, consulte o repositório sample-aws-devops-agent-acp-mcp em. GitHub

Webhooks

Os webhooks permitem que sistemas externos acionem automaticamente as investigações do AWS DevOps agente. Sistemas externos, como plataformas de emissão de bilhetes e ferramentas de monitoramento, podem enviar solicitações HTTP quando ocorrem incidentes. Para obter mais informações, consulte the section called “Invocando o DevOps Agente por meio do Webhook”.

AWS DevOps API do agente

AWS DevOps O agente fornece APIs para acesso programático aos recursos do agente. Você pode criar e gerenciar Agent Spaces, acionar investigações e recuperar descobertas. Para obter mais informações, consulte a Referência da API do AWS DevOps agente.

Configurando recursos para AWS DevOps Agente

AWS DevOps Os recursos do agente ampliam a funcionalidade do seu agente conectando-o às suas ferramentas e infraestrutura existentes. Configure esses recursos para permitir uma investigação abrangente de incidentes, fluxos de trabalho de resposta automatizados e integração perfeita com seu DevOps ecossistema.

Os recursos a seguir ajudam você a maximizar a eficácia do seu DevOps agente:

- AWS Configuração do EKS Access - Permita a introspecção de clusters, registros de pods e eventos de cluster do Kubernetes para ambientes EKS públicos e privados
- Integração com o Azure — Conecte as assinaturas do Azure e as DevOps organizações do Azure para investigar os recursos do Azure e correlacionar as implantações do Azure DevOps com incidentes
- CI/CD Integração de pipeline - Connect GitHub e GitLab pipelines para correlacionar implantações com incidentes e rastrear alterações de código durante investigações
- Conexões de servidor MCP - Estenda os recursos de investigação conectando ferramentas externas de observabilidade e sistemas de monitoramento personalizados por meio do Model Context Protocol
- Multi-Account AWS Acesso - Configure AWS contas secundárias para investigar recursos em toda a organização durante a resposta a incidentes
- Integração de fontes de telemetria — Conecte plataformas de monitoramento como Datadog, Dynatrace, Grafana, New Relic e Splunk para acesso abrangente aos dados de observabilidade
- Integração de emissão de tíquetes e bate-papo - Connect ServiceNow e Slack para automatizar fluxos de trabalho de resposta a incidentes e permitir a colaboração em equipe PagerDuty
- Configuração do webhook - Permita que sistemas externos acionem automaticamente as investigações do DevOps agente por meio de solicitações HTTP. Para obter detalhes sobre a configuração do webhook, os métodos de autenticação e o formato da solicitação, consultethe section called “Invocando o DevOps Agente por meio do Webhook”.
- EventBridge Integração com a Amazon — incorpore o AWS DevOps agente em aplicativos orientados por eventos roteando eventos do ciclo de vida de investigação e mitigação para os alvos da Amazon EventBridge

Você pode configurar cada recurso de forma independente com base nas necessidades específicas da sua equipe e na pilha de ferramentas existente. Comece com as integrações mais importantes

para seu fluxo de trabalho de resposta a incidentes e, em seguida, expanda para recursos adicionais conforme necessário.

Migração da versão prévia pública para a disponibilidade geral

Se você usou o AWS DevOps Agent durante a pré-visualização pública, você deve atualizar suas funções do IAM antes do lançamento do GA. Este guia explica como atualizar as funções de monitoramento e as funções do operador em suas contas.

O que está mudando

1. Os históricos de bate-papo sob demanda durante a pré-visualização não estão mais acessíveis
2. Novas políticas gerenciadas substituem as políticas disponíveis durante a versão prévia
3. Os Agent Spaces podem ter um escopo de acesso ao aplicativo IAM Identity Center desatualizado

Histórico de bate-papo sob demanda a partir da pré-visualização pública

A versão GA introduz medidas de segurança adicionais para fortalecer os controles de acesso aos históricos de bate-papo. Como resultado dessas mudanças, os históricos de bate-papo sob demanda do período de pré-visualização pública (antes de 30 de março de 2026) não estão mais acessíveis. Os diários de investigação e as descobertas criadas durante a pré-visualização pública não são afetados. Essa alteração se aplica somente às conversas de bate-papo sob demanda.

Novas políticas gerenciadas

Para o GA, AWS fornece novas políticas gerenciadas que substituem as políticas da era de pré-visualização:

Tipo de função	Remover	Adicionar
Monitoramento	Política gerenciada pelo `AIOpsAssistantPolicy`	Política gerenciada pelo `AIDevOpsAgentAccessPolicy`
Operador (IAM e IDC)	Política em linha	Política gerenciada pelo `AIDevOpsOperatorAppAccessPolicy`

Além disso, as funções do operador exigem políticas de confiança atualizadas, e as funções do operador da IDC exigem uma nova política em linha.

Pré-requisitos

- Acesso às AWS contas em que suas funções de DevOps agente estão configuradas (contas primárias e todas as secundárias)
- Permissões do IAM para modificar funções, políticas e relações de confiança
- Seu ID do Agent Space, ID da AWS conta e região (visíveis no console do DevOps agente)

Etapa 1: atualizar as funções de monitoramento

Atualize a função de monitoramento em sua conta principal e em cada conta secundária. Essas são as funções de Primary/Secondary origem configuradas na guia Capacidades em seu espaço de agente (exemplo de primary/secondary função:`DevOpsAgentRole-AgentSpace-3xj2396z`).

1. No console do DevOps agente, acesse seu Espaço do agente e escolha a guia Capacidades.
2. Encontre a função de monitoramento de suas Primary/Secondary fontes (por exemplo,`DevOpsAgentRole-AgentSpace-3xj2396z`) e escolha Editar.
3. Em Políticas de permissões, remova a política `AIOpsAssistantPolicy` AWS gerenciada.
4. Escolha Adicionar permissões, Anexar políticas e anexar a política `AIDevOpsAgentAccessPolicy` gerenciada.
5. Edite a política em linha e substitua seu conteúdo pelo seguinte, substituindo o ID da sua conta:

```
{
    "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. A política de confiança para a função de monitoramento não exige alterações. Verifique se ele corresponde ao seguinte:

```
{
    "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 as etapas de 2 a 6 para a função de monitoramento em cada conta secundária.

Etapa 2: atualizar a função do operador (IAM)

1. No console do DevOps agente, escolha a guia Acesso e encontre a função do operador.
2. No console do IAM, remova a política embutida existente da função de operador.
3. Escolha Adicionar permissões, Anexar políticas e anexar a política `AIDevOpsOperatorAppAccessPolicy` gerenciada.
4. Escolha a guia Relações de confiança e escolha Editar política de confiança. Substitua a política de confiança pela seguinte, substituindo seu ID da conta, região e ID do espaço do agente:

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

Etapa 3: Atualizar as funções do operador (IDC)

Se você usa o IAM Identity Center com o DevOps Agent, atualize cada função de operador da IDC.

1. No console do IAM, acesse Roles e pesquise `WebappIDC` para encontrar suas funções do DevOps Agent IDC (por exemplo,`DevOpsAgentRole-WebappIDC-<id>`).
2. Para cada função da IDC:

a. Remova a política embutida existente.

b. Escolha Adicionar permissões, Anexar políticas e anexar a política `AIDevOpsOperatorAppAccessPolicy` gerenciada.

c. Escolha a guia Relações de confiança e escolha Editar política de confiança. Substitua a política de confiança pela seguinte, substituindo seu ID da conta, região e ID do espaço do agente:

```
{
    "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. Crie uma nova política em linha com as seguintes permissões, substituindo o ID da sua conta:

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

Reconecte o IAM Identity Center (se aplicável)

Os Agent Spaces criados durante a pré-visualização pública podem ter um aplicativo IAM Identity Center configurado com um escopo de acesso desatualizado. Para GA, o escopo correto é **`aidevops:read_write`**. Se seu aplicativo do IAM Identity Center tiver o escopo anterior (**`awsaidevops:read_write`**), você deverá desconectar e reconectar o IAM Identity Center.

Como verificar o escopo do aplicativo do IAM Identity Center

Execute o seguinte comando da AWS CLI para verificar o escopo do seu aplicativo do IAM Identity Center. Você pode encontrar o ARN do aplicativo no console do IAM Identity Center em Aplicativos.

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

A saída deve mostrar o escopo correto **`aidevops:read_write`**:

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

Se o escopo aparecer **`awsaidevops:read_write`**, ele está desatualizado. Siga as etapas abaixo para atualizá-lo.

Como reconectar o IAM Identity Center

O escopo de acesso em um aplicativo AWS gerenciado do IAM Identity Center não pode ser atualizado diretamente. Você deve desconectar e reconectar:

1. No console do AWS DevOps agente, acesse seu Espaço do agente e escolha a guia Acesso.
2. Escolha Desconectar ao lado da configuração do IAM Identity Center.
3. Confirme a desconexão.
4. Escolha Connect para configurar o IAM Identity Center novamente. O serviço cria um novo aplicativo do IAM Identity Center com o escopo correto.
5. Reatribua usuários e grupos ao novo aplicativo no console do IAM Identity Center.

> **Important**
>
> A desconexão remove o bate-papo individual do usuário e o histórico de artefatos associados às contas de usuário do IAM Identity Center. Os usuários precisarão fazer login novamente após a reconexão.

Verificação

Depois de concluir todas as etapas:

1. Retorne ao console do DevOps Agente e verifique se nenhum erro de permissão aparece na guia Acesso ao Espaço do Agente.

2. Teste o aplicativo web do operador para confirmar se ele carrega e funciona corretamente.
3. Se você usa o IDC, verifique se os usuários podem autenticar e acessar a experiência do operador.

Solução de problemas

Erros de permissão negada após a migração

- Verifique se `AIOpsAssistantPolicy` foi removido e `AIDevOpsAgentAccessPolicy` está vinculado às funções de monitoramento.
- Verifique se as políticas embutidas antigas foram removidas e `AIDevOpsOperatorAppAccessPolicy` se estão vinculadas às funções do operador.
- Verifique se as políticas de confiança da operadora incluem`sts:TagSession`.
- Confirme se você substituiu todos os valores de espaço reservado (`<account-id><region>`,,`<agentspace-id>`) por valores reais.

Contas secundárias não funcionam

- A função de monitoramento de cada conta secundária deve ser atualizada de forma independente. Faça login em cada conta e repita a Etapa 1.

Falhas de autenticação do IDC

- Verifique se a política de confiança da IDC inclui tanto a `sts:TagSession` declaração`sts:AssumeRole`/quanto a `TrustedIdentityPropagation` declaração.
- Confirme se a política em linha com`sso:ListInstances`,`sso:DescribeInstance`, e `identitystore:DescribeUser` foi criada.

Histórico de bate-papo sob demanda ausente após a migração

- Os históricos de bate-papo sob demanda do período de pré-visualização pública não estão acessíveis após o lançamento do GA. Esse é o comportamento esperado devido às medidas de segurança aprimoradas introduzidas no GA. Os diários de investigação e as descobertas da prévia pública não são afetados.

AWS Configuração de acesso ao EKS

Você pode permitir que o AWS DevOps Agente investigue problemas em seus clusters do Amazon EKS executando `kubectl` comandos somente de leitura em clusters públicos e privados. Você pode conectar qualquer número de clusters EKS ao mesmo Agent Space.

Depois de conectado, o agente pode ajudar a diagnosticar problemas operacionais em seus clusters, descrevendo recursos, recuperando registros do pod, inspecionando eventos do cluster, verificando a integridade dos nós e muito mais. O agente não pode criar, modificar ou excluir nenhum recurso no seu cluster.

Pré-requisitos

Antes de configurar o acesso ao EKS, certifique-se de que o modo de autenticação do seu cluster EKS inclua a API EKS. Você pode verificar isso na guia Acesso no console do Amazon EKS. Se o modo não incluir a API EKS, selecione um modo que inclua antes de continuar.

Configuração

Essas etapas precisam ser concluídas no console do Amazon EKS para cada cluster para o qual você deseja criar uma entrada de acesso. Você pode encontrar o ARN da função do IAM no seu Espaço do agente (consultethe section called “Criação de um espaço de agente”) em Capacidades > Nuvem > Fonte primária > Editar.

1. Vá até a guia Acesso. Se o modo de autenticação já indicar API EKS, você poderá adicionar entradas de acesso. Caso contrário, selecione um modo que inclua a API EKS.
2. Na guia Acesso, crie uma nova entrada de acesso do IAM. Copie o ARN da função do IAM da fonte de nuvem primária e insira-o como o principal do IAM para a entrada de acesso. Clique em Next.
3. Selecione a política de AIOps AssistantPolicy acesso AWS gerenciada da Amazon e selecione Cluster para o escopo de acesso. (Como alternativa, se você quiser que o agente acesse apenas determinados namespaces, selecione os namespaces Kubernetes desejados). Clique em Adicionar política e, em seguida, clique em Avançar.
4. Analise as alterações e confirme se a política de entrada de acesso e a função do IAM corretas foram escolhidas e crie sua entrada de acesso clicando em “Criar”.

Para verificar se o acesso ao EKS foi configurado corretamente, navegue até o aplicativo Operator e inicie uma nova investigação, fazendo uma pergunta ao agente sobre seu cluster, como “listar todos os pods no namespace padrão” ou “mostrar eventos recentes em meu cluster”.

Solução de problemas

Se o agente não conseguir acessar seu cluster, verifique se a entrada de acesso está usando o ARN correto da função do IAM mostrado na caixa de diálogo de configuração e se a política de AIOps AssistantPolicy acesso da Amazon está anexada.

Conectando o Azure

A integração com o Azure permite que o AWS DevOps Agente investigue recursos em seu ambiente Azure e correlacione implantações de DevOps pipeline do Azure com incidentes operacionais. Ao conectar o Azure, o agente ganha visibilidade em sua infraestrutura do Azure e pode realizar análises de causa raiz em ambos os recursos AWS e no Azure.

A integração com o Azure consiste em dois recursos independentes:

- Recursos do Azure — Permite que o agente descubra e investigue os recursos de nuvem do Azure, como máquinas virtuais, clusters do Azure Kubernetes Service (AKS), bancos de dados e componentes de rede. O agente usa o Azure Resource Graph para consultar seus recursos durante investigações de incidentes.
- Azure DevOps — permite que o agente acesse os DevOps repositórios do Azure e o histórico de execução do pipeline. O agente pode correlacionar mudanças e implantações de código com incidentes para ajudar a identificar possíveis causas-raiz.

Cada recurso é registrado no nível da AWS conta e pode então ser associado a espaços de agentes individuais.

Métodos de registro

AWS DevOps O agente oferece suporte a dois métodos para se conectar ao Azure:

- Consentimento do administrador — Um fluxo simplificado baseado em consentimento em que você autoriza o aplicativo AWS DevOps Agent Entra em seu locatário do Azure. No console, isso aparece como a opção de consentimento do administrador. Esse método requer entrar com uma conta que tenha permissão para realizar o consentimento do administrador no Microsoft Entra ID.

- Registro de aplicativos — Uma abordagem autogerenciada em que você cria seu próprio aplicativo Entra com credenciais de identidade federadas usando o Outbound Identity Federation. No console, isso aparece como a opção Registro do aplicativo. Esse método é adequado quando você precisa de mais controle sobre a configuração do aplicativo ou quando as permissões de consentimento do administrador não estão disponíveis.

Ambos os métodos oferecem os mesmos recursos. Você pode usar um ou os dois métodos na mesma AWS conta.

Limitações conhecidas

- Consentimento do administrador: uma AWS conta por inquilino do Azure — Cada inquilino do Azure só pode ter seu aplicativo AWS DevOps Agent Entra associado a uma AWS conta por vez. Para associar o mesmo inquilino a uma AWS conta diferente, você deve primeiro cancelar o registro existente.
- Registro de aplicativo: aplicativo exclusivo por registro — Cada registro de aplicativo deve usar um aplicativo diferente (ID do cliente). Você não pode registrar várias configurações com o mesmo ID de cliente.
- Azure DevOps: acesso ao código-fonte — A DevOps integração com o Azure fornece acesso ao histórico de execução do pipeline, independentemente de onde o código-fonte esteja hospedado. No entanto, para acessar o código-fonte real, o repositório deve ser conectado separadamente por meio de um provedor de origem compatível (por exemplo,the section called “Conectando GitHub”). O código-fonte hospedado no Bitbucket não pode ser acessado diretamente por meio da DevOps integração com o Azure.

Tópicos

Conectando recursos do Azure

A integração com os Recursos do Azure permite que o AWS DevOps Agente descubra e investigue recursos em suas assinaturas do Azure durante investigações de incidentes. O agente usa o Azure

Resource Graph para descobrir recursos e pode acessar métricas, registros e dados de configuração em seu ambiente do Azure.

Essa integração segue um processo de duas etapas: registrar o Azure no nível da AWS conta e associar assinaturas específicas do Azure a Agent Spaces individuais.

Pré-requisitos

Antes de conectar os Recursos do Azure, verifique se você tem:

- Acesso ao console do AWS DevOps agente
- Uma conta do Azure com acesso à assinatura de destino
- Para o método de consentimento do administrador: uma conta com permissão para realizar o consentimento do administrador no Microsoft Entra ID
- Para o método de registro de aplicativos: um aplicativo Entra com permissões para configurar credenciais de identidade federada e federação de identidade de saída ativada em sua conta AWS

> Observação: você também pode iniciar o registro em um Espaço do Agente. Navegue até Fontes secundárias, clique em Adicionar e selecione Azure. Se o Azure Cloud ainda não estiver registrado, o console o guiará primeiro pelo registro.

Registrando recursos do Azure por meio do consentimento do administrador

O método Admin Consent usa um fluxo baseado em consentimento com o aplicativo gerenciado pelo AWS DevOps agente.

Etapa 1: iniciar o registro

1. Faça login no console AWS de gerenciamento e navegue até o console do AWS DevOps agente
2. Acesse a página Provedores de Capacidades
3. Localize a seção Azure Cloud e clique em Registrar
4. Selecione o método de registro do Admin Consent

Etapa 2: Consentimento completo do administrador

1. Revise as permissões que estão sendo solicitadas
2. Clique para continuar — você será redirecionado para a página de consentimento do administrador do Microsoft Entra

3. Faça login com uma conta principal de usuário que tenha permissão para realizar o consentimento do administrador
4. Revise e conceda consentimento para a inscrição do AWS DevOps Agente

Etapa 3: Concluir a autorização do usuário

1. Após o consentimento do administrador, você receberá uma solicitação de autorização do usuário para verificar sua identidade como membro do inquilino autorizado
2. Entre com uma conta pertencente ao mesmo locatário do Azure
3. Após a autorização, você é redirecionado de volta para o console do AWS DevOps agente com um status de sucesso

Etapa 4: atribuir funções

Consulte Atribuição de funções do Azure abaixo. Procure um AWS DevOps agente ao selecionar membros.

Registrando recursos do Azure por meio do registro de aplicativos

O método de registro do aplicativo usa seu próprio aplicativo Entra com credenciais de identidade federadas.

Etapa 1: iniciar o registro

1. No console do AWS DevOps agente, acesse a página Capability Providers
2. Localize a seção Azure Cloud e clique em Registrar
3. Selecione o método de registro do aplicativo

Etapa 2: Crie e configure seu aplicativo Entra

Siga as instruções exibidas no console para:

1. Ative a federação de identidade de saída em sua AWS conta (no console do IAM, acesse Configurações da conta → Federação de identidade de saída)
2. Crie um aplicativo Entra em seu Microsoft Entra ID ou use um existente
3. Configurar credenciais de identidade federada no aplicativo

Etapa 3: fornecer detalhes do registro

Preencha o formulário de inscrição com:

- ID do inquilino — Seu identificador de inquilino do Azure
- Nome do inquilino — Um nome de exibição para o inquilino
- ID do cliente — O ID do aplicativo (cliente) do aplicativo Entra que você criou
- Público — O identificador de público para a credencial federada

Etapa 4: criar a função do IAM

Uma função do IAM será criada automaticamente quando você enviar o registro pelo console. Ele permite que o AWS DevOps Agente assuma as credenciais e invoque. `sts:GetWebIdentityToken`

Etapa 5: atribuir funções

Consulte Atribuição de funções do Azure abaixo. Pesquise o aplicativo Entra que você criou ao selecionar membros.

Etapa 6: Concluir o registro

1. Confirme a configuração no console do AWS DevOps agente
2. Clique em Enviar para concluir o registro

Atribuição de funções do Azure

Após o registro, conceda ao aplicativo acesso de leitura à sua assinatura do Azure. Essa etapa é a mesma para os métodos de consentimento do administrador e registro do aplicativo.

1. No Portal do Azure, navegue até sua assinatura de destino
2. Vá para Controle de acesso (IAM)
3. Clique em Adicionar > Adicionar atribuição de função
4. Selecione a função Leitor e clique em Avançar
5. Clique em Selecionar membros, pesquise o aplicativo (AWS DevOps Agente para consentimento do administrador ou seu próprio aplicativo Entra para registro do aplicativo)
6. Selecione o aplicativo e clique em Revisar + atribuir

7. (Opcional) Para permitir que o agente acesse clusters do Azure Kubernetes Service (AKS), conclua a seguinte configuração de acesso ao AKS.

   Requisito de segurança: O responsável pelo serviço deve receber somente a função de Leitor (e, opcionalmente, as funções somente de leitura do AKS listadas abaixo). A função Reader serve como um limite de segurança que restringe o agente a operações somente de leitura e limita o impacto de ataques indiretos de injeção imediata. A atribuição de funções com permissões de gravação ou ação aumenta significativamente o raio de explosão da injeção imediata e pode resultar no comprometimento dos recursos do Azure. AWS DevOps O agente executa somente operações de leitura. O agente não modifica, cria nem exclui recursos do Azure.

Configuração de acesso AKS (opcional)

Etapa 1: acesso no nível do Azure Resource Manager (ARM)

Atribua a função de usuário do Azure Kubernetes Service Cluster ao aplicativo.

No Portal do Azure, acesse Assinaturas → selecione assinatura → Controle de Acesso (IAM) → Adicionar atribuição de função → selecione Função de Usuário do Azure Kubernetes Service Cluster → atribuir ao aplicativo (AWS DevOps Agente para Consentimento do Administrador ou seu próprio aplicativo Entra para Registro do Aplicativo).

Isso abrange todos os clusters do AKS na assinatura. Para definir o escopo para clusters específicos, atribua no nível do grupo de recursos ou do cluster individual.

Etapa 2: acesso à API Kubernetes

Escolha uma opção com base na configuração de autenticação do seu cluster:

Opção A: Controle de Acesso Baseado em Função (RBAC) do Azure para Kubernetes (recomendado)

1. Ative o Azure RBAC no cluster, se ainda não estiver habilitado: Portal do Azure → Cluster AKS → Configurações → Configuração de segurança → Autenticação e autorização → selecione Azure RBAC
2. Atribuir função somente para leitura: Portal do Azure → Assinaturas → selecionar assinatura → Controle de acesso (IAM) → Adicionar atribuição de função → selecionar Azure Kubernetes Service RBAC Reader → atribuir ao aplicativo

Isso abrange todos os clusters do AKS na assinatura.

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

Use isso se seu cluster já usa a configuração padrão de autenticação do Azure AD e você prefere não habilitar o RBAC do Azure. Isso requer `kubectl` configuração por cluster.

1. Salve o seguinte manifesto 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>`Substitua pelo ID do objeto do seu diretor de serviço. Para encontrá-lo: Portal do Azure → Inserir ID → Aplicativos corporativos → pesquise o nome do aplicativo (AWS DevOps Agente para consentimento do administrador ou seu próprio aplicativo Entra para registro do aplicativo).

2. Aplique a cada cluster:

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

> Observação: clusters usando somente contas locais (sem o Azure AD) não são compatíveis. Recomendamos habilitar a integração do Azure AD em seu cluster para usar esse recurso.

Função personalizada com menos privilégios (opcional)

Para um controle de acesso mais rígido, você pode criar uma função personalizada do Azure com escopo exclusivo para os provedores de recursos que o AWS DevOps Agente usa, em vez da função ampla do Reader:

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

```
  "NotDataActions": [],
  "AssignableScopes": [
    "/subscriptions/{your-subscription-id}"
  ]
}
```

Associando uma assinatura a um Agent Space

Depois de registrar o Azure no nível da conta, associe assinaturas específicas aos seus Agent Spaces:

1. No console do AWS DevOps agente, selecione seu Espaço do agente
2. Vá para a guia Capacidades
3. Na seção Fontes secundárias, clique em Adicionar
4. Selecione Azure
5. Forneça a ID de assinatura para a assinatura do Azure que você deseja associar
6. Clique em Adicionar para concluir a associação

Você pode associar várias assinaturas ao mesmo Espaço do Agente para dar visibilidade ao agente em todo o seu ambiente do Azure.

Gerenciando conexões do Azure Resources

- Visualizando assinaturas conectadas — Na guia Capacidades, a seção Fontes secundárias lista todas as assinaturas conectadas do Azure.
- Removendo uma assinatura — Para desconectar uma assinatura de um Espaço do Agente, selecione-a na lista Fontes secundárias e clique em Remover. Isso não afeta o registro no nível da conta.
- Removendo o registro — Para remover totalmente o registro do Azure Cloud, acesse a página Capability Providers e exclua o registro. Todas as associações do Agent Space devem ser removidas primeiro.

Conectando o Azure DevOps

DevOps A integração com o Azure permite que o AWS DevOps Agente acesse repositórios e o histórico de execução do pipeline em sua DevOps organização do Azure. O agente pode

correlacionar mudanças e implantações de código com incidentes operacionais para ajudar a identificar possíveis causas-raiz.

> Observação: os DevOps pipelines do Azure podem usar o código-fonte do Azure Repos ou do GitHub Bitbucket. A DevOps integração com o Azure fornece acesso ao histórico de execução do pipeline, independentemente do provedor de origem. No entanto, para acessar o código-fonte real durante as investigações, o repositório deve ser conectado separadamente por meio de uma integração compatível, comothe section called “Conectando GitHub”. O código-fonte no Bitbucket não está diretamente acessível por meio dessa integração.

Essa integração segue um processo de duas etapas: registrar o Azure DevOps no nível da AWS conta e associar projetos específicos a Agent Spaces individuais.

Pré-requisitos

Antes de conectar o Azure DevOps, verifique se você tem:

- Acesso ao console do AWS DevOps agente
- Uma DevOps organização do Azure com pelo menos um projeto contendo um histórico de repositório e pipeline
- Permissões para adicionar usuários à sua DevOps organização do Azure
- Para o método de consentimento do administrador: uma conta com permissão para realizar o consentimento do administrador no Microsoft Entra ID
- Para o método de registro de aplicativos: um aplicativo Entra com permissões para configurar credenciais de identidade federadas e federação de identidade de saída ativada em sua conta AWS

> Observação: você também pode iniciar o registro em um Espaço do Agente. Navegue até a seção Pipelines, clique em Adicionar e selecione Azure DevOps. Se o Azure ainda não DevOps estiver registrado, o console o guiará primeiro pelo registro.

Registrando o Azure DevOps por meio do consentimento do administrador

O método Admin Consent usa um fluxo baseado em consentimento com o aplicativo gerenciado pelo AWS DevOps agente.

Etapa 1: iniciar o registro

1. Faça login no console AWS de gerenciamento e navegue até o console do AWS DevOps agente
2. Acesse a página Provedores de Capacidades
3. Localize a DevOps seção Azure e clique em Registrar
4. Insira o nome DevOps da sua organização do Azure quando solicitado

Etapa 2: Consentimento completo do administrador

1. Clique para continuar - você será redirecionado para a página de consentimento do administrador do Microsoft Entra
2. Faça login com uma conta principal de usuário que tenha permissão para realizar o consentimento do administrador
3. Revise e conceda consentimento para a inscrição do AWS DevOps Agente

Etapa 3: Concluir a autorização do usuário

1. Após o consentimento do administrador, você receberá uma solicitação de autorização do usuário para verificar sua identidade como membro do inquilino autorizado
2. Entre com uma conta pertencente ao mesmo locatário do Azure
3. Após a autorização, você é redirecionado de volta para o console do AWS DevOps agente com um status de sucesso

Etapa 4: conceder acesso no Azure DevOps

Consulte Conceder acesso no Azure DevOps abaixo. Pesquise o AWS DevOps Agente ao adicionar usuários.

Registrando o Azure DevOps por meio do registro de aplicativos

O Registro do Aplicativo é compartilhado entre os Recursos do Azure e o Azure DevOps. Se você já concluiu o Registro do Aplicativo para Recursos do Azure, pode pular para Conceder acesso no Azure. DevOps

Etapa 1: iniciar o registro do aplicativo ADO

1. No console do AWS DevOps agente, acesse a página Capability Providers

2. Localize a seção Azure Cloud e clique em Registrar
3. Selecione o método de registro do aplicativo

Etapa 2: Crie e configure seu aplicativo Entra

Siga as instruções exibidas no console para:

1. Ative a federação de identidade de saída em sua AWS conta (no console do IAM, acesse Configurações da conta → Federação de identidade de saída)
2. Crie um aplicativo Entra em seu Microsoft Entra ID ou use um existente
3. Configurar credenciais de identidade federada no aplicativo

Etapa 3: fornecer detalhes do registro

Preencha o formulário de inscrição com:

- ID do inquilino — Seu identificador de inquilino do Azure
- Nome do inquilino — Um nome de exibição para o inquilino
- ID do cliente — O ID do aplicativo (cliente) do aplicativo Entra
- Público — O identificador de público para a credencial federada

Etapa 4: criar a função do IAM

Uma função do IAM será criada automaticamente quando você enviar o registro pelo console. Ele permite que o AWS DevOps Agente assuma as credenciais e invoque. `sts:GetWebIdentityToken`

Etapa 5: Concluir o registro

1. Confirme a configuração no console do AWS DevOps agente
2. Clique em Enviar para concluir o registro

Etapa 6: Conceder acesso no Azure DevOps

Consulte Conceder acesso no Azure DevOps abaixo. Pesquise o aplicativo Entra que você criou durante o registro do aplicativo ao adicionar usuários.

Concedendo acesso no Azure DevOps

Após o registro, conceda ao aplicativo acesso à sua DevOps organização do Azure. Essa etapa é a mesma para os métodos de consentimento do administrador e registro do aplicativo.

1. No Azure DevOps, vá para Configurações da organização **>** Usuários **>** Adicionar usuários
2. Pesquise o aplicativo (seja AWS DevOps Agent for Admin Consent ou seu próprio aplicativo Entra para registro de aplicativos)
3. Defina o nível de acesso como Básico
4. Em Adicionar aos projetos, selecione os projetos que você deseja que o agente acesse
5. Em DevOps Grupos do Azure, selecione Project Readers
6. Clique em Adicionar para concluir

> Requisito de segurança: atribua somente o grupo de leitores do projeto. O acesso somente leitura serve como um limite de segurança que restringe o agente a operações somente de leitura e limita o impacto de ataques indiretos de injeção imediata. Atribuir grupos com permissões de gravação ou ação aumenta significativamente o raio de explosão da injeção imediata e pode resultar no comprometimento dos recursos do Azure. DevOps

Associando um projeto a um Agent Space

Depois de registrar o Azure DevOps no nível da conta, associe projetos específicos aos seus Agent Spaces:

1. No console do AWS DevOps agente, selecione seu Espaço do agente
2. Vá para a guia Capacidades
3. Na seção Pipelines, clique em Adicionar
4. Selecione Azure DevOps na lista de provedores disponíveis
5. Selecione o projeto na lista suspensa de projetos disponíveis
6. Clique em Adicionar para concluir a associação

Gerenciando DevOps conexões do Azure

- Visualizando projetos conectados — Na guia Capacidades, a seção Pipelines lista todos os DevOps projetos conectados do Azure.

- Removendo um projeto — Para desconectar um projeto de um Espaço do Agente, selecione-o na seção Pipelines e clique em Remover.
- Removendo o registro — Para remover totalmente o DevOps registro do Azure, acesse a página Provedores de Capacidade e exclua o registro. Todas as associações do Agent Space devem ser removidas primeiro.

Conexão a CI/CD tubulações

A integração do pipeline de CI/CD permite que o AWS DevOps Agente monitore implantações e correlacione alterações de código com incidentes operacionais durante as investigações. Ao conectar seus CI/CD provedores, o agente pode rastrear eventos de implantação e associá-los a AWS recursos para ajudar a identificar possíveis causas-raiz durante a resposta a incidentes.

AWS DevOps O Agent oferece suporte à integração com CI/CD plataformas populares por meio de um processo de duas etapas:

1. Registro no nível da conta — Registre seu CI/CD provedor uma vez no nível da AWS conta
2. Conexão com o Agent Space — Conecte projetos ou repositórios específicos a Agent Spaces individuais com base em suas necessidades organizacionais

Essa abordagem permite que você compartilhe registros de CI/CD provedores em vários Agent Spaces, mantendo um controle granular sobre quais projetos são monitorados por cada espaço.

CI/CD Provedores compatíveis

AWS DevOps O Agent oferece suporte às seguintes CI/CD plataformas:

- GitHub— Conecte repositórios de GitHub.com usando o GitHub aplicativo AWS DevOps Agent.
- GitLab— Conecte projetos de GitLab.com, GitLab instâncias gerenciadas ou GitLab implantações auto-hospedadas acessíveis ao público.

Tópicos

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

Conectando GitHub

GitHub a integração permite que o AWS DevOps Agente acesse repositórios de código e receba eventos de implantação durante investigações de incidentes. Essa integração segue um processo de duas etapas: registro em nível de conta GitHub, seguido pela conexão de repositórios específicos a Agent Spaces individuais.

AWS DevOps O agente é compatível com GitHub instâncias.com (SaaS) e GitHub Enterprise Server (auto-hospedadas).

Pré-requisitos

Antes de se conectar GitHub, verifique se você tem:

- Acesso ao console de administração do AWS DevOps agente
- Uma conta de GitHub usuário ou organização com permissões de administrador
- Autorização para instalar GitHub aplicativos em sua conta ou organização

Para o GitHub Enterprise Server, você também precisa:

- Uma instância do GitHub Enterprise Server (versão 3.x ou posterior) acessível por HTTPS
- O URL HTTPS da sua instância do GitHub Enterprise Server (por exemplo,`https://github.example.com`)
- (Opcional) Uma conexão privada, se sua instância do GitHub Enterprise Server não estiver acessível publicamente

Registro GitHub (nível da conta)

GitHub é registrado no nível da AWS conta e compartilhado entre todos os Agent Spaces dessa conta. Você só precisa se registrar GitHub uma vez por AWS conta.

Etapa 1: Navegar até os fornecedores de funil

1. Faça login no console AWS de gerenciamento
2. Navegue até o console do AWS DevOps agente
3. Vá para a guia Capacidades
4. Na seção Pipeline, clique em Adicionar

5. GitHubSelecione na lista de provedores disponíveis

Se GitHub ainda não tiver sido registrado, você será solicitado a registrá-lo primeiro.

Etapa 2: Escolha o tipo de conexão

Na tela “Registrar GitHub conta/organização”, selecione se você está se conectando como usuário ou organização:

- Usuário — Sua GitHub conta pessoal com nome de usuário e perfil
- Organização — Uma GitHub conta compartilhada em que várias pessoas podem colaborar em vários projetos ao mesmo tempo

Se você estiver se conectando a uma instância do GitHub Enterprise Server, marque a caixa de seleção Usar GitHub Enterprise Server e insira a URL HTTPS da sua instância (por exemplo,`https://github.example.com`).

Se sua instância do GitHub Enterprise Server não estiver acessível publicamente, você pode, opcionalmente, configurar uma conexão privada para permitir que o AWS DevOps Agente acesse sua instância com segurança. Para obter mais informações, consulte the section called “Conectando-se a ferramentas hospedadas de forma privada”.

> **Note**
>
> Não `/api/v3` inclua nenhum caminho final no URL — insira somente o URL base.

Etapa 3: configurar o GitHub aplicativo

Clique em Enviar para iniciar o processo de configuração do aplicativo. As próximas etapas são diferentes dependendo se você está se conectando a GitHub .com ou ao GitHub Enterprise Server.

Para GitHub .com

1. Você será redirecionado para GitHub instalar o GitHub aplicativo AWS DevOps Agent.
2. Selecione em qual conta ou organização instalar o aplicativo.
3. O aplicativo permite que o AWS DevOps Agente receba eventos de repositórios conectados, incluindo eventos de implantação.

Para servidor GitHub corporativo

GitHub O Enterprise Server usa um fluxo de manifesto de GitHub aplicativo, que configura automaticamente um novo GitHub aplicativo na sua instância. Isso envolve dois redirecionamentos para sua instância do GitHub Enterprise Server.

1. Seu navegador será redirecionado para a página “Criar GitHub aplicativo” da sua instância do GitHub Enterprise Server.
2. Você verá o nome do aplicativo pré-preenchido. Sinta-se à vontade para alterar o nome conforme necessário. Clique em Criar GitHub aplicativo.
3. Você será redirecionado de volta para o AWS DevOps Agent, que troca o código do manifesto pelas credenciais do aplicativo.

Etapa 4: selecionar repositórios e concluir a instalação

1. Você verá a página Instalar e Autorizar do GitHub aplicativo.
2. Selecione quais repositórios permitir que o aplicativo acesse:
   - Todos os repositórios — Conceda acesso a todos os repositórios atuais e futuros
   - Selecione somente repositórios — Escolha repositórios específicos da sua conta ou organização
3. Clique em Instalar e autorizar.
4. Você será redirecionado de volta ao console do AWS DevOps agente, onde GitHub aparecerá como registrado no nível da conta.

Conectando repositórios a um Espaço do Agente

Depois de se registrar GitHub no nível da conta, você pode conectar repositórios específicos a Agent Spaces individuais:

1. No console do AWS DevOps agente, selecione seu Espaço do agente
2. Vá para a guia Capacidades
3. Na seção Pipeline, clique em Adicionar
4. GitHubSelecione na lista de provedores disponíveis
5. Selecione o subconjunto de repositórios relevantes para esse Espaço do Agente
6. Clique em Adicionar para concluir a conexão

Você pode conectar diferentes conjuntos de repositórios a diferentes Agent Spaces com base nas suas necessidades organizacionais.

Entendendo o GitHub aplicativo

O GitHub aplicativo AWS DevOps Agent:

- Solicita acesso aos seus repositórios — você pode revisar as permissões específicas durante a instalação do GitHub aplicativo
- Recebe eventos de implantação e outros eventos do repositório
- Permite que o AWS DevOps agente correlacione alterações de código com incidentes operacionais
- Pode ser desinstalado a qualquer momento por meio de suas configurações GitHub

Para o GitHub Enterprise Server, o GitHub aplicativo é criado automaticamente na sua instância durante o registro. Você pode gerenciar o acesso ao repositório do aplicativo ou desinstalá-lo em Configurações > Aplicativos > GitHub Aplicativos instalados. Para excluir totalmente a definição do aplicativo, acesse Configurações > Configurações do desenvolvedor > GitHub Aplicativos.

GitHub Atualizações de permissões do aplicativo

AWS DevOps O agente pode solicitar atualizações de permissão após a instalação do GitHub aplicativo para oferecer suporte a novos recursos. Quando isso acontece:

1. Você receberá uma notificação GitHub sobre a solicitação de atualização de permissão.
2. Analise os detalhes da atualização para entender quais novas permissões estão sendo solicitadas.
3. Aceite a solicitação para conceder as permissões atualizadas.

Nenhuma alteração é necessária em seu serviço ou aplicativo. Depois de aceitar as permissões atualizadas, o próximo token de acesso de instalação solicitado pelo AWS DevOps Agente GitHub incluirá automaticamente as novas permissões.

> **Note**
>
> Até que você aceite uma atualização de permissão, o AWS DevOps Agente continua operando com as permissões concedidas anteriormente. Novos recursos que dependem das permissões atualizadas não estarão disponíveis até que você aprove a solicitação.

Gerenciando GitHub conexões

- Atualizando o acesso ao repositório — Para alterar quais repositórios o GitHub aplicativo pode acessar, acesse as configurações da sua GitHub conta ou organização (ou as configurações da instância do GitHub Enterprise Server), navegue até os GitHub aplicativos instalados e modifique a configuração do aplicativo do AWS DevOps Agente.
- Visualizando repositórios conectados — No console do AWS DevOps Agente, selecione seu Espaço do Agente e vá até a guia Capacidades para visualizar os repositórios conectados na seção Pipeline.
- Removendo a GitHub conexão — Para se desconectar GitHub de um Espaço do Agente, selecione a conexão na seção Pipeline e clique em Remover. Para desinstalar completamente o GitHub aplicativo, desinstale-o das configurações da sua GitHub conta ou organização. Para o GitHub Enterprise Server, como o GitHub aplicativo é criado diretamente na sua instância durante o registro, você pode, opcionalmente, limpar o aplicativo por completo executando as duas ações a seguir:
  - Desinstalar o aplicativo — Vá para Configurações > Aplicativos > GitHub Aplicativos instalados, clique em Configurar no aplicativo e, em seguida, desinstale-o.
  - Excluir o aplicativo — Vá para Configurações > Configurações do desenvolvedor > GitHub Aplicativos, selecione o aplicativo, vá até a guia Avançado e escolha Excluir GitHub aplicativo. Aviso: a exclusão do GitHub aplicativo é permanente e não pode ser desfeita. Se você excluí-lo, precisará registrar novamente o GitHub Enterprise Server desde o início no console do AWS DevOps Agente para criar um novo aplicativo.

Conectando GitLab

GitLab a integração permite que o AWS DevOps Agente monitore as implantações do GitLab Pipelines para informar as investigações causais durante a resposta a incidentes. Essa integração segue um processo de duas etapas: registro em nível de conta GitLab, seguido pela conexão de projetos específicos a Agent Spaces individuais.

Registro GitLab (nível da conta)

GitLab é registrado no nível da AWS conta e compartilhado entre todos os Agent Spaces dessa conta. Os Agent Spaces individuais podem então escolher quais projetos específicos se aplicam ao seu Agent Space.

Etapa 1: Navegar até os fornecedores de funil

1. Faça login no console AWS de gerenciamento
2. Navegue até o console do AWS DevOps agente
3. Vá para a página Capability Providers (acessível na navegação lateral)
4. Encontre GitLabna seção Provedores disponíveis em Pipeline e clique em Registrar

Etapa 2: configurar a GitLab conexão

Na página GitLab de registro, configure o seguinte:

Tipo de conexão — Selecione se você está se conectando como pessoa ou grupo:

- Pessoal (padrão) — Sua conta de GitLab usuário individual com nome de usuário e perfil
- Grupo — Em GitLab, você usa grupos para gerenciar um ou mais projetos relacionados ao mesmo tempo

GitLab tipo de instância — Escolha a qual tipo de GitLab instância você está se conectando:

- GitLab.com (padrão) — O GitLab serviço público
- Auto-hospedado publicamente acessível GitLab — marque a caixa Usar endpoint GitLab auto-hospedado e forneça o URL para sua instância GitLab

> **Note**
>
> Atualmente, somente GitLab instâncias acessíveis ao público são suportadas.

Token de acesso — Forneça um token de acesso GitLab pessoal:

1. Em uma guia separada do navegador, faça login na sua GitLab conta

2. Navegue até suas configurações de usuário e selecione Tokens de acesso
3. Crie um novo token de acesso pessoal com as seguintes permissões:
   - `read_repository`— Necessário para acessar o conteúdo do repositório
   - `read_virtual_registry`— Necessário para acessar as informações do registro virtual
   - `read_registry`— Necessário para acessar as informações do registro
   - `api`— Necessário para acesso à API de leitura e gravação
   - `self_rotate`- Necessário para tokens rotativos. No momento, esse recurso não é suportado pelo AWS DevOps Agente, mas será suportado em uma data posterior. Adicionar agora evita a necessidade de criar um novo token no futuro.
4. Defina a expiração do token para um máximo de 365 dias a partir da data atual
5. Copie o token gerado
6. Retornar ao console do AWS DevOps agente
7. Cole o token no campo “Token de acesso”

Etapa 3: Concluir o registro

(Opcional) Tags — Adicione AWS tags ao GitLab registro para fins organizacionais.

Clique em Avançar para revisar sua configuração e, em seguida, clique em Enviar para concluir o processo de GitLab registro. O sistema validará seu token de acesso e estabelecerá a conexão.

Conectando projetos a um Agent Space

Depois de se registrar GitLab no nível da conta, você pode conectar projetos específicos a Agent Spaces individuais:

1. No console do AWS DevOps agente, selecione seu Espaço do agente
2. Vá para a guia Capacidades
3. Na seção Pipeline, clique em Adicionar
4. GitLabSelecione na lista de provedores disponíveis
5. Selecione os GitLab projetos relevantes para o seu Espaço do Agente
6. Clique em Salvar

AWS DevOps O agente monitorará esses projetos para implantações da GitLab Pipelines para informar as investigações causais.

Gerenciando GitLab conexões

- Atualização do token de acesso — Se seu token de acesso expirar ou precisar ser atualizado, você poderá atualizá-lo no console do AWS DevOps agente modificando o GitLab registro no nível da conta.
- Visualização de projetos conectados — No console do AWS DevOps agente, selecione seu Espaço do agente e vá até a guia Capacidades para visualizar os projetos conectados na seção Pipeline.
- Removendo a GitLab conexão — Para desconectar GitLab projetos de um Espaço do Agente, selecione a conexão na seção Pipeline e clique em Remover. Para remover completamente o GitLab registro, primeiro remova-o de todos os Agent Spaces e, em seguida, exclua o registro no nível da conta.

Conectando servidores MCP

Os servidores do Model Context Protocol (MCP) ampliam os recursos de investigação do AWS DevOps Agente fornecendo acesso aos dados de suas ferramentas externas de observabilidade, sistemas de monitoramento personalizados e fontes de dados operacionais. Este guia explica como conectar um servidor MCP ao AWS DevOps Agente.

Requisitos

Antes de conectar um servidor MCP, certifique-se de que seu servidor atenda aos seguintes requisitos:

- Protocolo de transporte HTTP simplificável — Somente servidores MCP que implementam o protocolo de transporte HTTP simplificável são suportados.
- Suporte à autenticação — Seu servidor MCP deve oferecer suporte a um dos seguintes métodos de autenticação: OAuth 2.0 (Credenciais do Cliente ou 3LO), autenticação baseada em chave de API/token ou AWS Signature Version 4 (SigV4).

Considerações sobre segurança

Ao conectar servidores MCP ao AWS DevOps Agente, considere estes aspectos de segurança:

- Lista de permissões de ferramentas — Você deve listar somente as ferramentas específicas de que seu Espaço do Agente precisa, em vez de expor todas as ferramentas do seu servidor MCP.

Consulte Configurando ferramentas MCP em um Espaço do Agente para saber como permitir ferramentas de lista por Espaço do Agente.

Observe que o comprimento máximo da ferramenta de qualquer ferramenta MCP é 64.

- Riscos de injeção imediata — servidores MCP personalizados podem introduzir riscos adicionais de ataques de injeção imediata. Consulte Proteção imediata por injeção: AWS DevOps Agent Security para obter mais informações.
- Ferramentas e acesso somente para leitura — Liste somente as ferramentas MCP somente para leitura e garanta que as credenciais de autenticação tenham acesso somente para leitura.

Consulte AWS DevOps Segurança do agente para obter mais informações sobre injeção imediata e o modelo de responsabilidade compartilhada.

> **Note**
>
> Se o servidor MCP estiver em uma rede privada, consulte the section called “Conectando-se a ferramentas hospedadas de forma privada”

Registrando um servidor MCP (nível de conta)

Os servidores MCP são registrados no nível da AWS conta e compartilhados entre todos os Agent Spaces nessa conta. Os Agent Spaces individuais podem então escolher quais ferramentas específicas precisam de cada servidor MCP.

Etapa 1: detalhes do servidor MCP

1. Faça login no console AWS de gerenciamento
2. Navegue até o console do AWS DevOps agente
3. Vá para a página Capability Providers (acessível na navegação lateral)
4. Encontre o MCP Server na seção Provedores disponíveis e clique em Registrar
5. Na página de detalhes do servidor MCP, insira as seguintes informações:
   - Nome — Insira um nome descritivo para seu servidor MCP
   - URL do endpoint — insira o URL HTTPS completo do endpoint do servidor MCP

- Descrição (opcional) — Adicione uma descrição para ajudar a identificar a finalidade do servidor
- Ativar registro dinâmico de clientes — Marque essa caixa de seleção se quiser permitir que o AWS DevOps Agente se registre automaticamente no servidor de autorização do seu servidor MCP.
- Conectar-se ao endpoint usando conexão privada — Marque essa caixa de seleção se quiser que o AWS DevOps Agente faça solicitações ao seu servidor MCP de forma privada. Você pode selecionar uma conexão privada existente ou criar uma nova. Se você usa OAuth autenticação, a conexão privada se aplica tanto ao endpoint do servidor MCP quanto ao endpoint de troca de tokens. Certifique-se de que a conexão privada esteja configurada com um endereço de host que possa rotear o tráfego para os dois endpoints. Para obter mais informações, consulte the section called “Conectando-se a ferramentas hospedadas de forma privada”.

6. Clique em Avançar.

> **Note**
>
> O URL do endpoint do servidor MCP será exibido nos AWS CloudTrail registros da sua conta.

Etapa 2: fluxo de autorização

Selecione o método de autenticação para seu servidor MCP:

OAuth Credenciais do cliente — Se o servidor MCP usar o fluxo de credenciais OAuth do cliente:

1. Selecione as credenciais OAuth do cliente
2. Clique em Avançar.

OAuth 3LO (Three-Legged OAuth) — Se o seu servidor MCP usa OAuth 3LO para autenticação:

1. Selecione OAuth 3LO
2. Clique em Avançar.

Chave de API — Se seu servidor MCP usa autenticação de chave de API:

1. Selecione a chave de API

2. Clique em Avançar.

AWS SigV4 — Se o seu servidor MCP usa a autenticação AWS Signature Version 4:

1. Selecione AWS SigV4
2. Clique em Avançar.

Etapa 3: configuração da autorização

Configure parâmetros de autorização adicionais com base no método de autenticação selecionado:

Para credenciais OAuth do cliente:

1. ID do cliente — Insira o ID do OAuth cliente
2. Segredo do cliente — Insira o segredo do OAuth cliente
3. URL do Exchange — Insira o URL do endpoint de troca de OAuth tokens
4. Parâmetros de troca — insira os parâmetros de troca de OAuth tokens para autenticação com o serviço
5. Adicionar escopo — Adicione OAuth escopos para autenticação
6. Clique em Avançar.

Para OAuth 3LO:

1. ID do cliente — Insira o ID do OAuth cliente
2. Segredo do cliente — Insira o segredo do OAuth cliente se for exigido pelo seu OAuth cliente
3. URL do Exchange — Insira o URL do endpoint de troca de OAuth tokens
4. URL de autorização - Insira a URL do endpoint de OAuth autorização
5. Suporte ao Code Challenge - Marque essa caixa de seleção se seu OAuth cliente suportar o Code Challenge
6. Adicionar escopo — Adicione OAuth escopos para autenticação
7. Clique em Avançar.

Para chave de API:

1. Insira um nome de chave de API
2. Insira o nome do cabeçalho que conterá a chave de API na solicitação
3. Insira o valor da sua chave de API
4. Clique em Avançar.

Para AWS SigV4:

AWS A autenticação SigV4 permite que o AWS DevOps Agente se conecte aos servidores MCP que usam o AWS Signature Versão 4 para assinatura de solicitações. Isso é útil para servidores MCP hospedados por trás do Amazon API Gateway ou outros AWS serviços que oferecem suporte à autenticação SigV4.

> Nota: As conexões privadas não são suportadas para servidores MCP usando a autenticação SigV4. O endpoint do servidor MCP deve estar acessível ao público. Para servidores MCP em redes privadas usando outros métodos de autenticação, consultethe section called “Conectando-se a ferramentas hospedadas de forma privada”.

1. Configurar a função do IAM — Escolha uma das seguintes opções:
   - Use uma função existente — Selecione uma função do IAM existente no menu suspenso. A função deve ter uma política de confiança que permita que o diretor do serviço do AWS DevOps agente a assuma (consulte Criação de uma função do IAM para autenticação SigV4).
   - Crie uma nova função manualmente — Siga as step-by-step instruções exibidas no console para criar uma nova função do IAM com a política de confiança correta.
2. AWS Região — Insira a AWS região para assinatura SigV4 (por exemplo,`us-east-1`). Para usar a assinatura multirregional SigV4a, digite. *
3. Nome do serviço — Insira o nome do AWS serviço para assinatura SigV4 (por exemplo, `execute-api` para API Gateway).
4. Cabeçalhos personalizados (opcional) — Adicione até 10 pares de cabeçalhos de valores-chave personalizados para incluir em cada solicitação assinada.
5. Clique em Avançar.

Etapa 4: revisar e enviar

1. Revise todos os detalhes da configuração do servidor MCP

2. Clique em Enviar para concluir o registro
3. AWS DevOps O agente validará a conexão com seu servidor MCP
4. Após a validação bem-sucedida, seu servidor MCP será registrado no nível da conta

Configurando ferramentas MCP em um espaço de agente

Depois de registrar um servidor MCP no nível da conta, você pode configurar quais ferramentas desse servidor estão disponíveis para Agent Spaces específicos:

1. No console do AWS DevOps agente, selecione seu Espaço do agente
2. Vá para a guia Capacidades
3. Na seção Servidores MCP, clique em Adicionar
4. Selecione o servidor MCP registrado que você deseja conectar a este Espaço do Agente.
5. Configure quais ferramentas desse servidor MCP devem estar disponíveis para o Espaço do Agente:
   - Permitir todas as ferramentas — Disponibiliza todas as ferramentas do servidor MCP
   - Selecionar ferramentas específicas — Permite que você escolha quais ferramentas serão permitidas na lista.
6. Clique em Adicionar para conectar o servidor MCP ao seu Espaço do Agente.

AWS DevOps Agora, o agente poderá usar as ferramentas da lista de permissões do seu servidor MCP durante as investigações neste Espaço do Agente.

Gerenciando conexões do servidor MCP

Atualização das credenciais de autenticação — Se suas credenciais de autenticação precisarem ser atualizadas, você precisará registrar novamente o servidor MCP. Navegue até a página Capability Providers no console do AWS DevOps Agente, localize seu servidor MCP, remova todas as associações ativas e clique em Cancelar registro. Em seguida, registre seu servidor MCP com as novas credenciais de autenticação e recrie todas as associações necessárias com seu Espaço do Agente.

Visualizando servidores MCP conectados — Para ver todos os servidores MCP conectados ao seu Espaço do Agente, selecione seu Espaço do Agente, vá até a guia Capacidades e verifique a seção Servidores MCP. Você também pode atualizar as ferramentas selecionadas aqui.

Removendo conexões do servidor MCP — Para desconectar um servidor MCP de um Espaço do Agente, selecione o servidor na seção Servidores MCP e clique em Remover. Para excluir completamente um registro do servidor MCP, primeiro remova-o de todos os Agent Spaces e, em seguida, exclua o registro no nível da conta.

Criação de uma função do IAM para autenticação SigV4

Ao usar a autenticação AWS SigV4, o AWS DevOps Agente assume uma função do IAM em sua conta para assinar solicitações no seu servidor MCP. Essa função deve ter uma política de confiança que permita que o AWS DevOps agente principal de serviço (`aidevops.amazonaws.com`) a assuma, com uma proteção adjunta confusa.

Política de confiança

Crie uma função do IAM com a seguinte política de confiança. `REGION`Substitua pela sua AWS região (por exemplo,`us-east-1`) e `ACCOUNT_ID` pelo ID AWS da sua conta.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "aidevops.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "ACCOUNT_ID"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:aidevops:REGION:ACCOUNT_ID:service/*"
        }
      }
    }
  ]
}
```

A política de confiança inclui as seguintes condições para evitar o confuso problema do deputado:

- `aws:SourceAccount`— Restringe a suposição de funções às solicitações provenientes de sua conta. AWS

- `aws:SourceArn`— Restringe a suposição de funções às solicitações provenientes dos recursos de serviço do AWS DevOps Agente em sua conta.

Política de permissões

Anexe uma política de permissões à função que conceda as permissões mínimas necessárias para invocar seu servidor MCP. Por exemplo, se seu servidor MCP estiver hospedado por trás do Amazon API Gateway, a função deverá ter `execute-api:Invoke` permissão no recurso API Gateway.

Assinatura multirregional (SigV4a)

Se o seu servidor MCP estiver implantado em várias AWS regiões, você poderá usar o SIGV4a (Sigv4a, Signature Version 4a) para assinatura multirregional. Para habilitar isso, insira `*` como AWS Região ao configurar a autorização SigV4. O Sigv4a usa assinatura assimétrica, o que permite que uma única solicitação assinada seja válida em várias regiões.

Tópicos relacionados

- Segurança no AWS DevOps Agent
- Configurando um espaço de agente
- Proteção imediata de injeção

Conectando várias AWS contas

AWS As contas secundárias permitem que o AWS DevOps Agente investigue recursos em várias AWS contas em sua organização. Quando seus aplicativos abrangem várias contas, adicionar contas secundárias garante que o agente tenha visibilidade de todos os recursos relevantes durante as investigações de incidentes. O maior acesso às contas e aos recursos que compõem um aplicativo garante maior precisão na investigação.

Pré-requisitos

Antes de adicionar uma AWS conta secundária, verifique se você tem:

- Acesso ao console do AWS DevOps agente na conta principal
- Acesso administrativo à AWS conta secundária
- Permissões do IAM para criar funções na conta secundária

Adicionar uma AWS conta secundária

Além das etapas abaixo, você pode usar o the section called “AWS DevOps Guia de integração do Agent CLI” para adicionar contas secundárias de forma programática.

Etapa 1: iniciar a configuração da conta secundária

1. Faça login no console AWS de gerenciamento e navegue até o console do AWS DevOps agente
2. Selecione seu espaço de agente
3. Vá para a guia Capacidades
4. Na seção Nuvem, localize a subseção Fontes secundárias
5. Clique em Adicionar

Etapa 2: especificar o nome da função

1. No campo Nome da sua função, insira um nome para a função que você criará na conta secundária
2. Anote esse nome: você o usará novamente ao criar a função na conta secundária
3. Copie a política de confiança fornecida no console e salve-a em um espaço rascunho

Etapa 3: criar a função na conta secundária

1. Abra uma nova guia do navegador e faça login no console do IAM na AWS conta secundária
2. Navegue até IAM > Funções > Criar função
3. Selecione Política de confiança personalizada
4. Cole a política de confiança que você copiou da Etapa 2
5. Clique em Avançar.

Etapa 4: anexar a política AWS gerenciada

1. Na seção Políticas de permissões, pesquise AIDevOpsAgentAccessPolicy
2. Marque a caixa de seleção ao lado da política AIDevOpsAgentAccessPolicygerenciada
3. Clique em Avançar.

Etapa 5: nomear e criar a função

1. No campo Nome da função, insira o mesmo nome da função que você forneceu na Etapa 2
2. (Opcional) Adicione uma descrição para ajudar a identificar a finalidade da função
3. Revise a política de confiança e as permissões anexadas
4. Clique em Criar função

Etapa 6: anexar a política em linha

1. No console do IAM, localize e selecione a função que você acabou de criar
2. Vá para a guia Permissões
3. Clique em Adicionar permissões > Criar política em linha
4. Mudar para a guia JSON
5. Cole a política que você salvou na Etapa 2
6. Cole a política no editor JSON no console do IAM
7. Clique em Avançar.
8. Forneça um nome para a política em linha (por exemplo, "DevOpsAgentInlinePolicy“)
9. Clique em Criar política

Etapa 7: Concluir a configuração

1. Retorne ao console do AWS DevOps agente na conta principal
2. Clique em Avançar para concluir a configuração da conta secundária
3. Verifique se o status da conexão é exibido como Ativo

Entendendo as políticas necessárias

AWS DevOps O agente exige três componentes de política para acessar recursos em uma conta secundária:

- Política de confiança — permite que o AWS DevOps agente na conta principal assuma a função na conta secundária. Isso estabelece a relação de confiança entre as contas.

- AIDevOpsAgentAccessPolicy (política AWS gerenciada) — Fornece as principais permissões de somente leitura que o AWS DevOps agente precisa para investigar recursos na conta secundária. Essa política é mantida AWS e atualizada à medida que novos recursos são adicionados.
- Política embutida — fornece permissões adicionais específicas para sua configuração do Agent Space. Essa política é gerada com base nas configurações do seu Espaço do Agente e pode incluir permissões para integrações ou recursos específicos.

Na conta principal, a função do AWS DevOps Agent IAM deve ser capaz de assumir a função criada na conta secundária.

Gerenciando contas secundárias

- Visualização de contas conectadas — Na guia Capacidades, a subseção Fontes secundárias lista todas as contas secundárias conectadas com seu status de conexão.
- Atualização da função do IAM — Se você precisar modificar as permissões, atualize a política embutida anexada à função na conta secundária. As alterações terão efeito imediatamente.
- Removendo uma conta secundária — Para desconectar uma conta secundária, selecione-a na lista Fontes secundárias e clique em Remover. Isso não exclui a função do IAM na conta secundária.

Conectando fontes de telemetria

AWS DevOps O agente fornece três maneiras de se conectar às suas fontes de telemetria.

Integração bidirecional integrada

Atualmente, o AWS DevOps Agent oferece suporte aos usuários do Dynatrace com uma integração bidirecional integrada que permite o seguinte:

- Mapeamento de recursos de topologia - O AWS DevOps agente aumentará sua topologia do espaço do DevOps agente com entidades e relacionamentos disponíveis por meio de um servidor Dynatrace MCP hospedado por um AWS DevOps agente.
- Acionamento automatizado de investigações - Os fluxos de trabalho do Dynatrace podem ser configurados para acionar investigações de resolução de incidentes a partir de problemas do Dynatrace.

- Introspecção de telemetria - O AWS DevOps agente pode fazer uma introspecção da telemetria do Dynatrace enquanto investiga um problema por meio do servidor Dynatrace MCP hospedado pelo agente. AWS DevOps
- Atualizações de status - O AWS DevOps agente publicará as principais descobertas da investigação, análises da causa raiz e planos de mitigação gerados na interface de usuário do Dynatrace.

Para saber mais sobre integrações bidirecionais, consulte

- the section called “Conectando o Dynatrace”

Integração unidirecional integrada

Atualmente, o AWS DevOps Agent oferece suporte AWS CloudWatch aos usuários do Amazon S3, Datadog, Grafana, New Relic e Splunk com integrações unidirecionais integradas.

Prática recomendada de segurança: ao configurar credenciais para integrações unidirecionais integradas, recomendamos definir o escopo das chaves e tokens da API para acesso somente leitura. AWS DevOps O agente usa essas credenciais somente para introspecção de telemetria e não exige acesso de gravação ao seu provedor de telemetria.

A integração unidirecional AWS CloudWatch integrada não requer configuração adicional e permite o seguinte:

- Mapeamento de recursos de topologia - O AWS DevOps agente aumentará sua topologia do espaço do DevOps agente com entidades e relacionamentos disponíveis por meio de suas contas de nuvem primária e secundária AWS configuradas.
- Introspecção de telemetria - O AWS DevOps agente pode fazer uma introspecção da AWS CloudWatch telemetria ao investigar um problema por meio das funções do IAM fornecidas durante a configuração da conta de nuvem primária e secundária. AWS

A integração unidirecional integrada do Amazon S3 permite o seguinte:

- Introspecção de telemetria - O AWS DevOps agente pode ler objetos dos buckets do Amazon S3 enquanto investiga um problema. Isso é útil para acessar registros, arquivos de configuração e outros artefatos armazenados no S3.

Para usar a integração com o Amazon S3, adicione as `s3:ListBucket` permissões `s3:GetObject` e à função IAM do DevOps agente. Seguindo o princípio do privilégio mínimo, defina essas permissões somente para os buckets específicos do S3 que o agente precisa acessar. Para obter mais informações sobre como configurar as permissões do IAM, consultethe section called "DevOps Permissões do Agent IAM".

As integrações unidirecionais integradas do Datadog, Grafana, New Relic e Splunk exigem configuração e permitem o seguinte:

- Acionamento automatizado de investigações - os eventos Datadog, Grafana, New Relic e Splunk podem ser configurados para AWS DevOps acionar investigações de resolução de incidentes do Agente por meio de webhooks do Agent. AWS DevOps
- Introspecção de telemetria - O AWS DevOps agente pode fazer uma introspecção da telemetria Datadog, Grafana, New Relic e Splunk enquanto investiga um problema por meio do servidor MCP remoto de cada provedor.

Para saber mais sobre integrações unidirecionais, veja o seguinte:

- the section called "Conectando DataDog"
- the section called "Conectando a Grafana"
- the section called "Conectando a New Relic"
- the section called "Conectando o Splunk"

Bring-your-own fontes de telemetria

Para qualquer outra fonte de telemetria, incluindo as métricas do Prometheus, você pode aproveitar o suporte do AWS DevOps Agent para integração de webhook e servidor MCP.

Para saber mais sobre bring-your-own integrações, consulte o seguinte

- the section called "Invocando o DevOps Agente por meio do Webhook"
- the section called "Conectando servidores MCP"

Conectando o Dynatrace

Built-in, integração bidirecional

Atualmente, o AWS DevOps Agent oferece suporte aos usuários do Dynatrace com uma integração bidirecional integrada que permite o seguinte:

- Mapeamento de recursos de topologia - O AWS DevOps agente aumentará sua topologia do espaço do DevOps agente com entidades e relacionamentos disponíveis em seu ambiente Dynatrace.
- Acionamento automatizado de investigações - Os fluxos de trabalho do Dynatrace podem ser configurados para acionar investigações de resolução de incidentes a partir de problemas do Dynatrace.
- Introspecção de telemetria - O AWS DevOps agente pode fazer uma introspecção da telemetria do Dynatrace enquanto investiga um problema por meio do servidor MCP do Dynatrace. AWS DevOps Agent-hosted
- Atualizações de status - O AWS DevOps agente publicará as principais descobertas da investigação, análises da causa raiz e planos de mitigação gerados na interface de usuário do Dynatrace.

Pré-requisitos

A integração do AWS DevOps agente com o Dynatrace requer o SaaS da Dynatrace. A integração depende dos recursos da plataforma Dynatrace (fluxos de trabalho, AppEngine aplicativos, incluindo o aplicativo SRE Agents e clientes OAuth) que estão disponíveis somente nos ambientes SaaS da Dynatrace.

O Dynatrace Managed (local) não é suportado, e a Dynatrace não tem planos de trazer esses recursos da plataforma para o Managed. Se você estiver executando o Dynatrace Managed, precisará fazer o upgrade para o Dynatrace SaaS antes de conectá-lo ao Agent. AWS DevOps Consulte Atualização do Dynatrace Managed para SaaS.

Onboarding

Processo de integração

A integração do seu sistema de observabilidade Dynatrace envolve três etapas:

1. Connect - Estabeleça uma conexão com o Dynatrace configurando as credenciais de acesso à conta, com todos os ambientes que você possa precisar
2. Habilitar - Ative o Dynatrace em espaços específicos do Agente com ambientes específicos do Dynatrace
3. Configure seu ambiente Dynatrace - use o aplicativo Dynatrace SRE Agents para concluir a conexão em 2 cliques

Etapa 1: Conectar

Estabeleça conexão com seu ambiente Dynatrace

Configuração

1. Vá para a página Capability Providers (acessível na navegação lateral)
2. Encontre o Dynatrace na seção Provedores disponíveis em Telemetria e clique em Registrar
3. Crie um cliente OAuth no Dynatrace, com as permissões detalhadas.
   a. Veja a documentação do Dynatrace
   b. Quando estiver pronto, pressione Avançar
   c. Você pode conectar vários ambientes do Dynatrace e, posteriormente, o escopo a ambientes específicos para cada espaço de DevOps agente que você possa ter.
4. Insira seus detalhes do Dynatrace na configuração do cliente OAuth:
   - Nome do cliente
   - ID do cliente
   - Segredo do cliente
   - URN da conta
5. Clique em Avançar.
6. Revise e adicione

Etapa 2: Ativar

Ative o Dynatrace em um espaço de agente específico e configure o escopo apropriado

Configuração

1. Na página de espaços do agente, selecione um espaço do agente e pressione visualizar detalhes

2. Selecione a guia Capacidades
3. Localize a seção Telemetria, pressione Adicionar
4. Você notará que o Dynatrace está com o status “Registrado”. Clique em adicionar para adicioná-lo ao seu espaço de agente
5. ID do ambiente Dynatrace - Forneça a ID do ambiente Dynatrace que você gostaria de associar a esse espaço do agente. DevOps
6. Insira uma ou mais IDs de entidade do Dynatrace - elas ajudam o DevOps agente a descobrir seus recursos mais importantes. Exemplos podem ser serviços ou aplicativos. Se você não tiver certeza, pode pressionar remover.
7. Revise e pressione Salvar
8. Copie o URL do Webhook e o Segredo do Webhook. Você os usará no aplicativo Dynatrace SRE Agents para concluir a conexão. Consulte a seção Introdução para obter detalhes.

Etapa 3: Configurar seu ambiente Dynatrace

Para concluir sua configuração do Dynatrace, use o aplicativo Dynatrace SRE Agents para configurar o lado Dynatrace da integração em 2 cliques — nenhuma configuração manual do fluxo de trabalho é necessária. Para obter detalhes, consulte a seção Introdução.

Esquemas de eventos compatíveis

AWS DevOps O agente oferece suporte a dois tipos de eventos do Dynatrace usando webhooks. Os esquemas de eventos compatíveis estão documentados abaixo:

Evento de incidente

Eventos incidentes são usados para acionar uma investigação. O esquema do evento é:

```
{
    "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 mitigação

Os eventos de mitigação são usados para acionar a geração de um relatório de mitigação para a investigação das próximas etapas. O esquema do evento é:

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

Remoção

A fonte de telemetria é conectada em dois níveis no nível do espaço do agente e no nível da conta. Para removê-lo completamente, você deve primeiro removê-lo de todos os espaços do agente em que ele é usado e, em seguida, ele pode ser cancelado.

Etapa 1: Remover do espaço do agente

1. Na página de espaços do agente, selecione um espaço do agente e pressione visualizar detalhes
2. Selecione a guia Capacidades
3. Role para baixo até a seção Telemetria
4. Selecione Dynatrace
5. Pressione remover

Etapa 2: Cancelar o registro da conta

1. Vá para a página Capability Providers (acessível na navegação lateral)
2. Role até a seção Registrado atualmente.
3. Verifique se a contagem de espaço do agente é zero (se não, repita a Etapa 1 acima em seus outros espaços do agente)
4. Pressione Cancelar registro ao lado de Dynatrace

Conectando DataDog

Built-in, integração unidirecional

Atualmente, o AWS DevOps Agent oferece suporte aos usuários do Datadog com integração unidirecional integrada, permitindo o seguinte:

- Acionamento automatizado de investigação - os eventos do Datadog podem ser configurados para acionar investigações de resolução de incidentes do AWS DevOps agente por meio de webhooks do agente. AWS DevOps
- Introspecção de telemetria - O AWS DevOps agente pode fazer uma introspecção da telemetria do Datadog enquanto investiga um problema por meio do servidor MCP remoto de cada provedor.

Onboarding

Etapa 1: Conectar

Estabeleça conexão com seu endpoint MCP remoto Datadog com credenciais de acesso à conta

Configuração

1. Vá para a página Capability Providers (acessível na navegação lateral)
2. Encontre Datadog na seção Provedores disponíveis em Telemetria e clique em Registrar
3. Insira os detalhes do seu servidor Datadog MCP:
   - Nome do servidor - Identificador exclusivo (por exemplo, my-datadog-server)
   - URL do endpoint - Seu endpoint do servidor Datadog MCP. O URL do endpoint varia dependendo do seu site Datadog. Veja a tabela de endpoints do site Datadog abaixo.
   - Descrição - Descrição opcional do servidor
4. Clique em Avançar.
5. Analisar e enviar

Endpoints do site Datadog

O URL do endpoint MCP varia dependendo do seu site Datadog. Para identificar seu site, verifique a URL em seu navegador quando estiver conectado ao Datadog ou consulte Acessar o site do Datadog.

Site Datadog	Domínio do site	URL do endpoint MCP
US1 (padrão)	`datadoghq.com`	`https://mcp.datadoghq.com/api/unstable/mcp-server/mcp`
NÓS 3	`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`
MAPA 1	`ap1.datadoghq.com`	`https://mcp.ap1.datadoghq.com/api/unstable/mcp-server/mcp`
MAPA 2	`ap2.datadoghq.com`	`https://mcp.ap2.datadoghq.com/api/unstable/mcp-server/mcp`

Autorização

Conclua a autorização do OAuth até:

- Autorizando como seu usuário na página OAuth do Datadog
- Se não estiver conectado, clique em Permitir, faça login e autorize

Depois de configurado, o Datadog fica disponível em todos os espaços do agente.

Etapa 2: ativar

Ative DataDog em um espaço de agente específico e configure o escopo apropriado

Configuração

1. Na página de espaços do agente, selecione um espaço do agente e pressione visualizar detalhes (se você ainda não criou um espaço do agente, consultethe section called “Criação de um espaço de agente”)
2. Selecione a guia Capacidades
3. Role para baixo até a seção Telemetria
4. Pressione Adicionar
5. Selecione Datadog
6. Próximo
7. Revise e pressione Salvar
8. Copie o URL do webhook e a chave da API

Etapa 3: configurar webhooks

Usando o URL do Webhook e a chave de API, você pode configurar o Datadog para enviar eventos para acionar uma investigação, por exemplo, a partir de um alarme.

Os webhooks do Datadog usam autenticação de token de portador. Para obter o formato completo da solicitação de webhook, o esquema de carga útil e o código de exemplo, consulte. the section called “Invocando o DevOps Agente por meio do Webhook” Use os exemplos da versão 2 (autenticação de token do portador), definindo o `Authorization: Bearer <Token>` cabeçalho com a chave de API da Etapa 2.

Envie webhooks com o Datadog https://docs.datadoghq.com/integrations/webhooks/(observe que não selecione nenhuma autorização e, em vez disso, use a opção de cabeçalho personalizado).

Saiba mais: Datadog Remote MCP Server

Remoção

A fonte de telemetria é conectada em dois níveis no nível do espaço do agente e no nível da conta. Para removê-lo completamente, você deve primeiro removê-lo de todos os espaços do agente em que ele é usado e, em seguida, ele pode ser cancelado.

Etapa 1: Remover do espaço do agente

1. Na página de espaços do agente, selecione um espaço do agente e pressione visualizar detalhes
2. Selecione a guia Capacidades
3. Role para baixo até a seção Telemetria
4. Selecione Datadog
5. Pressione remover

Etapa 2: Cancelar o registro da conta

1. Vá para a página Capability Providers (acessível na navegação lateral)
2. Role até a seção Registrado atualmente.
3. Verifique se a contagem de espaço do agente é zero (se não, repita a Etapa 1 acima em seus outros espaços do agente)
4. Pressione Cancelar registro ao lado do Datadog

Conectando a Grafana

A integração com o Grafana permite que o AWS DevOps Agente consulte métricas, painéis e dados de alerta da sua instância do Grafana durante investigações de incidentes. Essa integração segue um processo de duas etapas: registro em nível de conta do Grafana, seguido pela conexão com Agent Spaces individuais.

Para melhorar a segurança, a integração com o Grafana permite apenas ferramentas somente de leitura. As ferramentas de gravação estão desativadas e não podem ser habilitadas. Isso significa que o agente pode consultar e ler dados da sua instância do Grafana, mas não pode criar, modificar ou excluir nenhum recurso do Grafana, como painéis, alertas ou anotações. Para obter mais informações, consulte Segurança no AWS DevOps Agente.

Requisitos da Grafana

Antes de conectar a Grafana, certifique-se de ter:

- Grafana versão 9.0 ou posterior. Alguns recursos, principalmente operações relacionadas à fonte de dados, podem não funcionar corretamente com versões anteriores devido à falta de endpoints da API.
- Uma instância do Grafana acessível por HTTPS. Tanto os endpoints de rede pública quanto a privada são suportados. Com conectividade de rede privada, sua instância do Grafana pode ser hospedada em uma VPC sem acesso público à Internet. Para obter detalhes, consulte the section called “Conectando-se a ferramentas hospedadas de forma privada”.
- Uma conta de serviço da Grafana com um token de acesso que tem permissões de leitura apropriadas

Registrando a Grafana (nível da conta)

Grafana é registrada no nível da AWS conta e compartilhada entre todos os Agent Spaces dessa conta.

Etapa 1: Configurar o Grafana

1. Faça login no console AWS de gerenciamento
2. Navegue até o console do AWS DevOps agente
3. Vá para a página Capability Providers (acessível na navegação lateral)
4. Encontre Grafana na seção Provedores disponíveis em Telemetria e clique em Registrar
5. Na página Configurar Grafana, insira as seguintes informações:
   - Nome do serviço (obrigatório) — Insira um nome descritivo para seu servidor Grafana usando somente caracteres alfanuméricos, hífens e sublinhados. Por exemplo, .`my-grafana-server`
   - URL do Grafana (obrigatório) — Insira o URL HTTPS completo da sua instância do Grafana. Por exemplo, .`https://myinstance.grafana.net`
   - Token de acesso à conta de serviço (obrigatório) — Insira um token de acesso à conta de serviço da Grafana. Os tokens geralmente começam com`glsa_`. Para criar um token de conta de serviço, navegue até sua instância do Grafana, acesse Administração > Contas de serviço, crie uma conta de serviço com o papel de Visualizador e gere um token.
   - Descrição (opcional) — Adicione uma descrição para ajudar a identificar a finalidade do servidor. Por exemplo, .`Production Grafana server for monitoring`

6. (Opcional) Adicione AWS tags ao registro para fins organizacionais.
7. Clique em Avançar.

Etapa 2: revisar e enviar o registro da Grafana

1. Revise todos os detalhes de configuração da Grafana
2. Clique em Enviar para concluir o registro
3. Após o registro bem-sucedido, Grafana aparece na seção Atualmente registrado da página Provedores de Capacidades

Adicionando Grafana a um espaço de agente

Depois de registrar o Grafana no nível da conta, você pode conectá-lo a Agent Spaces individuais:

1. No console do AWS DevOps agente, selecione seu Espaço do agente
2. Vá para a guia Capacidades
3. Na seção Telemetria, clique em Adicionar
4. Selecione Grafana na lista de provedores disponíveis
5. Clique em Salvar

Configurando webhooks de alerta da Grafana

Você pode configurar o Grafana para acionar automaticamente as investigações do AWS DevOps Agente quando os alertas são disparados, enviando webhooks pelos pontos de contato da Grafana. Para obter detalhes sobre métodos de autenticação de webhook e gerenciamento de credenciais, consulte. the section called “Invocando o DevOps Agente por meio do Webhook”

Etapa 1: criar um modelo de notificação personalizado

Na sua instância do Grafana, navegue até Alertas > Pontos de contato > Modelos de notificação e crie um novo modelo com o seguinte conteúdo:

```
{{ 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 }}
```

Este modelo formata os alertas do Grafana na estrutura de carga útil do webhook esperada pelo Agente. AWS DevOps Ele mapeia rótulos de alerta, anotações e status nos campos apropriados e inclui todos os rótulos de alerta como metadados.

> Nota: Esse modelo processa somente o primeiro alerta em um grupo. Grafana agrupa vários alertas de disparo em uma única notificação por padrão. Para garantir que cada alerta seja enviado individualmente, configure suas políticas de notificação para agrupar por`alertname`. Além disso, esse modelo não escapa de caracteres JSON especiais em valores de rótulos ou anotações. Certifique-se de que os rótulos de alerta e a `summary` anotação não contenham caracteres como aspas duplas ou novas linhas, o que produziria um JSON inválido.

Etapa 2: criar um ponto de contato de webhook

1. No Grafana, navegue até Alertas > Pontos de contato e clique em Adicionar ponto de contato
2. Selecione Webhook como o tipo de integração
3. Defina a URL para seu endpoint de webhook do AWS DevOps agente
4. Em Configurações opcionais de webhook, configure os cabeçalhos de autenticação com base no seu tipo de webhook. Consulte Métodos de autenticação do Webhook para obter detalhes.
5. Defina o campo Carga personalizada para usar seu modelo personalizado: `{{ template "devops-agent-payload" . }}`
6. Clique em Salvar ponto de contato

Etapa 3: atribuir o ponto de contato a uma política de notificação

1. Navegue até Alertas > Políticas de notificação
2. Edite uma política existente ou crie uma nova
3. Defina o ponto de contato como o ponto de contato do webhook que você criou
4. Clique em Salvar política

Quando um alerta correspondente é acionado, a Grafana enviará a carga formatada ao AWS DevOps Agente, que iniciará uma investigação automaticamente.

Limitações

- ClickHouse ferramentas de fonte de ClickHouse dados — as ferramentas de fonte de dados não são suportadas atualmente.
- Prevenção proativa de incidentes — atualmente the section called “Prevenção proativa de incidentes” não usa as ferramentas da Grafana. Support está planejado para uma versão futura.

Considerações sobre o Amazon Managed Grafana

Se você estiver usando o Amazon Managed Grafana (AMG), esteja ciente das seguintes limitações:

- Os pontos de contato do webhook não são suportados — atualmente, o AMG não oferece suporte aos pontos de contato do webhook em sua configuração de alerta. Você não pode usar o AMG para enviar webhooks de alerta diretamente para o Agente. AWS DevOps Para obter detalhes, consulte Alertar pontos de contato no Amazon Managed Grafana.
- Expiração do token da conta de serviço — Os tokens da conta de serviço AMG têm uma expiração máxima de 30 dias. Você precisará alternar os tokens e atualizar seu registro da Grafana AWS DevOps no Agent antes que eles expirem. Consulte Gerenciando conexões do Grafana para saber como atualizar as credenciais. Para obter detalhes sobre os limites do token AMG, consulte Contas de serviço na Amazon Managed Grafana.

Gerenciando conexões da Grafana

- Atualização de credenciais — Se o token da sua conta de serviço expirar ou precisar ser atualizado, cancele o registro do Grafana na página Capability Providers e registre-se novamente com o novo token.

- Visualização de instâncias conectadas — No console do AWS DevOps agente, selecione seu Espaço do agente e vá até a guia Capacidades para visualizar as fontes de telemetria conectadas.
- Removendo o Grafana — Para desconectar o Grafana de um Espaço do Agente, selecione-o na seção Telemetria e clique em Remover. Para remover completamente o registro, primeiro remova-o de todos os Agent Spaces e, em seguida, cancele o registro da página Capability Providers.

Conectando a New Relic

Built-in, integração unidirecional

Atualmente, o AWS DevOps Agent oferece suporte aos usuários da New Relic com integração unidirecional integrada, permitindo o seguinte:

- Acionamento automatizado de investigações - os eventos do New Relic podem ser configurados para acionar investigações de resolução de incidentes do AWS DevOps agente por meio AWS DevOps de webhooks do agente.
- Introspecção de telemetria - O AWS DevOps agente pode fazer uma introspecção da telemetria da New Relic enquanto investiga um problema por meio do servidor MCP remoto de cada provedor.

Onboarding

Etapa 1: Conectar

Estabeleça conexão com seu endpoint MCP remoto da New Relic com credenciais de acesso à conta

Use um usuário de plataforma completa (não Basic/Core) na New Relic para habilitar as ferramentas MCP da New Relic.

Configuração

1. Vá para a página Capability Providers (acessível na navegação lateral)
2. Encontre a New Relic na seção Provedores disponíveis em Telemetria e clique em Registrar
3. Siga as instruções para obter sua chave de API New Relic
4. Insira os detalhes da chave de API do servidor New Relic MCP:
   - ID da conta: insira o ID da sua conta New Relic obtido acima

- Chave de API: insira a chave de API obtida acima
- Selecione a região dos EUA ou da UE com base na localização da sua conta New Relic.

5. Clique em Adicionar

Etapa 2: ativar

Ative o New Relic em um espaço de agente específico e configure o escopo apropriado

Configuração

1. Na página de espaços do agente, selecione um espaço do agente e pressione visualizar detalhes (se você ainda não criou um espaço do agente, consultethe section called “Criação de um espaço de agente”)
2. Selecione a guia Capacidades
3. Role para baixo até a seção Telemetria
4. Pressione Adicionar
5. Selecione New Relic
6. Próximo
7. Revise e pressione Salvar
8. Copie o URL do webhook e a chave da API

Etapa 3: configurar webhooks

Usando o URL do Webhook e a chave de API, você pode configurar o New Relic para enviar eventos para acionar uma investigação, por exemplo, a partir de um alarme. Para obter mais detalhes sobre como configurar webhooks, consulte Webhooks de rastreamento de alterações.

Os webhooks da New Relic usam autenticação de token de portador. Para obter o formato completo da solicitação de webhook, o esquema de carga útil e o código de exemplo, consulte. the section called “Invocando o DevOps Agente por meio do Webhook” Use os exemplos da versão 2 (autenticação de token do portador), definindo o `Authorization: Bearer <Token>` cabeçalho com a chave de API da Etapa 2.

Envie webhooks com a New Relic. https://newrelic.com/instant-observability/webhook-notifications Você pode selecionar o token do portador para o tipo de autorização ou selecionar sem autorização e adicioná-lo `Authorization: Bearer <Token>` como um cabeçalho personalizado.

Saiba mais: https://docs.newrelic.com/docs/agentic-ai/mcp/overview/

Remoção

A fonte de telemetria está conectada em dois níveis no nível do espaço do agente e no nível da conta. Para removê-lo completamente, você deve primeiro removê-lo de todos os espaços do agente em que ele é usado e, em seguida, ele pode ser cancelado.

Etapa 1: Remover do espaço do agente

1. Na página de espaços do agente, selecione um espaço do agente e pressione visualizar detalhes
2. Selecione a guia Capacidades
3. Role para baixo até a seção Telemetria
4. Selecione New Relic
5. Pressione remover

Etapa 2: Cancelar o registro da conta

1. Vá para a página Capability Providers (acessível na navegação lateral)
2. Role até a seção Registrado atualmente.
3. Verifique se a contagem de espaço do agente é zero (se não, repita a Etapa 1 acima em seus outros espaços do agente)
4. Pressione Cancelar registro ao lado da New Relic

Conectando o Splunk

Built-in, integração unidirecional

Atualmente, o AWS DevOps Agent oferece suporte aos usuários do Splunk com integração unidirecional integrada, permitindo o seguinte:

- Acionamento automatizado de investigações - os eventos do Splunk podem ser configurados para acionar investigações de resolução de incidentes do AWS DevOps agente por meio AWS DevOps de webhooks do agente.
- Introspecção de telemetria - O AWS DevOps agente pode fazer uma introspecção da telemetria do Splunk enquanto investiga um problema por meio do servidor MCP remoto de cada provedor.

Pré-requisitos

Obtendo um token da API Splunk

Você precisará de um URL e token do MCP para conectar o Splunk.

Etapas do administrador do Splunk

Seu administrador do Splunk precisa executar as seguintes etapas:

- habilitar o acesso à API REST
- habilite a autenticação de token na implantação.
- crie uma nova função 'mcp_user', a nova função não precisa ter nenhum recurso.
- atribua a função 'mcp_user' a qualquer usuário na implantação que esteja autorizado a usar o servidor MCP.
- crie o token para os usuários autorizados com o público como 'mcp' e defina a expiração apropriada, caso o usuário não tenha permissão para criar tokens por conta própria.

Etapas do usuário do Splunk

Um usuário do Splunk precisa realizar as seguintes etapas:

- Obtenha um token apropriado do administrador do Splunk ou crie um por conta própria, se ele tiver permissão. O público do token deve ser 'mcp'.

Onboarding

Etapa 1: Conectar

Estabeleça uma conexão com seu endpoint MCP remoto da Splunk com credenciais de acesso à conta

Configuração

1. Vá para a página Capability Providers (acessível na navegação lateral)
2. Encontre o Splunk na seção Provedores disponíveis em Telemetria e clique em Registrar
3. Insira os detalhes do seu servidor Splunk MCP:
   - Nome do servidor - Identificador exclusivo (por exemplo, my-splunk-server)

- URL do endpoint - Seu endpoint do servidor Splunk MCP:

```
https://<YOUR_SPLUNK_DEPLOYMENT_NAME>.api.scs.splunk.com/
<YOUR_SPLUNK_DEPLOYMENT_NAME>/mcp/v1/
```

- Descrição - Descrição opcional do servidor
- Nome do token - O nome do token portador para autenticação: `my-splunk-token`
- Valor do token O valor do token do portador para autenticação

Etapa 2: ativar

Ative o Splunk em um espaço de agente específico e configure o escopo apropriado

Configuração

1. Na página de espaços do agente, selecione um espaço do agente e pressione visualizar detalhes (se você ainda não criou um espaço do agente, consultethe section called “Criação de um espaço de agente”)
2. Selecione a guia Capacidades
3. Role para baixo até a seção Telemetria
4. Pressione Adicionar
5. Selecione Splunk
6. Próximo
7. Revise e pressione Salvar
8. Copie o URL do webhook e a chave da API

Etapa 3: configurar webhooks

Usando o URL do Webhook e a chave de API, você pode configurar o Splunk para enviar eventos para acionar uma investigação, por exemplo, a partir de um alarme.

Os webhooks do Splunk usam autenticação de token do portador. Para obter o formato completo da solicitação de webhook, o esquema de carga útil e o código de exemplo, consulte. the section called “Invocando o DevOps Agente por meio do Webhook” Use os exemplos da versão 2 (autenticação de token do portador), definindo o `Authorization: Bearer <Token>` cabeçalho com a chave de API da Etapa 2.

Envie webhooks com o Splunk https://help.splunk.com/en/splunk-enterprise/alert-and-respond/alerting-manual/9.4/configure-alert-actions/use-a-webhook-alert-action(observe que não selecione nenhuma autorização e, em vez disso, use a opção de cabeçalho personalizado)

Saiba mais:

- Documentação do servidor MCP da Splunk: https://help.splunk.com/en/splunk-cloud-platform/mcp-server-for-splunk-platform/about-mcp-server-for-splunk-platform
- Requisitos e limitações de acesso para a API REST do Splunk Cloud Platform: https://docs.splunk.com/Documentation/SplunkCloud/latest/RESTTUT/RESTandCloud
- Gerencie tokens de autenticação no 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
- Crie e gerencie funções com o Splunk Web: https://docs.splunk.com/Documentation/SplunkCloud/latest/Security/Addandeditroles

Remoção

A fonte de telemetria é conectada em dois níveis no nível do espaço do agente e no nível da conta. Para removê-lo completamente, você deve primeiro removê-lo de todos os espaços do agente em que ele é usado e, em seguida, ele pode ser cancelado.

Etapa 1: Remover do espaço do agente

1. Na página de espaços do agente, selecione um espaço do agente e pressione visualizar detalhes
2. Selecione a guia Capacidades
3. Role para baixo até a seção Telemetria
4. Selecione Splunk
5. Pressione remover

Etapa 2: Cancelar o registro da conta

1. Vá para a página Capability Providers (acessível na navegação lateral)
2. Role até a seção Registrado atualmente.
3. Verifique se a contagem de espaço do agente é zero (se não, repita a Etapa 1 acima em seus outros espaços do agente)

4. Pressione Cancelar registro ao lado de Splunk

Conectando-se à emissão de bilhetes e ao bate-papo

AWS DevOps O agente foi projetado para atuar como membro da sua equipe, participando dos canais de comunicação existentes da sua equipe. Você pode conectar o DevOps Agent aos seus sistemas de emissão de tíquetes e alarmes, como ServiceNow e PagerDuty, para iniciar automaticamente investigações a partir de tíquetes de incidentes, acelerando a resposta a incidentes em seus fluxos de trabalho existentes para reduzir o tempo médio de recuperação (MTTR). Você também pode conectar seu DevOps agente aos sistemas de colaboração de sua equipe, como o Slack, para receber resumos de atividades do seu DevOps agente em um canal de bate-papo.

Para saber mais sobre como conectar integrações de tickets e chat, veja o seguinte:

- the section called “Conectando PagerDuty”
- the section called “Conectando ServiceNow”
- the section called “Conectando o Slack”

Conectando PagerDuty

PagerDuty a integração permite que o AWS DevOps agente acesse e atualize dados de incidentes, agendas de plantão e informações de serviço de sua PagerDuty conta durante investigações de incidentes e respostas automatizadas. Essa integração usa OAuth 2.0 para autenticação segura.

> **Important**
>
> AWS DevOps O agente suporta somente a PagerDuty OAuth versão 2.0 mais recente (com escopo OAuth). O legado PagerDuty OAuth com uri de redirecionamento não é suportado.

PagerDuty requisitos

Antes de se conectar PagerDuty, verifique se você tem:

- Uma PagerDuty conta com seu ID de OAuth cliente e segredo de cliente
- O subdomínio PagerDuty da sua conta (por exemplo, se sua PagerDuty URL for`https://your-company.pagerduty.com`, o subdomínio é) `your-company`

Registrando PagerDuty

PagerDuty é registrado no nível da AWS conta e compartilhado entre todos os Agent Spaces dessa conta.

Etapa 1: configurar o acesso no PagerDuty

1. Faça login no console AWS de gerenciamento
2. Navegue até o console do AWS DevOps agente
3. Vá para a página Capability Providers (acessível na navegação lateral)
4. Encontre PagerDutyna seção Provedores disponíveis em Comunicação e clique em Registrar
5. Siga a configuração guiada na PagerDuty página Configurar acesso em:

Verifique sua região de serviço e subdomínio:

- Escopo da conta — Selecione sua PagerDuty região (EUA ou UE) e insira seu PagerDuty subdomínio. Por exemplo, se sua PagerDuty URL for`https://your-company.pagerduty.com`, insira`your-company`.

Crie um novo aplicativo em PagerDuty:

- Em uma guia separada do navegador, faça login PagerDuty e navegue até Integrações > Registro de aplicativos
- Crie um novo aplicativo usando OAuth 2.0 Scoped OAuth
- Em Permissões, conceda os seguintes escopos mínimos obrigatórios:`incidents.read`,`incidents.write`, e `services.read`
- Ative a integração de eventos para permitir a comunicação bidirecional entre o agente e AWS DevOps PagerDuty

Configure OAuth as credenciais:

- Escopo da permissão — Os escopos mínimos necessários são:`incidents.read`,, `incidents.write` `services.read`
- Nome do cliente — Insira um nome descritivo para seu OAuth cliente
- ID do cliente — insira o ID do OAuth cliente do registro do seu PagerDuty aplicativo
- Segredo do cliente — Insira o segredo do OAuth cliente do registro do seu PagerDuty aplicativo

Etapa 2: revisar e enviar o PagerDuty registro

1. Revise todos os detalhes da PagerDuty configuração
2. Clique em Enviar para concluir o registro
3. Após o registro bem-sucedido, PagerDuty aparece na seção Registrado atualmente da página Provedores de capacidades

Adicionando PagerDuty a um espaço de agente

Depois de se registrar PagerDuty no nível da conta, você pode conectá-la aos Agent Spaces individuais:

1. No console do AWS DevOps agente, selecione seu Espaço do agente
2. Vá para a guia Capacidades
3. Na seção Comunicações, clique em Adicionar
4. PagerDutySelecione na lista de provedores disponíveis
5. Clique em Salvar

Gerenciando PagerDuty conexões

- Atualização de credenciais — Se suas OAuth credenciais precisarem ser atualizadas, cancele o registro PagerDuty na página Capability Providers e registre-se novamente com as novas credenciais.
- Visualizando conexões — No console do AWS DevOps agente, selecione seu Espaço do agente e vá até a guia Capacidades para visualizar as integrações de comunicação conectadas.
- Removendo PagerDuty — Para se desconectar PagerDuty de um Espaço do Agente, selecione-o na seção Comunicações e clique em Remover. Para remover completamente o registro, primeiro remova-o de todos os Agent Spaces e, em seguida, cancele o registro da página Capability Providers.

Suporte para webhook

AWS DevOps O agente só oferece suporte a PagerDuty webhooks V3. As versões anteriores do webhook não são suportadas.

Para obter mais informações sobre assinaturas de webhook PagerDuty V3, consulte Visão geral de webhooks na documentação do desenvolvedor. PagerDuty

Conectando ServiceNow

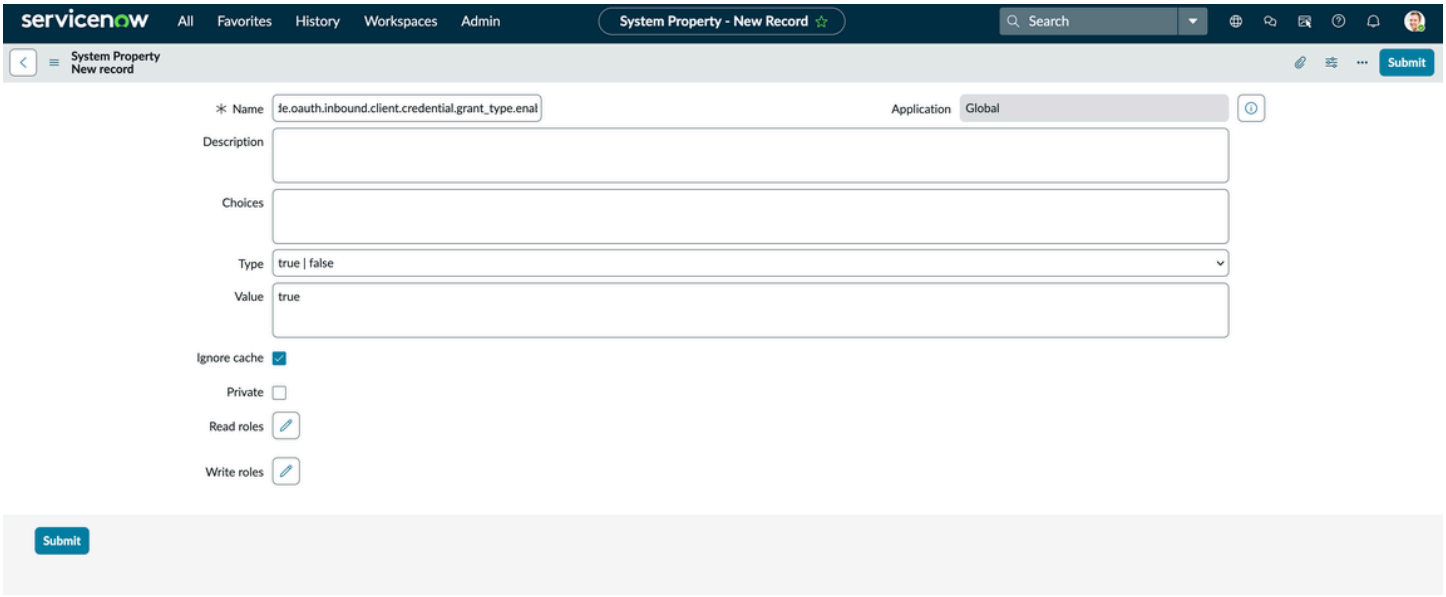
Este tutorial explica como conectar uma ServiceNow instância ao AWS DevOps Agent para permitir que ela inicie automaticamente investigações de resposta a incidentes quando um ticket é criado e publique suas principais descobertas no ticket de origem. Ele também contém exemplos de como configurar sua ServiceNow instância para enviar somente tickets específicos para um DevOps Agent Space e como orquestrar o roteamento de tickets em vários DevOps Agent Spaces.

Configuração inicial

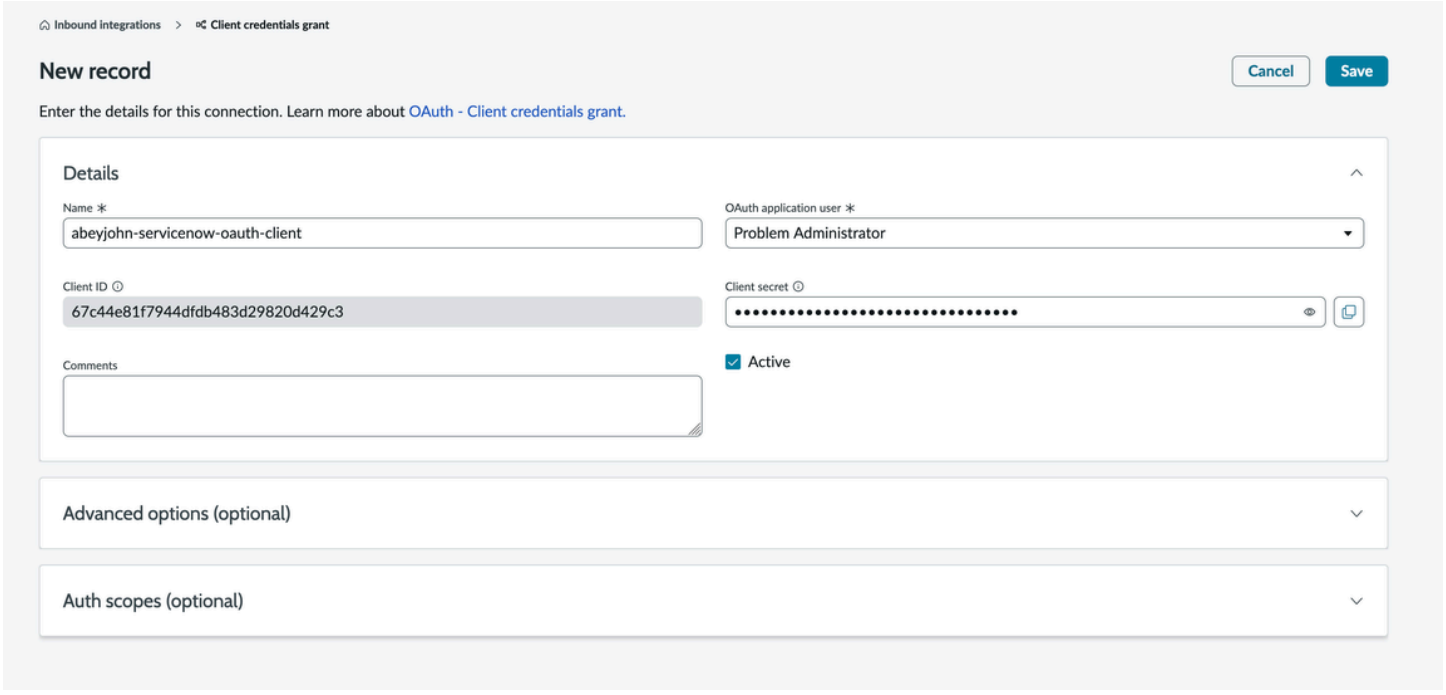
A primeira etapa é criar ServiceNow um cliente de aplicativo OAuth que AWS DevOps possa ser usado para acessar sua ServiceNow instância.

Crie um cliente de ServiceNow aplicativo OAuth

1. Ative a propriedade do sistema de credenciais do cliente da sua instância
   a. Pesquise `sys_properties.list` na caixa de pesquisa do filtro e pressione enter (não mostrará a opção, mas pressionar enter funciona)
   b. Escolha Novo
   c. Adicione o nome como `glide.oauth.inbound.client.credential.grant_type.enabled` e o valor como verdadeiro com o tipo verdadeiro | falso



1. Navegue até System OAuth > Application Registry na caixa de pesquisa do filtro
2. Escolha “Novo” > “Nova experiência de integração de entrada” > “Nova integração” > “OAuth - Concessão de credenciais de cliente”
3. Escolha um nome e defina o usuário do aplicativo OAuth como “Administrador do problema”, clique em “Salvar”

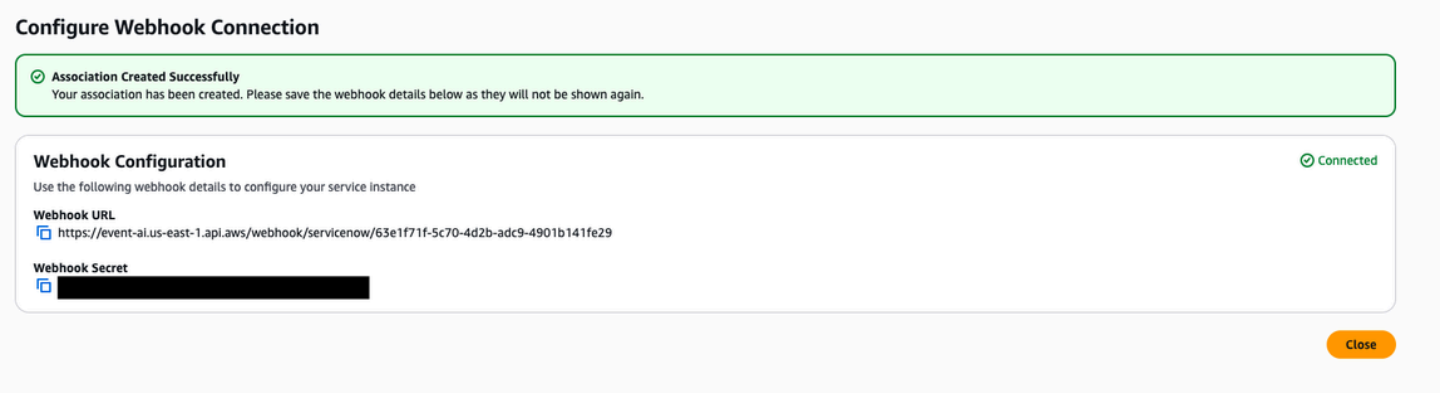


Conecte seu cliente ServiceNow OAuth ao AWS DevOps Agente

1. Você pode iniciar esse processo em dois lugares. Primeiro, navegando até a página Provedores de Capacidades, encontrando ServiceNowem Comunicação e, em seguida, clicando em Registrar. Como alternativa, você pode selecionar qualquer Espaço do DevOps Agente que você possa ter criado e navegar até Capacidades → Comunicações → Adicionar → ServiceNow e clicar em Registrar.
2. Em seguida, autorize o DevOps Agente a acessar sua ServiceNow instância usando o cliente do aplicativo OAuth que você acabou de criar.

- Siga as próximas etapas e salve as informações resultantes sobre o webhook

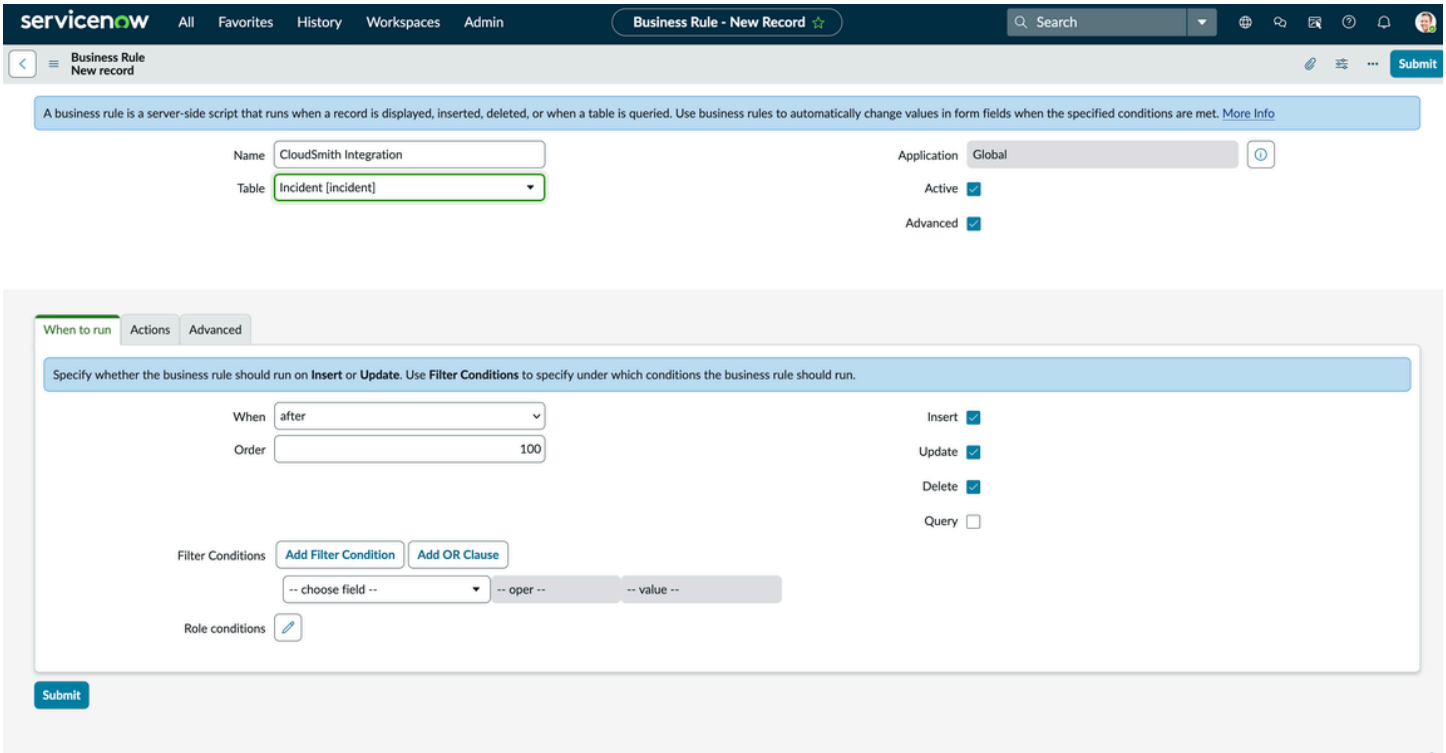
> **Important**
>
> Você não verá essas informações novamente



Configure sua regra ServiceNow de negócios

Depois de estabelecer a conectividade, você precisará configurar uma regra de negócios para enviar tickets ServiceNow para o (s) seu (s) espaço (s) de DevOps agente.

1. Navegue até Assinaturas de atividades → Administração → Regras de negócios e clique em Novo.
2. Defina o campo “Tabela” como “Incidente [incidente]”, marque a caixa “Avançado” e defina a regra a ser executada após Inserir, Atualizar e Excluir.



1. Navegue até a guia “Avançado” e adicione o seguinte script de webhook, inserindo o segredo e o URL do webhook onde indicado, e clique em 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) {
```

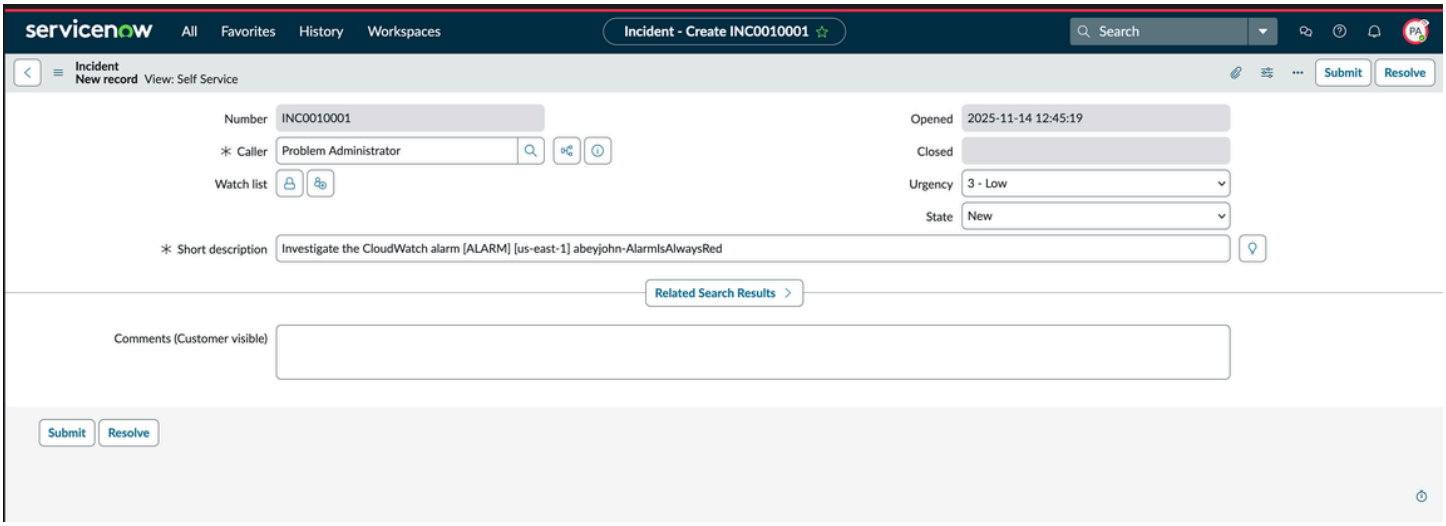
```
        try {
            var mac = new GlideCertificateEncryption();
            var signature = mac.generateMac(secret, "HmacSHA256", payloadString);
            return signature;
        } catch (e) {
            gs.error('HMAC generation failed: ' + e);
            return null;
        }
    }

    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);
```

Se você optou por registrar sua ServiceNow conexão na página Capability Providers, agora você precisa navegar até o DevOps Agent Space no qual deseja investigar os tickets de ServiceNow

incidentes, selecionar Capabilities → Communications e, em seguida, registrar a ServiceNow instância registrada na página Capability Providers. Agora, tudo deve estar configurado e todos os incidentes em que o chamador está configurado como “Administrador do problema” (para imitar as permissões que você concedeu ao cliente AWS DevOps OAuth) acionarão uma investigação de resposta a incidentes no Espaço do Agente configurado. DevOps Você pode testar isso criando um novo incidente ServiceNow e definindo o campo Chamador do incidente como “Administrador do problema”.



ServiceNow atualizações de ingressos

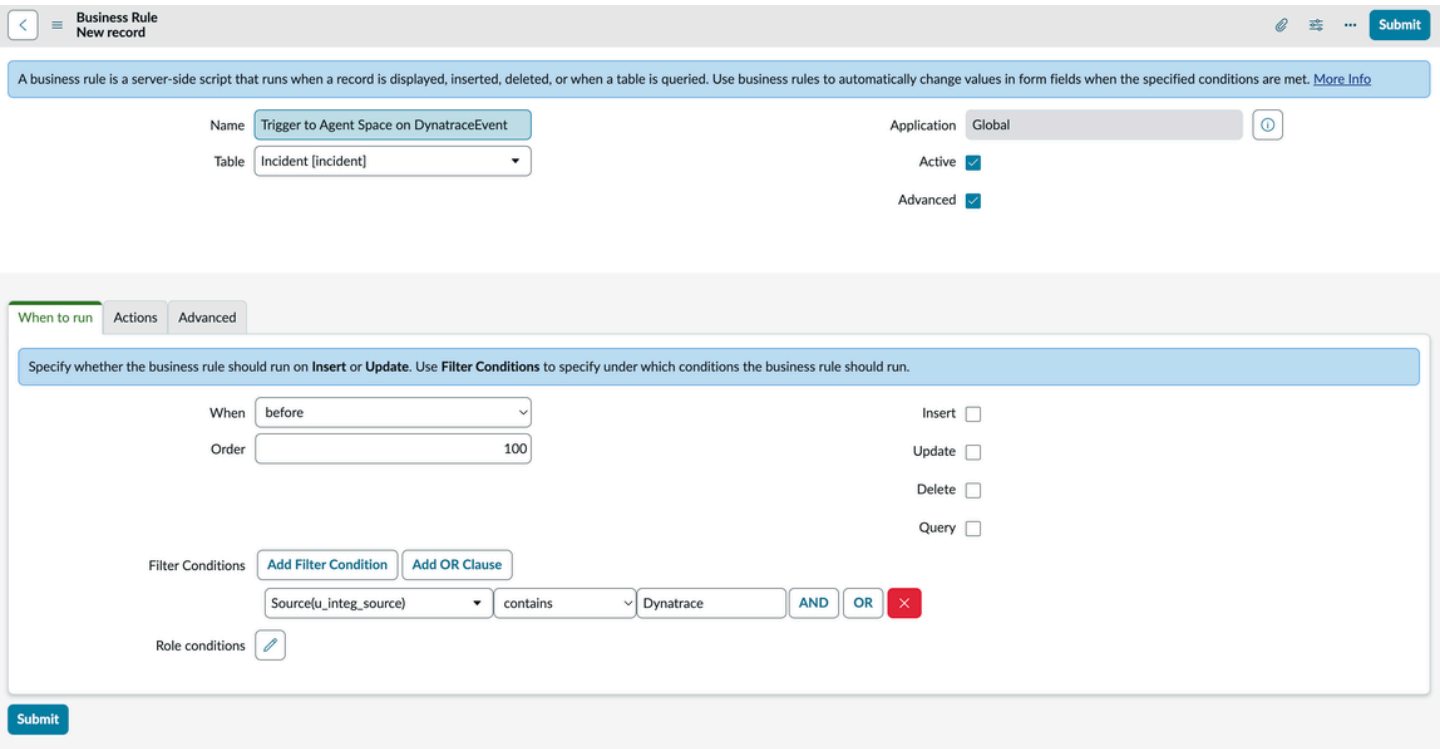
Durante todas as investigações de resposta a incidentes acionados, seu DevOps agente fornecerá atualizações de suas principais descobertas, análises da causa raiz e planos de mitigação no ticket de origem. As descobertas do agente são publicadas nos comentários de um incidente e, atualmente, publicaremos apenas registros do agente do tipo`finding`,, `cause` `investigation_summary``mitigation_summary`, e atualizações do status da investigação (por exemplo`AWS DevOps Agent started/finished its investigation`).

Exemplos de roteamento e orquestração de tickets

Cenário: filtrando quais incidentes são enviados para um espaço de agente DevOps

Esse é um cenário simples, mas precisa de alguma configuração ServiceNow para criar um campo ServiceNow para rastrear a origem do incidente. Para fins deste exemplo, crie um novo campo Fonte (u_source) usando o construtor de formulários SNOW. Isso permitirá rastrear a origem do incidente e usá-la para rotear solicitações de uma fonte específica para um Espaço do DevOps Agente. O roteamento é realizado criando uma regra de negócios do Service Now e, na guia Quando executar,

definindo os gatilhos “Quando” e “Condições do filtro”. Neste exemplo, as condições do filtro são definidas da seguinte forma:

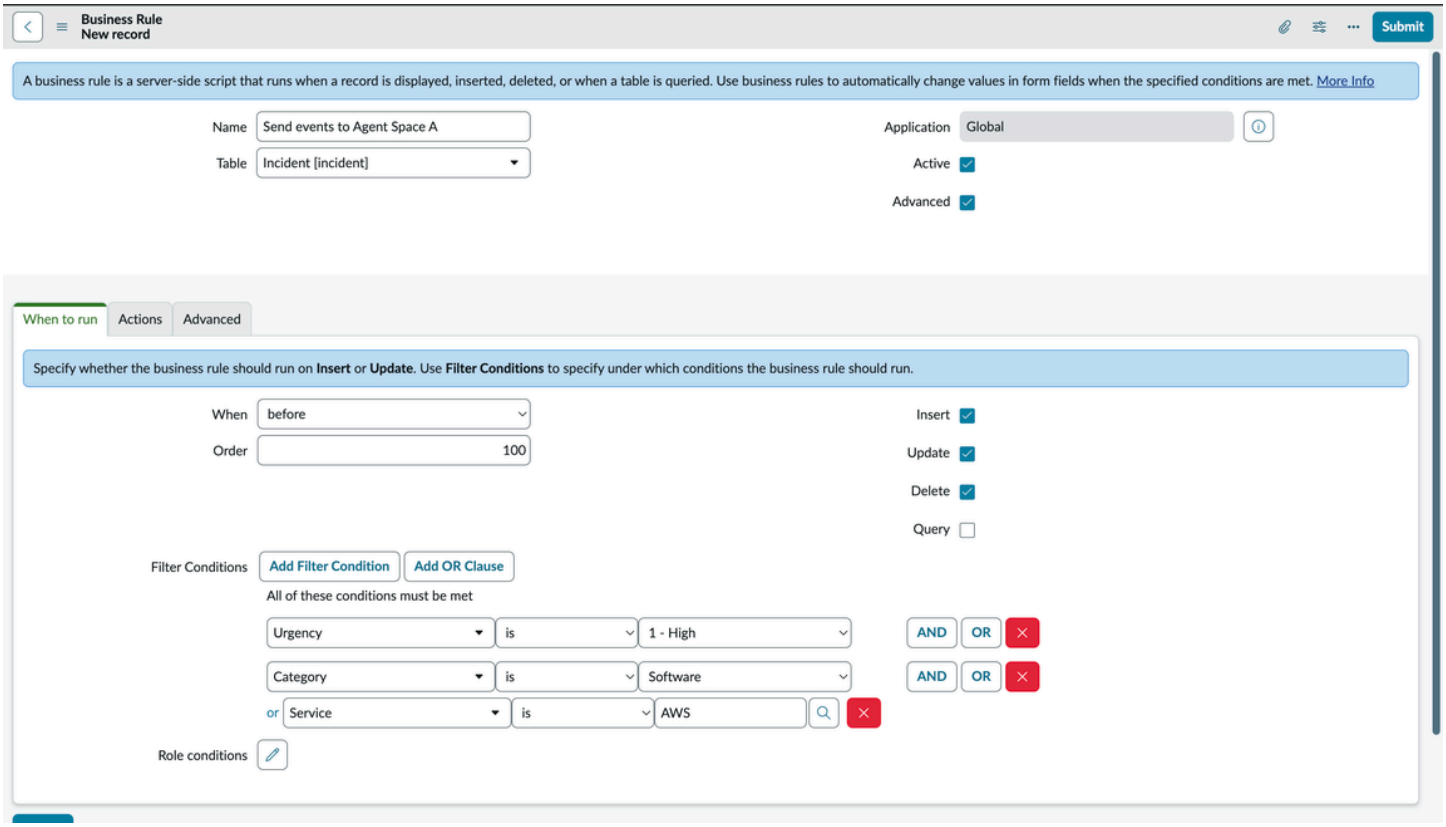


Cenário: roteamento de incidentes em vários DevOps espaços de agentes

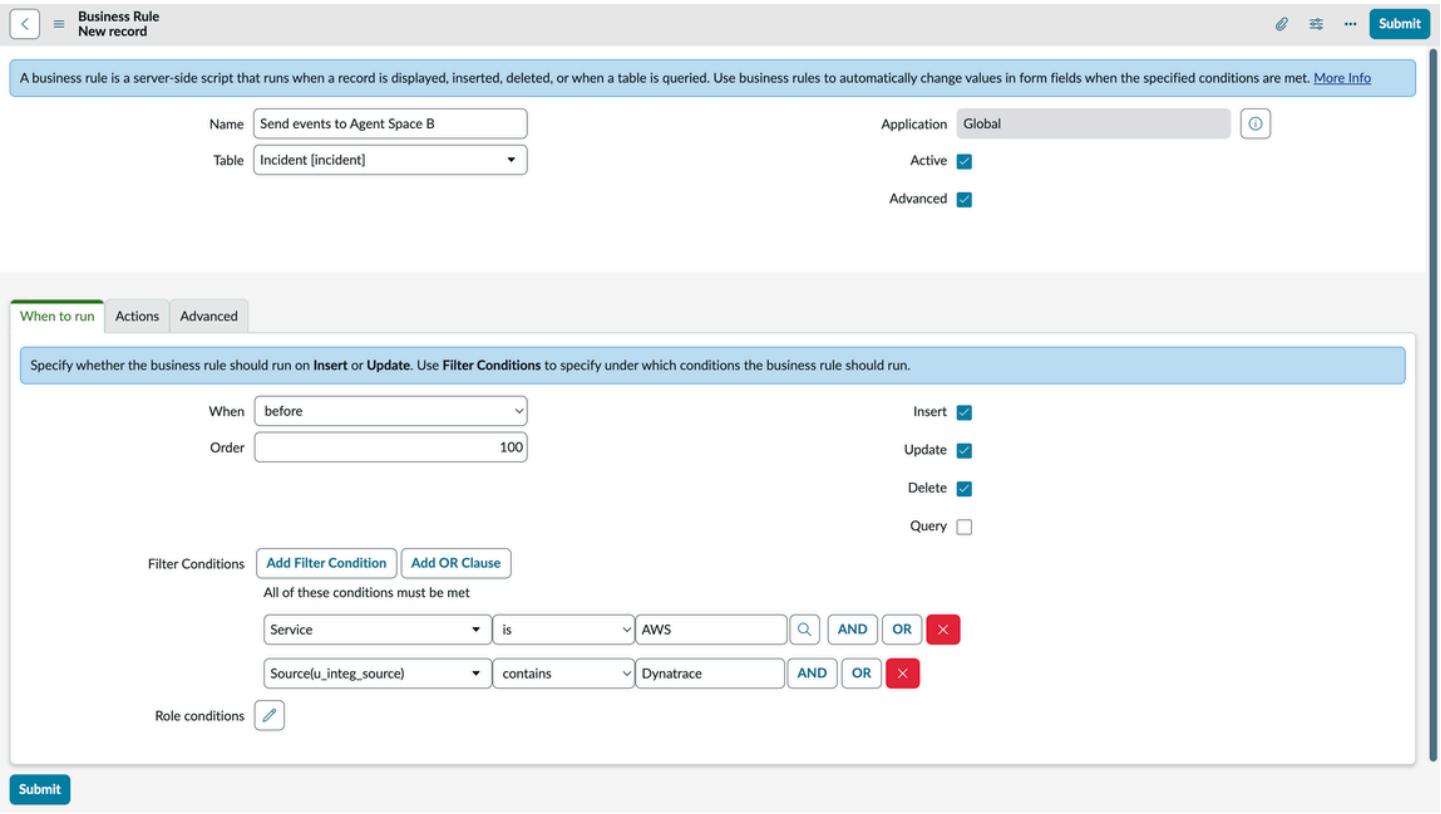
Este exemplo mostra como acionar uma Investigação no Espaço do DevOps Agente B quando a urgência é`1`, a categoria é `Software` ou o Serviço é`AWS`, e acionar uma Investigação no Espaço do DevOps Agente A quando o serviço é `AWS` e a origem é`Dynatrace`.

Esse cenário pode ser realizado de duas maneiras. O script do webhook em si pode ser atualizado para incluir essa lógica de negócios. Nesse cenário, mostraremos como fazer isso com uma regra de ServiceNow negócios, para transparência e simplificação da depuração. O roteamento é realizado com a criação de duas regras de negócios do Service Now.

- Crie uma regra de negócios ServiceNow para o DevOps Agent Space A e crie uma condição usando o construtor de condições para enviar somente os eventos com base em nossa condição especificada.



- Em seguida, crie outra regra de negócios ServiceNow para AgentSpace B, para a qual a regra de negócios só será acionada quando o serviço for AWS e a origem for o Dynatrace.



Agora, quando você cria um novo incidente que corresponda à condição especificada, ele aciona uma investigação no Espaço do DevOps Agente A ou no Espaço do DevOps Agente B, fornecendo a você um controle refinado sobre o roteamento de incidentes.

Conectando o Slack

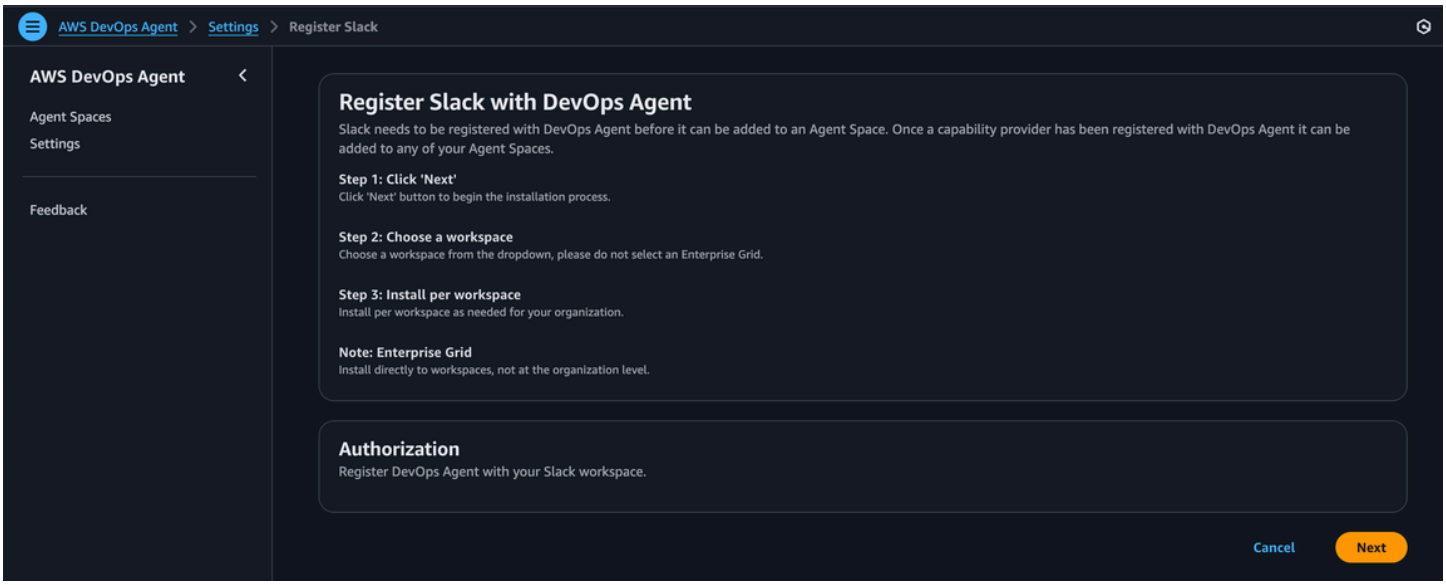
Você pode configurar o AWS DevOps Agente para atualizar um canal do Slack selecionado com as principais descobertas da investigação de resposta a incidentes, análises da causa raiz e planos de mitigação gerados.

Antes de começar

O Slack precisa ser registrado no DevOps Agent antes de poder ser adicionado ao Agent Space. Para integrar o AWS DevOps Agent ao Slack, você deve atender aos seguintes requisitos:

- Tenha acesso a um espaço de trabalho do Slack com a capacidade de instalar e autorizar aplicativos de terceiros
- Identificou os canais do Slack para os quais você deseja que o AWS DevOps agente envie notificações

Registre a integração do Slack com AWS DevOps o Agent



1. Na página Provedores de capacidades no console do AWS DevOps agente, encontre o Slack na seção Provedores disponíveis em Comunicação e clique em Registrar.
2. Escolha o botão Registrar.
3. Você será redirecionado para o Slack para autorizar a inscrição do AWS DevOps Agente no seu espaço de trabalho.
4. Na página de autorização do Slack, instale diretamente nos espaços de trabalho, não no nível da organização.
5. Escolha um espaço de trabalho no menu suspenso. Não selecione um Enterprise Grid.
6. Instale por espaço de trabalho conforme necessário para sua organização.
7. Revise os escopos solicitados e clique em Permitir para autorizar a integração.
8. Após a autorização, você retornará ao console do AWS DevOps agente.

Associe o Slack ao (s) seu (s) espaço (s) de DevOps agente

Depois de registrar o Slack no seu Espaço do DevOps agente, você pode associá-lo ao (s) seu (s) Espaço (s) do DevOps agente:

1. Na guia Capacidades em sua configuração AgentSpace, navegue até Comunicações > Slack.
2. Selecione Adicionar Slack
3. Insira o ID do canal

4. Escolha Criar para concluir a configuração do Slack.

> **Note**
>
> O usuário bot do agente deve ser adicionado aos canais privados antes de poder publicar mensagens.

> **Important**
>
> A desinstalação do aplicativo Slack pode fazer com que o aplicativo Slack não possa ser reinstalado. Evite desinstalar o aplicativo Slack.

Invocando o DevOps Agente por meio do Webhook

Os webhooks permitem que sistemas externos acionem automaticamente as investigações do AWS DevOps agente. Isso permite a integração com sistemas de emissão de bilhetes, ferramentas de monitoramento e outras plataformas que podem enviar solicitações HTTP quando ocorrem incidentes.

Pré-requisitos

Antes de configurar o acesso ao webhook, verifique se você tem:

- Um Espaço do Agente configurado no AWS DevOps Agente
- Acesso ao console do AWS DevOps agente
- O sistema externo que enviará solicitações de webhook

Tipos de webhook

AWS DevOps O Agent oferece suporte aos seguintes tipos de webhooks:

- Integration-specific webhooks — gerados automaticamente quando você configura integrações de terceiros, como Dynatrace, Splunk, Datadog, New Relic ou Slack. ServiceNow Esses webhooks estão associados à integração específica e usam métodos de autenticação determinados pelo tipo de integração.

- Webhooks genéricos — Podem ser criados manualmente para acionar investigações de qualquer fonte não coberta por uma integração específica. Atualmente, os webhooks genéricos usam autenticação HMAC (o token do portador não está disponível no momento).
- Webhooks de alerta do Grafana — O Grafana pode enviar notificações de alerta diretamente AWS DevOps ao Agente por meio de pontos de contato do webhook. Para obter instruções de configuração, incluindo um modelo de notificação personalizado, consulte Conectando o Grafana.

Métodos de autenticação de webhook

O método de autenticação do seu webhook depende da integração à qual ele está associado:

Autenticação HMAC — usada por:

- Webhooks de integração com o Dynatrace
- Webhooks genéricos (não vinculados a uma integração específica de terceiros)

Autenticação de token do portador — usada por:

- Webhooks de integração com o Splunk
- Webhooks de integração com Datadog
- Webhooks de integração com a New Relic
- ServiceNow webhooks de integração
- Webhooks de integração com o Slack
- Webhooks de integração com Grafana

Entendendo a autenticação HMAC

O HMAC (Código de Autenticação de Hash-based Mensagens) é um mecanismo criptográfico que verifica a integridade e a autenticidade de uma solicitação de webhook. Ao enviar um webhook com autenticação HMAC, você gera uma assinatura combinando o timestamp e a carga da solicitação usando sua chave secreta com o algoritmo. SHA-256 AWS DevOps O agente calcula de forma independente o mesmo hash em seu lado e compara as duas assinaturas. Se corresponderem, a solicitação será aceita.

Como o carimbo de data/hora está incluído na assinatura, o HMAC também fornece proteção de repetição — o AWS DevOps agente pode rejeitar solicitações com carimbos de data/hora muito antigos, impedindo que um invasor capture e reenvie uma solicitação válida.

Escolhendo entre HMAC e token Bearer

Consideração	HMAC	Token do portador
Complexidade da configuração	Mais complexo — seu cliente deve computar uma assinatura para cada solicitação usando o carimbo de data/hora e a carga	Mais simples — inclua um token estático no cabeçalho `Authorization`
Integridade da carga	Verificado — qualquer modificação na carga após a assinatura invalida a assinatura	Não verificado — o token autentica o remetente, mas não protege o conteúdo da carga
Proteção de repetição	Built-in — o timestamp na assinatura permite que o servidor rejeite solicitações obsoletas	Não incorporado — um token capturado pode ser reutilizado até ser rotacionado
Risco de exposição secreta	Inferior — o segredo nunca é transmitido na solicitação; somente a assinatura computada é enviada	Maior — o token é enviado em cada cabeçalho de solicitação, aumentando a exposição se o tráfego for interceptado
Quando usar	Recomendado quando você precisa de garantias de segurança mais fortes, como para webhooks genéricos ou ambientes com requisitos rígidos de conformidade	Adequado quando a facilidade de integração é uma prioridade e seu transporte de rede é confiável, como para integrações SaaS gerenciadas via HTTPS

Observação: o método de autenticação é determinado pelo tipo de integração. Integration-specific webhooks (Splunk, Datadog, New Relic, ServiceNow Slack, Grafana) usam autenticação de token de portador. O Dynatrace e os webhooks genéricos usam autenticação HMAC. Você não pode alterar o método de autenticação de um webhook específico da integração.

Configurando o acesso ao webhook

Etapa 1: Navegue até a configuração do webhook

1. Faça login no console AWS de gerenciamento e navegue até o console do AWS DevOps agente
2. Selecione seu espaço de agente
3. Vá para a guia Capacidades
4. Na seção Webhook, clique em Configurar

Etapa 2: gerar credenciais de webhook

Para webhooks específicos de integração:

Os webhooks são gerados automaticamente quando você conclui a configuração de uma integração de terceiros. O URL e as credenciais do endpoint do webhook são fornecidos no final do processo de configuração da integração.

Para webhooks genéricos:

1. Clique em Gerar webhook
2. O sistema gerará um par de chaves HMAC
3. Armazene com segurança a chave e o segredo gerados — você não poderá recuperá-los novamente
4. Copie o URL do endpoint do webhook fornecido

Etapa 3: configurar seu sistema externo

Use o URL e as credenciais do endpoint do webhook para configurar seu sistema externo para enviar solicitações ao Agente. AWS DevOps As etapas específicas de configuração dependem do seu sistema externo.

Gerenciando credenciais de webhook

Removendo credenciais — Para excluir as credenciais do webhook, acesse a seção de configuração do webhook e clique em Remover. Depois de remover as credenciais, o endpoint do webhook não aceitará mais solicitações até que você gere novas credenciais.

Regeneração de credenciais — Para gerar novas credenciais, primeiro remova as existentes e, em seguida, gere um novo token ou par de chaves.

Usando o webhook

Formato de solicitação de webhook

Para acionar uma investigação, seu sistema externo deve enviar uma solicitação HTTP POST para a URL do endpoint do webhook.

Para a versão 1 (autenticação HMAC):

Cabeçalhos:

- `Content-Type: application/json`
- `x-amzn-event-signature: <HMAC signature>`
- `x-amzn-event-timestamp: <+%Y-%m-%dT%H:%M:%S.000Z>`

A assinatura HMAC é gerada assinando o corpo da solicitação com sua chave secreta usando SHA-256.

Para a versão 2 (autenticação de token do portador):

Cabeçalhos:

- `Content-Type: application/json`
- `Authorization: Bearer <your-token>`

Corpo da solicitação:

O corpo da solicitação deve incluir informações sobre o 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"
    }
  }
}
```

Esquema de carga útil:

```
{
    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;
}
```

Código de exemplo

Versão 1 (autenticação 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);
});
```

Versão 1 (autenticação 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"
```

Versão 2 (autenticação do token do 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),
  });
}
```

Versão 2 (autenticação do token do 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"
```

Solução de problemas com webhooks

Se você não receber um 200

Um 200 e uma mensagem como webhook recebida indicam que a autenticação foi aprovada e a mensagem foi colocada na fila para o sistema verificar e processar. Se você não obtiver um 200, mas um 4xx, provavelmente há algo errado com a autenticação ou os cabeçalhos. Tente enviar manualmente usando as opções de curl para ajudar a depurar a autenticação.

Se você receber um 200, mas a investigação não começar

A causa provável é uma carga com formato incorreto.

1. Verifique se o timestamp e o ID do incidente estão atualizados e exclusivos. As mensagens duplicadas são desduplicadas.
2. Verifique se a mensagem é um JSON válido
3. Verifique se o formato está correto

Se você receber um 200 e a investigação for imediatamente cancelada

Provavelmente você atingiu o limite do mês. Fale com seu AWS contato para solicitar uma alteração do limite de tarifa, se apropriado.

Tópicos relacionados

- the section called “Criação de um espaço de agente”
- the section called “O que é um DevOps Agent Web App?”

- the section called “DevOps Permissões do Agent IAM”

Integrar AWS DevOps Agente da Amazon EventBridge

Você pode integrar o AWS DevOps Agent aos seus aplicativos orientados por eventos usando eventos que ocorrem durante os ciclos de vida de investigação e mitigação. AWS DevOps O agente envia eventos para a Amazon EventBridge quando o estado de uma investigação ou mitigação muda. Em seguida, você pode criar EventBridge regras que atuem com base nesses eventos.

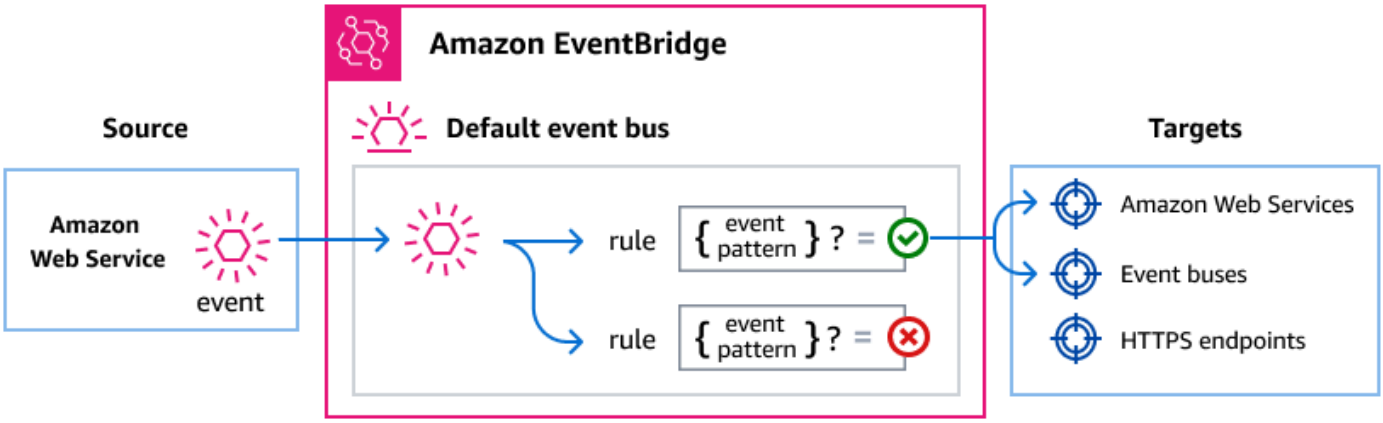
Por exemplo, você pode criar regras que executem as seguintes ações:

- Invoque uma função AWS Lambda para processar os resultados da investigação quando uma investigação for concluída.
- Envie uma notificação do Amazon SNS quando uma investigação falhar ou atingir o tempo limite.
- Atualize um sistema de emissão de tíquetes quando uma nova investigação for criada.
- Inicie um fluxo de trabalho do AWS Step Functions quando uma ação de mitigação for concluída.

Como são EventBridge as rotas AWS DevOps Eventos do agente

AWS DevOps O agente envia eventos para o barramento de eventos EventBridge padrão. EventBridge em seguida, avalia os eventos em relação às regras que você cria. Quando um evento corresponde ao padrão de eventos de uma regra, EventBridge envia o evento para os destinos especificados.

O diagrama a seguir mostra como EventBridge roteia os eventos AWS DevOps do Agente.



1. AWS DevOps O agente envia um evento para o barramento de eventos EventBridge padrão quando o estado do ciclo de vida de uma investigação ou mitigação muda.
2. EventBridge avalia o evento em relação às regras que você criou.
3. Se o evento corresponder ao padrão de evento de uma regra, EventBridge envia o evento para os destinos especificados na regra.

AWS DevOps Eventos do agente

AWS DevOps O agente envia os seguintes eventos para EventBridge. Todos os eventos usam a fonte`aws.aidevops`.

Eventos de investigação apoiados

detail-type	Description
`Investigation Created`	Uma investigação foi criada no espaço do agente.
`Investigation Priority Updated`	A prioridade de uma investigação foi alterada.
`Investigation In Progress`	Uma investigação iniciou uma análise ativa.
`Investigation Completed`	Uma investigação foi concluída com sucesso com as descobertas.
`Investigation Failed`	Uma investigação encontrou um erro e não pôde ser concluída.
`Investigation Timed Out`	Uma investigação excedeu a duração máxima permitida.
`Investigation Cancelled`	Uma investigação foi cancelada antes da conclusão.
`Investigation Pending Triage`	Uma investigação está aguardando a triagem antes do início da análise ativa.

detail-type	Description
`Investigation Linked`	Uma investigação foi vinculada a um incidente ou ticket relacionado.
`Investigation Skipped`	Uma investigação foi ignorada porque correspondia aos critérios de ignorar definidos em uma habilidade.

Eventos de mitigação suportados

detail-type	Description
`Mitigation In Progress`	Uma ação de mitigação foi iniciada.
`Mitigation Completed`	Uma ação de mitigação foi concluída com sucesso.
`Mitigation Failed`	Uma ação de mitigação encontrou um erro e não pôde ser concluída.
`Mitigation Timed Out`	Uma ação de mitigação excedeu a duração máxima permitida.
`Mitigation Cancelled`	Uma ação de mitigação foi cancelada antes da conclusão.

Para obter descrições de campo detalhadas e exemplos de eventos, consultethe section called “AWS DevOps Referência detalhada de eventos do agente”.

Criação de padrões de eventos que correspondam AWS DevOps Eventos do agente

EventBridge as regras usam padrões de eventos para selecionar eventos e encaminhá-los aos alvos. Um padrão de evento corresponde à estrutura dos eventos que ele manipula. Você cria padrões de eventos para filtrar eventos do AWS DevOps Agente com base nos campos de eventos.

Os exemplos a seguir mostram padrões de eventos para casos de uso comuns.

Combine todos os eventos AWS DevOps do Agent

O padrão de eventos a seguir corresponde a todos os eventos do AWS DevOps Agente.

```
{
  "source": ["aws.aidevops"]
}
```

Combine apenas eventos de investigação

O padrão de evento a seguir usa uma correspondência de prefixo para selecionar somente eventos do ciclo de vida da investigação.

```
{
  "source": ["aws.aidevops"],
  "detail-type": [{"prefix": "Investigation"}]
}
```

Combine apenas eventos de conclusão e falha

O padrão de eventos a seguir corresponde aos eventos de investigações e mitigações concluídas ou fracassadas.

```
{
  "source": ["aws.aidevops"],
  "detail-type": [
    "Investigation Completed",
    "Investigation Failed",
    "Mitigation Completed",
    "Mitigation Failed"
  ]
}
```

Combine eventos para um espaço de agente específico

O padrão de eventos a seguir corresponde aos eventos de um espaço de agente específico.

```
{
```

```
  "source": ["aws.aidevops"],
  "detail": {
    "metadata": {
      "agent_space_id": ["your-agent-space-id"]
    }
  }
}
```

Para obter mais informações sobre padrões de eventos, consulte os padrões de EventBridge eventos da Amazon no Guia EventBridge do usuário da Amazon.

EventBridge Permissões da Amazon

AWS DevOps O agente não precisa de permissões adicionais para realizar eventos EventBridge. Os eventos são enviados automaticamente para o barramento de eventos padrão.

Dependendo dos alvos que você configura para suas EventBridge regras, talvez seja necessário adicionar permissões específicas. Para obter mais informações sobre as permissões necessárias para os alvos, consulte Usando políticas baseadas em recursos para a Amazon EventBridge no Guia EventBridge do usuário da Amazon.

EventBridge Recursos adicionais

Para obter mais informações sobre EventBridge conceitos e configuração, consulte os seguintes tópicos no Guia do EventBridge usuário da Amazon:

- EventBridge ônibus para eventos
- EventBridge eventos
- EventBridge padrões de eventos
- EventBridge regras
- EventBridge alvos

AWS DevOps Referência detalhada de eventos do agente

Eventos de AWS serviços têm campos de metadados comuns`source`, incluindo`detail-type`,`account`,`region`, e. `time` Esses eventos também contêm um `detail` campo com dados específicos do serviço. Para eventos do AWS DevOps Agente, o `source` é sempre `aws.aidevops` e `detail-type` identifica o evento específico.

Eventos de investigação

Os `detail-type` valores a seguir identificam os eventos da investigação:

- `Investigation Created`
- `Investigation Priority Updated`
- `Investigation In Progress`
- `Investigation Completed`
- `Investigation Failed`
- `Investigation Timed Out`
- `Investigation Cancelled`
- `Investigation Pending Triage`
- `Investigation Linked`
- `Investigation Skipped`

Os `detail-type` campos `source` e estão incluídos abaixo porque contêm valores específicos para eventos AWS DevOps do agente. Para obter definições dos outros campos de metadados que estão incluídos em todos os eventos, consulte Estrutura de eventos na Amazon EventBridge Events Reference.

A seguir está a estrutura JSON para eventos de investigação.

```
{
  . . .,
  "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 o tipo de evento. Para eventos de investigação, esse é um dos nomes de eventos listados anteriormente.

`source`Identifica o serviço que gerou o evento. Para eventos AWS DevOps do Agent, esse valor é`aws.aidevops`.

`detail`Um objeto JSON que contém dados específicos do evento. O `detail` objeto inclui os seguintes campos:

- `version`(string) — A versão do esquema dos detalhes do evento. Atualmente`1.0.0`.
- `metadata.agent_space_id`(string) — O identificador exclusivo do espaço do agente em que o evento se originou.
- `metadata.task_id`(string) — O identificador exclusivo da tarefa.
- `metadata.execution_id`(string) — O identificador exclusivo da execução. Presente quando uma execução foi atribuída à investigação.
- `data.task_type`(string) — O tipo de tarefa. Valor:`INVESTIGATION`.
- `data.priority`(string) — O nível de prioridade. Valores:`CRITICAL`,`HIGH`,`MEDIUM`,`LOW`,`MINIMAL`.
- `data.status`(string) — O status atual. Valores:`PENDING_START`,`IN_PROGRESS`,`COMPLETED`,`FAILED`,`TIMED_OUT`,`CANCELLED`,`PENDING_TRIAG`
- `data.created_at`(string) — Registro de data e hora ISO 8601 quando a tarefa foi criada.
- `data.updated_at`(string) — Registro de data e hora ISO 8601 de quando a tarefa foi atualizada pela última vez.
- `data.summary_record_id`(string) — O identificador do registro resumido contendo os resultados da investigação. Incluído quando um resumo é gerado para a investigação concluída. Você pode recuperar o conteúdo resumido por meio da API do AWS DevOps agente usando esse identificador para pesquisar o registro do diário com um tipo de registro de`investigation_summary_md`.

Exemplo: Evento de investigação concluído

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

Exemplo: Evento de falha na investigação

```
{
  "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 mitigação

Os `detail-type` valores a seguir identificam eventos de mitigação:

- `Mitigation In Progress`
- `Mitigation Completed`
- `Mitigation Failed`
- `Mitigation Timed Out`
- `Mitigation Cancelled`

Os `detail-type` campos `source` e estão incluídos abaixo porque contêm valores específicos para eventos AWS DevOps do agente. Para obter definições dos outros campos de metadados que estão incluídos em todos os eventos, consulte Estrutura de eventos na Amazon EventBridge Events Reference.

A seguir está a estrutura JSON para eventos de mitigação.

```
{
  . . .,
  "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 o tipo de evento. Para eventos de mitigação, esse é um dos nomes de eventos listados anteriormente.

`source`Identifica o serviço que gerou o evento. Para eventos AWS DevOps do Agent, esse valor é`aws.aidevops`.

`detail`Um objeto JSON que contém dados específicos do evento. O `detail` objeto inclui os seguintes campos:

- `version`(string) — A versão do esquema dos detalhes do evento. Atualmente`1.0.0`.
- `metadata.agent_space_id`(string) — O identificador exclusivo do espaço do agente em que o evento se originou.
- `metadata.task_id`(string) — O identificador exclusivo da tarefa.
- `metadata.execution_id`(string) — O identificador exclusivo da execução. Presente quando uma execução foi atribuída à mitigação.
- `data.task_type`(string) — O tipo de tarefa. Valor:`INVESTIGATION`.
- `data.priority`(string) — O nível de prioridade. Valores:`CRITICAL`,`HIGH`,`MEDIUM`,`LOW`,`MINIMAL`.
- `data.status`(string) — O status atual. Valores:`IN_PROGRESS`,`COMPLETED`,`FAILED`,`TIMED_OUT`,`CANCELLED`.

- `data.created_at`(string) — Registro de data e hora ISO 8601 quando a tarefa foi criada.
- `data.updated_at`(string) — Registro de data e hora ISO 8601 de quando a tarefa foi atualizada pela última vez.
- `data.summary_record_id`(string) — O identificador do registro resumido contendo as descobertas de mitigação. Incluído quando um resumo é gerado para a mitigação concluída. Você pode recuperar o conteúdo resumido por meio da API do AWS DevOps agente usando esse identificador para pesquisar o registro do diário com um tipo de registro de`mitigation_summary_md`.

Exemplo: Evento de mitigação concluído

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

Exemplo: evento de falha na mitigação

```
{
  "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 e métricas vendidas

Você pode monitorar seus espaços de agentes e operações de serviço usando CloudWatch métricas e registros vendidos pela Amazon. Este tópico descreve as CloudWatch métricas que o AWS DevOps Agente publica automaticamente em sua conta e os registros vendidos que você pode configurar para entrega em seus destinos preferidos.

Métricas vendidas CloudWatch

AWS DevOps O agente publica automaticamente métricas CloudWatch na Amazon em sua conta. Essas métricas estão disponíveis sem nenhuma configuração. Você pode usá-los para monitorar o uso, rastrear a atividade operacional e criar alarmes.

Perfil vinculado ao serviço

Para que CloudWatch as métricas da Amazon sejam publicadas em sua conta para esse serviço, o AWS DevOps agente criará automaticamente a função vinculada ao serviço AWSServiceRoleForAIDevOps Service-Linked Role para você. Se a função do IAM que invoca a API não tiver a permissão apropriada, a criação do recurso falhará com um InvalidParameterException.

> **Important**
>
> Os clientes que criaram sua função AgentSpace antes de 13 de março de 2026 precisarão criar manualmente a função vinculada ao serviço de AWSServiceRoleForAIDevoperações para que CloudWatch as métricas do AWS DevOps agente sejam publicadas em suas contas.

Crie manualmente uma função vinculada ao serviço (para clientes existentes)

Execute um destes procedimentos:

- No console do IAM, crie a função AWSServiceRoleForAIDevOps no serviço AWS DevOps Agent.
- Na AWS CLI, execute o seguinte comando:

```
aws iam create-service-linked-role --aws-service-name aidevops.amazonaws.com
```

Namespace

Todas as métricas são publicadas no `AWS/AIDevOps` namespace.

Dimensões

Todas as métricas incluem a seguinte dimensão.

Dimensão	Description
`AgentSpaceUUID`	O identificador exclusivo do espaço do agente. Para agregar métricas em todos os espaços de agentes em sua conta, use expressões CloudWatch matemáticas ou omita o filtro de dimensão.

Referência de métricas

Nome da métrica	Description	Unidade	Frequência de publicação	Estatísticas úteis
ConsumedChatRequests	O número de solicitações de bate-papo que o espaço de um agente consumiu. Para obter a contagem total da sua conta, use a `SUM` estatística em todas as `AgentSpaceUUID` dimensões.	Contagem	A cada 5 minutos	Soma, média
ConsumedInvestigationTime	O tempo gasto conduzindo investigações em um espaço de agente.	Segundos	A cada 5 minutos	Soma, média, máximo

Nome da métrica	Description	Unidade	Frequência de publicação	Estatísticas úteis
ConsumedEvaluationTime	O tempo gasto executando avaliações em um espaço de agente.	Segundos	A cada 5 minutos	Soma, média, máximo
TopologyCompletionCount	O número de conclusões do processamento de topologia. AWS DevOps O agente emite essa métrica quando uma topologia termina o processamento, seja desde a criação inicial durante a integração, uma atualização manual ou uma atualização diária programada.	Contagem	Orientado por eventos (emitido em cada conclusão)	Soma, SampleCount

Visualizando métricas no CloudWatch console

1. Abra o console do CloudWatch .
2. No painel de navegação, escolha Metrics (Métricas) e, em seguida, All metrics (Todas as métricas).
3. Escolha o namespace AWS/AIDevOps.
4. Escolha Por AgentSpace para ver as métricas dos seus espaços de agente.

> **Note**
>
> Você pode criar CloudWatch alarmes sobre essas métricas para receber notificações quando o uso exceder um limite. Por exemplo, crie um alarme `ConsumedChatRequests` para monitorar o consumo de solicitações de bate-papo.

Pré-requisitos

Antes de configurar a entrega de registros, verifique se você tem o seguinte:

- Uma AWS conta ativa com acesso ao console do AWS DevOps agente
- Um diretor do IAM com permissões para entrega de CloudWatch registros APIs
- (Opcional) Um bucket do Amazon S3 ou um stream de entrega do Amazon Data Firehose, se você planeja usá-los como destinos de log

Logs fornecidos

AWS DevOps O agente oferece suporte a registros vendidos que fornecem visibilidade dos eventos que seus espaços de agentes e registros de serviços processam. Os registros vendidos usam a infraestrutura Amazon CloudWatch Logs para entregar os registros ao seu destino preferido.

Para usar registros vendidos, você deve configurar um destino de entrega. Os seguintes destinos são compatíveis:

- Amazon CloudWatch Logs — Um grupo de registros em sua conta
- Amazon S3 — Um bucket S3 em sua conta
- Amazon Data Firehose — Um stream de entrega do Firehose em sua conta

Tipos de log compatíveis

Um único tipo de log é suportado:`APPLICATION_LOGS`. Esse tipo de registro abrange todos os eventos operacionais que o serviço emite.

Registrar tipos de eventos

A tabela a seguir resume os eventos que o AWS DevOps Agente registra.

Event	Description	Nível de log
Evento de entrada do agente recebido	Um agente é acionado por uma fonte integrada e recebe um evento de entrada (por exemplo, um evento de PagerDuty incidente).	INFO
Evento de entrada de agentes cancelado	Um evento de entrada foi descartado antes que o agente o processasse. O registro inclui o motivo (por exemplo, dados malformad os).	A ser definido
Falha na comunicação de saída do agente	Uma comunicação externa com uma integração de terceiros falhou. O registro inclui o ID da tarefa e o identificador de destino (por exemplo, um erro de autentica ção).	A ser definido
Criação de topologia em fila	Um trabalho de criação de topologia foi colocado na fila para processamento.	INFO
A criação da topologia foi iniciada	Um trabalho de criação de topologia começou a ser processado.	INFO
Criação da topologia concluída	Um trabalho de criação de topologia concluiu o processamento. Esse evento se aplica à criação inicial, às atualizações e às atualizações diárias.	INFO

Event	Description	Nível de log
Falha na descoberta de recursos	A descoberta de recursos durante a criação da topologia encontrou uma falha.	ERRO
Falha no registro do serviço	O registro do serviço encontra uma falha irrecuperável	ERRO
Falha na validação do webhook	Quando o webhook recebido pelo agente Devops não corresponde ao esquema esperado	ERRO
Atualizações do status de validação da associação	Quando uma associação de espaço de agente (primary/ secondary conta típica), o status de validação muda de válido para inválido e vice-versa (por exemplo, causado por uma função malformada, que não pode ser assumida pelo serviço).	ERRO/INFORMAÇÃO

Permissões

AWS DevOps O agente usa registros CloudWatch vendidos (permissões V2) para entregar registros. Para configurar a entrega de registros, a função do IAM que configura a entrega deve ter as seguintes permissões:

- `aidevops:AllowVendedLogDeliveryForResource`— Necessário para permitir a entrega de registros para o recurso de espaço do agente.
- Permissões para a entrega de CloudWatch registros APIs (`logs:PutDeliverySource`,`logs:PutDeliveryDestination`,`logs:CreateDelivery`, e operações relacionadas).
- Permissões específicas para o destino de entrega escolhido.

Para ver a política completa do IAM que é necessária para cada tipo de destino, consulte os seguintes tópicos no Guia do usuário do Amazon CloudWatch Logs:

- Registros enviados para CloudWatch Logs
- Logs enviados ao Amazon S3
- Registros enviados para o Firehose

Configurar a entrega de registros (console)

AWS DevOps O agente fornece dois locais no console AWS de gerenciamento para configurar a entrega de registros:

- Página de configurações de registro de serviço — Configure a entrega de registros para eventos de nível de serviço. Esses registros usam o serviço ARN (`arn:aws:aidevops:<region>:<account-id>:service/<account-id>`) como recurso.
- Página Espaço do agente — Configure a entrega de registros para eventos específicos de um espaço de agente individual. Esses registros usam o espaço do agente ARN (`arn:aws:aidevops:<region>:<account-id>:agentspace/<agent-space-id>`) como recurso.

Para configurar a entrega de registros para um registro de serviço

1. Abra o console do AWS DevOps agente no console AWS de gerenciamento.
2. No painel de navegação, selecione Configurações.
3. Na guia Capability Providers > Logs, escolha Configure.
4. Em Tipo de destino, escolha uma das seguintes opções:
5. CloudWatch Registros — Selecione ou crie um grupo de registros.
6. Amazon S3 — Insira o ARN do bucket do S3.
7. Amazon Data Firehose — Selecione ou crie um stream de entrega do Firehose.
8. Para Configurações adicionais — opcionais, você pode especificar as seguintes opções:
   a. Em Seleção de campos, escolha o nome dos campos de log que você deseja entregar ao seu destino. Você pode selecionar campos de log de acesso e um subconjunto de campos de log de acesso em tempo real.
   b. (Somente para o Amazon S3) Em Particionamento, especifique o caminho para particionar os dados do arquivo de log.

c. (Somente para o Amazon S3) Em Formato de arquivo compatível com o Hive, você pode marcar a caixa de seleção para usar caminhos do S3 compatíveis com o Hive. Isso ajuda a simplificar o carregamento de novos dados em suas ferramentas compatíveis com o Hive.

d. Em Formato de saída, especifique o formato de sua preferência.

e. Em Delimitador de campo, especifique como separar os campos de log.

9. Escolha Salvar.

10. Verifique se o status da entrega mostra Ativo.

Para configurar a entrega de registros para um espaço de agente

1. Abra o console do AWS DevOps agente no console AWS de gerenciamento.
2. Escolha o espaço do agente que você deseja configurar.
3. Na guia Configuração, escolha Configurar.
4. Em Tipo de destino, escolha uma das seguintes opções:
5. CloudWatch Registros — Selecione ou crie um grupo de registros.
6. Amazon S3 — Insira o ARN do bucket do S3.
7. Amazon Data Firehose — Selecione ou crie um stream de entrega do Firehose.
8. Para Configurações adicionais — *opcional*, você pode especificar as seguintes opções:

a. Em Seleção de campos, escolha o nome dos campos de log que você deseja entregar ao seu destino. Você pode selecionar campos de log de acesso e um subconjunto de campos de log de acesso em tempo real.

b. (Somente para o Amazon S3) Em Particionamento, especifique o caminho para particionar os dados do arquivo de log.

c. (Somente para o Amazon S3) Em Formato de arquivo compatível com o Hive, você pode marcar a caixa de seleção para usar caminhos do S3 compatíveis com o Hive. Isso ajuda a simplificar o carregamento de novos dados em suas ferramentas compatíveis com o Hive.

d. Em Formato de saída, especifique o formato de sua preferência.

e. Em Delimitador de campo, especifique como separar os campos de log.

9. Escolha Salvar.

10. Verifique se o status da entrega mostra Ativo.

Configurar a entrega de registros (CloudWatch API)

Você também pode usar a API CloudWatch Logs para configurar a entrega de registros de forma programática. A entrega de um log de trabalho consiste em três elementos:

- A DeliverySource— Representa o recurso de espaço do AWS DevOps agente que gera os registros.
- A DeliveryDestination— Representa o destino em que os registros são gravados.
- Uma entrega — conecta uma fonte de entrega a um destino de entrega.

Etapa 1: criar uma fonte de entrega

Use a PutDeliverySourceoperação para criar uma fonte de entrega. Passe o ARN do seu recurso de espaço do AWS DevOps agente e especifique `APPLICATION_LOGS` como o tipo de registro.

O exemplo a seguir cria uma fonte de entrega para um espaço de agente:

```
{
    "name": "my-agent-space-delivery-source",
    "resourceArn": "arn:aws:aidevops:us-east-1:123456789012:agentspace/my-agent-space-
id",
    "logType": "APPLICATION_LOGS"
}
```

O exemplo a seguir cria uma fonte de entrega para o serviço:

```
{
    "name": "my-service-delivery-source",
    "resourceArn": "arn:aws:aidevops:us-east-1:123456789012:service",
    "logType": "APPLICATION_LOGS"
}
```

Etapa 2: criar um destino de entrega

Use a PutDeliveryDestinationoperação para configurar onde os registros são armazenados. Você pode escolher Amazon CloudWatch Logs, Amazon S3 ou Amazon Data Firehose.

O exemplo a seguir cria um destino de CloudWatch registros:

```
{
```

```
    "name": "my-cwl-destination",
    "deliveryDestinationConfiguration": {
        "destinationResourceArn": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/
aidevops/my-agent-space"
    },
    "outputFormat": "json"
}
```

O exemplo a seguir cria um destino do Amazon S3:

```
{
    "name": "my-s3-destination",
    "deliveryDestinationConfiguration": {
        "destinationResourceArn": "arn:aws:s3:::my-aidevops-logs-bucket"
    },
    "outputFormat": "json"
}
```

O exemplo a seguir cria um destino do Amazon Data Firehose:

```
{
    "name": "my-firehose-destination",
    "deliveryDestinationConfiguration": {
        "destinationResourceArn": "arn:aws:firehose:us-
east-1:123456789012:deliverystream/my-aidevops-log-stream"
    },
    "outputFormat": "json"
}
```

> **Note**
>
> Se você entregar registros em várias contas, deverá usá-los PutDeliveryDestinationPolicyna conta de destino para autorizar a entrega.

Se você quiser usar CloudFormation, você pode usar o seguinte:

- Delivery
- DeliveryDestination
- DeliverySource

`ResourceArn` é `AgentSpaceArn`, e `LogType` deve ser `APPLICATION_LOGS` como o tipo de log compatível.

Etapa 3: criar uma entrega

Use a CreateDeliveryoperação para vincular a origem da entrega ao destino da entrega.

```
{
    "deliverySourceName": "my-agent-space-delivery-source",
    "deliveryDestinationArn": "arn:aws:logs:us-east-1:123456789012:delivery-
destination:my-cwl-destination"
}
```

AWS CloudFormation

Você também pode configurar a entrega de registros usando AWS CloudFormation os seguintes recursos:

- AWS: :Registros:: DeliverySource
- AWS: :Registros:: DeliveryDestination
- AWS: :Logs: :Entrega

`ResourceArn`Defina o espaço do AWS DevOps agente ou o ARN do serviço e `LogType` defina como. `APPLICATION_LOGS`

Registro da referência de esquemas

AWS DevOps O agente usa um esquema de log compartilhado em todos os tipos de eventos. Nem todo evento de log usa todos os campos.

A tabela a seguir descreve os campos no esquema de log.

Campo	Tipo	Description
event_timestamp	Longo	Timestamp Unix de quando o evento ocorreu
resource_arn	String	ARN do recurso que gerou o evento

Campo	Tipo	Description
id_da_conta opcional	String	AWS ID da conta associada ao registro.
nível_opcional	String	Nível de registro:`INFO`,`WARN`, `ERROR`
id_do_espaço_agente opcional	String	Identificador do espaço do agente.
id_de_associação opcional	String	Identificador de associação para o registro.
status_opcional	String	Status da operação de topologia.
id_webhook_id opcional	String	Identificador de webhook.
url_mcp_endpoint_opcional	String	URL do endpoint do servidor MCP
tipo_de_serviço_opcional	String	Tipo de serviço:`DYNATRACE` ,`DATADOG`,`GITHUB`,`SLACK`,`SERVICENOW` .
url_de_endpoint_de_serviço opcional	String	URL do endpoint para integrações de terceiros.
id_de_serviço opcional	String	Identificador da fonte.
request_id	String	Identificador de solicitação para correlacionar com AWS CloudTrail nossos tickets de suporte.
operação_opcional	String	Nome da operação que foi executada.

Campo	Tipo	Description
tipo_de_tarefa_opcional	String	Tipo de tarefa da lista de pendências do agente: `INVESTIGATION` ou `EVALUATION`
id_da_tarefa_opcional	String	Agent Backlog Task Identific ador IDAgent de tarefa de backlog.
referência_opcional	String	Referência de uma tarefa do agente (por exemplo, um ticket do Jira).
tipo_de_erro opcional	String	Tipo de erro
mensagem_de_erro opcional	String	Descrição do erro quando uma operação falha.
detalhes_opcionais	Cadeia de caracteres (JSON)	Carga útil de eventos específic a do serviço que contém parâmetros e resultados da operação.

Gerenciar e desativar a entrega de registros

Você pode modificar ou remover a entrega de registros a qualquer momento no console do AWS DevOps agente no AWS Management Console ou usando a API de CloudWatch registros.

Gerenciar a entrega de registros (console)

1. Abra o console do AWS DevOps agente no console AWS de gerenciamento.
2. Navegue até a página Configurações (para registros de nível de serviço) ou a página específica do Espaço do Agente (para registros no nível do Espaço do Agente).
3. Na guia Configuração (para registros no nível do Agent Space) ou na guia Capability Providers > Logs (para registros no nível do serviço), escolha a entrega a ser modificada.
4. Atualize a configuração conforme necessário e escolha Salvar.

Observação: você não pode alterar o tipo de destino de uma entrega existente. Para alterar o tipo de destino, exclua a entrega atual e crie uma nova.

Desativar a entrega de registros (console)

1. Abra o console do AWS DevOps agente no console AWS de gerenciamento.
2. Navegue até a página Configurações (para registros de nível de serviço) ou a página específica do Espaço do Agente (para registros no nível do Espaço do Agente).
3. Na guia Configuração (para registros no nível do Agent Space) ou na guia Capability Providers > Logs (para registros no nível do serviço), selecione a entrega a ser removida.
4. Escolha Excluir e confirme.

Desativar a entrega de registros (API)

Para remover uma entrega de registros usando a API, exclua os recursos na seguinte ordem:

1. Exclua a entrega usando DeleteDelivery.
2. Exclua a fonte de entrega usando DeleteDeliverySource.
3. (Opcional) Se o destino da entrega não for mais necessário, exclua-o usando DeleteDeliveryDestination.

> **Important**
>
> Você é responsável por remover os recursos de entrega de registros depois de excluir o recurso de espaço do agente que gera os registros (por exemplo, depois de excluir um espaço do agente). Se você não remover esses recursos, as configurações de entrega órfã poderão permanecer.

Preços

O AWS DevOps agente não cobra pela ativação de registros vendidos. Entretanto, você pode incorrer em cobranças pela entrega, ingestão, armazenamento ou acesso, dependendo do destino de entrega de log selecionado. Para obter detalhes sobre preços, consulte Vended Logs na guia Logs em Amazon CloudWatch Pricing.

Para preços específicos do destino, consulte o seguinte:

- Preços do Amazon CloudWatch Logs
- Preços do Amazon S3
- Definição de preço do Amazon Data Firehose

Conectando-se a ferramentas hospedadas de forma privada

Visão geral das conexões privadas

AWS DevOps O agente pode ser estendido com ferramentas personalizadas do Model Context Protocol (MCP) e outras integrações que dão ao agente acesso a sistemas internos, como registros de pacotes privados, plataformas de observabilidade auto-hospedadas, APIs de documentação interna e instâncias de controle de origem (consulte:). Configurando recursos para AWS DevOps Agente Esses serviços geralmente são executados dentro de uma Amazon Virtual Private Cloud (Amazon VPC) com acesso restrito ou nenhum acesso público à Internet, o que significa que o AWS DevOps Agente não pode acessá-los por padrão.

As conexões privadas do AWS DevOps Agent permitem que você conecte com segurança seu Espaço do Agente aos serviços em execução na sua VPC sem expô-los à Internet pública. As conexões privadas funcionam com qualquer integração que precise alcançar um endpoint privado, incluindo servidores MCP, instâncias Grafana ou Splunk auto-hospedadas e sistemas de controle de origem, como Enterprise Server e. GitHub GitLab Self-Managed

> **Note**
>
> Se suas ferramentas hospedadas de forma privada fizerem solicitações de saída para o AWS DevOps Agente de dentro da sua VPC, esse tráfego também poderá ser protegido usando um VPC Endpoint para que ele permaneça na rede. AWS Por exemplo, isso pode ser usado com ferramentas que acionam o DevOps Agente por meio de eventos de webhook (consulte:the section called “Invocando o DevOps Agente por meio do Webhook”). Para obter mais informações, consulte the section called “VPC endpoints (AWS PrivateLink)”.

Como as conexões privadas funcionam

Uma conexão privada cria um caminho de rede seguro entre o AWS DevOps Agente e um recurso de destino em sua VPC. Nos bastidores, o AWS DevOps Agent usa o Amazon VPC Lattice para estabelecer esse caminho seguro de conectividade privada. O VPC Lattice é um serviço de rede de

aplicativos que permite conectar, proteger e monitorar a comunicação entre aplicativos em VPCs, contas e tipos de computação, sem gerenciar a infraestrutura de rede subjacente.

Quando você cria uma conexão privada, ocorre o seguinte:

- Você fornece a VPC, as sub-redes e (opcionalmente) os grupos de segurança que têm conectividade de rede com seu serviço de destino.
- AWS DevOps O agente cria um gateway de recursos gerenciado por serviços e provisiona suas interfaces de rede elástica (ENIs) nas sub-redes que você especificou.
- O agente usa o gateway de recursos para rotear o tráfego para o endereço IP ou nome DNS do serviço de destino pelo caminho da rede privada.

O gateway de recursos é totalmente gerenciado pelo AWS DevOps Agente e aparece como um recurso somente para leitura em sua conta (nomeado`aidevops-{your-private-connection-name}`). Você não precisa configurá-lo ou mantê-lo. Os únicos recursos criados em sua VPC são ENIs nas sub-redes que você especifica. Esses ENIs servem como ponto de entrada para tráfego privado e são gerenciados inteiramente pelo serviço. Eles não aceitam conexões de entrada da Internet e você mantém controle total sobre o tráfego deles por meio de seus próprios grupos de segurança.

Segurança

As conexões privadas são projetadas com várias camadas de segurança:

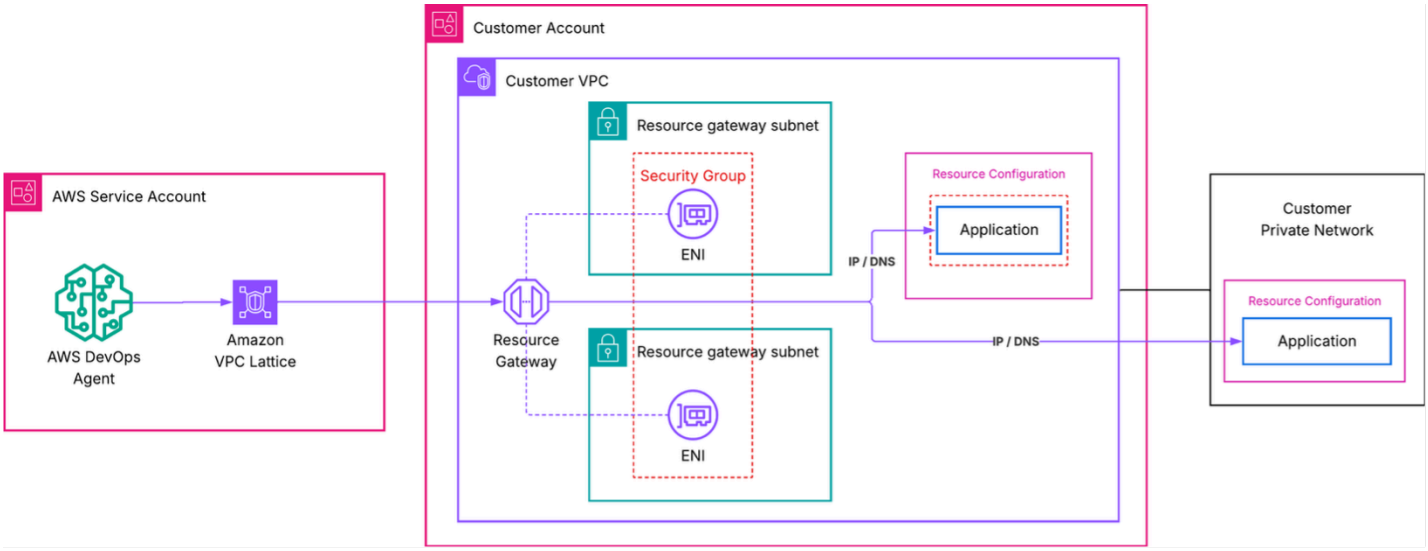
- Sem exposição pública à Internet — Todo o tráfego entre o AWS DevOps agente e seu serviço de destino permanece na AWS rede. Seu serviço nunca precisa de um endereço IP público ou gateway de internet.
- Service-controlled gateway de recursos — O gateway de recursos gerenciado por serviços é somente para leitura em sua conta. Ele só pode ser usado pelo AWS DevOps Agente, e nenhum outro serviço ou principal pode rotear o tráfego por ele. Você pode verificar isso nos AWS CloudTrailregistros, que registram todas as chamadas da API VPC Lattice.
- Seus grupos de segurança, suas regras — Você controla o tráfego de entrada e saída para os ENIs por meio de grupos de segurança que você possui e gerencia. Se você não especificar grupos de segurança, o AWS DevOps Agente cria um grupo de segurança padrão com o escopo das portas que você define.
- Service-linked funções com menos privilégios — O AWS DevOps agente usa uma função vinculada ao serviço para criar somente os recursos necessários do VPC Lattice e do Amazon

EC2. Essa função tem como escopo os recursos marcados com `AWSAIDevOpsManaged` e não pode acessar nenhum outro recurso em sua conta.

> **Note**
>
> Se sua organização tem políticas de controle de serviço (SCPs) que restringem as ações da API VPC Lattice, o gateway de recursos gerenciados por serviços é criado por meio de uma função vinculada ao serviço. Certifique-se de que seus SCPs permitam as ações necessárias para a função vinculada ao serviço.

Arquitetura

O diagrama a seguir mostra o caminho de rede para uma conexão privada.



Nesta arquitetura:

- AWS DevOps O agente inicia uma solicitação para seu serviço de destino.
- O Amazon VPC Lattice encaminha a solicitação por meio do gateway de recursos gerenciado por serviços em sua VPC. Para configurações avançadas usando seus próprios recursos do VPC Lattice, consulte Configuração avançada usando os recursos existentes do VPC Lattice.
- Uma ENI na sua VPC recebe o tráfego e o encaminha para o endereço IP ou nome DNS do seu serviço de destino.
- Seus grupos de segurança controlam qual tráfego é permitido por meio dos ENIs.

- Do ponto de vista do seu serviço de destino, a solicitação se origina de endereços IP privados de ENIs em sua VPC.

Crie uma conexão privada

Você pode criar uma conexão privada usando o AWS Management Console ou a AWS CLI.

> **Note**
>
> As seguintes zonas de disponibilidade não são compatíveis com o VPC Lattice:`use1-az3`,,,`usw1-az2`,`apne1-az3`,, `apne2-az2euc1-az2`,`euw1-az4`. `cac1-az3` `ilc1-az2`

Pré-requisitos

Antes de criar uma conexão privada, verifique se você tem o seguinte:

- Um Espaço do Agente ativo — Você precisa de um Espaço do Agente existente na sua conta. Se você não tiver uma, consulte Começando com o AWS DevOps Agent.
- Um serviço de destino com acesso privado — Seu servidor MCP, plataforma de observabilidade ou outro serviço deve estar acessível em um endereço IP privado conhecido ou nome DNS da VPC em que o gateway de recursos está implantado. O serviço pode ser executado na mesma VPC, em uma VPC emparelhada ou no local, desde que seja roteável a partir das sub-redes do gateway de recursos. O serviço deve fornecer tráfego HTTPS com uma versão TLS mínima de 1.2 em uma porta que você especifica ao criar a conexão.
- Sub-redes em sua VPC — identifique de 1 a 20 sub-redes nas quais as ENIs serão criadas. Recomendamos selecionar sub-redes em várias zonas de disponibilidade para alta disponibilidade. Essas sub-redes devem ter conectividade de rede com seu serviço de destino. Uma sub-rede por zona de disponibilidade pode ser usada pelo VPC Lattice.
- (Opcional) Grupos de segurança — Se você quiser controlar o tráfego com regras específicas, prepare até cinco IDs de grupos de segurança para anexar aos ENIs. Se você omitir grupos de segurança, o AWS DevOps Agente cria um grupo de segurança padrão.

Conexões privadas são recursos em nível de conta. Depois de criar uma conexão privada, você pode reutilizá-la em várias integrações e espaços de agentes que precisam alcançar o mesmo host.

Crie uma conexão privada usando o console

1. Abra o console do AWS DevOps agente.
2. No painel de navegação, escolha Provedores de capacidade e, em seguida, escolha Conexões privadas.
3. Escolha Criar uma conexão.
4. Em Nome, insira um nome descritivo para a conexão, como`my-mcp-tool-connection`.
5. Para VPC, selecione a VPC em que as ENIs do gateway de recursos serão implantadas.
6. Para Sub-redes, selecione uma ou mais sub-redes (até 20). Recomendamos escolher sub-redes em pelo menos duas zonas de disponibilidade.
7. Para o tipo de endereço IP, selecione o tipo de endereço IP do seu serviço de destino (`IPv4`,`IPv6`, ou`DualStack`).
8. (Opcional) Em Número de endereços IPv4, se você selecionou IPv4 ou Dualstack para o tipo de endereço IP, poderá inserir o número de endereços IPv4 por ENI para seu gateway de recursos. O padrão são 16 endereços IPv4 por ENI.
9. (Opcional) Para grupos de segurança, selecione grupos de segurança existentes (até 5) para restringir qual tráfego pode alcançar seu serviço de destino. Se você não selecionar nenhum, um grupo de segurança padrão será criado.
10. (Opcional) Para intervalos de portas, especifique as portas TCP que seu aplicativo de destino escuta (por exemplo, `443` ou`8080-8090`). Você pode especificar até 11 intervalos de portas.
11. Em Endereço do host, insira o endereço IP ou o nome DNS do seu serviço de destino (por exemplo, `mcp.internal.example.com` ou`10.0.1.50`). O serviço deve estar acessível a partir da VPC selecionada. Se você escolher um nome DNS, ele deverá ser resolvido publicamente.
12. (Opcional) Em Chave pública do certificado, se o endereço do host que você especificou usar certificados TLS emitidos por uma autoridade de certificação privada, insira a chave PEM-encoded pública do certificado. Isso permite que o AWS DevOps Agente confie na conexão TLS com seu serviço de destino.
13. Escolha Criar conexão.

O status da conexão muda para Criar em andamento. Esse processo pode levar até 10 minutos. Quando o status muda para Ativo, o caminho da rede está pronto.

Se o status mudar para Falha na criação, verifique o seguinte:

- As sub-redes que você especificou têm endereços IP disponíveis.

- Sua conta não atingiu as cotas de serviço do VPC Lattice.
- Nenhuma política restritiva do IAM está impedindo que a função vinculada ao serviço crie recursos.

> **Note**
>
> Essas etapas também podem ser executadas selecionando `Create a new private connection` durante o registro de um provedor de recursos. Para obter mais informações, consulte Usar uma conexão privada com um provedor de recursos.

Crie uma conexão privada usando o AWS CLI

Execute o comando a seguir para criar uma conexão privada. Substitua os valores dos espaços reservados pelos seus próprios.

```
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"]
        }
    }'
```

A resposta inclui o nome da conexão e um status 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 verificar o status da conexão, use o `describe-private-connection` comando:

```
aws devops-agent describe-private-connection \
    --name my-mcp-tool-connection
```

Quando o status for`ACTIVE`, sua conexão privada estará pronta para uso.

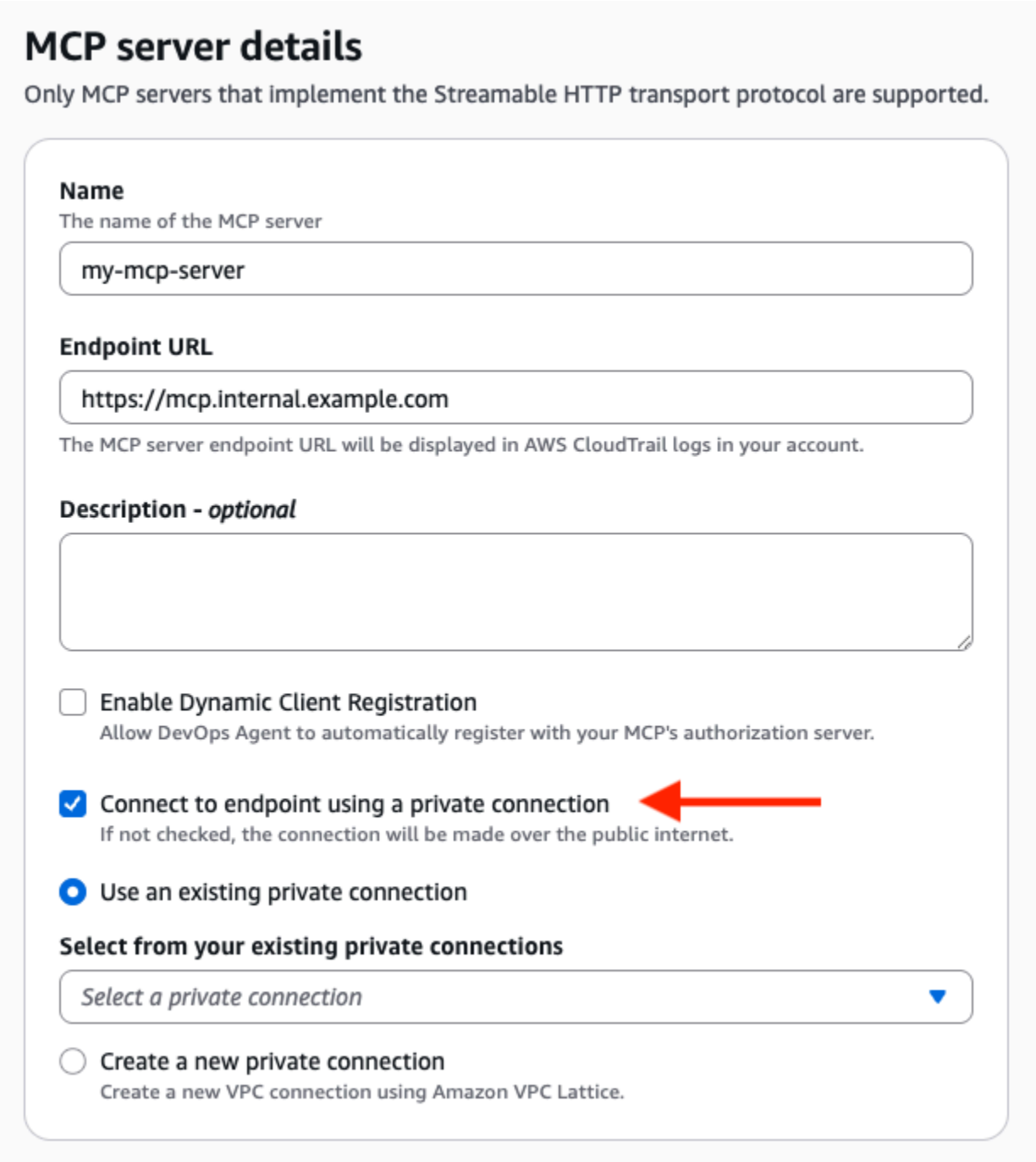
Use uma conexão privada com um provedor de recursos

Para usar uma conexão privada, você pode vinculá-la durante o registro de um provedor de recursos. Os recursos compatíveis que podem ser usados com conexões privadas incluem: `GitHub` `GitLabMCP Server`,, `Grafana` e. Você pode executar essa etapa usando o console AWS de gerenciamento ou a AWS CLI.

> **Note**
>
> Ao registrar um provedor de recursos, o AWS DevOps Agente valida se o endpoint está acessível e está respondendo. Certifique-se de que seu serviço de destino esteja funcionando e aceitando conexões antes de concluir o registro.

Use uma conexão privada com um provedor de recursos usando o console

No console do AWS DevOps agente, as conexões privadas podem ser vinculadas a um recurso durante o registro, selecionando a opção “Conectar-se ao endpoint usando uma conexão privada”.



1. Abra o console do AWS DevOps agente e navegue até seu Espaço do agente.
2. Na seção Provedores de recursos, escolha Registro.
3. Selecione Registrar para o tipo de capacidade que você deseja usar com a conexão privada.
4. Na visualização de detalhes do registro, insira o URL do Endpoint ao qual você deseja se conectar usando a conexão privada (por exemplo,`https://mcp.internal.example.com`).
5. Selecione Conectar ao endpoint usando uma conexão privada.

6. Selecione uma conexão privada existente que corresponda ao URL do Endpoint ao qual você deseja se conectar ou selecione Criar uma nova conexão privada para criar uma.
7. Conclua o processo de registro do provedor de recursos.

> **Note**
>
> Quando você seleciona uma conexão privada para um provedor de recursos que usa autenticação OAuth (Client Credentials ou 3LO), a conexão privada se aplica tanto ao endpoint do provedor de recursos quanto ao endpoint de troca de tokens. Certifique-se de que a conexão privada esteja configurada com um endereço de host que possa rotear o tráfego para os dois endpoints.

Use uma conexão privada com um provedor de recursos usando o AWS CLI

Você pode registrar recursos com uma conexão privada incluindo o `private-connection-name` argumento. Abaixo está um exemplo de registro de um servidor MCP com autorização de chave de API usando a conexão `my-mcp-tool-connection` privada. Substitua os valores dos espaços reservados pelos seus próprios.

```
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
```

Verificar uma conexão privada

Depois que a conexão privada atingir o estado Ativo e for utilizada por um provedor de recursos, verifique se o AWS DevOps Agente pode acessar seu serviço de destino:

1. Abra o console do AWS DevOps agente e navegue até seu Espaço do agente.
2. Inicie uma nova sessão de bate-papo.
3. Invoque um comando que usa a integração apoiada por sua conexão privada. Por exemplo, se sua ferramenta MCP fornece acesso a uma base de conhecimento interna, faça ao agente uma pergunta que exija essa base de conhecimento.
4. Confirme se o agente retorna os resultados do serviço privado.

Se a conexão falhar, verifique o seguinte:

- Limites do VPC Lattice — Verifique se você não atingiu nenhum gateway de recursos ou outros limites de cota do VPC Lattice
- Regras do grupo de segurança — Verifique se os grupos de segurança conectados às ENIs permitem tráfego de saída na porta que seu serviço escuta. Verifique também se o grupo de segurança do seu serviço permite tráfego de entrada na porta de destino. O tráfego chega dos IPs do plano de dados do VPC Lattice dentro do intervalo CIDR do VPC. Você pode usar a referência de grupos de segurança (permitindo o grupo de segurança ENI como fonte) ou permitir a entrada do CIDR da VPC.
- Conectividade de sub-rede — verifique se as sub-redes selecionadas podem rotear o tráfego para o seu serviço. Se o serviço for executado em uma sub-rede diferente, confirme se as tabelas de rotas permitem tráfego entre elas.
- Disponibilidade do serviço — confirme se seu serviço está em execução e aceitando conexões na porta esperada.
- Zona de disponibilidade não suportada - verifique se suas sub-redes estão em zonas de disponibilidade suportadas. Execute `aws ec2 describe-subnets --subnet-ids <your-subnet-ids> --query 'Subnets[*].[SubnetId,AvailabilityZoneId]'` e verifique as zonas de disponibilidade não suportadas listadas acima.

Excluir uma conexão privada

Você pode excluir conexões privadas não utilizadas usando o AWS Management Console ou a AWS CLI.

Excluir uma conexão privada usando o console

1. Abra o console do AWS DevOps agente.
2. No painel de navegação, escolha Provedores de capacidade e, em seguida, escolha Conexões privadas.
3. Selecione o menu Ações da conexão privada que você deseja excluir e selecione Remover.

A conexão privada será exibida com o status “Removendo conexão”, enquanto o AWS DevOps Agente remove o gateway de recursos gerenciados e os ENIs da sua VPC. Depois que a exclusão for concluída, a conexão não aparecerá mais na sua lista de conexões privadas.

Exclua uma conexão privada usando o AWS CLI

```
aws devops-agent delete-private-connection \
    --name my-mcp-tool-connection
```

A resposta retorna um status de`DELETE_IN_PROGRESS`. AWS DevOps O agente remove o gateway de recursos gerenciados e os ENIs da sua VPC. Depois que a exclusão for concluída, a conexão não aparecerá mais na sua lista de conexões privadas.

Configuração avançada usando os recursos existentes do VPC Lattice

Se sua organização já usa o Amazon VPC Lattice e gerencia suas próprias configurações de recursos, você pode criar uma conexão privada no modo autogerenciado. Em vez de fazer com que o AWS DevOps Agente crie um gateway de recursos para você, você fornece o Amazon Resource Name (ARN) de uma configuração de recurso existente que aponta para seu serviço de destino.

Essa abordagem é útil quando você:

- Quer controle total sobre o gateway de recursos e o ciclo de vida da configuração de recursos.
- Precisa compartilhar configurações de recursos em várias AWS contas ou serviços.
- Exija registros de acesso ao VPC Lattice para monitoramento detalhado do tráfego.

- Execute uma arquitetura de rede hub-and-spoke.

Para criar uma conexão privada autogerenciada com a 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 obter mais detalhes sobre como configurar gateways de recursos e configurações de recursos do VPC Lattice, consulte o Guia do usuário do Amazon VPC Lattice.

Tópicos relacionados

- the section called “VPC endpoints (AWS PrivateLink)”
- the section called “Conectando servidores MCP”
- Configurando recursos para AWS DevOps Agente
- AWS DevOps Segurança do agente
- the section called “DevOps Permissões do Agent IAM”

AWS DevOps Segurança do agente

Este documento fornece informações sobre considerações de segurança, proteção de dados, controles de acesso e recursos de conformidade do AWS DevOps Agent. Use essas informações para entender como o AWS DevOps Agent foi projetado para atender aos seus requisitos de segurança e conformidade.

Multi-layered segurança

AWS DevOps O agente implementa a segurança em várias camadas. Mesmo que permissões mais amplas sejam concedidas à função de IAM do agente, o agente aplica seus próprios controles de acesso internos para limitar o escopo de suas ações.

Recomendamos seguir o princípio do menor privilégio ao configurar as permissões do IAM para o AWS DevOps Agente e implementar a segurança em várias camadas. A defesa profunda garante que nenhuma configuração incorreta possa comprometer a segurança do seu ambiente.

Espaços para agentes

Os Agent Spaces servem como o principal limite de segurança no AWS DevOps Agent. Cada espaço de agente:

- Opera de forma independente com suas próprias configurações e permissões
- Define quais AWS contas e recursos o agente pode acessar
- Estabelece conexões com plataformas de terceiros

Os Agent Spaces mantêm um isolamento estrito para garantir a segurança e evitar o acesso não intencional em diferentes ambientes ou equipes.

Processamento regional e fluxo de dados

AWS DevOps O agente opera globalmente com recursos de processamento regionais. O agente recupera dados operacionais de AWS regiões em todas as AWS contas com acesso concedido no Espaço do Agente configurado. Essa coleta de dados multirregional entre contas garante uma análise abrangente de incidentes, respeitando os limites geográficos para o processamento de inferências.

Uso do Amazon Bedrock e inferência entre regiões

AWS DevOps O agente selecionará automaticamente a região ideal em sua geografia para processar suas solicitações de inferência. Isso maximiza os recursos computacionais disponíveis, a disponibilidade do modelo e oferece a melhor experiência ao cliente. Seus dados permanecerão armazenados somente na região em que seu Espaço do Agente foi criado, no entanto, as solicitações de entrada e os resultados de saída podem ser processados fora dessa região, conforme descrito na lista a seguir. Todos os dados serão transmitidos criptografados pela rede segura da Amazon.

AWS DevOps O agente encaminhará com segurança suas solicitações de inferência para os recursos computacionais disponíveis na área geográfica em que a solicitação foi originada, da seguinte forma:

- Solicitações de inferência originadas na União Europeia serão processadas dentro da União Europeia.
- Solicitações de inferência originadas nos Estados Unidos da América serão processadas nos Estados Unidos.
- Solicitações de inferência originadas na Austrália serão processadas na Austrália.
- Solicitações de inferência originadas no Japão serão processadas no Japão.
- Se uma solicitação de inferência for originada em uma área não listada, ela será processada por padrão nos Estados Unidos da América.
- DevOps Agent e Bedrock não são afetados pelas políticas do cliente nas Políticas de Controle de Serviços (SCPs) ou na Control Tower, que restringem o conteúdo do cliente a regiões específicas
- A Bedrock pode usar regiões diferentes da região de origem em sua geografia para realizar inferências sem estado para otimizar o desempenho e a disponibilidade.

Gerenciamento de identidade e acesso

Métodos de autenticação

AWS DevOps O agente fornece dois métodos de autenticação para fazer login no aplicativo web do AWS DevOps Agent Space:

- AWS Integração com o Identity Center — O método de autenticação principal usa OAuth 2.0 com autenticação baseada em sessão usando cookies. HTTP-only AWS O Identity Center pode se

federar com provedores de identidade externos por meio de protocolos OIDC e SAML padrão, incluindo provedores como Okta, Ping Identity e Microsoft Entra ID. Esse método oferece suporte à autenticação multifatorial por meio do seu provedor de identidade. AWS O Identity Center usa como padrão uma duração de sessão de até 12 horas e pode ser configurado para a duração desejada.

- Link de autenticação do IAM — Um método alternativo fornece acesso direto ao aplicativo web a partir do AWS Management Console usando JWT-based tokens derivados de uma sessão existente do AWS Management Console. Essa opção é útil para avaliar o AWS DevOps Agente antes de implementar a integração completa do Identity Center, bem como para obter acesso administrativo se o aplicativo web do AWS DevOps Agent ficar inacessível por meio da autenticação baseada no Identity Center. As sessões são limitadas a 10 minutos.

Perfis do IAM

AWS DevOps O agente usa funções do IAM para definir as permissões de acesso:

- Função da conta principal — concede ao agente acesso aos recursos na AWS conta em que você criou o Espaço do Agente, bem como acesso às funções secundárias da conta.
- Funções secundárias da conta — concede ao agente acesso aos recursos em AWS contas adicionais conectadas ao Espaço do Agente.
- Função do aplicativo Web — Concede aos usuários acesso aos dados e descobertas da investigação do AWS DevOps Agente no aplicativo web.

Essas funções devem ser configuradas seguindo o princípio do privilégio mínimo, concedendo somente as permissões de leitura necessárias para investigações.

Proteção de dados

Criptografia de dados

AWS DevOps O agente criptografa todos os dados do cliente:

- Criptografia em repouso — Todos os dados são criptografados com chaves AWS gerenciadas.
- Criptografia em trânsito — Todos os registros, métricas, itens de conhecimento, metadados de tickets e outros dados recuperados são criptografados em trânsito dentro da rede privada do agente e para redes externas.

Armazenamento e retenção de dados

Os dados são armazenados na região em que seu Espaço do Agente é criado, enquanto o processamento de inferência pode ocorrer dentro da sua geografia, conforme descrito na seção de uso do Amazon Bedrock acima.

Informações pessoais identificáveis (PII)

AWS DevOps O agente não filtra as informações de PII ao resumir os dados coletados durante investigações, avaliações de recomendações ou respostas de bate-papo. É recomendável que os dados de PII sejam editados antes de serem armazenados nos registros de observabilidade.

Diário do agente e registro de auditoria

Diário do agente

Tanto os recursos de Investigação quanto de Prevenção de Incidentes mantêm diários detalhados que:

- Registre todas as etapas de raciocínio e ações tomadas
- Crie transparência total nos processos de tomada de decisão dos agentes
- Não pode ser modificado pelos agentes depois de registrados, evitando que ataques, como injeção imediata, ocultem ações importantes
- Inclua todas as mensagens de bate-papo da página Investigação

AWS CloudTrail integração

Todas as chamadas da API do AWS DevOps agente são capturadas AWS CloudTrail automaticamente pela AWS conta de hospedagem. Usando as informações coletadas por CloudTrail, você pode determinar:

- A solicitação que foi feita ao agente
- O endereço IP do qual a solicitação foi feita.
- Quem fez a solicitação.
- Quando ela foi feita

Proteção imediata de injeção

Um ataque de injeção imediata ocorre quando um invasor incorpora instruções maliciosas em dados externos, como uma página da Web ou documento, que um sistema generativo de IA processará posteriormente. AWS DevOps O agente consome nativamente muitas fontes de dados como parte de suas operações normais, incluindo registros, tags de recursos e outros dados operacionais. AWS DevOps O agente protege contra ataques de injeção imediata por meio das proteções abaixo, mas é importante garantir que todas as fontes de dados conectadas e o acesso do usuário a essas fontes de dados sejam confiáveis. Consulte a seção Modelo de responsabilidade compartilhada para obter mais informações.

Proteções de injeção rápida:

- Capacidades de gravação limitadas — As ferramentas disponíveis para o agente não são capazes de alterar recursos, com exceção da abertura de tickets e casos de suporte. Isso evita que instruções maliciosas modifiquem sua infraestrutura ou seus aplicativos.
- Aplicação do limite da conta — O AWS DevOps agente opera somente dentro do limite permitido pelas funções atribuídas ao agente nas contas primária e secundária AWS conectada. O agente não pode acessar ou modificar recursos fora do escopo configurado.
- Proteções de segurança de IA — O AWS DevOps agente usa modelos com proteções de segurança de IA de nível 3 (ASL-3). Essas proteções incluem classificadores que detectam e previnem ataques imediatos de injeção antes que eles possam afetar o comportamento do agente.
- Trilha de auditoria imutável — O diário do agente registra todas as etapas de raciocínio e ações tomadas. As entradas do diário não podem ser modificadas pelo agente depois de registradas, evitando que ataques de injeção imediata ocultem ações maliciosas.

Embora o AWS DevOps Agent forneça várias camadas de proteção contra ataques imediatos de injeção, certas configurações podem aumentar o risco:

- Ferramentas personalizadas de servidor MCP — O recurso traga seu próprio MCP permite que você introduza ferramentas personalizadas ao agente, o que pode apresentar oportunidades adicionais para injeção imediata. As ferramentas personalizadas podem não ter os mesmos controles de segurança que as ferramentas nativas do AWS DevOps Agente, e instruções maliciosas podem potencialmente utilizar essas ferramentas de maneiras não intencionais. Consulte a seção Modelo de responsabilidade compartilhada para obter mais informações.

- Ataques de usuários autorizados — Usuários autorizados a operar dentro dos limites da AWS conta ou das ferramentas conectadas têm uma chance maior de tentar um ataque contra o agente. Esses usuários podem modificar as fontes de dados que o agente consome, como registros ou tags de recursos, facilitando a incorporação de instruções maliciosas que o agente processará.

Para mitigar esses riscos:

1. Analise e teste cuidadosamente os servidores MCP personalizados antes de implantá-los nos Agent Spaces.
   a. Certifique-se de que eles só tenham permissão para realizar ações somente para leitura
   b. Verifique se os usuários de ferramentas externas acessadas pelos servidores MCP são entidades confiáveis, pois os AWS DevOps agentes que fazem interface com o MCP dependem da relação de confiança implícita estabelecida entre esses usuários da ferramenta e o agente AWS DevOps
2. Aplique o princípio do menor privilégio ao conceder aos usuários acesso a sistemas que fornecem dados ao agente
3. Audite regularmente quais servidores MCP estão conectados aos seus Agent Spaces
4. Como qualquer conteúdo recuperado dos URLs da lista de permissões pode tentar manipular o comportamento do agente, inclua somente fontes confiáveis na sua lista de permissões.

Segurança de integração

AWS DevOps O agente oferece suporte a vários tipos de integração, cada um com seu próprio modelo de segurança:

- Integrações bidirecionais nativas — Built-in integrações que podem enviar dados ao agente e receber atualizações do agente. Isso usa os métodos de autenticação do fornecedor
- Servidores MCP — servidores Remote Model Context Protocol que utilizam fluxos de autenticação OAuth 2.0 e chaves de API para se comunicar com segurança com sistemas externos.
- Acionadores de webhook — gatilhos de investigação de serviços remotos, como tickets ou sistemas de observabilidade. Os webhooks usam o Código de Autenticação de Hash-based Mensagens (HMAC) para fins de segurança.
- Comunicação externa — Integrações como o Slack e os sistemas de emissão de bilhetes recebem atualizações do agente, mas ainda não oferecem suporte à comunicação bidirecional.

Provedores de registro

Algumas ferramentas externas são autenticadas no nível da conta e compartilhadas entre todos os Agent Spaces na conta. Ao registrar essas ferramentas, você se autentica uma vez no nível da conta e, em seguida, cada Espaço do Agente pode se conectar a recursos específicos dentro dessa conexão registrada.

As ferramentas a seguir usam o registro em nível de conta:

- GitHub— Usa o fluxo OAuth para autenticação. Depois de se registrar GitHub no nível da conta, cada Espaço do Agente pode se conectar a repositórios específicos em sua GitHub organização.
- Dynatrace — usa autenticação de token OAuth. Depois de registrar o Dynatrace no nível da conta, cada Agent Space pode se conectar a ambientes específicos do Dynatrace ou configurações de monitoramento.
- Slack — usa autenticação de token OAuth. Depois de registrar o Slack no nível da conta, cada espaço do agente pode se conectar a canais específicos do Slack.
- Datadog — usa MCP com fluxo OAuth para autenticação. Depois de registrar o Datadog no nível da conta, cada Espaço do Agente pode se conectar a recursos específicos de monitoramento do Datadog.
- New Relic — usa autenticação por chave de API. Depois de registrar a New Relic no nível da conta, cada Espaço do Agente pode se conectar a configurações específicas de monitoramento da New Relic.
- Splunk — Usa autenticação de token do portador. Depois de registrar o Splunk no nível da conta, cada Espaço do Agente pode se conectar a fontes de dados específicas do Splunk.
- GitLab— Usa autenticação por token de acesso. Depois de se registrar GitLab no nível da conta, cada Espaço do Agente pode se conectar a GitLab repositórios específicos.
- ServiceNow— Usa a autenticação do cliente key/token OAuth. Depois de se registrar ServiceNow no nível da conta, cada espaço do agente pode se conectar a ServiceNow instâncias específicas ou filas de tickets.
- Servidores MCP remotos acessíveis ao público em geral — Use o fluxo OAuth para autenticação. Depois de registrar um servidor MCP remoto no nível da conta, cada Espaço do Agente pode se conectar a recursos específicos expostos por esse servidor.

Conectividade de rede

AWS DevOps O agente se conecta a seus sistemas de terceiros e servidores MCP remotos para realizar investigações e outras operações.

Tráfego de entrada de AWS DevOps Agente para seus sistemas

AWS DevOps O agente inicia conexões de saída com seus sistemas de terceiros e servidores MCP remotos, que chegam como tráfego de entrada à sua infraestrutura. A forma como você protege esse tráfego depende de como suas ferramentas são hospedadas:

- Ferramentas hospedadas de forma privada — Se suas ferramentas puderem ser acessadas de dentro de uma AWS VPC, você poderá usar as conexões privadas do AWS DevOps Agente para manter o tráfego isolado AWS das redes e fora da Internet pública. Para obter mais informações, consulte the section called “Conectando-se a ferramentas hospedadas de forma privada”.
- Ferramentas hospedadas publicamente — Se suas ferramentas puderem ser acessadas pela Internet pública e usarem listas de permissões de IP ou regras de firewall, você deverá permitir o tráfego de entrada dos seguintes endereços IP de origem do AWS DevOps agente:
  - Ásia-Pacífico (Sydney) (ap-southeast-2)
    - `13.237.95.197`
    - `13.238.84.102`
    - `52.64.174.242`
    - `13.211.249.13`
    - `15.134.235.54`
    - `3.107.145.226`
  - Ásia Pacific (Tóquio) (ap-northeast-1)
    - `13.192.12.233`
    - `35.74.181.230`
    - `57.183.50.158`
    - `13.114.228.89`
    - `54.150.140.28`
    - `46.51.224.121`
  - Europa (Frankfurt) (eu-central-1)
    - `18.158.110.140`

  - 52.57.96.160
  - 52.59.55.56
  - 63.183.67.111
  - 63.184.95.132
  - 63.184.36.38
- Europa (Irlanda) (eu-west-1)
  - 34.251.85.24
  - 52.30.157.157
  - 52.51.192.222
  - 99.81.41.52
  - 54.246.170.103
  - 52.212.224.65
- Leste dos EUA (Norte da Virgínia) (us-east-1)
  - 34.228.181.128
  - 44.219.176.187
  - 54.226.244.221
  - 100.56.22.59
  - 3.234.39.4
  - 44.215.92.10
- Oeste dos EUA (Oregon) (us-west-2)
  - 34.212.16.133
  - 52.89.67.212
  - 54.187.135.61
  - 34.209.115.89
  - 44.224.219.86
  - 54.201.89.243

Tráfego de saída da sua VPC para AWS DevOps Agente

Para tráfego de saída da sua AWS VPC AWS DevOps para o agente (por exemplo, the section called “Invocando o DevOps Agente por meio do Webhook” usando), você pode usar VPC Endpoints

para manter esse tráfego de rede isolado das redes. AWS Para obter mais informações, consulte the section called “VPC endpoints (AWS PrivateLink)”.

Modelo de responsabilidade compartilhada

AWS responsabilidades

AWS é responsável por:

- Manter a segurança dos dados recuperados pelo agente
- Protegendo as ferramentas nativas disponíveis para uso pelo agente
- Protegendo a infraestrutura que executa o AWS DevOps Agent

Responsabilidades do cliente

Os clientes são responsáveis por:

- Gerenciando o acesso do usuário ao espaço do agente
- Limitar o acesso a usuários confiáveis de sistemas externos que fornecem informações ao agente, como serviços e recursos que produzem registros, CloudTrail eventos, tickets e muito mais, que podem ser usados para tentar uma injeção maliciosa imediata.
- Garanta que todas as fontes de dados conectadas tenham dados confiáveis que provavelmente não serão usados para tentar ataques de injeção imediata
- Garantindo que as integrações de servidores MCP “traga seu próprio” operem com segurança
- Garantir que as funções do IAM atribuídas ao agente tenham um escopo adequado
- Editando dados de PII antes de armazená-los em registros de observabilidade e outras fontes de dados do agente
- Seguindo a prática recomendada de conceder somente permissões de leitura às fontes de dados conectadas, incluindo servidores MCP “traga seus próprios”

Uso de dados

AWS não usa dados de agentes, mensagens de bate-papo ou dados de fontes de dados integradas para treinar modelos ou melhorar o produto. O Espaço do AWS DevOps Agente usa o feedback do

cliente no produto para melhorar as respostas e investigações do agente, mas AWS não o usa para melhorar o serviço em si.

DevOps Permissões do Agent IAM

AWS DevOps O agente usa ações de AWS Identity and Access Management (IAM) específicas do serviço para controlar o acesso aos seus recursos e capacidades. Essas ações determinam o que os usuários podem fazer no console do AWS DevOps agente e no Operator Web App. Isso é separado das permissões da API de AWS serviço que o próprio agente usa para investigar seus recursos.

Para obter mais informações sobre como limitar o acesso do agente, consulte Limitar o acesso do agente em uma AWS conta.

Ações de gerenciamento do Agent Space

Essas ações controlam o acesso à configuração e ao gerenciamento do Agent Space:

- aidevops: GetAgentSpace — Permite que os usuários visualizem detalhes sobre um Espaço do Agente, incluindo sua configuração, status e contas associadas. Os usuários precisam dessa permissão para acessar um Espaço do Agente no AWS Management Console.
- aidevops: GetAssociation — Permite que os usuários visualizem detalhes sobre uma associação de conta específica, incluindo a configuração da função do IAM e o status da conexão.
- aidevops: ListAssociations — Permite que os usuários listem todas as associações de AWS contas configuradas para um Espaço do Agente, incluindo contas primárias e secundárias.

Ações de investigação e execução

Essas ações controlam o acesso aos recursos de investigação de incidentes:

- aidevops: ListExecutions — Permite que os usuários visualizem metadados de execução, incluindo ID, status e muito mais, para investigações, mitigações, avaliações e conversas de bate-papo associadas a uma tarefa.
- aidevops: ListJournalRecords — Permite que os usuários acessem registros detalhados que mostram as etapas de raciocínio do agente, as ações tomadas e as fontes de dados consultadas durante uma investigação, mitigação, avaliação e conversa por chat. Isso é útil para entender como o agente chegou às suas conclusões.

Ações de gerenciamento de chat

O Chat exige as seguintes permissões do IAM para funcionar:

- aidevops: ListChats — Permite que os usuários listem e acessem o histórico de conversas do chat.
- aidevops: CreateChat — Permite que os usuários criem novas conversas de bate-papo.
- aidevops: SendMessage — Permite que os usuários enviem consultas e recebam respostas de streaming.

Ações de topologia e descoberta

Essas ações controlam o acesso aos recursos de mapeamento de recursos do aplicativo:

- aidevops: DiscoverTopology — Permite que os usuários acionem a descoberta e o mapeamento de topologia para um Agent Space. Essa ação inicia o processo de verificação de AWS contas e criação da topologia de recursos do aplicativo.

Ações de prevenção e recomendação

Essas ações controlam o acesso ao recurso Prevenção:

- aidevops: ListGoals — Permite que os usuários visualizem as metas e objetivos de prevenção pelos quais o agente está trabalhando com base nos padrões de incidentes recentes.
- aidevops: ListRecommendations — Permite que os usuários visualizem todas as recomendações geradas pelo recurso de Prevenção, incluindo sua prioridade e categoria.
- aidevops: GetRecommendation — Permite que os usuários visualizem informações detalhadas sobre uma recomendação específica, incluindo os incidentes que ela teria evitado e orientações de implementação.

Ações de gerenciamento de tarefas do backlog

Essas ações controlam a capacidade de gerenciar recomendações como tarefas de lista de pendências:

- aidevops: CreateBacklogTask — Permite que os usuários criem uma tarefa de investigação de incidentes ou avaliação de prevenção.

- aidevops: UpdateBacklogTask — Permite que os usuários aprovem um plano de mitigação ou cancelem uma investigação ou avaliação ativa.
- aidevops: GetBacklogTask — Permite que os usuários recuperem detalhes sobre uma tarefa específica.
- aidevops: ListBacklogTasks — Permite que os usuários listem tarefas para um Espaço do Agente, filtradas por tipo de tarefa, status, prioridade ou horário de criação.

Ações de gestão do conhecimento

Essas ações controlam a capacidade de adicionar e gerenciar conhecimento personalizado que o agente pode usar durante as investigações:

- aidevops: CreateKnowledgeItem — Permite que os usuários adicionem itens de conhecimento personalizados, como habilidades, guias de solução de problemas ou informações específicas do aplicativo que o agente deve consultar.
- aidevops: ListKnowledgeItems — Permite que os usuários visualizem todos os itens de conhecimento configurados para um Espaço do Agente.
- aidevops: GetKnowledgeItem — Permite que os usuários recuperem os detalhes de um item de conhecimento específico.
- aidevops: UpdateKnowledgeItem — Permite que os usuários modifiquem os itens de conhecimento existentes para manter as informações atualizadas.
- aidevops: DeleteKnowledgeItem — Permite que os usuários removam itens de conhecimento que não são mais relevantes.

AWS Ações de integração do Support

Essas ações controlam a integração com os casos de AWS Support:

- aidevops: InitiateChatForCase — Permite que os usuários iniciem uma sessão de bate-papo com o AWS Support diretamente de uma investigação, fornecendo automaticamente contexto sobre o incidente.
- aidevops: EndChatForCase — Permite que os usuários encerrem uma sessão ativa de bate-papo sobre casos do AWS Support.
- aidevops: DescribeSupportLevel — Permite que os usuários verifiquem o nível do plano AWS Support da conta para determinar as opções de suporte disponíveis.

Ações de uso e monitoramento

Essas ações controlam o acesso às informações de uso:

- aidevops: GetAccountUsage — Permite que os usuários visualizem a cota mensal do AWS DevOps agente para horas de investigação, horas de avaliação de prevenção e solicitações de bate-papo, bem como o uso do mês atual.

Exemplos comuns de políticas do IAM

Política do administrador

Essa política concede acesso total a todos os recursos AWS DevOps do Agente:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "aidevops:*",
      "Resource": "*"
    }
  ]
}
```

Política do operador

Essa política concede acesso a recursos de investigação e prevenção sem recursos administrativos:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aidevops:GetAgentSpace",
        "aidevops:InvokeAgent",
        "aidevops:ListExecutions",
        "aidevops:ListJournalRecords",
        "aidevops:ListAssociations",
        "aidevops:GetAssociation",
```

```
        "aidevops:DiscoverTopology",
        "aidevops:ListRecommendations",
        "aidevops:GetRecommendation",
        "aidevops:CreateBacklogTask",
        "aidevops:UpdateBacklogTask",
        "aidevops:GetBacklogTask",
        "aidevops:ListBacklogTasks",
        "aidevops:ListKnowledgeItems",
        "aidevops:GetKnowledgeItem",
        "aidevops:InitiateChatForCase",
        "aidevops:EndChatForCase",
        "aidevops:ListChats",
        "aidevops:CreateChat",
        "aidevops:SendMessage",
        "aidevops:ListGoals",
        "aidevops:CreateKnowledgeItem",
        "aidevops:UpdateKnowledgeItem",
        "aidevops:DescribeSupportLevel",
        "aidevops:ListPendingMessages"
      ],
      "Resource": "*"
    }
  ]
}
```

Read-only política

Essa política concede acesso somente para visualização às investigações e recomendações:

```
{
  "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": "*"
    }
  ]
}
```

Usando funções vinculadas a serviços para AWS DevOps Agente

AWS DevOps O agente usa AWS funções vinculadas ao serviço Identity and Access Management (IAM). Uma função vinculada ao serviço é um tipo exclusivo de função do IAM vinculada diretamente ao AWS DevOps Agente. Service-linked as funções são predefinidas pelo AWS DevOps Agente e incluem todas as permissões que o serviço exige para chamar outros AWS serviços em seu nome.

Service-linked permissões de função

A função vinculada a serviço `AWSServiceRoleForAIDevOps` confia na entidade principal do serviço `aidevops.amazonaws.com` para assumir a função.

A função usa a política gerenciada `AWSServiceRoleForAIDevOpsPolicy` com as seguintes permissões:

- `cloudwatch:PutMetricData`— Publique métricas de uso no `AWS/AIDevOps` CloudWatch namespace. Definido por uma `cloudwatch:namespace` condição para permitir apenas o `AWS/AIDevOps` namespace.
- `vpc-lattice:CreateResourceGateway`— Crie gateways de recursos VPC Lattice para conexões privadas. Definido por uma `aws:RequestTag/AWSAIDevOpsManaged` condição para que o serviço só possa criar gateways de recursos que contenham a `AWSAIDevOpsManaged` tag.
- `vpc-lattice:TagResource`— Identifique os gateways de recursos do VPC Lattice. Determinado por uma `aws:RequestTag/AWSAIDevOpsManaged` condição.
- `vpc-lattice:DeleteResourceGateway`— Exclua os gateways de recursos do VPC Lattice. Definido por uma `aws:ResourceTag/AWSAIDevOpsManaged` condição para que o serviço só possa excluir os gateways de recursos criados por ele.
- `vpc-lattice:GetResourceGateway`— Recupere informações sobre os gateways de recursos do VPC Lattice. Definido por uma `aws:ResourceTag/AWSAIDevOpsManaged` condição para que o serviço só possa ler os gateways de recursos criados por ele.

- `ec2:DescribeVpcs`,`ec2:DescribeSubnets`, `ec2:DescribeSecurityGroups` — Recupere informações sobre os recursos de rede VPC necessários para configurar gateways de recursos. Essas ações somente para leitura se aplicam a todos os recursos da VPC porque a API do EC2 não oferece suporte a permissões em nível de recurso para chamadas Describe.
- `iam:CreateServiceLinkedRole`— Crie a função vinculada ao serviço VPC Lattice necessária para operações de gateway de recursos. Essa permissão tem como escopo somente o principal do `vpc-lattice.amazonaws.com` serviço e não pode ser usada para criar funções vinculadas ao serviço para nenhum outro serviço.

Como criar uma função vinculada ao serviço

Você não precisa criar manualmente a função vinculada a serviço `AWSServiceRoleForAIDevOps`. Quando você começa a usar o AWS DevOps Agente, o serviço cria a função vinculada ao serviço para você.

Para permitir que o serviço crie a função em seu nome, você precisa ter a `iam:CreateServiceLinkedRole` permissão. Recomendamos definir o escopo dessa permissão com a `iam:AWSServiceName` condição de `aidevops.amazonaws.com` seguir o princípio do menor privilégio. Para obter mais informações, consulte permissões de Service-linked função.

Editando a função vinculada ao serviço

Você não pode editar a função vinculada a serviço `AWSServiceRoleForAIDevOps`. Depois que a função é criada, você não pode alterar o nome da função porque várias entidades podem referenciar a função pelo nome. No entanto, você poderá editar a descrição do perfil usando o IAM. Para obter mais informações, consulte Editando uma função vinculada ao serviço.

Excluindo uma função vinculada ao serviço

Se você não precisar mais usar o AWS DevOps Agente, recomendamos que você exclua a função `AWSServiceRoleForAIDevOps` vinculada ao serviço. Antes de excluir a função, você deve primeiro remover todas as conexões privadas configuradas no seu Espaço do Agente. A exclusão da função vinculada ao serviço não remove automaticamente os gateways de recursos do VPC Lattice marcados com os `AWSAIDevOpsManaged` que foram criados anteriormente pelo serviço. Você deve excluir esses gateways de recursos manualmente se eles não forem mais necessários. Para obter mais informações, consulte Excluindo uma função vinculada ao serviço.

AWS Políticas gerenciadas para AWS DevOps Agente

AWS aborda muitos casos de uso comuns fornecendo políticas autônomas do IAM que são criadas e administradas pela AWS. Essas políticas AWS gerenciadas concedem as permissões necessárias para casos de uso comuns, para que você possa evitar a necessidade de investigar quais permissões são necessárias. Para obter mais informações, consulte políticas AWS gerenciadas no _Guia do usuário do IAM_.

As políticas AWS gerenciadas a seguir, que você pode anexar aos usuários em sua conta, são específicas do AWS DevOps Agente.

AIDevOpsAgentReadOnlyAccess

Fornece acesso somente de leitura ao Amazon DevOps Agent por meio do AWS Management Console

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "AIDevOpsAgentReadOnlyAccess",
   "Effect": "Allow",
   "Action": [
    "aidevops:Get*",
    "aidevops:List*",
    "aidevops:SearchServiceAccessibleResource"
   ],
   "Resource": "*"
  }
 ]
}
```

AIDevOpsAgentFullAccess

Fornece acesso total ao Amazon DevOps Agent por meio do AWS Management Console

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

```
    "Action": [
     "aidevops:CreateAgentSpace",
     "aidevops:DeleteAgentSpace",
     "aidevops:GetAgentSpace",
     "aidevops:ListAgentSpaces",
     "aidevops:UpdateAgentSpace"
    ],
    "Resource": "*"
   },
   {
    "Sid": "AIDevOpsServiceAccess",
    "Effect": "Allow",
    "Action": [
     "aidevops:DeregisterService",
     "aidevops:GetService",
     "aidevops:ListServices",
     "aidevops:RegisterService",
     "aidevops:SearchServiceAccessibleResource"
    ],
    "Resource": "*"
   },
   {
    "Sid": "AIDevOpsAssociationAccess",
    "Effect": "Allow",
    "Action": [
     "aidevops:AssociateService",
     "aidevops:DisassociateService",
     "aidevops:GetAssociation",
     "aidevops:ListAssociations",
     "aidevops:UpdateAssociation",
     "aidevops:ValidateAwsAssociations"
    ],
    "Resource": "*"
   },
   {
    "Sid": "AIDevOpsWebhookAccess",
    "Effect": "Allow",
    "Action": [
     "aidevops:ListWebhooks"
    ],
    "Resource": "*"
   },
   {
    "Sid": "AIDevOpsOperatorAppAccess",
```

```
    "Effect": "Allow",
    "Action": [
     "aidevops:DisableOperatorApp",
     "aidevops:EnableOperatorApp",
     "aidevops:GetOperatorApp",
     "aidevops:UpdateOperatorAppIdpConfig"
    ],
    "Resource": "*"
   },
   {
    "Sid": "AIDevOpsKnowledgeAccess",
    "Effect": "Allow",
    "Action": [
     "aidevops:CreateKnowledgeItem",
     "aidevops:DeleteKnowledgeItem",
     "aidevops:GetKnowledgeItem",
     "aidevops:ListKnowledgeItems",
     "aidevops:ListKnowledgeItemVersions",
     "aidevops:UpdateKnowledgeItem"
    ],
    "Resource": "*"
   },
   {
    "Sid": "AIDevOpsBacklogAccess",
    "Effect": "Allow",
    "Action": [
     "aidevops:CreateBacklogTask",
     "aidevops:GetBacklogTask",
     "aidevops:ListBacklogTasks",
     "aidevops:ListGoals",
     "aidevops:UpdateBacklogTask",
     "aidevops:UpdateGoal"
    ],
    "Resource": "*"
   },
   {
    "Sid": "AIDevOpsRecommendationAccess",
    "Effect": "Allow",
    "Action": [
     "aidevops:GetRecommendation",
     "aidevops:ListRecommendations",
     "aidevops:UpdateRecommendation"
    ],
    "Resource": "*"
```

```
  },
  {
   "Sid": "AIDevOpsAgentChatAccess",
   "Effect": "Allow",
   "Action": [
    "aidevops:CreateChat",
    "aidevops:ListChats",
    "aidevops:ListPendingMessages",
    "aidevops:SendMessage"
   ],
   "Resource": "*"
  },
  {
   "Sid": "AIDevOpsJournalAccess",
   "Effect": "Allow",
   "Action": [
    "aidevops:ListExecutions",
    "aidevops:ListJournalRecords"
   ],
   "Resource": "*"
  },
  {
   "Sid": "AIDevOpsTopologyAccess",
   "Effect": "Allow",
   "Action": [
    "aidevops:DiscoverTopology"
   ],
   "Resource": "*"
  },
  {
   "Sid": "AIDevOpsSupportAccess",
   "Effect": "Allow",
   "Action": [
    "aidevops:DescribeServices",
    "aidevops:DescribeSupportLevel",
    "aidevops:EndChatForCase",
    "aidevops:InitiateChatForCase"
   ],
   "Resource": "*"
  },
  {
   "Sid": "AIDevOpsUsageAccess",
   "Effect": "Allow",
   "Action": [
```

```
    "aidevops:GetAccountUsage"
   ],
   "Resource": "*"
  },
  {
   "Sid": "AIDevOpsTaggingAccess",
   "Effect": "Allow",
   "Action": [
    "aidevops:ListTagsForResource",
    "aidevops:TagResource",
    "aidevops:UntagResource"
   ],
   "Resource": "*"
  },
  {
   "Sid": "AIDevOpsVendedLogs",
   "Effect": "Allow",
   "Action": [
    "aidevops:AllowVendedLogDeliveryForResource"
   ],
   "Resource": "*"
  }
 ]
}
```

AIDevOpsOperatorAppAccessPolicy

Fornece acesso para usar o aplicativo web do AWS DevOps operador para um Espaço do 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",
    "aidevops:DescribeServices"
   ],
   "Resource": "arn:aws:aidevops:*:*:agentspace/${aws:PrincipalTag/AgentSpaceId}",
   "Condition": {
    "StringEquals": {
     "aws:ResourceAccount": "${aws:PrincipalAccount}"
    }
   }
  },
  {
   "Sid": "AllowOperatorAccountActions",
   "Effect": "Allow",
   "Action": [
    "aidevops:GetAccountUsage"
   ],
   "Resource": "*",
   "Condition": {
    "StringEquals": {
     "aws:ResourceAccount": "${aws:PrincipalAccount}"
    }
   }
  },
  {
   "Sid": "AllowSupportOperatorActions",
```

```
    "Effect": "Allow",
    "Action": [
     "support:DescribeCases",
     "support:DescribeServices",
     "support:InitiateChatForCase",
     "support:DescribeSupportLevel"
    ],
    "Resource": "*",
    "Condition": {
     "StringEquals": {
      "aws:ResourceAccount": "${aws:PrincipalAccount}"
     }
    }
   },
   {
    "Sid": "AllowSecretsManagerOperatorActions",
    "Effect": "Allow",
    "Action": [
     "secretsmanager:CreateSecret",
     "secretsmanager:ListSecrets"
    ],
    "Resource": "*",
    "Condition": {
     "StringEquals": {
      "aws:ResourceAccount": "${aws:PrincipalAccount}"
     }
    }
   }
  ]
}
```

AIDevOpsAgentAccessPolicy

Fornece as permissões exigidas pelo AWS DevOps agente para conduzir investigações e realizar análises sobre AWS os recursos do cliente.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AIOPSServiceAccess",
            "Effect": "Allow",
            "Action": [
```

```
                "access-analyzer:GetAnalyzer",
                "access-analyzer:List*",
                "acm-pca:Describe*",
                "acm-pca:GetCertificate",
                "acm-pca:GetCertificateAuthorityCertificate",
                "acm-pca:GetCertificateAuthorityCsr",
                "acm-pca:List*",
                "acm:DescribeCertificate",
                "acm:GetAccountConfiguration",
                "aidevops:GetKnowledgeItem",
                "aidevops:ListKnowledgeItems",
                "airflow:List*",
                "amplify:GetApp",
                "amplify:GetBranch",
                "amplify:GetDomainAssociation",
                "amplify:List*",
                "aoss:BatchGetCollection",
                "aoss:BatchGetLifecyclePolicy",
                "aoss:BatchGetVpcEndpoint",
                "aoss:GetAccessPolicy",
                "aoss:GetSecurityConfig",
                "aoss:GetSecurityPolicy",
                "aoss:List*",
                "appconfig:GetApplication",
                "appconfig:GetConfigurationProfile",
                "appconfig:GetEnvironment",
                "appconfig:GetHostedConfigurationVersion",
                "appconfig:List*",
                "appflow:Describe*",
                "appflow:List*",
                "application-autoscaling:Describe*",
                "application-signals:BatchGetServiceLevelObjectiveBudgetReport",
                "application-signals:GetService",
                "application-signals:GetServiceLevelObjective",
                "application-signals:List*",
                "applicationinsights:Describe*",
                "applicationinsights:List*",
                "apprunner:Describe*",
                "apprunner:List*",
                "appstream:Describe*",
                "appstream:List*",
                "appsync:GetApiAssociation",
                "appsync:GetDataSource",
                "appsync:GetDomainName",
```

```
                "appsync:GetFunction",
                "appsync:GetGraphqlApi",
                "appsync:GetGraphqlApiEnvironmentVariables",
                "appsync:GetIntrospectionSchema",
                "appsync:GetResolver",
                "appsync:GetSourceApiAssociation",
                "appsync:List*",
                "aps:Describe*",
                "aps:List*",
                "arc-zonal-shift:GetManagedResource",
                "arc-zonal-shift:List*",
                "athena:GetCapacityAssignmentConfiguration",
                "athena:GetCapacityReservation",
                "athena:GetDataCatalog",
                "athena:GetNamedQuery",
                "athena:GetPreparedStatement",
                "athena:GetWorkGroup",
                "athena:List*",
                "auditmanager:GetAssessment",
                "auditmanager:List*",
                "autoscaling:Describe*",
                "backup-gateway:GetHypervisor",
                "backup-gateway:List*",
                "backup:Describe*",
                "backup:GetBackupPlan",
                "backup:GetBackupSelection",
                "backup:GetBackupVaultAccessPolicy",
                "backup:GetBackupVaultNotifications",
                "backup:GetRestoreTestingPlan",
                "backup:GetRestoreTestingSelection",
                "backup:List*",
                "batch:DescribeComputeEnvironments",
                "batch:DescribeJobQueues",
                "batch:DescribeSchedulingPolicies",
                "batch:List*",
                "bedrock:GetAgent",
                "bedrock:GetAgentActionGroup",
                "bedrock:GetAgentAlias",
                "bedrock:GetAgentKnowledgeBase",
                "bedrock:GetDataSource",
                "bedrock:GetGuardrail",
                "bedrock:GetKnowledgeBase",
                "bedrock:List*",
                "budgets:Describe*",
```

```
                "budgets:List*",
                "ce:Describe*",
                "ce:Get*",
                "ce:List*",
                "chatbot:Describe*",
                "chatbot:GetMicrosoftTeamsChannelConfiguration",
                "chatbot:List*",
                "cleanrooms-ml:GetTrainingDataset",
                "cleanrooms-ml:List*",
                "cleanrooms:GetAnalysisTemplate",
                "cleanrooms:GetCollaboration",
                "cleanrooms:GetConfiguredTable",
                "cleanrooms:GetConfiguredTableAnalysisRule",
                "cleanrooms:GetConfiguredTableAssociation",
                "cleanrooms:GetMembership",
                "cleanrooms:List*",
                "cloudformation:Describe*",
                "cloudformation:GetResource",
                "cloudformation:GetStackPolicy",
                "cloudformation:GetTemplate",
                "cloudformation:List*",
                "cloudfront:Describe*",
                "cloudfront:GetCachePolicy",
                "cloudfront:GetCloudFrontOriginAccessIdentity",
                "cloudfront:GetContinuousDeploymentPolicy",
                "cloudfront:GetDistribution",
                "cloudfront:GetDistributionConfig",
                "cloudfront:GetFunction",
                "cloudfront:GetKeyGroup",
                "cloudfront:GetMonitoringSubscription",
                "cloudfront:GetOriginAccessControl",
                "cloudfront:GetOriginRequestPolicy",
                "cloudfront:GetPublicKey",
                "cloudfront:GetRealtimeLogConfig",
                "cloudfront:GetResponseHeadersPolicy",
                "cloudfront:List*",
                "cloudtrail:Describe*",
                "cloudtrail:GetChannel",
                "cloudtrail:GetEventConfiguration",
                "cloudtrail:GetEventDataStore",
                "cloudtrail:GetEventSelectors",
                "cloudtrail:GetInsightSelectors",
                "cloudtrail:GetQueryResults",
                "cloudtrail:GetResourcePolicy",
```

```
                "cloudtrail:GetTrail",
                "cloudtrail:GetTrailStatus",
                "cloudtrail:List*",
                "cloudtrail:LookupEvents",
                "cloudtrail:StartQuery",
                "cloudwatch:Describe*",
                "cloudwatch:GenerateQuery",
                "cloudwatch:GetDashboard",
                "cloudwatch:GetInsightRuleReport",
                "cloudwatch:GetMetricData",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:GetMetricStream",
                "cloudwatch:GetService",
                "cloudwatch:GetServiceLevelObjective",
                "cloudwatch:List*",
                "codeartifact:Describe*",
                "codeartifact:GetDomainPermissionsPolicy",
                "codeartifact:GetRepositoryPermissionsPolicy",
                "codeartifact:List*",
                "codebuild:BatchGetFleets",
                "codebuild:List*",
                "codecommit:GetRepository",
                "codecommit:GetRepositoryTriggers",
                "codedeploy:BatchGetDeployments",
                "codedeploy:BatchGetDeploymentTargets",
                "codedeploy:GetApplication",
                "codedeploy:GetDeploymentConfig",
                "codedeploy:GetDeploymentTarget",
                "codedeploy:List*",
                "codeguru-profiler:Describe*",
                "codeguru-profiler:GetNotificationConfiguration",
                "codeguru-profiler:GetPolicy",
                "codeguru-profiler:List*",
                "codeguru-reviewer:Describe*",
                "codeguru-reviewer:List*",
                "codepipeline:GetPipeline",
                "codepipeline:GetPipelineState",
                "codepipeline:List*",
                "codestar-connections:GetConnection",
                "codestar-connections:GetRepositoryLink",
                "codestar-connections:GetSyncConfiguration",
                "codestar-connections:List*",
                "codestar-notifications:Describe*",
                "codestar-notifications:List*",
```

```
                "cognito-identity:DescribeIdentityPool",
                "cognito-identity:GetIdentityPoolRoles",
                "cognito-identity:ListIdentityPools",
                "cognito-identity:ListTagsForResource",
                "cognito-idp:AdminListGroupsForUser",
                "cognito-idp:DescribeIdentityProvider",
                "cognito-idp:DescribeResourceServer",
                "cognito-idp:DescribeRiskConfiguration",
                "cognito-idp:DescribeUserImportJob",
                "cognito-idp:DescribeUserPool",
                "cognito-idp:DescribeUserPoolDomain",
                "cognito-idp:GetGroup",
                "cognito-idp:GetLogDeliveryConfiguration",
                "cognito-idp:GetUICustomization",
                "cognito-idp:GetUserPoolMfaConfig",
                "cognito-idp:GetWebACLForResource",
                "cognito-idp:ListGroups",
                "cognito-idp:ListIdentityProviders",
                "cognito-idp:ListResourceServers",
                "cognito-idp:ListUserPoolClients",
                "cognito-idp:ListUserPools",
                "cognito-idp:ListTagsForResource",
                "comprehend:Describe*",
                "comprehend:List*",
                "config:Describe*",
                "config:GetStoredQuery",
                "config:List*",
                "connect:Describe*",
                "connect:GetTaskTemplate",
                "connect:List*",
                "databrew:Describe*",
                "databrew:List*",
                "datapipeline:Describe*",
                "datapipeline:GetPipelineDefinition",
                "datapipeline:List*",
                "datasync:Describe*",
                "datasync:List*",
                "deadline:GetFarm",
                "deadline:GetFleet",
                "deadline:GetLicenseEndpoint",
                "deadline:GetMonitor",
                "deadline:GetQueue",
                "deadline:GetQueueEnvironment",
                "deadline:GetQueueFleetAssociation",
```

```
                "deadline:GetStorageProfile",
                "deadline:List*",
                "detective:GetMembers",
                "detective:List*",
                "devicefarm:GetDevicePool",
                "devicefarm:GetInstanceProfile",
                "devicefarm:GetNetworkProfile",
                "devicefarm:GetProject",
                "devicefarm:GetTestGridProject",
                "devicefarm:GetVPCEConfiguration",
                "devicefarm:List*",
                "devops-guru:Describe*",
                "devops-guru:GetResourceCollection",
                "devops-guru:List*",
                "dms:Describe*",
                "dms:List*",
                "ds:Describe*",
                "dynamodb:Describe*",
                "dynamodb:GetResourcePolicy",
                "dynamodb:List*",
                "ec2:Describe*",
                "ec2:GetAssociatedEnclaveCertificateIamRoles",
                "ec2:GetIpamPoolAllocations",
                "ec2:GetIpamPoolCidrs",
                "ec2:GetManagedPrefixListEntries",
                "ec2:GetNetworkInsightsAccessScopeContent",
                "ec2:GetSnapshotBlockPublicAccessState",
                "ec2:GetTransitGatewayMulticastDomainAssociations",
                "ec2:GetTransitGatewayRouteTableAssociations",
                "ec2:GetTransitGatewayRouteTablePropagations",
                "ec2:GetVerifiedAccessEndpointPolicy",
                "ec2:GetVerifiedAccessGroupPolicy",
                "ec2:GetVerifiedAccessInstanceWebAcl",
                "ec2:SearchLocalGatewayRoutes",
                "ec2:SearchTransitGatewayRoutes",
                "ecr:Describe*",
                "ecr:GetLifecyclePolicy",
                "ecr:GetRegistryPolicy",
                "ecr:GetRepositoryPolicy",
                "ecr:List*",
                "ecs:Describe*",
                "ecs:List*",
                "eks:AccessKubernetesApi",
                "eks:Describe*",
```

```
                "eks:List*",
                "elasticache:Describe*",
                "elasticache:List*",
                "elasticbeanstalk:Describe*",
                "elasticbeanstalk:List*",
                "elasticfilesystem:Describe*",
                "elasticloadbalancing:GetResourcePolicy",
                "elasticloadbalancing:GetTrustStoreCaCertificatesBundle",
                "elasticloadbalancing:GetTrustStoreRevocationContent",
                "elasticloadbalancing:Describe*",
                "elasticmapreduce:Describe*",
                "elasticmapreduce:List*",
                "emr-containers:Describe*",
                "emr-containers:List*",
                "emr-serverless:GetApplication",
                "emr-serverless:List*",
                "es:Describe*",
                "es:List*",
                "events:Describe*",
                "events:List*",
                "evidently:GetExperiment",
                "evidently:GetFeature",
                "evidently:GetLaunch",
                "evidently:GetProject",
                "evidently:GetSegment",
                "evidently:List*",
                "firehose:Describe*",
                "firehose:List*",
                "fis:GetExperimentTemplate",
                "fis:GetTargetAccountConfiguration",
                "fis:List*",
                "fms:GetNotificationChannel",
                "fms:GetPolicy",
                "fms:List*",
                "forecast:Describe*",
                "forecast:List*",
                "frauddetector:BatchGetVariable",
                "frauddetector:Describe*",
                "frauddetector:GetDetectors",
                "frauddetector:GetDetectorVersion",
                "frauddetector:GetEntityTypes",
                "frauddetector:GetEventTypes",
                "frauddetector:GetExternalModels",
                "frauddetector:GetLabels",
```

```
                "frauddetector:GetListElements",
                "frauddetector:GetListsMetadata",
                "frauddetector:GetModelVersion",
                "frauddetector:GetOutcomes",
                "frauddetector:GetRules",
                "frauddetector:GetVariables",
                "frauddetector:List*",
                "fsx:Describe*",
                "gamelift:Describe*",
                "gamelift:List*",
                "globalaccelerator:Describe*",
                "globalaccelerator:List*",
                "glue:GetDatabase",
                "glue:GetDatabases",
                "glue:GetJob",
                "glue:GetRegistry",
                "glue:GetSchema",
                "glue:GetSchemaVersion",
                "glue:GetTable",
                "glue:GetTags",
                "glue:GetTrigger",
                "glue:List*",
                "glue:querySchemaVersionMetadata",
                "grafana:Describe*",
                "grafana:List*",
                "greengrass:Describe*",
                "greengrass:GetDeployment",
                "greengrass:List*",
                "groundstation:GetConfig",
                "groundstation:GetDataflowEndpointGroup",
                "groundstation:GetMissionProfile",
                "groundstation:List*",
                "guardduty:GetDetector",
                "guardduty:GetFilter",
                "guardduty:GetIPSet",
                "guardduty:GetMalwareProtectionPlan",
                "guardduty:GetMasterAccount",
                "guardduty:GetMembers",
                "guardduty:GetThreatIntelSet",
                "guardduty:List*",
                "health:DescribeEvents",
                "health:DescribeEventDetails",
                "healthlake:Describe*",
                "healthlake:List*",
```

```
                "iam:GetGroup",
                "iam:GetGroupPolicy",
                "iam:GetInstanceProfile",
                "iam:GetLoginProfile",
                "iam:GetOpenIDConnectProvider",
                "iam:GetPolicy",
                "iam:GetPolicyVersion",
                "iam:GetRole",
                "iam:GetRolePolicy",
                "iam:GetSAMLProvider",
                "iam:GetServerCertificate",
                "iam:GetServiceLinkedRoleDeletionStatus",
                "iam:GetUser",
                "iam:GetUserPolicy",
                "iam:ListAttachedRolePolicies",
                "iam:ListOpenIDConnectProviders",
                "iam:ListRolePolicies",
                "iam:ListRoles",
                "iam:ListServerCertificates",
                "iam:ListVirtualMFADevices",
                "identitystore:DescribeGroup",
                "identitystore:DescribeGroupMembership",
                "identitystore:ListGroupMemberships",
                "identitystore:ListGroups",
                "imagebuilder:GetComponent",
                "imagebuilder:GetContainerRecipe",
                "imagebuilder:GetDistributionConfiguration",
                "imagebuilder:GetImage",
                "imagebuilder:GetImagePipeline",
                "imagebuilder:GetImageRecipe",
                "imagebuilder:GetInfrastructureConfiguration",
                "imagebuilder:GetLifecyclePolicy",
                "imagebuilder:GetWorkflow",
                "imagebuilder:List*",
                "inspector2:List*",
                "inspector:Describe*",
                "inspector:List*",
                "internetmonitor:GetMonitor",
                "internetmonitor:List*",
                "iot:Describe*",
                "iot:GetPackage",
                "iot:GetPackageVersion",
                "iot:GetPolicy",
                "iot:GetThingShadow",
```

```
                "iot:GetTopicRule",
                "iot:GetTopicRuleDestination",
                "iot:GetV2LoggingOptions",
                "iot:List*",
                "iotanalytics:Describe*",
                "iotanalytics:List*",
                "iotevents:Describe*",
                "iotevents:List*",
                "iotsitewise:Describe*",
                "iotsitewise:List*",
                "iotwireless:GetDestination",
                "iotwireless:GetDeviceProfile",
                "iotwireless:GetFuotaTask",
                "iotwireless:GetMulticastGroup",
                "iotwireless:GetNetworkAnalyzerConfiguration",
                "iotwireless:GetServiceProfile",
                "iotwireless:GetWirelessDevice",
                "iotwireless:GetWirelessGateway",
                "iotwireless:GetWirelessGatewayTaskDefinition",
                "iotwireless:List*",
                "ivs:GetChannel",
                "ivs:GetEncoderConfiguration",
                "ivs:GetPlaybackRestrictionPolicy",
                "ivs:GetRecordingConfiguration",
                "ivs:GetStage",
                "ivs:List*",
                "ivschat:GetLoggingConfiguration",
                "ivschat:GetRoom",
                "ivschat:List*",
                "kafka:Describe*",
                "kafka:GetClusterPolicy",
                "kafka:List*",
                "kafkaconnect:Describe*",
                "kafkaconnect:List*",
                "kendra:Describe*",
                "kendra:List*",
                "kinesis:Describe*",
                "kinesis:GetResourcePolicy",
                "kinesis:List*",
                "kinesisanalytics:Describe*",
                "kinesisanalytics:List*",
                "kinesisvideo:Describe*",
                "kms:DescribeKey",
                "kms:ListResourceTags",
```

```
                "kms:ListKeys",
                "kms:GetKeyPolicy",
                "kms:GetKeyRotationStatus",
                "kms:ListAliases",
                "kms:ListKeyRotations",
                "lakeformation:Describe*",
                "lakeformation:GetLFTag",
                "lakeformation:GetResourceLFTags",
                "lakeformation:List*",
                "lambda:GetAlias",
                "lambda:GetCodeSigningConfig",
                "lambda:GetEventSourceMapping",
                "lambda:GetFunctionCodeSigningConfig",
                "lambda:GetFunctionConfiguration",
                "lambda:GetFunctionEventInvokeConfig",
                "lambda:GetFunctionRecursionConfig",
                "lambda:GetFunctionUrlConfig",
                "lambda:GetLayerVersion",
                "lambda:GetLayerVersionPolicy",
                "lambda:GetPolicy",
                "lambda:GetProvisionedConcurrencyConfig",
                "lambda:GetRuntimeManagementConfig",
                "lambda:List*",
                "launchwizard:GetDeployment",
                "launchwizard:List*",
                "license-manager:GetLicense",
                "license-manager:List*",
                "lightsail:GetAlarms",
                "lightsail:GetBuckets",
                "lightsail:GetCertificates",
                "lightsail:GetContainerServices",
                "lightsail:GetDisk",
                "lightsail:GetDisks",
                "lightsail:GetInstance",
                "lightsail:GetInstances",
                "lightsail:GetLoadBalancer",
                "lightsail:GetLoadBalancers",
                "lightsail:GetLoadBalancerTlsCertificates",
                "lightsail:GetStaticIp",
                "lightsail:GetStaticIps",
                "logs:Describe*",
                "logs:FilterLogEvents",
                "logs:GetDataProtectionPolicy",
                "logs:GetDelivery",
```

```
                "logs:GetDeliveryDestination",
                "logs:GetDeliveryDestinationPolicy",
                "logs:GetDeliverySource",
                "logs:GetLogAnomalyDetector",
                "logs:GetLogDelivery",
                "logs:GetLogGroupFields",
                "logs:GetQueryResults",
                "logs:List*",
                "logs:StartQuery",
                "logs:StopLiveTail",
                "logs:StopQuery",
                "logs:TestMetricFilter",
                "m2:GetApplication",
                "m2:GetEnvironment",
                "m2:List*",
                "macie2:GetAllowList",
                "macie2:GetCustomDataIdentifier",
                "macie2:GetFindingsFilter",
                "macie2:GetMacieSession",
                "macie2:List*",
                "mediaconnect:Describe*",
                "mediaconnect:List*",
                "medialive:Describe*",
                "medialive:GetCloudWatchAlarmTemplate",
                "medialive:GetCloudWatchAlarmTemplateGroup",
                "medialive:GetEventBridgeRuleTemplate",
                "medialive:GetEventBridgeRuleTemplateGroup",
                "medialive:GetSignalMap",
                "medialive:List*",
                "mediapackage-vod:Describe*",
                "mediapackage-vod:List*",
                "mediapackage:Describe*",
                "mediapackage:List*",
                "mediapackagev2:GetChannel",
                "mediapackagev2:GetChannelGroup",
                "mediapackagev2:GetChannelPolicy",
                "mediapackagev2:GetOriginEndpoint",
                "mediapackagev2:GetOriginEndpointPolicy",
                "mediapackagev2:List*",
                "memorydb:Describe*",
                "memorydb:List*",
                "mobiletargeting:GetInAppTemplate",
                "mobiletargeting:List*",
                "mq:Describe*",
```

```
                "mq:List*",
                "network-firewall:Describe*",
                "network-firewall:List*",
                "networkmanager:Describe*",
                "networkmanager:GetConnectAttachment",
                "networkmanager:GetConnectPeer",
                "networkmanager:GetCoreNetwork",
                "networkmanager:GetCoreNetworkPolicy",
                "networkmanager:GetCustomerGatewayAssociations",
                "networkmanager:GetDevices",
                "networkmanager:GetLinkAssociations",
                "networkmanager:GetLinks",
                "networkmanager:GetSites",
                "networkmanager:GetSiteToSiteVpnAttachment",
                "networkmanager:GetTransitGatewayPeering",
                "networkmanager:GetTransitGatewayRegistrations",
                "networkmanager:GetTransitGatewayRouteTableAttachment",
                "networkmanager:GetVpcAttachment",
                "networkmanager:List*",
                "oam:GetLink",
                "oam:GetSink",
                "oam:GetSinkPolicy",
                "oam:List*",
                "omics:GetAnnotationStore",
                "omics:GetReferenceStore",
                "omics:GetRunGroup",
                "omics:GetSequenceStore",
                "omics:GetVariantStore",
                "omics:GetWorkflow",
                "omics:List*",
                "organizations:Describe*",
                "organizations:List*",
                "osis:GetPipeline",
                "osis:List*",
                "payment-cryptography:GetAlias",
                "payment-cryptography:GetKey",
                "payment-cryptography:List*",
                "pca-connector-ad:GetConnector",
                "pca-connector-ad:GetDirectoryRegistration",
                "pca-connector-ad:GetServicePrincipalName",
                "pca-connector-ad:GetTemplate",
                "pca-connector-ad:GetTemplateGroupAccessControlEntry",
                "pca-connector-ad:List*",
                "pca-connector-scep:GetChallengeMetadata",
```

```
                "pca-connector-scep:GetConnector",
                "pca-connector-scep:List*",
                "personalize:Describe*",
                "personalize:List*",
                "pi:Describe*",
                "pi:Get*",
                "pi:List*",
                "pipes:Describe*",
                "pipes:List*",
                "proton:GetEnvironmentTemplate",
                "proton:GetServiceTemplate",
                "proton:List*",
                "qbusiness:GetApplication",
                "qbusiness:GetDataSource",
                "qbusiness:GetIndex",
                "qbusiness:GetPlugin",
                "qbusiness:GetRetriever",
                "qbusiness:GetWebExperience",
                "qbusiness:List*",
                "ram:GetPermission",
                "ram:GetResourceShares",
                "ram:List*",
                "rds:Describe*",
                "rds:List*",
                "redshift-serverless:GetNamespace",
                "redshift-serverless:GetWorkgroup",
                "redshift-serverless:List*",
                "redshift:Describe*",
                "refactor-spaces:GetApplication",
                "refactor-spaces:GetEnvironment",
                "refactor-spaces:GetRoute",
                "refactor-spaces:List*",
                "rekognition:Describe*",
                "rekognition:List*",
                "resiliencehub:Describe*",
                "resiliencehub:List*",
                "resource-explorer-2:GetDefaultView",
                "resource-explorer-2:GetIndex",
                "resource-explorer-2:GetView",
                "resource-explorer-2:List*",
                "resource-explorer-2:Search",
                "resource-groups:GetGroup",
                "resource-groups:GetGroupConfiguration",
                "resource-groups:GetGroupQuery",
```

```
                "resource-groups:GetTags",
                "resource-groups:List*",
                "route53-recovery-control-config:Describe*",
                "route53-recovery-control-config:List*",
                "route53-recovery-readiness:GetCell",
                "route53-recovery-readiness:GetReadinessCheck",
                "route53-recovery-readiness:GetRecoveryGroup",
                "route53-recovery-readiness:GetResourceSet",
                "route53-recovery-readiness:List*",
                "route53:GetDNSSEC",
                "route53:GetHealthCheck",
                "route53:GetHealthCheckStatus",
                "route53:GetHostedZone",
                "route53:List*",
                "route53profiles:GetProfile",
                "route53profiles:GetProfileAssociation",
                "route53profiles:GetProfileResourceAssociation",
                "route53profiles:List*",
                "route53resolver:GetFirewallDomainList",
                "route53resolver:GetFirewallRuleGroup",
                "route53resolver:GetFirewallRuleGroupAssociation",
                "route53resolver:GetOutpostResolver",
                "route53resolver:GetResolverConfig",
                "route53resolver:GetResolverQueryLogConfig",
                "route53resolver:GetResolverQueryLogConfigAssociation",
                "route53resolver:GetResolverRule",
                "route53resolver:GetResolverRuleAssociation",
                "route53resolver:List*",
                "rum:GetAppMonitor",
                "rum:List*",
                "s3-outposts:ListEndpoints",
                "s3-outposts:ListOutpostsWithS3",
                "s3:GetAccessGrant",
                "s3:GetAccessGrantsInstance",
                "s3:GetAccessGrantsLocation",
                "s3:GetAccessPoint",
                "s3:GetAccessPointConfigurationForObjectLambda",
                "s3:GetAccessPointForObjectLambda",
                "s3:GetAccessPointPolicy",
                "s3:GetAccessPointPolicyForObjectLambda",
                "s3:GetAccessPointPolicyStatusForObjectLambda",
                "s3:GetBucketAbac",
                "s3:GetBucketAcl",
                "s3:GetBucketCORS",
```

```
                "s3:GetBucketLocation",
                "s3:GetBucketLogging",
                "s3:GetBucketMetadataTableConfiguration",
                "s3:GetBucketNotification",
                "s3:GetBucketObjectLockConfiguration",
                "s3:GetBucketOwnershipControls",
                "s3:GetBucketPolicy",
                "s3:GetBucketPublicAccessBlock",
                "s3:GetBucketTagging",
                "s3:GetBucketVersioning",
                "s3:GetEncryptionConfiguration",
                "s3:GetLifecycleConfiguration",
                "s3:GetMultiRegionAccessPoint",
                "s3:GetMultiRegionAccessPointPolicy",
                "s3:GetMultiRegionAccessPointPolicyStatus",
                "s3:GetReplicationConfiguration",
                "s3:GetStorageLensConfiguration",
                "s3:GetStorageLensConfigurationTagging",
                "s3:GetStorageLensGroup",
                "s3:ListAllMyBuckets",
                "sagemaker:Describe*",
                "sagemaker:List*",
                "scheduler:GetSchedule",
                "scheduler:GetScheduleGroup",
                "scheduler:List*",
                "schemas:Describe*",
                "schemas:GetResourcePolicy",
                "schemas:List*",
                "secretsmanager:Describe*",
                "secretsmanager:GetResourcePolicy",
                "secretsmanager:List*",
                "securityhub:BatchGetAutomationRules",
                "securityhub:BatchGetSecurityControls",
                "securityhub:Describe*",
                "securityhub:GetConfigurationPolicy",
                "securityhub:GetConfigurationPolicyAssociation",
                "securityhub:GetEnabledStandards",
                "securityhub:GetFindingAggregator",
                "securityhub:GetInsights",
                "securityhub:List*",
                "securitylake:GetSubscriber",
                "securitylake:List*",
                "servicecatalog:Describe*",
                "servicecatalog:GetApplication",
```

```
                "servicecatalog:GetAttributeGroup",
                "servicecatalog:List*",
                "servicequotas:GetServiceQuota",
                "servicequotas:ListServiceQuotas",
                "ses:Describe*",
                "ses:GetAccount",
                "ses:GetAddonInstance",
                "ses:GetAddonSubscription",
                "ses:GetArchive",
                "ses:GetConfigurationSet",
                "ses:GetConfigurationSetEventDestinations",
                "ses:GetContactList",
                "ses:GetDedicatedIpPool",
                "ses:GetDedicatedIps",
                "ses:GetEmailIdentity",
                "ses:GetEmailTemplate",
                "ses:GetIngressPoint",
                "ses:GetRelay",
                "ses:GetRuleSet",
                "ses:GetTemplate",
                "ses:GetTrafficPolicy",
                "ses:List*",
                "shield:Describe*",
                "shield:List*",
                "signer:GetSigningProfile",
                "signer:List*",
                "sns:GetDataProtectionPolicy",
                "sns:GetSubscriptionAttributes",
                "sns:GetTopicAttributes",
                "sns:List*",
                "sqs:GetQueueAttributes",
                "sqs:GetQueueUrl",
                "sqs:List*",
                "ssm-contacts:GetContact",
                "ssm-contacts:GetContactChannel",
                "ssm-contacts:List*",
                "ssm-incidents:GetReplicationSet",
                "ssm-incidents:GetResponsePlan",
                "ssm-incidents:List*",
                "ssm-sap:GetApplication",
                "ssm-sap:List*",
                "ssm:Describe*",
                "ssm:GetDefaultPatchBaseline",
                "ssm:GetDocument",
```

```
                "ssm:GetParameters",
                "ssm:GetPatchBaseline",
                "ssm:GetResourcePolicies",
                "ssm:List*",
                "sso:GetInlinePolicyForPermissionSet",
                "sso:GetManagedApplicationInstance",
                "sso:GetPermissionsBoundaryForPermissionSet",
                "sso:GetSharedSsoConfiguration",
                "sso:ListAccountAssignments",
                "sso:ListApplicationAssignments",
                "sso:ListApplications",
                "sso:ListCustomerManagedPolicyReferencesInPermissionSet",
                "sso:ListInstances",
                "sso:ListManagedPoliciesInPermissionSet",
                "sso:ListTagsForResource",
                "states:GetExecutionHistory",
                "states:Describe*",
                "states:List*",
                "support:CreateCase",
                "support:DescribeCases",
                "synthetics:Describe*",
                "synthetics:GetCanary",
                "synthetics:GetCanaryRuns",
                "synthetics:GetGroup",
                "synthetics:List*",
                "tag:GetResources",
                "timestream:Describe*",
                "timestream:List*",
                "transfer:Describe*",
                "transfer:List*",
                "verifiedpermissions:GetIdentitySource",
                "verifiedpermissions:GetPolicy",
                "verifiedpermissions:GetPolicyStore",
                "verifiedpermissions:GetPolicyTemplate",
                "verifiedpermissions:GetSchema",
                "verifiedpermissions:List*",
                "vpc-lattice:GetAccessLogSubscription",
                "vpc-lattice:GetAuthPolicy",
                "vpc-lattice:GetListener",
                "vpc-lattice:GetResourcePolicy",
                "vpc-lattice:GetRule",
                "vpc-lattice:GetService",
                "vpc-lattice:GetServiceNetwork",
                "vpc-lattice:GetServiceNetworkServiceAssociation",
```

```
                "vpc-lattice:GetServiceNetworkVpcAssociation",
                "vpc-lattice:GetTargetGroup",
                "vpc-lattice:List*",
                "wafv2:GetIPSet",
                "wafv2:GetLoggingConfiguration",
                "wafv2:GetRegexPatternSet",
                "wafv2:GetRuleGroup",
                "wafv2:GetWebACL",
                "wafv2:GetWebACLForResource",
                "wafv2:List*",
                "workspaces-web:GetBrowserSettings",
                "workspaces-web:GetIdentityProvider",
                "workspaces-web:GetNetworkSettings",
                "workspaces-web:GetPortal",
                "workspaces-web:GetPortalServiceProviderMetadata",
                "workspaces-web:GetTrustStore",
                "workspaces-web:GetUserAccessLoggingSettings",
                "workspaces-web:GetUserSettings",
                "workspaces-web:List*",
                "workspaces:Describe*",
                "xray:BatchGetTraces",
                "xray:GetGroup",
                "xray:GetGroups",
                "xray:GetSamplingRules",
                "xray:GetServiceGraph",
                "xray:GetTraceSummaries",
                "xray:List*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AIOPSAPIGatewayAccess",
            "Effect": "Allow",
            "Action": [
                "apigateway:GET"
            ],
            "Resource": [
                "arn:aws:apigateway:*::/restapis",
                "arn:aws:apigateway:*::/restapis/*",
                "arn:aws:apigateway:*::/restapis/*/deployments",
                "arn:aws:apigateway:*::/restapis/*/deployments/*",
                "arn:aws:apigateway:*::/restapis/*/resources/*/methods/*/integrations",
                "arn:aws:apigateway:*::/restapis/*/resources/*/methods/*/integrations/
*",
```

```
                "arn:aws:apigateway:*::/restapis/*/stages",
                "arn:aws:apigateway:*::/restapis/*/stages/*",
                "arn:aws:apigateway:*::/apis",
                "arn:aws:apigateway:*::/apis/*",
                "arn:aws:apigateway:*::/apis/*/deployments",
                "arn:aws:apigateway:*::/apis/*/deployments/*",
                "arn:aws:apigateway:*::/apis/*/integrations",
                "arn:aws:apigateway:*::/apis/*/integrations/*",
                "arn:aws:apigateway:*::/apis/*/stages",
                "arn:aws:apigateway:*::/apis/*/stages/*",
                "arn:aws:apigateway:*::/domainnames/*"
            ]
        }
    ]
}
```

Limitando o acesso do agente em um AWS Conta

AWS DevOps O agente usa funções do IAM para descobrir e descrever AWS recursos durante investigações de incidentes e avaliações preventivas. Você pode controlar o nível de acesso que o agente tem configurando as políticas do IAM anexadas a essas funções. A topologia do aplicativo não mostra tudo ao qual o agente tem acesso — as políticas do IAM são a única maneira de realmente limitar quais APIs e recursos AWS de serviço o agente pode acessar.

Entendendo as funções do IAM para AWS DevOps Agente

AWS DevOps O agente usa funções do IAM para acessar recursos em dois tipos de contas:

- Função principal da conta — concede ao agente acesso aos recursos na AWS conta em que você criou o Espaço do Agente.
- Funções secundárias da conta — concede ao agente acesso aos recursos em AWS contas adicionais que você conecta ao Espaço do Agente.

Para qualquer tipo de conta, você pode restringir quais AWS serviços o agente pode acessar, limitar o acesso a recursos específicos dentro desses serviços e controlar em quais regiões o agente pode operar.

Entendendo as barreiras de proteção de permissão

AWS DevOps O agente aplica uma proteção de permissão a cada sessão que cria ao acessar seus AWS recursos. Essa grade de proteção funciona como um teto — ela define o conjunto máximo de permissões que o agente pode usar, independentemente das permissões que você concede na função do IAM.

Como funciona

Quando o agente assume sua função do IAM, ele passa uma política de sessão que limita as permissões efetivas para essa sessão. As permissões efetivas são a interseção de:

1. Suas políticas de função do IAM — A política gerenciada e todas as políticas embutidas que você anexar à função.
2. A barreira de permissão — Uma política de sessão aplicada pelo AWS DevOps Agente no momento de assumir a função.

Uma permissão deve estar presente nas duas camadas para entrar em vigor. Se você adicionar uma permissão à sua função que não esteja incluída na grade de proteção, o agente não poderá usá-la.

Permissões padrão

A política `AIDevOpsAgentAccessPolicy` gerenciada fornece o conjunto padrão de permissões somente para leitura que o agente usa para investigações. Essas permissões estão incluídas na grade de proteção, portanto, funcionam sem configuração adicional.

Estendendo as permissões além do padrão

AWS DevOps O agente oferece suporte a um conjunto selecionado de permissões adicionais além da política gerenciada padrão. Essas permissões estão incluídas na grade de proteção, mas não são habilitadas por padrão. Para usá-las, adicione as permissões específicas à sua função como uma política embutida.

Por exemplo, para permitir que o agente leia objetos de seus buckets do S3 durante as investigações, adicione uma política embutida à sua função:

```
{
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::my-application-bucket",
        "arn:aws:s3:::my-application-bucket/*"
      ]
    }
  ]
}
```

Por `s3:GetObject` estar `s3:ListBucket` incluída na grade de proteção, essa política em linha entra em vigor. Você pode definir o escopo `Resource` de dois compartimentos específicos para seguir o princípio do menor privilégio.

Permissões adicionais suportadas

As permissões a seguir estão incluídas na grade de proteção e podem ser ativadas adicionando-as à sua função como uma política embutida. Eles não são concedidos por padrão — você deve se inscrever explicitamente.

Serviço	Ações	Caso de uso
Amazon S3	`s3:GetObject` , `s3:ListBucket`	Leia dados, registros ou configurações do aplicativo armazenados no S3
AWS Direct Connect	`directconnect:DescribeConnections` , `directconnect:DescribeDirectConnectGatewayAssociations` , `directconnect:DescribeDirectConnectGateways` , `directconnect:DescribeLags` ,	Investigue problemas de conectividade de rede

Serviço	Ações	Caso de uso
	`directconnect:DescribeVirtualInterfaces`	

Observação: essa lista pode se expandir com o tempo, à medida que novos recursos são adicionados ao AWS DevOps Agente. As permissões não listadas aqui ou na política `AIDevOpsAgentAccessPolicy` gerenciada são bloqueadas pela grade de proteção.

Permissões bloqueadas pela grade de proteção

Se você adicionar uma permissão à sua função que não esteja na grade de proteção, o agente não poderá usá-la. Isso ocorre intencionalmente — a barreira impede que o agente execute ações fora do escopo pretendido, mesmo que a função as permitisse.

Por exemplo, operações de gravação como `s3:PutObjectec2:TerminateInstances`, ou não `dynamodb:DeleteItem` estão incluídas na grade de proteção. Mesmo que sua função conceda essas permissões, o agente não poderá realizar essas ações.

Resumo

Camada	Quem controla isso	Finalidade
Políticas de função do IAM	You	Defina o que você pretende que o agente seja capaz de fazer
Guardrail de permissão	AWS DevOps Agente	Define o máximo que o agente pode fazer
Permissões efetivas	Interseção de ambos	O que o agente pode realmente fazer

Esse modelo garante que o agente opere dentro de um limite de segurança bem definido, ao mesmo tempo em que oferece flexibilidade para ampliar seus recursos para seu caso de uso específico.

Escolhendo seus limites de recursos

Ao limitar o acesso aos recursos, você precisa incluir permissões suficientes para que o agente investigue com êxito os incidentes do aplicativo. Isso inclui:

- Todos os recursos para aplicativos dentro do escopo que o agente deve monitorar e investigar
- Toda a infraestrutura de suporte da qual esses aplicativos dependem

A infraestrutura de suporte pode incluir:

- Componentes de rede (VPCs, sub-redes, balanceadores de carga, gateways de API)
- Armazenamentos de dados (bancos de dados, caches, armazenamento de objetos)
- Recursos computacionais (instâncias EC2, funções Lambda, contêineres)
- Serviços de monitoramento e registro (CloudWatch, CloudTrail)
- Recursos de gerenciamento de identidade e acesso necessários para entender as permissões

Se você restringir o acesso de forma muito restrita, talvez o agente não consiga identificar as causas-raiz que se originam na infraestrutura de suporte fora dos limites definidos.

Restringindo o acesso ao serviço

Você pode limitar quais AWS serviços o agente pode acessar modificando as políticas do IAM anexadas às funções do agente. Ao criar políticas personalizadas, siga estas melhores práticas:

- Conceda somente permissões de leitura — O agente precisa ler as configurações, métricas e registros dos recursos durante as investigações. Evite conceder permissões que permitam ao agente modificar ou excluir recursos.
- Limite aos serviços necessários — inclua somente AWS os serviços que contêm recursos relevantes para seus aplicativos. Por exemplo, se seu aplicativo não usa o Amazon RDS, não inclua as permissões do RDS na política.
- Use ações específicas em vez de curingas — em vez de conceder `service:*` permissões, especifique ações individuais, como ou`cloudwatch:GetMetricData`. `ec2:DescribeInstances`

Exemplo de política de restrição a serviços 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": "*"
    }
  ]
}
```

Restringindo o acesso aos recursos

Para limitar o agente a recursos específicos em um serviço, use permissões em nível de recurso em suas políticas do IAM. Isso permite que você conceda acesso somente a recursos que correspondam a padrões específicos.

Usando padrões de ARN de recursos:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lambda:GetFunction",
        "lambda:GetFunctionConfiguration"
      ],
      "Resource": "arn:aws:lambda:*:*:function:production-*"
    }
  ]
```

```
}
```

Este exemplo limita o agente a acessar somente funções do Lambda com nomes que começam com “production-”.

Usando restrições baseadas em tags:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeInstanceStatus"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Environment": "production"
        }
      }
    }
  ]
}
```

Este exemplo limita o agente a acessar somente instâncias do EC2 marcadas com`Environment=production`.

Restringindo o acesso regional

Para limitar quais AWS regiões o agente pode acessar, use a chave de `aws:RequestedRegion` condição em suas políticas do 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 exemplo limita o agente a acessar recursos somente nas regiões us-east-1 e us-west-2.

Criação de políticas personalizadas do IAM

Ao criar um Agent Space ou adicionar contas secundárias, você tem a opção de criar uma função personalizada do IAM usando um modelo de política. Isso permite que você implemente o princípio do menor privilégio.

Ao criar um Espaço do Agente

No console do DevOps agente no console AWS de gerenciamento...

- Escolha Criar uma nova função de DevOps agente usando um documento de política e siga as instruções

Ao editar um Espaço do Agente

No console do DevOps agente no console AWS de gerenciamento...

- Selecione a guia Capacidades
- Selecione a conta secundária que você deseja editar na seção Nuvem e clique em Editar
- Escolha Criar uma nova política de DevOps agente usando um modelo e siga as instruções.

Práticas recomendadas de políticas personalizadas

- Conceda permissões somente de leitura — evite permissões que permitam a modificação ou exclusão de recursos
- Use permissões em nível de recurso quando possível — restrinja o acesso a recursos específicos usando padrões ou tags de ARN
- Revise e audite regularmente as permissões — revise periodicamente as políticas de IAM do agente para garantir que elas ainda estejam alinhadas aos seus requisitos de segurança

Configurando a autenticação do IAM Identity Center

A autenticação do IAM Identity Center fornece uma forma centralizada de gerenciar o acesso do usuário ao aplicativo web do AWS DevOps Agent Space. Este guia explica como configurar a autenticação do IAM Identity Center e gerenciar usuários.

Pré-requisitos

Antes de configurar a autenticação do IAM Identity Center, verifique se você tem:

- O IAM Identity Center está ativado em sua organização ou conta
- Permissões de administrador no AWS DevOps Agent
- Um espaço de agente configurado ou pronto para ser criado

Opções de autenticação

AWS DevOps O agente oferece dois métodos de autenticação para acessar o aplicativo web do Agent Space:

Autenticação do IAM Identity Center — recomendada para ambientes de produção. Fornece gerenciamento centralizado de usuários, integração com provedores de identidade externos e sessões de até 12 horas.

Acesso de administrador (autenticação do IAM) — fornece acesso rápido aos administradores durante a configuração e configuração iniciais. As sessões são limitadas a 30 minutos.

Configurando o IAM Identity Center durante a criação do Agent Space

Ao criar um Agent Space, você pode configurar a autenticação do IAM Identity Center na guia Acesso:

Etapa 1: Navegar até a configuração do aplicativo Web

1. Depois de configurar os detalhes do Agent Space e o acesso à AWS conta, vá para a guia Acesso
2. Você verá duas seções: “Connect IAM Identity Center” e “Acesso de administrador”

Etapa 2: configurar a integração do IAM Identity Center

Na seção Connect [Agent Space] ao IAM Identity Center:

1. Verifique a instância do IAM Identity Center — O console exibe qual instância do Identity Center gerenciará o acesso do usuário ao aplicativo Web (por exemplo,`ssoins-7223a9580931edbe`). Sua instância do IAM Identity Center mais próxima será automaticamente pré-preenchida.
2. Selecione a opção Nome da função do aplicativo IAM Identity Center — Escolha uma das três opções:

Crie automaticamente uma nova função de DevOps agente (recomendado):

- O sistema cria automaticamente uma nova função de serviço com as permissões apropriadas.
- Essa é a opção mais simples e funciona para a maioria dos casos de uso.

Atribua uma função existente:

- Use uma função do IAM existente que você já criou
- O sistema verificará se a função tem as permissões necessárias
- Escolha essa opção se sua organização tiver funções pré-criadas para AWS DevOps Agente

Crie uma nova função de DevOps agente usando um modelo de política:

- Use os detalhes da política fornecidos para criar seu próprio papel personalizado no console do IAM
- Escolha essa opção se precisar personalizar as permissões da função.

Depois de clicar em Connect, o sistema automaticamente:

- Cria ou configura a função do IAM especificada
- Configura um aplicativo IAM Identity Center para seu Agent Space
- Estabelece relações de confiança entre o IAM Identity Center e o aplicativo web Agent Space
- Configura fluxos de autenticação OAuth 2.0 para acesso seguro do usuário

Alternativa: usar o acesso de administrador

Se você quiser acessar o aplicativo web Agent Space imediatamente sem configurar o IAM Identity Center:

1. Na seção Acesso de administrador, observe o ARN da função do IAM que fornece acesso ao administrador (por exemplo,) `arn:aws:iam::440491339484:role/service-role/DevOpsAgentRole-WebappAdmin-15ppoc42`
2. Clique no botão azul de acesso do administrador para iniciar o aplicativo web Agent Space com autenticação IAM
3. As sessões que usam esse método são limitadas a 30 minutos.

> **Note**
>
> O acesso de administrador é destinado à instalação e configuração iniciais. Para uso em produção e operações contínuas, configure a autenticação do IAM Identity Center.

Adicione usuários e grupos.

Depois de configurar a autenticação do IAM Identity Center, você precisa conceder acesso ao aplicativo web Agent Space a usuários e grupos específicos:

Etapa 1: acessar o gerenciamento de usuários

1. No console do AWS DevOps agente, selecione seu Espaço do agente
2. Vá para a guia Acesso
3. Em Acesso do usuário, clique em Gerenciar usuários e grupos

Etapa 2: adicionar usuários ou grupos

1. Escolha Adicionar usuários ou grupos
2. Pesquise usuários ou grupos no diretório do IAM Identity Center
3. Marque as caixas de seleção ao lado dos usuários ou grupos que você deseja adicionar.
4. Clique em Adicionar para conceder acesso a eles

Agora, os usuários selecionados podem acessar o aplicativo web Agent Space usando suas credenciais do IAM Identity Center.

Trabalhando com provedores de identidade externos

Se você estiver usando um provedor de identidade externo (como Okta, Microsoft Entra ID ou Ping Identity) com o IAM Identity Center:

- Usuários e grupos são sincronizados do seu provedor de identidade externo com o IAM Identity Center
- Ao adicionar usuários e grupos ao aplicativo web do Agent Space, você está selecionando no diretório sincronizado
- Os atributos do usuário e as associações de grupos são mantidos pelo seu provedor de identidade externo.
- As alterações no seu provedor de identidade são refletidas automaticamente no IAM Identity Center após a sincronização

Como os usuários acessam o aplicativo web Agent Space

Depois de adicionar usuários ao seu Espaço do Agente:

1. Compartilhe o URL do aplicativo web Agent Space com usuários autorizados
2. Quando os usuários navegam até o URL, eles são redirecionados para a página de login do IAM Identity Center
3. Depois de inserir suas credenciais (e concluir o MFA, se configurado), eles são redirecionados de volta ao aplicativo web Agent Space
4. Sua sessão é válida por 8 horas por padrão (configurável pelo administrador do Identity Center)

Gerenciar acesso do usuário

Você pode atualizar o acesso do usuário a qualquer momento:

Adicionar mais usuários ou grupos:

- Siga as mesmas etapas descritas acima para adicionar mais usuários ou grupos.

Removendo o acesso:

1. Na seção Acesso do usuário, encontre o usuário ou grupo a ser removido
2. Clique no botão Remover ao lado do nome da pessoa
3. Confirme a remoção

Os usuários removidos perderão o acesso imediatamente, mas as sessões ativas podem continuar até expirarem.

Gerenciamento de sessões

As sessões do IAM Identity Center para o aplicativo web Agent Space têm as seguintes características:

- Duração padrão da sessão — 8 horas
- Segurança da sessão — cookies somente HTTP para proteção aprimorada
- Autenticação multifatorial — compatível quando configurada no IAM Identity Center
- Credenciais de API — Credenciais SigV4 de curta duração (15 minutos) são emitidas para chamadas de API e renovadas automaticamente

Para configurar a duração da sessão:

1. Navegue até o console do IAM Identity Center
2. Vá para Configurações > Autenticação
3. Em Duração da sessão, configure sua duração preferida (de 1 hora a 12 horas)
4. Selecione Save changes (Salvar alterações)

Desconectando o Identity Center

1. No console do seu Agent Space, clique em Ações no canto superior direito e selecione Desconectar do IAM Identity Center
2. Confirme na caixa de diálogo de confirmação

Configurando a autenticação do provedor de identidade externo (IdP)

A autenticação de provedor de identidade externo (IdP) permite que sua organização use um provedor de identidade existente compatível com OIDC, como Okta ou Microsoft Entra ID, para gerenciar o acesso do usuário ao aplicativo web do Agent Space. AWS DevOps Os usuários fazem login com suas credenciais corporativas diretamente por meio do seu IdP, sem AWS precisar do IAM Identity Center.

Pré-requisitos

Antes de configurar a autenticação de IdP externo, verifique se você tem:

- Um provedor de identidade compatível com OIDC (Okta ou Microsoft Entra ID)
- Acesso de administrador ao seu provedor de identidade
- Permissões de administrador para acessar o console do AWS DevOps agente
- Um espaço de agente configurado ou pronto para ser criado

Como funciona

Quando você configura a autenticação de IdP externo:

- Os usuários navegam até o URL do aplicativo web Agent Space
- Eles são redirecionados para a página de login do seu provedor de identidade
- Depois de se autenticarem com suas credenciais corporativas, eles são redirecionados de volta para o aplicativo web
- O aplicativo web troca o token de autenticação por AWS credenciais de curta duração com escopo no Espaço do Agente

As sessões são válidas por até 8 horas. As credenciais são atualizadas automaticamente usando tokens de atualização do OIDC sem exigir que os usuários se autentiquem novamente.

Configurando a autenticação de IdP externo

Etapa 1: registrar um aplicativo em seu provedor de identidade

Escolha seu provedor de identidade e siga as instruções de configuração correspondentes.

Opção A: Okta

1. No Okta Admin Console, navegue até Aplicativos > Aplicativos e escolha Criar integração de aplicativos
2. Selecione OIDC - OpenID Connect como método de login e Aplicativo Web como tipo de aplicativo. Escolha Próximo.
3. Defina um nome descritivo para o aplicativo (por exemplo,`AWS DevOps Agent`)
4. Em Tipo de concessão, verifique se o seguinte está marcado:
   - Código de autorização (padrão)
   - Token de atualização — Isso é necessário para a atualização da sessão. Se não estiver ativado, os usuários não conseguirão manter as sessões.

> **Note**
>
> O Okta não habilita o tipo de concessão Refresh Token por padrão. Você deve habilitá-lo explicitamente.

1. Deixe o redirecionamento de login URIs como o valor padrão por enquanto — você o atualizará após configurar o Espaço do Agente
2. Em Tarefas, atribua os usuários ou grupos que devem ter acesso
3. Escolha Salvar
4. Na guia Geral do aplicativo, observe os seguintes valores:
   - ID do cliente
   - Segredo do cliente — Escolha Copiar para salvar esse valor com segurança
5. Anote seu domínio Okta — este é o URL do emissor (por exemplo,`https://dev-12345678.okta.com`).

> **Note**
>
> Na guia Login, verifique se o emissor está definido como URL Okta (não dinâmico). Isso garante um URL de emissor estável.

> **Note**
>
> Não adicione uma reivindicação de grupos ao token de ID na guia Reivindicações do seu servidor de autorização. AWS DevOps O agente não usa a associação ao grupo do seu IdP.

Opção B: Microsoft Inserir ID

1. No portal do Azure, navegue até Microsoft Entra ID > Registros de aplicativos > Novo registro
2. Defina um nome descritivo (por exemplo,`AWS DevOps Agent`)
3. Em Tipos de conta compatíveis, selecione a opção apropriada para sua organização (normalmente Contas somente neste diretório organizacional)
4. Deixe o URI de redirecionamento em branco por enquanto. Escolha Registrar
5. Na página Visão geral do aplicativo, observe os seguintes valores:
   - ID do aplicativo (cliente) — usado como ID do cliente ao configurar o Agent Space
   - ID do diretório (inquilino) — usado para construir o URL do emissor
6. Navegue até Certificados e segredos > Novo segredo do cliente
   - Defina uma descrição e um período de expiração
   - Escolha Adicionar e copie o valor secreto imediatamente — ele não será exibido novamente
7. O URL do emissor do Entra ID segue esse formato. `{tenant-id}`Substitua pelo ID do diretório (inquilino) da etapa 5:
   - `https://login.microsoftonline.com/{tenant-id}/v2.0`

> **Note**
>
> Não habilite a reivindicação opcional do grupo na configuração do Token. AWS DevOps O agente não usa a associação ao grupo do seu IdP.

Etapa 2: habilitar o aplicativo do operador com autenticação IdP

1. No console do AWS DevOps agente, selecione seu Espaço do agente
2. Vá para a guia Acesso
3. Em Acesso do usuário, escolha Provedor de identidade externo
4. No formulário de configuração, configure o seguinte:
   - Provedor de identidade — Selecione seu provedor de identidade (Okta ou Microsoft Entra ID)
   - URL do emissor — O URL do emissor do OIDC do seu provedor de identidade
   - ID do cliente — O ID do cliente do aplicativo OIDC que você criou
   - Segredo do cliente — O segredo do cliente do seu aplicativo OIDC
5. Em Nome da função do aplicativo do provedor de identidade, escolha uma das três opções:
   - Criar automaticamente uma nova função de DevOps agente (recomendado) — Cria uma nova função de serviço com as permissões apropriadas
   - Atribuir uma função existente — Use uma função do IAM existente que você já criou
   - Crie uma nova função de DevOps agente usando um modelo de política — Use os detalhes fornecidos para criar sua própria função no console do IAM
6. Revise o alerta de aviso de URL de retorno de chamada exibido na parte inferior do formulário. Copie esse URL — você precisará adicioná-lo ao redirecionamento permitido do seu provedor de identidade URIs antes que os usuários possam fazer login.
7. Selecione Connect (Conectar).

Depois de escolher Connect, o console exibe a configuração do provedor de identidade externo com os seguintes detalhes:

- Provedor — O provedor de identidade que você selecionou
- URL do emissor — O URL configurado do emissor do OIDC
- ID do cliente — O ID do cliente configurado
- ARN da função do IAM — A função do IAM usada para acesso do usuário
- URL de retorno de chamada — Configure essa URL em seu provedor de identidade como uma URI de redirecionamento permitida
- URL de login — Use esse URL para acessar o aplicativo web por meio do seu provedor de identidade

Etapa 3: adicionar o URL de retorno de chamada ao seu provedor de identidade

Okta

1. No Okta Admin Console, navegue até a guia Geral do seu aplicativo
2. Em Login, escolha Editar
3. Adicione o URL de retorno de chamada como um URI de redirecionamento de login:
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/callback`
4. (Opcional) Defina o URI de início de login para ativar o login iniciado pelo IdP no painel do Okta:
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/login`
5. (Recomendado) Adicione um URI de redirecionamento de desconexão para redirecionar os usuários de volta ao aplicativo web após o logout. Sem isso, os usuários podem ver uma página de erro ao sair:
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome`
6. Escolha Salvar

Microsoft Entra ID

1. No portal do Azure, navegue até a página de Autenticação do seu aplicativo
2. Em Configurações da plataforma, escolha Adicionar uma plataforma > Web
3. Insira o URL de retorno de chamada como o URI de redirecionamento:
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/callback`
4. (Opcional) Adicione um URI de redirecionamento de desconexão para redirecionar os usuários de volta ao aplicativo web após o logout:
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome`
5. Escolha Configurar

Etapa 4: Verificar a configuração

1. Navegue até o URL de login mostrado no console:
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/login`
2. Você deve ser redirecionado para a página de login do seu provedor de identidade
3. Faça login com suas credenciais corporativas

4. Após a autenticação bem-sucedida, você será redirecionado de volta para o aplicativo web do Agent Space

Atualizando a configuração do IdP

Você pode alternar o segredo do cliente sem se desconectar:

1. No console do AWS DevOps agente, selecione seu Espaço do agente
2. Vá para a guia Acesso
3. Em Configuração do provedor de identidade externo, escolha Rotate client secret
4. Insira o novo segredo do cliente
5. Escolha Salvar

Para alterar qualquer outro campo de configuração do IdP (como URL do emissor, ID do cliente ou provedor de identidade), você deve desconectar o IdP existente e configurar um novo.

Como os usuários acessam o aplicativo web Agent Space

Depois de configurar a autenticação de IdP externo:

- Compartilhe o URL do aplicativo web Agent Space com usuários autorizados
- Quando os usuários navegam até o URL, eles são redirecionados para a página de login do seu provedor de identidade
- Depois de inserir suas credenciais (e concluir o MFA, se configurado pelo seu IdP), eles são redirecionados de volta ao aplicativo web Agent Space
- As sessões são atualizadas automaticamente — consulte Gerenciamento de sessões para obter detalhes

Gerenciamento de sessões

As sessões de IdP externo para o aplicativo web Agent Space têm as seguintes características:

- Duração da sessão — As sessões do navegador duram até 8 horas. Isso não é configurável no AWS DevOps Agente. Se a duração da sessão do seu IdP exceder 8 horas, os usuários poderão ser reautenticados automaticamente na próxima visita sem inserir credenciais. Configure a vida útil da sessão e do token do seu IdP de acordo com os requisitos de segurança da sua organização.

- Atualização de credenciais — As sessões são atualizadas automaticamente usando tokens de atualização do OIDC sem exigir que os usuários se autentiquem novamente
- Autenticação multifatorial — Compatível quando configurada em seu provedor de identidade. O IdP manipula o MFA durante o login — nenhuma configuração adicional é necessária no Agente AWS DevOps

Comportamento de logout

Quando um usuário clica em Sair no aplicativo web:

1. Todos os cookies de sessão são apagados imediatamente
2. O usuário é redirecionado para o endpoint de logout do OIDC do provedor de identidade para encerrar a sessão de SSO
3. Se um URI de redirecionamento de saída estiver configurado, o usuário será redirecionado de volta para a página de boas-vindas do aplicativo web

Revogando o acesso do usuário

Para revogar imediatamente o acesso de um usuário, você pode revogar suas sessões diretamente no portal administrativo do seu provedor de identidade:

- Okta — No Okta Admin Console, navegue até Diretório > Pessoas, selecione o usuário, escolha Mais ações > Limpar sessões de usuário
- Microsoft Entra ID — No portal do Azure, navegue até Usuários, selecione o usuário e escolha Revogar sessões

Considerações sobre segurança

Armazenamento secreto do cliente — O segredo do cliente que você fornece durante a configuração é criptografado usando sua chave KMS gerenciada pelo cliente, caso você tenha fornecido uma ao criar o Espaço do Agente, ou uma chave de propriedade do serviço. Ele nunca é retornado nas respostas da API ou exibido no console após a configuração inicial.

Rotação de segredos do cliente — Os segredos do cliente Entra têm uma expiração configurável. Defina um lembrete para alternar o segredo antes que ele expire usando a opção Girar segredo do cliente no console do agente. AWS DevOps Se o segredo expirar, os usuários não conseguirão fazer login até que ele seja rotacionado.

Gerenciamento da vida útil do token — A vida útil dos tokens (tokens de acesso, tokens de atualização) emitidos pelo seu provedor de identidade é controlada pela configuração do seu IdP. Recomendamos configurar a vida útil apropriada do token em seu IdP:

- Okta — Configure a vida útil do token em Segurança > API > Servidores de autorização > Políticas de acesso
- Microsoft Entra ID — Configure a vida útil do token usando políticas de vida útil do token

Declaração de grupos — Não habilite a declaração de grupos na configuração de token do seu provedor de identidade. AWS DevOps Atualmente, o agente não usa a associação ao grupo do seu IdP.

Identificador de usuário — O AWS DevOps agente usa uma declaração específica do provedor para identificar usuários de forma exclusiva:

- Okta — Usa a `sub` declaração do token de ID
- Microsoft Entra ID — usa a declaração `oid` (identificador de objeto) do token de ID

Esses identificadores são imutáveis e aparecem nos CloudTrail registros para fins de auditoria.

Desconectando o IdP externo

1. No console do AWS DevOps agente, selecione seu Espaço do agente
2. Vá para a guia Acesso
3. Em Acesso do usuário, escolha Desconectar
4. Analise os impactos listados na caixa de diálogo de confirmação e confirme

A desconexão irá:

- Remova a configuração do IdP do Espaço do Agente
- Impedir que os usuários façam login por meio do provedor de identidade externo
- Remova o histórico individual de bate-papo e artefatos associados às contas de usuário do IdP

As sessões ativas do usuário continuarão até expirarem ou a próxima atualização de credencial falhar.

Solução de problemas

- Falha no redirecionamento para o IdP — Verifique se o URL do emissor corresponde ao endpoint de descoberta do OIDC do seu IdP. Para Okta, certifique-se de que o Emissor esteja configurado como URL Okta (não Dinâmico) na guia Login. Para Entra, use o formato`https://login.microsoftonline.com/{tenant-id}/v2.0`.
- Acesso negado ou erro de política (Okta) — Verifique se o usuário ou seu grupo está atribuído ao aplicativo em Atribuições. Marque Sign On > Regras da política de login.
- Erro de configuração do IdP após o login — Seu provedor de identidade não retornou um token de atualização. Certifique-se de que o `offline_access` escopo e o tipo de concessão do token de atualização estejam habilitados:
  - Okta — Vá para a guia Geral do seu aplicativo e ative a caixa de seleção Atualizar Token em Tipo de concessão
  - Entrar — Acesse as permissões da API e verifique `offline_access` se está listado em permissões delegadas
- A autenticação é bem-sucedida, mas o aplicativo web mostra um erro — Verifique se o URI de redirecionamento em seu IdP corresponde exatamente ao URL de retorno de chamada mostrado no console do agente. AWS DevOps
- Falhas de autenticação — Se a declaração opcional do grupo estiver habilitada em seu IdP, desative-a. AWS DevOps O agente não usa reivindicações de grupo.
- O login falha após a autenticação do IdP — Para Entra, verifique se não **`requestedAccessTokenVersion`** está definido **`null`** no Manifesto do aplicativo. Para Okta, verifique se o URL do emissor está correto.
- Página de erro após clicar em Sair (Okta) — Se você ver um **`post_logout_redirect_uri`** erro após sair, adicione **`https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome`** como URI de redirecionamento de saída na guia Geral do seu aplicativo Okta.
- Os usuários permanecem na página do provedor de identidade após o logout (Entra) — Para redirecionar os usuários de volta ao aplicativo web após o logout, adicione **`https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome`** como URI de redirecionamento na página de autenticação do seu aplicativo Entra.

Criptografia em repouso para AWS DevOps Agent

AWS DevOps O agente criptografa todos os dados do cliente em repouso. Por padrão, o AWS DevOps Agente usa chaves AWS próprias para criptografar automaticamente seus dados sem custo adicional. Você não pode visualizar, gerenciar ou auditar o uso de chaves AWS próprias. No entanto, você não precisa realizar nenhuma ação para proteger essas chaves. Seus dados são protegidos automaticamente.

Você pode optar por criptografar seus dados usando uma chave simétrica gerenciada pelo cliente que você cria, possui e gerencia no AWS Key Management Service (AWS KMS). Como você tem controle total dessa camada de criptografia, você pode executar tarefas como as seguintes:

- Estabelecer e manter as políticas de chave
- Habilitar e desabilitar políticas de chaves
- Alternar os materiais de criptografia de chave
- Adicionar etiquetas
- Criar réplicas de chaves
- Chaves de agendamento para exclusão

Para obter mais informações, consulte Chaves gerenciadas pelo cliente no Guia do desenvolvedor do AWS Key Management Service.

> **Note**
>
> AWS DevOps O agente ativa automaticamente a criptografia em repouso usando chaves AWS próprias para proteger os dados do cliente sem nenhum custo. As cobranças padrão do AWS KMS se aplicam quando você usa uma chave gerenciada pelo cliente. Para obter mais informações sobre preços, consulte Preços do AWS Key Management Service.

Chaves gerenciadas pelo cliente

As chaves gerenciadas pelo cliente são chaves KMS em sua AWS conta que você cria, possui e gerencia. Você tem controle total sobre essas chaves do KMS, incluindo o estabelecimento e a manutenção de suas principais políticas.

Quando você configura uma chave gerenciada pelo cliente, o AWS DevOps Agente a usa para proteger dados confidenciais de recursos. AWS DevOps O agente usa criptografia de envelope com o chaveiro hierárquico do AWS Encryption SDK. Sua chave KMS é usada para gerar chaves de ramificação, que por sua vez protegem seus dados.

Você pode especificar uma chave gerenciada pelo cliente ao criar os seguintes recursos:

- Espaço do Agente — Criptografa os detalhes e o conteúdo do Agent Space criados a partir do DevOps Agent Web App relacionados a investigações, habilidades e bate-papo.
- Serviço — Criptografa as credenciais de serviços de terceiros em repouso.

Para configurar uma chave gerenciada pelo cliente no AWS DevOps Agent, siga estas etapas.

Etapa 1: criar uma chave gerenciada pelo cliente

Você pode criar uma chave simétrica gerenciada pelo cliente usando o console do AWS KMS ou a API do AWS KMS. A chave deve atender aos seguintes requisitos:

Propriedade	Requisito
Tipo de chave	Simétrica
Especificação da chave	`SYMMETRIC_DEFAULT`
Uso da chave	`ENCRYPT_DECRYPT`

> **Note**
>
> AWS DevOps O agente só oferece suporte a chaves KMS de criptografia simétrica com a especificação da `SYMMETRIC_DEFAULT` chave e o uso da `ENCRYPT_DECRYPT` chave. Atualmente, não há suporte para chaves multirregionais e chaves assimétricas.

Para obter mais informações, consulte Criação de uma chave simétrica gerenciada pelo cliente no Guia do desenvolvedor do AWS Key Management Service.

Etapa 2: definir a política de chaves

As políticas de chaves controlam o acesso à chave gerenciada pelo cliente. Cada chave gerenciada pelo cliente deve ter exatamente uma política de chave, que contém declarações que determinam quem pode usar a chave e como pode usá-la.

Sua política de chaves deve conceder permissões tanto para o responsável pela chamada (sua identidade do IAM) quanto para o serviço do AWS DevOps agente. AWS DevOps O agente acessa sua chave usando dois conjuntos de credenciais:

1. Suas credenciais de chamador — usadas para todas as operações síncronas, incluindo validação de chave, criptografia no momento da criação do recurso e qualquer chamada de API que retorne uma resposta direta ao chamador.
2. AWS DevOps Agent Service Principal — Usado para operações assíncronas executadas em segundo plano, como investigações operacionais, análise de incidentes, correlação de eventos e geração de análise de causa raiz.

A tabela a seguir lista as ações necessárias do KMS:

Ação KMS	Description
`kms:DescribeKey`	Valide a configuração da chave no momento da criação do recurso
`kms:GenerateDataKey`	Gere chaves de criptografia de dados para criptografia de envelope
`kms:Decrypt`	Descriptografar dados
`kms:Encrypt`	Criptografar dados
`kms:ReEncrypt`	Criptografe novamente os dados com a mesma chave ou com uma chave diferente

AWS DevOps O agente valida todas essas permissões no momento da configuração usando operações de execução a seco. Se alguma permissão estiver ausente, a solicitação falhará com uma exceção.

Veja a seguir um exemplo de política de chave. Substitua os valores do espaço reservado pelos seus.

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

A política contém as seguintes declarações:

- AllowKeyAdministration— Concede à raiz da conta acesso administrativo total à chave. `111122223333`Substitua pelo ID AWS da sua conta.
- AllowCallerAccessViaService— Concede aos seus diretores do IAM as permissões KMS necessárias para todas as operações AWS DevOps síncronas do agente. Isso inclui a validação da chave no momento da criação do recurso, bem como operações de criptografia e descriptografia para qualquer chamada de API que retorne uma resposta direta ao chamador. A `kms:ViaService` condição garante que você possa usar a chave somente por meio do serviço de AWS DevOps agente. `111122223333`Substitua pelo ID AWS da sua conta e `us-east-1` pela sua AWS região.
- AllowDevOpsAgentServiceAccessForAgentSpace/AllowDevOpsAgentServiceAccessForService— Concede ao responsável pelo `aidevops.amazonaws.com` serviço as permissões KMS necessárias para operações assíncronas. AWS DevOps O agente usa esse princípio de serviço para criptografar e descriptografar seus dados ao realizar operações em segundo plano, como investigações operacionais, analisar incidentes, correlacionar eventos entre serviços e gerar análises de causa raiz. Sem esse acesso, o AWS DevOps Agente não pode ler os dados criptografados necessários para realizar investigações em seu nome. A `aws:SourceArn` condição restringe o acesso às solicitações provenientes dos recursos do seu AWS DevOps Agente e garante que o `kms:EncryptionContext` contexto de criptografia corresponda ao seu recurso. ARNs `111122223333`Substitua pelo ID AWS da sua conta e `us-east-1` pela sua AWS região.

Para obter mais informações sobre políticas principais, consulte Políticas de chaves no AWS KMS no Guia do desenvolvedor do AWS Key Management Service.

Etapa 3: especificar a chave ao criar um recurso

Depois de criar sua chave e configurar a política de chaves, você pode especificar a chave ao criar recursos do AWS DevOps Agente.

Console

Para configurar uma chave gerenciada pelo cliente ao criar um Espaço do Agente no console:

1. Abra o console do AWS DevOps agente.
2. Escolha Criar espaço de agente ou Registrar serviço.
3. Insira os detalhes do espaço do agente (nome, descrição e função do IAM).

4. Expanda a seção Configuração avançada.
5. Em Tipo de chave de criptografia, selecione Chave gerenciada pelo cliente.
6. Escolha uma chave KMS na lista suspensa ou insira um ARN da chave KMS.
7. Revise a política de chaves exibida na seção Política de chaves expansível. Certifique-se de ter anexado essa política à sua chave KMS. Você pode usar o botão copiar para copiar a política.
8. Conclua a configuração restante e escolha Criar.

> **Note**
>
> Se você não vê sua chave KMS na lista suspensa, verifique se a chave atende aos requisitos da Etapa 1 e se você tem `kms:ListKeys` permissões. `kms:DescribeKey`

solicitações de

Criação de um espaço de agente com uma chave gerenciada pelo cliente

Especifique o `kmsKeyArn` parâmetro ao criar um espaço de agente. O valor deve ser o ARN completo da chave KMS.

```
{
  "name": "my-agent-space",
  "description": "An encrypted agent space",
  "kmsKeyArn": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

Registrando um serviço com uma chave gerenciada pelo cliente

Especifique o `kmsKeyArn` parâmetro ao registrar um serviço. O valor deve ser o ARN completo da chave KMS. Esse parâmetro é suportado em todos os tipos de serviço, incluindo servidores Dynatrace ServiceNow,, PagerDuty, GitLab GitHub, e MCP.

```
{
  "service": "dynatrace",
  "kmsKeyArn": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "serviceDetails": { ... }
}
```

> **Note**
>
> Você deve especificar a chave gerenciada pelo cliente no momento da criação do recurso. Você não pode adicionar ou alterar a chave gerenciada pelo cliente para um recurso existente.

AWS DevOps Contexto de criptografia do agente

Um contexto de criptografia é um conjunto de pares de valores-chave não secretos que contêm informações contextuais adicionais sobre os dados. AWS O KMS usa o contexto de criptografia como dados autenticados adicionais para oferecer suporte à criptografia autenticada. Quando você inclui um contexto de criptografia em uma solicitação para criptografar dados, o AWS KMS vincula o contexto de criptografia aos dados criptografados. Para descriptografar dados, você deve incluir o mesmo contexto de criptografia na solicitação.

AWS DevOps O agente usa o seguinte contexto de criptografia em todas as operações criptográficas:

```
{
  "aws-crypto-ec:aws:aidevops:arn": "arn:aws:aidevops:{region}:{accountId}:
{resourceType}/{resourceId}"
}
```

O valor do contexto de criptografia é o ARN do recurso do AWS DevOps agente que está sendo criptografado. Você pode usar esse contexto de criptografia em suas principais condições de política e em AWS CloudTrail registros para auditar como sua chave está sendo usada.

Gerenciamento de chaves

Se você desativar ou programar a exclusão da sua chave KMS, o AWS DevOps Agente não poderá descriptografar seus dados. Isso resulta em `AccessDeniedException` erros nas operações que lêem dados criptografados.

> **Important**
>
> Se você optar por usar uma chave gerenciada pelo cliente, será responsável por gerenciar a chave e suas permissões. Se a chave for desativada ou excluída, ou se o AWS

DevOps Agente perder a permissão para usar a chave, você perderá o acesso aos dados criptografados.

A tabela a seguir descreve cenários de falha comuns:

Ação	Impacto
Principais permissões de política revogadas	`AccessDeniedException` em operações de criptografia e descriptografia
A chave KMS está desativada	`DisabledException` em operações de criptografia e descriptografia
A chave KMS está programada para exclusão	`KMSInvalidStateException` em operações de criptografia e descriptografia
A chave KMS é excluída	Perda permanente de dados — dados criptografados não podem ser recuperados

Antes de desativar ou excluir uma chave:

1. Verifique se nenhum recurso ativo do AWS DevOps Agente depende da chave.
2. Considere desativar a chave primeiro para testar o impacto antes de programar a exclusão.
3. AWS O KMS impõe um período mínimo de espera antes da exclusão da chave, dando a você tempo para cancelar, se necessário.

Observação: o AWS DevOps agente não recriptografa automaticamente os dados com uma nova chave. Se você precisar alternar para uma nova chave gerenciada pelo cliente, deverá criar um novo recurso com a nova chave.

Monitorar suas chaves de criptografia

Ao usar uma chave gerenciada pelo cliente com o AWS DevOps Agent, você pode usar AWS CloudTrailpara rastrear as solicitações que o AWS DevOps Agente envia ao AWS KMS.

Você pode filtrar CloudTrail eventos por:

- Fonte do evento — `kms.amazonaws.com`
- Chave de contexto de criptografia — `aws-crypto-ec:aws:aidevops:arn`
- ARN da chave — Seu cliente gerenciou o ARN da chave nos parâmetros da solicitação

Para obter mais informações, consulte Registrar chamadas da API AWS KMS com AWS CloudTrail o AWS Key Management Service Developer Guide.

VPC endpoints (AWS PrivateLink)

Você pode usar AWS PrivateLink para criar uma conexão privada entre sua VPC e AWS DevOps o agente. Você pode acessar o AWS DevOps Agente como se estivesse em sua VPC, sem o uso de um gateway de internet, dispositivo NAT, conexão VPN ou conexão Direct Connect. As instâncias em sua VPC não precisam de endereços IP públicos para acessar AWS DevOps o Agente.

Você estabelece essa conexão privada criando um endpoint de interface, alimentado por AWS PrivateLink. Criaremos um endpoint de interface de rede em cada sub-rede que você habilitar para o endpoint de interface. Essas são interfaces de rede gerenciadas pelo solicitante que servem como ponto de entrada para o tráfego destinado ao Agente. AWS DevOps

Para obter mais informações, consulte Acesse AWS os serviços AWS PrivateLink no _AWS PrivateLink Guide_.

Considerações sobre endpoints VPC AWS DevOps do Agent

Antes de configurar um endpoint de interface para o AWS DevOps Agent, revise as considerações no PrivateLink _AWS Guide_.

AWS DevOps O agente oferece suporte para fazer chamadas de API por meio dos seguintes VPC endpoints.

Categoria	Sufixo do endpoint
AWS DevOps Ações da API do plano de controle do agente	`aidevops`
AWS DevOps Operações de tempo de execução do	`aidevops-dataplane`

Categoria	Sufixo do endpoint
AWS DevOps Eventos do Agent Webhook	`event-ai`

Crie um endpoint de interface para o Agent AWS DevOps

Você pode criar um endpoint de interface para o AWS DevOps Agent usando o console Amazon VPC ou AWS a Interface de Linha de Comando (AWS CLI). Para obter mais informações, consulte Criar um endpoint de interface no _AWS PrivateLink Guide_.

Crie um endpoint de interface para o AWS DevOps Agent usando os seguintes nomes de serviço:

- com.amazonaws. {região} .aidevops
- com.amazonaws. {region} .aidevops-dataplane
- com.amazonaws. {região} .event-ai

Após criar o endpoint, você tem a opção de habilitar um nome de host DNS privado. Habilite essa configuração selecionando Habilitar nome DNS privado no console da VPC ao criar o VPC endpoint.

Se você habilitar o DNS privado para o endpoint da interface, poderá fazer solicitações de API ao AWS DevOps Agente usando seu nome DNS regional padrão. O exemplo a seguir mostra o formato do nome DNS regional padrão.

- cp.aidevops. {região} .api.aws
- dp.aidevops. {região} .api.aws
- auxílio do evento. {região} .api.aws

Criar uma política de endpoint para o endpoint de interface

Uma política de endpoint é um recurso do IAM que pode ser anexado ao endpoint de interface. A política de endpoint padrão permite acesso total ao AWS DevOps Agente por meio do endpoint da interface. Para controlar o acesso permitido ao AWS DevOps Agente a partir de sua VPC, anexe uma política de endpoint personalizada ao endpoint da interface.

Uma política de endpoint especifica as seguintes informações:

- Os diretores que podem realizar ações (AWS contas, usuários do IAM e funções do IAM).

- As ações que podem ser realizadas.
- Os recursos nos quais as ações podem ser executadas.

Para obter mais informações, consulte Controle o acesso aos serviços usando políticas de endpoint no _AWS PrivateLink Guide_.

Validação de conformidade para o AWS DevOps agente

Auditores terceirizados avaliam a segurança e a conformidade dos AWS serviços como parte de vários programas de AWS conformidade. AWS DevOps O agente está no escopo dos seguintes programas de conformidade: BIO, C5, CISPE, CPSTIC, ENS High, FINMA, GNS, GSMA, HITRUST, IRAP, ISMAP, ISO (ISO/IEC 27001, 27017, 27018, 27701, 22301, 20000, 9001), CSA STAR, MTCS, OSPAR, PCI, Pinakes e SOC. PiTuKri Além disso, o AWS DevOps agente é elegível para a HIPAA. Nossos auditores terceirizados analisarão e testarão o AWS DevOps Agente durante os próximos ciclos de auditoria desses programas de conformidade.

Para obter uma lista de AWS serviços no escopo de programas de conformidade específicos, consulte AWS serviços no escopo por programa de conformidade. Para obter informações gerais, consulte Programas de conformidade da AWS.

Você pode baixar relatórios de auditoria de terceiros usando o AWS Artifact. Para obter mais informações, consulte Como baixar relatórios no AWS Artifact.

Sua responsabilidade de conformidade ao usar o AWS DevOps Agent é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade da sua empresa e pelas leis e regulamentos aplicáveis. AWS fornece os seguintes recursos para ajudar na conformidade:

- Guias de início rápido sobre segurança e conformidade — Esses guias de implantação discutem considerações arquitetônicas e fornecem etapas para a implantação de ambientes básicos focados em segurança e conformidade em. AWS
- AWS recursos de conformidade — uma coleção de pastas de trabalho e guias que podem ser aplicados ao seu setor e localização.
- AWS Config — Esse AWS serviço avalia o quão bem suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes do setor e os regulamentos.
- AWS Security Hub — Esse AWS serviço fornece uma visão abrangente do seu estado de segurança interno AWS. O Security Hub usa controles de segurança para avaliar seus AWS

recursos e verificar sua conformidade com os padrões e as melhores práticas do setor de segurança.

Cotas

AWS DevOps As cotas de agentes incluem número de espaços para agentes, investigações simultâneas e muito mais. Algumas cotas podem ter seus limites aumentados mediante solicitação, mas nem todas são passíveis de aumento. Esses aumentos não são concedidos imediatamente, portanto, pode levar de algumas horas a dias para que seu aumento entre em vigor. A menos que especificado de outra forma, cada cota é específica para a região.

A tabela a seguir descreve as cotas do AWS DevOps Agente.

Nome	Padrão	Ajustável	Description
Espaços de agente por conta por região	100	Sim	O número máximo de espaços de agente que você pode criar por conta em cada AWS região.
Investigações simultâneas por espaço de agente	3	Sim	O número máximo de investigações de resolução de incidentes que podem ser executadas simultaneamente em um único espaço de agente.
Avaliações simultâneas por espaço de agente	1	Não	O número máximo de avaliações de prevenção de incidentes que podem ser executadas simultaneamente em um único espaço de agente.

Nome	Padrão	Ajustável	Description
Invocações simultâneas sob demanda por espaço de agente	10	Sim	O número máximo de DevOps invocações sob demanda que podem ser executadas simultaneamente em um único espaço de agente.

Solicitar um aumento de cota

Você pode solicitar um aumento de cota usando uma das seguintes opções:

- No AWS Management Console — Abra o console Service Quotas. No painel de navegação, escolha Serviços da AWS . Selecione DevOps Agente, selecione uma cota e siga as instruções para solicitar um aumento de cota. Para obter mais informações, consulte Solicitando um Aumento de Cota no Guia do Usuário do Service Quotas.
- Da AWS CLI — Use o comando da CLI request-service-quota-increase AWS . Para obter mais informações, consulte Solicitando um Aumento de Cota no Guia do Usuário do Service Quotas.

Histórico do documento

A tabela a seguir descreve as mudanças importantes na documentação desde a última versão do AWS DevOps Agent. Para receber notificações sobre atualizações dessa documentação, é possível inscrever-se em um feed RSS.

Alteração	Descrição	Data
Instruções do agente	Foram adicionadas instruções globais e específicas do agente (AGENTS.md) que se aplicam a todas as sessões.	21 de maio de 2026
Ignorar habilidade e status de triagem IGNORADA	Foi adicionado um exemplo de habilidade de filtragem de incidentes (salto da janela de manutenção), decisão de triagem IGNORADA, instruções para corrigir decisões de triagem e o evento Investigação ignorada. EventBridge	20 de maio de 2026
Enviando anexos de arquivo	Foi adicionada documentação para anexar imagens, documentos e arquivos de código às mensagens de bate-papo. Isso inclui tipos de arquivo, limites e casos de uso compatíveis.	19 de maio de 2026
Priorização de recomendações	Foi adicionada a classificação da lista de AI-powered pendências, incluindo personalização de prioridades via chat e estabilidade de classificação.	13 de maio de 2026

Alteração	Descrição	Data
Plugin Claude Code para MCP	Foi adicionada referência ao exemplo do plug-in Claude Code na seção de integração do MCP.	12 de maio de 2026
Guardrails de permissão	Foi adicionado o modelo de proteção da política de sessão, abrangendo permissões padrão, permissões adicionais suportadas e permissões bloqueadas pela grade de proteção.	7 de maio de 2026
Novos IPs estáticos	Foram adicionados novos endereços IP estáticos para conexões de saída em todas as regiões suportadas.	7 de maio de 2026
Histórico de documentos	Página de histórico de documentos adicionada para rastrear a nova documentação.	5 de maio de 2026
Interface com o agente DevOps	Foi adicionada documentação para cinco métodos de acesso: aplicativo web, MCP, ACP, webhooks e API.	28 de abril de 2026
Validação de conformidade	Foi adicionada uma página dedicada de validação de conformidade.	15 de abril de 2026
Começando a usar AWS CloudFormation	Guia de CloudFormation introdução adicionado.	29 de março de 2026

Alteração	Descrição	Data
Conectando-se a ferramentas hospedadas de forma privada	Documentação adicionada para conexões privadas.	29 de março de 2026
Endpoints da interface VPC	Foi adicionada a documentação do VPC endpoint ()AWS PrivateLink.	29 de março de 2026
EventBridge Integração com a Amazon	Guia de EventBridge integração adicionado para aplicativos orientados a eventos.	28 de março de 2026
EventBridge referência de detalhes de eventos	Referência detalhada do evento adicionada para EventBridge integração.	28 de março de 2026
Cotas	Página de cotas de serviço adicionada.	28 de março de 2026
Conectando Grafana	Foi adicionada a documentação de integração de telemetria da Grafana.	27 de março de 2026
Conectando o Azure	Documentação de integração do Azure adicionada.	27 de março de 2026
Conectando recursos do Azure	Guia de conexão do Azure Resources adicionado.	27 de março de 2026
Conectando o Azure DevOps	Guia de DevOps conexão do Azure adicionado.	27 de março de 2026
Conectando PagerDuty	Documentação PagerDuty de integração de comunicação adicionada.	27 de março de 2026

Alteração	Descrição	Data
Migração do Public Preview para o GA	Guia de migração do Public Preview to General Availability foi adicionado.	27 de março de 2026
Disponibilidade geral	Esta é a versão inicial de disponibilidade geral do AWS DevOps Agent.	30 de março de 2026

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.