Guide de l’utilisateur

AWS DevOps Agent

AWS DevOps Agent: Guide de l’utilisateur

Table of Contents

À propos de AWS DevOps l'agent

AWS DevOps L'agent est un agent pionnier qui résout et prévient les incidents de manière proactive, en améliorant continuellement la fiabilité et les performances.

AWS DevOps L'agent enquête sur les incidents et identifie les améliorations opérationnelles en tant qu' DevOps ingénieur expérimenté.

L'agent travaille en :

- Découvrez vos ressources et leurs relations.
- Utilisation de vos outils d'observabilité, de vos compétences, de vos référentiels de code et CI/CD de vos pipelines.
- Corréler les données de télémétrie, de code et de déploiement pour comprendre les relations entre les ressources de votre application.
- Prise en charge d'applications dans des environnements multicloud et hybrides.

Fonctions principales

AWS DevOps L'agent fournit des fonctionnalités complètes de réponse et de prévention des incidents grâce aux fonctionnalités suivantes :

Réponse autonome et permanente aux incidents

AWS DevOps L'agent enquête de manière autonome sur les problèmes dès qu'ils surviennent :

- Enquête automatisée sur les incidents : commence à enquêter immédiatement lorsqu'une alerte ou un ticket d'assistance arrive
- AWS DevOps Agent Chat : interrogez votre infrastructure, analysez l'état du système et dirigez les enquêtes en langage naturel dans l'application Web DevOps Agent Space. Chat fournit des réponses contextuelles en fonction de la page que vous consultez, qu'il s'agisse de poser des questions sur les ressources dans Topology, de diriger une enquête ou de filtrer des recommandations dans Prevention.
- Plans d'atténuation détaillés : fournit des actions spécifiques pour résoudre les incidents, valider le succès et annuler les modifications si nécessaire

- Coordination automatisée des incidents : achemine les observations, les conclusions et les mesures d'atténuation par le biais de vos canaux de communication préférés tels que Slack et ServiceNow
- AWS Intégration du support : créez des dossiers de AWS support directement à partir d'une enquête avec un contexte immédiat fourni aux experts du AWS support

Prévenir les futurs incidents

AWS DevOps L'agent analyse les tendances des incidents historiques pour vous aider à passer d'une lutte réactive contre les incendies à une amélioration opérationnelle proactive :

- Recommandations ciblées : fournit des améliorations spécifiques et exploitables qui renforcent quatre domaines clés : l'observabilité (surveillance, alertes, journalisation), l'optimisation de l'infrastructure (mise à l'échelle automatique, réglage des capacités) et l'amélioration du pipeline de déploiement (tests, validation).
- Apprentissage continu : affine les recommandations en fonction des commentaires de votre équipe

Tirez le meilleur parti de vos DevOps outils

AWS DevOps L'agent s'intègre à vos outils existants sans modifier vos flux de travail :

- Cartographie des ressources de l'application : crée un graphe topologique des ressources de votre application et de leurs relations
- Intégrations intégrées : fonctionne avec les outils d'observabilité les plus courants (Amazon CloudWatch, Dynatrace, Datadog, New Relic et Splunk), les référentiels de code et les pipelines (actions et référentiels, flux de travail et CI/CD référentiels) GitHub GitLab
- Intégration d'outils personnalisés : étendez les fonctionnalités en vous connectant à vos propres serveurs MCP (Model Context Protocol) pour obtenir des outils supplémentaires
- Requêtes d'infrastructure conversationnelle : utilisez le langage naturel pour interroger les AWS ressources, les métriques du système et l'état des alarmes sans avoir à naviguer sur plusieurs consoles. Chat comprend le contexte et conserve l'historique des conversations pour les questions de suivi.

Comment fonctionne AWS DevOps l'agent

AWS DevOps L'agent fonctionne via une architecture à deux consoles. Les administrateurs utilisent la console AWS de gestion pour créer et gérer des espaces d'agents, configurer des intégrations et configurer des contrôles d'accès. Les équipes opérationnelles utilisent l'application Web AWS DevOps Agent pour les activités de réponse aux day-to-day incidents et d'investigation. L'application Web permet aux opérateurs d'interagir avec les investigations des agents, de parcourir la topologie des applications entre comptes et de découvrir les améliorations préventives apportées à l'observabilité, au code, aux pipelines et aux architectures d'infrastructure. Pour en savoir plus, veuillez consulter la section the section called “Prévention proactive des incidents”.

Le service est organisé autour des espaces d'agent, qui sont des conteneurs logiques qui définissent les éléments auxquels l'agent peut accéder et ce à quoi l' AWS DevOps agent peut accéder et étudier. Chaque espace d'agent contient les configurations de votre AWS compte, les intégrations d'outils tiers et les autorisations d'accès. Pour en savoir plus, veuillez consulter la section the section called “Que sont les DevOps Agent Spaces ?”.

AWS DevOps L'agent crée automatiquement une topologie d'application qui cartographie vos ressources et leurs relations. Cette topologie aide le service à comprendre l'architecture de votre application lors des investigations. Pour en savoir plus, veuillez consulter la section the section called “Qu'est-ce qu'une topologie d' DevOps agent ?”.

Avantages

- Réduction du délai moyen de résolution (MTTR) : les enquêtes autonomes démarrent immédiatement, ce qui accélère la résolution des incidents de quelques heures à quelques minutes
- Prévenir les incidents récurrents : des recommandations ciblées s'attaquent aux causes profondes et renforcent la résilience du système
- Améliorez l'efficacité opérationnelle : libérez votre équipe des tâches d'investigation répétitives pour se concentrer sur l'innovation
- Travaillez dans le cadre des flux de travail existants : s'intègre à vos outils et processus existants sans interruption

Qu'est-ce qu'une application Web pour DevOps agents ?

AWS DevOps L'agent utilise une architecture à deux consoles qui sépare les fonctions administratives des activités day-to-day opérationnelles. Cette conception permet aux administrateurs de configurer le service tandis que les équipes opérationnelles se concentrent sur la réponse et la prévention des incidents.

Consoles

AWS DevOps L'agent fournit deux interfaces distinctes :

- AWS Console de gestion : les administrateurs utilisent la console de AWS gestion pour configurer et gérer AWS DevOps l'agent. Dans cette console, vous pouvez the section called "Création d'un espace d'agents" connecter AWS des services et des outils tiers, et gérer les autorisations d'accès pour votre organisation.
- DevOps Application Web Agent : les équipes opérationnelles utilisent les applications Web DevOps Agent Space pour leurs activités quotidiennes de réponse aux incidents. Cette application autonome fournit une interface permettant aux ingénieurs de garde de lancer des enquêtes, d'interagir avec l'agent par le biais d'un chat en langage naturel, de visualiser les topologies des applications et de passer en revue les recommandations de prévention des incidents.

Fonctionnalités de l'application Web

L'application Web DevOps Agent fournit les fonctionnalités principales suivantes :

- Réponse aux incidents : cette page vous permet de créer et de suivre les enquêtes sur les incidents, ainsi que de générer des plans d'atténuation pour résoudre les incidents.
- Prévention des incidents — Sur la page Prévention, vous trouverez des recommandations pour améliorer votre posture d'observabilité, vos processus de livraison et votre architecture d'infrastructure afin de prévenir de futurs incidents.
- Topologie : la page Topologie fournit une représentation visuelle interactive des ressources du compte et de leurs relations entre toutes les ressources des comptes connectés. Vous pouvez afficher la topologie avec différents niveaux de détail à l'aide de la liste déroulante « Afficher » pour basculer entre les vues Système, Conteneur et Ressources.
- Compétences — Des ensembles d'instructions modulaires qui ajoutent à AWS DevOps l'Agent des fonctionnalités spécialisées. Les compétences incluent la connaissance du domaine, les méthodologies d'investigation et les configurations d'outils adaptées à votre infrastructure. Chaque

compétence permet d'utiliser des outils spécifiques et fournit une divulgation progressive des instructions uniquement lorsque cela est pertinent pour l'enquête.

- Interface de chat en langage naturel — Disponible via l'application Web, Chat est un assistant conversationnel alimenté par l'IA qui vous permet d'interroger votre infrastructure, d'analyser l'état du système et de mener des enquêtes en langage naturel. Le chat fournit des réponses contextuelles en fonction de la page que vous consultez.

Authentification

AWS DevOps L'agent prend en charge des méthodes d'authentification flexibles pour répondre aux différentes exigences de l'organisation :

- Intégration à IAM Identity Center (accès utilisateur) — Les organisations peuvent utiliser AWS Identity Center (IAM Identity Center) pour gérer de manière centralisée l'accès des utilisateurs aux applications Web DevOps Agent Space. IAM Identity Center peut se fédérer avec des fournisseurs d'identité externes via les protocoles OIDC et SAML standard, notamment des fournisseurs tels qu'Okta, Ping Identity et Microsoft Entra ID. Cette méthode prend en charge l'authentification multifactorielle auprès de votre fournisseur d'identité.
- Authentification par fournisseur d'identité externe (IdP) : les organisations peuvent connecter un fournisseur d'identité compatible OIDC, tel qu'Okta ou Microsoft Entra ID, directement à l'application Web Agent Space sans avoir besoin d'IAM Identity Center. Les utilisateurs se connectent à l'aide de leurs identifiants d'entreprise via l'IdP. Pour les instructions de configuration, voirthe section called “Configuration de l'authentification par fournisseur d'identité externe (IdP)”.
- Lien d'authentification IAM (accès administrateur) : une autre méthode fournit un accès direct à l'application Web depuis la console de AWS gestion en utilisant votre session de console existante. Cette option est utile avant de mettre en œuvre l'intégration complète d'Identity Center, mais les sessions sont limitées à 10 minutes.

Que sont les DevOps Agent Spaces ?

Un espace d' DevOps agent est un conteneur logique qui définit les outils et l'infrastructure auxquels AWS DevOps l'agent a accès. Chaque espace d'agent fonctionne de manière indépendante avec son propre accès au AWS compte, des intégrations tierces et des autorisations utilisateur.

Un espace d'agent représente la limite de ce à quoi AWS DevOps l'agent peut accéder et étudier pendant la réponse à un incident. Lorsque vous créez un espace agent, vous définissez les AWS

comptes auxquels l'agent peut accéder, les outils externes auxquels il peut se connecter et les utilisateurs de votre organisation qui peuvent interagir avec l'agent.

Chaque espace agent fonctionne comme un déploiement indépendant de l' AWS DevOps agent. Vous configurez l'espace agent via la console de AWS gestion, tandis que vos équipes opérationnelles utilisent l'application Web de l'espace agent pour mener des enquêtes et examiner les recommandations dans cet espace.

Comment les espaces d'agent sont isolés

Les espaces d'agent maintiennent l'isolement pour garantir la sécurité et empêcher tout accès involontaire entre différents environnements ou équipes :

- AWS isolation des comptes : chaque espace agent utilise des rôles IAM dédiés qui n'accordent l'accès qu'à des AWS comptes et à des ressources spécifiques. L'agent ne peut pas accéder à des AWS ressources autres que celles explicitement configurées pour l'espace agent.
- Isolation de l'accès des utilisateurs : vous contrôlez les utilisateurs ou les groupes autorisés à accéder à chaque espace d'agent. Cela vous permet d'aligner les autorisations d'accès sur votre structure organisationnelle, en veillant à ce que les équipes n'interagissent qu'avec leurs espaces d'agents désignés.
- Isolation des données — Les données d'enquête, l'historique des incidents et les recommandations sont conservés séparément dans chaque espace d'agent. Les informations d'un espace agent ne sont pas visibles ou accessibles depuis un autre espace agent.
- Isolation des données de chat - L'historique des conversations de chat est également isolé au sein de chaque espace agent. Les conversations et les requêtes dans un espace agent ne sont pas visibles ou accessibles depuis un autre espace agent.

Application Web Agent Space

Chaque espace d'agent dispose d'une application Web dédiée accessible en dehors de la console de AWS gestion. Consultez the section called “Qu'est-ce qu'une application Web pour DevOps agents ?” pour en savoir plus sur l'application Web.

Quand utiliser plusieurs espaces d'agent

Envisagez de créer plusieurs espaces d'agent pour répondre aux différents besoins organisationnels :

- Séparation des équipes : créez des espaces d'agent dédiés pour les différentes équipes d'application ou unités commerciales afin de maintenir des limites de propriété claires dans l'espace des agents.
- Isolation de l'environnement : séparez les environnements de production et de non-production en différents espaces d'agents afin d'empêcher tout accès accidentel entre environnements.
- Limites de service — Alignez les espaces des agents avec les limites des services ou des applications spécifiques pour que les enquêtes restent ciblées et pertinentes.
- Exigences de conformité — Configurez des espaces d'agent distincts avec différents contrôles d'accès ou paramètres de résidence des données pour répondre aux exigences réglementaires.

> **Note**
>
> Lorsque vous créez plusieurs espaces d'agent, vous pouvez utiliser un AWS compte dédié comme compte principal pour un espace d'agent et connecter des comptes d'application distincts en tant que comptes secondaires. Cette approche vous permet de maintenir des contrôles d'accès précis tout en garantissant que chaque espace d'agent ne peut accéder qu'aux ressources spécifiques à l'étendue prévue, même lors de la création automatique de rôles.

Qu'est-ce qu'une topologie d' DevOps agent ?

AWS DevOps L'agent découvre et visualise automatiquement les ressources et les relations au sein de vos applications et utilise la topologie obtenue pour comprendre votre infrastructure lors des enquêtes sur les incidents et lors de la formulation de recommandations préventives.

Comment sont créés les graphes topologiques

AWS DevOps L'agent crée des graphes topologiques par le biais de plusieurs processus automatisés :

- Découverte des ressources : l'agent analyse automatiquement vos AWS comptes pour identifier les ressources telles que les instances de calcul, les services de stockage, les composants réseau et les bases de données qui font partie de vos applications.
- Détection des relations : l'agent analyse les données de configuration, les CloudFormation piles et les balises de ressources pour déterminer comment les ressources sont liées les unes aux autres.

- Cartographie du code et du déploiement : lorsqu'il est connecté à des CI/CD pipelines, l'agent relie les ressources de l'infrastructure à leurs processus de déploiement et modifie le code de l'application et de l'infrastructure.
- Cartographie des comportements d'observabilité : les données issues de systèmes d'observabilité tels qu'Amazon CloudWatch Application Signals et Dynatrace sont utilisées pour identifier les comportements observés qui indiquent les relations entre les ressources.

Capacités clés

La cartographie des ressources fournit plusieurs fonctionnalités qui améliorent l'investigation et la prévention des incidents :

- Visualisation interactive : explorez la topologie de votre application à l'aide d'un graphique interactif dans l'application Operator Web. Vous pouvez zoomer et parcourir la topologie pour comprendre les relations complexes entre les ressources. Vous pouvez également utiliser Chat pour demander des informations topologiques en langage naturel, par exemple « Afficher toutes les fonctions Lambda connectées à cette table DynamoDB » ou « Quelles ressources sont affectées par cette alarme ? ».
- Investigation contextuelle — Au cours des enquêtes sur les incidents, l' AWS DevOps agent est assisté par la topologie des ressources pour identifier les composants affectés, comprendre le rayon d'explosion et tracer la trajectoire d'impact dans vos systèmes.
- Analyse des causes profondes : la compréhension détaillée des relations entre les ressources permet de déterminer l'origine des problèmes, même dans les systèmes distribués complexes présentant de nombreuses interdépendances.
- Évaluation de l'impact : lors de l'analyse des incidents, l'agent peut mieux déterminer quels services en aval sont susceptibles d'être affectés en identifiant les chaînes de dépendance dans la topologie.
- Recommandations préventives : l'agent utilise les informations topologiques pour formuler des recommandations ciblées visant à améliorer la résilience, en suggérant les modifications qui auront l'impact le plus significatif sur la stabilité du système.

Vues topologiques

La visualisation de la topologie sur la page Topologie de l'Operator Web App offre plusieurs niveaux de détail :

- Appris : vue par défaut, générée à partir de la compétence Agent Space Understanding. Affiche un résumé structuré de votre infrastructure organisé par services logiques et chemins de demande.
- Système — Affiche les limites générales du compte et de la région.
- Conteneur : affiche les piles de déploiement sous forme de CloudFormation piles contenant des ressources connexes.
- Composants : affiche les composants individuels au sein des conteneurs et leurs relations.
- Toutes les ressources : affiche une vue complète de toutes les ressources découvertes et de leurs relations.

Découverte de ressources

Les ressources sont découvertes par le biais de deux méthodes :

- CloudFormation piles — L'agent répertorie toutes les CloudFormation piles et leurs ressources dans le AWS compte principal et dans tous les comptes secondaires connectés. Cela est pris en charge pour tous les infrastructure-as-code outils utilisés CloudFormation pour le déploiement, y compris AWS le Cloud Development Kit (AWS CDK).
- Explorateur de ressources : pour les ressources non déployées depuis CloudFormation, les ressources balisées sont découvertes à partir de l'explorateur de AWS ressources. L'explorateur de ressources doit être activé sur le AWS compte cible. Cela est utile pour identifier les limites des applications pour les ressources déployées via la console de AWS gestion, le AWS service APIs ou d'autres infrastructure-as-code frameworks.

Champ d'investigation au-delà de la topologie

Bien que la topologie de l'application fournisse un contexte important lors des investigations, AWS DevOps Agent ne se limite pas à étudier uniquement les ressources indiquées dans la topologie. L'agent peut utiliser des sources de données supplémentaires, telles que des AWS services APIs ou des outils d'observabilité connectés, pour étudier les ressources qui ne figurent pas dans la topologie de l'application.

Pour limiter les ressources auxquelles l'agent a accès, limitez la politique relative au rôle attribué à l'agent pour accéder aux ressources entre comptes. Pour de plus amples informations, veuillez consulter the section called “Limiter l'accès des agents à un AWS compte”.

Compétence de compréhension de la topologie et de l'espace des agents

Le graphe topologique alimente la compétence acquise dans Agent Space Understanding, qui code un résumé structuré de votre infrastructure à utiliser lors des enquêtes. Lorsque la découverte de la topologie est terminée pour un nouvel espace d'agent, le système génère automatiquement la compétence Agent Space Understanding. Pour plus d'informations sur les compétences acquises, consultezthe section called "Compétences apprises".

DevOps Compétences des agents

AWS DevOps Les compétences des agents sont des ensembles d'instructions modulaires qui étendent les capacités de l'agent grâce à des connaissances spécialisées dans le domaine et à des méthodologies d'investigation adaptées à votre infrastructure et à vos flux de travail opérationnels.

Que sont les compétences

Les compétences sont des répertoires autonomes contenant des instructions Markdown qui fournissent des fonctionnalités spécialisées à l' AWS DevOps Agent. AWS DevOps Agent prend en charge un sous-ensemble de la spécification Agent Skills, une norme ouverte pour les instructions et les ressources relatives aux agents d'emballage, qui ne prend en charge que les documents non exécutables : instructions Markdown, PDFs images et fichiers de données.

Chaque compétence nécessite un fichier Skill.md contenant les instructions que vous souhaitez fournir à votre AWS DevOps agent. Outre le fichier Skill.md requis, les compétences peuvent inclure :

- Flux de travail d'investigation pour des scénarios ou des types d'infrastructure spécifiques.
- Matériaux de référence, y compris les modèles d'architecture et les procédures opérationnelles.
- Ciblage par type d'agent — Les compétences peuvent être ciblées sur des types d'agents spécifiques (générique, à la demande, triage des incidents, RCA des incidents, atténuation des incidents, évaluation) afin de réduire la consommation de contexte et d'améliorer la concentration des agents.

Pourquoi utiliser Skills

Grâce à ses compétences, AWS DevOps l'agent passe du statut d'assistant généraliste à celui de spécialiste de votre infrastructure et de vos flux de travail opérationnels. Contrairement aux instructions ponctuelles fournies dans un message de chat, les compétences sont des fonctionnalités

réutilisables qui se chargent automatiquement lorsqu'elles sont pertinentes pour les tâches effectuées par AWS DevOps l'Agent.

Principaux avantages :

- Spécialisez votre agent : adaptez AWS DevOps l'agent aux procédures d'enquête, aux meilleures pratiques et aux connaissances organisationnelles spécifiques à votre infrastructure et à vos modèles opérationnels.
- Réduisez les répétitions : créez des flux de travail d'investigation une seule fois et l' AWS DevOps agent les utilisera automatiquement pour toutes les enquêtes pertinentes, éliminant ainsi le besoin de fournir les mêmes instructions à plusieurs reprises.
- Capacités de composition : combinez plusieurs compétences pour créer des flux de travail end-to-end d'investigation. AWS DevOps L'agent lit plusieurs compétences pendant l'exécution, comme une compétence pour récupérer des déploiements depuis votre CI/CD pipeline personnalisé et une compétence pour rechercher dans vos référentiels de code.
- Amplifier les outils personnalisés : créez des compétences qui guideront l' AWS DevOps agent dans l'utilisation efficace de vos outils de serveur MCP personnalisés. Les compétences peuvent documenter quand invoquer des outils spécifiques, quels paramètres utiliser pour différents scénarios et comment interpréter les résultats pour réaliser des flux de travail spécifiques à votre infrastructure.

Comment fonctionnent les compétences

Lorsque AWS DevOps l'agent est confronté à une tâche pertinente, il charge les compétences appropriées et suit les instructions pour guider son enquête. Par exemple, une compétence « Investigation des performances des bases de données » peut inclure des step-by-step procédures d'analyse des problèmes de régulation RDS, permettant à l'agent de vérifier systématiquement l'état des alarmes, d'analyser les métriques de connexion et d'identifier les requêtes lentes.

Structure des compétences

Une compétence est organisée sous la forme d'un répertoire contenant :

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

Skill.MD

`SKILL.md`Il s'agit du seul fichier obligatoire. Il contient les instructions de base écrites au format Markdown. Ce fichier doit :

- Décrivez quand et comment utiliser la compétence.
- Fournir des procédures step-by-step d'enquête.
- Incluez des arbres de décision pour différents scénarios.
- Documenter les résultats attendus et les critères de réussite.

Matière première

Le Frontmatter est le bloc de métadonnées situé en haut d'un `SKILL.md` fichier, entre des `---` délimiteurs. Il contient les `description` champs `name` et que AWS DevOps l'agent utilise pour déterminer quand activer la compétence lors d'une enquête ou d'une tâche.

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

name — Identifiant unique de la compétence. Utilisez uniquement des lettres minuscules, des chiffres et des traits d'union (64 caractères maximum). Ne doit pas commencer ou se terminer par un trait d'union.

description — Une explication détaillée du moment et des raisons pour lesquelles AWS DevOps l'agent doit utiliser cette compétence. AWS DevOps L'agent évalue ce champ pour déterminer si la compétence est pertinente pour la tâche en cours. Une description vague ou manquante peut amener l'agent à ignorer complètement la compétence, même si les instructions sont bien rédigées.

Important — Rédigez la description du point de vue de l'agent. Incluez les scénarios, les services, les types d'erreur ou les symptômes spécifiques qui devraient déclencher la compétence. Par exemple, « Utiliser cette compétence pour étudier la latence de la base de données, les erreurs de connexion ou les délais de requête pour les instances Amazon RDS » est plus efficace que la « compétence RDS ».

Lorsque vous créez une compétence dans l'interface utilisateur, le système génère automatiquement des informations préliminaires à partir du nom et de la description que vous fournissez. Les compétences téléchargées sous forme de fichiers zip doivent inclure des éléments de base dans le `SKILL.md` fichier.

Exemple : compétence complète

L'exemple suivant montre une compétence complète et bien formée pour étudier les problèmes de performances RDS. Il présente la structure du répertoire, le dossier de Skill.md, les procédures d'enquête exploitables et un fichier de références supplémentaire.

Structure du répertoire :

```
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
```

références/ .md rds-metrics-reference :

```
# RDS CloudWatch Metrics Reference


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

Création de compétences

Avant de créer des compétences, vous devez disposer d'un espace agent. Pour de plus amples informations, veuillez consulter the section called “Création d'un espace d'agents”.

Vous pouvez créer des compétences de deux manières en fonction de vos préférences en matière de flux de travail et de la complexité des compétences :

Création d'une compétence dans l'interface utilisateur

Les compétences créées dans l'application Web AWS DevOps Agent Operator contiennent un nom, une description et des instructions dans un seul fichier Skill.md.

Pour créer une compétence dans l'interface utilisateur :

- Accédez à la page Compétences de votre application Web Agent Space Operator.
- Cliquez sur « Ajouter une compétence ».
- Sélectionnez « Créer une compétence » dans le modal.
- Remplissez le formulaire de compétence :
  - Nom : lettres minuscules, chiffres et tirets uniquement (64 caractères maximum). Ne doit pas commencer ou se terminer par un trait d'union. Exemple : `rds-throttling-investigation`
  - Description — Brève explication des circonstances dans lesquelles utiliser cette compétence (minimum 100 caractères recommandés, maximum 1 024 caractères). Cela permet à l'agent de déterminer quand activer la compétence.
  - État : défini sur Actif (par défaut) ou Inactif. Les compétences inactives ne sont pas utilisées par l'agent.
  - Type d'agent — Sélectionnez un ou plusieurs types d'agents pouvant utiliser cette compétence. L'option Générique est sélectionnée par défaut et met la compétence à la disposition de tous les types d'agents. Pour cibler des agents spécifiques, désélectionnez Générique et choisissez parmi : On-Demand, Incident Triage, Incident RCA, Incident Mitigation ou Evaluation.
  - Instructions — Step-by-step procédures au format Markdown. Soyez précis et exploitable.
- Cliquez sur « Créer » pour enregistrer la compétence.

Le système génère automatiquement un fichier Skill.md avec la structure de fond appropriée.

Pour modifier une compétence créée dans l'interface utilisateur :

- Accédez à la compétence dans la liste des compétences et cliquez dessus pour l'ouvrir.
- Cliquez sur Modifier.
- Modifiez le nom, la description ou les instructions.

- Cliquez sur Enregistrer pour mettre à jour la compétence.

Téléchargement d'une compétence

Les compétences téléchargées sous forme de fichiers zip contiennent un fichier Skill.md ainsi que des ressources supplémentaires telles que des documents de référence ou des actifs.

Structure des compétences :

```
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
```

Exigences relatives à la face avant Skill.md :

Les compétences téléchargées sous forme de fichiers zip doivent inclure le frontmatter dans Skill.md avec `name` les champs et. `description` AWS DevOps L'agent utilise ces champs pour déterminer quand activer la compétence. Pour plus de détails sur la rédaction d'un texte de base efficace, consultez la section du sujet principal plus haut dans cette rubrique.

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

Pour créer une compétence par téléchargement zip :

- Créez un répertoire avec vos fichiers de compétences en suivant la structure ci-dessus.

- Assurez-vous que Skill.md inclut une entrée appropriée (nom et description).
- Compressez le répertoire dans un fichier .zip.
- Accédez à la page Compétences de votre application Web Agent Space Operator.
- Cliquez sur « Ajouter une compétence ».
- Sélectionnez « Télécharger une compétence » dans le modal.
- Glissez et déposez votre fichier .zip ou cliquez pour le parcourir (fichiers ZIP uniquement, maximum 6 Mo).
- Sélectionnez un ou plusieurs types d'agents qui peuvent utiliser cette compétence (le mode générique est sélectionné par défaut et s'applique à tous les types d'agents ; désélectionnez cette option pour cibler spécifiquement On-Demand, le triage des incidents, le RCA des incidents, l'atténuation des incidents ou l'évaluation).
- Passez en revue les exigences relatives au fichier zip et les résultats de validation.
- Cliquez sur « Télécharger » pour ajouter la compétence à votre espace agent.

Restrictions importantes concernant les compétences téléchargées sous forme de fichiers zip :

- Les scripts ne sont actuellement pas pris en charge : les compétences contenant des scripts dans le `scripts/` répertoire seront rejetées lors du téléchargement. L'exécution de scripts sera activée dans une future version une fois que les agents auront accès à un environnement de codage sécurisé.
- Limite de taille — La taille totale du fichier zip ne doit pas dépasser 6 Mo (y compris tous les fichiers).
- Skill.md obligatoire — Le fichier zip doit contenir un fichier Skill.md avec un frontal valide.

Bonnes pratiques en matière de compétences en matière de dénomination :

Utilisez des noms clairs et descriptifs tels que « rds-throttling-investigation » plutôt que des noms génériques. Un bon nom de compétence reflète le scénario ou le service spécifique auquel il répond, ce qui permet d'identifier plus facilement la bonne compétence en un coup d'œil.

Gestion des skills

AWS DevOps L'agent fournit des fonctionnalités complètes de gestion des compétences via l'application Web Operator :

Répertorier les compétences — Afficher toutes les compétences de votre espace d'agent. La page Compétences affiche le nom de la compétence, son statut actif ou inactif, sa date de création, la date de dernière mise à jour et les actions disponibles.

Visualisation des compétences : cliquez sur n'importe quelle compétence pour voir sa vue détaillée. Les compétences créées dans l'interface utilisateur affichent un contenu modifiable dont vous pouvez modifier le nom, la description ou les instructions directement dans l'interface utilisateur et cliquer sur « Enregistrer » pour les mettre à jour. Les compétences téléchargées sous forme de fichiers zip affichent une arborescence de fichiers contenant Skill.md et tous les répertoires supplémentaires tels que references/ et assets/. Cliquez sur les fichiers dans l'arborescence pour afficher leur contenu en mode lecture seule.

Sélection d'agents pour une compétence : configurez les types d'agents autorisés à utiliser chaque compétence lors de sa création ou de sa modification. Dans le menu déroulant Type d'agent, sélectionnez un ou plusieurs types d'agents à l'aide des cases à cocher : générique (par défaut, s'applique à tous les types d'agents), à la demande (requêtes conversationnelles), triage des incidents (évaluation initiale des incidents), RCA des incidents (analyse des causes profondes), atténuation des incidents (réponse automatique aux incidents) ou évaluation (recommandations proactives). L'option Générique est sélectionnée par défaut et met la compétence à la disposition de tous les types d'agents. Les compétences destinées à des agents spécifiques réduisent la consommation de contexte et améliorent la concentration des agents.

Activation et désactivation des compétences : désactivez temporairement les compétences sans les supprimer à l'aide du bouton Active/Inactive . Ouvrez la vue détaillée des compétences et basculez sur « Inactif » pour empêcher l'agent de la charger pour de nouvelles enquêtes, tout en préservant l'ensemble du contenu et des configurations. Les enquêtes en cours continuent d'utiliser cette compétence. Revenez à « Active » pour que la compétence soit à nouveau immédiatement disponible.

Mise à jour des compétences : modifiez les compétences existantes en fonction de la façon dont elles ont été créées. Pour les compétences créées dans l'interface utilisateur, cliquez sur « Modifier » dans la vue détaillée des compétences, modifiez le nom, la description ou les instructions, puis cliquez sur « Enregistrer » pour les mettre à jour. Pour les compétences téléchargées sous forme de fichiers zip, modifiez les fichiers localement, créez un nouveau fichier zip et importez une nouvelle version.

Supprimer des compétences : supprimez définitivement des compétences de votre espace agent. Ouvrez l'affichage de la liste des compétences, cliquez sur le menu Autres options () et sélectionnez « Supprimer », consultez l'avertissement concernant la suppression définitive, saisissez le nom de

la compétence pour confirmer, puis cliquez sur « Supprimer la compétence ». La suppression ne peut pas être annulée. Les enquêtes en cours peuvent être affectées si elles tentent de charger la compétence supprimée. Pour les compétences téléchargées sous forme de fichiers zip, téléchargez le fichier zip avant de le supprimer en tant que sauvegarde. Envisagez de désactiver la compétence au lieu de la supprimer si vous en avez à nouveau besoin.

Migration depuis Runbooks

Les Runbooks existants sont automatiquement migrés vers Skills sans qu'aucune action du client ne soit requise. Lorsque votre espace d'agent passe au modèle de compétences, tous les Runbooks sont convertis en compétences et apparaissent dans votre interface utilisateur de compétences. Après la migration, vous pouvez :

- Vérifiez les compétences migrées : vérifiez que la migration automatique a correctement converti vos Runbooks.
- Mise à jour selon les besoins : modifiez les compétences directement dans l'interface utilisateur pour affiner les instructions, mettre à jour les descriptions ou configurer le ciblage par type d'agent.
- Développez avec des références — Pour les compétences qui bénéficieraient de documents de référence ou de diagrammes d'architecture supplémentaires, recréez-les sous forme de compétences de téléchargement zip avec un répertoire references/ ou assets/.
- Créez de nouvelles compétences — Ajoutez de nouvelles compétences pour les flux de travail d'investigation qui n'étaient pas couverts auparavant par Runbooks.

Contactez le AWS Support si vous rencontrez des problèmes avec les compétences migrées automatiquement ou si vous avez besoin d'aide pour les mises à jour après la migration.

Compétences apprises

Quelles sont les compétences apprises ?

Les compétences acquises sont des fichiers de connaissances structurés que l' DevOps agent génère à partir des données de votre espace agent. Chaque compétence acquise code un type spécifique de connaissances que l' AWS DevOps agent utilise dans le cadre de ses tâches. Au lancement, deux compétences apprises sont disponibles : la compréhension de l'espace des agents et les meilleures pratiques d'utilisation des outils.

Comprendre l'espace des agents

La compétence Agent Space Understanding (`understanding-agent-space`) analyse vos comptes cloud connectés, vos référentiels de code et vos intégrations de télémétrie afin de créer une carte des ressources et des relations au sein d'un agent space.

La compétence produit un `SKILL.md` fichier principal et un ensemble de fichiers de référence. Le fichier principal contient une présentation du système en langage clair avec les principaux concepts de domaine, les environnements de déploiement (paires de AWS comptes et de régions, abonnements Azure et régions, etc.), un schéma d'architecture au niveau du conteneur montrant comment les services logiques se connectent, les chemins de demande essentiels à votre application avec les composants qu'ils traversent, et un mappage des référentiels de code aux conteneurs.

Chaque conteneur logique reçoit un fichier de référence dédié décrivant ses composants internes (calcul, données, messagerie, réseau, etc.) avec les types de ressources et les identifiants physiques tels que ARNs les noms des tables et les files d'attente URLs. Le fichier de référence capture également la couverture de l'observabilité, y compris les alarmes, les tableaux de bord et les moniteurs liés à chaque composant. Il associe également chaque composant à ses référentiels de code, packages et infrastructure-as-code définitions associés, fournissant ainsi une chaîne de traçabilité complète depuis le code source jusqu'aux ressources déployées.

Chaque chemin de demande critique reçoit un fichier de référence dédié décrivant le flux de end-to-end demandes complet selon la granularité des composants, depuis le point d'entrée jusqu'à chaque service intermédiaire, magasin de données et dépendance externe. Le fichier comprend un organigramme séquencé montrant l'ordre des opérations et les mécanismes d'interaction entre les composants, ainsi que la responsabilité de chaque participant. Il répertorie également les signaux d'observabilité relatifs au chemin : modèles de groupes de logs pour chaque saut, indicateurs clés (latence, taux d'erreur, limitation, quotas de jetons) avec leurs noms et dimensions d'alarme, et intervalles de suivi distribués qui peuvent être corrélés entre les services et les comptes.

Meilleures pratiques d'utilisation des outils

La compétence Bonnes pratiques d'utilisation des outils analyse les utilisations passées des outils d'investigation afin d'extraire des modèles d'utilisation efficaces, des modes de défaillance courants et des indications sur les paramètres. Cela permet à l' DevOps agent d'éviter les pièges connus et de mener des enquêtes en réduisant le nombre d'étapes inutiles. La compétence produit un fichier principal et un ensemble de fichiers de référence par outil. Le fichier principal sert d'index de routage qui répertorie chaque outil avec les scénarios d'investigation qu'il prend en charge et renvoie au fichier de référence correspondant.

Chaque fichier de référence par outil peut inclure jusqu'à trois sections :

- Meilleures pratiques — Techniques axées sur l'investigation issues d'une utilisation réussie des outils, telles que les modèles de requêtes CloudWatch Logs Insights, les espaces de noms et dimensions de métriques spécifiques à l'environnement et les filtres de source d'événements. CloudTrail Chaque entrée est organisée autour d'un scénario d'enquête et inclut des valeurs de paramètres concrets et des exemples observés lors d'enquêtes antérieures.
- Erreurs courantes — Modes de défaillance récurrents et leurs correctifs. Chaque entrée décrit une condition d'erreur spécifique, telle que l'interrogation d'un compte inaccessible ou la création d'une requête d'agrégation mal formée, et fournit une action corrective afin que l'agent puisse éviter ou corriger l'erreur sans gaspiller les étapes d'investigation.
- Gestion des résultats : conseils pour les appels d'outils qui ont tendance à renvoyer des réponses volumineuses. Chaque entrée décrit un changement de paramètre ou une stratégie de traitement qui réduit la taille de sortie tout en préservant la valeur diagnostique.

Lorsque l'accès à l'infrastructure en direct est disponible, la compétence valide les modèles par rapport à votre environnement avant de les inclure. Les modèles confirmés sont énoncés avec confiance, les modèles non confirmés utilisent un langage prudent et les modèles réfutés sont exclus. Cela permet de maintenir les compétences en phase avec l'état actuel de votre infrastructure.

Gestion des compétences acquises

Mises à jour — L' DevOps agent génère et met à jour automatiquement les compétences acquises en fonction de l'activité dans votre espace agent. Ce qui suit décrit le moment où chaque compétence est mise à jour.

L' DevOps agent génère une compétence mise à jour des meilleures pratiques d'utilisation des outils toutes les 30 enquêtes.

La compétence Agent Space Understanding est générée par l'agent d'apprentissage, qui s'exécute chaque fois que vous ajoutez, mettez à jour ou supprimez une fonctionnalité ou une intégration Agent Space.

Pour régénérer les compétences apprises manuellement, cliquez sur le bouton Régénérer sur la page Topologie de l'application de l'opérateur, ou discutez avec l'agent et demandez-lui de mettre à jour les compétences acquises.

Désactivation : les compétences apprises sont actives par défaut. Lorsqu'ils sont actifs, l' DevOps agent les charge au début de chaque tâche de DevOps l'agent. Pour empêcher l'application d'une

compétence apprise, désactivez-la dans l'afficheur de compétences de l'application pour opérateurs. La désactivation d'une compétence ne la supprime pas. La compétence est conservée et peut être réactivée à tout moment. Lorsqu'une compétence est désactivée, l' DevOps agent agit à son insu.

Vue topologie : la page Topologie de l'application Web de votre agent Space utilise la compétence Agent Space Understanding pour afficher visuellement votre environnement Agent Space sous forme de conteneurs et de composants logiques. Cliquez sur n'importe quel conteneur pour voir ses composants, ses identificateurs de ressources et sa télémétrie.

Régions prises en charge

Cette rubrique décrit les AWS régions dans lesquelles vous pouvez utiliser l' AWS DevOps agent. Pour plus d'informations sur AWS les régions, voir Spécifier les AWS régions que votre compte peut utiliser dans le Guide de référence sur la gestion des AWS comptes.

Surveillance des ressources entre les régions

AWS DevOps L'agent peut surveiller et étudier les ressources des AWS comptes situés dans n'importe quelle AWS région, quelle que soit la région prise en charge dans laquelle vous créez votre espace d'agent. Lorsque vous associez un AWS compte à un espace d'agent, l'agent découvre et cartographie les ressources de toutes les régions de ce compte. Cela signifie que vous n'avez pas besoin d'un espace d'agent dans chaque région où s'exécutent vos charges de travail.

Choisissez une région prise en charge en fonction de la résidence de vos données préférée, de la proximité de votre équipe opérationnelle ou des exigences organisationnelles.

Régions prises en charge

AWS DevOps L'agent est disponible dans les AWS régions suivantes.

Nom de la région	Code région	Lien vers la console
USA Est (Virginie du Nord)	us-east-1	Console ouverte
USA Ouest (Oregon)	us-west-2	Console ouverte
Asie-Pacifique (Sydney)	ap-southeast-2	Console ouverte
Asie-Pacifique (Tokyo)	ap-northeast-1	Console ouverte

Nom de la région	Code région	Lien vers la console
Europe (Francfort)	eu-central-1	Console ouverte
Europe (Irlande)	eu-west-1	Console ouverte

Points de terminaison de service

Nom de la région	Code région	Endpoint	Protocole
USA Est (Virginie du Nord)	us-east-1	aidevops.us-east-1.amazonaws.com	HTTPS
USA Ouest (Oregon)	us-west-2	aidevops.us-west-2.amazonaws.com	HTTPS
Asie-Pacifique (Sydney)	ap-southeast-2	aidevops.ap-southeast-2.amazonaws.com	HTTPS
Asie-Pacifique (Tokyo)	ap-northeast-1	aidevops.ap-northeast-1.amazonaws.com	HTTPS
Europe (Francfort)	eu-central-1	aidevops.eu-central-1.amazonaws.com	HTTPS
Europe (Irlande)	eu-west-1	aidevops.eu-west-1.amazonaws.com	HTTPS

Considérations

- Sélection de la région de l'espace des agents — Un espace d'agents et ses données (enquêtes, topologie, recommandations) sont stockées dans la région où vous les créez. Choisissez une région qui répond à vos exigences en matière de résidence des données.

- Surveillance interrégionale — Ressources dans les AWS comptes associés à un agent

L'espace est surveillé quelle que soit la région dans laquelle ces ressources sont déployées. Il n'est pas nécessaire de créer des espaces d'agent distincts dans chaque région où s'exécutent vos charges de travail.

- Intégrations tierces — Connexions aux CI/CD fournisseurs (GitHub, GitLab),

les outils d'observabilité (Dynatrace, Datadog, New Relic, Splunk) et les serveurs MCP sont configurés par agent Space et ne dépendent pas de la région.

Commencer à utiliser AWS DevOps Agent

Dans ce guide de démarrage, vous allez créer un espace agent de base, configurer des autorisations minimales et mener votre première enquête basée sur l'IA.

Rubriques :

Création d'un espace d'agents

Un espace d'agent définit les outils et l'infrastructure auxquels AWS DevOps l'agent a accès. Ce guide explique comment créer un espace agent, configurer l'accès au compte principal et activer l'application Web DevOps Agent. Consultez la section « Qu'est-ce qu'un espace agent » pour en savoir plus sur le concept d'espace agent.

Création d'un espace d'agents

Accédez à la console de AWS DevOps l'agent

1. Connectez-vous à la console AWS de gestion
2. Accédez à la console de AWS DevOps l'agent

Nommez l'agent Space

1. Cliquez sur Créer un espace d'agent

Dans la section Détails de l'espace agent, indiquez :

1. Dans le champ Nom, saisissez le nom de votre espace agent

2. (Facultatif) Dans le champ Description, ajoutez des détails sur l'objectif de l'espace agent
3. (Facultatif) Dans le menu déroulant Langue de réponse de l'agent, sélectionnez la langue utilisée par l'agent lors de la génération des réponses, des conclusions et des résultats d'enquête. Les options incluent : bahasa indonésien, chinois (Simplified/PRC), Chinese (Traditional/Taiwan), anglais (Royaume-Uni), français (France), allemand (Allemagne), italien (Italie), japonais (Japon), coréen (Corée), portugais (Brésil), espagnol (Amérique latine), turc (Turquie), arabe (Arabie Saoudite), thaï (Thaïlande) et vietnamien (Vietnam). Si aucune langue n'est sélectionnée, l'agent répond dans la langue de saisie.

Configuration de l'accès au compte principal

Dans la section Accorder l'accès aux AWS ressources de cet espace d'agent, vous allez configurer un rôle IAM pour accorder à cet espace d'agent l'accès au AWS compte principal. Le compte principal est le AWS compte sur lequel vous créez votre espace agent. AWS DevOps L'agent a besoin d'un rôle IAM pour découvrir et accéder aux AWS ressources de ce compte pendant les enquêtes.

Choisissez une méthode de configuration des rôles. Sélectionnez l'une des options suivantes :

Option 1 : créer automatiquement un nouveau rôle d' AWS DevOps agent (recommandé)

Cette option crée automatiquement un rôle doté des autorisations appropriées pour que AWS DevOps l'agent puisse examiner les ressources de votre compte.

> **Note**
>
> Pour utiliser cette option, vous devez disposer des autorisations IAM pour créer de nouveaux rôles.

1. Sélectionnez Créer automatiquement un nouveau rôle d' AWS DevOps agent
2. (Facultatif) Mettez à jour le nom du rôle Agent Space à créer

Option 2 : attribuer un rôle existant

Utilisez cette option lorsqu'un autre administrateur a déjà créé un rôle spécifique pour AWS DevOps l'agent.

1. Sélectionnez Attribuer un rôle existant
2. Dans le menu déroulant, sélectionnez un rôle existant doté des autorisations appropriées

Option 3 : créer un nouveau rôle d' AWS DevOps agent à l'aide d'un modèle de politique

Utilisez cette option lorsque vous devez limiter les services et les ressources auxquels l'agent peut accéder dans le compte principal.

1. Sélectionnez Créer un nouveau rôle d' AWS DevOps agent à l'aide d'un modèle de politique
2. Suivez les instructions pour créer la politique de confiance et la politique en ligne du nouveau rôle.

Activation de l'application Web Agent Space

L'application Web est l'endroit où le personnel interagit avec AWS DevOps l'agent pour les enquêtes sur les incidents et l'examen des recommandations. Voir Architecture de la console de l' AWS DevOps agent [lien] pour en savoir plus. Lorsque cette option est activée, les utilisateurs peuvent accéder à l'application Web Agent Space via un lien d'authentification IAM depuis la console de AWS gestion.

Sélectionnez l'une des options suivantes :

Option 1 : créer automatiquement un nouveau rôle d' AWS DevOps agent (recommandé)

Cette option crée automatiquement un rôle doté des autorisations appropriées pour accéder à l'application Web de l' DevOps agent.

> **Note**
>
> Pour utiliser cette option, vous devez disposer des autorisations IAM pour créer de nouveaux rôles.

1. Sélectionnez Créer automatiquement un nouveau rôle d' AWS DevOps agent
2. Vérifiez les autorisations qui seront accordées au rôle

Option 2 : attribuer un rôle existant

Utilisez cette option lorsqu'un autre administrateur a déjà créé un rôle d'opérateur.

1. Sélectionnez Attribuer un rôle existant
2. Dans le menu déroulant, sélectionnez un rôle existant doté des autorisations appropriées

Option 3 : créer un nouveau rôle d' AWS DevOps agent à l'aide d'un modèle de politique

Utilisez cette option lorsque vous devez personnaliser les autorisations d'accès aux applications Web.

1. Sélectionnez Créer un nouveau rôle d' AWS DevOps agent à l'aide d'un modèle de politique
2. Suivez les instructions pour créer la politique de confiance et la politique en ligne du nouveau rôle.

Ajouter des tags (facultatif)

Vous pouvez ajouter des AWS tags à votre espace agent lors de la création. Les balises sont des paires clé-valeur qui vous aident à organiser et à identifier vos ressources. Vous pouvez ajouter jusqu'à 50 balises par agent Space. Pour ajouter des tags, développez la section Tags sur la page Create Agent Space et cliquez sur Ajouter un nouveau tag.

Création complète d'un espace d'agent

Une fois toutes les sections remplies, cliquez sur Créer

Vérification de la configuration de votre espace agent

Une fois configuré, le bouton d'accès de l'opérateur apparaît sur la page de détails de l'espace agent. Cliquez dessus pour ouvrir l'application Web dans un nouvel onglet et vous authentifier avec succès.

Étapes suivantes

Après avoir configuré votre espace agent, considérez les étapes suivantes :

- Ajoutez des comptes secondaires si vos applications couvrent plusieurs AWS comptes
- Configurer des intégrations tierces telles que des outils d'observabilité ou des systèmes de billetterie
- Configurer AWS l'authentification Identity Center pour les environnements de production
- Explorez le mappage des ressources de votre application pour aider AWS DevOps l'agent à comprendre votre infrastructure

AWS DevOps Guide d'intégration de l'Agent CLI

Présentation de

Avec AWS DevOps Agent, vous pouvez surveiller et gérer votre AWS infrastructure. Ce guide explique comment configurer l' AWS DevOps agent à l'aide de l'interface de ligne de AWS commande (AWS CLI). Vous créez des rôles IAM, configurez un espace d'agent et associez votre AWS compte. Vous activez également l'application opérateur et connectez éventuellement des intégrations tierces. Il faut environ 20 minutes pour compléter ce guide.

AWS DevOps L'agent est disponible dans six AWS régions : États-Unis Est (Virginie du Nord), États-Unis Ouest (Oregon), Asie-Pacifique (Sydney), Asie-Pacifique (Tokyo), Europe (Francfort) et Europe (Irlande). Pour plus d'informations sur les régions prises en charge, consultezthe section called "Régions prises en charge".

Conditions préalables

Avant de commencer, assurez-vous de disposer des éléments suivants :

- AWS CLI version 2 installée et configurée
- Authentification auprès de votre compte AWS de surveillance
- Autorisations permettant de créer des rôles AWS Identity and Access Management (IAM) et d'associer des politiques
- Un AWS compte à utiliser comme compte de surveillance
- Connaissance de la AWS CLI et de la syntaxe JSON

Tout au long de ce guide, remplacez les valeurs d'espace réservé suivantes par les vôtres :

- `<MONITORING_ACCOUNT_ID>`— Votre identifiant de AWS compte à 12 chiffres pour le compte de surveillance (principal)
- `<EXTERNAL_ACCOUNT_ID>`— L'identifiant de AWS compte à 12 chiffres du compte secondaire à surveiller (utilisé à l'étape 4)
- `<REGION>`— Le code de AWS région de votre espace d'agent (par exemple, `us-east-1` ou`eu-central-1`)
- `<AGENT_SPACE_ID>`— L'identifiant de l'espace agent renvoyé par la `create-agent-space` commande

Configuration des rôles IAM

1. Création du rôle DevOps d'espace Agent

Créez la politique de confiance IAM en exécutant la commande suivante :

```
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
```

Créez le rôle IAM :

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

Enregistrez l'ARN du rôle en exécutant la commande suivante :

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

Joignez la politique AWS gérée :

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

Créez et attachez une politique en ligne pour autoriser la création du rôle lié au service 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. Création du rôle IAM de l'application opérateur

Créez la politique de confiance IAM en exécutant la commande suivante :

```
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
```

Créez le rôle IAM :

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

Enregistrez l'ARN du rôle en exécutant la commande suivante :

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

Joignez la politique de l'application AWS gérée pour les opérateurs :

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

Cette politique gérée accorde à l'application de l'opérateur les autorisations nécessaires pour accéder aux fonctionnalités de l'espace agent. Ces fonctionnalités incluent les enquêtes, les

recommandations, la gestion des connaissances, le chat et l'intégration du AWS Support. La politique définit l'accès à l'espace d'agent spécifique en utilisant la `aws:PrincipalTag/AgentSpaceId` condition. Pour plus d'informations sur la liste complète des actions, consultezthe section called "DevOps Autorisations IAM de l'agent".

Étapes d'intégration

1. Création d'un espace d'agent

Exécutez la commande suivante pour créer un espace d'agent :

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

Spécifiez éventuellement `--kms-key-arn` l'utilisation d'une clé AWS KMS gérée par le client pour le chiffrement. Vous pouvez également les utiliser `--tags` pour ajouter des balises de ressources et `--locale` définir la langue des réponses des agents.

Enregistrez le `agentSpaceId` depuis la réponse (située à l'adresse`agentSpace.agentSpaceId`).

Pour répertorier vos espaces d'agent ultérieurement, exécutez la commande suivante :

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

2. Associez votre AWS compte

Associez votre AWS compte pour activer la découverte de topologies. Définissez `accountType` l'une des valeurs suivantes :

- `monitor`— Le compte principal sur lequel se trouve l'espace agent. Ce compte héberge l'agent et est utilisé pour la découverte de la topologie.
- `source`— Un compte supplémentaire surveillé par l'agent. Utilisez ce type lorsque vous associez des comptes externes à l'étape 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. Activez l'application pour opérateurs

Les flux d'authentification peuvent utiliser IAM, IAM Identity Center (IDC) ou un fournisseur d'identité externe (IdP). Exécutez la commande suivante pour activer l'application opérateur pour votre espace agent :

```
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>
```

Pour l'authentification IAM Identity Center, utilisez `--auth-flow idc` et fournissez`--idc-instance-arn`. Pour un fournisseur d'identité externe, utilisez `--auth-flow idp` et fournissez `--issuer-url--idp-client-id`, et`--idp-client-secret`. Pour plus d'informations, consultez the section called "Configuration de l'authentification IAM Identity Center" et the section called "Configuration de l'authentification par fournisseur d'identité externe (IdP)".

> Remarque : Si vous avez déjà créé un rôle d'application opérateur pour un autre espace agent de votre compte, vous pouvez réutiliser l'ARN de ce rôle.

4. (Facultatif) Associer des comptes sources supplémentaires

Pour surveiller des comptes supplémentaires avec AWS DevOps l'Agent, créez un rôle multi-comptes IAM.

Création du rôle multi-comptes dans le compte externe

Passez au compte externe et créez la politique de confiance. `MONITORING_ACCOUNT_ID`Il s'agit du compte principal qui héberge l'espace d'agent que vous avez configuré à l'étape 2. Cette

configuration permet au service AWS DevOps Agent d'assumer un rôle dans les comptes sources secondaires au nom du compte de surveillance.

Exécutez la commande suivante pour créer la politique de confiance :

```
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
```

Créez le rôle IAM entre comptes :

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

Enregistrez l'ARN du rôle en exécutant la commande suivante :

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

Joignez la politique AWS gérée :

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

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

Joignez la politique intégrée pour autoriser la création du rôle lié au service Resource Explorer dans le compte externe :

```
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
```

Associer le compte externe

Revenez à votre compte de surveillance, puis exécutez la commande suivante pour associer le compte externe :

```
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. (Facultatif) Associer GitHub

> Remarque : vous devez d'abord vous enregistrer GitHub via la console de l' AWS DevOps agent en utilisant le OAuth flux avant de pouvoir l'associer via la CLI.

Pour obtenir des instructions sur l'enregistrement GitHub via la console, consultezthe section called "Raccordement aux CI/CD pipelines".

Répertoriez les services enregistrés :

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

Enregistrez le <SERVICE_ID> pour `github` ServiceType :.

Après vous être enregistré GitHub dans la console, associez GitHub des référentiels en exécutant la commande suivante :

```
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. (Facultatif) Inscrivez-vous et associez-vous ServiceNow

Tout d'abord, enregistrez le ServiceNow service avec des OAuth informations d'identification :

```
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>
```

Enregistrez le résultat<SERVICE_ID>, puis associez 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. (Facultatif) Enregistrez et associez Dynatrace

Tout d'abord, enregistrez le service Dynatrace avec des informations d'identification : OAuth

```
aws devops-agent register-service \
  --service dynatrace \
  --service-details '{
  "dynatrace": {
    "accountUrn": "<DYNATRACE_ACCOUNT_URN>",
    "authorizationConfig": {
        "oAuthClientCredentials": {
            "clientName": "<DYNATRACE_CLIENT_NAME>",
            "clientId": "<DYNATRACE_CLIENT_ID>",
            "clientSecret": "<DYNATRACE_CLIENT_SECRET>"
        }
      }
```

```
    }
  }' \
  --region <REGION>
```

Enregistrez le résultat<SERVICE_ID>, puis associez Dynatrace. Les ressources sont facultatives. L'environnement indique à quel environnement Dynatrace doit être associé.

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

La réponse inclut des informations sur le webhook pour l'intégration. Vous pouvez utiliser ce webhook pour déclencher une enquête auprès de Dynatrace. Pour de plus amples informations, veuillez consulter the section called “Connecter Dynatrace”.

8. (Facultatif) Enregistrez et associez Splunk

Enregistrez d'abord le service Splunk avec des BearerToken informations d'identification.

Le point de terminaison utilise le format suivant : `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>
```

Enregistrez le résultat`<SERVICE_ID>`, puis associez Splunk :

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

La réponse inclut des informations sur le webhook pour l'intégration. Vous pouvez utiliser ce webhook pour déclencher une enquête depuis Splunk. Pour de plus amples informations, veuillez consulter the section called “Connecter Splunk”.

9. (Facultatif) Enregistrez et associez New Relic

Enregistrez d'abord le service New Relic avec les informations d'identification clés de l'API.

Région : L'un `US` ou l'autre`EU`.

Champs facultatifs :`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>
```

Enregistrez le résultat`<SERVICE_ID>`, puis associez New Relic :

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

La réponse inclut des informations sur le webhook pour l'intégration. Vous pouvez utiliser ce webhook pour lancer une enquête auprès de New Relic. Pour de plus amples informations, veuillez consulter the section called “Connecter New Relic”.

10. (Facultatif) Enregistrez et associez Datadog

Vous devez d'abord enregistrer Datadog via la console de l' AWS DevOps Agent en utilisant le OAuth flux avant de pouvoir l'associer via la CLI. Pour de plus amples informations, veuillez consulter the section called “Connecter DataDog”.

Répertoriez les services enregistrés :

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

Enregistrez le `<SERVICE_ID>` pour `mcpserverdatadog` ServiceType :.

Associez ensuite Datadog :

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

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

La réponse inclut des informations sur le webhook pour l'intégration. Vous pouvez utiliser ce webhook pour déclencher une enquête depuis Datadog. Pour de plus amples informations, veuillez consulter the section called “Connecter DataDog”.

11. (Facultatif) Supprimer un espace d'agent

La suppression d'un espace d'agent entraîne la suppression de toutes les associations, configurations et données d'investigation associées à cet espace d'agent. Cette action ne peut être annulée.

Pour supprimer un espace d'agent, exécutez la commande suivante :

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

Vérification

Pour vérifier votre configuration, exécutez les commandes suivantes :

```
# 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>
```

Étapes suivantes

- Pour connecter des intégrations supplémentaires, voirConfiguration des fonctionnalités de AWS DevOps l'agent.
- Pour en savoir plus sur les compétences et les capacités des agents, voirthe section called "DevOps Compétences des agents".
- Pour comprendre l'application Web destinée aux opérateurs, voirthe section called "Qu'est-ce qu'une application Web pour DevOps agents ?".

Remarques

- Remplacez `<AGENT_SPACE_ID><MONITORING_ACCOUNT_ID>,<EXTERNAL_ACCOUNT_ID>,<REGION>`, et ainsi de suite par vos valeurs réelles.
- Pour obtenir une liste des régions prises en charge, consultez the section called "Régions prises en charge".

Création d'un environnement de test

Ce guide propose des tests pratiques pour valider la fonctionnalité de réponse aux incidents de l' AWS DevOps agent à l'aide d'un exemple d'architecture. Utilisez ce supplément si vous souhaitez tester l' DevOps agent avant de connecter vos systèmes de production.

Conditions préalables

- AWS compte avec accès administratif
- AWS DevOps Espace d'agent créé et configuré à l'aide du flux de rôles d' DevOps agent Auto Create

Vue d'ensemble des coûts et de la sécurité

Protection des coûts

- Test EC2 : GRATUIT (niveau AWS gratuit) ou ~0,02 $ pour 2 heures
- Test Lambda : GRATUIT (niveau gratuit de 1 million requests/month)

- CloudWatch: GRATUIT (10 alarmes, mesures de base incluses)
- Coût total estimé prévu : 0,00$ - 0,05$ pour un test complet

Caractéristiques de sécurité de ces tests

- Arrêt automatique : arrêt automatique intégré
- Éligibilité au niveau gratuit : utilise les plus petits types d'instances
- Champ d'application limité : ressources de test isolées et minimales
- Nettoyage facile : étapes simples sur console pour tout retirer
- Aucun impact sur la production : environnement de test complètement séparé

Configurez votre AWS compte pour les tests

> **Important**
>
> Les ressources d'infrastructure doivent être déployées dans le AWS compte sur lequel vous avez créé le compte cloud principal de votre DevOps Agent Space. La région spécifique n'a pas d'importance.

1. Connectez-vous à AWS la console : https://console.aws.amazon.com
2. Assurez-vous que vous travaillez sur le même AWS compte que celui où se trouve votre espace d' DevOps agent
3. Vous pouvez utiliser n'importe quelle région pour vos ressources de test

> **Note**
>
> Le mappage 1:1 entre le compte principal de votre DevOps agent et les ressources de l'environnement de test que vous créez simplifie la configuration des tests. Vous pouvez facilement étendre votre espace d' DevOps agent pour inclure des comptes secondaires et permettre des enquêtes entre comptes.

Choisissez votre test

Vous pouvez exécuter l'un des tests indépendamment ou les deux ensemble :

Option de test A : test de capacité du processeur EC2

Objectif : valider la capacité de l' AWS DevOps agent à détecter et à étudier les problèmes de performance EC2

Temps estimé : 5 minutes de configuration+10 minutes d'exécution automatique

Difficulté : Entièrement automatisé (aucune étape manuelle requise)

Option de test B : test du taux d'erreur Lambda

Objectif : Valider la capacité de l' AWS DevOps agent à détecter et à étudier les erreurs liées aux fonctions Lambda

Temps estimé : 10 minutes de configuration + 2 minutes de déclenchement

Difficulté : Très facile

Option de test A : test de capacité du processeur EC2

Étape 1 : Déployer la CloudFormation pile pour le test EC2

Nous les utiliserons CloudFormation pour créer nos ressources de test, ce qui permettra à AWS DevOps l'agent de les suivre et de les étudier correctement.

1. Naviguez vers CloudFormation :
   a. Dans AWS la console, recherchez « CloudFormation » et cliquez sur CloudFormation
   b. Cliquez sur Créer une pile > Avec de nouvelles ressources (standard)
2. Téléchargez le modèle :
   a. Créez un nouveau fichier local appelé `AWS-DevOpsAgent-ec2-test.yaml`
   b. Copiez et collez ce CloudFormation modèle dans le fichier :
      i.
```
AWSTemplateFormatVersion: '2010-09-09'
Description: 'AWS DevOps Agent EC2 CPU Test Stack'
Parameters:
  MyIP:
    Type: String
```

```
    Description: Your current IP address for SSH access (find at https://
whatismyipaddress.com)
    Default: '0.0.0.0/0'
Resources:
  # Security Group for SSH access
  TestSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupName: AWS-DevOpsAgent-test-sg
      GroupDescription: AWS DevOps Agent beta testing security group
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 22
          ToPort: 22
          CidrIp: !Ref MyIP
          Description: SSH access from your IP
      Tags:
        - Key: Name
          Value: AWS-DevOpsAgent-Test-SG
        - Key: Purpose
          Value: AWS-DevOpsAgent-Testing
  # Key Pair for SSH access
  TestKeyPair:
    Type: AWS::EC2::KeyPair
    Properties:
      KeyName: AWS-DevOpsAgent-test-key
      KeyType: rsa
      Tags:
        - Key: Name
          Value: AWS-DevOpsAgent-Test-Key
        - Key: Purpose
          Value: AWS-DevOpsAgent-Testing
  # EC2 Instance for CPU testing
  TestInstance:
    Type: AWS::EC2::Instance
    Properties:
      InstanceType: t3.micro
      ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-
kernel-6.1-x86_64}}'
      KeyName: !Ref TestKeyPair
      SecurityGroupIds:
        - !Ref TestSecurityGroup
      UserData:
        Fn::Base64: !Sub |
```

```
          #!/bin/bash
          yum update -y
          yum install -y htop

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

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

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

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

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

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

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

          # Create auto-shutdown script (safety mechanism)
```

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

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

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

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

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

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

c. Dans la CloudFormation console, sélectionnez Télécharger un fichier modèle

d. Cliquez sur Choisir un fichier

e. Sélectionnez le `AWS-DevOpsAgent-ec2-test.yaml` fichier

f. Cliquez sur Suivant

3. Configurer la pile :

   a. Nom de la pile : `AWS-DevOpsAgent-EC2-Test`

   b. Paramètres :

      i. MyIP : Laisser par défaut `0.0.0.0/0` (vous pouvez le sécuriser ultérieurement si nécessaire)

   c. Cliquez sur Suivant

4. Configurez les options de pile :

   a. Laissez les valeurs par défaut, cliquez sur Suivant

5. Vérifiez et créez :

   a. Vérifiez que je reconnais que cela AWS CloudFormation peut créer des ressources IAM

   b. Cliquez sur Soumettre

6. Attendez la fin :

   a. La création d'une pile prend 3 à 5 minutes

   b. Le statut passera de `CREATE_IN_PROGRESS` à `CREATE_COMPLETE`

   c. Important : votre instance EC2 fait désormais partie d'une CloudFormation pile AWS DevOpsAgent capable de suivre !

Facultatif : accès SSH sécurisé (uniquement si vous prévoyez de vous connecter à l'instance)

Ignorez cette étape si vous souhaitez simplement exécuter le test automatique

1. Accédez aux groupes de sécurité EC2 :
   a. Dans AWS la console, accédez à EC2 → Groupes de sécurité
   b. Trouvez `AWS-DevOpsAgent-test-sg`
2. Mettre à jour la règle SSH :
   a. Sélectionnez le groupe de sécurité → onglet Règles entrantes → Modifier les règles entrantes
   b. Trouvez la règle SSH (port 22)
   c. Changez la source `0.0.0.0/0` de votre adresse IP : `[YOUR_IP]/32`
   d. Obtenez votre adresse IP auprès de https://whatismyipaddress.com
   e. Cliquez sur Enregistrer les règles

Étape 2 : Attendre l'exécution automatique du test

1. Exécution automatique des tests :
   - Le test de stress du processeur démarre automatiquement 5 minutes après le lancement de l'instance
   - Aucune intervention manuelle requise : attendez, le test s'exécute complètement en arrière-plan
2. Surveillez le test :
   - L'instance démarre et prépare le test automatiquement
   - Le script s'exécutera pendant 5 minutes et générera une utilisation du processeur supérieure à 70 %
   - CloudWatch l'alarme devrait se déclencher dans un délai de 8 à 10 minutes au total (délai de 5 minutes + 3 à 5 minutes pour l'alarme)
3. Facultatif : réexécution manuelle (pour des tests supplémentaires) :
   - Connectez-vous à votre instance : console EC2 → → Connect **`AWS-DevOpsAgent-Test-Instance`** → Gestionnaire de session
   - Réexécutez le test de stress : `./cpu-stress-test.sh`
   - Parfait pour tester AWS DevOpsAgent la réponse à plusieurs reprises

Option de test B : test du taux d'erreur Lambda

Étape 1 : Déployer la CloudFormation pile pour le test Lambda

1. Naviguez vers CloudFormation :

a. Dans AWS la console, accédez à CloudFormation

b. Cliquez sur Créer une pile → Avec de nouvelles ressources (standard)

2. Téléchargez le modèle :

a. Créez un nouveau fichier local appelé `AWS-DevOpsAgent-lambda-test.yaml`

b. Copiez et collez ce CloudFormation modèle dans le fichier :

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. Dans la CloudFormation console, sélectionnez Télécharger un fichier modèle

d. Cliquez sur Choisir un fichier

e. Sélectionnez le `AWS-DevOpsAgent-lambda-test.yaml` fichier

f. Cliquez sur Suivant

3. Configurer la pile :

a. Nom de la pile : `AWS-DevOpsAgent-Lambda-Test`

b. Cliquez sur Suivant

4. Configurez les options de pile :

a. Laissez les valeurs par défaut, cliquez sur Suivant

5. Vérifiez et créez :

a. Vérifiez que je reconnais que cela AWS CloudFormation peut créer des ressources IAM

b. Cliquez sur Soumettre

6. Attendez la fin :

a. La création d'une pile prend 2 à 3 minutes

b. Le statut passera à `CREATE_COMPLETE`

Étape 2 : Déclencher des erreurs Lambda

1. Accédez à la console Lambda :

a. Accéder à la console AWS Lambda

b. Trouvez votre fonction `AWS-DevOpsAgent-test-lambda`

2. Testez la fonction :
   a. Cliquez sur l'onglet Test
   b. Cliquez sur Créer un nouvel événement
   c. Nom de l'événement : `AWS-DevOpsAgent-test-event`
   d. Utilisez cette charge utile JSON :
      i.
```
{
"test": "AWS DevOpsAgent validation",
"timestamp": "2024-01-01T00:00:00Z"
}
```
   e. Cliquez sur Enregistrer
3. Générer des erreurs :
   a. Cliquez sur le bouton Test 3 fois (attendez 10 secondes entre chaque)
   b. Chaque test générera une erreur intentionnelle
   c. CloudWatch l'alarme devrait se déclencher dans les 2 à 3 minutes
   d. AWS DevOpsAgentdevrait désormais être en mesure de détecter l'alarme grâce à une investigation dans l'application Operator que vous allez configurer ensuite.

Valider la détection des AWS DevOps agents

Étape 1 : CloudWatch alarmes de contrôle d'intégrité (en option)

Cette étape permet de s'assurer que les tests ci-dessus sont maintenant en état d'alarme.

Pour le test EC2 :

- Dans CloudWatch la console, accédez à Alarmes
- Attendez 3 à 5 minutes après le début du test de stress
- Votre alarme devrait afficher « En état d'alarme »
- Si le message est toujours « OK » : attendez encore 2 à 3 minutes (CloudWatch les mesures peuvent être retardées)

Pour le test Lambda :

- Vérifiez l'`AWS-DevOpsAgent-Lambda-Error-Test`alarme
- Devrait s'afficher en alarme dans les 2 à 3 minutes suivant l'exécution des tests

Étape 2 : Lancer une enquête sur un AWS DevOps agent

1. Ouvrez votre AWS DevOps agent AgentSpace
2. Cliquez sur Accès administrateur. Cela ouvrira l'application Web DevOps Agent Space dans une nouvelle fenêtre
3. Cliquez sur le bouton Démarrer l'enquête sur le côté droit de l'écran
4. Complétez le formulaire suivant :
   a. Détails de l'enquête : Décrivez l'enquête que vous souhaitez mener. Incluez tous les détails possibles sur les objectifs de l'enquête, les domaines à explorer ou les informations pertinentes.
   b. Point de départ de l'enquête : Décrivez les informations à partir desquelles vous souhaitez démarrer l'enquête. Vous pouvez mentionner une alarme, une métrique, un extrait de journal ou tout autre élément pour donner à l' DevOps Agent un point de départ à partir duquel travailler. Dans ce cas, fournissez un résumé des alarmes que vous venez de créer.
   c. Date et heure de l'incident (norme ISO 8601 préférée) ::MMZ YYYY-MM-DDTHH
   d. Donnez un nom à votre enquête : exemple : `Oncall_investigation_1:2025-10-27`
   e. AWS Numéro de compte associé à l'incident
   f. Région où l'incident s'est produit
   g. Priorité : AWS DevOpsAgent permet de mener deux enquêtes simultanées. La Priorité vous permet de définir l'ordre d'exécution de vos enquêtes.
5. Cliquez sur Enquêter pour lancer l'enquête.
6. Cliquez sur votre enquête répertoriée dans le tableau de bord. Vous serez redirigé vers l'écran Détails de l'enquête où vous pourrez voir les étapes détaillées suivies par DevOps l'agent.

Résultats attendus

Résultats du test EC2 :

- Détecte l'alarme du processeur EC2
- Identifie la cause première : « Charge de travail liée aux tests de stress du processeur »
- Affiche la chronologie : Stress test → CPU spike → Alarm

- Fournit des recommandations pour la surveillance et le dimensionnement

Résultats du test Lambda :

- Détecte le pic du taux d'erreur Lambda
- Identifie la cause première : « Exceptions de test intentionnelles »
- Affiche la chronologie : Invocations de fonctions → Erreurs → Alarme
- Fournit des recommandations pour la gestion et la surveillance des erreurs

Instructions de nettoyage

Test de nettoyage A (test EC2)

Nettoyage automatique

- L'instance se terminera automatiquement au bout de 2 heures (intégrée au CloudFormation modèle)

Nettoyage manuel (immédiat)

1. Supprimer la CloudFormation pile :
   a. Accéder à la CloudFormation console
   b. Sélectionnez une `AWS-DevOpsAgent-EC2-Test` pile
   c. Cliquez sur Supprimer
   d. Confirmer la suppression
   e. Cela supprimera automatiquement toutes les ressources : instance EC2, groupe de sécurité, paire de clés et alarme CloudWatch

Test de nettoyage B (test Lambda)

1. Supprimer la CloudFormation pile :
   a. Accéder à la CloudFormation console
   b. Sélectionnez une `AWS-DevOpsAgent-Lambda-Test` pile
   c. Cliquez sur Supprimer

d. Confirmer la suppression

e. Cela supprimera automatiquement toutes les ressources : fonction Lambda, rôle IAM et alarme CloudWatch

Résolution des problèmes

Problèmes courants

« Impossible de se connecter à l'instance EC2 »

- Vérifiez le groupe de sécurité : assurez-vous que le SSH (port 22) est ouvert à votre adresse IP
- Vérifier les autorisations clés : Exécuter `chmod 400 AWS-DevOpsAgent-test-key.pem`
- Vérifier l'adresse IP publique : l'instance doit avoir une adresse IP publique attribuée
- Attendre l'instance : assurez-vous que l'instance est en état « En cours d'exécution »

« L'alarme ne se déclenche pas »

- Attendre les mesures : CloudWatch les mesures peuvent prendre de 2 à 5 minutes pour apparaître
- Vérifiez la charge du processeur : SSH vers l'instance et exécutez pour vérifier que `top` le processeur est supérieur à 70 %
- Vérifier le test de résistance : exécuter `ps aux | grep yes` pour voir si les processus de chargement sont en cours d'exécution
- Attente prolongée : le déclenchement de la première alarme prend parfois jusqu'à 7 à 8 minutes

Validation des tests

Le test de votre AWS DevOp agent est réussi lorsque :

Validation technique

- Précision de l'enquête : les résultats du test EC2 doivent correctement indiquer que l'alarme a été déclenchée en raison de la charge du processeur. Le résultat du test Lambda devrait indiquer qu'il s'agissait d'une défaillance intentionnelle.
- Précision de la chronologie : séquence correcte des événements affichée
- Qualité des recommandations : suggestions exploitables fournies

Commencer à utiliser l' AWS DevOps agent à l'aide du AWS CDK

Présentation de

Ce guide explique comment utiliser le AWS Cloud Development Kit (AWS CDK) pour créer et déployer des ressources d' AWS DevOps agent. L'application AWS CDK automatise la création d'un espace d'agents, de rôles AWS Identity and Access Management (IAM), d'une application d'opérateur et d'associations de comptes via. AWS AWS CloudFormation

L'approche AWS CDK automatise les étapes manuelles décrites dans le guide d'intégration de la CLI en définissant toutes les ressources requises sous forme d'infrastructure sous forme de code.

AWS DevOps L'agent est disponible dans les 6 AWS régions suivantes : USA Est (Virginie du Nord), USA Ouest (Oregon), Asie-Pacifique (Sydney), Asie-Pacifique (Tokyo), Europe (Francfort) et Europe (Irlande). Pour plus d'informations sur les régions prises en charge, consultezthe section called "Régions prises en charge".

Conditions préalables

Avant de commencer, assurez-vous de disposer des éléments suivants :

- AWS Interface de ligne de commande (AWS CLI) installée et configurée avec les informations d'identification appropriées
- Node.js version 18 ou ultérieure
- AWS Interface de ligne de commande (CLI) CDK installée dans le monde entier. Pour installer la CLI AWS CDK, exécutez la commande suivante :

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

- Un AWS compte pour le compte de surveillance (principal)
- (Facultatif) Un deuxième AWS compte si vous souhaitez configurer la surveillance entre comptes

Ce que couvre ce guide

Ce guide est divisé en deux parties :

- Partie 1 — Déployez un espace d'agent avec une application d'opérateur et une AWS association dans votre compte de surveillance. Une fois cette partie terminée, l'agent peut surveiller les problèmes liés à ce compte.
- Partie 2 (facultatif) — Ajoutez une AWS association source pour un compte de service et déployez un rôle IAM entre comptes dans ce compte. Cette configuration permet à l'espace des agents de surveiller les ressources entre les comptes.

Ressources créées

Partie 1 : DevOpsAgentStack (compte de surveillance)

- Rôle IAM (`DevOpsAgentRole-AgentSpace`) : assumé par le service DevOps Agent pour surveiller le compte. Inclut la politique `AIDevOpsAgentAccessPolicy` gérée et une politique en ligne qui permet de créer le rôle lié au service Resource Explorer.
- Rôle IAM (**`DevOpsAgentRole-WebappAdmin`**) : rôle d'application opérateur avec la politique `AIDevOpsOperatorAppAccessPolicy` gérée pour les opérations de l'agent.
- Espace d'agent (`MyCDKAgentSpace`) : espace d'agent central, créé à l'aide de la `AWS::DevOpsAgent::AgentSpace` CloudFormation ressource. Inclut la configuration de l'application pour les opérateurs.
- Association (AWS moniteur) — Lie le compte de surveillance à l'espace des agents en utilisant la `AWS::DevOpsAgent::Association` CloudFormation ressource.
- Association (AWS source) — (Facultatif) Lie le compte de service à l'espace des agents pour la surveillance entre comptes.

Partie 2 : ServiceStack (compte de service, facultatif)

- Rôle IAM (**`DevOpsAgentRole-SecondaryAccount`**) : rôle multicompte avec un nom fixe. Approuvé par l'espace réservé aux agents dans le compte de surveillance. Inclut la politique `AIDevOpsAgentAccessPolicy` gérée et une politique en ligne qui permet de créer le rôle lié au service Resource Explorer.
- Fonction Lambda (`echo-service`) : exemple de service simple qui renvoie les événements d'entrée.

Configuration

Étape 1 : Cloner le référentiel d'échantillons

Exécutez les commandes suivantes pour cloner le référentiel et accéder au répertoire du projet :

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

Étape 2 : Installation des dépendances

Exécutez la commande suivante pour installer les dépendances du projet :

```
npm install
```

Partie 1 : Déploiement de l'espace agent

Dans cette section, vous allez créer l'espace agent, les rôles IAM, l'application opérateur et une AWS association dans votre compte de surveillance.

Étape 1 : configurer l'ID du compte de surveillance

Ouvrez `lib/constants.ts` et définissez l'identifiant de votre compte de surveillance :

L'exemple suivant montre la constante à mettre à jour :

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

Étape 2 : démarrer l'environnement AWS CDK

Si vous n'avez pas démarré le AWS CDK dans votre compte de surveillance, exécutez la commande suivante :

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

Étape 3 : Création et déploiement

Exécutez les commandes suivantes pour créer le TypeScript code et déployer la pile :

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

Étape 4 : Enregistrez les sorties de la pile

Une fois le déploiement terminé, le AWS CDK imprime les sorties de la pile. Enregistrez ces valeurs pour une utilisation ultérieure.

L'exemple suivant montre le résultat attendu :

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

Si vous prévoyez de terminer la partie 2, enregistrez la `AgentSpaceArn` valeur. Vous en avez besoin pour configurer la pile de comptes de service.

Étape 5 : vérifier le déploiement

Pour vérifier que l'espace agent a bien été créé, exécutez la commande AWS CLI suivante :

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

À ce stade, votre espace d'agent est déployé avec l'application opérateur activée et votre compte de surveillance associé. L'agent peut surveiller les problèmes liés à ce compte.

Partie 2 (facultatif) : Ajouter une surveillance entre comptes

Dans cette section, vous étendez la configuration afin que votre espace agent puisse surveiller les ressources d'un deuxième AWS compte (le compte de service). Cela implique deux actions :

1. Ajouter une AWS association source dans le DevOpsAgentStack qui pointe vers le compte de service.

2. Déployer le ServiceStack dans le compte de service avec un rôle IAM qui fait confiance à l'espace de l'agent.

> **Important**
>
> Vous devez terminer la partie 1 avant de continuer. ServiceStack Requiert le résultat `AgentSpaceArn` du DevOpsAgentStack déploiement.

Étape 1 : configurer l'ID du compte de service

Ouvrez `lib/constants.ts` et définissez l'identifiant de votre compte de service :

L'exemple suivant montre la constante à mettre à jour :

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

DevOpsAgentStack crée une AWS association source à l'aide de cet identifiant de compte. Si vous avez déployé le DevOpsAgentStack avant de définir cette valeur, redéployez-le pour créer l'association :

Exécutez les commandes suivantes pour le redéploiement :

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

Étape 2 : définir l'ARN de l'espace agent

Copiez la `AgentSpaceArn` valeur de la DevOpsAgentStack sortie (partie 1, étape 4) et définissez-la dans `lib/constants.ts` :

L'exemple suivant montre la constante à mettre à jour :

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

ServiceStack Utilise cette valeur pour définir la politique de confiance relative au rôle de compte secondaire. Le n' ServiceStack est synthétisé que lorsque cette valeur est définie.

Étape 3 : démarrer le compte de service

Si vous n'avez pas démarré le AWS CDK dans votre compte de service, exécutez la commande suivante :

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

Étape 4 : Déployez le ServiceStack

Exécutez les commandes suivantes pour créer et déployer le ServiceStack en utilisant les informations d'identification du compte de service :

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

Cela crée les ressources suivantes dans le compte de service :

- Un rôle IAM (`DevOpsAgentRole-SecondaryAccount`) qui fait confiance à l'espace des agents dans le compte de surveillance
- Une fonction Echo Lambda (`echo-service`) comme exemple de service

Étape 5 : vérifier le déploiement

Pour vérifier que la fonction Lambda a été déployée avec succès, exécutez les commandes suivantes pour tester le service Echo :

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

Résolution des problèmes

Cette section décrit les problèmes courants et explique comment les résoudre.

CloudFormation type de ressource introuvable

- Vérifiez que vous déployez dans unthe section called "Régions prises en charge".
- Vérifiez que votre AWS CLI est configurée avec les autorisations appropriées.

La création du rôle IAM a échoué

- Vérifiez que votre rôle de déploiement dispose des autorisations nécessaires pour créer des rôles IAM.
- Vérifiez que les conditions de la politique de confiance correspondent à votre numéro de compte.

Le déploiement entre comptes échoue avec le message « Impossible d'assumer le rôle dans le compte cible »

- Chaque pile doit être déployée avec les informations d'identification du compte cible. Utilisez l'`--profile`indicateur pour spécifier le profil de AWS CLI correct.
- Vérifiez que le AWS CDK a été amorcé dans le compte cible.

Retards de propagation de l'IAM

- La propagation des modifications de rôle IAM peut prendre quelques minutes. Si la création de l'espace agent échoue immédiatement après la création du rôle, attendez quelques minutes avant de procéder au redéploiement.

Nettoyage

Pour supprimer toutes les ressources, détruisez les piles dans l'ordre inverse.

Exécutez les commandes suivantes pour détruire les piles :

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

Avertissement : Cette action supprime définitivement votre espace agent et toutes les données associées. Cette action ne peut être annulée. Assurez-vous d'avoir sauvegardé toutes les informations importantes avant de continuer.

Considérations sur la sécurité

- L'application AWS CDK crée des rôles IAM avec des politiques de confiance qui autorisent uniquement le principal du `aidevops.amazonaws.com` service à les assumer.
- Les politiques de confiance incluent des conditions qui limitent l'accès à votre AWS compte spécifique et à l'ARN de votre espace agent.
- Toutes les politiques suivent le principe du moindre privilège. Passez en revue et personnalisez les politiques IAM en fonction des exigences de sécurité de votre organisation.
- Le rôle entre comptes (`DevOpsAgentRole-SecondaryAccount`) utilise un nom fixe et est limité à un ARN d'espace agent spécifique.

Étapes suivantes

Après avoir déployé votre AWS DevOps agent à l'aide du AWS CDK :

1. Découvrez la gamme complète des fonctionnalités de l' DevOps agent dans le guide de l'utilisateur de l'AWS DevOps agent.
2. Envisagez d'intégrer le déploiement du AWS CDK dans vos CI/CD pipelines pour une gestion automatisée de l'infrastructure.

Ressources supplémentaires

- AWS DevOps Guide de l'utilisateur de l'agent
- Exemple de référentiel CDK sur le site Web GitHub
- Guide d'intégration à la CLI

Commencer à utiliser l' AWS DevOps Agent à l'aide de AWS CloudFormation

Présentation de

Ce guide explique comment utiliser des AWS CloudFormation modèles pour créer et déployer des ressources d' AWS DevOps agent. Les modèles automatisent la création d'un espace d'agent, de

rôles AWS Identity and Access Management (IAM), d'une application d'opérateur et d'associations de AWS comptes sous forme d'infrastructure et de code.

L' CloudFormation approche automatise les étapes manuelles décrites dans le guide d'intégration de la CLI en définissant toutes les ressources requises dans des modèles YAML déclaratifs.

AWS DevOps L'agent est disponible dans les 6 AWS régions suivantes : USA Est (Virginie du Nord), USA Ouest (Oregon), Asie-Pacifique (Sydney), Asie-Pacifique (Tokyo), Europe (Francfort) et Europe (Irlande). Pour plus d'informations sur les régions prises en charge, consultezthe section called "Régions prises en charge".

Conditions préalables

Avant de commencer, assurez-vous de disposer des éléments suivants :

- AWS Interface de ligne de commande (AWS CLI) installée et configurée avec les informations d'identification appropriées
- Autorisations pour créer des rôles et CloudFormation des piles IAM
- Un AWS compte pour le compte de surveillance (principal)
- (Facultatif) Un deuxième AWS compte si vous souhaitez configurer la surveillance entre comptes

Ce que couvre ce guide

Ce guide est divisé en deux parties :

- Partie 1 — Déployez un espace d'agent avec une application d'opérateur et une AWS association dans votre compte de surveillance. Une fois cette partie terminée, l'agent peut surveiller les problèmes liés à ce compte.
- Partie 2 (facultatif) — Déployez un rôle IAM entre comptes dans un compte secondaire et ajoutez une association source AWS . Cette configuration permet à l'espace agent de surveiller les ressources entre les comptes.

Partie 1 : Déploiement de l'espace agent

Dans cette section, vous allez créer un CloudFormation modèle qui fournit l'espace des agents, les rôles IAM, l'application opérateur et une AWS association dans votre compte de surveillance.

Étape 1 : Création du CloudFormation modèle

Enregistrez le modèle suivant sous le nom `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
```

Étape 2 : Déployer la pile

Exécutez la commande suivante pour déployer la pile. Remplacez <REGION> par un the section called "Régions prises en charge" (par exemple,`us-east-1`).

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

Étape 3 : Enregistrez les sorties de la pile

Une fois le déploiement terminé, exécutez la commande suivante pour récupérer les sorties de la pile. Enregistrez ces valeurs pour une utilisation ultérieure.

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

L'exemple suivant montre le résultat attendu :

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

Si vous prévoyez de terminer la partie 2, enregistrez la `AgentSpaceArn` valeur. Vous en avez besoin pour configurer le rôle multi-comptes.

Étape 4 : vérifier le déploiement

Pour vérifier que l'espace agent a bien été créé, exécutez la commande AWS CLI suivante :

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

À ce stade, votre espace d'agent est déployé avec l'application opérateur activée et votre compte de surveillance associé. L'agent peut surveiller les problèmes liés à ce compte.

Partie 2 (facultatif) : Ajouter une surveillance entre comptes

Dans cette section, vous étendez la configuration afin que votre espace agent puisse surveiller les ressources d'un deuxième AWS compte (le compte de service). Cela implique deux actions :

1. Déploiement d'un rôle IAM dans le compte de service qui fait confiance à l'espace agent.
2. Ajout d'une AWS association source dans le compte de surveillance pointant vers le compte de service.

> Remarque : Vous devez terminer la partie 1 avant de continuer. Le modèle de compte de service nécessite les sorties `AgentSpaceArn` de la pile de la partie 1.

Étape 1 : Création du modèle de compte de service

Enregistrez le modèle suivant sous`devops-agent-service-account.yaml`. Ce modèle crée un rôle IAM entre comptes dans le compte secondaire.

```
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
```

Étape 2 : Déployer la pile de comptes de service

À l'aide des informations d'identification du compte de service, exécutez la commande suivante :

```
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>
```

Étape 3 : ajouter l' AWS association source

Revenez au compte de surveillance et créez une AWS association de source. Vous pouvez le faire en créant une pile séparée ou en mettant à jour le modèle d'origine. L'exemple suivant utilise un modèle autonome.

Enregistrez le modèle suivant sous le nom `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
```

Déployez la pile d'associations à l'aide des informations d'identification du compte de surveillance :

```
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>
```

Vérification

Vérifiez votre configuration en exécutant les commandes AWS CLI suivantes :

```
# 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>
```

Résolution des problèmes

Cette section décrit les problèmes courants et explique comment les résoudre.

CloudFormation type de ressource introuvable

- Vérifiez que vous déployez dans unthe section called “Régions prises en charge”.
- Vérifiez que votre AWS CLI est configurée avec les autorisations appropriées.

La création du rôle IAM a échoué

- Vérifiez que vos informations d'identification de déploiement sont autorisées à créer des rôles IAM avec des noms personnalisés (`CAPABILITY_NAMED_IAM`).
- Vérifiez que les conditions de la politique de confiance correspondent à votre numéro de compte.

Le déploiement entre comptes échoue

- Chaque pile doit être déployée avec les informations d'identification du compte cible. Utilisez l'`--profile`indicateur pour spécifier le profil de AWS CLI correct.
- Vérifiez que le `AgentSpaceArn` paramètre correspond exactement à l'ARN des sorties de la pile de la partie 1.

Retards de propagation de l'IAM

- La propagation des modifications de rôle IAM peut prendre quelques minutes. Si la création de l'espace agent échoue immédiatement après la création du rôle, attendez quelques minutes avant de procéder au redéploiement.

Nettoyage

Pour supprimer toutes les ressources, supprimez les piles dans l'ordre inverse.

Avertissement : Cette action supprime définitivement votre espace agent et toutes les données associées. Cette action ne peut être annulée. Assurez-vous d'avoir sauvegardé toutes les informations importantes avant de continuer.

Exécutez les commandes suivantes pour supprimer les piles :

```
# 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>
```

Étapes suivantes

Après avoir déployé votre AWS DevOps agent en utilisant AWS CloudFormation :

- Pour connecter des intégrations supplémentaires, voirConfiguration des fonctionnalités de AWS DevOps l'agent.
- Pour en savoir plus sur les compétences et les capacités des agents, voirthe section called "DevOps Compétences des agents".
- Pour comprendre l'application Web destinée aux opérateurs, voirthe section called "Qu'est-ce qu'une application Web pour DevOps agents ?".

Commencer à utiliser l' AWS DevOps agent à l'aide de Terraform

Présentation de

Ce guide vous explique comment utiliser Terraform pour créer et déployer des ressources d' AWS DevOps agent. La configuration Terraform automatise la création d'un espace d'agent, de rôles IAM, d'une application d'opérateur et d'associations de comptes. AWS

L'approche Terraform automatise les étapes manuelles décrites dans le guide d'intégration de la CLI en définissant toutes les ressources requises sous forme d'infrastructure sous forme de code.

AWS DevOps L'agent est disponible dans les 6 AWS régions suivantes : USA Est (Virginie du Nord), USA Ouest (Oregon), Asie-Pacifique (Sydney), Asie-Pacifique (Tokyo), Europe (Francfort) et Europe (Irlande). Pour plus d'informations sur les régions prises en charge, consultezthe section called "Régions prises en charge".

Conditions préalables

Avant de commencer, assurez-vous de disposer des éléments suivants :

- Terraform >= 1.0 installé
- AWS CLI installée et configurée avec les informations d'identification appropriées

- Un AWS compte pour le compte de surveillance (principal)
- (Facultatif) Un deuxième AWS compte si vous souhaitez configurer la surveillance entre comptes

Ce que couvre ce guide

Ce guide est divisé en deux parties :

- Partie 1 — Déployez un espace d'agent avec une application d'opérateur et une AWS association dans votre compte de surveillance. Une fois cette partie terminée, l'agent peut surveiller les problèmes liés à ce compte.
- Partie 2 (facultatif) — Ajoutez une AWS association source pour un compte de service et déployez un rôle IAM entre comptes ainsi qu'un echo Lambda dans ce compte. Cela permet à l'espace des agents de surveiller les ressources entre les comptes.

Ressources créées

Partie 1 : Compte de surveillance

- Rôle IAM (`DevOpsAgentRole-AgentSpace-*`) : assumé par le service DevOps Agent pour surveiller le compte. Inclut la politique `AIDevOpsAgentAccessPolicy` gérée et une politique en ligne qui permet de créer le rôle lié au service Resource Explorer.
- Rôle IAM (**`DevOpsAgentRole-WebappAdmin-*`**) : rôle d'application opérateur avec la politique `AIDevOpsOperatorAppAccessPolicy` gérée pour les opérations des agents.
- Espace d'agent (nom configurable) : espace d'agent central, créé à l'aide de la `awscc_devopsagent_agent_space` ressource. Inclut la configuration de l'application pour les opérateurs.
- Association (AWS moniteur) — Lie le compte de surveillance à l'espace des agents utilisant la `awscc_devopsagent_association` ressource.
- Association (AWS source) — (Facultatif) Lie le compte de service à l'espace des agents pour la surveillance entre comptes.

Partie 2 : Compte de service (facultatif)

- Rôle IAM (**`DevOpsAgentRole-SecondaryAccount-TF`**) : rôle multicompte avec un nom fixe. Approuvé par l'espace réservé aux agents dans le compte de surveillance. Inclut la politique

`AIDevOpsAgentAccessPolicy` gérée et une politique en ligne qui permet de créer le rôle lié au service Resource Explorer.

- Fonction Lambda (`echo-service-tf`) : exemple de service simple qui renvoie les événements d'entrée.

Configuration

Étape 1 : cloner le référentiel d'échantillons

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

Étape 2 : Configuration des variables

Copiez le fichier de variables d'exemple et personnalisez-le en fonction de votre environnement :

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

Modifiez `terraform.tfvars` avec le nom et la description de votre espace agent :

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

Partie 1 : Déploiement de l'espace agent

Dans cette section, vous allez créer l'espace agent, les rôles IAM, l'application opérateur et une AWS association dans votre compte de surveillance.

Étape 1 : déploiement automatique (recommandé)

Utilisez le script de déploiement fourni pour une configuration rationalisée :

```
./deploy.sh
```

Ce script :

- Vérifie les prérequis (Terraform, AWS CLI, informations d'identification)

- Crée `terraform.tfvars` à partir d'un exemple si nécessaire
- Initialise, valide, planifie et applique Terraform

Sinon, si vous préférez le contrôle manuel :

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

Tapez `yes` lorsque vous êtes invité à confirmer le déploiement.

Étape 2 : Enregistrez les sorties

Une fois le déploiement terminé, Terraform imprime les sorties. Enregistrez ces valeurs pour une utilisation ultérieure :

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

Si vous prévoyez de terminer la partie 2, enregistrez la `agent_space_arn` valeur. Vous en aurez besoin pour configurer les ressources du compte de service.

Étape 3 : vérifier le déploiement

Exécutez le script de vérification après le déploiement :

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

Vous pouvez également utiliser la AWS CLI pour vérifier que l'espace agent a été créé avec succès :

```
aws devops-agent get-agent-space \
```

```
  --agent-space-id <AGENT_SPACE_ID> \
  --region <REGION>
```

À ce stade, votre espace d'agent est déployé avec l'application opérateur activée et votre compte de surveillance associé. L'agent peut surveiller les problèmes liés à ce compte.

Partie 2 (facultatif) : Ajouter une surveillance entre comptes

Dans cette section, vous étendez la configuration afin que l'espace agent puisse surveiller les ressources d'un deuxième AWS compte (le compte de service). Cela implique deux actions :

1. Ajout d'une AWS association source pointant vers le compte de service.
2. Déploiement d'un rôle IAM entre comptes et d'une fonction Echo Lambda dans le compte de service.

> **Important**
>
> Vous devez terminer la partie 1 avant de continuer. Les ressources du compte de service nécessitent le résultat `agent_space_arn` de déploiement de la partie 1.

Étape 1 : configurer l'ID du compte de service

Dans`terraform.tfvars`, définissez l'identifiant de votre compte de service :

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

Étape 2 : définir l'ARN de l'espace agent

Copiez la `agent_space_arn` valeur de la sortie de la partie 1 (étape 2) et définissez-la dans `terraform.tfvars` :

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

Les ressources du compte de service utilisent cette valeur pour définir la politique de confiance relative au rôle du compte secondaire. Ces ressources ne sont créées que lorsque cette valeur est définie.

Étape 3 : Configuration du fournisseur `aws.service`

Dans`main.tf`, configurez l'alias du `aws.service` fournisseur avec les informations d'identification du compte de service. Vous pouvez utiliser un profil nommé ou un rôle d'emprunt :

À l'aide d'un profil :

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

Ou en utilisant assume le rôle :

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

Étape 4 : Déploiement

Appliquez la configuration mise à jour :

```
terraform apply
```

Cela crée les ressources suivantes dans le compte de service :

- Un rôle IAM (`DevOpsAgentRole-SecondaryAccount-TF`) qui fait confiance à l'espace des agents dans le compte de surveillance
- Une fonction Echo Lambda (`echo-service-tf`) comme exemple de service

Il crée également une AWS association de source dans le compte de surveillance qui lie le compte de service.

Étape 5 : vérifier le déploiement

Testez le service Echo pour vérifier que la fonction Lambda a été déployée avec succès :

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

Résolution des problèmes

Retards de propagation de l'IAM

- La configuration inclut un délai de 30 secondes `time_sleep` entre la création du rôle IAM et la création de l'espace agent. Le service DevOps Agent valide la politique de confiance du rôle d'opérateur lors de la création de l'espace agent, ce qui peut échouer si IAM ne s'est pas complètement propagé. Si des erreurs liées à la politique de confiance persistent, attendez une minute et `terraform apply` réexécutez. Les rôles IAM existeront déjà et l'application reprendra là où elle s'est arrêtée.

Erreurs d'autorisation

- Vérifiez que vos AWS informations d'identification disposent des autorisations IAM nécessaires pour créer des rôles et des politiques.
- Vérifiez que les conditions de la politique de confiance correspondent à votre numéro de compte.

Le déploiement entre comptes échoue

- Le `aws.service` fournisseur doit être configuré avec les informations d'identification du compte de service. Utilisez un profil nommé ou un bloc de rôle assumé.
- Vérifiez que la `agent_space_arn` valeur correspond à l'ARN de la sortie de la partie 1.

Type de ressource Terraform introuvable

- Vérifiez que vous disposez de la version du `awscc` fournisseur `~> 1.0` ou d'une version ultérieure. Les `awscc_devopsagent_association` ressources `awscc_devopsagent_agent_space` et nécessitent le fournisseur AWS Cloud Control.

Nettoyage

Pour supprimer toutes les ressources, détruisez-les dans l'ordre inverse si vous avez déployé la partie 2 :

```
./cleanup.sh
```

Ou manuellement :

```
terraform destroy
```

Avertissement : Cela supprime définitivement votre espace d'agent et toutes les données associées. Assurez-vous d'avoir sauvegardé toutes les informations importantes avant de continuer.

Considérations sur la sécurité

- La configuration Terraform crée des rôles IAM avec des politiques de confiance qui autorisent uniquement le principal du `aidevops.amazonaws.com` service à les assumer.
- Les politiques de confiance incluent des conditions qui limitent l'accès à votre AWS compte spécifique et à l'ARN de votre espace agent.
- Toutes les politiques suivent le principe du moindre privilège. Passez en revue et personnalisez les politiques IAM en fonction des exigences de sécurité de votre organisation.
- Le rôle entre comptes (`DevOpsAgentRole-SecondaryAccount-TF`) utilise un nom fixe et est limité à un ARN d'espace agent spécifique.

Étapes suivantes

Après avoir déployé votre AWS DevOps agent à l'aide de Terraform :

1. Découvrez la gamme complète des fonctionnalités de l' DevOps agent dans le guide de l'utilisateur de l'AWS DevOps agent.
2. Envisagez d'intégrer le déploiement de Terraform dans vos CI/CD pipelines pour une gestion automatisée de l'infrastructure.

Ressources supplémentaires

- AWS DevOps Guide de l'utilisateur de l'agent

- Exemple de référentiel Terraform
- Guide d'intégration à la CLI

Travailler avec l' DevOps agent

Travailler avec l' DevOps agent

AWS DevOps L'agent travaille aux côtés de votre équipe opérationnelle tout au long du cycle de vie des incidents, de la détection à l'investigation, en passant par le rétablissement et la prévention. Les rubriques suivantes décrivent comment utiliser l' DevOps agent pour gérer chaque phase de ce cycle de vie.

Réponse autonome aux incidents

Lorsqu'un incident est détecté, que ce soit par le biais d'une intégration intégrée à votre système de billetterie, d'un webhook créé par vos outils de surveillance ou d'un déclencheur manuel, DevOps l'agent lance automatiquement une enquête. L'agent analyse les métriques, les journaux, les traces, les modifications de code et l'historique des déploiements afin de déterminer la cause première et de proposer un plan d'atténuation. Si vous avez besoin d'une aide supplémentaire, vous pouvez accéder directement au AWS Support depuis l'application Web DevOps Agent Space, qui partage automatiquement le contexte de l'enquête avec les ingénieurs du support afin que vous n'ayez pas à répéter ce que l'agent a déjà découvert. Pour de plus amples informations, veuillez consulter the section called “Réponse autonome aux incidents”.

DevOps Tâches sur demande

À tout moment du cycle de vie de l'incident, vous pouvez interagir avec DevOps l'agent via une interface de chat conversationnelle. Posez des questions sur vos AWS ressources, l'état de votre système, l'état des alarmes et l'historique des déploiements en langage naturel. Le chat tient compte du contexte : lorsque vous consultez une enquête spécifique, vous pouvez demander à l'agent d'explorer des hypothèses spécifiques, de se concentrer sur des journaux spécifiques ou de mettre à jour son analyse des causes profondes. Vous pouvez également interroger les configurations des ressources, les tendances en matière d'erreurs et les informations d'investigation dans l'ensemble de votre environnement sans devoir passer d'une console à l'autre. Pour de plus amples informations, veuillez consulter the section called “DevOps Tâches à la demande”.

Prévention proactive des incidents

Après avoir résolu les incidents, DevOps l'agent analyse les tendances de l'historique de vos enquêtes afin de générer des recommandations visant à prévenir de futurs incidents et à réduire le délai moyen de détection. Les recommandations portent sur quatre domaines : posture d'observabilité, lacunes dans les tests, modifications du code et architecture de l'infrastructure. L'agent effectue des évaluations chaque semaine et met à jour les recommandations au fur et à mesure que de nouveaux incidents surviennent. Vous pouvez accepter, rejeter ou suivre les recommandations, et l'agent s'appuiera sur vos commentaires pour affiner les suggestions futures. Pour de plus amples informations, veuillez consulter the section called “Prévention proactive des incidents”.

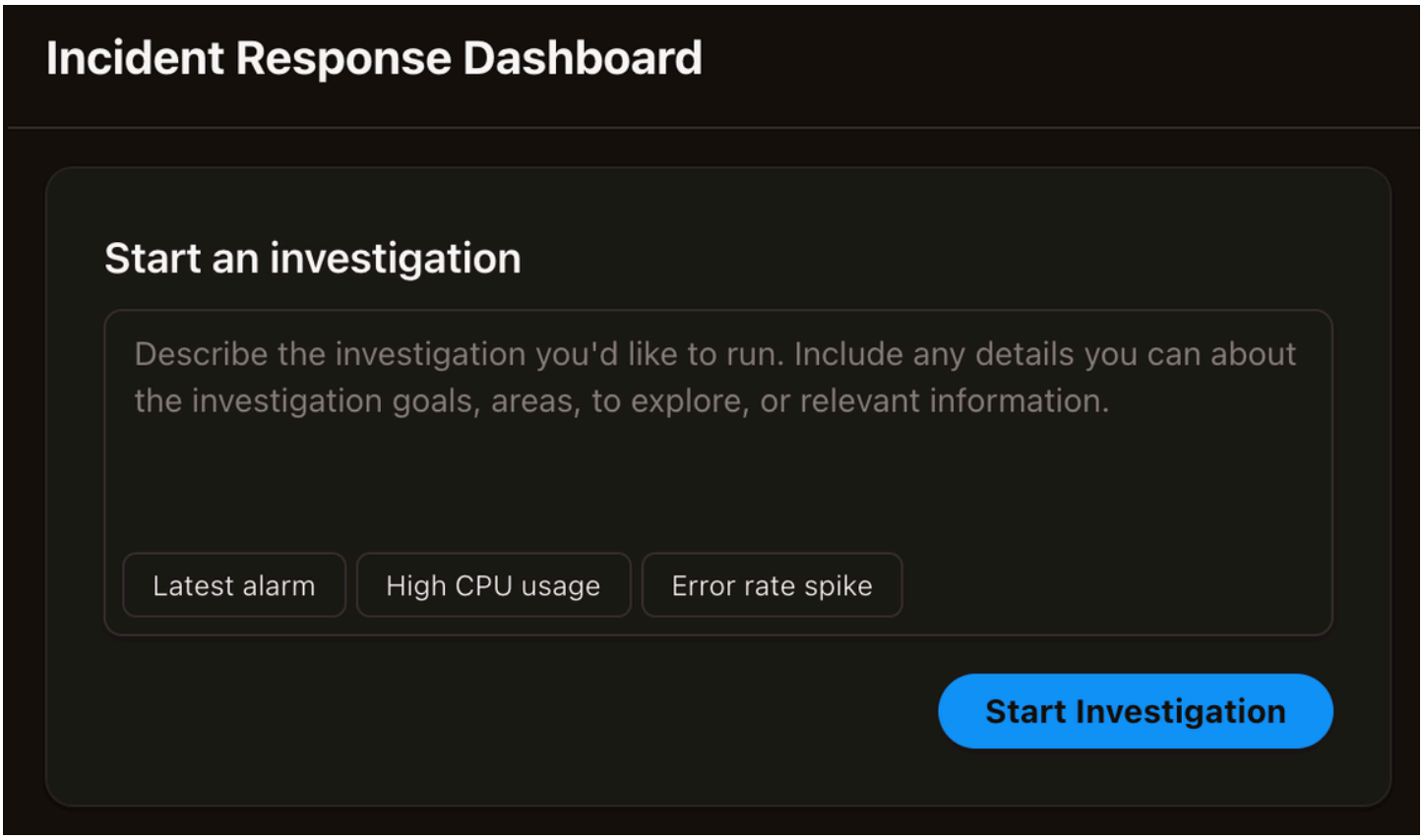
Réponse autonome aux incidents

Commencer les enquêtes

Les enquêtes de réponse aux incidents peuvent être lancées de trois manières différentes.

- Intégrations intégrées - Vous pouvez connecter un espace DevOps agent à des systèmes de billetterie, par exemple ServiceNow en utilisant des intégrations intégrées. Une fois connecté, les enquêtes de réponse aux incidents de l' DevOps agent seront automatiquement déclenchées à partir des tickets d'assistance, et votre DevOps agent fournira des mises à jour de ses principales conclusions, des analyses des causes profondes et des plans d'atténuation dans le ticket d'origine.
- Webhooks - Vous pouvez utiliser des webhooks pour envoyer des événements à l' AWS DevOps Agent. Par exemple, vous pouvez utiliser des webhooks pour déclencher des enquêtes de réponse aux incidents à partir de PagerDuty tickets ou d'alarmes Grafana.
- Manuellement : vous pouvez lancer manuellement des enquêtes sur la réponse aux incidents depuis l'onglet Réponse aux incidents de n'importe quelle application Web DevOps Agent Space. Vous pouvez soit saisir un texte en format libre décrivant l'incident sur lequel vous souhaitez que votre DevOps agent enquête, qui créera un plan d'enquête, collectera les conclusions, en déterminera la cause première et proposera de générer un plan d'atténuation. Vous pouvez également choisir parmi plusieurs points de départ préconfigurés pour démarrer rapidement votre enquête : dernière alarme pour examiner votre dernière alarme déclenchée et analyser les indicateurs et journaux sous-jacents pour en déterminer la cause première, utilisation élevée du processeur pour étudier les indicateurs d'utilisation élevée du processeur sur l'ensemble de vos ressources informatiques et identifier les processus ou services consommant trop de ressources,

ou pic du taux d'erreur pour étudier la récente augmentation du taux d'erreur des applications en analysant les métriques, les journaux des applications et en identifiant la source des défaillances.



Une fois que vous aurez cliqué sur « Démarrer l'enquête », il vous sera demandé de fournir des informations supplémentaires pour aider l'agent à concentrer son travail. La boîte de dialogue d'investigation inclut les champs suivants :

- Détails de l'enquête — Pré-remplis avec votre description. Vous pouvez le modifier pour affiner la portée de l'enquête.
- Point de départ de l'enquête : décrivez éventuellement une alarme, une métrique, un extrait de journal ou un autre point de départ spécifique pour l'agent.
- Date et heure de l'incident : saisie automatique de l'heure actuelle au format UTC. Ajustez si l'incident s'est produit plus tôt.
- Donnez un nom à votre enquête : générée automatiquement avec un horodatage. Vous pouvez le personnaliser (400 caractères maximum).
- Priorité — Sélectionnez la priorité de l'enquête dans la liste déroulante (moyenne est la priorité par défaut).

Passez en revue et ajustez ces champs selon vos besoins, puis cliquez sur « Commencer à étudier... » pour commencer. Vous serez ensuite redirigé vers la page des détails de l'enquête où vous pourrez voir votre DevOps agent en action !

Triage des incidents

La phase de triage est la première étape du système de réponse aux incidents de l' AWS DevOps agent. Lorsqu'un événement externe se déclenche, comme une alarme de Datadog, un ticket d'incident ou un problème de Dynatrace ServiceNow, l' AWS DevOps Agent le traite automatiquement en quelques secondes pour déterminer s'il doit être étudié de manière indépendante ou lié à une enquête existante.

La principale fonction de la phase de triage est la corrélation des incidents, c'est-à-dire l'identification des incidents connexes et leur regroupement en une seule enquête afin d'éviter le double emploi et le gaspillage de ressources. Lorsqu'un nouvel incident arrive, l' AWS DevOps agent l'analyse en même temps que les enquêtes en cours dans un délai rétrospectif (généralement 20 minutes). À l'aide d'une analyse basée sur l'IA, il examine des facteurs tels que les similitudes entre les composants, la région géographique et les modèles temporels afin de déterminer les relations entre les incidents.

AWS DevOps L'agent prend l'une des deux décisions suivantes :

- Lié — Corrèle l'incident avec une enquête en cours et envoie un message directeur à cette enquête avec le contexte du nouvel incident.
- Poursuivre — Planifie une nouvelle enquête indépendante sur l'incident.

Afficher les décisions de triage

Lorsque des incidents sont liés, l'enquête principale reçoit un message de direction contenant les détails de l'incident lié et le raisonnement de corrélation. Sur votre application Web AWS DevOps Agent Space, vous verrez le statut LINKED ainsi qu'un raisonnement de corrélation expliquant pourquoi les incidents ont été liés. L'enquête principale affiche une liste de tous les incidents liés, ce qui vous permet de voir l'ensemble des problèmes connexes étudiés ensemble. Votre système de tickets externe (ServiceNow, PagerDuty, etc.) et votre canal de communication (Slack) recevront une notification indiquant que l'incident est lié, ainsi qu'un raisonnement de corrélation.

Dissociation des incidents et règles de corrélation personnalisées

Si AWS DevOps l'Agent établit une corrélation incorrecte entre les incidents, vous pouvez les dissocier manuellement via l'application Web AWS DevOps Agent Space. Cela reprogrammera

l'incident non lié en tant qu'enquête indépendante. Vous pouvez également fournir des règles de corrélation personnalisées pour guider AWS DevOps l'agent en créant une compétence d' AWS DevOps agent contenant votre logique de corrélation et en l'associant à la phase de triage.

Demandez un soutien humain

AWS DevOps L'agent peut se connecter directement au AWS Support pour rationaliser votre processus de réponse aux incidents. Lorsque vous avez besoin d'une aide supplémentaire de la part du AWS Support, depuis votre application Web DevOps Agent Space, vous pouvez créer des dossiers d'assistance qui partagent automatiquement le contexte de l'enquête avec les ingénieurs du AWS support, réduisant ainsi le temps nécessaire pour expliquer votre problème.

Comment ça marche

Lorsqu'il enquête sur un incident, AWS DevOps l'agent crée un journal complet de son analyse, y compris :

- Résultats de l'enquête sur les causes profondes
- Métriques, journaux et traces analysés
- Modifications du code et historique des déploiements passés en revue
- Mesures correctives recommandées
- Chronologie des événements et du comportement du système

Vous pouvez transmettre votre enquête au AWS Support directement depuis l'application Web AWS DevOps Agent Space. Lorsque vous le faites, AWS DevOps l'Agent transmet automatiquement son journal d'investigation au AWS Support, fournissant à l'ingénieur du support le contexte complet de votre enquête sans que vous ayez à collecter et à expliquer manuellement les détails.

Discuter avec AWS le Support

Une fois que vous avez créé un dossier d'assistance, vous pouvez communiquer avec le AWS support dans une fenêtre de discussion séparée au sein de votre application Web AWS DevOps Agent Space. Cela vous permet de :

- Discutez de votre problème avec les ingénieurs du AWS Support ainsi que du calendrier d'investigation de votre AWS DevOps agent
- Consultez l'analyse automatisée de l' AWS DevOps agent et les conseils d'experts du AWS support dans la même interface

- Partagez facilement des informations supplémentaires ou des éclaircissements selon les besoins

L'expérience de chat permet d'accéder facilement à AWS DevOps l'enquête de votre agent et à la conversation avec le AWS Support, ce qui permet une collaboration et une résolution plus rapides.

Exigences du plan de support

Votre capacité à créer des demandes de support et à interagir avec celles-ci par le biais de AWS DevOps l'Agent dépend de votre plan de AWS support. Consultez le guide de l'utilisateur des plans de Support pour en savoir plus sur vos droits.

Remarque : les clients du support de base ne peuvent pas créer de dossiers de support technique et ne peuvent donc pas transférer les enquêtes des AWS DevOps agents au AWS support. Les clients du support aux développeurs peuvent créer des dossiers via l' AWS DevOps agent, mais doivent se rendre au centre de AWS support pour correspondre avec les ingénieurs du support, car le support aux développeurs n'inclut pas d'assistance par chat. Tous les autres plans peuvent utiliser l'expérience de chat intégrée dans Agent. AWS DevOps Pour plus de détails sur les droits des plans de support, y compris les temps de réponse et la sévérité des cas disponibles, consultez le guide de l'utilisateur des plans de AWS support.

Quelles informations sont partagées avec le AWS Support

Lorsque vous créez un dossier d'assistance depuis l'application Web AWS DevOps Agent Space, les informations suivantes sont automatiquement partagées avec le AWS support :

- Chronologie de l'enquête : enregistrement chronologique de l'analyse de l' AWS DevOps agent
- Informations sur les ressources : AWS Ressources concernées
- Données d'observabilité : mesures, journaux et traces pertinents issus de vos outils de surveillance intégrés
- Changements récents : déploiements de code, modifications de l'infrastructure et mises à jour de configuration
- Tentatives de correction : AWS DevOps agent Actions recommandé
- Évaluation d'impact : portée et gravité de l'incident

Toutes les données partagées avec le AWS Support respectent vos configurations de résidence et de sécurité des AWS données existantes. AWS DevOps L'agent partage uniquement les informations

relatives à votre enquête spécifique et respecte les politiques de gouvernance des données de votre organisation.

Prise en main

Pour utiliser l'intégration du AWS Support de l' AWS DevOps agent :

1. Assurez-vous d'avoir un plan de AWS Support actif.
2. Vérifiez que les autorisations IAM de votre AWS DevOps agent incluent la création de dossiers d'assistance (support :CreateCase, support :DescribeCases).
3. Lorsque AWS DevOps l'agent enquête sur un problème et que vous avez besoin d' AWS assistance, choisissez Demander une assistance humaine depuis votre application Web DevOps Agent Space.
4. Consultez le résumé de l'enquête qui sera communiqué au AWS Support.
5. Sélectionnez la gravité du cas appropriée en fonction des droits auxquels vous avez droit dans le cadre de votre plan de support.
6. Soumettre le dossier - AWS DevOps L'agent inclut automatiquement votre journal d'enquête.

La fenêtre de discussion s'ouvre automatiquement, vous permettant de commencer immédiatement à collaborer avec le AWS Support.

Prévention proactive des incidents

AWS DevOps L'agent analyse les tendances issues de vos enquêtes sur les incidents afin de fournir des recommandations ciblées qui améliorent continuellement votre posture opérationnelle et préviennent de futurs incidents. Accédez à la prévention proactive des incidents via la page Ops Backlog de l'application Web Operator.

Comment fonctionne la prévention proactive des incidents

AWS DevOps L'agent évalue les enquêtes récentes sur les incidents afin d'identifier des améliorations durables afin de prévenir de futurs incidents et d'accélérer le délai moyen de détection (MTTD). L'agent analyse plusieurs incidents afin d'identifier des recommandations susceptibles de prévenir des catégories entières d'incidents à l'avenir, en se concentrant sur les recommandations les plus pertinentes afin de garantir qu'elles sont exploitables.

Par défaut, l'agent exécute automatiquement des évaluations chaque semaine. Vous pouvez suspendre le calendrier si vous préférez exécuter des évaluations uniquement à la demande. Des évaluations manuelles sont toujours disponibles, ce qui est utile lorsqu'une enquête récente justifie une mise en œuvre rapide des améliorations recommandées.

L'agent identifie les améliorations dans quatre catégories, comme indiqué dans le tableau de catégorisation des recommandations sur la page Ops Backlog :

- Observabilité : recommandations pour améliorer la surveillance, les alertes, la journalisation et la visibilité du système afin de détecter les problèmes plus rapidement et avec plus de précision.
- Infrastructure : recommandations pour optimiser la configuration des ressources, le réglage des capacités et la résilience architecturale.
- Gouvernance — Recommandations pour renforcer les processus de déploiement, les améliorations du pipeline, les pratiques de test et les contrôles opérationnels.
- Optimisation du code : recommandations pour améliorer la qualité du code des applications, la gestion des erreurs et la résilience du code.

Cette catégorisation vous aide à comprendre les domaines dans lesquels vos améliorations opérationnelles sont les plus nécessaires et vous permet de hiérarchiser les recommandations en fonction des domaines d'intérêt de votre équipe.

Avantages

- Prévenir les incidents récurrents — Traitez systématiquement les causes profondes plutôt que de répondre de manière répétée aux mêmes types de problèmes
- Réduisez le travail opérationnel : libérez votre équipe de la lutte répétitive contre les incendies pour se concentrer sur l'innovation et les améliorations stratégiques
- Améliorez la résilience du système : renforcez votre infrastructure, votre observabilité et vos processus de déploiement sur la base de données réelles sur les incidents
- Tirez les leçons des modèles historiques — Tirez parti des informations tirées des incidents passés pour apporter des améliorations ciblées ayant le plus grand impact

Résumé de l'agent

Le résumé des agents figurant sur la page Ops Backlog de l'application Web fournit une description des résultats de la dernière évaluation des incidents récents. Le résumé explique le nombre

d'enquêtes sur les incidents analysées, quels incidents sont similaires aux précédents et quelles recommandations ont été créées ou mises à jour avec de nouvelles informations.

Le résumé vous aide à comprendre rapidement ce que l'agent a découvert lors de sa dernière évaluation et met en évidence les recommandations les plus importantes susceptibles d'avoir le plus d'impact sur votre posture opérationnelle.

Contrôler les évaluations

Vous pouvez contrôler le moment où AWS DevOps l'agent évalue les incidents et génère des recommandations :

- Exécution manuelle des évaluations : cliquez sur le bouton Exécuter maintenant sur la page Ops Backlog pour démarrer immédiatement une évaluation. Cela est utile lorsqu'une enquête récente justifie une mise en œuvre rapide des améliorations recommandées.
- Arrêt des évaluations actives : cliquez sur le bouton Arrêter l'évaluation dans la page Ops Backlog pour arrêter une évaluation en cours.

Gérer les recommandations

AWS DevOps L'agent fournit des recommandations sur la page Ops Backlog, où vous pouvez les consulter et les gérer :

- Afficher les détails des recommandations : cliquez sur une recommandation pour ouvrir la page des détails de la recommandation, où vous pouvez voir plus d'informations sur l'amélioration suggérée, notamment les incidents qui ont inspiré la recommandation, les impacts attendus et les prochaines étapes. Pour les recommandations concernant les modifications de code, vous pouvez également consulter la spécification prête à être transmise à un agent de codage pour mise en œuvre.
- Conserver : cliquez sur « Conserver » pour conserver une recommandation dans votre carnet de commandes à des fins de suivi. Cela vous permet de suivre les améliorations que vous prévoyez de mettre en œuvre et de suivre leur progression.
- Supprimer — Cliquez sur « Supprimer » pour supprimer une recommandation de votre backlog. Lorsque vous annulez une recommandation, vous pouvez expliquer en langage naturel pourquoi elle ne répond pas à vos besoins. L'agent tire les leçons de ces commentaires et les utilise pour élaborer de futures recommandations, en veillant à ce qu'elles correspondent mieux à vos priorités et exigences opérationnelles au fil du temps.

- Mise en œuvre — Cliquez sur « Mise en œuvre » pour marquer une recommandation comme terminée. Cela vous permet de suivre les améliorations qui ont été appliquées et permet à l'agent de mesurer l'efficacité de ses recommandations au fil du temps.
- Suppression automatique : les recommandations qui n'ont pas été marquées comme conservées ou mises en œuvre peuvent être supprimées au bout de 6 semaines environ si aucun nouvel incident n'aurait été évité grâce à la mise en œuvre de la recommandation. Cela garantit que la page Ops Backlog se concentre sur les améliorations les plus pertinentes pour relever vos défis opérationnels.
- Mises à jour des recommandations : les recommandations existantes sont mises à jour lorsque de nouveaux incidents qui auraient pu être évités par la recommandation sont découverts. Les mises à jour peuvent modifier la priorité de la recommandation ou l'affiner en fonction de nouvelles informations.

Spécifications prêtes à être utilisées par les agents

Pour les recommandations impliquant des modifications de code ou de configuration, l' AWS DevOps agent peut générer une spécification prête à être utilisée par l'agent. Cette spécification fournit un document structuré qui peut être transmis directement à un agent de codage pour la mise en œuvre.

La spécification inclut :

- Exposé du problème : résumé du problème et de sa cause première
- Résumé de la solution : description détaillée de l'approche recommandée
- Référentiels cibles : référentiels spécifiques dans lesquels des modifications doivent être apportées
- Modifications du code : descriptions détaillées de ce qui doit être modifié et pourquoi, avec des chemins de fichiers spécifiques et des considérations relatives à la mise en œuvre
- Exigences relatives aux tests — Quels scénarios doivent être testés
- Plan de mise en œuvre — Une approche progressive pour la mise en œuvre des changements

Les spécifications prêtes pour les agents accélèrent la mise en œuvre en fournissant aux agents de codage le contexte dont ils ont besoin pour apporter des modifications prêtes à être mises en production sans avoir à faire appel à de nombreux ingénieurs. back-and-forth

Mise en œuvre des recommandations

Pour optimiser la valeur des recommandations proactives en matière de prévention des incidents, considérez les pratiques suivantes pour y donner suite :

- Utilisation de spécifications prêtes pour l'agent : pour les recommandations relatives aux modifications de code, utilisez la spécification générée pour accélérer la mise en œuvre en la remettant à un agent de codage ou en l'utilisant comme guide détaillé pour la mise en œuvre manuelle.
- Ajouter des recommandations à votre carnet de tickets : copiez les recommandations dans le système de billetterie ou l'outil de gestion de projet de votre équipe pour vous assurer qu'elles sont priorisées aux côtés des autres travaux d'ingénierie.
- Hiérarchisation des recommandations en fonction de leur impact — Concentrez-vous d'abord sur les recommandations qui concernent les types d'incidents les plus fréquents ou les plus graves, ou ceux qui affectent les systèmes critiques.
- Suivi des progrès de la mise en œuvre — Surveillez les recommandations qui ont été mises en œuvre et mesurez leur efficacité en observant si le nombre d'incidents similaires diminue au fil du temps.
- Coordination avec les équipes de développement : partagez les recommandations avec les équipes appropriées qui possèdent les systèmes concernés, en veillant à ce qu'elles disposent du contexte et des ressources nécessaires pour mettre en œuvre les améliorations.

DevOps Tâches à la demande

AWS DevOps Agent On Demand Tasks est un assistant conversationnel basé sur l'intelligence artificielle générative (IA) qui permet aux équipes opérationnelles d'interroger l'architecture de leur application, d'analyser l'état du système et d'accéder aux informations issues des enquêtes en langage naturel. Vous pouvez poser des questions sur vos AWS ressources, les indicateurs du système, l'état des alarmes, l'historique des déploiements et les modèles d'incidents. Le chat fournit des réponses immédiates basées sur votre infrastructure et vos données opérationnelles réelles, éliminant ainsi le besoin de naviguer entre plusieurs AWS consoles ou outils de surveillance.

Le chat est intégré à l'ensemble de l'application Web DevOps Agent Space et fournit des réponses contextuelles en fonction de la page que vous consultez. L'interface conserve l'historique des conversations, ce qui vous permet de poursuivre les discussions précédentes et de vous appuyer sur des requêtes antérieures.

Capacités des tâches

AWS DevOps Agent On Demand Tasks fournit des fonctionnalités complètes pour vous aider à gérer et à comprendre votre infrastructure :

Requêtes sur les ressources : renseignez-vous sur les AWS ressources de votre espace agent, notamment les fonctions Lambda, les tables DynamoDB, les déploiements EKS, les certificats et les configurations d'infrastructure. Le chat peut filtrer et analyser les ressources en fonction d'attributs tels que les versions d'exécution, les paramètres de capacité ou l'état du déploiement. Par exemple, demandez « Combien de Lambdas utilisent Python 3.8 ? » ou « Est-ce que j'ai des certificats sur le point d'expirer ? »

Analyse de l'état du système : interrogez les indicateurs de santé actuels et historiques du système, notamment l'état des alarmes, les taux d'erreur, l'utilisation du processeur et la disponibilité des services. Le chat peut générer des résumés de santé couvrant des périodes spécifiques et identifier les tendances du comportement du système. Posez des questions telles que « Quelles alarmes se sont déclenchées au cours des dernières 24 heures ? » ou « Y a-t-il eu des erreurs 5xx au cours de la dernière heure ? »

Informations sur les enquêtes : accédez aux informations issues des enquêtes achevées et en cours, notamment l'analyse des causes profondes, les hypothèses explorées, les journaux examinés et les modèles de résolution. Le chat permet d'identifier les causes courantes des incidents et de fournir des recommandations basées sur des données historiques. Demande « Quelle est la cause la plus fréquente des incidents survenus le mois dernier ? » ou « Quel est le délai moyen de résolution des enquêtes terminées ? »

Direction de l'enquête : lorsque vous consultez la page détaillée d'une enquête, orientez l'enquête en demandant à l'agent de se concentrer sur des journaux spécifiques, d'explorer des hypothèses particulières ou de mettre à jour l'analyse des causes profondes. Fournissez des informations de pilotage telles que « Concentrez-vous sur les journaux du service de paiement et mettez à jour votre RCA » ou « Explorez l'hypothèse selon laquelle le problème est dû à la régulation de DynamoDB ».

Artefacts de chat : générez des rapports et des documents structurés, tels que des résumés de santé opérationnelle, des rapports d'erreurs et des analyses d'incidents. Les artefacts apparaissent dans un panneau dédié et permettent l'édition versionnée au sein de la conversation.

Filtrage des recommandations : interrogez les recommandations de prévention des incidents à l'aide de critères spécifiques, tels que les recommandations relatives à des services particuliers ou à des préoccupations opérationnelles. Chat explique les considérations relatives à l'impact et à la mise en

œuvre de chaque recommandation. Par exemple, « Afficher les recommandations qui empêcheront les incidents impliquant DynamoDB » ou « Quelles recommandations pourraient m'aider à détecter plus rapidement les problèmes de latence des demandes ? »

Accès au chat

Le chat est disponible sous forme de panneau permanent sur le côté gauche de l'application Web DevOps Agent Space. La barre latérale gauche comprend un bouton + Nouveau chat, une section Pages pour accéder aux incidents, au carnet des opérations et à la topologie, ainsi qu'une section Discussions qui affiche vos conversations récentes. Choisissez Afficher tout pour voir l'historique complet de vos conversations.

Le chat fournit des réponses contextuelles en fonction de l'endroit où vous y accédez :

Topologie : posez des questions générales sur les ressources, l'architecture et la santé opérationnelle de votre espace agent. Le chat offre une visibilité complète sur tous les comptes et services connectés. Dans ce contexte, vous pouvez interroger les configurations des ressources, l'historique des déploiements, les informations topologiques et les intégrations d'outils d'observabilité.

Réponse aux incidents : lorsque vous consultez la page de réponse aux incidents, posez des questions sur les tendances des enquêtes, les délais de résolution et les modèles d'incidents dans votre espace d'agents. Le chat peut analyser les données d'enquête historiques afin d'identifier les causes courantes et les opportunités d'amélioration.

Détails de l'enquête : lorsque vous consultez une enquête spécifique, Chat fournit des réponses contextuelles concernant cette enquête. Renseignez-vous sur les journaux examinés, les hypothèses explorées, les conclusions relatives aux causes profondes et les plans d'atténuation. Vous pouvez également fournir des informations de direction pour orienter l'enquête.

Prévention : depuis la page de prévention, interrogez les recommandations à l'aide de filtres, comprenez pourquoi elles ont été formulées et explorez les approches de mise en œuvre. Le chat vous aide à hiérarchiser les priorités et à comprendre l'impact des recommandations de prévention des incidents.

L'interface de chat reste disponible lorsque vous passez d'une page à l'autre, mais le contexte change pour fournir des informations pertinentes pour votre affichage actuel. Lorsque vous entamez une nouvelle conversation, elle commence sans contexte préalable. Lorsque vous poursuivez une conversation existante, Chat conserve l'historique complet des conversations pour les questions de suivi.

Réponses adaptées au contexte

Chat adapte ses réponses en fonction de la page que vous consultez dans l'application Web DevOps Agent Space. Cette connaissance du contexte vous garantit de recevoir des informations pertinentes sans avoir à préciser l'enquête ou l'étendue des ressources que vous demandez.

Lorsque vous consultez la page détaillée d'une enquête, Chat comprend automatiquement que vous posez une question sur cette enquête spécifique. Des questions telles que « Quels journaux avez-vous consultés ? » ou « Quelles hypothèses avez-vous explorées ? » faites référence à l'enquête actuellement affichée. Lorsque vous fournissez des informations de pilotage, Chat les applique à l'enquête en cours et crée une nouvelle version de la cause première, le cas échéant.

Sur la page de prévention, Chat comprend que vous êtes intéressé par les recommandations de prévention des incidents. Les requêtes filtrent et analysent automatiquement les recommandations dans le contexte de votre espace agent. Le système reconnaît si vous demandez des recommandations générales ou des recommandations spécifiques.

Lorsque vous accédez à Chat depuis la page Topologie, Chat offre une large visibilité sur toutes les ressources, statistiques et données historiques de votre espace d'agent. Vous pouvez poser des questions sur toute ressource, service ou préoccupation opérationnelle sans préciser le contexte de l'enquête ou de la recommandation.

Cette connaissance du contexte élimine le besoin de spécifier à plusieurs reprises à quelle enquête, recommandation ou étendue de ressource vous faites référence, créant ainsi un flux de conversation plus naturel.

Gérer les conversations

Chat conserve l'historique des conversations pour vous permettre de poursuivre les discussions précédentes et de faire référence à des requêtes antérieures.

Création de nouvelles conversations — Cliquez sur le bouton « Nouvelle session » dans le panneau de discussion pour démarrer une nouvelle conversation sans contexte préalable. Les nouvelles conversations ne reprennent pas les informations des discussions précédentes, ce qui vous permet de poser des questions indépendantes sans confusion.

Accès à l'historique des conversations : cliquez sur « Historique » pour afficher toutes les conversations précédentes dans votre espace agent. Les conversations sont organisées par ordre chronologique avec des horodatages et un texte d'aperçu. L'historique des conversations est conservé pendant 90 jours et est réservé à votre compte utilisateur dans l'espace agent.

Poursuite des conversations : sélectionnez n'importe quelle conversation dans votre historique pour reprendre là où vous vous êtes arrêtée. Le chat conserve le contexte complet des messages précédents, ce qui vous permet de poser des questions de suivi qui font référence à des parties antérieures de la conversation. Lorsque vous changez de page pendant que vous consultez une conversation, le contexte de la conversation est conservé, mais le contexte spécifique à la page est mis à jour en fonction de votre position actuelle.

Notez que l'historique des conversations est isolé dans chaque espace agent. Les conversations dans un espace agent ne sont ni visibles ni accessibles depuis les autres espaces agent. Cette isolation garantit que les informations sensibles restent compartimentées en fonction des limites de votre organisation.

Génération d'artefacts

AWS DevOps L'agent prend en charge les artefacts du chat, c'est-à-dire des documents structurés et versionnés générés par l'agent au cours d'une conversation. Les artefacts fournissent un panneau interactif dédié dans l'interface utilisateur du chat pour examiner et modifier le contenu généré par l'IA, tel que les rapports opérationnels, les résumés d'erreurs et les évaluations de santé.

Vous pouvez demander des artefacts depuis n'importe quelle page de l'application Web DevOps Agent Space. Le chat utilise le contexte de page actuel pour définir le contenu de l'artefact.

Comment fonctionnent les artefacts

Lorsque vous demandez à Chat de créer ou de mettre à jour du contenu, Chat génère un artefact (généralement un document formaté) et l'affiche dans le panneau des artefacts à côté de la conversation.

Générer — Envoyez une demande en langage naturel pour créer un rapport ou un document. Par exemple, demandez « Générer un rapport de santé opérationnel hebdomadaire pour mon agent Space » ou « Montrez-moi un rapport pour mes 4xx erreurs de la semaine dernière ».

Révision — L'artefact apparaît dans un panneau dédié à côté de la conversation. Vous pouvez consulter le contenu complet tout en continuant à interagir avec le chat.

Modifier — Demandez des modifications à l'artefact via le chat. Par exemple, demandez « Ajouter une section sur les démarrages à froid Lambda » ou « Mettre à jour le rapport pour inclure les données du mois dernier ». Le chat crée une nouvelle version de l'artefact avec les modifications que vous avez demandées.

Exemples de requêtes

Les exemples suivants illustrent les types de questions que vous pouvez poser à Chat. Ces exemples sont organisés par cas d'utilisation et par contexte.

Requêtes de génération d'artifacts

Depuis n'importe quelle page de l'application Web DevOps Agent Space :

- Générer un résumé hebdomadaire de l'état de santé opérationnel de mon agent Space
- Créez un rapport de toutes les erreurs 4xx de la semaine dernière
- Créez un rapport récapitulatif des incidents des 30 derniers jours
- Créez un résumé de l'activité des alarmes pour le service de paiement cette semaine
- Générer un rapport sur l'historique des déploiements des 7 derniers jours
- Résumez toutes les recommandations ouvertes dans un rapport

Demandes d'informations sur les ressources

Depuis n'importe quelle page de l'application Web DevOps Agent Space :

- Combien de fonctions Lambda utilisent Python 3.8 ?
- Est-ce que j'ai des certificats sur le point d'expirer ?
- Répertorier toutes les tables DynamoDB avec facturation à la demande
- Montrez-moi les clusters EKS en production
- Quelles fonctions Lambda n'ont pas été déployées au cours des 90 derniers jours ?
- Répertorier les compartiments S3 sans activation de la gestion des versions
- Quelles instances RDS exécutent la version X de la base de données ?

Requêtes sur l'état du système

À partir des pages Topologie ou Réponse aux incidents :

- Quelles alarmes se sont déclenchées au cours des dernières 24 heures ?
- Y a-t-il eu des erreurs 5xx au cours de la dernière heure ?
- Afficher les tendances en matière d'erreurs Lambda pour le service de paiement

- Quelle est l'utilisation du processeur pour mon cluster ECS ?
- Mes équilibreurs de charge contiennent-ils des cibles insalubres ?
- Afficher les événements de régulation d'API Gateway survenus hier
- Quels services ont enregistré le taux d'erreur le plus élevé la semaine dernière ?
- Donnez-moi un bilan de santé global couvrant les dernières 24 heures

Requêtes relatives à l'outil d'observabilité

À partir de Topology :

- Répertorier les groupes de logs Splunk
- Montrez-moi les métriques de Prometheus et leurs seuils d'alarme
- Quels sont les moniteurs Datadog configurés pour ce service ?
- Répertorier les politiques d'alerte de New Relic
- Montrez-moi les configurations du tableau de bord Dynatrace

Requêtes d'information sur les enquêtes

À partir de la page de réponse aux incidents :

- Quelle est la cause la plus fréquente des incidents survenus le mois dernier ?
- Quel est le délai moyen de résolution des enquêtes terminées ?
- Résumez les enquêtes de la semaine dernière et leur RCA
- Combien d'incidents ont été provoqués par la régulation DynamoDB ?
- Montrez-moi les tendances des enquêtes au cours du dernier trimestre
- Quels sont les services concernés par les incidents les plus fréquents ?

Demandes détaillées relatives à l'enquête

À partir de la page détaillée de l'enquête :

- Quels journaux avez-vous consultés ?
- Quelles hypothèses avez-vous explorées ?
- Dans quelle mesure les mesures d'atténuation que vous proposez sont-elles risquées ?

- Quelle a été la chronologie des événements lors de cet incident ?
- Pourquoi en avez-vous conclu que c'était la cause première ?
- Quelles preuves appuient votre analyse des causes profondes ?
- Qui a assuré le pilotage au cours de votre enquête ?
- Donnez-moi un résumé de cette enquête sur l'incident

Demandes relatives au pilotage des enquêtes

À partir de la page détaillée de l'enquête :

- Concentrez-vous sur les journaux du service de paiement entre 14h00 et 15h00 UTC et mettez à jour votre RCA
- Explorez l'hypothèse selon laquelle la régulation de DynamoDB est à l'origine du problème
- Vérifiez la configuration du cluster ECS pour voir si cela a provoqué l'alarme
- Vérifiez uniquement les journaux des 2 dernières heures, pas de la journée complète
- Enquêtez sur le pic d'erreurs à 15 h
- Consultez les journaux d'API Gateway au lieu des journaux Lambda

Demandes de recommandation de prévention

À partir de la page Prévention :

- Quelles sont mes 3 principales recommandations en matière de prévention des incidents ?
- Afficher les recommandations qui permettront d'éviter les incidents impliquant DynamoDB
- Quelles recommandations pourraient m'aider à détecter plus rapidement les problèmes de latence des demandes ?
- Répertorier les améliorations d'observabilité susceptibles de prévenir des incidents similaires
- Afficher les recommandations d'infrastructure pour le service de paiement
- Quelles recommandations ont le plus d'impact sur la résilience du système ?

Activer le chat dans votre espace d'agent

Le chat est disponible dans toutes les applications Web DevOps Agent Space. Le processus de configuration varie selon que vous disposez d'un espace agent nouveau ou existant.

Nouveaux espaces pour agents

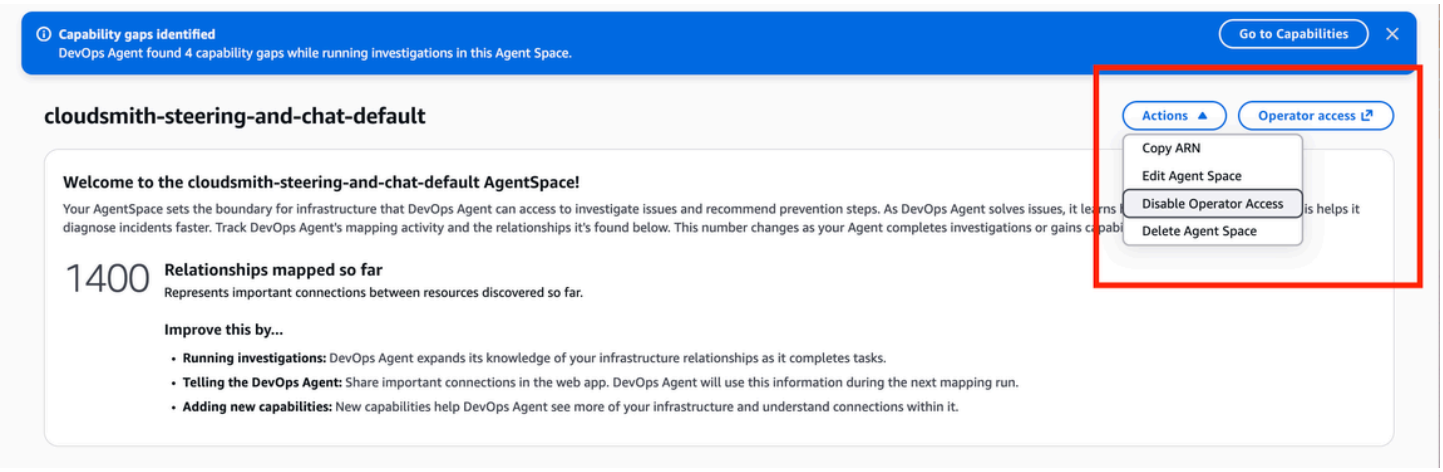
Le chat est automatiquement activé lorsque vous créez un nouvel espace agent. Aucune configuration supplémentaire ou configuration d'autorisations IAM n'est requise. Une fois que vous avez configuré votre application Web DevOps Agent Space, Chat est immédiatement disponible sous forme de panneau permanent sur le côté gauche de n'importe quelle page.

Espaces d'agents existants

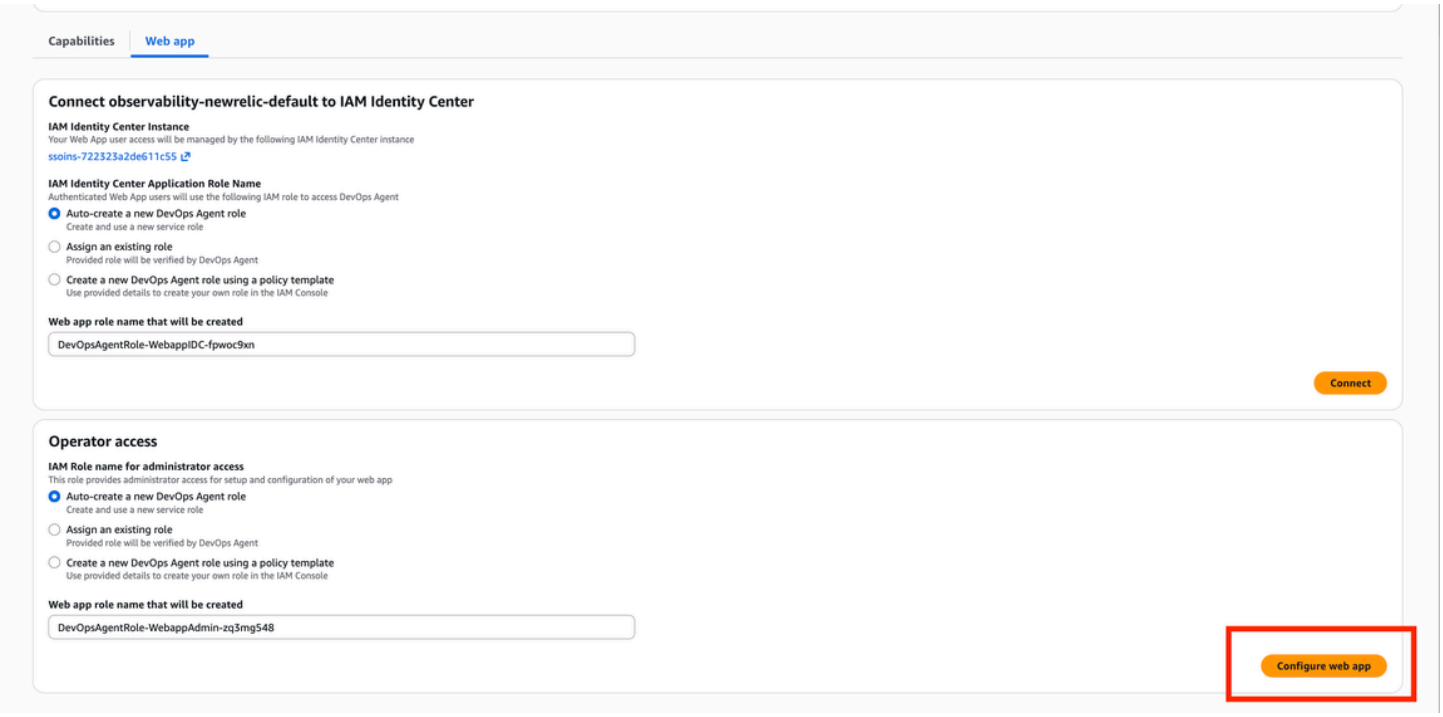
Si vous avez créé votre espace agent avant le lancement de Chat, vous devez activer les autorisations IAM requises. Vous avez deux options :

Option 1 : révoquer et réactiver l'accès à l'application de l'opérateur

Accédez à la console d'administration de l' AWS DevOps agent, recherchez le menu déroulant Action dans le coin supérieur droit et désactivez la configuration d'accès opérateur actuelle.



Activez ensuite l'option de création automatique pour l'accès de l'opérateur.



Cela applique automatiquement les autorisations IAM requises pour Chat ainsi que toutes les autres autorisations d'opérateur actuelles.

Option 2 : ajouter des autorisations IAM manuellement

Ajoutez les autorisations IAM suivantes à votre rôle d'accès d'opérateur existant :

- `aidevops:ListChats`— Afficher l'historique des conversations par chat
- `aidevops:CreateChat`— Créez de nouvelles conversations par chat
- `aidevops:SendMessage`— Envoyer des messages et recevoir des réponses

Accédez à la console AWS IAM, recherchez votre rôle d'opérateur d' DevOps agent et ajoutez ces autorisations à la politique de rôle. Le chat devient disponible immédiatement après l'ajout des autorisations.

Après avoir sélectionné l'une ou l'autre des options, actualisez votre application Web DevOps Agent Space pour que le panneau de discussion apparaisse sur le côté gauche de n'importe quelle page.

Configuration des fonctionnalités de AWS DevOps l'agent

AWS DevOps Les fonctionnalités d'agent étendent les fonctionnalités de votre agent en le connectant à vos outils et à votre infrastructure existants. Configurez ces fonctionnalités pour permettre une investigation complète des incidents, des workflows de réponse automatisés et une intégration parfaite à votre DevOps écosystème.

Les fonctionnalités suivantes vous aident à optimiser l'efficacité de votre DevOps agent :

- AWS Configuration d'EKS Access - Activez l'introspection des clusters Kubernetes, des journaux de pods et des événements de cluster pour les environnements EKS publics et privés
- Intégration Azure - Connectez les abonnements Azure et les DevOps organisations Azure pour étudier les ressources Azure et corréler les DevOps déploiements Azure avec les incidents
- Intégration du pipeline CI/CD - Connect GitHub et GitLab pipelines pour corréler les déploiements aux incidents et suivre les modifications du code au cours des enquêtes
- Connexions aux serveurs MCP : étendez les capacités d'investigation en connectant des outils d'observabilité externes et des systèmes de surveillance personnalisés via le protocole Model Context Protocol
- AWS Accès multicompte : configurez des AWS comptes secondaires pour examiner les ressources de l'ensemble de votre organisation lors de la réponse aux incidents
- Intégration des sources de télémétrie : connectez des plateformes de surveillance telles que Datadog, Dynatrace, Grafana, New Relic et Splunk pour un accès complet aux données d'observabilité
- Intégration de la billetterie et du chat : Connect ServiceNow et Slack pour automatiser les flux de travail de réponse aux incidents et permettre la collaboration en équipe PagerDuty
- Configuration du webhook - Autoriser les systèmes externes à déclencher automatiquement des investigations sur les DevOps agents via des requêtes HTTP
- EventBridge Intégration avec Amazon : intégrez l' AWS DevOps agent dans des applications pilotées par des événements en acheminant les événements du cycle de vie d'investigation et d'atténuation vers les cibles Amazon EventBridge

Vous pouvez configurer chaque fonctionnalité indépendamment en fonction des besoins spécifiques de votre équipe et de la pile d'outils existante. Commencez par les intégrations les plus essentielles à votre flux de travail de réponse aux incidents, puis étendez les fonctionnalités supplémentaires selon les besoins.

Migration de la version préliminaire publique à la disponibilité générale

Si vous avez utilisé AWS DevOps l'Agent lors de la version préliminaire publique, vous devez mettre à jour vos rôles IAM avant la sortie de GA. Ce guide explique comment mettre à jour les rôles de surveillance et les rôles d'opérateur dans vos comptes.

Ce qui change

1. L'historique des discussions à la demande pendant la prévisualisation n'est plus accessible
2. Les nouvelles politiques gérées remplacent les politiques disponibles lors de la version préliminaire
3. La portée d'accès à l'application IAM Identity Center d'agent Spaces est peut-être obsolète

Historique des discussions à la demande depuis l'aperçu public

La version GA introduit des mesures de sécurité supplémentaires pour renforcer les contrôles d'accès aux historiques de discussion. En raison de ces modifications, les historiques des discussions à la demande datant de la période de prévisualisation publique (avant le 30 mars 2026) ne sont plus accessibles. Les journaux d'investigation et les résultats créés lors de l'avant-première publique ne sont pas affectés. Cette modification s'applique uniquement aux conversations par chat à la demande.

Nouvelles politiques gérées

Pour GA, AWS fournit de nouvelles politiques gérées qui remplacent les politiques de l'ère de prévisualisation :

Type de rôle	Supprimer	Addition
Contrôle	Stratégie gérée par `AIOpsAssistantPolicy`	Stratégie gérée par `AIDevOpsAgentAccessPolicy`
Opérateur (IAM et IDC)	Politique en ligne	Stratégie gérée par `AIDevOpsOperatorAppAccessPolicy`

En outre, les rôles d'opérateur nécessitent des politiques de confiance mises à jour, et les rôles d'opérateur IDC nécessitent une nouvelle politique intégrée.

Conditions préalables

- Accès aux AWS comptes sur lesquels vos rôles d' DevOps agent sont configurés (comptes principaux et tous les comptes secondaires)
- Autorisations IAM pour modifier les rôles, les politiques et les relations de confiance
- Votre identifiant d'espace agent, votre identifiant de AWS compte et votre région (visibles dans la console DevOps Agent)

Étape 1 : Mettre à jour les rôles de surveillance

Mettez à jour le rôle de surveillance dans votre compte principal et dans chaque compte secondaire. Il s'agit des rôles Primary/Secondary source configurés sous l'onglet Capabilities de votre espace agent (exemple de primary/secondary rôle :`DevOpsAgentRole-AgentSpace-3xj2396z`).

1. Dans la console de l' DevOps agent, accédez à votre espace agent et choisissez l'onglet Fonctionnalités.
2. Trouvez le rôle de surveillance de vos Primary/Secondary sources (par exemple,`DevOpsAgentRole-AgentSpace-3xj2396z`) et choisissez Modifier.
3. Sous Politiques d'autorisations, supprimez la politique `AIOpsAssistantPolicy` AWS gérée.
4. Choisissez Ajouter des autorisations, Joindre des politiques, puis attachez la politique `AIDevOpsAgentAccessPolicy` gérée.
5. Modifiez la politique en ligne et remplacez son contenu par ce qui suit, en remplaçant votre identifiant de compte :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateServiceLinkedRoles",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": [
```

```
                "arn:aws:iam::<account-id>:role/aws-service-role/resource-
explorer-2.amazonaws.com/AWSServiceRoleForResourceExplorer"
            ]
        }
    ]
}
```

1. La politique de confiance relative au rôle de surveillance ne nécessite aucune modification. Vérifiez qu'il correspond aux critères suivants :

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

- Répétez les étapes 2 à 6 pour le rôle de surveillance dans chaque compte secondaire.

Étape 2 : Mettre à jour le rôle d'opérateur (IAM)

1. Dans la console de l' DevOps agent, choisissez l'onglet Accès et recherchez le rôle de l'opérateur.
2. Dans la console IAM, supprimez la politique intégrée existante du rôle d'opérateur.

3. Choisissez Ajouter des autorisations, Joindre des politiques, puis attachez la politique `AIDevOpsOperatorAppAccessPolicy` gérée.
4. Choisissez l'onglet Relations de confiance, puis sélectionnez Modifier la politique de confiance. Remplacez la politique de confiance par la suivante, en remplaçant votre identifiant de compte, votre région et votre identifiant Agent Space :

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

Étape 3 : Mettre à jour les rôles des opérateurs (IDC)

Si vous utilisez IAM Identity Center avec un DevOps agent, mettez à jour chaque rôle d'opérateur IDC.

1. Dans la console IAM, accédez à Rôles et recherchez **`WebappIDC`** vos rôles IDC d' DevOps agent (par exemple,`DevOpsAgentRole-WebappIDC-<id>`).
2. Pour chaque rôle IDC :

a. Supprimez la politique intégrée existante.

b. Choisissez Ajouter des autorisations, Joindre des politiques, puis attachez la politique `AIDevOpsOperatorAppAccessPolicy` gérée.

c. Choisissez l'onglet Relations de confiance, puis sélectionnez Modifier la politique de confiance. Remplacez la politique de confiance par la suivante, en remplaçant votre identifiant de compte, votre région et votre identifiant Agent Space :

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

```
                        "arn:aws:iam::aws:contextProvider/IdentityCenter"
                    ]
                },
                "Null": {
                    "sts:RequestContextProviders": "false"
                }
            }
        }
    ]
}
```

d. Créez une nouvelle politique en ligne avec les autorisations suivantes, en remplaçant votre identifiant de compte :

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

Reconnectez le centre d'identité IAM (le cas échéant)

Les espaces d'agent créés lors de la préversion publique peuvent comporter une application IAM Identity Center configurée avec une étendue d'accès obsolète. Pour GA, le champ d'application

correct est **aidevops:read_write**. Si votre application IAM Identity Center possède l'étendue précédente (**awsaidevops:read_write**), vous devez déconnecter puis reconnecter IAM Identity Center.

Comment vérifier le périmètre de votre application IAM Identity Center

Exécutez la commande AWS CLI suivante pour vérifier le périmètre de votre application IAM Identity Center. Vous trouverez l'ARN de l'application dans la console IAM Identity Center sous Applications.

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

La sortie doit afficher la portée correcte **aidevops:read_write**:

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

Si le scope l'indique **awsaidevops:read_write**, il est obsolète. Suivez les étapes ci-dessous pour le mettre à jour.

Comment reconnecter IAM Identity Center

L'étendue d'accès d'une application IAM Identity Center AWS gérée ne peut pas être mise à jour directement. Vous devez vous déconnecter puis vous reconnecter :

1. Dans la console de l' AWS DevOps agent, accédez à votre espace agent et choisissez l'onglet Accès.
2. Choisissez Déconnecter à côté de la configuration du centre d'identité IAM.
3. Confirmez la déconnexion.
4. Choisissez Connect pour configurer à nouveau IAM Identity Center. Le service crée une nouvelle application IAM Identity Center avec le périmètre approprié.
5. Réaffectez les utilisateurs et les groupes à la nouvelle application dans la console IAM Identity Center.

> **Important**
>
> La déconnexion supprime le chat individuel des utilisateurs et l'historique des artefacts associés aux comptes utilisateurs d'IAM Identity Center. Les utilisateurs devront se reconnecter après leur reconnexion.

Vérification

Après avoir effectué toutes les étapes :

1. Retournez à la console de l' DevOps agent et vérifiez qu'aucune erreur d'autorisation n'apparaît dans l'onglet Agent Space Access.
2. Testez l'application Web de l'opérateur pour vérifier qu'elle se charge et fonctionne correctement.
3. Si vous utilisez IDC, vérifiez que les utilisateurs peuvent s'authentifier et accéder à l'expérience de l'opérateur.

Résolution des problèmes

Erreurs d'autorisation refusée après la migration

- Vérifiez qu'il `AIOpsAssistantPolicy` a été supprimé et qu'`AIDevOpsAgentAccessPolicy`il est associé aux rôles de surveillance.
- Vérifiez que les anciennes politiques intégrées ont été supprimées et qu'elles `AIDevOpsOperatorAppAccessPolicy` sont associées aux rôles des opérateurs.
- Vérifiez que les politiques de confiance des opérateurs incluent`sts:TagSession`.
- Vérifiez que vous avez remplacé toutes les valeurs d'espace réservé (`<account-id>`,`<region>`,`<agentspace-id>`) par des valeurs réelles.

Les comptes secondaires ne fonctionnent pas

- Le rôle de surveillance de chaque compte secondaire doit être mis à jour indépendamment. Connectez-vous à chaque compte et répétez l'étape 1.

Défaillances d'authentification IDC

- Vérifiez que la politique de confiance d'IDC inclut à la fois l'`sts:TagSession`instruction`sts:AssumeRole`/et la `TrustedIdentityPropagation` déclaration.
- Confirmez la politique en ligne avec `sso:ListInstances``sso:DescribeInstance`, et `identitystore:DescribeUser` a été créée.

L'historique des discussions à la demande est manquant après la migration

- L'historique des discussions à la demande datant de la période de préversion publique ne sera plus accessible après la sortie de GA. Ce comportement est attendu en raison des mesures de sécurité renforcées introduites en Géorgie. Les journaux d'investigation et les résultats publiés en avant-première ne sont pas affectés.

AWS Configuration de l'accès EKS

Vous pouvez permettre à AWS DevOps l'agent d'étudier les problèmes dans vos clusters Amazon EKS en exécutant des `kubectl` commandes en lecture seule sur des clusters publics et privés. Vous pouvez connecter un nombre illimité de clusters EKS au même agent Space.

Une fois connecté, l'agent peut aider à diagnostiquer les problèmes opérationnels dans vos clusters, en décrivant les ressources, en récupérant les journaux des pods, en inspectant les événements du cluster, en vérifiant l'état des nœuds, etc. L'agent ne peut pas créer, modifier ou supprimer de ressources dans votre cluster.

Conditions préalables

Avant de configurer l'accès EKS, assurez-vous que le mode d'authentification de votre cluster EKS inclut l'API EKS. Vous pouvez le vérifier dans l'onglet Accès de la console Amazon EKS. Si le mode n'inclut pas l'API EKS, sélectionnez un mode qui en inclut un avant de continuer.

Configuration

Ces étapes doivent être effectuées depuis la console Amazon EKS pour chaque cluster pour lequel vous souhaitez créer une entrée d'accès. Vous pouvez trouver l'ARN de votre rôle IAM dans votre espace agent (voirthe section called “Création d'un espace d'agents”) sous Fonctionnalités > Cloud > Source primaire > Modifier.

1. Accédez à l'onglet Accès. Si le mode d'authentification indique déjà l'API EKS, vous pouvez ajouter des entrées d'accès. Sinon, sélectionnez un mode qui inclut l'API EKS.
2. Dans l'onglet Accès, créez une nouvelle entrée d'accès IAM. Copiez l'ARN de votre rôle IAM source principal dans le cloud et saisissez-le comme principal IAM pour l'entrée d'accès. Cliquez sur Suivant.
3. Sélectionnez la politique AIOps AssistantPolicy d'accès de AWS Managed Amazon, puis sélectionnez Cluster pour l'étendue d'accès. (Sinon, si vous souhaitez que l'agent n'accède qu'à certains espaces de noms, sélectionnez les espaces de noms Kubernetes souhaités). Cliquez sur Ajouter une politique, puis sur Suivant.
4. Passez en revue les modifications et confirmez que la politique de saisie d'accès et le rôle IAM appropriés ont été choisis, puis créez votre entrée d'accès en cliquant sur « Créer ».

Pour vérifier que l'accès EKS a été correctement configuré, accédez à l'application Operator et lancez une nouvelle enquête en posant à l'agent une question sur votre cluster, par exemple « listez tous les pods dans l'espace de noms par défaut » ou « montrez-moi les événements récents de mon cluster ».

Résolution des problèmes

Si l'agent ne parvient pas à accéder à votre cluster, vérifiez que l'entrée d'accès utilise le bon ARN du rôle IAM indiqué dans la boîte de dialogue de configuration et que la politique AIOps AssistantPolicy d'accès Amazon est jointe.

Connecter Azure

L'intégration Azure permet à l' AWS DevOps agent d'étudier les ressources de votre environnement Azure et de corréler les déploiements du DevOps pipeline Azure avec les incidents opérationnels. En connectant Azure, l'agent gagne en visibilité sur votre infrastructure Azure et peut effectuer une analyse des causes profondes à la fois sur les ressources Azure AWS et sur celles d'Azure.

L'intégration Azure comprend deux fonctionnalités indépendantes :

- Ressources Azure : permet à l'agent de découvrir et d'étudier les ressources du cloud Azure telles que les machines virtuelles, les clusters Azure Kubernetes Service (AKS), les bases de données et les composants réseau. L'agent utilise Azure Resource Graph pour interroger vos ressources lors des enquêtes sur les incidents.

- Azure DevOps : permet à l'agent d'accéder aux DevOps référentiels Azure et à l'historique d'exécution du pipeline. L'agent peut corréler les modifications de code et les déploiements avec les incidents afin d'identifier les causes profondes potentielles.

Chaque fonctionnalité est enregistrée au niveau du AWS compte et peut ensuite être associée à des espaces d'agent individuels.

Modes d'enregistrement

AWS DevOps L'agent prend en charge deux méthodes pour se connecter à Azure :

- Consentement de l'administrateur : flux rationalisé basé sur le consentement dans lequel vous autorisez l'application AWS DevOps Agent Entra dans votre client Azure. Dans la console, cela apparaît sous la forme de l'option de consentement de l'administrateur. Cette méthode nécessite de se connecter avec un compte autorisé à obtenir le consentement de l'administrateur dans Microsoft Entra ID.
- Enregistrement des applications : approche autogérée dans le cadre de laquelle vous créez votre propre application Entra avec des informations d'identification fédérées à l'aide de la fédération d'identité sortante. Dans la console, cela apparaît sous la forme de l'option d'enregistrement de l'application. Cette méthode convient lorsque vous avez besoin de plus de contrôle sur la configuration de l'application ou lorsque les autorisations de consentement de l'administrateur ne sont pas disponibles.

Les deux méthodes offrent les mêmes fonctionnalités. Vous pouvez utiliser l'une ou les deux méthodes au sein du même AWS compte.

Limitations connues

- Consentement de l'administrateur : un AWS compte par locataire Azure — Chaque locataire Azure ne peut avoir son application AWS DevOps Agent Entra associée qu'à un seul AWS compte à la fois. Pour associer le même locataire à un autre AWS compte, vous devez d'abord annuler l'enregistrement existant.
- Enregistrement de l'application : application unique par enregistrement — Chaque enregistrement d'application doit utiliser une application différente (ID client). Vous ne pouvez pas enregistrer plusieurs configurations avec le même ID client.
- Azure DevOps : accès au code source — L' DevOps intégration Azure permet d'accéder à l'historique d'exécution du pipeline quel que soit l'endroit où le code source est hébergé. Toutefois,

pour accéder au code source proprement dit, le référentiel doit être connecté séparément via un fournisseur de source compatible (par exemple,the section called "Connecter GitHub"). Le code source hébergé dans Bitbucket n'est pas directement accessible via l' DevOps intégration Azure.

Rubriques

Connecter les ressources Azure

L'intégration d'Azure Resources permet à AWS DevOps l'agent de découvrir et d'étudier les ressources de vos abonnements Azure lors d'enquêtes sur des incidents. L'agent utilise Azure Resource Graph pour la découverte des ressources et peut accéder aux métriques, aux journaux et aux données de configuration dans votre environnement Azure.

Cette intégration suit un processus en deux étapes : enregistrer Azure au niveau du AWS compte, puis associer des abonnements Azure spécifiques à des espaces d'agent individuels.

Conditions préalables

Avant de connecter Azure Resources, assurez-vous d'avoir :

- Accès à la console de AWS DevOps l'agent
- Un compte Azure avec accès à l'abonnement cible
- Pour la méthode de consentement de l'administrateur : un compte autorisé à effectuer le consentement de l'administrateur dans le Microsoft Entra ID
- Pour la méthode d'enregistrement des applications : une application Entra autorisée à configurer les informations d'identification fédérées, et la fédération d'identité sortante activée sur votre compte AWS

> Remarque : Vous pouvez également démarrer l'enregistrement depuis un espace agent. Accédez aux sources secondaires, cliquez sur Ajouter, puis sélectionnez Azure. Si Azure Cloud n'est pas encore enregistré, la console vous guide d'abord tout au long de l'enregistrement.

Enregistrement des ressources Azure via le consentement de l'administrateur

La méthode Admin Consent utilise un flux basé sur le consentement avec l'application gérée par l' AWS DevOps agent.

Étape 1 : Commencez l'enregistrement

1. Connectez-vous à la console AWS de gestion et accédez à la console de l' AWS DevOps agent
2. Accédez à la page Capability Providers
3. Localisez la section Azure Cloud et cliquez sur Enregistrer
4. Sélectionnez la méthode d'enregistrement du consentement de l'administrateur

Étape 2 : Compléter le consentement de l'administrateur

1. Vérifiez les autorisations demandées
2. Cliquez pour continuer : vous êtes redirigé vers la page de consentement de l'administrateur Microsoft Entra
3. Connectez-vous avec un compte utilisateur principal autorisé à obtenir le consentement de l'administrateur
4. Examiner et donner son accord pour la demande d' AWS DevOps agent

Étape 3 : Autorisation complète de l'utilisateur

1. Après le consentement de l'administrateur, vous êtes invité à obtenir l'autorisation de l'utilisateur pour vérifier votre identité en tant que membre du locataire autorisé
2. Connectez-vous avec un compte appartenant au même client Azure
3. Après autorisation, vous êtes redirigé vers la console de l' AWS DevOps agent avec un statut de réussite

Étape 4 : Attribuer des rôles

Consultez la section Affectation de rôles Azure ci-dessous. Recherchez l'AWS DevOps agent lors de la sélection des membres.

Enregistrement des ressources Azure via l'enregistrement des applications

La méthode d'enregistrement des applications utilise votre propre application Entra avec des identifiants d'identité fédérés.

Étape 1 : Commencez l'enregistrement

1. Dans la console de l' AWS DevOps agent, accédez à la page Capability Providers
2. Localisez la section Azure Cloud et cliquez sur Enregistrer
3. Sélectionnez la méthode d'enregistrement de l'application

Étape 2 : Créez et configurez votre application Entra

Suivez les instructions affichées dans la console pour :

1. Activez la fédération des identités sortantes dans votre AWS compte (dans la console IAM, allez dans Paramètres du compte → Fédération des identités sortantes)
2. Créez une application Entra dans votre identifiant Microsoft Entra ou utilisez-en une existante
3. Configurer les informations d'identification fédérées sur l'application

Étape 3 : Fournissez les détails de l'enregistrement

Remplissez le formulaire d'inscription avec :

- ID de locataire : votre identifiant de locataire Azure
- Nom du locataire : nom d'affichage du locataire
- ID client — L'identifiant de l'application (client) de l'application Entra que vous avez créée
- Audience : identifiant d'audience pour l'identifiant fédéré

Étape 4 : Création du rôle IAM

Un rôle IAM est automatiquement créé lorsque vous soumettez l'enregistrement via la console. Cela permet à AWS DevOps l'agent d'assumer les informations d'identification et d'appeler`sts:GetWebIdentityToken`.

Étape 5 : Attribuer des rôles

Consultez la section Affectation de rôles Azure ci-dessous. Recherchez l'application Entra que vous avez créée lors de la sélection des membres.

Étape 6 : Compléter l'enregistrement

1. Confirmez la configuration dans la console de AWS DevOps l'agent
2. Cliquez sur Soumettre pour terminer l'enregistrement

Attribuer des rôles Azure

Après l'enregistrement, accordez à l'application un accès en lecture à votre abonnement Azure. Cette étape est la même pour les méthodes de consentement de l'administrateur et d'enregistrement de l'application.

1. Dans le portail Azure, accédez à votre abonnement cible
2. Accédez au contrôle d'accès (IAM)
3. Cliquez sur Ajouter > Ajouter une attribution de rôle
4. Sélectionnez le rôle de lecteur et cliquez sur Suivant
5. Cliquez sur Sélectionner les membres, recherchez l'application (soit AWS DevOps agent pour le consentement de l'administrateur, soit votre propre application Entra pour l'enregistrement de l'application)
6. Sélectionnez l'application et cliquez sur Réviser + attribuer
7. (Facultatif) Pour permettre à l'agent d'accéder aux clusters Azure Kubernetes Service (AKS), effectuez la configuration d'accès AKS suivante.

> Exigence de sécurité : le principal du service doit se voir attribuer uniquement le rôle de lecteur (et éventuellement les rôles en lecture seule AKS répertoriés ci-dessous). Le rôle de lecteur sert de limite de sécurité qui limite l'agent aux opérations en lecture seule et limite l'impact des attaques indirectes par injection rapide. L'attribution de rôles dotés d'autorisations d'écriture ou d'action augmente considérablement le rayon d'action et peut compromettre les ressources Azure. AWS DevOps L'agent effectue uniquement des opérations de lecture. L'agent ne modifie, ne crée ni ne supprime les ressources Azure.

Configuration de l'accès AKS (facultatif)

Étape 1 : accès au niveau Azure Resource Manager (ARM)

Attribuez le rôle d'utilisateur du cluster de services Azure Kubernetes à l'application.

Sur le portail Azure, accédez à Abonnements → sélectionnez abonnement → Contrôle d'accès (IAM) → Ajouter une attribution de rôle → sélectionnez Rôle utilisateur du cluster de services Azure Kubernetes → attribuer à l'application (soit AWS DevOps agent pour le consentement de l'administrateur, soit votre propre application Entra pour l'enregistrement des applications).

Cela couvre tous les clusters AKS inclus dans l'abonnement. Pour étendre la portée à des clusters spécifiques, attribuez-les plutôt au niveau du groupe de ressources ou du cluster individuel.

Étape 2 : Accès à l'API Kubernetes

Choisissez une option en fonction de la configuration d'authentification de votre cluster :

Option A : contrôle d'accès basé sur les rôles (RBAC) Azure pour Kubernetes (recommandé)

1. Activez Azure RBAC sur le cluster s'il n'est pas déjà activé : Portail Azure → Cluster AKS → Paramètres → Configuration de sécurité → Authentification et autorisation → sélectionnez Azure RBAC
2. Attribuer un rôle en lecture seule : Portail Azure → Abonnements → sélectionner un abonnement → Contrôle d'accès (IAM) → Ajouter une attribution de rôle → sélectionner Azure Kubernetes Service RBAC Reader → attribuer à l'application

Cela couvre tous les clusters AKS inclus dans l'abonnement.

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

Utilisez-le si votre cluster utilise déjà la configuration d'authentification Azure AD par défaut et que vous préférez ne pas activer Azure RBAC. Cela nécessite une `kubectl` configuration par cluster.

1. Enregistrez le manifeste suivant sous le nom `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>Remplacez-le par l'ID d'objet de votre directeur de service. Pour le trouver : Portail Azure → Entra ID → Applications d'entreprise → recherchez le nom de l'application (soit AWS DevOps Agent pour le consentement de l'administrateur, soit votre propre application Entra pour l'enregistrement des applications).
2. Appliquer à chaque cluster :

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

> Remarque : Les clusters utilisant uniquement des comptes locaux (sans Azure AD) ne sont pas pris en charge. Nous vous recommandons d'activer l'intégration Azure AD sur votre cluster pour utiliser cette fonctionnalité.

Rôle personnalisé le moins privilégié (facultatif)

Pour un contrôle d'accès plus strict, vous pouvez créer un rôle Azure personnalisé limité uniquement aux fournisseurs de ressources utilisés par l' AWS DevOps agent, au lieu du rôle général de lecteur :

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

Associer un abonnement à un espace agent

Après avoir enregistré Azure au niveau du compte, associez des abonnements spécifiques à vos espaces d'agent :

1. Dans la console AWS DevOps Agent, sélectionnez votre espace agent
2. Accédez à l'onglet Fonctionnalités
3. Dans la section Sources secondaires, cliquez sur Ajouter
4. Sélectionnez Azure

5. Fournissez l'ID d'abonnement pour l'abonnement Azure que vous souhaitez associer
6. Cliquez sur Ajouter pour terminer l'association

Vous pouvez associer plusieurs abonnements au même espace agent pour donner à l'agent une visibilité sur l'ensemble de votre environnement Azure.

Gestion des connexions Azure Resources

- Affichage des abonnements connectés : dans l'onglet Fonctionnalités, la section Sources secondaires répertorie tous les abonnements Azure connectés.
- Supprimer un abonnement : pour déconnecter un abonnement d'un espace agent, sélectionnez-le dans la liste des sources secondaires et cliquez sur Supprimer. Cela n'affecte pas l'enregistrement au niveau du compte.
- Suppression de l'enregistrement : pour supprimer complètement l'enregistrement Azure Cloud, rendez-vous sur la page Capability Providers et supprimez l'enregistrement. Toutes les associations Agent Space doivent d'abord être supprimées.

Connecter Azure DevOps

DevOps L'intégration Azure permet à l' AWS DevOps agent d'accéder aux référentiels et à l'historique d'exécution du pipeline dans votre DevOps organisation Azure. L'agent peut corréler les modifications de code et les déploiements avec les incidents opérationnels pour aider à identifier les causes profondes potentielles.

> Remarque : les DevOps pipelines Azure peuvent utiliser le code source d'Azure Repos ou de Bitbucket. GitHub L' DevOps intégration Azure permet d'accéder à l'historique d'exécution du pipeline quel que soit le fournisseur source. Toutefois, pour accéder au code source réel pendant les enquêtes, le référentiel doit être connecté séparément via une intégration prise en charge telle quethe section called “Connecter GitHub”. Le code source de Bitbucket n'est pas directement accessible via cette intégration.

Cette intégration suit un processus en deux étapes : enregistrer Azure DevOps au niveau du AWS compte, puis associer des projets spécifiques à des espaces d'agent individuels.

Conditions préalables

Avant de connecter Azure DevOps, assurez-vous d'avoir :

- Accès à la console de AWS DevOps l'agent
- Une DevOps organisation Azure avec au moins un projet contenant un historique de référentiel et de pipeline
- Autorisations pour ajouter des utilisateurs à votre DevOps organisation Azure
- Pour la méthode de consentement de l'administrateur : un compte autorisé à effectuer le consentement de l'administrateur dans le Microsoft Entra ID
- Pour la méthode d'enregistrement des applications : une application Entra autorisée à configurer les informations d'identification fédérées, et la fédération d'identité sortante activée sur votre compte AWS

> Remarque : Vous pouvez également démarrer l'enregistrement depuis un espace agent. Accédez à la section Pipelines, cliquez sur Ajouter, puis sélectionnez Azure DevOps. Si Azure n' DevOps est pas encore enregistré, la console vous guide d'abord tout au long de l'enregistrement.

Enregistrement d'Azure DevOps via le consentement de l'administrateur

La méthode Admin Consent utilise un flux basé sur le consentement avec l'application gérée par l' AWS DevOps agent.

Étape 1 : Commencez l'enregistrement

1. Connectez-vous à la console AWS de gestion et accédez à la console de l' AWS DevOps agent
2. Accédez à la page Capability Providers
3. Localisez la DevOps section Azure et cliquez sur Enregistrer
4. Entrez le nom de votre DevOps organisation Azure lorsque vous y êtes invité

Étape 2 : Compléter le consentement de l'administrateur

1. Cliquez pour continuer - vous êtes redirigé vers la page de consentement de l'administrateur Microsoft Entra
2. Connectez-vous avec un compte utilisateur principal autorisé à obtenir le consentement de l'administrateur
3. Examiner la demande d' AWS DevOps agent et donner son accord

Étape 3 : Autorisation complète de l'utilisateur

1. Après le consentement de l'administrateur, vous êtes invité à obtenir l'autorisation de l'utilisateur pour vérifier votre identité en tant que membre du locataire autorisé
2. Connectez-vous avec un compte appartenant au même client Azure
3. Après autorisation, vous êtes redirigé vers la console de l' AWS DevOps agent avec un statut de réussite

Étape 4 : accorder l'accès dans Azure DevOps

Consultez la section Octroi d'accès dans Azure DevOps ci-dessous. Recherchez AWS DevOps Agent lors de l'ajout d'utilisateurs.

Enregistrement d'Azure DevOps via l'enregistrement d'applications

L'enregistrement des applications est partagé entre Azure Resources et Azure DevOps. Si vous avez déjà terminé l'enregistrement des applications pour Azure Resources, vous pouvez passer à la section Accorder l'accès dans Azure DevOps.

Étape 1 : démarrer l'enregistrement de l'application ADO

1. Dans la console de l' AWS DevOps agent, accédez à la page Capability Providers
2. Localisez la section Azure Cloud et cliquez sur Enregistrer
3. Sélectionnez la méthode d'enregistrement de l'application

Étape 2 : Créez et configurez votre application Entra

Suivez les instructions affichées dans la console pour :

1. Activez la fédération des identités sortantes dans votre AWS compte (dans la console IAM, allez dans Paramètres du compte → Fédération des identités sortantes)
2. Créez une application Entra dans votre identifiant Microsoft Entra ou utilisez-en une existante
3. Configurer les informations d'identification fédérées sur l'application

Étape 3 : Fournissez les informations d'enregistrement

Remplissez le formulaire d'inscription avec :

- ID de locataire : votre identifiant de locataire Azure
- Nom du locataire : nom d'affichage du locataire
- ID client — L'identifiant de l'application (client) de l'application Entra
- Audience : identifiant d'audience pour l'identifiant fédéré

Étape 4 : Création du rôle IAM

Un rôle IAM est automatiquement créé lorsque vous soumettez l'enregistrement via la console. Cela permet à AWS DevOps l'agent d'assumer les informations d'identification et d'appeler`sts:GetWebIdentityToken`.

Étape 5 : Compléter l'enregistrement

1. Confirmez la configuration dans la console de AWS DevOps l'agent
2. Cliquez sur Soumettre pour terminer l'enregistrement

Étape 6 : accorder l'accès dans Azure DevOps

Consultez la section Octroi d'accès dans Azure DevOps ci-dessous. Recherchez l'application Entra que vous avez créée lors de l'enregistrement de l'application lors de l'ajout d'utilisateurs.

Octroi d'accès dans Azure DevOps

Après l'enregistrement, accordez à l'application l'accès à votre DevOps organisation Azure. Cette étape est la même pour les méthodes de consentement de l'administrateur et d'enregistrement de l'application.

1. Dans Azure DevOps, accédez à Paramètres de l'organisation > Utilisateurs > Ajouter des utilisateurs
2. Recherchez l'application (soit AWS DevOps agent pour le consentement de l'administrateur, soit votre propre application Entra pour l'enregistrement de l'application)
3. Définissez le niveau d'accès sur Basic
4. Sous Ajouter aux projets, sélectionnez les projets auxquels vous souhaitez que l'agent accède
5. Sous Azure DevOps Groups, sélectionnez Project Readers
6. Cliquez sur Ajouter pour terminer

Exigence de sécurité : Attribuez uniquement le groupe des lecteurs de projet. L'accès en lecture seule constitue une limite de sécurité qui limite l'agent aux opérations en lecture seule et limite l'impact des attaques indirectes par injection rapide. L'attribution à des groupes d'autorisations d'écriture ou d'action augmente considérablement le rayon d'action d'injection rapide et peut compromettre les DevOps ressources Azure.

Associer un projet à un agent Space

Après avoir enregistré Azure DevOps au niveau du compte, associez des projets spécifiques à vos espaces d'agent :

1. Dans la console AWS DevOps Agent, sélectionnez votre espace agent
2. Accédez à l'onglet Fonctionnalités
3. Dans la section Pipelines, cliquez sur Ajouter
4. Sélectionnez Azure DevOps dans la liste des fournisseurs disponibles
5. Sélectionnez le projet dans le menu déroulant des projets disponibles
6. Cliquez sur Ajouter pour terminer l'association

Gestion des DevOps connexions Azure

- Affichage des projets connectés : dans l'onglet Fonctionnalités, la section Pipelines répertorie tous les DevOps projets Azure connectés.
- Supprimer un projet : pour déconnecter un projet d'un espace agent, sélectionnez-le dans la section Pipelines et cliquez sur Supprimer.
- Suppression de l'enregistrement : pour supprimer complètement l' DevOps enregistrement Azure, rendez-vous sur la page Capability Providers et supprimez l'enregistrement. Toutes les associations Agent Space doivent d'abord être supprimées.

Raccordement aux CI/CD pipelines

L'intégration du pipeline CI/CD permet à l' AWS DevOps agent de surveiller les déploiements et de corréler les modifications de code avec les incidents opérationnels au cours des enquêtes. En connectant vos CI/CD fournisseurs, l'agent peut suivre les événements de déploiement et les associer à AWS des ressources pour aider à identifier les causes profondes potentielles lors de la réponse aux incidents.

AWS DevOps L'agent prend en charge l'intégration avec les CI/CD plateformes populaires grâce à un processus en deux étapes :

1. Enregistrement au niveau du compte — Enregistrez votre CI/CD fournisseur une fois au niveau du AWS compte
2. Connexion à l'espace agent : connectez des projets ou des référentiels spécifiques à des espaces d'agent individuels en fonction des besoins de votre organisation

Cette approche vous permet de partager les inscriptions des CI/CD fournisseurs entre plusieurs espaces d'agents tout en gardant un contrôle précis sur les projets surveillés par chaque espace.

CI/CD Fournisseurs pris en charge

AWS DevOps L'agent prend en charge les CI/CD plateformes suivantes :

- GitHub— Connectez les référentiels depuis GitHub.com à l'aide de l' GitHub application AWS DevOps Agent.
- GitLab— Connectez des projets à partir de GitLab.com, d' GitLab instances gérées ou de GitLab déploiements auto-hébergés accessibles au public.

Rubriques

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

Connecter GitHub

GitHub l'intégration permet à l' AWS DevOps agent d'accéder aux référentiels de code et de recevoir les événements de déploiement lors des enquêtes sur les incidents. Cette intégration suit un processus en deux étapes : enregistrement au niveau du compte GitHub, suivi de la connexion de référentiels spécifiques à des espaces d'agent individuels.

AWS DevOps L'agent prend en charge les instances GitHub .com (SaaS) et GitHub Enterprise Server (auto-hébergées).

Conditions préalables

Avant de vous connecter GitHub, assurez-vous d'avoir :

- Accès à la console d'administration de AWS DevOps l'agent
- Un compte GitHub utilisateur ou une organisation avec des autorisations d'administrateur
- Autorisation d'installer GitHub des applications dans votre compte ou votre organisation

Pour GitHub Enterprise Server, vous devez également :

- Une instance de serveur GitHub d'entreprise (version 3.x ou ultérieure) accessible via HTTPS
- L'URL HTTPS de votre instance de serveur GitHub d'entreprise (par exemple,`https://github.example.com`)
- (Facultatif) Une connexion privée, si votre instance de serveur GitHub d'entreprise n'est pas accessible au public

Inscription GitHub (au niveau du compte)

GitHub est enregistré au niveau du AWS compte et partagé entre tous les espaces d'agent de ce compte. Vous ne devez vous inscrire qu' GitHub une seule fois par AWS compte.

Étape 1 : Accédez aux fournisseurs de pipelines

1. Connectez-vous à la console AWS de gestion
2. Accédez à la console de AWS DevOps l'agent
3. Accédez à l'onglet Fonctionnalités
4. Dans la section Pipeline, cliquez sur Ajouter
5. Sélectionnez GitHubdans la liste des fournisseurs disponibles

Si ce GitHub n'est pas encore le cas, il vous sera demandé de l'enregistrer d'abord.

Étape 2 : Choisissez le type de connexion

Sur l'écran « Enregistrer un GitHub compte/une organisation », indiquez si vous vous connectez en tant qu'utilisateur ou en tant qu'organisation :

- Utilisateur — Votre GitHub compte personnel avec un nom d'utilisateur et un profil
- Organisation — Un GitHub compte partagé où plusieurs personnes peuvent collaborer sur de nombreux projets à la fois

Si vous vous connectez à une instance de serveur GitHub d'entreprise, cochez la case Utiliser un serveur GitHub d'entreprise et entrez l'URL HTTPS de votre instance (par exemple,`https://github.example.com`).

Si votre instance GitHub Enterprise Server n'est pas accessible au public, vous pouvez éventuellement configurer une connexion privée pour permettre à l' AWS DevOps agent d'accéder à votre instance en toute sécurité. Pour de plus amples informations, veuillez consulter the section called “Connexion à des outils hébergés en privé”.

> **Note**
>
> N'incluez `/api/v3` aucun chemin de fin dans l'URL. Entrez uniquement l'URL de base.

Étape 3 : configurer l' GitHub application

Cliquez sur Soumettre pour démarrer le processus de configuration de l'application. Les étapes suivantes varient selon que vous vous connectez à GitHub .com ou à GitHub Enterprise Server.

Pour GitHub .com

1. Vous serez redirigé GitHub vers l' GitHub application AWS DevOps Agent pour y installer.
2. Sélectionnez le compte ou l'organisation dans lequel vous souhaitez installer l'application.
3. L'application permet à l' AWS DevOps agent de recevoir des événements provenant de référentiels connectés, y compris des événements de déploiement.

Pour GitHub Enterprise Server

GitHub Enterprise Server utilise un flux GitHub App Manifest, qui configure automatiquement une nouvelle GitHub application sur votre instance. Cela implique deux redirections vers votre instance de serveur GitHub d'entreprise.

1. Votre navigateur sera redirigé vers la page « Créer une GitHub application » de votre instance GitHub Enterprise Server.
2. Vous verrez le nom de l'application prérempli. N'hésitez pas à modifier le nom selon vos besoins. Cliquez sur Créer une GitHub application.
3. Vous serez redirigé vers l' AWS DevOps Agent, qui échange le code manifeste contre les informations d'identification de l'application.

Étape 4 : sélectionner les référentiels et terminer l'installation

1. Vous verrez la page d'installation et d'autorisation de l' GitHub application.
2. Sélectionnez les référentiels auxquels l'application doit accéder :
   - Tous les référentiels : accordez l'accès à tous les référentiels actuels et futurs
   - Sélectionnez uniquement les référentiels : choisissez des référentiels spécifiques à partir de votre compte ou de votre organisation
3. Cliquez sur Installer et autoriser.
4. Vous serez redirigé vers la console de l' AWS DevOps agent, où GitHub vous serez enregistré au niveau du compte.

Connexion de référentiels à un espace d'agents

Une fois enregistré GitHub au niveau du compte, vous pouvez connecter des référentiels spécifiques à des espaces d'agent individuels :

1. Dans la console AWS DevOps Agent, sélectionnez votre espace agent
2. Accédez à l'onglet Fonctionnalités
3. Dans la section Pipeline, cliquez sur Ajouter
4. Sélectionnez GitHubdans la liste des fournisseurs disponibles
5. Sélectionnez le sous-ensemble de référentiels correspondant à cet espace agent
6. Cliquez sur Ajouter pour terminer la connexion

Vous pouvez connecter différents ensembles de référentiels à différents espaces d'agent en fonction des besoins de votre organisation.

Comprendre l' GitHub application

L' GitHub application AWS DevOps Agent :

- Demande un accès en lecture seule à vos référentiels
- Reçoit les événements de déploiement et autres événements du référentiel
- Permet à AWS DevOps l'agent de corréler les modifications de code avec les incidents opérationnels

- Peut être désinstallé à tout moment via vos GitHub paramètres

Pour GitHub Enterprise Server, l' GitHub application est automatiquement créée sur votre instance lors de l'enregistrement. Vous pouvez gérer l'accès au référentiel de l'application ou la désinstaller via Paramètres > Applications > GitHub Applications installées. Pour supprimer complètement la définition de l'application, accédez à Réglages > Paramètres du développeur > GitHub Applications.

Gestion des GitHub connexions

- Mise à jour de l'accès aux référentiels : pour modifier les référentiels auxquels GitHub l'application peut accéder, accédez aux paramètres de votre GitHub compte ou de votre organisation (ou aux paramètres de votre instance GitHub Enterprise Server), accédez aux GitHub applications installées et modifiez la configuration de l'application AWS DevOps Agent.
- Affichage des référentiels connectés : dans la console de l' AWS DevOps agent, sélectionnez votre espace agent et accédez à l'onglet Fonctionnalités pour afficher les référentiels connectés dans la section Pipeline.
- Suppression de la GitHub connexion : pour vous déconnecter GitHub d'un espace agent, sélectionnez la connexion dans la section Pipeline et cliquez sur Supprimer. Pour désinstaller complètement l' GitHub application, désinstallez-la dans les paramètres de votre GitHub compte ou de votre organisation. Pour GitHub Enterprise Server, étant donné que l' GitHub application est créée directement sur votre instance lors de l'enregistrement, vous pouvez éventuellement nettoyer entièrement l'application en effectuant les deux opérations suivantes :
  - Désinstallez l'application : accédez à Paramètres > Applications > GitHub Applications installées, cliquez sur Configurer dans l'application, puis désinstallez-la.
  - Supprimer l'application : accédez à Réglages > Paramètres du développeur > GitHub Applications, sélectionnez l'application, accédez à l'onglet Avancé, puis choisissez Supprimer GitHub l'application. Avertissement : La suppression de l' GitHub application est définitive et ne peut pas être annulée. Si vous le supprimez, vous devrez réenregistrer GitHub Enterprise Server depuis le début dans la console de l' AWS DevOps agent pour créer une nouvelle application.

Connecter GitLab

GitLab l'intégration permet à l' AWS DevOps agent de surveiller les déploiements à partir de GitLab pipelines afin d'éclairer les enquêtes causales lors de la réponse à un incident. Cette intégration suit un processus en deux étapes : enregistrement au niveau du compte GitLab, suivi de la connexion de projets spécifiques à des espaces d'agent individuels.

Inscription GitLab (au niveau du compte)

GitLab est enregistré au niveau du AWS compte et partagé entre tous les espaces d'agent de ce compte. Les espaces d'agent individuels peuvent ensuite choisir les projets spécifiques qui s'appliquent à leur espace d'agent.

Étape 1 : Accédez aux fournisseurs de pipelines

1. Connectez-vous à la console AWS de gestion
2. Accédez à la console de AWS DevOps l'agent
3. Accédez à la page Capability Providers (accessible depuis la navigation latérale)
4. Recherchez GitLabdans la section Fournisseurs disponibles sous Pipeline et cliquez sur Enregistrer

Étape 2 : Configuration de la GitLab connexion

Sur la page GitLab d'enregistrement, configurez les éléments suivants :

Type de connexion : indiquez si vous vous connectez en tant que personne ou en tant que groupe :

- Personnel (par défaut) — Votre compte GitLab utilisateur individuel avec un nom d'utilisateur et un profil
- Groupe — Dans GitLab, vous utilisez des groupes pour gérer un ou plusieurs projets connexes en même temps

GitLab type d'instance — Choisissez le type d' GitLab instance auquel vous vous connectez :

- GitLab.com (par défaut) — Le GitLab service public
- Auto-hébergé accessible au public GitLab : cochez la case Utiliser un point de terminaison GitLab auto-hébergé et fournissez l'URL de votre instance GitLab

> **Note**
>
> Actuellement, seules les GitLab instances accessibles au public sont prises en charge.

Jeton d'accès — Fournissez un jeton d'accès GitLab personnel :

1. Dans un onglet de navigateur distinct, connectez-vous à votre GitLab compte
2. Accédez à vos paramètres utilisateur et sélectionnez Access Tokens
3. Créez un nouveau jeton d'accès personnel avec les autorisations suivantes :
   - `read_repository`— Nécessaire pour accéder au contenu du référentiel
   - `read_virtual_registry`— Nécessaire pour accéder aux informations du registre virtuel
   - `read_registry`— Nécessaire pour accéder aux informations du registre
   - `api`— Nécessaire pour accéder à l'API en lecture et en écriture
   - `self_rotate`- Nécessaire pour la rotation des jetons. Cette fonctionnalité n'est actuellement pas prise en charge par l' AWS DevOps Agent mais le sera ultérieurement. En ajoutant maintenant, il n'est plus nécessaire de créer un nouveau jeton dans le futur.
4. Définissez l'expiration du jeton sur un maximum de 365 jours à compter de la date actuelle
5. Copiez le jeton généré
6. Retournez à la console de AWS DevOps l'agent
7. Collez le jeton dans le champ « Jeton d'accès »

Étape 3 : Compléter l'enregistrement

Tags (Facultatif) — Ajoutez des AWS tags à l' GitLab enregistrement à des fins d'organisation.

Cliquez sur Suivant pour vérifier votre configuration, puis sur Soumettre pour terminer le processus GitLab d'enregistrement. Le système validera votre jeton d'accès et établira la connexion.

Connecter des projets à un espace d'agents

Après vous être enregistré GitLab au niveau du compte, vous pouvez connecter des projets spécifiques à des espaces d'agent individuels :

1. Dans la console AWS DevOps Agent, sélectionnez votre espace agent
2. Accédez à l'onglet Fonctionnalités
3. Dans la section Pipeline, cliquez sur Ajouter
4. Sélectionnez GitLabdans la liste des fournisseurs disponibles
5. Sélectionnez les GitLab projets pertinents pour votre espace d'agents
6. Cliquez sur Enregistrer

AWS DevOps L'agent surveillera ces projets pour détecter les déploiements à partir de GitLab pipelines afin d'éclairer les enquêtes causales.

Gestion des GitLab connexions

- Mise à jour du jeton d'accès : si votre jeton d'accès expire ou doit être mis à jour, vous pouvez le mettre à jour dans la console de l' AWS DevOps agent en modifiant l' GitLab enregistrement au niveau du compte.
- Affichage des projets connectés : dans la console AWS DevOps Agent, sélectionnez votre espace agent et accédez à l'onglet Fonctionnalités pour afficher les projets connectés dans la section Pipeline.
- Suppression de la GitLab connexion : pour déconnecter GitLab des projets d'un espace agent, sélectionnez la connexion dans la section Pipeline et cliquez sur Supprimer. Pour supprimer complètement l' GitLab enregistrement, supprimez-le d'abord de tous les agents Spaces, puis supprimez l'enregistrement au niveau du compte.

Connexion de serveurs MCP

Les serveurs MCP (Model Context Protocol) étendent les capacités d'investigation de l' AWS DevOps agent en fournissant un accès aux données provenant de vos outils d'observabilité externes, de vos systèmes de surveillance personnalisés et de vos sources de données opérationnelles. Ce guide explique comment connecter un serveur MCP à l' AWS DevOps agent.

Exigences

Avant de connecter un serveur MCP, assurez-vous que celui-ci répond aux exigences suivantes :

- Protocole de transport HTTP streamable : seuls les serveurs MCP implémentant le protocole de transport HTTP Streamable sont pris en charge.
- Prise en charge de l'authentification : votre serveur MCP doit prendre en charge les flux d'authentification OAuth 2.0 ou l'authentification basée sur une clé API/un jeton.

Considérations sur la sécurité

Lorsque vous connectez des serveurs MCP à l' AWS DevOps agent, tenez compte des aspects de sécurité suivants :

- Liste des outils autorisés : vous devez uniquement autoriser les outils spécifiques dont votre agent Space a besoin, plutôt que d'exposer tous les outils de votre serveur MCP. Voir Configuration des outils MCP dans un espace d'agent pour savoir comment autoriser les outils de liste par espace d'agent.

Veuillez noter que la longueur maximale d'un outil MCP est de 64.

- Risques d'injection rapide — Les serveurs MCP personnalisés peuvent introduire un risque supplémentaire d'attaques par injection rapide. Voir Protection contre les injections rapides : sécurité des AWS DevOps agents pour plus d'informations.
- Outils et accès en lecture seule : autorisez uniquement la liste des outils MCP en lecture seule et assurez-vous que les informations d'authentification ne sont autorisées qu'en lecture seule.

Consultez AWS DevOps Sécurité des agents pour plus d'informations sur l'injection rapide et le modèle de responsabilité partagée.

> **Note**
>
> Si votre serveur MCP se trouve sur un réseau privé, voir the section called “Connexion à des outils hébergés en privé”

Enregistrement d'un serveur MCP (au niveau du compte)

Les serveurs MCP sont enregistrés au niveau du AWS compte et partagés entre tous les agents Spaces de ce compte. Chaque agent Spaces peut ensuite choisir les outils spécifiques dont il a besoin sur chaque serveur MCP.

Étape 1 : détails du serveur MCP

1. Connectez-vous à la console AWS de gestion
2. Accédez à la console de AWS DevOps l'agent
3. Accédez à la page Capability Providers (accessible depuis la navigation latérale)
4. Trouvez le serveur MCP dans la section Fournisseurs disponibles et cliquez sur Enregistrer
5. Sur la page de détails du serveur MCP, entrez les informations suivantes :
   - Nom — Entrez un nom descriptif pour votre serveur MCP

- URL du point de terminaison : entrez l'URL HTTPS complète du point de terminaison de votre serveur MCP
- Description (facultatif) — Ajoutez une description pour aider à identifier l'objectif du serveur
- Activer l'enregistrement dynamique des clients : cochez cette case si vous souhaitez autoriser l' AWS DevOps agent à s'enregistrer automatiquement auprès du serveur d'autorisation de votre serveur MCP

6. Cliquez sur Suivant

> **Note**
>
> L'URL du point de terminaison du serveur MCP sera affichée dans AWS CloudTrail les journaux de votre compte.

Étape 2 : flux d'autorisation

Sélectionnez la méthode d'authentification pour votre serveur MCP :

OAuth Informations d'identification du client : si votre serveur MCP utilise le flux d'informations d'identification du OAuth client :

1. Sélectionnez les informations d'identification OAuth du client
2. Cliquez sur Suivant

OAuth 3LO (Three-Legged OAuth) — Si votre serveur MCP utilise OAuth 3LO pour l'authentification :

1. Sélectionnez OAuth 3LO
2. Cliquez sur Suivant

Clé d'API — Si votre serveur MCP utilise l'authentification par clé d'API :

1. Sélectionnez la clé d'API
2. Cliquez sur Suivant

Étape 3 : Configuration de l'autorisation

Configurez des paramètres d'autorisation supplémentaires en fonction de la méthode d'authentification sélectionnée :

Pour les informations d'identification OAuth du client :

1. ID client — Entrez l'ID client du OAuth client
2. Secret client — Entrez le secret client du OAuth client
3. URL d'échange — Entrez l'URL du point de terminaison de l'échange de OAuth jetons
4. Paramètres d'échange — Entrez les paramètres d'échange de OAuth jetons pour vous authentifier auprès du service
5. Ajouter une étendue — Ajouter des OAuth étendues pour l'authentification
6. Cliquez sur Suivant

Pour OAuth 3LO :

1. ID client — Entrez l'ID client du OAuth client
2. Secret du client — Entrez le secret du OAuth client s'il est exigé par votre OAuth client
3. URL d'échange — Entrez l'URL du point de terminaison de l'échange de OAuth jetons
4. URL d'autorisation : entrez l'URL du point de terminaison OAuth d'autorisation
5. Support pour les défis de code : cochez cette case si votre OAuth client prend en charge les défis de code
6. Ajouter une étendue — Ajouter des OAuth étendues pour l'authentification
7. Cliquez sur Suivant

Pour la clé API :

1. Entrez un nom de clé d'API
2. Entrez le nom de l'en-tête qui contiendra la clé d'API dans la demande
3. Entrez la valeur de votre clé d'API
4. Cliquez sur Suivant

Étape 4 : Réviser et soumettre

1. Passez en revue tous les détails de configuration du serveur MCP
2. Cliquez sur Soumettre pour terminer l'enregistrement
3. AWS DevOps L'agent validera la connexion à votre serveur MCP
4. Une fois la validation réussie, votre serveur MCP sera enregistré au niveau du compte

Configuration des outils MCP dans un espace d'agents

Après avoir enregistré un serveur MCP au niveau du compte, vous pouvez configurer les outils disponibles sur ce serveur pour des agents Spaces spécifiques :

1. Dans la console AWS DevOps Agent, sélectionnez votre espace agent
2. Accédez à l'onglet Fonctionnalités
3. Dans la section Serveurs MCP, cliquez sur Ajouter
4. Sélectionnez le serveur MCP enregistré que vous souhaitez connecter à cet agent Space
5. Configurez les outils de ce serveur MCP qui doivent être accessibles à l'espace agent :
   - Autoriser tous les outils : rend disponibles tous les outils du serveur MCP
   - Sélectionnez des outils spécifiques : vous permet de choisir les outils à autoriser dans la liste
6. Cliquez sur Ajouter pour connecter le serveur MCP à votre agent Space

AWS DevOps L'agent pourra désormais utiliser les outils autorisés de votre serveur MCP lors d'enquêtes dans cet espace d'agents.

Gestion des connexions au serveur MCP

Mise à jour des informations d'authentification — Si vos informations d'authentification doivent être mises à jour, vous devrez réenregistrer votre serveur MCP. Accédez à la page Capability Providers dans la console de l' AWS DevOps agent, localisez votre serveur MCP, supprimez toutes les associations actives, puis cliquez sur Désenregistrer. Enregistrez ensuite votre serveur MCP avec les nouvelles informations d'authentification et recréez toutes les associations nécessaires avec votre espace agent.

Affichage des serveurs MCP connectés : pour voir tous les serveurs MCP connectés à votre espace agent, sélectionnez votre espace agent, accédez à l'onglet Fonctionnalités et consultez la section Serveurs MCP. Vous pouvez également mettre à jour les outils sélectionnés ici.

Suppression des connexions au serveur MCP — Pour déconnecter un serveur MCP d'un espace agent, sélectionnez le serveur dans la section Serveurs MCP et cliquez sur Supprimer. Pour supprimer complètement un enregistrement de serveur MCP, supprimez-le d'abord de tous les agents Spaces, puis supprimez l'enregistrement au niveau du compte.

Rubriques en relation

- Sécurité dans l' AWS DevOps agent
- Configuration d'un espace d'agents
- Protection contre les injections rapides

Connexion de plusieurs AWS comptes

AWS Les comptes secondaires permettent à AWS DevOps l'agent d'examiner les ressources de plusieurs AWS comptes de votre organisation. Lorsque vos applications s'étendent sur plusieurs comptes, l'ajout de comptes secondaires garantit à l'agent une visibilité sur toutes les ressources pertinentes lors des enquêtes sur les incidents. Un meilleur accès aux comptes et aux ressources composant une application garantit une plus grande précision des enquêtes.

Conditions préalables

Avant d'ajouter un AWS compte secondaire, assurez-vous d'avoir :

- Accès à la console de AWS DevOps l'agent dans le compte principal
- Accès administratif au AWS compte secondaire
- Autorisations IAM pour créer des rôles dans le compte secondaire

Ajouter un AWS compte secondaire

Outre les étapes ci-dessous, vous pouvez utiliser le the section called “AWS DevOps Guide d'intégration de l'Agent CLI” pour ajouter des comptes secondaires par programmation.

Étape 1 : démarrer la configuration du compte secondaire

1. Connectez-vous à la console AWS de gestion et accédez à la console de l' AWS DevOps agent
2. Sélectionnez votre espace d'agent

3. Accédez à l'onglet Fonctionnalités
4. Dans la section Cloud, recherchez la sous-section Sources secondaires
5. Cliquez sur Ajouter

Étape 2 : Spécifiez le nom du rôle

1. Dans le champ Nommez votre rôle, saisissez le nom du rôle que vous allez créer dans le compte secondaire
2. Notez ce nom : vous l'utiliserez à nouveau lors de la création du rôle dans le compte secondaire
3. Copiez la politique de confiance fournie dans la console et enregistrez-la dans un espace de travail

Étape 3 : créer le rôle dans le compte secondaire

1. Ouvrez un nouvel onglet de navigateur et connectez-vous à la console IAM dans le compte secondaire AWS
2. Accédez à IAM > Rôles > Créer un rôle
3. Sélectionnez une politique de confiance personnalisée
4. Collez la politique de confiance que vous avez copiée à partir de l'étape 2
5. Cliquez sur Suivant

Étape 4 : joindre la politique AWS gérée

1. Dans la section Politiques d'autorisations, recherchez AIOpsAssistantPolicy
2. Cochez la case à côté de la politique AIOpsAssistantPolicygérée
3. Cliquez sur Suivant

Étape 5 : Nommez et créez le rôle

1. Dans le champ Nom du rôle, entrez le même nom de rôle que celui que vous avez indiqué à l'étape 2
2. (Facultatif) Ajoutez une description pour aider à identifier l'objectif du rôle
3. Passez en revue la politique de confiance et les autorisations associées
4. Cliquez sur Créer un rôle

Étape 6 : Joindre la politique en ligne

1. Dans la console IAM, recherchez et sélectionnez le rôle que vous venez de créer
2. Accédez à l'onglet Autorisations
3. Cliquez sur Ajouter des autorisations > Créer une politique en ligne
4. Passez à l'onglet JSON
5. Collez la politique que vous avez enregistrée à l'étape 2
6. Collez la politique dans l'éditeur JSON de la console IAM
7. Cliquez sur Suivant
8. Donnez un nom à la politique intégrée (par exemple, DevOpsAgentInlinePolicy « »)
9. Cliquez sur Créer une politique

Étape 7 : terminer la configuration

1. Retournez à la console de l' AWS DevOps agent dans le compte principal
2. Cliquez sur Suivant pour terminer la configuration du compte secondaire
3. Vérifiez que l'état de la connexion est indiqué comme actif

Comprendre les politiques requises

AWS DevOps L'agent a besoin de trois composants de politique pour accéder aux ressources d'un compte secondaire :

- Politique de confiance — Permet à l' AWS DevOps agent du compte principal d'assumer le rôle du compte secondaire. Cela établit la relation de confiance entre les comptes.
- AIOpsAssistantPolicy (politique AWS gérée) — Fournit les autorisations de lecture seule de base dont AWS DevOps l'agent a besoin pour étudier les ressources du compte secondaire. Cette politique est maintenue AWS et mise à jour au fur et à mesure que de nouvelles fonctionnalités sont ajoutées.
- Politique intégrée : fournit des autorisations supplémentaires spécifiques à la configuration de votre espace agent. Cette politique est générée en fonction des paramètres de votre espace agent et peut inclure des autorisations pour des intégrations ou des fonctionnalités spécifiques.

Dans le compte principal, le rôle d' AWS DevOps agent IAM doit pouvoir assumer le rôle créé dans le compte secondaire.

Gestion des comptes secondaires

- Affichage des comptes connectés : dans l'onglet Fonctionnalités, la sous-section Sources secondaires répertorie tous les comptes secondaires connectés avec leur état de connexion.
- Mise à jour du rôle IAM : si vous devez modifier les autorisations, mettez à jour la politique intégrée associée au rôle dans le compte secondaire. Les modifications prennent effet immédiatement.
- Suppression d'un compte secondaire : pour déconnecter un compte secondaire, sélectionnez-le dans la liste des sources secondaires et cliquez sur Supprimer. Cela ne supprime pas le rôle IAM dans le compte secondaire.

Connexion de sources de télémétrie

AWS DevOps L'agent propose trois méthodes pour se connecter à vos sources de télémétrie.

Intégration bidirectionnelle intégrée

Actuellement, AWS DevOps Agent prend en charge les utilisateurs de Dynatrace grâce à une intégration bidirectionnelle intégrée permettant ce qui suit :

- Cartographie des ressources topologiques - AWS DevOps L'agent augmentera la topologie de votre espace DevOps agent avec des entités et des relations disponibles via un serveur Dynatrace MCP hébergé par l' AWS DevOps agent.
- Déclenchement automatique des enquêtes - Les flux de travail Dynatrace peuvent être configurés pour déclencher des enquêtes de résolution d'incidents liés à des problèmes Dynatrace.
- Introspection de la télémétrie : l' AWS DevOps agent peut introspecter la télémétrie Dynatrace lorsqu'il étudie un problème via le serveur Dynatrace MCP hébergé par l'agent. AWS DevOps
- Mises à jour de statut - AWS DevOps L'agent publiera les principaux résultats de l'enquête, les analyses des causes profondes et les plans d'atténuation générés sur l'interface utilisateur de Dynatrace.

Pour en savoir plus sur les intégrations bidirectionnelles, voir

- the section called “Connecter Dynatrace”

Intégration unidirectionnelle intégrée

Actuellement, AWS DevOps Agent prend en charge les utilisateurs de Datadog AWS CloudWatch, Grafana, New Relic et Splunk grâce à des intégrations unidirectionnelles intégrées.

Bonnes pratiques en matière de sécurité : lors de la configuration des informations d'identification pour les intégrations unidirectionnelles intégrées, nous vous recommandons de définir les clés d'API et les jetons pour un accès en lecture seule. AWS DevOps L'agent utilise ces informations d'identification uniquement pour l'introspection télémétrique et ne nécessite pas d'accès en écriture à votre fournisseur de télémétrie.

L'intégration unidirectionnelle AWS CloudWatch intégrée ne nécessite aucune configuration supplémentaire et permet ce qui suit :

- Cartographie des ressources topologiques - AWS DevOps L'agent augmentera la topologie de votre espace DevOps agent avec les entités et les relations disponibles via vos comptes cloud principaux et secondaires AWS configurés.
- Introspection de la télémétrie : l' AWS DevOps agent peut effectuer une introspection de la AWS CloudWatch télémétrie lorsqu'il étudie un problème via le ou les rôles IAM fournis lors de la configuration du compte cloud principal et secondaire. AWS

Les intégrations unidirectionnelles intégrées à Datadog, Grafana, New Relic et Splunk nécessitent une configuration et permettent les fonctionnalités suivantes :

- Déclenchement automatique des enquêtes : les événements Datadog, Grafana, New Relic et Splunk peuvent être configurés pour déclencher des enquêtes de résolution d'incidents par des agents via des webhooks d' AWS DevOps agents. AWS DevOps
- Introspection de la télémétrie : l' AWS DevOps agent peut examiner la télémétrie de Datadog, Grafana, New Relic et Splunk lorsqu'il enquête sur un problème via le serveur MCP distant de chaque fournisseur.

Pour en savoir plus sur les intégrations unidirectionnelles, consultez les rubriques suivantes :

- the section called “Connecter DataDog”
- the section called “Connecter Grafana”
- the section called “Connecter New Relic”
- the section called “Connecter Splunk”

Bring-your-own sources de télémétrie

Pour toute autre source de télémétrie, y compris les métriques Prometheus, vous pouvez tirer parti du support de l' AWS DevOps Agent pour l'intégration du webhook et du serveur MCP.

Pour en savoir plus sur bring-your-own les intégrations, consultez ce qui suit

- the section called “Invocation de DevOps l'agent via Webhook”
- the section called “Connexion de serveurs MCP”

Connecter Dynatrace

Intégration bidirectionnelle intégrée

Actuellement, AWS DevOps Agent prend en charge les utilisateurs de Dynatrace grâce à une intégration bidirectionnelle intégrée permettant ce qui suit :

- Cartographie des ressources topologiques - AWS DevOps L'agent augmentera la topologie de votre espace DevOps agent avec les entités et les relations disponibles dans votre environnement Dynatrace.
- Déclenchement automatique des enquêtes - Les flux de travail Dynatrace peuvent être configurés pour déclencher des enquêtes de résolution d'incidents liés à des problèmes Dynatrace.
- Introspection de la télémétrie : l' AWS DevOps agent peut introspecter la télémétrie Dynatrace lorsqu'il étudie un problème via le serveur Dynatrace MCP hébergé par l'agent. AWS DevOps
- Mises à jour de statut - AWS DevOps L'agent publiera les principaux résultats de l'enquête, les analyses des causes profondes et les plans d'atténuation générés sur l'interface utilisateur de Dynatrace.

Intégration

Processus d'intégration

L'intégration de votre système d'observabilité Dynatrace comprend trois étapes :

1. Connect - Établissez une connexion à Dynatrace en configurant les identifiants d'accès au compte, avec tous les environnements dont vous pourriez avoir besoin
2. Activer - Activez Dynatrace dans des espaces d'agent spécifiques avec des environnements Dynatrace spécifiques

3. Configurez votre environnement Dynatrace : téléchargez les flux de travail et le tableau de bord, puis importez-les dans Dynatrace, en prenant note des détails du webhook pour déclencher des enquêtes dans les espaces réservés aux agents

Étape 1 : Connect

Établissez une connexion à votre environnement Dynatrace

Configuration

1. Accédez à la page Capability Providers (accessible depuis la navigation latérale)
2. Trouvez Dynatrace dans la section Fournisseurs disponibles sous Télémétrie et cliquez sur S'inscrire
3. Créez un OAuth client dans Dynatrace, avec les autorisations détaillées.
   a. Voir la documentation de Dynatrace
   b. Lorsque vous êtes prêt, appuyez sur Suivant
   c. Vous pouvez connecter plusieurs environnements Dynatrace et les étendre ultérieurement à des environnements spécifiques pour chaque DevOps agent Space dont vous disposez.
4. Entrez vos informations Dynatrace depuis la configuration du OAuth client :
   - Nom du client
   - Identifiant du client
   - Secret du client
   - URN du compte
5. Cliquez sur Suivant
6. Réviser et ajouter

Étape 2 : activer

Activez Dynatrace dans un espace d'agent spécifique et configurez le scoping approprié

Configuration

1. Sur la page des espaces d'agent, sélectionnez un espace d'agent et appuyez sur Afficher les détails
2. Sélectionnez l'onglet Fonctionnalités

3. Localisez la section Télémétrie, appuyez sur Ajouter
4. Vous remarquerez que Dynatrace a le statut « Enregistré ». Cliquez sur Ajouter pour l'ajouter à votre espace agent
5. ID d'environnement Dynatrace - Indiquez l'identifiant d'environnement Dynatrace que vous souhaitez associer à cet espace d'agents. DevOps
6. Entrez une ou plusieurs entités IDs Dynatrace. Ces DevOps agents aident à découvrir vos ressources les plus importantes, par exemple des services ou des applications. Si vous n'êtes pas sûr, vous pouvez appuyer sur Supprimer.
7. Vérifiez et appuyez sur Enregistrer
8. Copiez l'URL du webhook et le secret du webhook. Consultez la documentation de Dynatrace pour ajouter ces informations d'identification à Dynatrace.

Étape 3 : Configuration de votre environnement Dynatrace

Pour terminer la configuration de Dynatrace, vous devrez effectuer certaines étapes de configuration dans votre environnement Dynatrace. Suivez les instructions de la documentation de Dynatrace.

Schémas d'événements pris en charge

AWS DevOps L'agent prend en charge deux types d'événements de Dynatrace à l'aide de webhooks. Les schémas d'événements pris en charge sont documentés ci-dessous :

Événement d'incident

Les incidents sont utilisés pour déclencher une enquête. Le schéma de l'événement est le suivant :

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

Événement d'atténuation

Les événements d'atténuation sont utilisés pour déclencher la génération d'un rapport d'atténuation pour l'enquête sur les prochaines étapes. Le schéma de l'événement est le suivant :

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

Enlèvement

La source de télémétrie est connectée à deux niveaux, au niveau de l'espace agent et au niveau du compte. Pour le supprimer complètement, vous devez d'abord le supprimer de tous les espaces d'agent où il est utilisé, puis il peut être désenregistré.

Étape 1 : Supprimer de l'espace agent

1. Sur la page des espaces d'agent, sélectionnez un espace d'agent et appuyez sur Afficher les détails
2. Sélectionnez l'onglet Fonctionnalités
3. Faites défiler la page jusqu'à la section Télémétrie
4. Sélectionnez Dynatrace
5. Appuyez sur Supprimer

Étape 2 : Désenregistrer du compte

1. Accédez à la page Capability Providers (accessible depuis la navigation latérale)
2. Accédez à la section Actuellement enregistré.
3. Vérifiez que le nombre d'espaces d'agent est nul (sinon, répétez l'étape 1 ci-dessus dans vos autres espaces d'agent)
4. Appuyez sur Désenregistrer à côté de Dynatrace

Connecter DataDog

Intégration unidirectionnelle intégrée

À l'heure actuelle, AWS DevOps l'Agent prend en charge les utilisateurs de Datadog grâce à une intégration unidirectionnelle intégrée, qui permet ce qui suit :

- Déclenchement automatique des enquêtes : les événements Datadog peuvent être configurés pour déclencher AWS DevOps des enquêtes de résolution d'incidents par le biais des webhooks des AWS DevOps agents.
- Introspection de la télémétrie : l' AWS DevOps agent peut examiner la télémétrie Datadog lorsqu'il étudie un problème via le serveur MCP distant de chaque fournisseur.

Intégration

Étape 1 : Connect

Établissez une connexion à votre point de terminaison MCP distant Datadog à l'aide des identifiants d'accès au compte

Configuration

1. Accédez à la page Capability Providers (accessible depuis la navigation latérale)
2. Trouvez Datadog dans la section Fournisseurs disponibles sous Télémétrie, puis cliquez sur Enregistrer
3. Entrez les détails de votre serveur Datadog MCP :
   - Nom du serveur - Identifiant unique (par exemple, my-datadog-server)
   - URL du point de terminaison : le point de terminaison de votre serveur MCP Datadog. L'URL du point de terminaison varie en fonction de votre site Datadog. Consultez le tableau des points de terminaison du site Datadog ci-dessous.
   - Description : description facultative du serveur
4. Cliquez sur Suivant
5. Vérification et soumission

Points de terminaison du site Datadog

L'URL du point de terminaison MCP varie en fonction de votre site Datadog. Pour identifier votre site, vérifiez l'URL dans votre navigateur lorsque vous êtes connecté à Datadog, ou consultez Accéder au site Datadog.

Site Datadog	Domaine du site	URL du point de terminaison MCP
US1 (par défaut)	`datadoghq.com`	`https://mcp.datadoghq.com/api/unstable/mcp-server/mcp`
US3	`us3.datadoghq.com`	`https://mcp.us3.datadoghq.com/api/unstable/mcp-server/mcp`
US5	`us5.datadoghq.com`	`https://mcp.us5.datadoghq.com/api/unstable/mcp-server/mcp`
EU1	`datadoghq.eu`	`https://mcp.datadoghq.eu/api/unstable/mcp-server/mcp`
AP1	`ap1.datadoghq.com`	`https://mcp.ap1.datadoghq.com/api/unstable/mcp-server/mcp`
AP2	`ap2.datadoghq.com`	`https://mcp.ap2.datadoghq.com/api/unstable/mcp-server/mcp`

Autorisation

OAuth Autorisation complète par :

- Autorisation en tant qu'utilisateur sur la page Datadog OAuth
- Si vous n'êtes pas connecté, cliquez sur Autoriser, connectez-vous, puis sur Autoriser

Une fois configuré, Datadog est disponible dans tous les espaces Agent.

Étape 2 : activer

Activez DataDog dans un espace d'agent spécifique et configurez le scoping approprié

Configuration

1. Sur la page des espaces d'agent, sélectionnez un espace d'agent et appuyez sur Afficher les détails (si vous n'avez pas encore créé d'espace d'agent, voirthe section called "Création d'un espace d'agents")
2. Sélectionnez l'onglet Fonctionnalités
3. Faites défiler la page jusqu'à la section Télémétrie
4. Appuyez sur Ajouter
5. Sélectionnez Datadog
6. Suivant
7. Vérifiez et appuyez sur Enregistrer
8. Copiez l'URL du webhook et la clé API

Étape 3 : Configuration des webhooks

À l'aide de l'URL du webhook et de la clé d'API, vous pouvez configurer Datadog pour qu'il envoie des événements afin de déclencher une enquête, par exemple à partir d'une alarme.

Pour garantir que les événements envoyés peuvent être utilisés par l' DevOps agent, assurez-vous que les données transmises au webhook correspondent au schéma de données spécifié ci-dessous. Les événements qui ne correspondent pas à ce schéma peuvent être ignorés par DevOps l'agent.

Définissez la méthode et les en-têtes

```
    method: "POST",
```

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

Envoyez le corps sous forme de chaîne JSON.

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

Envoyez des webhooks avec Datadog https://docs.datadoghq.com/integrations/webhooks/ (attention, ne sélectionnez aucune autorisation et utilisez plutôt l'option d'en-tête personnalisé).

En savoir plus : Serveur MCP distant Datadog

Enlèvement

La source de télémétrie est connectée à deux niveaux, au niveau de l'espace agent et au niveau du compte. Pour le supprimer complètement, vous devez d'abord le supprimer de tous les espaces d'agent où il est utilisé, puis il peut être désenregistré.

Étape 1 : Supprimer de l'espace agent

1. Sur la page des espaces d'agent, sélectionnez un espace d'agent et appuyez sur Afficher les détails
2. Sélectionnez l'onglet Fonctionnalités
3. Faites défiler la page jusqu'à la section Télémétrie
4. Sélectionnez Datadog
5. Appuyez sur Supprimer

Étape 2 : Désenregistrer du compte

1. Accédez à la page Capability Providers (accessible depuis la navigation latérale)
2. Accédez à la section Actuellement enregistré.
3. Vérifiez que le nombre d'espaces d'agent est nul (sinon, répétez l'étape 1 ci-dessus dans vos autres espaces d'agent)
4. Appuyez sur Désenregistrer à côté de Datadog

Connecter Grafana

L'intégration de Grafana permet à l' AWS DevOps agent d'interroger les métriques, les tableaux de bord et les données d'alerte à partir de votre instance Grafana lors des enquêtes sur les incidents. Cette intégration suit un processus en deux étapes : enregistrement de Grafana au niveau du compte, suivi de sa connexion à des espaces d'agent individuels.

Pour améliorer la sécurité, l'intégration Grafana active uniquement les outils en lecture seule. Les outils d'écriture sont désactivés et ne peuvent pas être activés. Cela signifie que l'agent peut interroger et lire les données de votre instance Grafana, mais ne peut pas créer, modifier ou supprimer des ressources Grafana telles que des tableaux de bord, des alertes ou des annotations. Pour plus d'informations, consultez la section Sécurité dans AWS DevOps l'agent.

Exigences relatives à Grafana

Avant de connecter Grafana, assurez-vous d'avoir :

- Grafana version 9.0 ou ultérieure. Certaines fonctionnalités, en particulier les opérations liées aux sources de données, peuvent ne pas fonctionner correctement avec les versions antérieures en raison de l'absence de points de terminaison d'API.
- Une instance de Grafana accessible via HTTPS. Les points de terminaison des réseaux publics et privés sont pris en charge. Grâce à la connectivité réseau privée, votre instance Grafana peut être hébergée dans un VPC sans accès public à Internet. Pour en savoir plus, consultez the section called “Connexion à des outils hébergés en privé”.
- Un compte de service Grafana avec un jeton d'accès doté des autorisations de lecture appropriées

Enregistrer Grafana (au niveau du compte)

Grafana est enregistré au niveau du AWS compte et partagé entre tous les espaces d'agent de ce compte.

Étape 1 : Configuration de Grafana

1. Connectez-vous à la console AWS de gestion
2. Accédez à la console de AWS DevOps l'agent
3. Accédez à la page Capability Providers (accessible depuis la navigation latérale)
4. Trouvez Grafana dans la section Fournisseurs disponibles sous Télémétrie et cliquez sur Enregistrer
5. Sur la page Configurer Grafana, entrez les informations suivantes :
   - Nom du service (obligatoire) — Entrez un nom descriptif pour votre serveur Grafana en utilisant uniquement des caractères alphanumériques, des traits d'union et des traits de soulignement. Par exemple, `my-grafana-server`.
   - URL Grafana (obligatoire) — Entrez l'URL HTTPS complète de votre instance Grafana. Par exemple, `https://myinstance.grafana.net`.
   - Jeton d'accès au compte de service (obligatoire) — Entrez un jeton d'accès au compte de service Grafana. Les jetons commencent généralement par`glsa_`. Pour créer un jeton de compte de service, accédez à votre instance Grafana, accédez à Administration > Comptes de service, créez un compte de service avec le rôle Viewer et générez un jeton.
   - Description (facultatif) — Ajoutez une description pour aider à identifier l'objectif du serveur. Par exemple, `Production Grafana server for monitoring`.
6. (Facultatif) Ajoutez des AWS balises à l'enregistrement à des fins d'organisation.
7. Cliquez sur Suivant

Étape 2 : Vérifiez et soumettez votre inscription à Grafana

1. Passez en revue tous les détails de configuration de Grafana
2. Cliquez sur Soumettre pour terminer l'enregistrement
3. Une fois l'inscription réussie, Grafana apparaît dans la section Actuellement enregistré de la page Capability Providers

Ajouter Grafana à un espace d'agents

Après avoir enregistré Grafana au niveau du compte, vous pouvez le connecter à des espaces d'agent individuels :

1. Dans la console AWS DevOps Agent, sélectionnez votre espace agent
2. Accédez à l'onglet Fonctionnalités
3. Dans la section Télémétrie, cliquez sur Ajouter
4. Sélectionnez Grafana dans la liste des fournisseurs disponibles
5. Cliquez sur Enregistrer

Configuration des webhooks d'alerte Grafana

Vous pouvez configurer Grafana pour déclencher automatiquement les enquêtes des AWS DevOps agents lorsque des alertes sont déclenchées en envoyant des webhooks via les points de contact Grafana. Pour plus de détails sur les méthodes d'authentification par webhook et la gestion des informations d'identification, consultez. the section called “Invocation de DevOps l'agent via Webhook”

Étape 1 : créer un modèle de notification personnalisé

Dans votre instance Grafana, accédez à Alertes > Points de contact > Modèles de notification et créez un nouveau modèle avec le contenu suivant :

```
{{ 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 }}
```

Ce modèle formate les alertes Grafana selon la structure de charge utile du webhook attendue par l'Agent. AWS DevOps Il mappe les étiquettes, les annotations et le statut des alertes dans les champs appropriés, et inclut toutes les étiquettes d'alerte sous forme de métadonnées.

> Remarque : Ce modèle traite uniquement la première alerte d'un groupe. Grafana regroupe plusieurs alertes de déclenchement en une seule notification par défaut. Pour vous assurer que chaque alerte est envoyée individuellement, configurez vos politiques de notification pour qu'elles soient groupées par `alertname`. De plus, ce modèle n'échappe pas aux caractères JSON spéciaux dans les valeurs d'étiquette ou les annotations. Assurez-vous que les étiquettes d'alerte et les `summary` annotations ne contiennent pas de caractères tels que des guillemets ou des nouvelles lignes, ce qui produirait un JSON non valide.

Étape 2 : créer un point de contact Webhook

1. Dans Grafana, accédez à Alertes > Points de contact et cliquez sur Ajouter un point de contact
2. Sélectionnez Webhook comme type d'intégration
3. Définissez l'URL du point de terminaison de votre AWS DevOps agent Webhook
4. Sous Paramètres optionnels du webhook, configurez les en-têtes d'authentification en fonction de votre type de webhook. Voir Méthodes d'authentification Webhook pour plus de détails.
5. Définissez le champ Message pour utiliser votre modèle personnalisé : `{{ template "devops-agent-payload" . }}`
6. Cliquez sur Enregistrer le point de contact

Étape 3 : Affecter le point de contact à une politique de notification

1. Accédez à Alertes > Politiques de notification
2. Modifier une politique existante ou en créer une nouvelle
3. Définissez le point de contact sur le point de contact Webhook que vous avez créé

4. Cliquez sur Enregistrer la politique

Lorsqu'une alerte correspondante se déclenche, Grafana envoie la charge utile formatée à l' AWS DevOps agent, qui lance automatiquement une enquête.

Limitations

- ClickHouse outils de source de ClickHouse données : les outils de source de données ne sont actuellement pas pris en charge.
- Prévention proactive des incidents : the section called "Prévention proactive des incidents" n'utilise pas actuellement les outils Grafana. Support est prévu pour une future version.

Considérations relatives à Amazon Managed Grafana

Si vous utilisez Amazon Managed Grafana (AMG), tenez compte des limites suivantes :

- Les points de contact Webhook ne sont pas pris en charge — AMG ne prend actuellement pas en charge les points de contact Webhook dans sa configuration d'alerte. Vous ne pouvez pas utiliser AMG pour envoyer des webhooks d'alerte directement à l' AWS DevOps Agent. Pour plus de détails, consultez la section Alerter les points de contact dans Amazon Managed Grafana.
- Expiration des jetons de compte de service — Les jetons de compte de service AMG ont une expiration maximale de 30 jours. Vous devrez alterner les jetons et mettre à jour votre inscription à Grafana dans AWS DevOps Agent avant leur expiration. Voir Gestion des connexions Grafana pour savoir comment mettre à jour les informations d'identification. Pour en savoir plus sur les limites des jetons AMG, consultez la section Comptes de service dans Amazon Managed Grafana.

Gestion des connexions Grafana

- Mise à jour des informations d'identification — Si le jeton de votre compte de service expire ou doit être mis à jour, désenregistrez Grafana de la page Capability Providers et enregistrez-vous à nouveau avec le nouveau jeton.
- Affichage des instances connectées : dans la console de l' AWS DevOps agent, sélectionnez votre espace agent et accédez à l'onglet Fonctionnalités pour afficher les sources de télémétrie connectées.
- Supprimer Grafana — Pour déconnecter Grafana d'un espace agent, sélectionnez-le dans la section Télémétrie et cliquez sur Supprimer. Pour supprimer complètement l'enregistrement,

supprimez-le d'abord de tous les agents Spaces, puis désenregistrez-vous de la page Capability Providers.

Connecter New Relic

Intégration unidirectionnelle intégrée

Actuellement, AWS DevOps Agent prend en charge les utilisateurs de New Relic grâce à une intégration unidirectionnelle intégrée, permettant ce qui suit :

- Déclenchement automatique des enquêtes - Les événements New Relic peuvent être configurés pour déclencher des enquêtes de résolution d'incidents par l' AWS DevOps agent via des webhooks d' AWS DevOps agent.
- Introspection de la télémétrie : l' AWS DevOps agent peut introspecter la télémétrie New Relic lorsqu'il étudie un problème via le serveur MCP distant de chaque fournisseur.

Intégration

Étape 1 : Connect

Établissez une connexion à votre point de terminaison MCP distant New Relic à l'aide des informations d'accès au compte

Veuillez utiliser un utilisateur de plateforme complète (et non Basic/Core) dans New Relic pour activer les outils New Relic MCP.

Configuration

1. Accédez à la page Capability Providers (accessible depuis la navigation latérale)
2. Trouvez New Relic dans la section Fournisseurs disponibles sous Télémétrie et cliquez sur Enregistrer
3. Suivez les instructions pour obtenir votre clé d'API New Relic
4. Entrez les détails de la clé d'API de votre serveur New Relic MCP :
   - ID de compte : Entrez votre identifiant de compte New Relic obtenu ci-dessus
   - Clé d'API : entrez la clé d'API obtenue ci-dessus
   - Sélectionnez la région des États-Unis ou de l'UE en fonction de l'endroit où se trouve votre compte New Relic.

5. Cliquez sur Ajouter

Étape 2 : activer

Activez New Relic dans un espace d'agent spécifique et configurez le scope approprié

Configuration

1. Sur la page des espaces d'agent, sélectionnez un espace d'agent et appuyez sur Afficher les détails (si vous n'avez pas encore créé d'espace d'agent, voirthe section called "Création d'un espace d'agents")
2. Sélectionnez l'onglet Fonctionnalités
3. Faites défiler la page jusqu'à la section Télémétrie
4. Appuyez sur Ajouter
5. Sélectionnez New Relic
6. Suivant
7. Vérifiez et appuyez sur Enregistrer
8. Copiez l'URL du webhook et la clé API

Étape 3 : Configuration des webhooks

À l'aide de l'URL du webhook et de la clé API, vous pouvez configurer New Relic pour envoyer des événements afin de déclencher une enquête, par exemple à partir d'une alarme. Pour plus de détails sur la configuration des webhooks, consultez la section Suivi des modifications des webhooks.

Pour garantir que les événements envoyés peuvent être utilisés par l' DevOps agent, assurez-vous que les données transmises au webhook correspondent au schéma de données spécifié ci-dessous. Les événements qui ne correspondent pas à ce schéma peuvent être ignorés par DevOps l'agent.

Définissez la méthode et les en-têtes

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

Envoyez le corps sous forme de chaîne JSON.

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

Envoyez des webhooks avec les notifications de webhook de New Relic. https://newrelic.com/instant-observability/ Vous pouvez soit sélectionner le jeton Bearer pour le type d'autorisation, soit sélectionner aucune autorisation et l'ajouter `Authorization: Bearer <Token>` en tant qu'en-tête personnalisé à la place.

Pour en savoir plus : https://docs.newrelic.com/docs/agentic-/ai/mcp/overview

Enlèvement

La source de télémétrie est connectée à deux niveaux, au niveau de l'espace agent et au niveau du compte. Pour le supprimer complètement, vous devez d'abord le supprimer de tous les espaces d'agent où il est utilisé, puis il peut être désenregistré.

Étape 1 : Supprimer de l'espace agent

1. Sur la page des espaces d'agent, sélectionnez un espace d'agent et appuyez sur Afficher les détails
2. Sélectionnez l'onglet Fonctionnalités
3. Faites défiler la page jusqu'à la section Télémétrie
4. Sélectionnez New Relic
5. Appuyez sur Supprimer

Étape 2 : Désenregistrer du compte

1. Accédez à la page Capability Providers (accessible depuis la navigation latérale)
2. Accédez à la section Actuellement enregistré.

3. Vérifiez que le nombre d'espaces d'agent est nul (sinon, répétez l'étape 1 ci-dessus dans vos autres espaces d'agent)
4. Appuyez sur Désenregistrer à côté de New Relic

Connecter Splunk

Intégration unidirectionnelle intégrée

Actuellement, AWS DevOps Agent prend en charge les utilisateurs de Splunk grâce à une intégration unidirectionnelle intégrée, permettant ce qui suit :

- Déclenchement automatique des enquêtes : les événements Splunk peuvent être configurés pour déclencher AWS DevOps des enquêtes de résolution d'incidents par l'intermédiaire des webhooks des AWS DevOps agents.
- Introspection de la télémétrie : l' AWS DevOps agent peut effectuer une introspection de la télémétrie Splunk lorsqu'il étudie un problème via le serveur MCP distant de chaque fournisseur.

Conditions préalables

Obtenir un jeton d'API Splunk

Vous aurez besoin d'une URL MCP et d'un jeton pour vous connecter à Splunk.

Étapes d'administration de Splunk

Votre administrateur Splunk doit effectuer les étapes suivantes :

- activer l'accès à l'API REST
- activer l'authentification par jeton lors du déploiement.
- créez un nouveau rôle « mcp_user », le nouveau rôle n'a pas besoin de fonctionnalités.
- attribuez le rôle « mcp_user » à tous les utilisateurs du déploiement autorisés à utiliser le serveur MCP.
- créez le jeton pour les utilisateurs autorisés dont l'audience est « mcp » et définissez le délai d'expiration approprié, si l'utilisateur n'est pas autorisé à créer lui-même des jetons.

Étapes pour les utilisateurs de Splunk

Un utilisateur de Splunk doit effectuer les étapes suivantes :

- Obtenez un jeton approprié auprès de l'administrateur Splunk ou créez-en un lui-même, s'il en a l'autorisation. L'audience du jeton doit être « mcp ».

Intégration

Étape 1 : Connect

Établissez une connexion à votre point de terminaison MCP distant Splunk à l'aide des informations d'accès au compte

Configuration

1. Accédez à la page Capability Providers (accessible depuis la navigation latérale)
2. Trouvez Splunk dans la section Fournisseurs disponibles sous Télémétrie et cliquez sur Enregistrer
3. Entrez les détails de votre serveur Splunk MCP :
   - Nom du serveur - Identifiant unique (par exemple, my-splunk-server)
   - URL du point de terminaison : point de terminaison de votre serveur Splunk MCP :

```
https://<YOUR_SPLUNK_DEPLOYMENT_NAME>.api.scs.splunk.com/
<YOUR_SPLUNK_DEPLOYMENT_NAME>/mcp/v1/
```

- Description : description facultative du serveur
- Nom du jeton : nom du jeton porteur pour l'authentification : `my-splunk-token`
- Valeur du jeton La valeur du jeton porteur pour l'authentification

Étape 2 : activer

Activez Splunk dans un espace d'agent spécifique et configurez le scoping approprié

Configuration

1. Sur la page des espaces d'agent, sélectionnez un espace d'agent et appuyez sur Afficher les détails (si vous n'avez pas encore créé d'espace d'agent, voirthe section called “Création d'un espace d'agents”)
2. Sélectionnez l'onglet Fonctionnalités

3. Faites défiler la page jusqu'à la section Télémétrie
4. Appuyez sur Ajouter
5. Sélectionnez Splunk
6. Suivant
7. Vérifiez et appuyez sur Enregistrer
8. Copiez l'URL du webhook et la clé API

Étape 3 : Configuration des webhooks

À l'aide de l'URL et de la clé API du webhook, vous pouvez configurer Splunk pour envoyer des événements afin de déclencher une enquête, par exemple à partir d'une alarme.

Pour garantir que les événements envoyés peuvent être utilisés par l' DevOps agent, assurez-vous que les données transmises au webhook correspondent au schéma de données spécifié ci-dessous. Les événements qui ne correspondent pas à ce schéma peuvent être ignorés par DevOps l'agent.

Définissez la méthode et les en-têtes

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

Envoyez le corps sous forme de chaîne JSON.

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

```
}
```

Envoyez des webhooks avec Splunk https://help.splunk.com/en/splunk enterprise/alert-and-respond/alerting-manual/9.4/configure-alert-actions/use - a-webhook-alert-action (notez que vous ne sélectionnez aucune autorisation et utilisez plutôt l'option d'en-tête personnalisé)

En savoir plus :

- Documentation du serveur MCP de Splunk : https://help.splunk.com/en/splunk-cloud-platform/- platform/ mcp-server-for-splunk -splunk-platform about-mcp-server-for
- Exigences et limites d'accès pour l'API REST de Splunk Cloud Platform : https://docs.splunk.com/Documentation/SplunkCloud/latest/RESTTUT/RESTandCloud
- Gérez les jetons d'authentification dans 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
- Créez et gérez des rôles avec Splunk Web : https://docs.splunk.com/Documentation/SplunkCloud/latest/Security/Addandeditroles

Enlèvement

La source de télémétrie est connectée à deux niveaux, au niveau de l'espace agent et au niveau du compte. Pour le supprimer complètement, vous devez d'abord le supprimer de tous les espaces d'agent où il est utilisé, puis il peut être désenregistré.

Étape 1 : Supprimer de l'espace agent

1. Sur la page des espaces d'agent, sélectionnez un espace d'agent et appuyez sur Afficher les détails
2. Sélectionnez l'onglet Fonctionnalités
3. Faites défiler la page jusqu'à la section Télémétrie
4. Sélectionnez Splunk
5. Appuyez sur Supprimer

Étape 2 : Désenregistrer du compte

1. Accédez à la page Capability Providers (accessible depuis la navigation latérale)
2. Accédez à la section Actuellement enregistré.

3. Vérifiez que le nombre d'espaces d'agent est nul (sinon, répétez l'étape 1 ci-dessus dans vos autres espaces d'agent)
4. Appuyez sur Désenregistrer à côté de Splunk

Connexion à la billetterie et au chat

AWS DevOps L'agent est conçu pour agir en tant que membre de votre équipe en participant aux canaux de communication existants de votre équipe. Vous pouvez connecter DevOps l'agent à vos systèmes de billetterie et d'alarme, par exemple, pour ServiceNow lancer automatiquement des enquêtes à partir de tickets d'incident, en accélérant ainsi la réponse aux incidents dans vos flux de travail existants afin de réduire le temps moyen de restauration (MTTR). PagerDuty Vous pouvez également connecter votre DevOps agent aux systèmes de collaboration de votre équipe tels que Slack pour recevoir des résumés d'activités de votre DevOps agent sur un canal de discussion.

Pour en savoir plus sur la connexion des intégrations de billetterie et de chat, consultez ce qui suit :

- the section called “Connecter PagerDuty”
- the section called “Connecter ServiceNow”
- the section called “Connecter Slack”

Connecter PagerDuty

PagerDuty l'intégration permet à l' AWS DevOps agent d'accéder aux données sur les incidents, aux horaires d'appel et aux informations de service à partir de votre PagerDuty compte et de les mettre à jour lors des enquêtes sur les incidents et des réponses automatisées. Cette intégration utilise la OAuth version 2.0 pour une authentification sécurisée.

> **Important**
>
> AWS DevOps L'agent ne prend en charge que la version PagerDuty OAuth 2.0 la plus récente (Scoped OAuth). L'ancienne PagerDuty OAuth version avec URI de redirection n'est pas prise en charge.

PagerDuty exigences

Avant de vous connecter PagerDuty, assurez-vous d'avoir :

- Un PagerDuty compte avec votre identifiant OAuth client et votre secret client
- Le sous-domaine de votre PagerDuty compte (par exemple, si votre PagerDuty URL est`https://your-company.pagerduty.com`, le sous-domaine est) `your-company`

S'inscrire PagerDuty

PagerDuty est enregistré au niveau du AWS compte et partagé entre tous les espaces d'agent de ce compte.

Étape 1 : configurer l'accès dans PagerDuty

1. Connectez-vous à la console AWS de gestion
2. Accédez à la console de AWS DevOps l'agent
3. Accédez à la page Capability Providers (accessible depuis la navigation latérale)
4. Recherchez PagerDutydans la section Fournisseurs disponibles sous Communication et cliquez sur Enregistrer
5. Suivez les instructions de configuration sur la PagerDuty page Configurer l'accès dans :

Vérifiez votre région de service et votre sous-domaine :

- Étendue du compte — Sélectionnez votre PagerDuty région (États-Unis ou UE) et entrez votre PagerDuty sous-domaine. Par exemple, si votre PagerDuty URL est`https://your-company.pagerduty.com`, entrez`your-company`.

Créez une nouvelle application dans PagerDuty :

- Dans un onglet de navigateur distinct, connectez-vous PagerDuty et accédez à Intégrations > Enregistrement des applications
- Créez une nouvelle application à l'aide de OAuth 2.0 Scoped OAuth
- Sous Autorisations, accordez les étendues minimales requises suivantes : `incidents.read`, `incidents.write`, et `services.read`
- Activez l'intégration des événements pour permettre une communication bidirectionnelle entre AWS DevOps l'agent et PagerDuty

Configurer les OAuth informations d'identification :

- Étendue des autorisations — Les étendues minimales requises sont les suivantes :`incidents.read`,, `incidents.write` `services.read`
- Nom du client — Entrez un nom descriptif pour votre OAuth client
- ID client — Entrez l'identifiant OAuth client indiqué lors de PagerDuty l'enregistrement de votre application
- Secret client — Entrez le secret OAuth client indiqué lors de PagerDuty l'enregistrement de votre application

Étape 2 : Vérifiez et soumettez PagerDuty l'enregistrement

1. Passez en revue tous les détails PagerDuty de configuration
2. Cliquez sur Soumettre pour terminer l'enregistrement
3. Une fois l'enregistrement réussi, PagerDuty apparaît dans la section Actuellement enregistré de la page Fournisseurs de capacités

Ajouter PagerDuty à un espace d'agent

Une fois enregistré PagerDuty au niveau du compte, vous pouvez le connecter à des espaces d'agent individuels :

1. Dans la console AWS DevOps Agent, sélectionnez votre espace agent
2. Accédez à l'onglet Fonctionnalités
3. Dans la section Communications, cliquez sur Ajouter
4. Sélectionnez PagerDutydans la liste des fournisseurs disponibles
5. Cliquez sur Enregistrer

Gestion des PagerDuty connexions

- Mise à jour des informations d'identification — Si vos OAuth informations d'identification doivent être mises à jour, désenregistrez-vous PagerDuty de la page des fournisseurs de capacités et enregistrez-vous à nouveau avec les nouvelles informations d'identification.
- Affichage des connexions : dans la console de l' AWS DevOps agent, sélectionnez votre espace agent et accédez à l'onglet Fonctionnalités pour afficher les intégrations de communication connectées.

- Suppression PagerDuty : pour vous PagerDuty déconnecter d'un espace agent, sélectionnez-le dans la section Communications et cliquez sur Supprimer. Pour supprimer complètement l'enregistrement, supprimez-le d'abord de tous les agents Spaces, puis désenregistrez-vous de la page Capability Providers.

Support du webhook

AWS DevOps L'agent prend uniquement en charge les PagerDuty webhooks V3. Les versions antérieures du webhook ne sont pas prises en charge.

Pour plus d'informations sur les abonnements aux webhooks PagerDuty V3, consultez la section Présentation des webhooks dans la PagerDuty documentation destinée aux développeurs.

Connecter ServiceNow

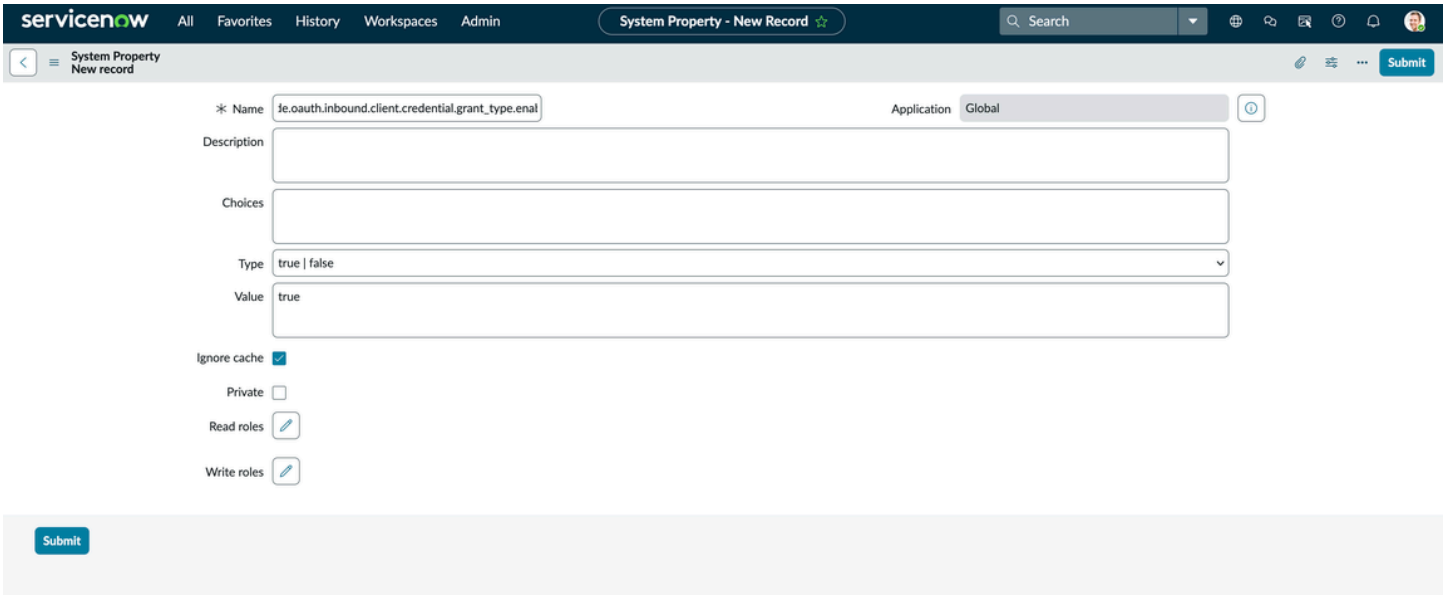
Ce didacticiel explique comment connecter une ServiceNow instance à l' AWS DevOps Agent pour lui permettre de lancer automatiquement des enquêtes de réponse aux incidents lorsqu'un ticket est créé et de publier ses principales conclusions dans le ticket d'origine. Il contient également des exemples expliquant comment configurer votre ServiceNow instance pour envoyer uniquement des tickets spécifiques à un espace DevOps agent et comment orchestrer le routage des tickets sur plusieurs espaces DevOps agent.

Configuration initiale

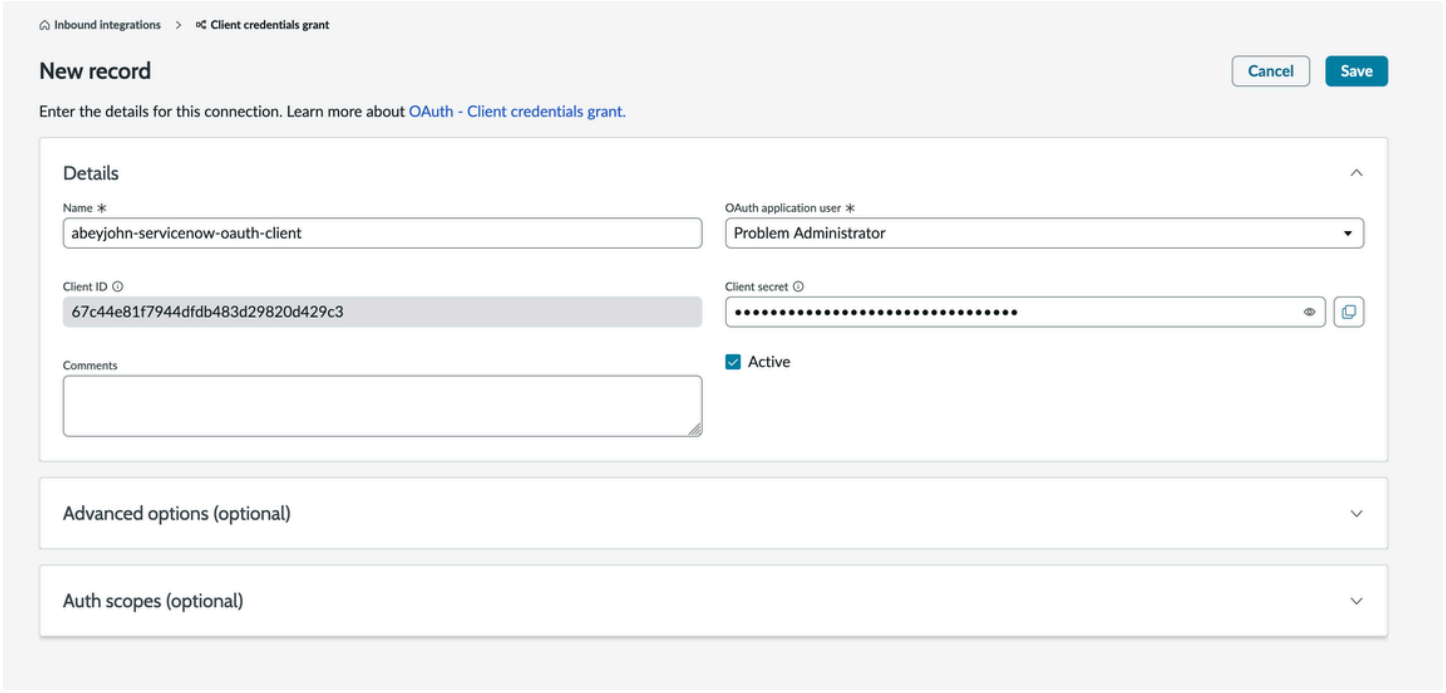
La première étape consiste à créer ServiceNow un client d' OAuth application qui AWS DevOps peut être utilisé pour accéder à votre ServiceNow instance.

Création d'un client ServiceNow OAuth d'application

1. Activez la propriété du système d'identification client de votre instance
   a. Recherchez `sys_properties.list` dans le champ de recherche du filtre, puis appuyez sur Entrée (l'option n'apparaîtra pas, mais appuyer sur Entrée fonctionne)
   b. Choisissez Nouveau
   c. Ajoutez le nom as `glide.oauth.inbound.client.credential.grant_type.enabled` et la valeur à true avec le type true | false



1. Accédez à Système OAuth > Registre des applications à partir du champ de recherche du filtre
2. Choisissez « Nouveau » > « Nouvelle expérience d'intégration entrante » > « Nouvelle intégration » > « OAuth - Octroi des informations d'identification du client »
3. Choisissez un nom et définissez l'utilisateur de l' OAuth application sur « Administrateur des problèmes », puis cliquez sur « Enregistrer »

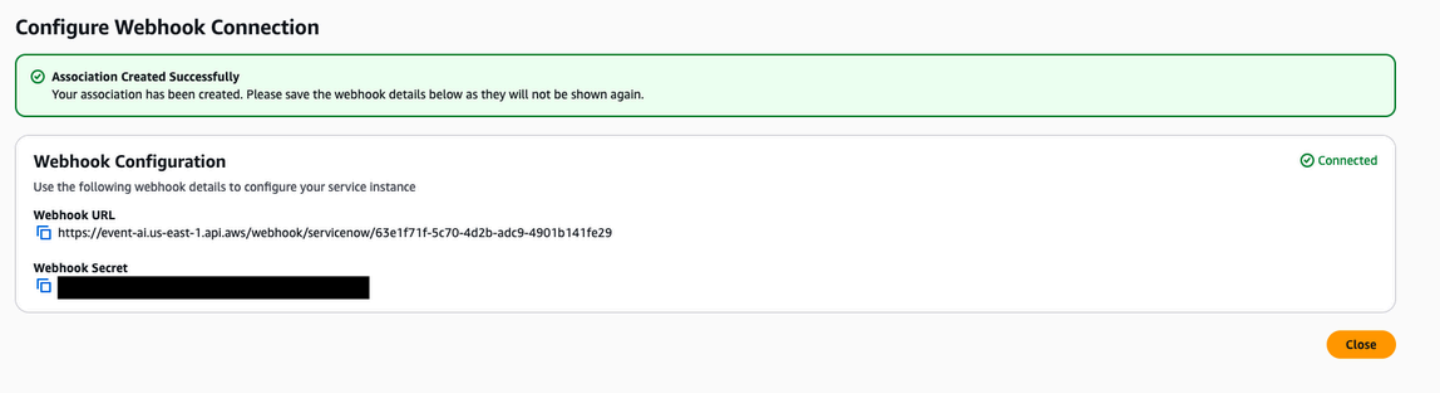


Connectez votre ServiceNow OAuth client à l' AWS DevOps agent

1. Vous pouvez démarrer ce processus à deux endroits. Tout d'abord, en accédant à la page des fournisseurs de capacités et en recherchant la ServiceNowsection Communication, puis en cliquant sur Enregistrer. Vous pouvez également sélectionner n'importe quel espace d' DevOps agent que vous avez créé et accéder à Fonctionnalités → Communications → Ajouter → ServiceNow puis cliquer sur Enregistrer.
2. Ensuite, autorisez DevOps l'Agent à accéder à votre ServiceNow instance à l'aide du client OAuth d'application que vous venez de créer.

- Suivez les étapes suivantes et enregistrez les informations obtenues sur le webhook

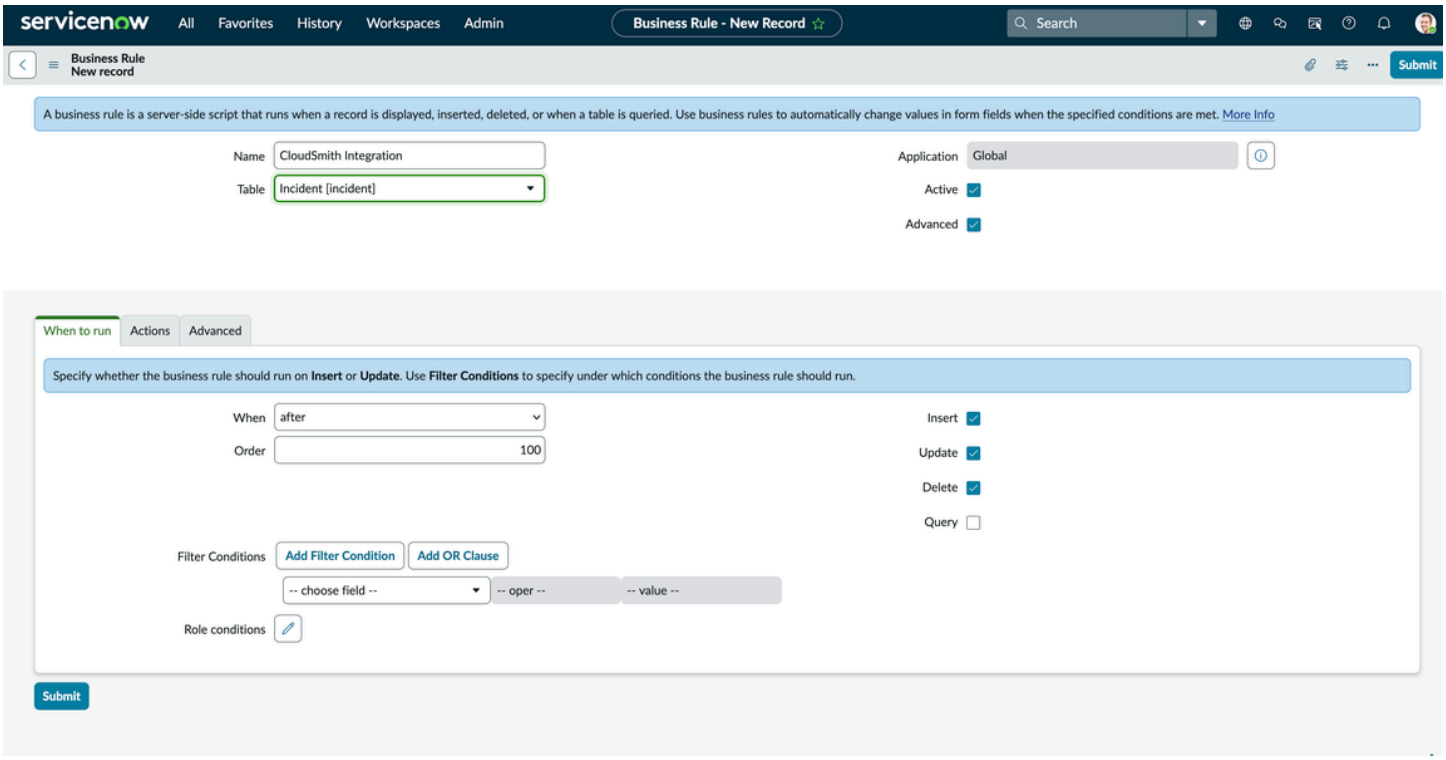
> **Important**
>
> Vous ne verrez plus ces informations



Configurez votre règle ServiceNow métier

Une fois que vous avez établi la connectivité, vous devez configurer une règle métier ServiceNow pour envoyer des tickets à vos espaces d' DevOps agents.

1. Accédez à Abonnements aux activités → Administration → Règles de gestion, puis cliquez sur Nouveau.
2. Définissez le champ « Table » sur « Incident [incident] », cochez la case « Avancé » et définissez la règle à exécuter après l'insertion, la mise à jour et la suppression.



1. Accédez à l'onglet « Avancé » et ajoutez le script de webhook suivant, en insérant le secret et l'URL de votre webhook à l'endroit indiqué, puis cliquez sur Soumettre.

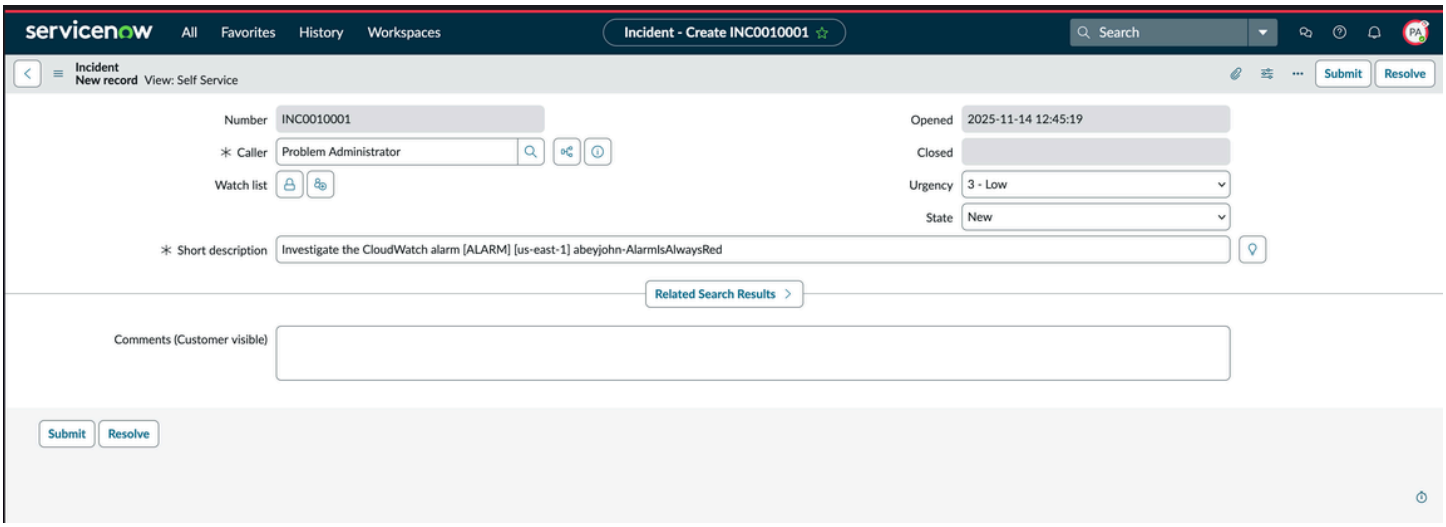
```
(function executeRule(current, previous /*null when async*/ ) {

    var WEBHOOK_CONFIG = {
        webhookSecret: GlideStringUtil.base64Encode('<<< INSERT WEBHOOK SECRET HERE
 >>>'),
        webhookUrl: '<<< INSERT WEBHOOK URL HERE >>>'
    };
```

```
    function generateHMACSignature(payloadString, secret) {
        try {
            var mac = new GlideCertificateEncryption();
            var signature = mac.generateMac(secret, "HmacSHA256", payloadString);
            return signature;
        } catch (e) {
            gs.error('HMAC generation failed: ' + e);
            return null;
        }
    }

    function callWebhook(payload, config) {
        try {
            var timestamp = new Date().toISOString();
            var payloadString = JSON.stringify(payload);
            var payloadWithTimestamp =`${timestamp}:${payloadString}`;

            var signature = generateHMACSignature(payloadWithTimestamp,
config.webhookSecret);

            if (!signature) {
                gs.error('Failed to generate signature');
                return false;
            }

            gs.info('Generated signature: ' + signature);

            var request = new sn_ws.RESTMessageV2();
            request.setEndpoint(config.webhookUrl);
            request.setHttpMethod('POST');

            request.setRequestHeader('Content-Type', 'application/json');
            request.setRequestHeader('x-amzn-event-signature', signature);
            request.setRequestHeader('x-amzn-event-timestamp', timestamp);

            request.setRequestBody(payloadString);

            var response = request.execute();
            var httpStatus = response.getStatusCode();
            var responseBody = response.getBody();

            if (httpStatus >= 200 && httpStatus < 300) {
                gs.info('Webhook sent successfully. Status: ' + httpStatus);
```

```
                return true;
            } else {
                gs.error('Webhook failed. Status: ' + httpStatus + ', Response: ' +
responseBody);
                return false;
            }

        } catch (ex) {
            gs.error('Error sending webhook: ' + ex.getMessage());
            return false;
        }
    }

    function createReference(field) {
        if (!field || field.nil()) {
            return null;
        }

        return {
            link: field.getLink(true),
            value: field.toString()
        };
    }

    function getStringValue(field) {
        if (!field || field.nil()) {
            return null;
        }
        return field.toString();
    }

    function getIntValue(field) {
        if (!field || field.nil()) {
            return null;
        }
        var val = parseInt(field.toString());
        return isNaN(val) ? null : val;
    }

    var eventType = (current.operation() == 'insert') ? "create" : "update";

    var incidentEvent = {
        eventType: eventType.toString(),
        sysId: current.sys_id.toString(),
```

```
        priority: getStringValue(current.priority),
        impact: getStringValue(current.impact),
        active: getStringValue(current.active),
        urgency: getStringValue(current.urgency),
        description: getStringValue(current.description),
        shortDescription: getStringValue(current.short_description),
        parent: getStringValue(current.parent),
        incidentState: getStringValue(current.incident_state),
        severity: getStringValue(current.severity),
        problem: createReference(current.problem),
        additionalContext: {}
    };

    incidentEvent.additionalContext = {
        number: current.number.toString(),
        opened_at: getStringValue(current.opened_at),
        opened_by: current.opened_by.nil() ? null :
 current.opened_by.getDisplayValue(),
        assigned_to: current.assigned_to.nil() ? null :
 current.assigned_to.getDisplayValue(),
        category: getStringValue(current.category),
        subcategory: getStringValue(current.subcategory),
        knowledge: getStringValue(current.knowledge),
        made_sla: getStringValue(current.made_sla),
        major_incident: getStringValue(current.major_incident)
    };

    for (var key in incidentEvent.additionalContext) {
        if (incidentEvent.additionalContext[key] === null) {
            delete incidentEvent.additionalContext[key];
        }
    }

    gs.info(JSON.stringify(incidentEvent, null, 2)); // Pretty print for logging only

    if (WEBHOOK_CONFIG.webhookUrl && WEBHOOK_CONFIG.webhookSecret) {
        callWebhook(incidentEvent, WEBHOOK_CONFIG);
    } else {
        gs.info('Webhook not configured.');
    }

})(current, previous);
```

Si vous avez choisi d'enregistrer votre ServiceNow connexion depuis la page des fournisseurs de capacités, vous devez maintenant accéder à l'espace des DevOps agents dans lequel vous souhaitez examiner les tickets d' ServiceNow incident, sélectionner Capabilities → Communications, puis enregistrer l' ServiceNow instance que vous avez enregistrée sur la page Capability Providers. À présent, tout doit être configuré, et tous les incidents pour lesquels l'appelant est défini sur « Administrateur des problèmes » (pour imiter les autorisations que vous avez accordées au AWS DevOps OAuth client) déclencheront une enquête sur la réponse à l'incident dans l'espace DevOps agent configuré. Vous pouvez le tester en créant un nouvel incident ServiceNow et en définissant le champ Caller de l'incident comme « Administrateur des problèmes ».



ServiceNow mises à jour des tickets

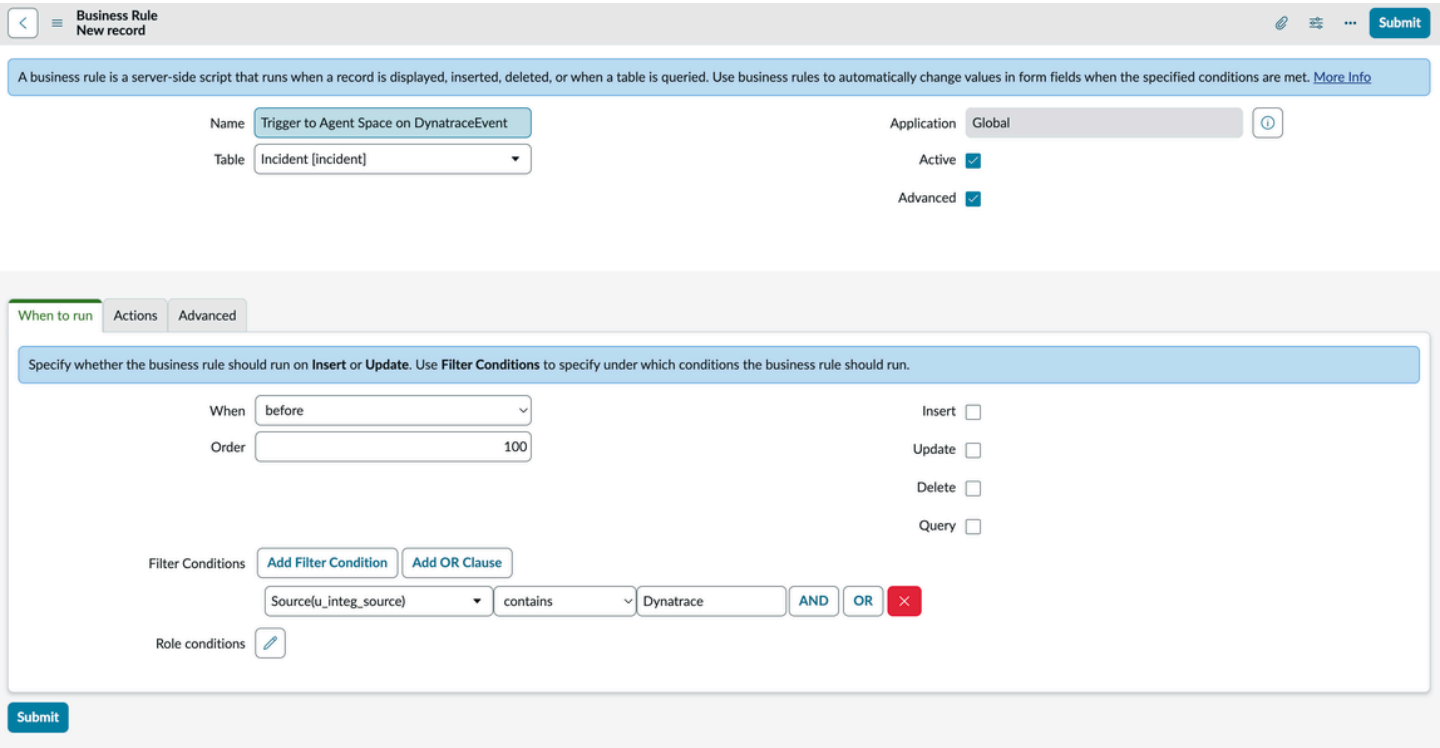
Au cours de toutes les enquêtes de réponse aux incidents déclenchées, votre DevOps agent fournira des mises à jour de ses principales conclusions, des analyses des causes profondes et des plans d'atténuation dans le ticket d'origine. Les conclusions de l'agent sont publiées dans les commentaires d'un incident, et nous ne publions actuellement que les dossiers des agents contenant le type`finding`,, `cause` `investigation_summary``mitigation_summary`, et les mises à jour de l'état de l'enquête (par exemple`AWS DevOps Agent started/finished its investigation`).

Exemples de routage et d'orchestration des tickets

Scénario : Filtrer les incidents envoyés à un espace d' DevOps agents

Il s'agit d'un scénario simple mais nécessite une certaine configuration ServiceNow pour créer un champ permettant de ServiceNow suivre la source de l'incident. Dans le cadre de cet exemple, créez un nouveau champ Source (u_source) à l'aide du générateur de formulaires SNOW. Cela permettra

de suivre la source de l'incident et de l'utiliser pour acheminer les demandes d'une source particulière vers un espace d' DevOps agent. Le routage est effectué en créant une règle métier de Service Now et en définissant dans l'onglet Quand exécuter les déclencheurs et les « Conditions de filtrage » dans l'onglet Quand exécuter. Dans cet exemple, les conditions du filtre sont définies comme suit :

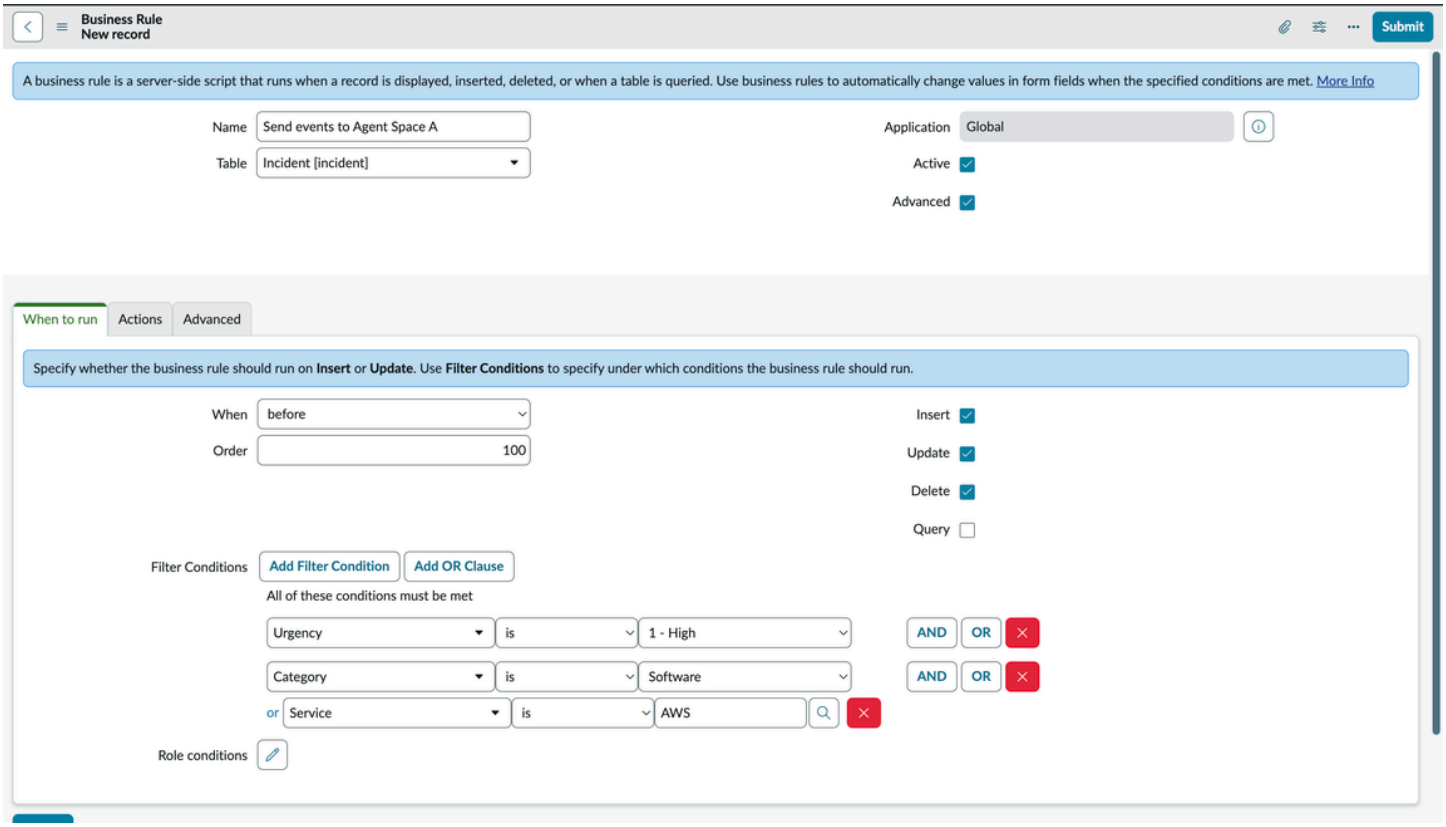


Scénario : routage des incidents sur plusieurs espaces d' DevOps agents

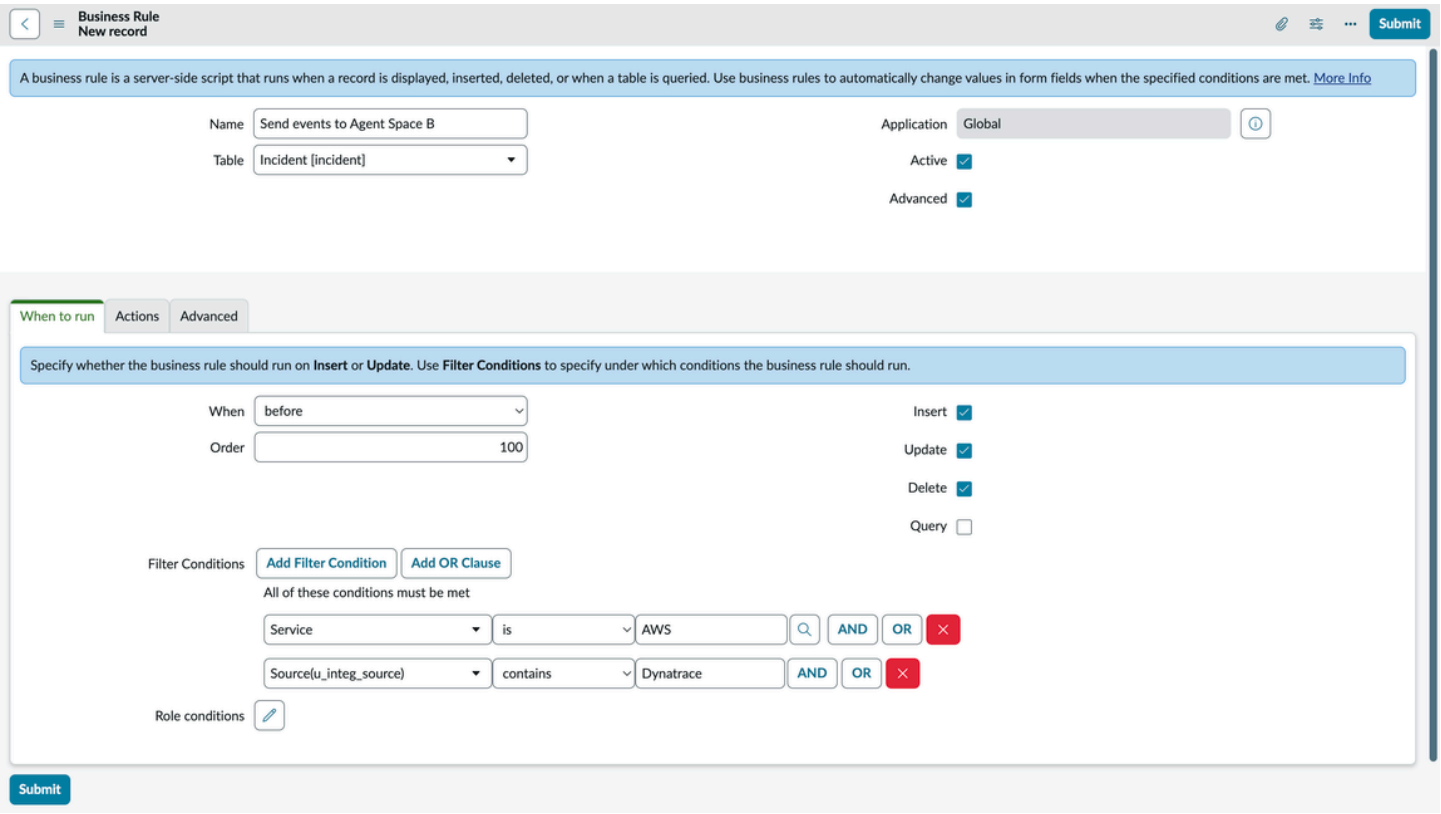
Cet exemple montre comment déclencher une enquête dans l'espace DevOps agent B lorsque l'urgence est1, la catégorie est `Software` ou le service est`AWS`, et déclencher une enquête dans l'espace DevOps agent A lorsque le service existe et que la source l'est`Dynatrace`. `AWS`

Ce scénario peut être réalisé de deux manières. Le script webhook lui-même peut être mis à jour pour inclure cette logique métier. Dans ce scénario, nous allons montrer comment y parvenir à l'aide d'une règle ServiceNow métier, pour plus de transparence et pour simplifier le débogage. Le routage est effectué en créant deux règles commerciales de Service Now.

- Créez une règle métier ServiceNow pour DevOps Agent Space A et créez une condition à l'aide du générateur de conditions pour envoyer uniquement les événements en fonction de la condition spécifiée.



- Créez ensuite une autre règle métier dans le champ AgentSpace B ServiceNow pour laquelle la règle métier ne se déclenchera que lorsque le service existe AWS et que la source est Dynatrace.



Désormais, lorsque vous créez un nouvel incident répondant à la condition spécifiée, il déclenche une enquête sur l'espace DevOps agent A ou sur l'espace DevOps agent B, ce qui vous permet de contrôler avec précision le routage des incidents.

Connecter Slack

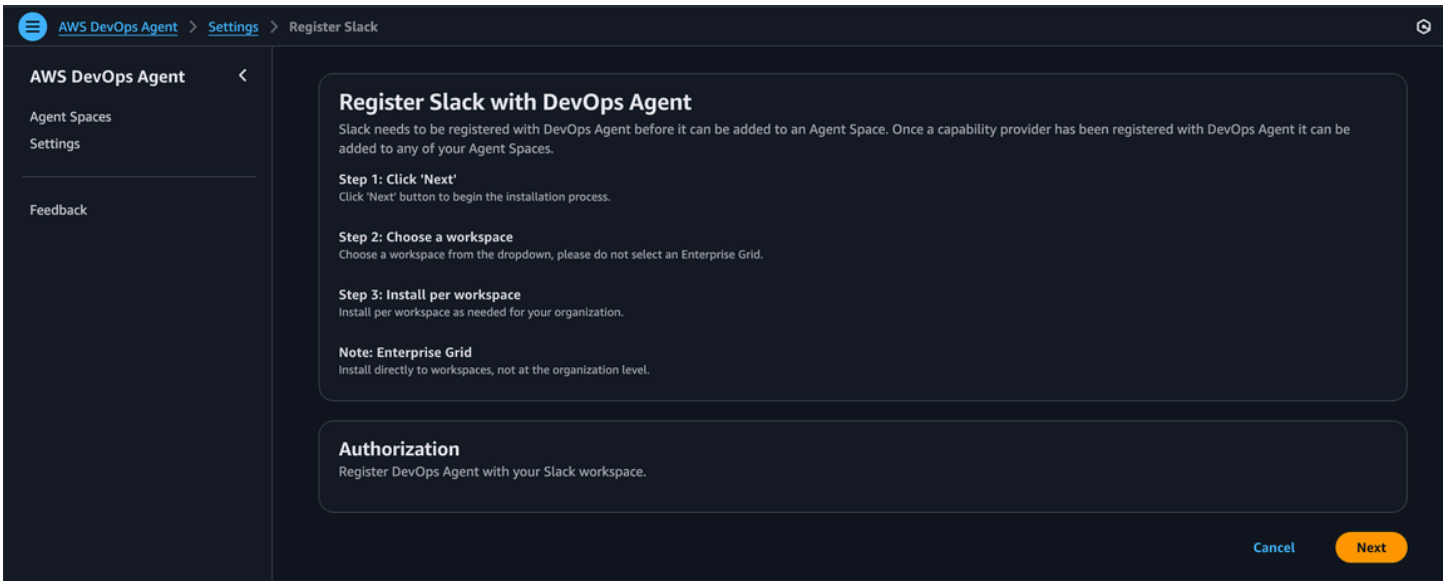
Vous pouvez configurer AWS DevOps l'Agent pour mettre à jour une chaîne Slack que vous sélectionnez en fonction des principaux résultats de l'enquête sur la réponse aux incidents, des analyses des causes profondes et des plans d'atténuation générés.

Avant de commencer

Slack doit être enregistré auprès de DevOps l'agent avant de pouvoir être ajouté à un espace agent. Pour intégrer AWS DevOps Agent à Slack, vous devez répondre aux exigences suivantes :

- Accédez à un espace de travail Slack avec la possibilité d'installer et d'autoriser des applications tierces
- Vous avez identifié les chaînes Slack sur lesquelles vous souhaitez que AWS DevOps l'agent envoie des notifications

Enregistrer l'intégration de Slack avec Agent AWS DevOps



1. Sur la page Fournisseurs de capacités de la console AWS DevOps Agent, recherchez Slack dans la section Fournisseurs disponibles sous Communication, puis cliquez sur Enregistrer.
2. Cliquez sur le bouton Enregistrer.
3. Vous serez redirigé vers Slack pour autoriser l'application AWS DevOps Agent pour votre espace de travail.
4. Sur la page d'autorisation de Slack, effectuez l'installation directement dans les espaces de travail, et non au niveau de l'organisation.
5. Choisissez un espace de travail dans le menu déroulant. Ne sélectionnez pas de réseau d'entreprise.
6. Effectuez l'installation par espace de travail selon les besoins de votre organisation.
7. Passez en revue les étendues demandées et cliquez sur Autoriser pour autoriser l'intégration.
8. Après autorisation, vous retournerez à la console de l' AWS DevOps agent.

Associez Slack à vos espaces d' DevOps agent

Après avoir enregistré Slack dans votre espace d' DevOps agent, vous pouvez l'associer à votre ou vos espaces d' DevOps agent :

1. Dans l'onglet Fonctionnalités de votre configuration AgentSpace, accédez à Communications > Slack.
2. Sélectionnez Ajouter Slack

3. Entrez l'ID de la chaîne
4. Choisissez Créer pour terminer la configuration de Slack.

> **Note**
>
> L'utilisateur bot de l'agent doit être ajouté aux chaînes privées pour que celui-ci puisse publier des messages.

> **Important**
>
> La désinstallation de l'application Slack peut empêcher sa réinstallation. Évitez de désinstaller l'application Slack.

Invocation de DevOps l'agent via Webhook

Les webhooks permettent aux systèmes externes de déclencher automatiquement les investigations des AWS DevOps agents. Cela permet l'intégration aux systèmes de billetterie, aux outils de surveillance et à d'autres plateformes qui peuvent envoyer des requêtes HTTP en cas d'incident.

Conditions préalables

Avant de configurer l'accès au webhook, assurez-vous d'avoir :

- Un espace d'agent configuré dans AWS DevOps l'agent
- Accès à la console de AWS DevOps l'agent
- Le système externe qui enverra les demandes de webhook

Types de webhooks

AWS DevOps L'agent prend en charge les types de webhooks suivants :

- Webhooks spécifiques à l'intégration : générés automatiquement lorsque vous configurez des intégrations tierces telles que Dynatrace, Splunk, Datadog, New Relic ou Slack. ServiceNow Ces webhooks sont associés à l'intégration spécifique et utilisent des méthodes d'authentification déterminées par le type d'intégration.

- Webhooks génériques : ils peuvent être créés manuellement pour déclencher des enquêtes à partir de n'importe quelle source non couverte par une intégration spécifique. Les webhooks génériques utilisent actuellement l'authentification HMAC (le jeton porteur n'est pas disponible actuellement).
- Webhooks d'alerte Grafana — Grafana peut envoyer des notifications d'alerte directement à l' AWS DevOps agent via les points de contact des webhooks. Pour les instructions de configuration, y compris un modèle de notification personnalisé, voir Connecting Grafana.

Méthodes d'authentification Webhook

La méthode d'authentification de votre webhook dépend de l'intégration à laquelle il est associé :

Authentification HMAC — Utilisée par :

- Webhooks d'intégration de Dynatrace
- Webhooks génériques (non liés à une intégration tierce spécifique)

Authentification par jeton au porteur : utilisée par :

- Webhooks d'intégration Splunk
- Webhooks d'intégration à Datadog
- Webhooks d'intégration New Relic
- ServiceNow webhooks d'intégration
- Webhooks d'intégration à Slack

Configuration de l'accès au webhook

Étape 1 : Accédez à la configuration du webhook

1. Connectez-vous à la console AWS de gestion et accédez à la console de l' AWS DevOps agent
2. Sélectionnez votre espace d'agent
3. Accédez à l'onglet Fonctionnalités
4. Dans la section Webhook, cliquez sur Configurer

Étape 2 : générer les informations d'identification du webhook

Pour les webhooks spécifiques à l'intégration :

Les webhooks sont automatiquement générés lorsque vous terminez la configuration d'une intégration tierce. L'URL et les informations d'identification du point de terminaison du webhook sont fournies à la fin du processus de configuration de l'intégration.

Pour les webhooks génériques :

1. Cliquez sur Générer un webhook
2. Le système générera une paire de clés HMAC
3. Stockez en toute sécurité la clé et le secret générés, vous ne pourrez plus les récupérer
4. Copiez l'URL du point de terminaison du webhook fournie

Étape 3 : Configuration de votre système externe

Utilisez l'URL et les informations d'identification du point de terminaison du webhook pour configurer votre système externe afin d'envoyer des demandes à l' AWS DevOps agent. Les étapes de configuration spécifiques dépendent de votre système externe.

Gestion des informations d'identification du webhook

Suppression des informations d'identification : pour supprimer les informations d'identification du webhook, accédez à la section de configuration du webhook et cliquez sur Supprimer. Après avoir supprimé les informations d'identification, le point de terminaison du webhook n'acceptera plus les demandes tant que vous n'aurez pas généré de nouvelles informations d'identification.

Régénération des informations d'identification : pour générer de nouvelles informations d'identification, supprimez d'abord les informations d'identification existantes, puis générez une nouvelle paire de clés ou un nouveau jeton.

Utilisation du webhook

Format de demande de webhook

Pour déclencher une enquête, votre système externe doit envoyer une requête HTTP POST à l'URL du point de terminaison du webhook.

Pour la version 1 (authentification HMAC) :

En-têtes :

- `Content-Type: application/json`
- `x-amzn-event-signature: <HMAC signature>`
- `x-amzn-event-timestamp: <+%Y-%m-%dT%H:%M:%S.000Z>`

La signature HMAC est générée en signant le corps de la demande avec votre clé secrète en utilisant SHA-256.

Pour la version 2 (authentification par jeton porteur) :

En-têtes :

- `Content-Type: application/json`
- `Authorization: Bearer <your-token>`

Corps de la demande :

Le corps de la demande doit inclure des informations sur l'incident :

```
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"
    }
  }
}
```

Exemple de code

Version 1 (authentification 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);
});
```

Version 1 (authentification 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"
```

Version 2 (authentification par jeton porteur) - : JavaScript

```
function sendEventToWebhook(webhookUrl, secret) {
  const timestamp = new Date().toISOString();

  const payload = {
    eventType: 'incident',
    incidentId: 'incident-123',
    action: 'created',
    priority: "HIGH",
    title: 'Test Alert',
    description: 'Test description',
    timestamp: timestamp,
    service: 'TestService',
    data: {}
  };

  fetch(webhookUrl, {
    method: "POST",
    headers: {
      "Content-Type": "application/json",
      "x-amzn-event-timestamp": timestamp,
      "Authorization": `Bearer ${secret}`,  // Fixed: template literal
    },
    body: JSON.stringify(payload),
  });
}
```

Version 2 (authentification par jeton porteur) - 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"
```

Résolution des problèmes liés aux webhooks

Si vous ne recevez pas de 200

Un 200 et un message tel que webhook reçus indiquent que l'authentification a été réussie et que le message a été mis en file d'attente pour que le système le vérifie et le traite. Si vous n'obtenez pas un 200 mais un 4xx, il y a probablement un problème avec l'authentification ou les en-têtes. Essayez d'envoyer manuellement à l'aide des options curl pour aider à débuguer l'authentification.

Si vous recevez un 200 mais qu'une enquête ne démarre pas

La cause probable est une charge utile mal formatée.

1. Vérifiez que l'horodatage et l'identifiant de l'incident sont à jour et uniques. Les messages dupliqués sont dédupliqués.
2. Vérifiez que le message est valide au format JSON
3. Vérifiez que le format est correct

Si vous recevez un 200$ et que l'enquête est immédiatement annulée

Vous avez probablement atteint la limite du mois. Adressez-vous à votre AWS contact pour demander une modification de la limite de taux, le cas échéant.

Rubriques en relation

- the section called "Création d'un espace d'agents"
- the section called "Qu'est-ce qu'une application Web pour DevOps agents ?"
- the section called "DevOps Autorisations IAM de l'agent"

Intégration de AWS DevOps l'agent à Amazon EventBridge

Vous pouvez intégrer AWS DevOps l'agent à vos applications axées sur les événements en utilisant les événements qui se produisent pendant les cycles de vie des enquêtes et des mesures d'atténuation. AWS DevOps L'agent envoie des événements à Amazon EventBridge lorsque l'état d'une enquête ou des mesures d'atténuation change. Vous pouvez ensuite créer des EventBridge règles qui prennent des mesures en fonction de ces événements.

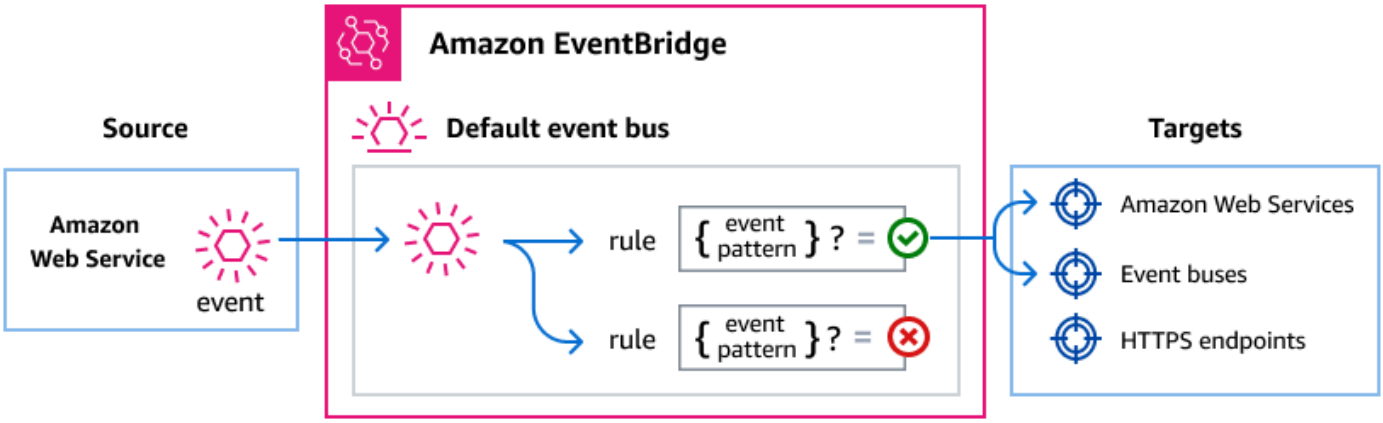
Par exemple, vous pouvez créer des règles qui exécutent les actions suivantes :

- Invoquez une fonction AWS Lambda pour traiter les résultats d'une enquête lorsqu'une enquête est terminée.
- Envoyez une notification Amazon SNS en cas d'échec ou d'expiration d'une enquête.
- Mettez à jour un système de billetterie lorsqu'une nouvelle enquête est créée.
- Démarrez un flux de travail AWS Step Functions lorsqu'une action d'atténuation est terminée.

Comment EventBridge achemine les événements des AWS DevOps agents

AWS DevOps L'agent envoie les événements au bus d'événements EventBridge par défaut. EventBridge évalue ensuite les événements par rapport aux règles que vous créez. Lorsqu'un événement correspond au modèle d'événement d'une règle, EventBridge envoie l'événement aux cibles spécifiées.

Le schéma suivant montre comment EventBridge achemine les événements de AWS DevOps l'agent.



1. AWS DevOps L'agent envoie un événement au bus d'événements EventBridge par défaut lorsque l'état du cycle de vie d'une investigation ou d'une atténuation change.
2. EventBridge évalue l'événement par rapport aux règles que vous avez créées.
3. Si l'événement correspond au modèle d'événement d'une règle, EventBridge envoie l'événement aux cibles spécifiées dans la règle.

AWS DevOps Événements pour les agents

AWS DevOps L'agent envoie les événements suivants à EventBridge. Tous les événements utilisent la source`aws.aidevops`.

Événements d'enquête pris en charge

detail-type	Description
`Investigation Created`	Une enquête a été créée dans l'espace des agents.
`Investigation Priority Updated`	La priorité d'une enquête a été modifiée.
`Investigation In Progress`	Une enquête a permis de lancer une analyse active.
`Investigation Completed`	Une enquête s'est terminée avec succès et a donné lieu à des conclusions.

detail-type	Description
`Investigation Failed`	Une enquête a rencontré une erreur et n'a pas pu être terminée.
`Investigation Timed Out`	Une enquête a dépassé la durée maximale autorisée.
`Investigation Cancelled`	Une enquête a été annulée avant d'être terminée.
`Investigation Pending Triage`	Une enquête est en attente de triage avant le début de l'analyse active.
`Investigation Linked`	Une enquête était liée à un incident ou à un ticket connexe.

Événements d'atténuation pris en charge

detail-type	Description
`Mitigation In Progress`	Une action d'atténuation a été lancée.
`Mitigation Completed`	Une action d'atténuation s'est terminée avec succès.
`Mitigation Failed`	Une action d'atténuation a rencontré une erreur et n'a pas pu être exécutée.
`Mitigation Timed Out`	Une mesure d'atténuation a dépassé la durée maximale autorisée.
`Mitigation Cancelled`	Une action d'atténuation a été annulée avant d'être terminée.

Pour obtenir des descriptions détaillées des champs et des exemples d'événements, voirthe section called “AWS DevOps Référence détaillée des événements de l'agent”.

Création de modèles d'événements correspondant aux événements de AWS DevOps l'agent

EventBridge les règles utilisent des modèles d'événements pour sélectionner les événements et les acheminer vers des cibles. Un modèle d'événement correspond à la structure des événements qu'il gère. Vous créez des modèles d'événements pour filtrer les événements de l' AWS DevOps agent en fonction des champs d'événements.

Les exemples suivants présentent des modèles d'événements pour des cas d'utilisation courants.

Faites correspondre tous les événements des AWS DevOps agents

Le modèle d'événements suivant correspond à tous les événements de AWS DevOps l'Agent.

```
{
  "source": ["aws.aidevops"]
}
```

Faire correspondre uniquement les événements d'enquête

Le modèle d'événements suivant utilise une correspondance de préfixe pour sélectionner uniquement les événements du cycle de vie de l'investigation.

```
{
  "source": ["aws.aidevops"],
  "detail-type": [{"prefix": "Investigation"}]
}
```

Faire correspondre uniquement les événements d'achèvement et d'échec

Le schéma d'événements suivant correspond aux événements relatifs aux enquêtes achevées ou échouées et aux mesures d'atténuation.

```
{
  "source": ["aws.aidevops"],
  "detail-type": [
    "Investigation Completed",
    "Investigation Failed",
    "Mitigation Completed",
    "Mitigation Failed"
```

```
  ]
}
```

Faites correspondre des événements pour un espace d'agent spécifique

Le modèle d'événements suivant correspond aux événements d'un espace d'agents spécifique.

```
{
  "source": ["aws.aidevops"],
  "detail": {
    "metadata": {
      "agent_space_id": ["your-agent-space-id"]
    }
  }
}
```

Pour plus d'informations sur les modèles d'événements, consultez la section Modèles EventBridge d'événements Amazon dans le guide de EventBridge l'utilisateur Amazon.

EventBridge Autorisations Amazon

AWS DevOps L'agent n'a pas besoin d'autorisations supplémentaires pour transmettre des événements à EventBridge. Les événements sont automatiquement envoyés au bus d'événements par défaut.

Selon les cibles que vous configurez pour vos EventBridge règles, vous devrez peut-être ajouter des autorisations spécifiques. Pour plus d'informations sur les autorisations requises pour les cibles, consultez la section Utilisation de politiques basées sur les ressources pour Amazon EventBridge dans le guide de EventBridge l'utilisateur Amazon.

EventBridge Ressources supplémentaires

Pour plus d'informations sur EventBridge les concepts et la configuration, consultez les rubriques suivantes du guide de EventBridge l'utilisateur Amazon :

- EventBridge bus événementiels
- EventBridge événements
- EventBridge modèles d'événements
- EventBridge règles

- EventBridge cibles

AWS DevOps Référence détaillée des événements de l'agent

Les événements issus AWS des services ont des champs de métadonnées communs`source`, notamment `detail-type`account,,`region`, et`time`. Ces événements contiennent également un `detail` champ contenant des données spécifiques au service. Pour les événements d' AWS DevOps agent, le `source` est toujours `aws.aidevops` et le `detail-type` identifie l'événement spécifique.

Événements liés à l'enquête

Les `detail-type` valeurs suivantes identifient les événements d'enquête :

- `Investigation Created`
- `Investigation Priority Updated`
- `Investigation In Progress`
- `Investigation Completed`
- `Investigation Failed`
- `Investigation Timed Out`
- `Investigation Cancelled`
- `Investigation Pending Triage`
- `Investigation Linked`

Les `detail-type` champs `source` et sont inclus ci-dessous car ils contiennent des valeurs spécifiques pour les événements de AWS DevOps l'agent. Pour les définitions des autres champs de métadonnées inclus dans tous les événements, consultez la section Structure des événements dans Amazon EventBridge Events Reference.

La structure JSON des événements d'investigation est la suivante.

```
{
  . . .,
  "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`Identifie le type d'événement. Pour les événements d'enquête, il s'agit de l'un des noms d'événements répertoriés précédemment.

`source`Identifie le service qui a généré l'événement. Pour les événements de l' AWS DevOps agent, cette valeur est`aws.aidevops`.

`detail`Objet JSON contenant des données spécifiques à un événement. L'`detail`objet inclut les champs suivants :

- `version`(string) — Version du schéma du détail de l'événement. Actuellement`1.0.0`.
- `metadata.agent_space_id`(chaîne) — Identifiant unique de l'espace d'agent d'origine de l'événement.
- `metadata.task_id`(chaîne) — Identifiant unique de la tâche.
- `metadata.execution_id`(string) — Identifiant unique de l'exécution. Présent lorsqu'une exécution a été affectée à l'enquête.
- `data.task_type`(string) — Type de tâche. Valeur :`INVESTIGATION`.
- `data.priority`(string) — Le niveau de priorité.
  Valeurs :`CRITICAL`,`HIGH`,`MEDIUM`,`LOW`,`MINIMAL`.
- `data.status`(string) — L'état actuel.
  Valeurs :`PENDING_START`,`IN_PROGRESS`,`COMPLETED`,`FAILED`,`TIMED_OUT`,`CANCELLED`,`PENDING_TRIA`
- `data.created_at`(string) — Horodatage ISO 8601 lors de la création de la tâche.
- `data.updated_at`(string) — Horodatage ISO 8601 de la dernière mise à jour de la tâche.

- `data.summary_record_id`(chaîne) — Identifiant du compte rendu sommaire contenant les résultats de l'enquête. Inclus lorsqu'un résumé est généré pour l'enquête terminée. Vous pouvez récupérer le contenu du résumé via l'API de l' AWS DevOps agent en utilisant cet identifiant pour rechercher la notice du journal avec un type d'enregistrement de`investigation_summary_md`.

Exemple : événement d'enquête terminé

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

Exemple : échec de l'enquête

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

Événements d'atténuation

Les `detail-type` valeurs suivantes identifient les événements d'atténuation :

- `Mitigation In Progress`
- `Mitigation Completed`
- `Mitigation Failed`
- `Mitigation Timed Out`
- `Mitigation Cancelled`

Les `detail-type` champs `source` et sont inclus ci-dessous car ils contiennent des valeurs spécifiques pour les événements de AWS DevOps l'agent. Pour les définitions des autres champs de métadonnées inclus dans tous les événements, consultez la section Structure des événements dans Amazon EventBridge Events Reference.

Voici la structure JSON des événements d'atténuation.

```
{
  . . .,
  "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`Identifie le type d'événement. Pour les événements d'atténuation, il s'agit de l'un des noms d'événements répertoriés précédemment.

`source`Identifie le service qui a généré l'événement. Pour les événements de l' AWS DevOps agent, cette valeur est`aws.aidevops`.

`detail`Objet JSON contenant des données spécifiques à un événement. L'`detail`objet inclut les champs suivants :

- `version`(string) — Version du schéma du détail de l'événement. Actuellement`1.0.0`.
- `metadata.agent_space_id`(chaîne) — Identifiant unique de l'espace d'agent d'origine de l'événement.
- `metadata.task_id`(chaîne) — Identifiant unique de la tâche.
- `metadata.execution_id`(string) — Identifiant unique de l'exécution. Présent lorsqu'une exécution a été affectée à l'atténuation.
- `data.task_type`(string) — Type de tâche. Valeur :`INVESTIGATION`.

- `data.priority`(string) — Le niveau de priorité.
  Valeurs :`CRITICAL`,`HIGH`,`MEDIUM`,`LOW`,`MINIMAL`.
- `data.status`(string) — L'état actuel.
  Valeurs :`IN_PROGRESS`,`COMPLETED`,`FAILED`,`TIMED_OUT`,`CANCELLED`.
- `data.created_at`(string) — Horodatage ISO 8601 lors de la création de la tâche.
- `data.updated_at`(string) — Horodatage ISO 8601 de la dernière mise à jour de la tâche.
- `data.summary_record_id`(chaîne) — Identifiant de l'enregistrement récapitulatif contenant les résultats des mesures d'atténuation. Inclus lorsqu'un résumé est généré pour l'atténuation terminée. Vous pouvez récupérer le contenu du résumé via l'API de l' AWS DevOps agent en utilisant cet identifiant pour rechercher la notice du journal avec un type d'enregistrement de`mitigation_summary_md`.

Exemple : événement d'atténuation terminé

```
{
  "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"
```

```
    }
  }
}
```

Exemple : échec de l'atténuation

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

Logs et statistiques vendus

Vous pouvez surveiller les espaces de vos agents et les opérations de service en utilisant les CloudWatch statistiques et les journaux Amazon vendus. Cette rubrique décrit les CloudWatch métriques que l' AWS DevOps agent publie automatiquement sur votre compte et les journaux vendus que vous pouvez configurer pour la livraison vers vos destinations préférées.

Indicateurs vendus CloudWatch

AWS DevOps L'agent publie automatiquement les statistiques sur Amazon CloudWatch dans votre compte. Ces métriques sont disponibles sans aucune configuration. Vous pouvez les utiliser pour surveiller l'utilisation, suivre l'activité opérationnelle et créer des alarmes.

Rôle lié à un service

Pour que CloudWatch les statistiques Amazon soient publiées sur votre compte pour ce service, l' AWS DevOps agent créera automatiquement le rôle lié au service AWSServiceRoleForAIDevOps Service-Linked Role pour vous. Si le rôle IAM invoquant l'API ne dispose pas de l'autorisation appropriée, la création de la ressource échouera avec un. InvalidParameterException

> **Important**
>
> Les clients qui ont créé leur rôle AgentSpace avant le 13 mars 2026 devront créer manuellement le rôle lié au service AWSServiceRoleForAIDevOps pour que CloudWatch les statistiques relatives à AWS DevOps l'agent soient publiées sur leur compte.

Créer manuellement un rôle lié à un service (pour les clients existants)

Effectuez l'une des actions suivantes :

- Dans la console IAM, créez le rôle AWSServiceRoleForAIDevOps sous le service AWS DevOps Agent.
- À partir de la AWS CLI, exécutez la commande suivante :

```
aws iam create-service-linked-role --aws-service-name aidevops.amazonaws.com
```

Namespace

Toutes les métriques sont publiées dans l'espace de `AWS/AIDevOps` noms.

Dimensions

Toutes les mesures incluent la dimension suivante.

Dimension	Description
`AgentSpaceUUID`	Identifiant unique de l'espace des agents. Pour agréger les statistiques de tous les espaces d'agent de votre compte, utilisez des expressions CloudWatch mathématiques ou omettez le filtre de dimension.

Référence des métriques

Nom des métriques	Description	Unit	Fréquence de publication	Statistiques utiles
ConsumedChatRequests	Le nombre de demandes de chat consommées par un agent. Pour obtenir le nombre total de données de votre compte, utilisez les `SUM` statistiques pour toutes les `AgentSpaceUUID` dimensions.	Nombre	Toutes les 5 minutes	Somme, moyenne
ConsumedInvestigationTime	Le temps passé à mener des enquêtes dans un espace réservé aux agents.	Secondes	Toutes les 5 minutes	Somme, moyenne, maximale

Nom des métriques	Description	Unit	Fréquence de publication	Statistiques utiles
ConsumedEvaluationTime	Le temps passé à effectuer des évaluations dans un espace réservé aux agents.	Secondes	Toutes les 5 minutes	Somme, moyenne, maximale
TopologyCompletionCount	Le nombre de traitements topologiques achevés. AWS DevOps L'agent émet cette métrique lorsque le traitement d'une topologie est terminé, qu'il s'agisse de sa création initiale lors de l'intégration, d'une mise à jour manuelle ou d'une actualisation quotidienne planifiée.	Nombre	Dirigé par les événements (émis à chaque achèvement)	Somme, SampleCount

Afficher les métriques dans la CloudWatch console

1. Ouvrez la CloudWatch console.
2. Dans le panneau de navigation, choisissez Metrics (Métriques), puis choisissez All metrics (Toutes les métriques).
3. Choisissez l'espace de noms AWS/AIDevOps.

4. Choisissez Par AgentSpace pour afficher les statistiques relatives à vos espaces d'agent.

> **Note**
>
> Vous pouvez créer des CloudWatch alarmes sur ces métriques pour recevoir des notifications lorsque l'utilisation dépasse un seuil. Par exemple, créez une alarme `ConsumedChatRequests` pour surveiller la consommation des demandes de chat.

Conditions préalables

Avant de configurer la livraison des journaux, assurez-vous que vous disposez des éléments suivants :

- Un AWS compte actif avec accès à la console AWS DevOps Agent
- Un directeur IAM avec des autorisations pour la livraison CloudWatch des journaux APIs
- (Facultatif) Un bucket Amazon S3 ou un flux de diffusion Amazon Data Firehose, si vous prévoyez de les utiliser comme destinations de log

Journaux vended

AWS DevOps L'agent prend en charge les journaux vendus qui fournissent une visibilité sur les événements traités par vos espaces d'agent et les enregistrements de services. Les journaux automatiques utilisent l'infrastructure Amazon CloudWatch Logs pour acheminer les journaux vers votre destination préférée.

Pour utiliser les journaux vendus, vous devez configurer une destination de livraison. Les destinations suivantes sont prises en charge :

- Amazon CloudWatch Logs — Un groupe de logs dans votre compte
- Amazon S3 — Un compartiment S3 dans votre compte
- Amazon Data Firehose : un flux de diffusion Firehose sur votre compte

Types de journaux pris en charge

Un seul type de journal est pris en charge :`APPLICATION_LOGS`. Ce type de journal couvre tous les événements opérationnels émis par le service.

Types d'événements du journal

Le tableau suivant récapitule les événements enregistrés par l' AWS DevOps agent.

Événement	Description	Niveau du journal
Événement entrant reçu par l'agent	Un agent est déclenché par une source intégrée et reçoit un événement entrant (par exemple, un événement PagerDuty incident).	INFO
Événement entrant lié à l'agent annulé	Un événement entrant a été supprimé avant que l'agent ne le traite. Le journal inclut la raison (par exemple, des données mal formées).	À définir
Échec des communications sortantes de l'agent	Une communication sortante vers une intégration tierce a échoué. Le journal inclut l'ID de tâche et l'identifiant de destination (par exemple, une erreur d'authentification).	À définir
Création de topologie en file d'attente	Une tâche de création de topologie a été mise en file d'attente pour traitement.	INFO
La création de la topologie a commencé	Le traitement d'une tâche de création de topologie a commencé.	INFO

Événement	Description	Niveau du journal
Création de topologie terminée	Une tâche de création de topologie a terminé le traitement. Cet événement s'applique à la création initiale, aux mises à jour et aux actualisations quotidiennes.	INFO
Echec de la découverte des ressources	La découverte des ressource s lors de la création de la topologie a rencontré un échec.	ERROR
L'enregistrement du service a échoué	L'enregistrement du service rencontre une défaillance irréparable	ERROR
La validation du webhook échoue	Lorsque le webhook reçu par l'agent Devops ne correspond pas au schéma attendu	ERROR
Mises à jour de l'état de validation	Lors d'une association avec un espace Agent (primary/s econdary compte classique), le statut de validation passe de valide à non valide et vice versa (par exemple, en raison d'un rôle mal formé, qui n'est pas assumé par le service).	ERREUR/INFO

Permissions

AWS DevOps L'agent utilise les CloudWatch journaux vendus (autorisations V2) pour fournir les journaux. Pour configurer la livraison du journal, le rôle IAM qui configure la livraison doit disposer des autorisations suivantes :

- `aidevops:AllowVendedLogDeliveryForResource`— Nécessaire pour autoriser la livraison du journal pour la ressource d'espace de l'agent.
- Autorisations pour la livraison CloudWatch des journaux APIs (`logs:PutDeliverySourcelogs:PutDeliveryDestination`,`logs:CreateDelivery`,, et opérations associées).
- Autorisations spécifiques à la destination de livraison que vous avez choisie.

Pour connaître la politique IAM complète requise pour chaque type de destination, consultez les rubriques suivantes du guide de l'utilisateur Amazon CloudWatch Logs :

- Logs envoyés à CloudWatch Logs
- Journaux envoyés à Amazon S3
- Logs envoyés à Firehose

Configurer la livraison du journal (console)

AWS DevOps L'agent fournit deux emplacements dans la console AWS de gestion pour configurer la livraison des journaux :

- Page des paramètres d'enregistrement du service : configurez la livraison des journaux pour les événements de niveau de service. Ces journaux utilisent le service ARN (`arn:aws:aidevops:<region>:<account-id>:service/<account-id>`) comme ressource.
- Page Espace agent : configurez la diffusion du journal pour les événements spécifiques à un espace agent individuel. Ces journaux utilisent l'espace agent ARN (`arn:aws:aidevops:<region>:<account-id>:agentspace/<agent-space-id>`) comme ressource.

Pour configurer la livraison des journaux pour l'enregistrement d'un service

1. Ouvrez la console de AWS DevOps l'agent dans la console AWS de gestion.
2. Dans le panneau de navigation, sélectionnez Settings (Paramètres).
3. Dans l'onglet Capability Providers > Logs, sélectionnez Configurer.
4. Pour Type de destination, choisissez l'une des options suivantes :
5. CloudWatch Journaux : sélectionnez ou créez un groupe de journaux.

6. Amazon S3 — Entrez l'ARN du compartiment S3.
7. Amazon Data Firehose : sélectionnez ou créez un flux de diffusion Firehose.
8. Pour Paramètres supplémentaires (facultatif), vous pouvez définir les options suivantes :
   a. Pour la Sélection des champs, sélectionnez les noms de champs de journaux que vous souhaitez envoyer vers votre destination. Vous pouvez sélectionner les champs du journal d'accès et un sous-ensemble de champs du journal d'accès en temps réel.
   b. (Amazon S3 uniquement) Pour le Partitionnement, indiquez le chemin permettant de partitionner les données de votre fichier journal.
   c. (Amazon S3 uniquement) Pour le Format de nom de fichier compatible avec hive, vous pouvez cocher la case afin d'utiliser des chemins S3 compatibles avec Hive. Vous pourrez ainsi charger plus facilement de nouvelles données dans vos outils compatibles avec Hive.
   d. Pour Format de sortie, indiquez le format que vous souhaitez utiliser.
   e. Pour Délimiteur de champ, indiquez comment séparer les champs du journal.
9. Choisissez Enregistrer.
10. Vérifiez que le statut de livraison indique Actif.

Pour configurer la livraison des journaux pour un espace d'agent

1. Ouvrez la console de AWS DevOps l'agent dans la console AWS de gestion.
2. Choisissez l'espace d'agent que vous souhaitez configurer.
3. Dans l'onglet Configuration, choisissez Configurer.
4. Pour Type de destination, choisissez l'une des options suivantes :
5. CloudWatch Journaux : sélectionnez ou créez un groupe de journaux.
6. Amazon S3 — Entrez l'ARN du compartiment S3.
7. Amazon Data Firehose : sélectionnez ou créez un flux de diffusion Firehose.
8. Pour Paramètres supplémentaires — *optionnel*, vous pouvez définir les options suivantes :
   a. Pour la Sélection des champs, sélectionnez les noms de champs de journaux que vous souhaitez envoyer vers votre destination. Vous pouvez sélectionner les champs du journal d'accès et un sous-ensemble de champs du journal d'accès en temps réel.
   b. (Amazon S3 uniquement) Pour le Partitionnement, indiquez le chemin permettant de partitionner les données de votre fichier journal.

c. (Amazon S3 uniquement) Pour le Format de nom de fichier compatible avec hive, vous pouvez cocher la case afin d'utiliser des chemins S3 compatibles avec Hive. Vous pourrez ainsi charger plus facilement de nouvelles données dans vos outils compatibles avec Hive.

d. Pour Format de sortie, indiquez le format que vous souhaitez utiliser.

e. Pour Délimiteur de champ, indiquez comment séparer les champs du journal.

9. Choisissez Enregistrer.

10. Vérifiez que le statut de livraison indique Actif.

Configuration de la livraison des journaux (CloudWatch API)

Vous pouvez également utiliser l'API CloudWatch Logs pour configurer la livraison des journaux par programmation. La livraison d'un journal de travail comprend trois éléments :

- A DeliverySource— Représente la ressource d'espace de l' AWS DevOps agent qui génère les journaux.
- A DeliveryDestination— Représente la destination où les journaux sont écrits.
- Une livraison — Connecte une source de livraison à une destination de livraison.

Étape 1 : créer une source de diffusion

Utilisez l'PutDeliverySourceopération pour créer une source de livraison. Transmettez l'ARN de votre ressource d'espace AWS DevOps Agent et `APPLICATION_LOGS` spécifiez-le comme type de journal.

L'exemple suivant crée une source de diffusion pour un espace agent :

```
{
    "name": "my-agent-space-delivery-source",
    "resourceArn": "arn:aws:aidevops:us-east-1:123456789012:agentspace/my-agent-space-
id",
    "logType": "APPLICATION_LOGS"
}
```

L'exemple suivant crée une source de livraison pour le service :

```
{
    "name": "my-service-delivery-source",
```

```
    "resourceArn": "arn:aws:aidevops:us-east-1:123456789012:service",
    "logType": "APPLICATION_LOGS"
}
```

Étape 2 : créer une destination de livraison

Utilisez cette PutDeliveryDestinationopération pour configurer l'emplacement de stockage des journaux. Vous pouvez choisir Amazon CloudWatch Logs, Amazon S3 ou Amazon Data Firehose.

L'exemple suivant crée une destination CloudWatch Logs :

```
{
    "name": "my-cwl-destination",
    "deliveryDestinationConfiguration": {
        "destinationResourceArn": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/
aidevops/my-agent-space"
    },
    "outputFormat": "json"
}
```

L'exemple suivant crée une destination Amazon S3 :

```
{
    "name": "my-s3-destination",
    "deliveryDestinationConfiguration": {
        "destinationResourceArn": "arn:aws:s3:::my-aidevops-logs-bucket"
    },
    "outputFormat": "json"
}
```

L'exemple suivant crée une destination Amazon Data Firehose :

```
{
    "name": "my-firehose-destination",
    "deliveryDestinationConfiguration": {
        "destinationResourceArn": "arn:aws:firehose:us-
east-1:123456789012:deliverystream/my-aidevops-log-stream"
    },
    "outputFormat": "json"
}
```

> **Note**
>
> Si vous livrez des journaux entre comptes, vous devez les utiliser PutDeliveryDestinationPolicydans le compte de destination pour autoriser la livraison.

Si vous souhaitez utiliser CloudFormation, vous pouvez utiliser ce qui suit :

- Delivery
- DeliveryDestination
- DeliverySource

L'`ResourceArn` correspond à l'`AgentSpaceArn` et `LogType` doit être défini sur `APPLICATION_LOGS`, comme étant le type de journaux pris en charge.

Étape 3 : créer une livraison

Utilisez cette CreateDeliveryopération pour lier la source de livraison à la destination de livraison.

```
{
    "deliverySourceName": "my-agent-space-delivery-source",
    "deliveryDestinationArn": "arn:aws:logs:us-east-1:123456789012:delivery-
destination:my-cwl-destination"
}
```

AWS CloudFormation

Vous pouvez également configurer la livraison des journaux à l' AWS CloudFormation aide des ressources suivantes :

- AWS: :Journaux : : DeliverySource
- AWS: :Journaux : : DeliveryDestination
- AWS: :Logs : :Livraison

Défini `ResourceArn` sur l'espace de l' AWS DevOps agent ou sur l'ARN du service de l'agent, puis défini `LogType` sur`APPLICATION_LOGS`.

Référence d'un schéma de journal

AWS DevOps L'agent utilise un schéma de journal partagé pour tous les types d'événements. Tous les événements du journal n'utilisent pas tous les champs.

Le tableau suivant décrit les champs du schéma du journal.

Champ	Type	Description
event_timestamp	Long	Horodatage Unix indiquant le moment où l'événement s'est produit
resource_arn	String	ARN de la ressource qui a généré l'événement
identifiant_compte_facultatif	String	AWS ID de compte associé au journal.
niveau_facultatif	String	Niveau du journal :`INFO`,`WARN`, `ERROR`
identifiant_espace_agent facultatif	String	Identifiant de l'espace des agents.
identifiant_associatif facultatif	String	Identifiant d'association pour le journal.
statut_facultatif	String	État de l'opération de topologie.
identifiant_webhook facultatif	String	Identifiant du webhook.
URL_endpoint_mcp facultative	String	URL du point de terminaison du serveur MCP
type_de_service facultatif	String	Type de service :`DYNATRACE` ,`DATADOG`,`GITHUB`,`SLACK`,`SERVICENOW` .

Champ	Type	Description
URL_endpoint_du_service facultatif	String	URL du point de terminaison pour les intégrations tierces.
identifiant_de_service facultatif	String	Identifiant de la source.
request_id	String	Identifiant de demande pour établir une corrélation avec les tickets AWS CloudTrail ou les prendre en charge.
opération_optionnelle	String	Nom de l'opération qui a été effectuée.
type_de_tâche facultatif	String	Type de tâche de backlog de l'agent : ou `INVESTIGATION` `EVALUATION`
identifiant_de_tâche facultatif	String	Agent Backlog Task Identifia nt de la tâche du IDAgent backlog.
référence_optionnelle	String	Référence provenant d'une tâche d'agent (par exemple, un ticket Jira).
type_d'erreur facultatif	String	Error type (Type d'erreur)
message_d'erreur facultatif	String	Description de l'erreur en cas d'échec d'une opération.
détails_facultatifs	Chaîne (JSON)	Charge utile d'événements spécifiques au service qui contient les paramètres et les résultats des opérations.

Gérer et désactiver la livraison des journaux

Vous pouvez modifier ou supprimer la livraison des journaux à tout moment depuis la console de l' AWS DevOps agent dans la console AWS de gestion ou à l'aide de l'API CloudWatch Logs.

Gérer la livraison des journaux (console)

1. Ouvrez la console de AWS DevOps l'agent dans la console AWS de gestion.
2. Accédez à la page Paramètres (pour les journaux au niveau du service) ou à la page spécifique de l'espace agent (pour les journaux au niveau de l'espace agent).
3. Dans l'onglet Configuration (pour les journaux au niveau de l'espace de l'agent) ou dans l'onglet Fournisseurs de capacités > Journaux (pour les journaux au niveau du service), choisissez la diffusion à modifier.
4. Mettez à jour la configuration selon vos besoins et choisissez Enregistrer.

> Remarque : vous ne pouvez pas modifier le type de destination d'une livraison existante. Pour modifier le type de destination, supprimez la livraison actuelle et créez-en une nouvelle.

Désactiver la livraison du journal (console)

1. Ouvrez la console de AWS DevOps l'agent dans la console AWS de gestion.
2. Accédez à la page Paramètres (pour les journaux au niveau du service) ou à la page spécifique de l'espace agent (pour les journaux au niveau de l'espace agent).
3. Dans l'onglet Configuration (pour les journaux au niveau de l'espace de l'agent) ou dans l'onglet Fournisseurs de capacités > Journaux (pour les journaux au niveau du service), sélectionnez la diffusion à supprimer.
4. Choisissez Supprimer et confirmez.

Désactiver la livraison du journal (API)

Pour supprimer la livraison d'un journal à l'aide de l'API, supprimez les ressources dans l'ordre suivant :

1. Supprimez la livraison en utilisant DeleteDelivery.
2. Supprimez la source de diffusion en utilisant DeleteDeliverySource.

3. (Facultatif) Si la destination de livraison n'est plus nécessaire, supprimez-la en utilisant DeleteDeliveryDestination.

> **Important**
>
> Vous êtes responsable de la suppression des ressources de diffusion des journaux une fois que vous avez supprimé la ressource d'espace agent qui génère les journaux (par exemple, après avoir supprimé un espace agent). Si vous ne supprimez pas ces ressources, les configurations de livraison orphelines risquent de rester.

Tarification

L' AWS DevOps agent ne facture pas l'activation des journaux vendus. Cependant, vous pouvez être facturé pour la livraison, l'ingestion, le stockage ou l'accès, selon la destination de livraison des journaux que vous avez choisie. Pour plus de détails sur les tarifs, consultez Vended Logs dans l'onglet Logs d'Amazon CloudWatch Pricing.

Pour les tarifs spécifiques à la destination, consultez les informations suivantes :

- Tarification d'Amazon CloudWatch Logs
- Tarification Amazon S3
- Tarification Amazon Data Firehose

Connexion à des outils hébergés en privé

Vue d'ensemble des connexions privées

AWS DevOps L'agent peut être étendu à l'aide d'outils MCP (Model Context Protocol) personnalisés et d'autres intégrations qui permettent à l'agent d'accéder à des systèmes internes tels que des registres de packages privés, des plateformes d'observabilité auto-hébergées, de la documentation APIs interne et des instances de contrôle de source (voir :). Configuration des fonctionnalités de AWS DevOps l'agent Ces services s'exécutent souvent au sein d'un Amazon Virtual Private Cloud (Amazon VPC) avec un accès Internet public restreint ou inexistant, ce qui signifie que AWS DevOps l'agent ne peut pas les atteindre par défaut.

Les connexions privées pour AWS DevOps Agent vous permettent de connecter en toute sécurité votre espace agent aux services exécutés dans votre VPC sans les exposer à l'Internet public. Les connexions privées fonctionnent avec toute intégration qui doit atteindre un point de terminaison privé, y compris les serveurs MCP, les instances Grafana ou Splunk auto-hébergées et les systèmes de contrôle de source tels que GitHub Enterprise Server et Self-Managed. GitLab

> **Note**
>
> Si vos outils hébergés en privé envoient des demandes sortantes à l' AWS DevOps agent depuis votre VPC, ce trafic peut également être sécurisé à l'aide d'un point de terminaison VPC afin qu'il reste sur le réseau. AWS Par exemple, cela peut être utilisé avec des outils qui déclenchent l' DevOps agent via des événements webhook (voir :the section called "Invocation de DevOps l'agent via Webhook"). Pour de plus amples informations, veuillez consulter the section called "Points de terminaison d'un VPC AWS PrivateLink".

Comment fonctionnent les connexions privées

Une connexion privée crée un chemin réseau sécurisé entre AWS DevOps l'agent et une ressource cible dans votre VPC. En arrière-plan, l' AWS DevOps agent utilise Amazon VPC Lattice pour établir ce chemin de connectivité privé sécurisé. VPC Lattice est un service de mise en réseau d'applications qui vous permet de connecter, de sécuriser et de surveiller les communications entre les applications VPCs, les comptes et les types de calcul, sans gérer l'infrastructure réseau sous-jacente.

Lorsque vous créez une connexion privée, les événements suivants se produisent :

- Vous fournissez le VPC, les sous-réseaux et (éventuellement) les groupes de sécurité dotés d'une connectivité réseau avec votre service cible.
- AWS DevOps L'agent crée une passerelle de ressources gérée par des services et provisionne ses interfaces réseau élastiques (ENIs) dans les sous-réseaux que vous avez spécifiés.
- L'agent utilise la passerelle de ressources pour acheminer le trafic vers l'adresse IP ou le nom DNS de votre service cible via le chemin du réseau privé.

La passerelle de ressources est entièrement gérée par AWS DevOps l'Agent et apparaît sous forme de ressource en lecture seule dans votre compte (nomméeaidevops-{your-private-connection-name}). Vous n'avez pas besoin de le configurer ou de le maintenir. Les seules

ressources créées dans votre VPC se trouvent ENIs dans les sous-réseaux que vous spécifiez. Ils ENIs servent de point d'entrée au trafic privé et sont entièrement gérés par le service. Ils n'acceptent pas les connexions entrantes en provenance d'Internet, et vous conservez le contrôle total de leur trafic par le biais de vos propres groupes de sécurité.

Sécurité

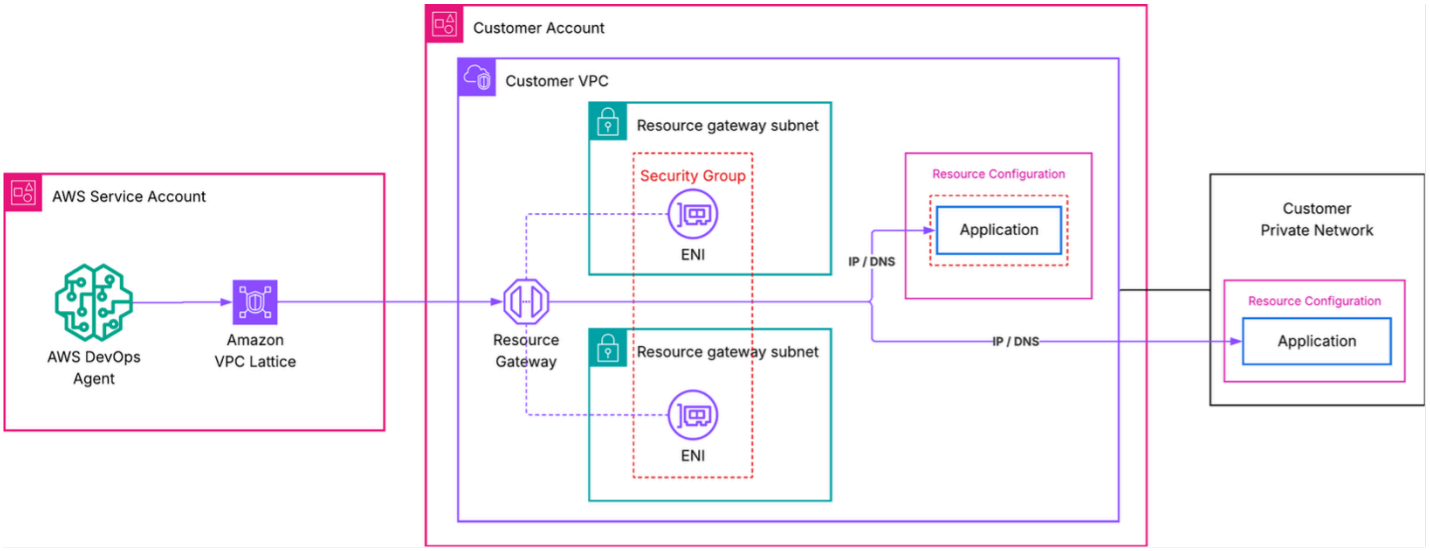
Les connexions privées sont conçues avec plusieurs niveaux de sécurité :

- Aucune exposition à Internet public : tout le trafic entre AWS DevOps l'agent et votre service cible reste sur le AWS réseau. Votre service n'a jamais besoin d'une adresse IP publique ou d'une passerelle Internet.
- Passerelle de ressources contrôlée par le service : la passerelle de ressources gérée par le service est en lecture seule dans votre compte. Il ne peut être utilisé que par AWS DevOps l'agent, et aucun autre service ou principal ne peut acheminer le trafic via celui-ci. Vous pouvez le vérifier dans les AWS CloudTrailjournaux, qui enregistrent tous les appels d'API VPC Lattice.
- Vos groupes de sécurité, vos règles — Vous contrôlez le trafic entrant et sortant vers les groupes ENIs de sécurité que vous possédez et gérez. Si vous ne spécifiez aucun groupe de sécurité, AWS DevOps l'agent crée un groupe de sécurité par défaut limité aux ports que vous définissez.
- Rôles liés à un service avec le moins de privilèges : AWS DevOps l'agent utilise un rôle lié à un service pour créer uniquement les ressources VPC Lattice et Amazon EC2 nécessaires. Ce rôle est limité aux ressources étiquetées avec `AWSAIDevOpsManaged` et ne peut accéder à aucune autre ressource de votre compte.

> **Note**
>
> Si votre organisation dispose de politiques de contrôle des services (SCPs) qui limitent les actions de l'API VPC Lattice, la passerelle de ressources gérée par les services est créée via un rôle lié à un service. Assurez-vous d' SCPs autoriser les actions nécessaires pour le rôle lié au service.

Architecture

Le schéma suivant montre le chemin réseau pour une connexion privée.



Dans cette architecture :

- AWS DevOps L'agent lance une demande auprès de votre service cible.
- Amazon VPC Lattice achemine la demande via la passerelle de ressources gérée par les services de votre VPC. Pour les configurations avancées utilisant vos propres ressources VPC Lattice, voir Configuration avancée à l'aide des ressources VPC Lattice existantes.
- Un ENI de votre VPC reçoit le trafic et le transmet à l'adresse IP ou au nom DNS de votre service cible.
- Vos groupes de sécurité déterminent le trafic autorisé via le ENIs.
- Du point de vue de votre service cible, la demande provient d'adresses IP privées situées ENIs au sein de votre VPC.

Création d'une connexion privée

Vous pouvez créer une connexion privée à l'aide de la console AWS de gestion ou de la AWS CLI.

> **Note**
>
> Les zones de disponibilité suivantes ne sont pas prises en charge par VPC Lattice :`use1-az3`,,`usw1-az2`,`apne1-az3`,, `apne2-az2euc1-az2`,`euw1-az4`. `cac1-az3` `ilc1-az2`

Conditions préalables

Avant de créer une connexion privée, vérifiez que vous disposez des éléments suivants :

- Un espace agent actif : vous avez besoin d'un espace agent existant dans votre compte. Si vous n'en avez pas, veuillez consulter Commencer à utiliser AWS DevOps Agent.
- Un service cible accessible de manière privée : votre serveur MCP, votre plateforme d'observabilité ou tout autre service doit être accessible via une adresse IP privée ou un nom DNS connu depuis le VPC sur lequel la passerelle de ressources est déployée. Le service peut s'exécuter dans le même VPC, un VPC pair ou sur site, à condition qu'il soit routable à partir des sous-réseaux de la passerelle de ressources. Le service doit desservir le trafic HTTPS avec une version TLS minimale de 1.2 sur un port que vous spécifiez lors de la création de la connexion.
- Sous-réseaux de votre VPC : identifiez 1 à 20 sous-réseaux dans lesquels ENIs ils seront créés. Nous vous recommandons de sélectionner des sous-réseaux dans plusieurs zones de disponibilité pour une haute disponibilité. Ces sous-réseaux doivent disposer d'une connectivité réseau avec votre service cible. Un sous-réseau par zone de disponibilité peut être utilisé par VPC Lattice.
- (Facultatif) Groupes de sécurité — Si vous souhaitez contrôler le trafic à l'aide de règles spécifiques, préparez jusqu'à cinq groupes IDs de sécurité à associer au ENIs. Si vous omettez les groupes de sécurité, AWS DevOps l'agent crée un groupe de sécurité par défaut.

Les connexions privées sont des ressources au niveau du compte. Après avoir créé une connexion privée, vous pouvez la réutiliser dans plusieurs intégrations et espaces d'agent qui doivent atteindre le même hôte.

Création d'une connexion privée à l'aide de la console

1. Ouvrez la console de AWS DevOps l'agent.
2. Dans le volet de navigation, choisissez Capability providers, puis Private connections.
3. Choisissez Créer une nouvelle connexion.
4. Dans Nom, entrez un nom descriptif pour la connexion, tel que `my-mcp-tool-connection`.
5. Pour le VPC, sélectionnez le VPC sur lequel la passerelle ENIs de ressources sera déployée.
6. Pour les sous-réseaux, sélectionnez un ou plusieurs sous-réseaux (jusqu'à 20). Nous vous recommandons de choisir des sous-réseaux dans au moins deux zones de disponibilité.
7. Pour le type d'adresse IP, sélectionnez le type d'adresse IP de votre service cible (`IPv4`, `IPv6`, ou `DualStack`).

8. (Facultatif) Pour Nombre d' IPv4 adresses, si vous avez sélectionné IPv4 Dualstack pour le type d'adresse IP, vous pouvez saisir le nombre d' IPv4 adresses par ENI pour votre passerelle de ressources. La valeur par défaut est de 16 IPv4 adresses par ENI.
9. (Facultatif) Pour les groupes de sécurité, sélectionnez les groupes de sécurité existants (jusqu'à 5) afin de limiter le trafic autorisé à atteindre votre service cible. Si vous n'en sélectionnez aucun, un groupe de sécurité par défaut est créé.
10. (Facultatif) Pour les plages de ports, spécifiez les ports TCP que votre application cible écoute (par exemple, `443` ou `8080-8090`). Vous pouvez spécifier jusqu'à 11 plages de ports.
11. Pour Adresse de l'hôte, entrez l'adresse IP ou le nom DNS de votre service cible (par exemple, `mcp.internal.example.com` ou `10.0.1.50`). Le service doit être accessible depuis le VPC sélectionné. Si vous choisissez un nom DNS, il doit pouvoir être résolu à partir du VPC sélectionné.
12. (Facultatif) Pour la clé publique du certificat, si l'adresse d'hôte que vous avez spécifiée utilise des certificats TLS émis par une autorité de certification privée, entrez la clé publique codée PEM du certificat. Cela permet à AWS DevOps l'agent de faire confiance à la connexion TLS à votre service cible.
13. Choisissez Créer une connexion.

L'état de la connexion passe à Créer en cours. Ce processus peut prendre jusqu'à 10 minutes. Lorsque le statut passe à Actif, le chemin réseau est prêt.

Si le statut passe à l'état « échec de la création », vérifiez les points suivants :

- Les sous-réseaux que vous avez spécifiés ont des adresses IP disponibles.
- Votre compte n'a pas atteint les quotas de service VPC Lattice.
- Aucune politique IAM restrictive n'empêche le rôle lié au service de créer des ressources.

> **Note**
>
> Ces étapes peuvent également être effectuées en sélectionnant un fournisseur de capacités `Create a new private connection` lors de l'enregistrement. Pour plus d'informations, voir Utiliser une connexion privée avec un fournisseur de fonctionnalités.

Création d'une connexion privée à l'aide de la AWS CLI

Exécutez la commande suivante pour créer une connexion privée. Remplacez les valeurs de l'espace réservé par les vôtres.

```
aws devops-agent create-private-connection \
    --name my-mcp-tool-connection \
    --mode '{
        "serviceManaged": {
            "hostAddress": "mcp.internal.example.com",
            "vpcId": "vpc-0123456789abcdef0",
            "subnetIds": [
                "subnet-0123456789abcdef0",
                "subnet-0123456789abcdef1"
            ],
            "securityGroupIds": [
                "sg-0123456789abcdef0"
            ],
            "portRanges": ["443"]
        }
    }'
```

La réponse inclut le nom de la connexion et le statut suivant `CREATE_IN_PROGRESS` :

```
{
    "name": "my-mcp-tool-connection",
    "status": "CREATE_IN_PROGRESS",
    "resourceGatewayId": "rgw-0123456789abcdef0",
    "hostAddress": "mcp.internal.example.com",
    "vpcId": "vpc-0123456789abcdef0"
}
```

Pour vérifier l'état de la connexion, utilisez la `describe-private-connection` commande suivante :

```
aws devops-agent describe-private-connection \
    --name my-mcp-tool-connection
```

Lorsque le statut est défini`ACTIVE`, votre connexion privée est prête à être utilisée.

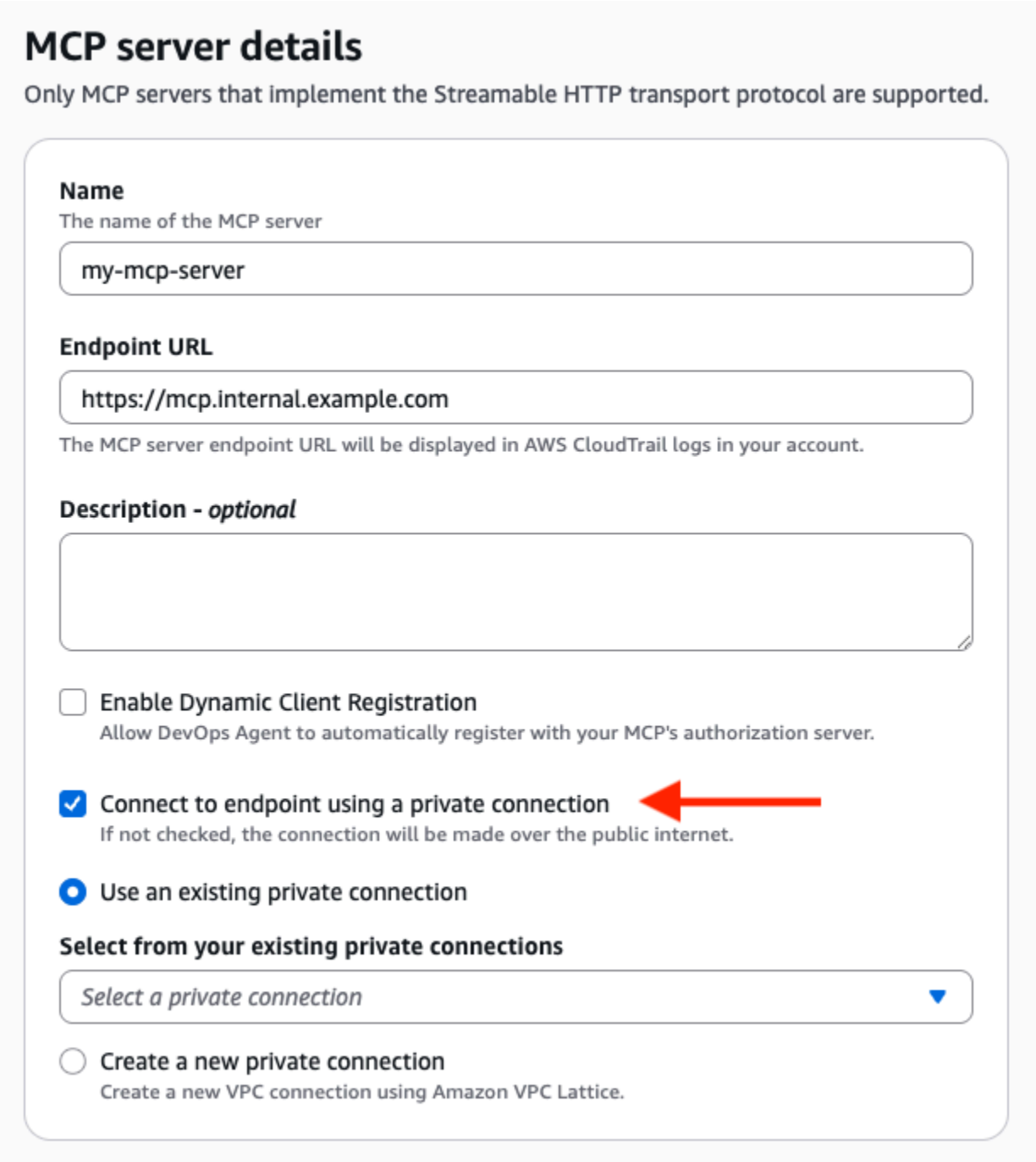
Utiliser une connexion privée avec un fournisseur de fonctionnalités

Pour utiliser une connexion privée, vous pouvez créer un lien vers celle-ci lors de l'enregistrement d'un fournisseur de fonctionnalités. Les fonctionnalités prises en charge qui peuvent être utilisées avec des connexions privées sont les suivantes : `GitHub` `GitLabMCP Server`,, et`Grafana`. Vous pouvez effectuer cette étape à l'aide de la console AWS de gestion ou de la AWS CLI.

> **Note**
>
> Lors de l'enregistrement d'un fournisseur de fonctionnalités, AWS DevOps l'agent vérifie que le point de terminaison est accessible et répond. Assurez-vous que votre service cible fonctionne et accepte les connexions avant de terminer l'enregistrement.

Utiliser une connexion privée avec un fournisseur de fonctionnalités à l'aide de la console

Dans la console de l' AWS DevOps agent, les connexions privées peuvent être liées à une fonctionnalité lors de l'enregistrement en sélectionnant l'option « Se connecter au point de terminaison à l'aide d'une connexion privée ».



1. Ouvrez la console de l' AWS DevOps agent et accédez à votre espace agent.
2. Dans la section Fournisseurs de capacités, choisissez Registration.
3. Sélectionnez Enregistrer pour le type de fonctionnalité que vous souhaitez utiliser avec la connexion privée.

4. Dans la vue des détails de l'enregistrement, entrez l'URL du point de terminaison auquel vous souhaitez vous connecter à l'aide de la connexion privée (par exemple,`https://mcp.internal.example.com`).
5. Sélectionnez Se connecter au point de terminaison à l'aide d'une connexion privée.
6. Sélectionnez une connexion privée existante qui correspond à l'URL du point de terminaison auquel vous souhaitez vous connecter, ou sélectionnez Créer une nouvelle connexion privée pour en créer une.
7. Terminez le processus d'enregistrement pour le fournisseur de capacités.

Utiliser une connexion privée avec un fournisseur de capacités à l'aide de la AWS CLI

Vous pouvez enregistrer des fonctionnalités avec une connexion privée en incluant l'`private-connection-name`argument. Vous trouverez ci-dessous un exemple d'enregistrement d'un serveur MCP avec une autorisation par clé d'API à l'aide de la connexion `my-mcp-tool-connection` privée. Remplacez les valeurs de l'espace réservé par les vôtres.

```
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
```

Vérifier une connexion privée

Une fois que la connexion privée a atteint l'état Active et a été utilisée par un fournisseur de fonctionnalités, vérifiez que AWS DevOps l'agent peut atteindre votre service cible :

1. Ouvrez la console de l' AWS DevOps agent et accédez à votre espace agent.
2. Démarrez une nouvelle session de chat.
3. Appelez une commande qui utilise l'intégration soutenue par votre connexion privée. Par exemple, si votre outil MCP donne accès à une base de connaissances interne, posez à l'agent une question qui nécessite cette base de connaissances.
4. Vérifiez que l'agent renvoie les résultats du service privé.

Si la connexion échoue, vérifiez les points suivants :

- Limites du réseau VPC : vérifiez que vous n'avez atteint aucune passerelle de ressources ni aucune autre limite de quota du réseau VPC
- Règles relatives aux groupes de sécurité : vérifiez que les groupes de sécurité attachés au ENIs réseau autorisent le trafic sortant sur le port sur lequel votre service écoute. Vérifiez également que le groupe de sécurité de votre service autorise le trafic entrant sur le port cible. Le trafic arrive du plan de données VPC Lattice dans la plage de IPs votre VPC CIDR. Vous pouvez utiliser le référencement du groupe de sécurité (en autorisant le groupe de sécurité ENI comme source) ou autoriser le trafic entrant depuis le CIDR VPC.
- Connectivité des sous-réseaux : vérifiez que les sous-réseaux que vous avez sélectionnés peuvent acheminer le trafic vers votre service. Si le service s'exécute dans un sous-réseau différent, vérifiez que les tables de routage autorisent le trafic entre elles.
- Disponibilité du service — Vérifiez que votre service fonctionne et accepte les connexions sur le port prévu.
- Zone de disponibilité non prise en charge : vérifiez que vos sous-réseaux se trouvent dans des zones de disponibilité prises en charge. Exécutez `aws ec2 describe-subnets --subnet-ids <your-subnet-ids> --query 'Subnets[*].[SubnetId,AvailabilityZoneId]'` et vérifiez par rapport aux zones de disponibilité non prises en charge répertoriées ci-dessus.

Supprimer une connexion privée

Vous pouvez supprimer les connexions privées non utilisées à l'aide de la console AWS de gestion ou de la AWS CLI.

Supprimer une connexion privée à l'aide de la console

1. Ouvrez la console de AWS DevOps l'agent.

2. Dans le volet de navigation, choisissez Capability providers, puis Private connections.
3. Sélectionnez le menu Actions pour la connexion privée que vous souhaitez supprimer, puis sélectionnez Supprimer.

La connexion privée sera affichée avec le statut « Suppression de la connexion » pendant que l' AWS DevOps agent supprime la passerelle de ressources gérées et ENIs de votre VPC. Une fois la suppression terminée, la connexion n'apparaît plus dans votre liste de connexions privées.

Supprimer une connexion privée à l'aide de la AWS CLI

```
aws devops-agent delete-private-connection \
    --name my-mcp-tool-connection
```

La réponse renvoie un statut de`DELETE_IN_PROGRESS`. AWS DevOps L'agent supprime la passerelle de ressources gérées et la supprime ENIs de votre VPC. Une fois la suppression terminée, la connexion n'apparaît plus dans votre liste de connexions privées.

Configuration avancée à l'aide des ressources VPC Lattice existantes

Si votre organisation utilise déjà Amazon VPC Lattice et gère ses propres configurations de ressources, vous pouvez créer une connexion privée en mode autogéré. Au lieu de demander à AWS DevOps l'agent de créer une passerelle de ressources pour vous, vous fournissez le nom de ressource Amazon (ARN) d'une configuration de ressources existante qui pointe vers votre service cible.

Cette approche est utile lorsque vous :

- Vous souhaitez avoir un contrôle total sur la passerelle de ressources et le cycle de vie de configuration des ressources.
- Nécessité de partager les configurations de ressources entre plusieurs AWS comptes ou services.
- Exigez des journaux d'accès VPC Lattice pour une surveillance détaillée du trafic.
- Exécutez une architecture hub-and-spoke réseau.

Pour créer une connexion privée autogérée avec la AWS CLI :

```
aws devops-agent create-private-connection \
    --name my-advanced-connection \
```

```
    --mode '{
        "selfManaged": {
            "resourceConfigurationId": "arn:aws:vpc-lattice:us-
east-1:123456789012:resourceconfiguration/rcfg-0123456789abcdef0"
        }
    }'
```

Pour plus de détails sur la configuration des passerelles de ressources VPC Lattice et des configurations de ressources, consultez le guide de l'utilisateur Amazon VPC Lattice.

Rubriques en relation

- the section called “Points de terminaison d'un VPC AWS PrivateLink”
- the section called “Connexion de serveurs MCP”
- Configuration des fonctionnalités de AWS DevOps l'agent
- AWS DevOps Sécurité des agents
- the section called “DevOps Autorisations IAM de l'agent”

AWS DevOps Sécurité des agents

Ce document fournit des informations sur les considérations de sécurité, la protection des données, les contrôles d'accès et les fonctionnalités de conformité de AWS DevOps l'Agent. Utilisez ces informations pour comprendre comment AWS DevOps l'agent est conçu pour répondre à vos exigences de sécurité et de conformité.

Sécurité à plusieurs niveaux

AWS DevOps L'agent implémente la sécurité à plusieurs niveaux. Même si des autorisations plus larges sont accordées au rôle IAM de l'agent, celui-ci applique ses propres contrôles d'accès internes afin de limiter la portée de ses actions. Par exemple, si un client ajoute une politique IAM d'accès complète à Amazon S3 au rôle IAM de l' AWS DevOps agent, celui-ci veillera à ce que seuls les journaux lus après le `AWSLogs` préfixe soient lus à des fins de résolution des problèmes.

Nous recommandons de suivre le principe du moindre privilège lors de la configuration des autorisations IAM pour AWS DevOps l'agent et de mettre en œuvre la sécurité à plusieurs niveaux. Une défense approfondie garantit qu'aucune erreur de configuration ne peut compromettre la sécurité de votre environnement.

Espaces réservés aux agents

Les espaces d'agent constituent la principale limite de sécurité dans AWS DevOps Agent. Chaque espace d'agent :

- Fonctionne indépendamment avec ses propres configurations et autorisations
- Définit AWS les comptes et les ressources auxquels l'agent peut accéder
- Établit des connexions à des plateformes tierces

Les espaces d'agent maintiennent une isolation stricte afin de garantir la sécurité et d'empêcher tout accès involontaire entre différents environnements ou équipes.

Traitement régional et flux de données

AWS DevOps L'agent opère dans le monde entier avec des capacités de traitement régionales. L'agent récupère les données opérationnelles des AWS régions de tous les AWS comptes autorisés

à accéder au sein de l'espace agent configuré. Cette collecte de données multi-régions entre comptes garantit une analyse complète des incidents tout en respectant les limites géographiques pour le traitement des inférences.

Utilisation d'Amazon Bedrock et inférence entre régions

AWS DevOps L'agent sélectionnera automatiquement la région optimale de votre zone géographique pour traiter vos demandes d'inférence. Cela permet d'optimiser les ressources informatiques disponibles, la disponibilité des modèles et d'offrir la meilleure expérience client. Vos données resteront stockées uniquement dans la région où votre espace agent a été créé. Toutefois, les demandes de saisie et les résultats de sortie peuvent être traités en dehors de cette région, comme décrit dans la liste suivante. Toutes les données seront transmises chiffrées sur le réseau sécurisé d’Amazon.

AWS DevOps L'agent acheminera en toute sécurité vos demandes d'inférence vers les ressources informatiques disponibles dans la zone géographique d'origine de la demande, comme suit :

- Les demandes d'inférence provenant de l'Union européenne seront traitées au sein de l'Union européenne.
- Les demandes d'inférence provenant des États-Unis seront traitées aux États-Unis.
- Les demandes d'inférence provenant de l'Australie seront traitées en Australie.
- Les demandes d'inférence provenant du Japon seront traitées au Japon.
- Si une demande d'inférence provient d'une zone non répertoriée, elle sera traitée par défaut aux États-Unis d'Amérique.
- DevOps Agent et Bedrock ne sont pas concernés par les politiques relatives aux clients énoncées dans Service Control Policies (SCPs) ou Control Tower qui limitent le contenu client à des régions spécifiques
- Bedrock peut utiliser des régions autres que la région d'origine au sein de votre zone géographique pour effectuer une inférence apatride afin d'optimiser les performances et la disponibilité

Gestion des identités et des accès

Méthodes d’authentification

AWS DevOps L'agent propose deux méthodes d'authentification pour se connecter à l'application Web AWS DevOps Agent Space :

- AWS Intégration à Identity Center : la principale méthode d'authentification utilise la OAuth version 2.0 avec une authentification basée sur les sessions à l'aide de cookies HTTP uniquement. AWS Identity Center peut se fédérer avec des fournisseurs d'identité externes via les protocoles OIDC et SAML standard, notamment des fournisseurs tels qu'Okta, Ping Identity et Microsoft Entra ID. Cette méthode prend en charge l'authentification multifactorielle par le biais de votre fournisseur d'identité. AWS Identity Center utilise par défaut des sessions d'une durée maximale de 12 heures et peut être configuré selon la durée souhaitée.
- Lien d'authentification IAM : une autre méthode fournit un accès direct à l'application Web depuis la console de AWS gestion à l'aide de jetons JWT dérivés d'une session de console de gestion existante AWS . Cette option est utile pour évaluer l' AWS DevOps agent avant de mettre en œuvre l'intégration complète d'Identity Center, ainsi que pour obtenir un accès administratif si l'application Web de l' AWS DevOps agent devient inaccessible via l'authentification basée sur Identity Center. Les séances sont limitées à 10 minutes.

Rôles IAM

AWS DevOps L'agent utilise les rôles IAM pour définir les autorisations d'accès :

- Rôle de compte principal : accorde à l'agent l'accès aux ressources du AWS compte sur lequel vous créez l'espace d'agent ainsi que l'accès aux rôles de compte secondaires.
- Rôles de compte secondaires — Permet à l'agent d'accéder aux ressources de AWS comptes supplémentaires connectés à l'espace agent.
- Rôle d'application Web — Permet aux utilisateurs d'accéder aux données et aux résultats des enquêtes de l' AWS DevOps agent dans l'application Web.

Ces rôles doivent être configurés selon le principe du moindre privilège, en accordant uniquement les autorisations de lecture seule nécessaires aux enquêtes.

Protection des données

Chiffrement des données

AWS DevOps L'agent chiffre toutes les données des clients :

- Chiffrement au repos : toutes les données sont AWS chiffrées à l'aide de clés gérées.

- Chiffrement en transit : tous les journaux, indicateurs, éléments de connaissances, métadonnées des tickets et autres données récupérés sont chiffrés pendant le transfert au sein du réseau privé de l'agent et vers des réseaux externes.

Stockage et conservation des données

Les données sont stockées dans la région où votre espace d'agent a été créé, tandis que le traitement des inférences peut avoir lieu dans votre zone géographique, comme décrit dans la section sur l'utilisation d'Amazon Bedrock ci-dessus.

Informations personnelles identifiables (PII)

AWS DevOps L'agent ne filtre pas les informations personnelles lorsqu'il résume les données collectées lors d'enquêtes, d'évaluations de recommandations ou de réponses au chat. Il est recommandé de supprimer les données d'identification personnelle avant de les stocker dans des journaux d'observabilité.

Journal de l'agent et journalisation des audits

Journal de l'agent

Les fonctionnalités d'investigation et de prévention des incidents tiennent à jour des journaux détaillés qui :

- Enregistrez chaque étape de raisonnement et chaque action entreprise
- Créez une transparence totale dans les processus décisionnels des agents
- Ne peut pas être modifié par les agents une fois enregistré, ce qui permet de minimiser les attaques telles que l'injection rapide pour masquer des actions importantes
- Inclure tous les messages de chat de la page d'investigation

AWS CloudTrail intégration

Tous les appels d'API de l' AWS DevOps agent sont automatiquement AWS CloudTrail capturés par le AWS compte d'hébergement. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer :

- La demande qui a été faite à l'agent
- L'adresse IP à partir de laquelle la demande a été effectuée
- La personne ayant effectué la demande
- Le moment où la demande a été formulée

Protection contre les injections rapides

Une attaque par injection rapide se produit lorsqu'un attaquant intègre des instructions malveillantes dans des données externes, telles qu'une page Web ou un document, qu'un système d'IA générative traitera ultérieurement. AWS DevOps L'agent consomme nativement de nombreuses sources de données dans le cadre de ses opérations normales, notamment les journaux, les balises de ressources et d'autres données opérationnelles. AWS DevOps L'agent protège contre les attaques par injection rapide grâce aux mesures de protection ci-dessous, mais il est important de s'assurer que toutes les sources de données connectées et l'accès des utilisateurs à ces sources de données sont fiables. Voir la section Modèle de responsabilité partagée pour en savoir plus.

Garanties d'injection rapide :

- Capacités d'écriture limitées — Les outils mis à la disposition de l'agent ne sont pas en mesure de modifier les ressources, à l'exception de l'ouverture de tickets et de demandes d'assistance. Cela empêche les instructions malveillantes de modifier votre infrastructure ou vos applications.
- Application des limites de compte — AWS DevOps L'agent n'opère que dans les limites autorisées par les rôles assignés à l'agent dans le compte principal et le AWS compte secondaire connecté. L'agent ne peut pas accéder aux ressources ou les modifier en dehors de son périmètre configuré.
- Protections de sécurité basées sur l'IA — AWS DevOps L'agent utilise des modèles dotés de protections de niveau 3 (ASL-3) basées sur l'IA. Ces protections incluent des classificateurs qui détectent et empêchent les attaques par injection rapide avant qu'elles n'affectent le comportement des agents.
- Piste d'audit immuable : le journal de l'agent enregistre chaque étape de raisonnement et chaque action entreprise. Les entrées du journal ne peuvent pas être modifiées par l'agent une fois enregistrées, ce qui empêche les attaques par injection rapide de masquer des actions malveillantes.

Bien que AWS DevOps l'agent fournisse plusieurs niveaux de protection contre les attaques par injection rapide, certaines configurations peuvent augmenter les risques :

- Outils de serveur MCP personnalisés — La fonctionnalité bring-your-own MCP vous permet d'introduire des outils personnalisés dans l'agent, ce qui peut offrir des opportunités supplémentaires d'injection rapide. Les outils personnalisés peuvent ne pas avoir les mêmes contrôles de sécurité que les outils d' AWS DevOps agent natifs, et des instructions malveillantes peuvent potentiellement exploiter ces outils de manière involontaire. Voir la section Modèle de responsabilité partagée pour en savoir plus.
- Attaques d'utilisateurs autorisés — Les utilisateurs autorisés à opérer dans les limites du AWS compte ou des outils connectés ont plus de chances de tenter une attaque contre l'agent. Ces utilisateurs peuvent avoir la possibilité de modifier les sources de données consommées par l'agent, telles que les journaux ou les balises de ressources, afin de faciliter l'intégration d'instructions malveillantes que l'agent traitera.

Pour atténuer ces risques :

1. Passez en revue et testez attentivement les serveurs MCP personnalisés avant de les déployer dans Agent Spaces.
   a. Assurez-vous qu'ils ne sont autorisés à effectuer que des actions en lecture seule
   b. Vérifiez que les utilisateurs des outils externes auxquels accèdent les serveurs MCP sont des entités fiables, car les AWS DevOps agents interagissant avec MCP s'appuient sur la relation de confiance implicite établie entre ces utilisateurs d'outils et l'agent AWS DevOps
2. Appliquer le principe du moindre privilège lorsque vous accordez aux utilisateurs l'accès aux systèmes qui fournissent des données à l'agent
3. Vérifiez régulièrement quels serveurs MCP sont connectés à vos agents Spaces
4. Étant donné que tout contenu extrait de la liste d'autorisation URLs peut tenter de manipuler le comportement de l'agent, n'incluez que des sources fiables dans votre liste d'autorisation.

Sécurité de l'intégration

AWS DevOps L'agent prend en charge plusieurs types d'intégration, chacun ayant son propre modèle de sécurité :

- Intégrations bidirectionnelles natives : intégrations intégrées qui peuvent envoyer des données à l'agent et recevoir des mises à jour de la part de l'agent. Cela utilise les méthodes d'authentification du fournisseur

- Serveurs MCP : serveurs Remote Model Context Protocol qui utilisent des flux d'authentification OAuth 2.0 et des clés d'API pour communiquer en toute sécurité avec des systèmes externes.
- Déclencheurs Webhook : déclencheurs d'investigation provenant de services distants tels que des tickets ou des systèmes d'observabilité. Les webhooks utilisent le code d'authentification des messages basé sur le hachage (HMAC) pour des raisons de sécurité.
- Communication sortante : les intégrations telles que Slack et les systèmes de billetterie reçoivent des mises à jour de l'agent mais ne prennent pas encore en charge la communication bidirectionnelle.

Fournisseurs d'enregistrement

Certains outils externes sont authentifiés au niveau du compte et partagés entre tous les espaces d'agent du compte. Lorsque vous enregistrez ces outils, vous vous authentifiez une fois au niveau du compte, puis chaque espace agent peut se connecter à des ressources spécifiques au sein de cette connexion enregistrée.

Les outils suivants utilisent l'enregistrement au niveau du compte :

- GitHub— Utilise OAuth le flux pour l'authentification. Une fois enregistré GitHub au niveau du compte, chaque agent Space peut se connecter à des référentiels spécifiques au sein de votre GitHub organisation.
- Dynatrace — Utilise OAuth l'authentification par jeton. Après avoir enregistré Dynatrace au niveau du compte, chaque agent Space peut se connecter à des environnements Dynatrace ou à des configurations de surveillance spécifiques.
- Slack — Utilise l'authentification par OAuth jeton. Après avoir enregistré Slack au niveau du compte, chaque espace agent peut se connecter à des chaînes Slack spécifiques.
- Datadog — Utilise MCP avec un OAuth flux pour l'authentification. Après avoir enregistré Datadog au niveau du compte, chaque Agent Space peut se connecter à des ressources de supervision Datadog spécifiques.
- New Relic — Utilise l'authentification par clé d'API. Après avoir enregistré New Relic au niveau du compte, chaque agent Space peut se connecter à des configurations de surveillance New Relic spécifiques.
- Splunk — Utilise l'authentification par jeton porteur. Après avoir enregistré Splunk au niveau du compte, chaque agent Space peut se connecter à des sources de données Splunk spécifiques.

- GitLab— Utilise l'authentification par jeton d'accès. Une fois enregistré GitLab au niveau du compte, chaque agent Space peut se connecter à des GitLab référentiels spécifiques.
- ServiceNow— Utilise key/token l'authentification OAuth du client. Après s'être enregistré ServiceNow au niveau du compte, chaque espace agent peut se connecter à des ServiceNow instances ou à des files d'attente de tickets spécifiques.
- Serveurs MCP distants accessibles au grand public : utilisez le OAuth flux pour l'authentification. Après avoir enregistré un serveur MCP distant au niveau du compte, chaque agent Space peut se connecter à des ressources spécifiques exposées par ce serveur.

La connectivité réseau

AWS DevOps L'agent se connecte à vos systèmes tiers et à vos serveurs MCP distants pour effectuer des enquêtes et d'autres opérations.

Trafic entrant de l' AWS DevOps agent vers vos systèmes

AWS DevOps L'agent initie des connexions sortantes vers vos systèmes tiers et vos serveurs MCP distants, qui arrivent sous forme de trafic entrant vers votre infrastructure. La façon dont vous sécurisez ce trafic dépend de la manière dont vos outils sont hébergés :

- Outils hébergés en privé : si vos outils sont accessibles depuis un AWS VPC, vous pouvez utiliser les connexions privées des AWS DevOps agents pour isoler le trafic des AWS réseaux et le maintenir hors de l'Internet public. Pour de plus amples informations, veuillez consulter the section called “Connexion à des outils hébergés en privé”.
- Outils hébergés publiquement : si vos outils sont accessibles via Internet public et utilisent des listes d'adresses IP autorisées ou des règles de pare-feu, vous devez autoriser le trafic entrant provenant des adresses IP sources des AWS DevOps agents suivantes :
  - Asie-Pacifique (Sydney) (ap-southeast-2)
    - `13.237.95.197`
    - `13.238.84.102`
  - Asie-Pacifique (Tokyo) (ap-northeast-1)
    - `13.192.12.233`
    - `35.74.181.230`
    - `57.183.50.158`
  - Europe (Francfort) (eu-central-1)

    - 18.158.110.140
    - 52.57.96.160
    - 52.59.55.56
- Europe (Irlande) (eu-west-1)
    - 34.251.85.24
    - 52.30.157.157
    - 52.51.192.222
- USA Est (Virginie du Nord) (us-east-1)
    - 34.228.181.128
    - 44.219.176.187
    - 54.226.244.221
- USA Ouest (Oregon) (us-west-2)
    - 34.212.16.133
    - 52.89.67.212
    - 54.187.135.61

Trafic sortant de votre AWS DevOps VPC vers l'agent

Pour le trafic sortant de votre AWS VPC AWS DevOps vers l'agent (par exemple, the section called "Invocation de DevOps l'agent via Webhook" en utilisant), vous pouvez utiliser des points de terminaison VPC pour isoler ce trafic réseau des réseaux. AWS Pour de plus amples informations, veuillez consulter the section called "Points de terminaison d'un VPC AWS PrivateLink".

Modèle de responsabilité partagée

AWS responsabilités

AWS est chargé de :

- Maintien de la sécurité des données récupérées par l'agent
- Sécurisation des outils natifs mis à la disposition de l'agent
- Protection de l'infrastructure qui exécute AWS DevOps l'agent

Responsabilités client

Les clients sont responsables de :

- Gestion de l'accès des utilisateurs à l'espace des agents
- Limiter l'accès aux utilisateurs fiables des systèmes externes qui fournissent des entrées à l'agent, tels que les services et les ressources qui produisent des journaux, CloudTrail des événements, des tickets, etc., qui peuvent être utilisés pour tenter une injection rapide malveillante.
- Assurez-vous que toutes les sources de données connectées disposent de données fiables peu susceptibles d'être utilisées pour tenter des attaques par injection rapide
- Garantir le fonctionnement sécurisé des intégrations de serveurs bring-your-own MCP
- S'assurer que les rôles IAM attribués à l'agent sont correctement définis
- Rédaction des données d'identification personnelle avant de les stocker dans les journaux d'observabilité et autres sources de données des agents
- Respect de la pratique recommandée consistant à n'accorder des autorisations en lecture seule qu'aux sources de données connectées, y compris bring-your-own les serveurs MCP

Utilisation des données

AWS n'utilise pas les données des agents, les messages de chat ou les données provenant de sources de données intégrées pour entraîner des modèles ou améliorer le produit. The AWS DevOps Agent Space utilise les commentaires des clients intégrés au produit pour améliorer les réponses et les enquêtes des agents, mais AWS ne les utilise pas pour améliorer le service lui-même.

Conformité d’

Lors de la version préliminaire, AWS DevOps l'agent n'est pas conforme aux normes telles que SOC 2, PCI-DSS, ISO 27001 ou FedRAMP. AWS annoncera les certifications de conformité qui seront disponibles ultérieurement.

DevOps Autorisations IAM de l'agent

AWS DevOps L'agent utilise des actions AWS Identity and Access Management (IAM) spécifiques au service pour contrôler l'accès à ses fonctionnalités et capacités. Ces actions déterminent ce que les utilisateurs peuvent faire dans la console AWS DevOps Agent et l'Operator Web App. Cela

est distinct des autorisations d'API de AWS service que l'agent lui-même utilise pour étudier vos ressources.

Pour plus d'informations sur la limitation de l'accès des agents, voir Limiter l'accès des agents dans un AWS compte.

Actions de gestion de l'espace agent

Ces actions contrôlent l'accès à la configuration et à la gestion de l'espace agent :

- aidevops : GetAgentSpace — Permet aux utilisateurs de consulter les détails d'un espace agent, notamment sa configuration, son statut et les comptes associés. Les utilisateurs ont besoin de cette autorisation pour accéder à un espace d'agent dans la console AWS de gestion.
- aidevops : GetAssociation — Permet aux utilisateurs d'afficher les détails d'une association de comptes spécifique, notamment la configuration du rôle IAM et l'état de la connexion.
- aidevops : ListAssociations — Permet aux utilisateurs de répertorier toutes les associations de AWS comptes configurées pour un espace agent, y compris les comptes principaux et secondaires.

Actions d'investigation et d'exécution

Ces actions contrôlent l'accès aux fonctionnalités d'enquête sur les incidents :

- aidevops : ListExecutions — Permet aux utilisateurs de consulter les métadonnées d'exécution, notamment l'identifiant, le statut, etc., pour les enquêtes, les mesures d'atténuation, les évaluations et les conversations par chat associées à une tâche.
- aidevops : ListJournalRecords — Permet aux utilisateurs d'accéder à des journaux détaillés qui indiquent les étapes de raisonnement de l'agent, les actions entreprises et les sources de données consultées lors d'une enquête, d'une atténuation, d'une évaluation et d'une conversation par chat. Cela est utile pour comprendre comment l'agent est parvenu à ses conclusions.

Actions de gestion du chat

Le chat nécessite les autorisations IAM suivantes pour fonctionner :

- aidevops : ListChats — Permet aux utilisateurs de répertorier et d'accéder à l'historique des conversations par chat.

- aidevops : CreateChat — Permet aux utilisateurs de créer de nouvelles conversations par chat.
- aidevops : SendMessage — Permet aux utilisateurs de soumettre des requêtes et de recevoir des réponses en streaming.

Actions de topologie et de découverte

Ces actions contrôlent l'accès aux fonctionnalités de mappage des ressources de l'application :

- aidevops : DiscoverTopology — Permet aux utilisateurs de déclencher la découverte de la topologie et le mappage d'un espace agent. Cette action lance le processus d'analyse des AWS comptes et de création de la topologie des ressources de l'application.

Actions de prévention et de recommandation

Les actions suivantes contrôlent l'accès à la fonctionnalité de prévention :

- aidevops : ListGoals — Permet aux utilisateurs de visualiser les buts et objectifs de prévention poursuivis par l'agent en fonction des récents modèles d'incidents.
- aidevops : ListRecommendations — Permet aux utilisateurs de consulter toutes les recommandations générées par la fonctionnalité de prévention, y compris leur priorité et leur catégorie.
- aidevops : GetRecommendation — Permet aux utilisateurs de consulter des informations détaillées sur une recommandation spécifique, y compris les incidents qu'elle aurait évités et les conseils de mise en œuvre.

Actions de gestion des tâches en attente

Ces actions contrôlent la capacité à gérer les recommandations sous forme de tâches en attente :

- aidevops : CreateBacklogTask — Permet aux utilisateurs de créer une tâche d'enquête sur les incidents ou d'évaluation de la prévention.
- aidevops : UpdateBacklogTask — Permet aux utilisateurs d'approuver un plan d'atténuation ou d'annuler une enquête ou une évaluation en cours.
- aidevops : GetBacklogTask — Permet aux utilisateurs de récupérer les détails d'une tâche spécifique.

- aidevops : ListBacklogTasks — Permet aux utilisateurs de répertorier les tâches d'un espace agent, filtrées par type de tâche, statut, priorité ou heure de création.

Actions de gestion des connaissances

Ces actions contrôlent la possibilité d'ajouter et de gérer des connaissances personnalisées que l'agent peut utiliser au cours des enquêtes :

- aidevops : CreateKnowledgeItem — Permet aux utilisateurs d'ajouter des éléments de connaissances personnalisés, tels que des compétences, des guides de dépannage ou des informations spécifiques à l'application auxquelles l'agent doit se référer.
- aidevops : ListKnowledgeItems — Permet aux utilisateurs de visualiser tous les éléments de connaissances configurés pour un espace agent.
- aidevops : GetKnowledgeItem — Permet aux utilisateurs de récupérer les détails d'un élément de connaissance spécifique.
- aidevops : UpdateKnowledgeItem — Permet aux utilisateurs de modifier les éléments de connaissances existants pour maintenir les informations à jour.
- aidevops : DeleteKnowledgeItem — Permet aux utilisateurs de supprimer les éléments de connaissances qui ne sont plus pertinents.

AWS Support aux actions d'intégration

Ces actions contrôlent l'intégration avec les dossiers de AWS Support :

- aidevops : InitiateChatForCase — Permet aux utilisateurs de démarrer une session de chat avec le AWS Support directement à partir d'une enquête, en fournissant automatiquement le contexte de l'incident.
- aidevops : EndChatForCase — Permet aux utilisateurs de mettre fin à une session de discussion active sur un dossier AWS Support.
- aidevops : DescribeSupportLevel — Permet aux utilisateurs de vérifier le niveau du plan de AWS support du compte afin de déterminer les options de support disponibles.

Actions d'utilisation et de surveillance

Ces actions contrôlent l'accès aux informations d'utilisation :

- aidevops : GetAccountUsage — Permet aux utilisateurs de consulter le quota mensuel de l' AWS DevOps agent pour les heures d'investigation, les heures d'évaluation de la prévention et les demandes de chat, ainsi que l'utilisation du mois en cours.

Exemples de politiques IAM courantes

Politique administrateur

Cette politique accorde un accès complet à toutes les fonctionnalités de AWS DevOps l'agent :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "aidevops:*",
      "Resource": "*"
    }
  ]
}
```

Politique de l'opérateur

Cette politique donne accès aux fonctionnalités d'investigation et de prévention sans fonctionnalités administratives :

```
{
  "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": "*"
    }
  ]
}
```

Politique de lecture seule

Cette politique accorde un accès en lecture seule aux enquêtes et aux recommandations :

```
{
  "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": "*"
    }
  ]
}
```

Utilisation de rôles liés à un service pour l'agent AWS DevOps

AWS DevOps L'agent utilise des AWS rôles liés au service Identity and Access Management (IAM). Un rôle lié à un service est un type unique de rôle IAM directement lié à l'agent. AWS DevOps Les rôles liés au service sont prédéfinis par AWS DevOps l'agent et incluent toutes les autorisations dont le service a besoin pour appeler d'autres AWS services en votre nom.

Autorisations de rôles liés à un service

Le rôle lié à un service `AWSServiceRoleForAIDevOps` fait confiance au mandataire de service `aidevops.amazonaws.com` qui assume le rôle.

Le rôle utilise la politique gérée `AWSServiceRoleForAIDevOpsPolicy` avec les autorisations suivantes :

- `cloudwatch:PutMetricData`— Publiez les métriques d'utilisation dans l'espace de `AWS/AIDevOps` CloudWatch noms. Délimité par une `cloudwatch:namespace` condition autorisant uniquement l'espace de `AWS/AIDevOps` noms.
- `vpc-lattice:CreateResourceGateway`— Créez des passerelles de ressources VPC Lattice pour les connexions privées. Délimité par une `aws:RequestTag/AWSAIDevOpsManaged` condition afin que le service ne puisse créer que des passerelles de ressources portant le `AWSAIDevOpsManaged` tag.
- `vpc-lattice:TagResource`— Marquez les passerelles de ressources VPC Lattice. Délimité par une `aws:RequestTag/AWSAIDevOpsManaged` condition.
- `vpc-lattice:DeleteResourceGateway`— Supprimez les passerelles de ressources VPC Lattice. Délimité par une `aws:ResourceTag/AWSAIDevOpsManaged` condition afin que le service ne puisse supprimer que les passerelles de ressources qu'il a créées.
- `vpc-lattice:GetResourceGateway`— Récupère des informations sur les passerelles de ressources VPC Lattice. Délimité par une `aws:ResourceTag/AWSAIDevOpsManaged` condition afin que le service ne puisse lire que les passerelles de ressources qu'il a créées.

- `ec2:DescribeVpcs`,`ec2:DescribeSubnets`, `ec2:DescribeSecurityGroups` — Récupérez des informations sur les ressources réseau VPC requises pour configurer les passerelles de ressources. Ces actions en lecture seule s'appliquent à toutes les ressources VPC car l'API EC2 ne prend pas en charge les autorisations au niveau des ressources pour les appels Describe.
- `iam:CreateServiceLinkedRole`— Créez le rôle lié au service VPC Lattice requis pour les opérations de passerelle de ressources. Cette autorisation est limitée au principal du `vpc-lattice.amazonaws.com` service et ne peut pas être utilisée pour créer des rôles liés au service pour un autre service.

Création du rôle lié à un service

Vous n'avez pas besoin de créer manuellement un rôle lié au service `AWSServiceRoleForAIDevOps`. Lorsque vous commencez à utiliser l' AWS DevOps Agent, le service crée le rôle lié au service pour vous.

Pour autoriser le service à créer le rôle en votre nom, vous devez en avoir l'`iam:CreateServiceLinkedRole`autorisation. Nous recommandons de définir la portée de cette autorisation `aidevops.amazonaws.com` à la `iam:AWSServiceName` condition de respecter le principe du moindre privilège. Pour plus d'informations, consultez Autorisations des rôles liés à un service.

Modifier le rôle lié à un service

Vous ne pouvez pas modifier le rôle lié à un service `AWSServiceRoleForAIDevOps`. Une fois le rôle créé, vous ne pouvez pas le modifier car différentes entités peuvent le référencer par son nom. Néanmoins, vous pouvez modifier la description du rôle à l'aide d'IAM. Pour plus d'informations, consultez la section Modification d'un rôle lié à un service.

Suppression du rôle lié à un service

Si vous n'avez plus besoin d'utiliser l' AWS DevOps Agent, nous vous recommandons de supprimer le rôle `AWSServiceRoleForAIDevOps` lié au service. Avant de pouvoir supprimer le rôle, vous devez d'abord supprimer toutes les connexions privées configurées dans votre espace agent. La suppression du rôle lié au service ne supprime pas automatiquement les passerelles de ressources VPC Lattice `AWSAIDevOpsManaged` qui ont été précédemment créées par le service. Vous devez supprimer ces passerelles de ressources manuellement si elles ne sont plus nécessaires. Pour plus d'informations, consultez Supprimer un rôle lié à un service.

AWS Politiques gérées pour AWS DevOps l'agent

AWS répond à de nombreux cas d'utilisation courants en fournissant des politiques IAM autonomes créées et administrées par. AWS Ces politiques AWS gérées accordent les autorisations nécessaires pour les cas d'utilisation courants afin que vous puissiez éviter d'avoir à rechercher les autorisations nécessaires. Pour plus d'informations, consultez les politiques AWS gérées dans le _Guide de l'utilisateur IAM_.

Les politiques AWS gérées suivantes, que vous pouvez associer aux utilisateurs de votre compte, sont spécifiques à AWS DevOps Agent.

AIDevOpsAgentReadOnlyAccess

Fournit un accès en lecture seule à Amazon DevOps Agent via la console AWS de gestion

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "AIDevOpsAgentReadOnlyAccess",
   "Effect": "Allow",
   "Action": [
    "aidevops:Get*",
    "aidevops:List*",
    "aidevops:SearchServiceAccessibleResource"
   ],
   "Resource": "*"
  }
 ]
}
```

AIDevOpsAgentFullAccess

Fournit un accès complet à Amazon DevOps Agent via la console AWS de gestion

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "AIDevOpsAgentSpaceAccess",
   "Effect": "Allow",
   "Action": [
```

```
    "aidevops:CreateAgentSpace",
    "aidevops:DeleteAgentSpace",
    "aidevops:GetAgentSpace",
    "aidevops:ListAgentSpaces",
    "aidevops:UpdateAgentSpace"
   ],
   "Resource": "*"
  },
  {
   "Sid": "AIDevOpsServiceAccess",
   "Effect": "Allow",
   "Action": [
    "aidevops:DeregisterService",
    "aidevops:GetService",
    "aidevops:ListServices",
    "aidevops:RegisterService",
    "aidevops:SearchServiceAccessibleResource"
   ],
   "Resource": "*"
  },
  {
   "Sid": "AIDevOpsAssociationAccess",
   "Effect": "Allow",
   "Action": [
    "aidevops:AssociateService",
    "aidevops:DisassociateService",
    "aidevops:GetAssociation",
    "aidevops:ListAssociations",
    "aidevops:UpdateAssociation",
    "aidevops:ValidateAwsAssociations"
   ],
   "Resource": "*"
  },
  {
   "Sid": "AIDevOpsWebhookAccess",
   "Effect": "Allow",
   "Action": [
    "aidevops:ListWebhooks"
   ],
   "Resource": "*"
  },
  {
   "Sid": "AIDevOpsOperatorAppAccess",
   "Effect": "Allow",
```

```
    "Action": [
     "aidevops:DisableOperatorApp",
     "aidevops:EnableOperatorApp",
     "aidevops:GetOperatorApp",
     "aidevops:UpdateOperatorAppIdpConfig"
    ],
    "Resource": "*"
   },
   {
    "Sid": "AIDevOpsKnowledgeAccess",
    "Effect": "Allow",
    "Action": [
     "aidevops:CreateKnowledgeItem",
     "aidevops:DeleteKnowledgeItem",
     "aidevops:GetKnowledgeItem",
     "aidevops:ListKnowledgeItems",
     "aidevops:ListKnowledgeItemVersions",
     "aidevops:UpdateKnowledgeItem"
    ],
    "Resource": "*"
   },
   {
    "Sid": "AIDevOpsBacklogAccess",
    "Effect": "Allow",
    "Action": [
     "aidevops:CreateBacklogTask",
     "aidevops:GetBacklogTask",
     "aidevops:ListBacklogTasks",
     "aidevops:ListGoals",
     "aidevops:UpdateBacklogTask",
     "aidevops:UpdateGoal"
    ],
    "Resource": "*"
   },
   {
    "Sid": "AIDevOpsRecommendationAccess",
    "Effect": "Allow",
    "Action": [
     "aidevops:GetRecommendation",
     "aidevops:ListRecommendations",
     "aidevops:UpdateRecommendation"
    ],
    "Resource": "*"
   },
```

```
{
 "Sid": "AIDevOpsAgentChatAccess",
 "Effect": "Allow",
 "Action": [
  "aidevops:CreateChat",
  "aidevops:ListChats",
  "aidevops:ListPendingMessages",
  "aidevops:SendMessage"
 ],
 "Resource": "*"
},
{
 "Sid": "AIDevOpsJournalAccess",
 "Effect": "Allow",
 "Action": [
  "aidevops:ListExecutions",
  "aidevops:ListJournalRecords"
 ],
 "Resource": "*"
},
{
 "Sid": "AIDevOpsTopologyAccess",
 "Effect": "Allow",
 "Action": [
  "aidevops:DiscoverTopology"
 ],
 "Resource": "*"
},
{
 "Sid": "AIDevOpsSupportAccess",
 "Effect": "Allow",
 "Action": [
  "aidevops:DescribeSupportLevel",
  "aidevops:EndChatForCase",
  "aidevops:InitiateChatForCase"
 ],
 "Resource": "*"
},
{
 "Sid": "AIDevOpsUsageAccess",
 "Effect": "Allow",
 "Action": [
  "aidevops:GetAccountUsage"
 ],
```

```
    "Resource": "*"
   },
   {
    "Sid": "AIDevOpsTaggingAccess",
    "Effect": "Allow",
    "Action": [
     "aidevops:ListTagsForResource",
     "aidevops:TagResource",
     "aidevops:UntagResource"
    ],
    "Resource": "*"
   },
   {
    "Sid": "AIDevOpsVendedLogs",
    "Effect": "Allow",
    "Action": [
     "aidevops:AllowVendedLogDeliveryForResource"
    ],
    "Resource": "*"
   }
  ]
}
```

AIDevOpsOperatorAppAccessPolicy

Permet d'utiliser l'application Web de l' AWS DevOps opérateur pour un agent Space.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "AllowOperatorAgentSpaceActions",
   "Effect": "Allow",
   "Action": [
    "aidevops:GetAgentSpace",
    "aidevops:GetAssociation",
    "aidevops:ListAssociations",
    "aidevops:CreateBacklogTask",
    "aidevops:GetBacklogTask",
    "aidevops:UpdateBacklogTask",
    "aidevops:ListBacklogTasks",
    "aidevops:ListJournalRecords",
    "aidevops:DiscoverTopology",
```

```
    "aidevops:ListGoals",
    "aidevops:ListRecommendations",
    "aidevops:ListExecutions",
    "aidevops:GetRecommendation",
    "aidevops:UpdateRecommendation",
    "aidevops:CreateKnowledgeItem",
    "aidevops:ListKnowledgeItems",
    "aidevops:ListKnowledgeItemVersions",
    "aidevops:GetKnowledgeItem",
    "aidevops:UpdateKnowledgeItem",
    "aidevops:DeleteKnowledgeItem",
    "aidevops:ListPendingMessages",
    "aidevops:InitiateChatForCase",
    "aidevops:EndChatForCase",
    "aidevops:DescribeSupportLevel",
    "aidevops:ListChats",
    "aidevops:CreateChat",
    "aidevops:SendMessage"
   ],
   "Resource": "arn:aws:aidevops:*:*:agentspace/${aws:PrincipalTag/AgentSpaceId}",
   "Condition": {
    "StringEquals": {
     "aws:ResourceAccount": "${aws:PrincipalAccount}"
    }
   }
  },
  {
   "Sid": "AllowOperatorAccountActions",
   "Effect": "Allow",
   "Action": [
    "aidevops:GetAccountUsage"
   ],
   "Resource": "*",
   "Condition": {
    "StringEquals": {
     "aws:ResourceAccount": "${aws:PrincipalAccount}"
    }
   }
  },
  {
   "Sid": "AllowSupportOperatorActions",
   "Effect": "Allow",
   "Action": [
    "support:DescribeCases",
```

```
    "support:InitiateChatForCase",
    "support:DescribeSupportLevel"
   ],
   "Resource": "*",
   "Condition": {
    "StringEquals": {
     "aws:ResourceAccount": "${aws:PrincipalAccount}"
    }
   }
  }
 ]
}
```

AIDevOpsAgentAccessPolicy

Fournit les autorisations requises par l' AWS DevOps agent pour mener des enquêtes et effectuer des analyses sur les AWS ressources des clients.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AIOPSServiceAccess",
            "Effect": "Allow",
            "Action": [
                "access-analyzer:GetAnalyzer",
                "access-analyzer:List*",
                "acm-pca:Describe*",
                "acm-pca:GetCertificate",
                "acm-pca:GetCertificateAuthorityCertificate",
                "acm-pca:GetCertificateAuthorityCsr",
                "acm-pca:List*",
                "acm:DescribeCertificate",
                "acm:GetAccountConfiguration",
                "aidevops:GetKnowledgeItem",
                "aidevops:ListKnowledgeItems",
                "airflow:List*",
                "amplify:GetApp",
                "amplify:GetBranch",
                "amplify:GetDomainAssociation",
                "amplify:List*",
                "aoss:BatchGetCollection",
                "aoss:BatchGetLifecyclePolicy",
```

```
                "aoss:BatchGetVpcEndpoint",
                "aoss:GetAccessPolicy",
                "aoss:GetSecurityConfig",
                "aoss:GetSecurityPolicy",
                "aoss:List*",
                "appconfig:GetApplication",
                "appconfig:GetConfigurationProfile",
                "appconfig:GetEnvironment",
                "appconfig:GetHostedConfigurationVersion",
                "appconfig:List*",
                "appflow:Describe*",
                "appflow:List*",
                "application-autoscaling:Describe*",
                "application-signals:BatchGetServiceLevelObjectiveBudgetReport",
                "application-signals:GetService",
                "application-signals:GetServiceLevelObjective",
                "application-signals:List*",
                "applicationinsights:Describe*",
                "applicationinsights:List*",
                "apprunner:Describe*",
                "apprunner:List*",
                "appstream:Describe*",
                "appstream:List*",
                "appsync:GetApiAssociation",
                "appsync:GetDataSource",
                "appsync:GetDomainName",
                "appsync:GetFunction",
                "appsync:GetGraphqlApi",
                "appsync:GetGraphqlApiEnvironmentVariables",
                "appsync:GetIntrospectionSchema",
                "appsync:GetResolver",
                "appsync:GetSourceApiAssociation",
                "appsync:List*",
                "aps:Describe*",
                "aps:List*",
                "arc-zonal-shift:GetManagedResource",
                "arc-zonal-shift:List*",
                "athena:GetCapacityAssignmentConfiguration",
                "athena:GetCapacityReservation",
                "athena:GetDataCatalog",
                "athena:GetNamedQuery",
                "athena:GetPreparedStatement",
                "athena:GetWorkGroup",
                "athena:List*",
```

```
                "auditmanager:GetAssessment",
                "auditmanager:List*",
                "autoscaling:Describe*",
                "backup-gateway:GetHypervisor",
                "backup-gateway:List*",
                "backup:Describe*",
                "backup:GetBackupPlan",
                "backup:GetBackupSelection",
                "backup:GetBackupVaultAccessPolicy",
                "backup:GetBackupVaultNotifications",
                "backup:GetRestoreTestingPlan",
                "backup:GetRestoreTestingSelection",
                "backup:List*",
                "batch:DescribeComputeEnvironments",
                "batch:DescribeJobQueues",
                "batch:DescribeSchedulingPolicies",
                "batch:List*",
                "bedrock:GetAgent",
                "bedrock:GetAgentActionGroup",
                "bedrock:GetAgentAlias",
                "bedrock:GetAgentKnowledgeBase",
                "bedrock:GetDataSource",
                "bedrock:GetGuardrail",
                "bedrock:GetKnowledgeBase",
                "bedrock:List*",
                "budgets:Describe*",
                "budgets:List*",
                "ce:Describe*",
                "ce:GetAnomalyMonitors",
                "ce:GetAnomalySubscriptions",
                "ce:List*",
                "chatbot:Describe*",
                "chatbot:GetMicrosoftTeamsChannelConfiguration",
                "chatbot:List*",
                "cleanrooms-ml:GetTrainingDataset",
                "cleanrooms-ml:List*",
                "cleanrooms:GetAnalysisTemplate",
                "cleanrooms:GetCollaboration",
                "cleanrooms:GetConfiguredTable",
                "cleanrooms:GetConfiguredTableAnalysisRule",
                "cleanrooms:GetConfiguredTableAssociation",
                "cleanrooms:GetMembership",
                "cleanrooms:List*",
                "cloudformation:Describe*",
```

```
                "cloudformation:GetResource",
                "cloudformation:GetStackPolicy",
                "cloudformation:GetTemplate",
                "cloudformation:List*",
                "cloudfront:Describe*",
                "cloudfront:GetCachePolicy",
                "cloudfront:GetCloudFrontOriginAccessIdentity",
                "cloudfront:GetContinuousDeploymentPolicy",
                "cloudfront:GetDistribution",
                "cloudfront:GetDistributionConfig",
                "cloudfront:GetFunction",
                "cloudfront:GetKeyGroup",
                "cloudfront:GetMonitoringSubscription",
                "cloudfront:GetOriginAccessControl",
                "cloudfront:GetOriginRequestPolicy",
                "cloudfront:GetPublicKey",
                "cloudfront:GetRealtimeLogConfig",
                "cloudfront:GetResponseHeadersPolicy",
                "cloudfront:List*",
                "cloudtrail:Describe*",
                "cloudtrail:GetChannel",
                "cloudtrail:GetEventConfiguration",
                "cloudtrail:GetEventDataStore",
                "cloudtrail:GetEventSelectors",
                "cloudtrail:GetInsightSelectors",
                "cloudtrail:GetQueryResults",
                "cloudtrail:GetResourcePolicy",
                "cloudtrail:GetTrail",
                "cloudtrail:GetTrailStatus",
                "cloudtrail:List*",
                "cloudtrail:LookupEvents",
                "cloudtrail:StartQuery",
                "cloudwatch:Describe*",
                "cloudwatch:GenerateQuery",
                "cloudwatch:GetDashboard",
                "cloudwatch:GetInsightRuleReport",
                "cloudwatch:GetMetricData",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:GetMetricStream",
                "cloudwatch:GetService",
                "cloudwatch:GetServiceLevelObjective",
                "cloudwatch:List*",
                "codeartifact:Describe*",
                "codeartifact:GetDomainPermissionsPolicy",
```

```
                "codeartifact:GetRepositoryPermissionsPolicy",
                "codeartifact:List*",
                "codebuild:BatchGetFleets",
                "codebuild:List*",
                "codecommit:GetRepository",
                "codecommit:GetRepositoryTriggers",
                "codedeploy:BatchGetDeployments",
                "codedeploy:BatchGetDeploymentTargets",
                "codedeploy:GetApplication",
                "codedeploy:GetDeploymentConfig",
                "codedeploy:GetDeploymentTarget",
                "codedeploy:List*",
                "codeguru-profiler:Describe*",
                "codeguru-profiler:GetNotificationConfiguration",
                "codeguru-profiler:GetPolicy",
                "codeguru-profiler:List*",
                "codeguru-reviewer:Describe*",
                "codeguru-reviewer:List*",
                "codepipeline:GetPipeline",
                "codepipeline:GetPipelineState",
                "codepipeline:List*",
                "codestar-connections:GetConnection",
                "codestar-connections:GetRepositoryLink",
                "codestar-connections:GetSyncConfiguration",
                "codestar-connections:List*",
                "codestar-notifications:Describe*",
                "codestar-notifications:List*",
                "cognito-identity:DescribeIdentityPool",
                "cognito-identity:GetIdentityPoolRoles",
                "cognito-identity:ListIdentityPools",
                "cognito-identity:ListTagsForResource",
                "cognito-idp:AdminListGroupsForUser",
                "cognito-idp:DescribeIdentityProvider",
                "cognito-idp:DescribeResourceServer",
                "cognito-idp:DescribeRiskConfiguration",
                "cognito-idp:DescribeUserImportJob",
                "cognito-idp:DescribeUserPool",
                "cognito-idp:DescribeUserPoolDomain",
                "cognito-idp:GetGroup",
                "cognito-idp:GetLogDeliveryConfiguration",
                "cognito-idp:GetUICustomization",
                "cognito-idp:GetUserPoolMfaConfig",
                "cognito-idp:GetWebACLForResource",
                "cognito-idp:ListGroups",
```

```
                "cognito-idp:ListIdentityProviders",
                "cognito-idp:ListResourceServers",
                "cognito-idp:ListUserPoolClients",
                "cognito-idp:ListUserPools",
                "cognito-idp:ListTagsForResource",
                "comprehend:Describe*",
                "comprehend:List*",
                "config:Describe*",
                "config:GetStoredQuery",
                "config:List*",
                "connect:Describe*",
                "connect:GetTaskTemplate",
                "connect:List*",
                "databrew:Describe*",
                "databrew:List*",
                "datapipeline:Describe*",
                "datapipeline:GetPipelineDefinition",
                "datapipeline:List*",
                "datasync:Describe*",
                "datasync:List*",
                "deadline:GetFarm",
                "deadline:GetFleet",
                "deadline:GetLicenseEndpoint",
                "deadline:GetMonitor",
                "deadline:GetQueue",
                "deadline:GetQueueEnvironment",
                "deadline:GetQueueFleetAssociation",
                "deadline:GetStorageProfile",
                "deadline:List*",
                "detective:GetMembers",
                "detective:List*",
                "devicefarm:GetDevicePool",
                "devicefarm:GetInstanceProfile",
                "devicefarm:GetNetworkProfile",
                "devicefarm:GetProject",
                "devicefarm:GetTestGridProject",
                "devicefarm:GetVPCEConfiguration",
                "devicefarm:List*",
                "devops-guru:Describe*",
                "devops-guru:GetResourceCollection",
                "devops-guru:List*",
                "dms:Describe*",
                "dms:List*",
                "ds:Describe*",
```

```
                "dynamodb:Describe*",
                "dynamodb:GetResourcePolicy",
                "dynamodb:List*",
                "ec2:Describe*",
                "ec2:GetAssociatedEnclaveCertificateIamRoles",
                "ec2:GetIpamPoolAllocations",
                "ec2:GetIpamPoolCidrs",
                "ec2:GetManagedPrefixListEntries",
                "ec2:GetNetworkInsightsAccessScopeContent",
                "ec2:GetSnapshotBlockPublicAccessState",
                "ec2:GetTransitGatewayMulticastDomainAssociations",
                "ec2:GetTransitGatewayRouteTableAssociations",
                "ec2:GetTransitGatewayRouteTablePropagations",
                "ec2:GetVerifiedAccessEndpointPolicy",
                "ec2:GetVerifiedAccessGroupPolicy",
                "ec2:GetVerifiedAccessInstanceWebAcl",
                "ec2:SearchLocalGatewayRoutes",
                "ec2:SearchTransitGatewayRoutes",
                "ecr:Describe*",
                "ecr:GetLifecyclePolicy",
                "ecr:GetRegistryPolicy",
                "ecr:GetRepositoryPolicy",
                "ecr:List*",
                "ecs:Describe*",
                "ecs:List*",
                "eks:AccessKubernetesApi",
                "eks:Describe*",
                "eks:List*",
                "elasticache:Describe*",
                "elasticache:List*",
                "elasticbeanstalk:Describe*",
                "elasticbeanstalk:List*",
                "elasticfilesystem:Describe*",
                "elasticloadbalancing:GetResourcePolicy",
                "elasticloadbalancing:GetTrustStoreCaCertificatesBundle",
                "elasticloadbalancing:GetTrustStoreRevocationContent",
                "elasticloadbalancing:Describe*",
                "elasticmapreduce:Describe*",
                "elasticmapreduce:List*",
                "emr-containers:Describe*",
                "emr-containers:List*",
                "emr-serverless:GetApplication",
                "emr-serverless:List*",
                "es:Describe*",
```

```
                "es:List*",
                "events:Describe*",
                "events:List*",
                "evidently:GetExperiment",
                "evidently:GetFeature",
                "evidently:GetLaunch",
                "evidently:GetProject",
                "evidently:GetSegment",
                "evidently:List*",
                "firehose:Describe*",
                "firehose:List*",
                "fis:GetExperimentTemplate",
                "fis:GetTargetAccountConfiguration",
                "fis:List*",
                "fms:GetNotificationChannel",
                "fms:GetPolicy",
                "fms:List*",
                "forecast:Describe*",
                "forecast:List*",
                "frauddetector:BatchGetVariable",
                "frauddetector:Describe*",
                "frauddetector:GetDetectors",
                "frauddetector:GetDetectorVersion",
                "frauddetector:GetEntityTypes",
                "frauddetector:GetEventTypes",
                "frauddetector:GetExternalModels",
                "frauddetector:GetLabels",
                "frauddetector:GetListElements",
                "frauddetector:GetListsMetadata",
                "frauddetector:GetModelVersion",
                "frauddetector:GetOutcomes",
                "frauddetector:GetRules",
                "frauddetector:GetVariables",
                "frauddetector:List*",
                "fsx:Describe*",
                "gamelift:Describe*",
                "gamelift:List*",
                "globalaccelerator:Describe*",
                "globalaccelerator:List*",
                "glue:GetDatabase",
                "glue:GetDatabases",
                "glue:GetJob",
                "glue:GetRegistry",
                "glue:GetSchema",
```

```
                "glue:GetSchemaVersion",
                "glue:GetTable",
                "glue:GetTags",
                "glue:GetTrigger",
                "glue:List*",
                "glue:querySchemaVersionMetadata",
                "grafana:Describe*",
                "grafana:List*",
                "greengrass:Describe*",
                "greengrass:GetDeployment",
                "greengrass:List*",
                "groundstation:GetConfig",
                "groundstation:GetDataflowEndpointGroup",
                "groundstation:GetMissionProfile",
                "groundstation:List*",
                "guardduty:GetDetector",
                "guardduty:GetFilter",
                "guardduty:GetIPSet",
                "guardduty:GetMalwareProtectionPlan",
                "guardduty:GetMasterAccount",
                "guardduty:GetMembers",
                "guardduty:GetThreatIntelSet",
                "guardduty:List*",
                "health:DescribeEvents",
                "health:DescribeEventDetails",
                "healthlake:Describe*",
                "healthlake:List*",
                "iam:GetGroup",
                "iam:GetGroupPolicy",
                "iam:GetInstanceProfile",
                "iam:GetLoginProfile",
                "iam:GetOpenIDConnectProvider",
                "iam:GetPolicy",
                "iam:GetPolicyVersion",
                "iam:GetRole",
                "iam:GetRolePolicy",
                "iam:GetSAMLProvider",
                "iam:GetServerCertificate",
                "iam:GetServiceLinkedRoleDeletionStatus",
                "iam:GetUser",
                "iam:GetUserPolicy",
                "iam:ListAttachedRolePolicies",
                "iam:ListOpenIDConnectProviders",
                "iam:ListRolePolicies",
```

```
                "iam:ListRoles",
                "iam:ListServerCertificates",
                "iam:ListVirtualMFADevices",
                "identitystore:DescribeGroup",
                "identitystore:DescribeGroupMembership",
                "identitystore:ListGroupMemberships",
                "identitystore:ListGroups",
                "imagebuilder:GetComponent",
                "imagebuilder:GetContainerRecipe",
                "imagebuilder:GetDistributionConfiguration",
                "imagebuilder:GetImage",
                "imagebuilder:GetImagePipeline",
                "imagebuilder:GetImageRecipe",
                "imagebuilder:GetInfrastructureConfiguration",
                "imagebuilder:GetLifecyclePolicy",
                "imagebuilder:GetWorkflow",
                "imagebuilder:List*",
                "inspector2:List*",
                "inspector:Describe*",
                "inspector:List*",
                "internetmonitor:GetMonitor",
                "internetmonitor:List*",
                "iot:Describe*",
                "iot:GetPackage",
                "iot:GetPackageVersion",
                "iot:GetPolicy",
                "iot:GetThingShadow",
                "iot:GetTopicRule",
                "iot:GetTopicRuleDestination",
                "iot:GetV2LoggingOptions",
                "iot:List*",
                "iotanalytics:Describe*",
                "iotanalytics:List*",
                "iotevents:Describe*",
                "iotevents:List*",
                "iotsitewise:Describe*",
                "iotsitewise:List*",
                "iotwireless:GetDestination",
                "iotwireless:GetDeviceProfile",
                "iotwireless:GetFuotaTask",
                "iotwireless:GetMulticastGroup",
                "iotwireless:GetNetworkAnalyzerConfiguration",
                "iotwireless:GetServiceProfile",
                "iotwireless:GetWirelessDevice",
```

```
                "iotwireless:GetWirelessGateway",
                "iotwireless:GetWirelessGatewayTaskDefinition",
                "iotwireless:List*",
                "ivs:GetChannel",
                "ivs:GetEncoderConfiguration",
                "ivs:GetPlaybackRestrictionPolicy",
                "ivs:GetRecordingConfiguration",
                "ivs:GetStage",
                "ivs:List*",
                "ivschat:GetLoggingConfiguration",
                "ivschat:GetRoom",
                "ivschat:List*",
                "kafka:Describe*",
                "kafka:GetClusterPolicy",
                "kafka:List*",
                "kafkaconnect:Describe*",
                "kafkaconnect:List*",
                "kendra:Describe*",
                "kendra:List*",
                "kinesis:Describe*",
                "kinesis:GetResourcePolicy",
                "kinesis:List*",
                "kinesisanalytics:Describe*",
                "kinesisanalytics:List*",
                "kinesisvideo:Describe*",
                "kms:DescribeKey",
                "kms:ListResourceTags",
                "kms:ListKeys",
                "kms:GetKeyPolicy",
                "kms:GetKeyRotationStatus",
                "kms:ListAliases",
                "kms:ListKeyRotations",
                "lakeformation:Describe*",
                "lakeformation:GetLFTag",
                "lakeformation:GetResourceLFTags",
                "lakeformation:List*",
                "lambda:GetAlias",
                "lambda:GetCodeSigningConfig",
                "lambda:GetEventSourceMapping",
                "lambda:GetFunctionCodeSigningConfig",
                "lambda:GetFunctionConfiguration",
                "lambda:GetFunctionEventInvokeConfig",
                "lambda:GetFunctionRecursionConfig",
                "lambda:GetFunctionUrlConfig",
```

```
                "lambda:GetLayerVersion",
                "lambda:GetLayerVersionPolicy",
                "lambda:GetPolicy",
                "lambda:GetProvisionedConcurrencyConfig",
                "lambda:GetRuntimeManagementConfig",
                "lambda:List*",
                "launchwizard:GetDeployment",
                "launchwizard:List*",
                "license-manager:GetLicense",
                "license-manager:List*",
                "lightsail:GetAlarms",
                "lightsail:GetBuckets",
                "lightsail:GetCertificates",
                "lightsail:GetContainerServices",
                "lightsail:GetDisk",
                "lightsail:GetDisks",
                "lightsail:GetInstance",
                "lightsail:GetInstances",
                "lightsail:GetLoadBalancer",
                "lightsail:GetLoadBalancers",
                "lightsail:GetLoadBalancerTlsCertificates",
                "lightsail:GetStaticIp",
                "lightsail:GetStaticIps",
                "logs:Describe*",
                "logs:FilterLogEvents",
                "logs:GetDataProtectionPolicy",
                "logs:GetDelivery",
                "logs:GetDeliveryDestination",
                "logs:GetDeliveryDestinationPolicy",
                "logs:GetDeliverySource",
                "logs:GetLogAnomalyDetector",
                "logs:GetLogDelivery",
                "logs:GetLogGroupFields",
                "logs:GetQueryResults",
                "logs:List*",
                "logs:StartQuery",
                "logs:StopLiveTail",
                "logs:StopQuery",
                "logs:TestMetricFilter",
                "m2:GetApplication",
                "m2:GetEnvironment",
                "m2:List*",
                "macie2:GetAllowList",
                "macie2:GetCustomDataIdentifier",
```

```
                "macie2:GetFindingsFilter",
                "macie2:GetMacieSession",
                "macie2:List*",
                "mediaconnect:Describe*",
                "mediaconnect:List*",
                "medialive:Describe*",
                "medialive:GetCloudWatchAlarmTemplate",
                "medialive:GetCloudWatchAlarmTemplateGroup",
                "medialive:GetEventBridgeRuleTemplate",
                "medialive:GetEventBridgeRuleTemplateGroup",
                "medialive:GetSignalMap",
                "medialive:List*",
                "mediapackage-vod:Describe*",
                "mediapackage-vod:List*",
                "mediapackage:Describe*",
                "mediapackage:List*",
                "mediapackagev2:GetChannel",
                "mediapackagev2:GetChannelGroup",
                "mediapackagev2:GetChannelPolicy",
                "mediapackagev2:GetOriginEndpoint",
                "mediapackagev2:GetOriginEndpointPolicy",
                "mediapackagev2:List*",
                "memorydb:Describe*",
                "memorydb:List*",
                "mobiletargeting:GetInAppTemplate",
                "mobiletargeting:List*",
                "mq:Describe*",
                "mq:List*",
                "network-firewall:Describe*",
                "network-firewall:List*",
                "networkmanager:Describe*",
                "networkmanager:GetConnectAttachment",
                "networkmanager:GetConnectPeer",
                "networkmanager:GetCoreNetwork",
                "networkmanager:GetCoreNetworkPolicy",
                "networkmanager:GetCustomerGatewayAssociations",
                "networkmanager:GetDevices",
                "networkmanager:GetLinkAssociations",
                "networkmanager:GetLinks",
                "networkmanager:GetSites",
                "networkmanager:GetSiteToSiteVpnAttachment",
                "networkmanager:GetTransitGatewayPeering",
                "networkmanager:GetTransitGatewayRegistrations",
                "networkmanager:GetTransitGatewayRouteTableAttachment",
```

```
                "networkmanager:GetVpcAttachment",
                "networkmanager:List*",
                "oam:GetLink",
                "oam:GetSink",
                "oam:GetSinkPolicy",
                "oam:List*",
                "omics:GetAnnotationStore",
                "omics:GetReferenceStore",
                "omics:GetRunGroup",
                "omics:GetSequenceStore",
                "omics:GetVariantStore",
                "omics:GetWorkflow",
                "omics:List*",
                "organizations:Describe*",
                "organizations:List*",
                "osis:GetPipeline",
                "osis:List*",
                "payment-cryptography:GetAlias",
                "payment-cryptography:GetKey",
                "payment-cryptography:List*",
                "pca-connector-ad:GetConnector",
                "pca-connector-ad:GetDirectoryRegistration",
                "pca-connector-ad:GetServicePrincipalName",
                "pca-connector-ad:GetTemplate",
                "pca-connector-ad:GetTemplateGroupAccessControlEntry",
                "pca-connector-ad:List*",
                "pca-connector-scep:GetChallengeMetadata",
                "pca-connector-scep:GetConnector",
                "pca-connector-scep:List*",
                "personalize:Describe*",
                "personalize:List*",
                "pi:DescribeDimensionKeys",
                "pi:GetResourceMetadata",
                "pi:GetResourceMetrics",
                "pi:ListAvailableResourceDimensions",
                "pi:ListAvailableResourceMetrics",
                "pipes:Describe*",
                "pipes:List*",
                "proton:GetEnvironmentTemplate",
                "proton:GetServiceTemplate",
                "proton:List*",
                "qbusiness:GetApplication",
                "qbusiness:GetDataSource",
                "qbusiness:GetIndex",
```

```
                "qbusiness:GetPlugin",
                "qbusiness:GetRetriever",
                "qbusiness:GetWebExperience",
                "qbusiness:List*",
                "ram:GetPermission",
                "ram:GetResourceShares",
                "ram:List*",
                "rds:Describe*",
                "rds:List*",
                "redshift-serverless:GetNamespace",
                "redshift-serverless:GetWorkgroup",
                "redshift-serverless:List*",
                "redshift:Describe*",
                "refactor-spaces:GetApplication",
                "refactor-spaces:GetEnvironment",
                "refactor-spaces:GetRoute",
                "refactor-spaces:List*",
                "rekognition:Describe*",
                "rekognition:List*",
                "resiliencehub:Describe*",
                "resiliencehub:List*",
                "resource-explorer-2:GetDefaultView",
                "resource-explorer-2:GetIndex",
                "resource-explorer-2:GetView",
                "resource-explorer-2:List*",
                "resource-explorer-2:Search",
                "resource-groups:GetGroup",
                "resource-groups:GetGroupConfiguration",
                "resource-groups:GetGroupQuery",
                "resource-groups:GetTags",
                "resource-groups:List*",
                "route53-recovery-control-config:Describe*",
                "route53-recovery-control-config:List*",
                "route53-recovery-readiness:GetCell",
                "route53-recovery-readiness:GetReadinessCheck",
                "route53-recovery-readiness:GetRecoveryGroup",
                "route53-recovery-readiness:GetResourceSet",
                "route53-recovery-readiness:List*",
                "route53:GetDNSSEC",
                "route53:GetHealthCheck",
                "route53:GetHealthCheckStatus",
                "route53:GetHostedZone",
                "route53:List*",
                "route53profiles:GetProfile",
```

```
                "route53profiles:GetProfileAssociation",
                "route53profiles:GetProfileResourceAssociation",
                "route53profiles:List*",
                "route53resolver:GetFirewallDomainList",
                "route53resolver:GetFirewallRuleGroup",
                "route53resolver:GetFirewallRuleGroupAssociation",
                "route53resolver:GetOutpostResolver",
                "route53resolver:GetResolverConfig",
                "route53resolver:GetResolverQueryLogConfig",
                "route53resolver:GetResolverQueryLogConfigAssociation",
                "route53resolver:GetResolverRule",
                "route53resolver:GetResolverRuleAssociation",
                "route53resolver:List*",
                "rum:GetAppMonitor",
                "rum:List*",
                "s3-outposts:ListEndpoints",
                "s3-outposts:ListOutpostsWithS3",
                "s3:GetAccessGrant",
                "s3:GetAccessGrantsInstance",
                "s3:GetAccessGrantsLocation",
                "s3:GetAccessPoint",
                "s3:GetAccessPointConfigurationForObjectLambda",
                "s3:GetAccessPointForObjectLambda",
                "s3:GetAccessPointPolicy",
                "s3:GetAccessPointPolicyForObjectLambda",
                "s3:GetAccessPointPolicyStatusForObjectLambda",
                "s3:GetBucketAbac",
                "s3:GetBucketAcl",
                "s3:GetBucketCORS",
                "s3:GetBucketLocation",
                "s3:GetBucketLogging",
                "s3:GetBucketMetadataTableConfiguration",
                "s3:GetBucketNotification",
                "s3:GetBucketObjectLockConfiguration",
                "s3:GetBucketOwnershipControls",
                "s3:GetBucketPolicy",
                "s3:GetBucketPublicAccessBlock",
                "s3:GetBucketTagging",
                "s3:GetBucketVersioning",
                "s3:GetEncryptionConfiguration",
                "s3:GetLifecycleConfiguration",
                "s3:GetMultiRegionAccessPoint",
                "s3:GetMultiRegionAccessPointPolicy",
                "s3:GetMultiRegionAccessPointPolicyStatus",
```

```
                "s3:GetReplicationConfiguration",
                "s3:GetStorageLensConfiguration",
                "s3:GetStorageLensConfigurationTagging",
                "s3:GetStorageLensGroup",
                "s3:ListAllMyBuckets",
                "sagemaker:Describe*",
                "sagemaker:List*",
                "scheduler:GetSchedule",
                "scheduler:GetScheduleGroup",
                "scheduler:List*",
                "schemas:Describe*",
                "schemas:GetResourcePolicy",
                "schemas:List*",
                "secretsmanager:Describe*",
                "secretsmanager:GetResourcePolicy",
                "secretsmanager:List*",
                "securityhub:BatchGetAutomationRules",
                "securityhub:BatchGetSecurityControls",
                "securityhub:Describe*",
                "securityhub:GetConfigurationPolicy",
                "securityhub:GetConfigurationPolicyAssociation",
                "securityhub:GetEnabledStandards",
                "securityhub:GetFindingAggregator",
                "securityhub:GetInsights",
                "securityhub:List*",
                "securitylake:GetSubscriber",
                "securitylake:List*",
                "servicecatalog:Describe*",
                "servicecatalog:GetApplication",
                "servicecatalog:GetAttributeGroup",
                "servicecatalog:List*",
                "servicequotas:GetServiceQuota",
                "ses:Describe*",
                "ses:GetAccount",
                "ses:GetAddonInstance",
                "ses:GetAddonSubscription",
                "ses:GetArchive",
                "ses:GetConfigurationSet",
                "ses:GetConfigurationSetEventDestinations",
                "ses:GetContactList",
                "ses:GetDedicatedIpPool",
                "ses:GetDedicatedIps",
                "ses:GetEmailIdentity",
                "ses:GetEmailTemplate",
```

```
                "ses:GetIngressPoint",
                "ses:GetRelay",
                "ses:GetRuleSet",
                "ses:GetTemplate",
                "ses:GetTrafficPolicy",
                "ses:List*",
                "shield:Describe*",
                "shield:List*",
                "signer:GetSigningProfile",
                "signer:List*",
                "sns:GetDataProtectionPolicy",
                "sns:GetSubscriptionAttributes",
                "sns:GetTopicAttributes",
                "sns:List*",
                "sqs:GetQueueAttributes",
                "sqs:GetQueueUrl",
                "sqs:List*",
                "ssm-contacts:GetContact",
                "ssm-contacts:GetContactChannel",
                "ssm-contacts:List*",
                "ssm-incidents:GetReplicationSet",
                "ssm-incidents:GetResponsePlan",
                "ssm-incidents:List*",
                "ssm-sap:GetApplication",
                "ssm-sap:List*",
                "ssm:Describe*",
                "ssm:GetDefaultPatchBaseline",
                "ssm:GetDocument",
                "ssm:GetParameters",
                "ssm:GetPatchBaseline",
                "ssm:GetResourcePolicies",
                "ssm:List*",
                "sso:GetInlinePolicyForPermissionSet",
                "sso:GetManagedApplicationInstance",
                "sso:GetPermissionsBoundaryForPermissionSet",
                "sso:GetSharedSsoConfiguration",
                "sso:ListAccountAssignments",
                "sso:ListApplicationAssignments",
                "sso:ListApplications",
                "sso:ListCustomerManagedPolicyReferencesInPermissionSet",
                "sso:ListInstances",
                "sso:ListManagedPoliciesInPermissionSet",
                "sso:ListTagsForResource",
                "states:GetExecutionHistory",
```

```
                "states:Describe*",
                "states:List*",
                "support:CreateCase",
                "support:DescribeCases",
                "synthetics:Describe*",
                "synthetics:GetCanary",
                "synthetics:GetCanaryRuns",
                "synthetics:GetGroup",
                "synthetics:List*",
                "tag:GetResources",
                "timestream:Describe*",
                "timestream:List*",
                "transfer:Describe*",
                "transfer:List*",
                "verifiedpermissions:GetIdentitySource",
                "verifiedpermissions:GetPolicy",
                "verifiedpermissions:GetPolicyStore",
                "verifiedpermissions:GetPolicyTemplate",
                "verifiedpermissions:GetSchema",
                "verifiedpermissions:List*",
                "vpc-lattice:GetAccessLogSubscription",
                "vpc-lattice:GetAuthPolicy",
                "vpc-lattice:GetListener",
                "vpc-lattice:GetResourcePolicy",
                "vpc-lattice:GetRule",
                "vpc-lattice:GetService",
                "vpc-lattice:GetServiceNetwork",
                "vpc-lattice:GetServiceNetworkServiceAssociation",
                "vpc-lattice:GetServiceNetworkVpcAssociation",
                "vpc-lattice:GetTargetGroup",
                "vpc-lattice:List*",
                "wafv2:GetIPSet",
                "wafv2:GetLoggingConfiguration",
                "wafv2:GetRegexPatternSet",
                "wafv2:GetRuleGroup",
                "wafv2:GetWebACL",
                "wafv2:GetWebACLForResource",
                "wafv2:List*",
                "workspaces-web:GetBrowserSettings",
                "workspaces-web:GetIdentityProvider",
                "workspaces-web:GetNetworkSettings",
                "workspaces-web:GetPortal",
                "workspaces-web:GetPortalServiceProviderMetadata",
                "workspaces-web:GetTrustStore",
```

```
                "workspaces-web:GetUserAccessLoggingSettings",
                "workspaces-web:GetUserSettings",
                "workspaces-web:List*",
                "workspaces:Describe*",
                "xray:BatchGetTraces",
                "xray:GetGroup",
                "xray:GetGroups",
                "xray:GetSamplingRules",
                "xray:GetServiceGraph",
                "xray:GetTraceSummaries",
                "xray:List*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AIOPSAPIGatewayAccess",
            "Effect": "Allow",
            "Action": [
                "apigateway:GET"
            ],
            "Resource": [
                "arn:aws:apigateway:*::/restapis",
                "arn:aws:apigateway:*::/restapis/*",
                "arn:aws:apigateway:*::/restapis/*/deployments",
                "arn:aws:apigateway:*::/restapis/*/deployments/*",
                "arn:aws:apigateway:*::/restapis/*/resources/*/methods/*/integrations",
                "arn:aws:apigateway:*::/restapis/*/resources/*/methods/*/integrations/
*",
                "arn:aws:apigateway:*::/restapis/*/stages",
                "arn:aws:apigateway:*::/restapis/*/stages/*",
                "arn:aws:apigateway:*::/apis",
                "arn:aws:apigateway:*::/apis/*",
                "arn:aws:apigateway:*::/apis/*/deployments",
                "arn:aws:apigateway:*::/apis/*/deployments/*",
                "arn:aws:apigateway:*::/apis/*/integrations",
                "arn:aws:apigateway:*::/apis/*/integrations/*",
                "arn:aws:apigateway:*::/apis/*/stages",
                "arn:aws:apigateway:*::/apis/*/stages/*",
                "arn:aws:apigateway:*::/domainnames/*"
            ]
        }
    ]
}
```

Limiter l'accès des agents à un AWS compte

AWS DevOps L'agent utilise les rôles IAM pour découvrir et décrire les AWS ressources lors des enquêtes sur les incidents et des évaluations préventives. Vous pouvez contrôler le niveau d'accès de l'agent en configurant les politiques IAM associées à ces rôles. La topologie de l'application ne montre pas tout ce à quoi l'agent a accès. Les politiques IAM sont le seul moyen de réellement limiter les AWS services APIs et les ressources auxquels l'agent peut accéder.

Comprendre les rôles IAM pour l'agent AWS DevOps

AWS DevOps L'agent utilise les rôles IAM pour accéder aux ressources de deux types de comptes :

- Rôle du compte principal — Permet à l'agent d'accéder aux ressources du AWS compte sur lequel vous créez l'espace agent.
- Rôles de compte secondaires — Permet à l'agent d'accéder aux ressources des AWS comptes supplémentaires que vous connectez à l'espace agent.

Quel que soit le type de compte, vous pouvez restreindre les AWS services auxquels l'agent peut accéder, limiter l'accès à des ressources spécifiques au sein de ces services et contrôler les régions dans lesquelles l'agent peut opérer.

Choix des limites de vos ressources

Lorsque vous limitez l'accès aux ressources, vous devez inclure suffisamment d'autorisations pour que l'agent puisse enquêter avec succès sur les incidents liés aux applications. Cela inclut notamment les éléments suivants :

- Toutes les ressources pour les applications concernées que l'agent doit surveiller et étudier
- Toute l'infrastructure de support dont dépendent ces applications

L'infrastructure de soutien peut inclure :

- Composants réseau (sous-réseauxVPCs, équilibreurs de charge, passerelles d'API)
- Magasins de données (bases de données, caches, stockage d'objets)
- Ressources de calcul (instances EC2, fonctions Lambda, conteneurs)
- Services de surveillance et de journalisation (CloudWatch, CloudTrail)
- Ressources de gestion des identités et des accès nécessaires pour comprendre les autorisations

Si vous limitez trop étroitement l'accès, l'agent risque de ne pas être en mesure d'identifier les causes profondes liées au soutien de l'infrastructure en dehors des limites que vous avez définies.

Restreindre l'accès aux services

Vous pouvez limiter les AWS services auxquels l'agent peut accéder en modifiant les politiques IAM associées aux rôles de l'agent. Lorsque vous créez des politiques personnalisées, suivez les meilleures pratiques suivantes :

- Accordez uniquement des autorisations en lecture seule : l'agent doit lire les configurations des ressources, les métriques et les journaux pendant les enquêtes. Évitez d'accorder des autorisations permettant à l'agent de modifier ou de supprimer des ressources.
- Limiter les services nécessaires — N'incluez que les AWS services contenant des ressources pertinentes pour vos applications. Par exemple, si votre application n'utilise pas Amazon RDS, n'incluez pas les autorisations RDS dans la politique.
- Utilisez des actions spécifiques plutôt que des caractères génériques : au lieu d'accorder des `service:*` autorisations, spécifiez des actions individuelles telles que `cloudwatch:GetMetricData` ou`ec2:DescribeInstances`.

Exemple de politique limitée à des services spécifiques :

```
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": "*"
    }
```

```
  ]
}
```

Restreindre l'accès aux ressources

Pour limiter l'agent à des ressources spécifiques au sein d'un service, utilisez les autorisations au niveau des ressources dans vos politiques IAM. Cela vous permet de n'accorder l'accès qu'aux ressources qui correspondent à des modèles spécifiques.

À l'aide de modèles d'ARN de ressources :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lambda:GetFunction",
        "lambda:GetFunctionConfiguration"
      ],
      "Resource": "arn:aws:lambda:*:*:function:production-*"
    }
  ]
}
```

Cet exemple limite l'agent à accéder uniquement aux fonctions Lambda dont le nom commence par « production- ».

Utilisation de restrictions basées sur des balises :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeInstanceStatus"
      ],
      "Resource": "*",
      "Condition": {
```

```
        "StringEquals": {
          "aws:ResourceTag/Environment": "production"
        }
      }
    }
  ]
}
```

Cet exemple limite l'agent à accéder uniquement aux instances EC2 étiquetées avec`Environment=production`.

Restreindre l'accès régional

Pour limiter AWS les régions auxquelles l'agent peut accéder, utilisez la clé de `aws:RequestedRegion` condition dans vos politiques 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"
          ]
        }
      }
    }
  ]
}
```

Cet exemple limite l'agent à accéder aux ressources uniquement dans les régions us-east-1 et us-west-2.

Création de politiques IAM personnalisées

Lorsque vous créez un espace agent ou que vous ajoutez des comptes secondaires, vous avez la possibilité de créer un rôle IAM personnalisé à l'aide d'un modèle de politique. Cela vous permet de mettre en œuvre le principe du moindre privilège.

Lors de la création d'un espace d'agent

Depuis la console de DevOps l'agent dans la console AWS de gestion...

- Choisissez Créer un nouveau rôle d' DevOps agent à l'aide d'un document de politique et suivez les instructions

Lors de la modification d'un espace d'agent

Depuis la console de DevOps l'agent dans la console AWS de gestion...

- Sélectionnez l'onglet Fonctionnalités
- Sélectionnez le compte secondaire que vous souhaitez modifier dans la section Cloud et cliquez sur Modifier
- Choisissez Créer une nouvelle politique d' DevOps agent à l'aide d'un modèle et suivez les instructions

Bonnes pratiques en matière de politiques personnalisées

- Accorder uniquement des autorisations en lecture seule : évitez les autorisations qui autorisent la modification ou la suppression de ressources
- Utilisez des autorisations au niveau des ressources lorsque cela est possible : limitez l'accès à des ressources spécifiques à l'aide de modèles ou de balises ARN
- Vérifiez et auditez régulièrement les autorisations : passez régulièrement en revue les politiques IAM de l'agent pour vous assurer qu'elles sont toujours conformes à vos exigences de sécurité

Configuration de l'authentification IAM Identity Center

L'authentification IAM Identity Center fournit un moyen centralisé de gérer l'accès des utilisateurs à l'application Web AWS DevOps Agent Space. Ce guide explique comment configurer l'authentification IAM Identity Center et comment gérer les utilisateurs.

Conditions préalables

Avant de configurer l'authentification IAM Identity Center, assurez-vous d'avoir :

- IAM Identity Center activé dans votre organisation ou votre compte
- Autorisations d'administrateur dans AWS DevOps l'agent
- Un espace agent configuré ou prêt à être créé

Options d'authentification

AWS DevOps L'agent propose deux méthodes d'authentification pour accéder à l'application Web Agent Space :

Authentification IAM Identity Center : recommandée pour les environnements de production. Assure une gestion centralisée des utilisateurs, une intégration avec des fournisseurs d'identité externes et des sessions d'une durée maximale de 12 heures.

Accès administrateur (authentification IAM) : fournit un accès rapide aux administrateurs lors de l'installation et de la configuration initiales. Les séances sont limitées à 30 minutes.

Configuration d'IAM Identity Center lors de la création de l'espace agent

Lorsque vous créez un espace agent, vous pouvez configurer l'authentification IAM Identity Center dans l'onglet Accès :

Étape 1 : Accédez à la configuration de l'application Web

1. Après avoir configuré les détails de votre espace agent et l'accès à votre AWS compte, passez à l'onglet Accès
2. Vous verrez deux sections : « Connect IAM Identity Center » et « Accès administrateur »

Étape 2 : configurer l'intégration d'IAM Identity Center

Dans la section Connect [Agent Space] à IAM Identity Center :

1. Vérifiez l'instance IAM Identity Center : la console indique quelle instance Identity Center gérera l'accès des utilisateurs de l'application Web (par exemple,`ssoins-7223a9580931edbe`). L'instance IAM Identity Center la plus proche sera automatiquement préremplie.

2. Sélectionnez l'option Nom du rôle de l'application IAM Identity Center : choisissez l'une des trois options suivantes :

Création automatique d'un nouveau rôle DevOps d'agent (recommandé) :

- Le système crée automatiquement un nouveau rôle de service avec les autorisations appropriées
- Il s'agit de l'option la plus simple et elle fonctionne pour la plupart des cas d'utilisation

Attribuez un rôle existant :

- Utiliser un rôle IAM existant que vous avez déjà créé
- Le système vérifiera que le rôle dispose des autorisations requises
- Choisissez cette option si votre organisation a précréé des rôles pour l'agent AWS DevOps

Créez un nouveau rôle d' DevOps agent à l'aide d'un modèle de politique :

- Utilisez les détails de politique fournis pour créer votre propre rôle personnalisé dans la console IAM
- Choisissez cette option si vous devez personnaliser les autorisations des rôles

Après avoir cliqué sur Connect, le système :

- Crée ou configure le rôle IAM spécifié
- Configure une application IAM Identity Center pour votre espace d'agent
- Établit des relations de confiance entre IAM Identity Center et l'application Web Agent Space
- Configure les flux d'authentification OAuth 2.0 pour un accès utilisateur sécurisé

Alternative : utilisation de l'accès administrateur

Si vous souhaitez accéder immédiatement à l'application Web Agent Space sans configurer IAM Identity Center :

1. Dans la section Accès administrateur, notez l'ARN du rôle IAM qui fournit un accès administrateur (par exemple,`arn:aws:iam::440491339484:role/service-role/DevOpsAgentRole-WebappAdmin-15ppoc42`)

2. Cliquez sur le bouton bleu d'accès administrateur pour lancer l'application Web Agent Space avec l'authentification IAM
3. Les sessions utilisant cette méthode sont limitées à 30 minutes

> **Note**
>
> L'accès administrateur est destiné à l'installation et à la configuration initiales. Pour une utilisation en production et pour les opérations en cours, configurez l'authentification IAM Identity Center.

Ajout d'utilisateurs et de groupes

Après avoir configuré l'authentification IAM Identity Center, vous devez autoriser des utilisateurs et des groupes spécifiques à accéder à l'application Web Agent Space :

Étape 1 : Accès à la gestion des utilisateurs

1. Dans la console AWS DevOps Agent, sélectionnez votre espace agent
2. Accédez à l'onglet Accès
3. Sous Accès utilisateur, cliquez sur Gérer les utilisateurs et les groupes

Étape 2 : ajouter des utilisateurs ou des groupes

1. Choisissez Ajouter des utilisateurs ou des groupes
2. Recherchez des utilisateurs ou des groupes dans votre annuaire IAM Identity Center
3. Cochez les cases à côté des utilisateurs ou des groupes que vous souhaitez ajouter
4. Cliquez sur Ajouter pour leur accorder l'accès

Les utilisateurs sélectionnés peuvent désormais accéder à l'application Web Agent Space à l'aide de leurs informations d'identification IAM Identity Center.

Travailler avec des fournisseurs d'identité externes

Si vous utilisez un fournisseur d'identité externe (tel qu'Okta, Microsoft Entra ID ou Ping Identity) avec IAM Identity Center :

- Les utilisateurs et les groupes sont synchronisés entre votre fournisseur d'identité externe et IAM Identity Center
- Lorsque vous ajoutez des utilisateurs et des groupes à l'application Web Agent Space, vous effectuez une sélection dans le répertoire synchronisé
- Les attributs des utilisateurs et les appartenances aux groupes sont gérés par votre fournisseur d'identité externe
- Les modifications apportées à votre fournisseur d'identité sont automatiquement reflétées dans IAM Identity Center après la synchronisation

Comment les utilisateurs accèdent à l'application Web Agent Space

Après avoir ajouté des utilisateurs à votre espace agent :

1. Partagez l'URL de l'application Web Agent Space avec les utilisateurs autorisés
2. Lorsque les utilisateurs accèdent à l'URL, ils sont redirigés vers la page de connexion d'IAM Identity Center
3. Après avoir saisi leurs informations d'identification (et terminé le MFA si configuré), ils sont redirigés vers l'application Web Agent Space
4. Leur session est valide pendant 8 heures par défaut (configurable par l'administrateur d'Identity Center)

Gestion de l'accès des utilisateurs

Vous pouvez mettre à jour l'accès des utilisateurs à tout moment :

Ajouter des utilisateurs ou des groupes supplémentaires :

- Suivez les mêmes étapes décrites ci-dessus pour ajouter des utilisateurs ou des groupes supplémentaires

Suppression de l'accès :

1. Dans la section Accès utilisateur, recherchez l'utilisateur ou le groupe à supprimer
2. Cliquez sur le bouton Supprimer à côté de son nom
3. Confirmez la suppression

Les utilisateurs supprimés perdront immédiatement leur accès, mais les sessions actives peuvent se poursuivre jusqu'à leur expiration.

Gestion de session

Les sessions IAM Identity Center pour l'application Web Agent Space présentent les caractéristiques suivantes :

- Durée de session par défaut : 8 heures
- Sécurité de session : cookies HTTP uniquement pour une protection renforcée
- Authentification multifactorielle : prise en charge lorsqu'elle est configurée dans IAM Identity Center
- Informations d'identification API — Des informations d'identification SigV4 de courte durée (15 minutes) sont émises pour les appels d'API et renouvelées automatiquement

Pour configurer la durée de session, procédez comme suit :

1. Accédez à la console IAM Identity Center
2. Accédez à Réglages > Authentification
3. Sous Durée de session, configurez votre durée préférée (de 1 heure à 12 heures)
4. Choisissez Enregistrer les modifications

Déconnexion du centre d'identité

1. Dans la console de votre Agent Space, cliquez sur Actions en haut à droite et sélectionnez Déconnecter du centre d'identité IAM
2. Confirmer dans la boîte de dialogue de confirmation

Configuration de l'authentification par fournisseur d'identité externe (IdP)

L'authentification par fournisseur d'identité externe (IdP) permet à votre organisation d'utiliser un fournisseur d'identité compatible OIDC existant, tel qu'Okta ou Microsoft Entra ID, pour gérer l'accès des utilisateurs à l' AWS DevOps application Web Agent Space. Les utilisateurs se connectent à l'aide de leurs identifiants d'entreprise directement via votre IdP, sans avoir besoin d' AWS IAM Identity Center.

Conditions préalables

Avant de configurer l'authentification IdP externe, assurez-vous d'avoir :

- Un fournisseur d'identité compatible OIDC (Okta ou Microsoft Entra ID)
- Accès administrateur à votre fournisseur d'identité
- Autorisations d'administrateur pour accéder à la console de AWS DevOps l'agent
- Un espace agent configuré ou prêt à être créé

Comment ça marche

Lorsque vous configurez l'authentification IdP externe :

- Les utilisateurs accèdent à l'URL de l'application Web Agent Space
- Ils sont redirigés vers la page de connexion de votre fournisseur d'identité
- Après s'être authentifiés à l'aide de leurs informations d'identification professionnelles, ils sont redirigés vers l'application Web
- L'application Web échange le jeton d'authentification contre des AWS informations d'identification de courte durée accessibles à l'espace agent

Les sessions sont valides pour une durée maximale de 8 heures. Les informations d'identification sont automatiquement actualisées à l'aide de jetons d'actualisation OIDC sans que les utilisateurs n'aient à s'authentifier à nouveau.

Configuration de l'authentification IdP externe

Étape 1 : Enregistrez une demande auprès de votre fournisseur d'identité

Choisissez votre fournisseur d'identité et suivez les instructions de configuration correspondantes.

Option A : Okta

1. Dans la console d'administration Okta, accédez à Applications > Applications et choisissez Create App Integration
2. Sélectionnez OIDC - OpenID Connect comme méthode de connexion et Application Web comme type d'application. Choisissez Next (Suivant)

3. Définissez un nom descriptif pour l'application (par exemple,`AWS DevOps Agent`)
4. Sous Type de subvention, assurez-vous que les éléments suivants sont cochés :
   - Code d'autorisation (par défaut)
   - Jeton d'actualisation — Ceci est nécessaire pour actualiser la session. Si cette option n'est pas activée, les utilisateurs ne seront pas en mesure de maintenir les sessions.

> **Note**
>
> Okta n'active pas le type de subvention Refresh Token par défaut. Vous devez l'activer explicitement.

1. Laissez la redirection de connexion URIs comme valeur par défaut pour le moment. Vous la mettrez à jour après avoir configuré l'espace agent
2. Sous Attributions, attribuez les utilisateurs ou les groupes qui devraient avoir accès
3. Choisissez Enregistrer.
4. Dans l'onglet Général de l'application, notez les valeurs suivantes :
   - Identifiant du client
   - Secret client — Choisissez Copier pour enregistrer cette valeur en toute sécurité
5. Notez votre domaine Okta : il s'agit de l'URL de votre émetteur (par exemple,`https://dev-12345678.okta.com`).

> **Note**
>
> Dans l'onglet Connexion, vérifiez que l'émetteur est défini sur Okta URL (et non sur Dynamic). Cela garantit la stabilité de l'URL de l'émetteur.

> **Note**
>
> N'ajoutez pas de réclamation de groupe au jeton d'identification dans l'onglet Réclamations de votre serveur d'autorisation. AWS DevOps L'agent n'utilise pas l'appartenance à un groupe depuis votre IdP.

Option B : identifiant Microsoft Entra

1. Sur le portail Azure, accédez à Microsoft Entra ID > Enregistrements d'applications > Nouvel enregistrement
2. Définissez un nom descriptif (par exemple,`AWS DevOps Agent`)
3. Sous Types de comptes pris en charge, sélectionnez l'option appropriée pour votre organisation (généralement, les comptes de ce répertoire d'organisation uniquement)
4. Laissez l'URI de redirection vide pour le moment. Choisissez S'inscrire
5. Sur la page de présentation de l'application, notez les valeurs suivantes :
   - ID de l'application (client) : utilisé comme identifiant client lors de la configuration de l'espace agent
   - ID du répertoire (tenant) : utilisé pour créer l'URL de l'émetteur
6. Accédez à Certificats et secrets > Nouveau secret client
   - Définissez une description et une période d'expiration
   - Choisissez Ajouter et copiez immédiatement la valeur secrète ; elle ne sera plus affichée
7. L'URL de l'émetteur de l'identifiant Entra suit ce format. `{tenant-id}`Remplacez-le par votre ID de répertoire (tenant) indiqué à l'étape 5 :
   - `https://login.microsoftonline.com/{tenant-id}/v2.0`

> **Note**
>
> N'activez pas la réclamation facultative du groupe dans la configuration des jetons. AWS DevOps L'agent n'utilise pas l'appartenance à un groupe depuis votre IdP.

Étape 2 : activer l'application Operator avec l'authentification IdP

1. Dans la console AWS DevOps Agent, sélectionnez votre espace agent
2. Accédez à l'onglet Accès
3. Sous Accès utilisateur, choisissez Fournisseur d'identité externe
4. Dans le formulaire de configuration, configurez les éléments suivants :
   - Fournisseur d'identité : sélectionnez votre fournisseur d'identité (Okta ou Microsoft Entra ID)
   - URL de l'émetteur : URL de l'émetteur OIDC de votre fournisseur d'identité

- ID client : ID client de l'application OIDC que vous avez créée
- Secret client — Le secret client de votre application OIDC

5. Sous Nom du rôle de l'application du fournisseur d'identité, choisissez l'une des trois options suivantes :
   - Création automatique d'un nouveau rôle d' DevOps agent (recommandé) : crée un nouveau rôle de service avec les autorisations appropriées
   - Attribuer un rôle existant : utilisez un rôle IAM existant que vous avez déjà créé
   - Création d'un nouveau rôle d' DevOps agent à l'aide d'un modèle de politique : utilisez les informations fournies pour créer votre propre rôle dans la console IAM
6. Consultez l'alerte d'avertissement relative à l'URL de rappel affichée au bas du formulaire. Copiez cette URL : vous devrez l'ajouter à la redirection autorisée par votre fournisseur d'identité URIs avant que les utilisateurs puissent se connecter.
7. Choisissez Connect (Connexion).

Après avoir choisi Connect, la console affiche la configuration du fournisseur d'identité externe avec les détails suivants :

- Fournisseur : le fournisseur d'identité que vous avez sélectionné
- URL de l'émetteur : URL de l'émetteur OIDC configurée
- ID client : ID client configuré
- ARN du rôle IAM — Le rôle IAM utilisé pour l'accès des utilisateurs
- URL de rappel — Configurez cette URL dans votre fournisseur d'identité en tant qu'URI de redirection autorisée
- URL de connexion : utilisez cette URL pour accéder à l'application Web via votre fournisseur d'identité

Étape 3 : Ajoutez l'URL de rappel à votre fournisseur d'identité

Okta

1. Dans la console d'administration Okta, accédez à l'onglet Général de votre application
2. Sous Connexion, choisissez Modifier
3. Ajoutez l'URL de rappel en tant qu'URI de redirection de connexion :

- `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/callback`

4. (Facultatif) Définissez l'URI d'initiation de connexion pour activer la connexion initiée par l'IdP depuis le tableau de bord Okta :
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/login`
5. (Recommandé) Ajoutez une URI de redirection de déconnexion pour rediriger les utilisateurs vers l'application Web après la déconnexion. Sans cela, les utilisateurs peuvent voir une page d'erreur s'afficher lorsqu'ils se déconnectent :
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome`
6. Choisissez Enregistrer.

Identifiant Microsoft Entra

1. Dans le portail Azure, accédez à la page d'authentification de votre application
2. Sous Configurations de plate-forme, choisissez Ajouter une plate-forme > Web
3. Entrez l'URL de rappel en tant qu'URI de redirection :
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/callback`
4. (Facultatif) Ajoutez une URI de redirection de déconnexion pour rediriger les utilisateurs vers l'application Web après la déconnexion :
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome`
5. Choisissez Configurer

Étape 4 : vérifier la configuration

1. Accédez à l'URL de connexion affichée dans la console :
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/login`
2. Vous devriez être redirigé vers la page de connexion de votre fournisseur d'identité
3. Connectez-vous à l'aide des informations d'identification de votre entreprise
4. Une fois l'authentification réussie, vous êtes redirigé vers l'application Web Agent Space

Mise à jour de la configuration de l'IdP

Vous pouvez faire pivoter le secret du client sans vous déconnecter :

1. Dans la console AWS DevOps Agent, sélectionnez votre espace agent
2. Accédez à l'onglet Accès
3. Sous Configuration du fournisseur d'identité externe, choisissez Rotate client secret
4. Entrez le nouveau secret du client
5. Choisissez Enregistrer.

Pour modifier un autre champ de configuration d'IdP (tel que l'URL de l'émetteur, l'ID client ou le fournisseur d'identité), vous devez déconnecter l'IdP existant et en configurer un nouveau.

Comment les utilisateurs accèdent à l'application Web Agent Space

Après avoir configuré l'authentification IdP externe :

- Partagez l'URL de l'application Web Agent Space avec les utilisateurs autorisés
- Lorsque les utilisateurs accèdent à l'URL, ils sont redirigés vers la page de connexion de votre fournisseur d'identité
- Après avoir saisi leurs informations d'identification (et terminé le MFA si configuré par votre IdP), ils sont redirigés vers l'application Web Agent Space
- Les sessions sont actualisées automatiquement — voir Gestion des sessions pour plus de détails

Gestion de session

Les sessions IdP externes pour l'application Web Agent Space présentent les caractéristiques suivantes :

- Durée de session — Les sessions du navigateur durent jusqu'à 8 heures. Ceci n'est pas configurable dans AWS DevOps l'Agent. Si la durée de vie de la session de votre IdP dépasse 8 heures, les utilisateurs peuvent être réauthentifiés automatiquement lors de leur prochaine visite sans saisir d'informations d'identification. Configurez la durée de vie des sessions et des jetons de votre IdP conformément aux exigences de sécurité de votre organisation.
- Actualisation des informations d'identification : les sessions sont automatiquement actualisées à l'aide de jetons d'actualisation OIDC sans que les utilisateurs n'aient à s'authentifier à nouveau
- Authentification multifactorielle : prise en charge lorsqu'elle est configurée dans votre fournisseur d'identité. L'IdP gère le MFA lors de la connexion ; aucune configuration supplémentaire n'est nécessaire dans l'Agent AWS DevOps

Comportement de déconnexion

Lorsqu'un utilisateur clique sur Déconnexion dans l'application Web :

1. Tous les cookies de session sont immédiatement effacés
2. L'utilisateur est redirigé vers le point de terminaison de déconnexion OIDC du fournisseur d'identité pour mettre fin à la session SSO
3. Si une URI de redirection de déconnexion est configurée, l'utilisateur est redirigé vers la page d'accueil de l'application Web

Révocation de l'accès utilisateur

Pour révoquer immédiatement l'accès d'un utilisateur, vous pouvez révoquer ses sessions directement sur le portail d'administration de votre fournisseur d'identité :

- Okta — Dans la console d'administration Okta, accédez à Annuaire > Personnes, sélectionnez l'utilisateur, choisissez Plus d'actions > Effacer les sessions utilisateur
- Microsoft Entra ID : dans le portail Azure, accédez à Utilisateurs, sélectionnez l'utilisateur, puis choisissez Révoquer les sessions

Considérations sur la sécurité

Stockage du secret client : le secret client que vous fournissez lors de la configuration est chiffré à l'aide de votre clé KMS gérée par le client si vous en avez fourni une lors de la création de l'espace agent, ou d'une clé appartenant au service dans le cas contraire. Il n'est jamais renvoyé dans les réponses de l'API ni affiché dans la console après la configuration initiale.

Rotation des secrets clients — Les secrets clients Entra ont une date d'expiration configurable. Définissez un rappel pour faire pivoter le secret avant son expiration à l'aide de l'option Rotation du secret client dans la console de l' AWS DevOps agent. Si le secret expire, les utilisateurs ne pourront pas se connecter tant qu'il n'aura pas été modifié.

Gestion de la durée de vie des jetons — La durée de vie des jetons (jetons d'accès, jetons d'actualisation) émis par votre fournisseur d'identité est contrôlée par la configuration de votre IdP. Nous vous recommandons de configurer les durées de vie des jetons appropriées dans votre IdP :

- Okta — Configurez la durée de vie des jetons sous Sécurité > API > Serveurs d'autorisation > Politiques d'accès

- Microsoft Entra ID : configurez la durée de vie des jetons à l'aide des politiques de durée de vie des jetons

Réclamation de groupe : n'activez pas la réclamation de groupe dans la configuration des jetons de votre fournisseur d'identité. AWS DevOps L'agent n'utilise actuellement pas l'appartenance à un groupe depuis votre IdP.

Identifiant utilisateur — AWS DevOps L'agent utilise une réclamation spécifique au fournisseur pour identifier les utilisateurs de manière unique :

- Okta — Utilise la `sub` réclamation du jeton d'identification
- Microsoft Entra ID — Utilise la réclamation `oid` (identifiant de l'objet) contenue dans le jeton d'identification

Ces identifiants sont immuables et apparaissent dans les CloudTrail journaux à des fins d'audit.

Déconnexion d'un IdP externe

1. Dans la console AWS DevOps Agent, sélectionnez votre espace agent
2. Accédez à l'onglet Accès
3. Sous Accès utilisateur, choisissez Déconnecter
4. Passez en revue les impacts répertoriés dans la boîte de dialogue de confirmation et confirmez

La déconnexion permettra de :

- Supprimer la configuration IdP de l'espace agent
- Empêcher les utilisateurs de se connecter via le fournisseur d'identité externe
- Supprimer l'historique des discussions individuelles et des artefacts associés aux comptes utilisateurs IdP

Les sessions utilisateur actives se poursuivront jusqu'à leur expiration ou jusqu'à ce que la prochaine actualisation des informations d'identification échoue.

Résolution des problèmes

- La redirection vers l'IdP échoue : vérifiez que l'URL de l'émetteur correspond au point de terminaison de découverte OIDC de votre IdP. Pour Okta, assurez-vous que l'émetteur est défini sur l'URL Okta (et non sur Dynamic) dans l'onglet Sign On. Pour Entra, utilisez le format`https://login.microsoftonline.com/{tenant-id}/v2.0`.
- Accès refusé ou erreur de politique (Okta) : vérifiez que l'utilisateur ou son groupe est affecté à l'application sous Affectations. Cochez la case Connexion **>** Règles relatives à la politique de connexion.
- Erreur de configuration de l'IdP après la connexion : votre fournisseur d'identité n'a pas renvoyé de jeton d'actualisation. Assurez-vous que le `offline_access` champ d'application et le type d'autorisation du jeton d'actualisation sont activés :
  - Okta — Accédez à l'onglet Général de votre application et cochez la case Actualiser le jeton sous Type de subvention
  - Entra — Accédez aux autorisations de l'API et assurez-vous qu'elles `offline_access` figurent dans la section Autorisations déléguées
- L'authentification réussit mais l'application Web affiche une erreur. Vérifiez que l'URI de redirection de votre IdP correspond exactement à l'URL de rappel affichée dans AWS DevOps la console de l'agent.
- Échecs d'authentification — Si la réclamation facultative du groupe est activée dans votre IdP, désactivez-la. AWS DevOps L'agent n'utilise pas les réclamations de groupe.
- La connexion échoue après l'authentification IdP — Pour Entra, la vérification n'**`requestedAccessTokenVersion`**est pas définie sur **`null`** dans le manifeste de l'application. Pour Okta, vérifiez que l'URL de l'émetteur est correcte.
- Page d'erreur après avoir cliqué sur Déconnexion (Okta) : si un **`post_logout_redirect_uri`** message d'erreur s'affiche après la déconnexion, ajoutez **`https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome`** une URI de redirection de déconnexion dans l'onglet Général de votre application Okta.
- Les utilisateurs restent sur la page du fournisseur d'identité après la déconnexion (Entra) — Pour rediriger les utilisateurs vers l'application Web après la déconnexion, ajoutez `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome` un URI de redirection sur la page d'authentification de votre application Entra.

Chiffrement au repos pour AWS DevOps l'agent

AWS DevOps L'agent chiffre toutes les données clients au repos. Par défaut, AWS DevOps l'Agent AWS utilise ses propres clés pour chiffrer automatiquement vos données sans frais supplémentaires. Vous ne pouvez pas consulter, gérer ou auditer l'utilisation des clés AWS détenues. Cependant, vous n'avez aucune action à effectuer pour protéger ces clés. Vos données sont automatiquement sécurisées.

Vous pouvez choisir de chiffrer vos données à l'aide d'une clé symétrique gérée par le client que vous créez, détenez et gérez dans AWS Key Management Service (AWS KMS). Comme vous avez le contrôle total de cette couche de chiffrement, vous pouvez effectuer des tâches telles que les suivantes :

- Établissement et gestion des stratégies de clé
- Activation et désactivation des stratégies de clé
- Rotation des matériaux de chiffrement de clé
- Ajout de balises
- Création d'alias de clé
- Planification des clés pour la suppression

Pour plus d'informations, consultez la section Clés gérées par le client dans le Guide du développeur du service de gestion des AWS clés.

> **Note**
>
> AWS DevOps L'agent active automatiquement le chiffrement au repos à l'aide de clés AWS détenues pour protéger gratuitement les données des clients. Les frais AWS KMS standard s'appliquent lorsque vous utilisez une clé gérée par le client. Pour plus d'informations sur la tarification, consultez la section Tarification du service de gestion des AWS clés.

Clés gérées par le client

Les clés gérées par le client sont des clés KMS de votre AWS compte que vous créez, détenez et gérez. Vous avez le contrôle total de ces clés KMS, y compris l'établissement et la mise à jour de leurs politiques clés.

Lorsque vous configurez une clé gérée par le client, AWS DevOps l'agent l'utilise pour protéger les données de ressources sensibles. AWS DevOps L'agent utilise le chiffrement des enveloppes avec le jeu de clés hiérarchique du SDK de AWS chiffrement. Votre clé KMS est utilisée pour générer des clés de branche qui, à leur tour, protègent vos données.

Vous pouvez spécifier une clé gérée par le client lorsque vous créez les ressources suivantes :

- Espace agent : chiffre les détails de l'espace agent et le contenu créés à partir de l'application Web de l' DevOps agent en rapport avec les enquêtes, les compétences et le chat.
- Service : chiffre les informations d'identification des services tiers au repos.

Pour configurer une clé gérée par le client dans AWS DevOps l'Agent, procédez comme suit.

Étape 1 : Créer une clé gérée par le client

Vous pouvez créer une clé symétrique gérée par le client à l'aide de la console AWS KMS ou de l'API AWS KMS. La clé doit répondre aux exigences suivantes :

Propriété	Exigence
Type de clé	Symétrique
Spécifications de la clé	`SYMMETRIC_DEFAULT`
Utilisation de la clé	`ENCRYPT_DECRYPT`

> **Note**
>
> AWS DevOps L'agent ne prend en charge que les clés KMS de chiffrement symétriques avec la spécification de la `SYMMETRIC_DEFAULT` clé et l'utilisation de la `ENCRYPT_DECRYPT` clé. Les clés multirégionales et les clés asymétriques ne sont actuellement pas prises en charge.

Pour plus d'informations, consultez la section Création d'une clé symétrique gérée par le client dans le Guide du développeur du service de gestion des AWS clés.

Étape 2 : définir la politique clé

Les stratégies de clés contrôlent l’accès à votre clé gérée par le client. Chaque clé gérée par le client doit avoir exactement une stratégie de clé, qui contient des instructions qui déterminent les personnes pouvant utiliser la clé et comment elles peuvent l’utiliser.

Votre politique clé doit accorder des autorisations à la fois au principal appelant (votre identité IAM) et au service de l' AWS DevOps agent. AWS DevOps L'agent accède à votre clé à l'aide de deux ensembles d'informations d'identification :

1. Vos informations d'identification de l'appelant : utilisées pour toutes les opérations synchrones, y compris la validation des clés, le chiffrement au moment de la création des ressources et tout appel d'API renvoyant une réponse directe à l'appelant.
2. AWS DevOps Agent de service principal : utilisé pour les opérations asynchrones exécutées en arrière-plan, telles que les enquêtes opérationnelles, l'analyse des incidents, la corrélation des événements et la génération d'analyses des causes premières.

Le tableau suivant répertorie les actions KMS requises :

Action KMS	Description
`kms:DescribeKey`	Valider la configuration des clés au moment de la création de la ressource
`kms:GenerateDataKey`	Génération de clés de chiffrement des données pour le chiffrement des enveloppes
`kms:Decrypt`	Déchiffrer des données
`kms:Encrypt`	Chiffrer des données
`kms:ReEncrypt`	Chiffrez à nouveau les données sous la même clé ou une clé différente

AWS DevOps L'agent valide toutes ces autorisations au moment de la configuration à l'aide d'opérations d'exécution à sec. Si une autorisation est manquante, la demande échoue avec une exception.

Voici un exemple de stratégie de clé. Remplacez les valeurs de l'espace réservé par les vôtres.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCallerAccessViaService",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/DevOpsAgentUserRole"
      },
      "Action": [
        "kms:DescribeKey",
        "kms:GenerateDataKey*",
        "kms:Decrypt",
        "kms:Encrypt",
        "kms:ReEncrypt*"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "aidevops.us-east-1.amazonaws.com"
        }
      }
    },
    {
      "Sid": "AllowDevOpsAgentServiceDescribeKeyAccess",
      "Effect": "Allow",
      "Principal": {
        "Service": "aidevops.amazonaws.com"
      },
      "Action": [
        "kms:DescribeKey"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowDevOpsAgentAccessForAgentSpace",
      "Effect": "Allow",
      "Principal": {
        "Service": "aidevops.amazonaws.com"
      },
      "Action": [
        "kms:GenerateDataKey*",
```

```
        "kms:Decrypt",
        "kms:Encrypt",
        "kms:ReEncrypt*"
      ],
      "Resource": "*",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:aidevops:us-east-1:111122223333:agentspace/*"
        },
        "StringLike": {
          "kms:EncryptionContext:aws-crypto-ec:aws:aidevops:arn": "arn:aws:aidevops:us-
east-1:111122223333:agentspace/*"
        }
      }
    },
    {
      "Sid": "AllowDevOpsAgentAccessForService",
      "Effect": "Allow",
      "Principal": {
        "Service": "aidevops.amazonaws.com"
      },
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt",
        "kms:Encrypt",
        "kms:ReEncrypt*"
      ],
      "Resource": "*",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:aidevops:us-east-1:111122223333:service/*"
        },
        "StringLike": {
          "kms:EncryptionContext:aws-crypto-ec:aws:aidevops:arn": "arn:aws:aidevops:us-
east-1:111122223333:service/*"
        }
      }
    }
  ]
}
```

La politique contient les déclarations suivantes :

- AllowKeyAdministration— Accorde au root du compte un accès administratif complet à la clé. Remplacez `111122223333` par votre identifiant de AWS compte.
- AllowCallerAccessViaService— Accorde à vos principaux IAM les autorisations KMS requises pour toutes les opérations synchrones AWS DevOps de l'agent. Cela inclut la validation des clés au moment de la création de la ressource, ainsi que les opérations de chiffrement et de déchiffrement pour tout appel d'API renvoyant une réponse directe à l'appelant. Cette `kms:ViaService` condition garantit que vous ne pouvez utiliser la clé que via le service AWS DevOps Agent. `111122223333`Remplacez-le par votre numéro de AWS compte et `us-east-1` par votre AWS région.
- AllowDevOpsAgentServiceAccessForAgentSpace/AllowDevOpsAgentServiceAccessForService— Accorde au principal du `aidevops.amazonaws.com` service les autorisations KMS requises pour les opérations asynchrones. AWS DevOps L'agent utilise ce principe de service pour chiffrer et déchiffrer vos données lorsqu'il effectue des opérations en arrière-plan, telles que des enquêtes opérationnelles, l'analyse d'incidents, la corrélation d'événements entre les services et la génération d'analyses des causes premières. Sans cet accès, AWS DevOps l'agent ne peut pas lire les données chiffrées nécessaires pour mener des enquêtes en votre nom. La `aws:SourceArn` condition restreint l'accès aux demandes provenant des ressources de votre AWS DevOps agent et garantit que le `kms:EncryptionContext` contexte de chiffrement correspond à votre ressource ARNs. `111122223333`Remplacez-le par votre numéro de AWS compte et `us-east-1` par votre AWS région.

Pour plus d'informations sur les politiques clés, consultez la section Politiques clés dans AWS KMS dans le Guide du développeur du service de gestion des AWS clés.

Étape 3 : Spécifier la clé lors de la création d'une ressource

Après avoir créé votre clé et configuré la politique des clés, vous pouvez spécifier la clé lors de la création des ressources de AWS DevOps l'agent.

Console

Pour configurer une clé gérée par le client lors de la création d'un espace agent dans la console :

1. Ouvrez la console de AWS DevOps l'agent.
2. Choisissez Create Agent Space ou Register Service.
3. Entrez les détails de l'espace de l'agent (nom, description et rôle IAM).
4. Développez la section Configuration avancée.

5. Sous Type de clé de chiffrement, sélectionnez Clé gérée par le client.
6. Choisissez une clé KMS dans la liste déroulante ou entrez un ARN de clé KMS.
7. Passez en revue la politique clé affichée dans la section extensible Politique clé. Assurez-vous d'avoir joint cette politique à votre clé KMS. Vous pouvez utiliser le bouton de copie pour copier la politique.
8. Terminez la configuration restante et choisissez Create.

> **Note**
>
> Si votre clé KMS ne figure pas dans la liste déroulante, vérifiez qu'elle répond aux exigences de l'étape 1 et que vous disposez des `kms:ListKeys` `kms:DescribeKey` autorisations nécessaires.

API

Création d'un espace d'agent avec une clé gérée par le client

Spécifiez le `kmsKeyArn` paramètre lors de la création d'un espace d'agent. La valeur doit être l'ARN complet de la clé KMS.

```
{
  "name": "my-agent-space",
  "description": "An encrypted agent space",
  "kmsKeyArn": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

Enregistrement d'un service avec une clé gérée par le client

Spécifiez le `kmsKeyArn` paramètre lors de l'enregistrement d'un service. La valeur doit être l'ARN complet de la clé KMS. Ce paramètre est pris en charge par tous les types de services, y compris les serveurs Dynatrace, ServiceNow, PagerDuty, GitLab GitHub, et MCP.

```
{
  "service": "dynatrace",
  "kmsKeyArn": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
```

```
  "serviceDetails": { ... }
}
```

> **Note**
>
> Vous devez spécifier la clé gérée par le client au moment de la création de la ressource. Vous ne pouvez pas ajouter ou modifier la clé gérée par le client pour une ressource existante.

AWS DevOps Contexte de chiffrement de l'agent

Un contexte de chiffrement est un ensemble de paires clé-valeur non secrètes qui contiennent des informations contextuelles supplémentaires sur les données. AWS KMS utilise le contexte de chiffrement comme données authentifiées supplémentaires pour prendre en charge le chiffrement authentifié. Lorsque vous incluez un contexte de chiffrement dans une demande de chiffrement de données, AWS KMS lie le contexte de chiffrement aux données chiffrées. Pour déchiffrer les données, vous devez inclure le même contexte de chiffrement dans la demande.

AWS DevOps L'agent utilise le contexte de chiffrement suivant pour toutes les opérations cryptographiques :

```
{
  "aws-crypto-ec:aws:aidevops:arn": "arn:aws:aidevops:{region}:{accountId}:
{resourceType}/{resourceId}"
}
```

La valeur du contexte de chiffrement est l'ARN de la ressource de l' AWS DevOps agent cryptée. Vous pouvez utiliser ce contexte de chiffrement dans les conditions de votre politique de clé et dans AWS CloudTrail les journaux pour vérifier la manière dont votre clé est utilisée.

Gestion des clés

Si vous désactivez ou planifiez la suppression de votre clé KMS, AWS DevOps l'agent ne peut pas déchiffrer vos données. Cela entraîne `AccessDeniedException` des erreurs lors des opérations de lecture de données cryptées.

> **Important**
>
> Si vous choisissez d'utiliser une clé gérée par le client, vous êtes responsable de la gestion de la clé et de ses autorisations. Si la clé est désactivée ou supprimée, ou si AWS DevOps l'agent perd l'autorisation d'utiliser la clé, vous perdez l'accès aux données chiffrées.

Le tableau suivant décrit les scénarios de défaillance courants :

Action	Impact
Principales autorisations politiques révoquées	`AccessDeniedException` sur les opérations de chiffrement et de déchiffrement
La clé KMS est désactivée	`DisabledException` sur les opérations de chiffrement et de déchiffrement
La suppression de la clé KMS est planifiée	`KMSInvalidStateException` sur les opérations de chiffrement et de déchiffrement
La clé KMS est supprimée	Perte de données permanente : les données cryptées ne peuvent pas être récupérées

Avant de désactiver ou de supprimer une clé :

1. Vérifiez qu'aucune ressource active de AWS DevOps l'agent ne dépend de la clé.
2. Envisagez de désactiver d'abord la clé pour tester l'impact avant de planifier la suppression.
3. AWS KMS impose une période d'attente minimale avant la suppression de la clé, ce qui vous laisse le temps d'annuler si nécessaire.

   Remarque : AWS DevOps L'agent ne rechiffre pas automatiquement les données sous une nouvelle clé. Si vous devez passer à une nouvelle clé gérée par le client, vous devez créer une nouvelle ressource avec la nouvelle clé.

Surveillance de vos clés de chiffrement

Lorsque vous utilisez une clé gérée par le client avec AWS DevOps l'Agent, vous pouvez l'utiliser AWS CloudTrailpour suivre les demandes que AWS DevOps l'agent envoie à AWS KMS.

Vous pouvez filtrer CloudTrail les événements par :

- Source de l'événement — `kms.amazonaws.com`
- Clé contextuelle de chiffrement — `aws-crypto-ec:aws:aidevops:arn`
- ARN clé — Votre client a géré l'ARN clé dans les paramètres de la demande

Pour plus d'informations, consultez la section Journalisation des appels d'API AWS KMS AWS CloudTrail dans le Guide du développeur du service de gestion des AWS clés.

Points de terminaison d'un VPC AWS PrivateLink

Vous pouvez l'utiliser AWS PrivateLink pour créer une connexion privée entre votre VPC et AWS DevOps l'agent. Vous pouvez accéder à AWS DevOps l'agent comme s'il se trouvait dans votre VPC, sans utiliser de passerelle Internet, de périphérique NAT, de connexion VPN ou de connexion Direct Connect. Les instances de votre VPC n'ont pas besoin d'adresses IP publiques pour accéder AWS DevOps à l'Agent.

Vous établissez cette connexion privée en créant un point de terminaison d'interface, alimenté par AWS PrivateLink. Nous créons une interface réseau de point de terminaison dans chaque sous-réseau que vous activez pour le point de terminaison d'interface. Il s'agit d'interfaces réseau gérées par le demandeur qui servent de point d'entrée pour le trafic destiné AWS DevOps à l'Agent.

Pour plus d'informations, consultez la section Accès aux AWS services AWS PrivateLink dans le AWS PrivateLink Guide.

Considérations relatives aux points de AWS DevOps terminaison Agent-VPC

Avant de configurer un point de terminaison d'interface pour AWS DevOps l'agent, consultez les considérations du AWS PrivateLink guide.

AWS DevOps L'agent prend en charge les appels d'API via les points de terminaison VPC suivants.

Catégorie	Suffixe de point de terminaison
AWS DevOps Actions de l'API Agent Control Plane	`aidevops`
AWS DevOps Opérations d'exécution de l'agent	`aidevops-dataplane`
AWS DevOps Événements pour agents Webhook	`event-ai`

Création d'un point de terminaison d'interface pour AWS DevOps l'agent

Vous pouvez créer un point de terminaison d'interface pour l' AWS DevOps agent à l'aide de la console Amazon VPC ou de l'interface de ligne de AWS commande (AWS CLI). Pour plus d'informations, consultez Création d'un point de terminaison d'interface dans le Guide AWS PrivateLink .

Créez un point de terminaison d'interface pour l' AWS DevOps agent en utilisant les noms de service suivants :

- com.amazonaws. {région} .aidevops
- com.amazonaws. {région} .aidevops dataplane
- com.amazonaws. {région} .event-ai

Après avoir créé le point de terminaison, vous avez la possibilité d'activer un nom d'hôte DNS privé. Activez ce nom d'hôte en sélectionnant Activer le nom de DNS privé dans la console VPC lorsque vous créez le point de terminaison d'un VPC.

Si vous activez le DNS privé pour le point de terminaison de l'interface, vous pouvez envoyer des demandes d'API à l' AWS DevOps agent en utilisant son nom DNS régional par défaut. L'exemple suivant montre le format du nom DNS régional par défaut.

- aidevops. {région} .api.aws
- plan de données aidevops. {région} .amazonaws.com
- event-ai. {région} .api.aws

Création d’une politique de point de terminaison pour votre point de terminaison d’interface

Une politique de point de terminaison est une ressource IAM que vous pouvez attacher à votre point de terminaison d’interface. La politique de point de terminaison par défaut autorise un accès complet à l' AWS DevOps agent via le point de terminaison de l'interface. Pour contrôler l'accès autorisé à l' AWS DevOps Agent depuis votre VPC, associez une politique de point de terminaison personnalisée au point de terminaison de l'interface.

Une politique de point de terminaison spécifie les informations suivantes :

- Principaux habilités à effectuer des actions (AWS comptes, utilisateurs IAM et rôles IAM).
- Les actions qui peuvent être effectuées.
- La ressource sur laquelle les actions peuvent être effectuées.

Pour plus d’informations, consultez Contrôle de l’accès aux services à l’aide de politiques de point de terminaison dans le Guide AWS PrivateLink .

Quotas

AWS DevOps Les quotas d'agents incluent le nombre d'espaces d'agents, les enquêtes simultanées, etc. Vous pouvez demander des augmentations pour certains quotas, mais tous les quotas ne peuvent pas être augmentés. Ces augmentations ne sont pas accordées immédiatement, de sorte que votre augmentation peut prendre de quelques heures à quelques jours pour entrer en vigueur. Sauf indication contraire, chaque quota est spécifique à une région.

Le tableau suivant décrit les quotas pour AWS DevOps l'agent.

Nom	Par défaut	Ajustable	Description
Espaces d'agents par compte et par région	10	Oui	Le nombre maximum d'espaces d'agent que vous pouvez créer par compte dans chaque AWS région.
Enquêtes simultanées par espace d'agent	3	Oui	Nombre maximal d'enquêtes de résolution d'inciden ts pouvant être exécutées simultané ment dans un seul espace d'agent.
Évaluations simultané es par espace d'agent	1	Non	Nombre maximal d'évaluations de prévention des incidents pouvant être exécutées simultané ment dans un seul espace d'agent.
Invocations simultané es à la demande par espace d'agent	10	Oui	Nombre maximal d'appels à la demande DevOps

Nom	Par défaut	Ajustable	Description
			pouvant être exécutés simultanément dans un seul espace d'agent.

Demande d'augmentation de quota

Vous pouvez demander une augmentation de quota en utilisant l'une des options suivantes :

- Depuis la console AWS de gestion : ouvrez la console Service Quotas. Dans le panneau de navigation, choisissez Services AWS . Sélectionnez DevOps Agent, sélectionnez un quota et suivez les instructions pour demander une augmentation de quota. Pour plus d'informations, consultez Demande d'augmentation de quota dans le Guide de l'utilisateur Service Quotas.
- Depuis la AWS CLI : utilisez la commande request-service-quota-increase AWS CLI. Pour de plus amples informations, veuillez consulter Demande d'augmentation de quota dans le Guide de l'utilisateur Service Quotas.

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.