使用者指南

AWS DevOps 代理程式

AWS DevOps 代理程式: 使用者指南

Table of Contents

About AWS DevOps 代理程式

AWS DevOps 代理程式是前沿代理程式，可解決並主動防止事件，持續改善可靠性和效能。

AWS DevOps 代理程式會調查事件，並將營運改進識別為經驗豐富的 DevOps 工程師。

代理程式的運作方式：

- 了解您的資源及其關係。
- 使用可觀測性工具、技能、程式碼儲存庫和 CI/CD 管道。
- 關聯遙測、程式碼和部署資料，以了解應用程式資源之間的關係。
- 支援多雲端和混合環境中的應用程式。

主要功能

AWS DevOps Agent 透過下列功能提供完整的事件回應和預防功能：

Always-on、自動事件回應

AWS DevOps 代理程式會自動調查問題發生的那一刻：

- 自動化事件調查 – 在警示或支援票證送達時立即開始調查
- AWS DevOps Agent Chat - 查詢您的基礎設施、分析系統運作狀態，並在整個 DevOps Agent Space Web 應用程式中使用自然語言引導調查。聊天會根據您正在檢視的頁面提供內容感知回應，無論是詢問拓撲中的資源、引導調查，還是篩選預防中的建議。
- 詳細的緩解計劃 – 提供特定動作來解決事件、驗證成功，並視需要還原變更
- 自動化事件協調 – 透過 Slack 和 ServiceNow 等您偏好的通訊管道，路由觀察、調查結果和緩解步驟
- AWS 支援整合 – 直接從調查中建立 AWS 支援案例，並提供即時內容給 AWS Support 專家

防止未來的事件

AWS DevOps 代理程式會分析歷史事件的模式，協助您從被動的消防轉向主動的操作改進：

- 針對性建議 – 提供具體、可行的改進，以強化四個關鍵領域：可觀測性（監控、提醒、記錄）、基礎設施最佳化（自動擴展、容量調校）和部署管道增強功能（測試、驗證）。

- 持續學習 – 根據團隊的意見回饋精簡建議

從您的 DevOps 工具取得更多

AWS DevOps Agent 會與您現有的工具整合，而無需變更您的工作流程：

- 應用程式資源映射 – 建立應用程式資源及其關係的拓撲圖
- 內建整合 – 使用熱門的可觀測性工具 (Amazon CloudWatch、Dynatrace、Datadog、New Relic 和 Splunk)、程式碼儲存庫和 CI/CD 管道 (GitHub 動作和儲存庫、GitLab 工作流程和儲存庫)
- 自訂工具整合 – 連接至您自己的模型內容通訊協定 (MCP) 伺服器以擴展其他工具的功能
- 對話式基礎設施查詢 – 使用自然語言查詢 AWS 資源、系統指標和警示狀態，而無需導覽多個主控台。聊天了解內容，並維護後續問題的對話歷史記錄。

AWS DevOps 代理程式的運作方式

AWS DevOps Agent 透過雙主控台架構運作。管理員使用 AWS 管理主控台來建立和管理 Agent Spaces、設定整合，以及設定存取控制。營運團隊使用 AWS DevOps Agent Web 應用程式進行day-to-day事件回應和調查活動。Web 應用程式可讓操作員與客服人員調查互動、瀏覽跨帳戶應用程式拓撲，以及了解可觀測性、程式碼、管道和基礎設施架構的預防性改善。如需詳細資訊，請參閱 the section called “主動事件預防”。

服務是以 Agent Spaces 組織，這是邏輯容器，可定義 AWS DevOps Agent 可存取和調查的內容。每個 Agent Space 都包含 AWS 您的帳戶組態、第三方工具整合和存取許可。如需詳細資訊，請參閱 the section called “什麼是 DevOps Agent Spaces？”。

AWS DevOps Agent 會自動建置應用程式拓撲，以映射您的資源及其關係。此拓撲可協助服務在調查期間了解您的應用程式架構。如需詳細資訊，請參閱 the section called “什麼是 DevOps 代理程式拓撲？”。

優勢

- 縮短解決的平均時間 (MTTR) – 自動調查會立即開始，將事件解決從數小時加速到幾分鐘
- 防止重複事件 – 針對性建議可解決根本原因並增強系統彈性
- 提高營運效率 – 讓您的團隊免於重複調查任務，專注於創新
- 在現有的工作流程中工作 – 與現有的工具和程序整合，而不會中斷

什麼是 DevOps Agent Web 應用程式？

AWS DevOps Agent 使用雙主控台架構，將管理函數與day-to-day營運活動分開。此設計可讓管理員設定服務，同時讓營運團隊專注於事件回應和預防。

主控台

AWS DevOps 代理程式提供兩種不同的界面：

- AWS 管理主控台 – 管理員使用 AWS 管理主控台來設定和管理 AWS DevOps 代理程式。在此主控台中，您可以the section called “建立 代理程式空間”連接 AWS 服務和第三方工具，並管理組織的存取許可。
- DevOps Agent Web 應用程式 - 營運團隊使用 DevOps Agent Space Web 應用程式進行每日事件回應活動。此獨立應用程式提供一個界面，可讓待命工程師啟動調查、透過自然語言聊天與客服人員互動、檢視應用程式拓撲，以及檢閱事件預防建議。

Web 應用程式功能

DevOps Agent Web 應用程式提供下列主要功能：

- 事件回應 – 此頁面可讓您建立和追蹤事件調查，以及產生解決事件的緩解計劃。
- 事件預防 – 在預防頁面中，您可以在這裡找到改善可觀測性狀態、交付程序和基礎設施架構以防止未來事件的建議。
- 拓撲 – 拓撲頁面提供帳戶資源及其在連線帳戶中所有資源間關係的互動式視覺化呈現。您可以使用「顯示」下拉式清單，在系統、容器和資源檢視之間切換，以不同層級的詳細資訊檢視拓撲。
- Skills – 模組化指令集，可延伸具有特殊功能的 AWS DevOps Agent。技能包含領域知識、調查方法和為您的基礎設施量身打造的工具組態。每項技能都會啟用特定工具，並僅在與調查相關時提供指示的漸進式揭露。
- 自然語言聊天界面 – 聊天功能是 AI 支援的對話式助理，可讓您查詢基礎設施、分析系統運作狀態，並使用自然語言進行調查。聊天功能會根據您正在檢視的頁面提供內容感知回應。

身分驗證

AWS DevOps 代理程式支援靈活的身分驗證方法，以適應不同的組織需求：

- IAM Identity Center 整合 （ 使用者存取 ） – Organizations 可以使用 AWS Identity Center (IAM Identity Center) 集中管理使用者對 DevOps Agent Space Web 應用程式的存取。IAM Identity

Center 可以透過標準 OIDC 和 SAML 通訊協定與外部身分提供者聯合，包括 Okta、Ping Identity 和 Microsoft Entra ID 等提供者。此方法支援來自身分提供者的多重驗證。

- 外部身分提供者 (IdP) 身分驗證 – Organizations 可以將 Okta 或 Microsoft Entra ID 等 OIDC 相容身分提供者直接連線至 Agent Space Web 應用程式，而不需要 IAM Identity Center。使用者透過 IdP 使用其公司登入資料登入。如需設定說明，請參閱 the section called “設定外部身分提供者 (IdP) 身分驗證”。
- IAM 身分驗證連結（管理員存取）– 一種替代方法可讓您使用現有的主控台工作階段，從 AWS 管理主控台直接存取 Web 應用程式。此選項在實作完整的 Identity Center 整合之前很有用，但工作階段限制為 10 分鐘。

什麼是 DevOps Agent Spaces？

DevOps Agent Space 是邏輯容器，可定義 AWS DevOps Agent 可存取的工具和基礎設施。每個 Agent Space 會使用自己的 AWS 帳戶存取、第三方整合和使用者許可獨立運作。

Agent Space 代表在事件回應期間可以存取和調查的 What AWS DevOps 代理程式界限。當您建立 代理程式空間時，您可以定義代理程式可以存取哪些 AWS 帳戶、可以連線哪些外部工具，以及組織中哪些使用者可以與代理程式互動。

每個 Agent Space 都做為獨立部署的 AWS DevOps Agent。您可以透過 AWS 管理主控台設定 Agent Space，而您的營運團隊會使用 Agent Space 的 Web 應用程式在該空間內進行調查和檢閱建議。

如何隔離 Agent Spaces

Agent Spaces 會維持隔離，以確保安全性，並防止不同環境或團隊的意外存取：

- AWS 帳戶隔離 – 每個客服人員空間都使用專用 IAM 角色，僅授予特定 AWS 帳戶和資源的存取權。代理程式無法存取代理程式空間明確設定 AWS 的資源。
- 使用者存取隔離 – 您可以控制哪些使用者或群組可以存取每個客服人員空間。這可讓您將存取許可與您的組織結構保持一致，確保團隊僅與其指定的客服人員空間互動。
- 資料隔離 – 調查資料、事件歷史記錄和建議會在每個客服人員空間中個別維護。來自某個客服人員空間的資訊無法顯示，也無法從另一個客服人員空間存取。
- 聊天資料隔離 - 聊天對話歷史記錄也會在每個客服人員空間中隔離。無法從另一個客服人員空間看見或存取一個客服人員空間中的對話和查詢。

客服人員空間 Web 應用程式

每個 Agent Space 都有可在 AWS 管理主控台外部存取的專用 Web 應用程式。請參閱 the section called “什麼是 DevOps Agent Web 應用程式？” 以進一步了解 Web 應用程式。

何時使用多個客服人員空間

考慮建立多個客服人員空間以支援不同的組織需求：

- 團隊區隔 – 為不同的應用程式團隊或業務單位建立專用客服人員空間，以在客服人員空間中維持明確的擁有權界限。
- 環境隔離 – 將生產環境和非生產環境分隔為不同的 Agent Spaces，以防止意外的跨環境存取。
- 服務界限 – 使 Agent Spaces 與特定服務或應用程式界限保持一致，以保持調查的重點和相關性。
- 合規要求 – 使用不同的存取控制或資料駐留設定來設定個別的 Agent Spaces，以符合法規要求。

> **Note**
>
> 建立多個客服人員空間時，您可以使用專用 AWS 帳戶做為客服人員空間的主要帳戶，並將不同的應用程式帳戶連接為次要帳戶。此方法可讓您維持精細的存取控制，同時確保每個客服人員空間只能存取其預期範圍的特定資源，即使使用自動建立角色也是如此。

什麼是 DevOps 代理程式拓撲？

AWS DevOps Agent 會自動探索和視覺化應用程式中的資源和關係，並使用產生的拓撲，在事件調查期間和提出預防性建議時了解您的基礎設施。

拓撲圖表的建立方式

AWS DevOps Agent 透過數個自動化程序建置拓撲圖表：

- 資源探索 – 代理程式會自動掃描 AWS 您的帳戶，以識別屬於您應用程式一部分的運算執行個體、儲存服務、聯網元件和資料庫等資源。
- 關係偵測 – 代理程式會分析組態資料、CloudFormation 堆疊和資源標籤，以判斷資源彼此之間的關係。
- 程式碼和部署映射 – 連線至 CI/CD 管道時，代理程式會將基礎設施資源連結至其部署程序，並變更應用程式和基礎設施程式碼。

- 可觀測性行為映射 – 來自可觀測性系統的資料，例如 Amazon CloudWatch Application Signals 和 Dynatrace，用於識別指出資源之間關係的觀察行為。

關鍵功能

資源映射提供多種功能，可增強事件調查和預防：

- 互動式視覺化 – 透過 Operator Web 應用程式中的互動式圖形探索應用程式拓撲。您可以縮放和導覽拓撲，以了解資源之間的複雜關係。您也可以使用聊天功能查詢使用自然語言的拓撲資訊，例如「顯示所有連接到此 DynamoDB 資料表的 Lambda 函數」或「哪些資源受此警示影響？」。
- 內容調查 – 在事件調查期間， AWS DevOps 代理程式受到資源拓撲的協助，以識別受影響的元件、了解爆量半徑，並透過您的系統追蹤影響路徑。
- 根本原因分析 – 詳細了解資源關係有助於找出問題的來源，即使在具有許多相互依存性的複雜分散式系統中也是如此。
- 影響評估 – 分析事件時，代理程式可以透過識別拓撲中的相依性鏈，更好地判斷哪些下游服務可能受到影響。
- 預防性建議 – 代理程式使用拓撲洞察，針對彈性改善提出有針對性的建議，建議對系統穩定性影響最大的變更。

拓撲檢視

Operator Web 應用程式拓撲頁面中的拓撲視覺化提供多個層級的詳細資訊：

- 已學習 – 從客服人員空間理解技能產生的預設檢視。顯示由邏輯服務和請求路徑組織之基礎設施的結構化摘要。
- 系統 – 顯示高階帳戶和區域邊界。
- 容器 – 顯示部署堆疊，例如包含相關資源的 CloudFormation 堆疊。
- 元件 – 顯示容器內的個別元件及其關係。
- 所有資源 – 顯示包含所有探索資源及其關係的完整檢視。

資源探索

透過兩種方法探索資源：

- CloudFormation 堆疊 – 代理程式會列出主要 AWS 帳戶和任何連線次要帳戶中的所有 CloudFormation 堆疊及其資源。這支援使用 CloudFormation 進行部署的任何infrastructure-as-code工具，包括 AWS 雲端開發套件 (AWS CDK)。
- Resource Explorer – 對於未從 CloudFormation 部署的資源，會從 AWS Resource Explorer 探索標記的資源。目標 AWS 帳戶必須啟用 Resource Explorer。這有助於識別透過 AWS 管理主控台、AWS 服務 APIs 或其他infrastructure-as-code架構部署之資源的應用程式邊界。

拓撲以外的調查範圍

雖然應用程式拓撲在調查期間提供重要的內容， AWS DevOps 代理程式並不僅限於調查拓撲中顯示的資源。代理程式可能會使用其他資料來源，例如 AWS 服務 APIs 或連線的可觀測性工具，來調查不在應用程式拓撲中的資源。

若要限制客服人員可存取的資源，請限制指派給客服人員之角色的政策，以存取跨帳戶資源。如需詳細資訊，請參閱the section called “限制 AWS 帳戶中的客服人員存取”。

拓撲和客服人員空間了解技能

拓撲圖表會饋送至客服人員空間了解學習到的技能，該技能會編碼基礎設施的結構化摘要，以便在調查期間使用。當新的客服人員空間拓撲探索完成時，系統會自動產生客服人員空間理解技能。如需學習技能的詳細資訊，請參閱 the section called “學習的技能”。

DevOps 代理程式技能

AWS DevOps 代理程式技能是模組化的指令集，可根據您的基礎設施和操作工作流程量身打造專業領域知識和調查方法，來擴展代理程式的功能。

什麼是技能

Skills 是包含 Markdown 指示的獨立目錄，可提供專業功能給 AWS DevOps 代理程式。 AWS DevOps 代理程式支援代理程式技能規格的子集，這是封裝代理程式指示和資源的開放標準，僅支援不可執行的文件：Markdown 指示、PDFs、影像和資料檔案。

每項技能都需要一個 SKILL.md 檔案，其中包含您要為 AWS DevOps 代理程式提供的指示。除了必要的 SKILL.md 檔案之外，技能還可以包括：

- 特定案例或基礎設施類型的調查工作流程。

- 參考資料，包括架構模式和操作程序。
- 代理程式類型目標 – 技能可以針對特定代理程式類型（通用、隨需、事件分類、事件 RCA、事件緩解、評估），以減少內容消耗並改善代理程式焦點。

為什麼要使用 Skills

技能將一般用途助理中的 transform AWS DevOps 代理程式轉換為基礎設施和操作工作流程的專家。與聊天訊息中提供的一次性指示不同，技能是可重複使用的功能，可在與 AWS DevOps 代理程式執行的任務相關時自動載入。

主要優點：

- 專用於您的代理程式 – Tailor AWS DevOps 代理程式，其中包含調查程序、最佳實務，以及專屬於基礎設施和營運模式的組織知識。
- 減少重複 – 建立一次調查工作流程， AWS DevOps 代理程式會在所有相關調查中自動使用這些工作流程，無需重複提供相同的指引。
- 編寫功能 – 結合多種技能來建置end-to-end調查工作流程。 AWS DevOps 代理程式會在執行期間讀取多種技能，例如從您的自訂 CI/CD 管道擷取部署的技能，以及搜尋程式碼儲存庫的技能。
- Amplify 自訂工具 – 建立技能，以有效地使用您的自訂 MCP 伺服器工具在 中引導 AWS DevOps 代理程式。技能可以記錄何時叫用特定工具、用於不同案例的參數，以及如何解譯結果以完成基礎設施特定的工作流程。

技能的運作方式

當 AWS DevOps 代理程式遇到相關任務時，它會載入適當的技能並遵循指示來引導其調查。例如，「資料庫效能調查」技能可能包括step-by-step程序，讓代理程式能夠有系統地檢查警示狀態、分析連線指標，以及識別慢速查詢。

技能結構

技能會組織為目錄，其中包含：

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

SKILL.md

SKILL.md 是唯一的強制性檔案。它包含以 Markdown 格式撰寫的核心指示。此檔案應該：

- 描述何時及如何使用技能。
- 提供step-by-step調查程序。
- 包含不同案例的決策樹。
- 記錄預期的輸出和成功條件。

Frontmatter

Frontmatter 是SKILL.md檔案頂端的中繼資料區塊，包含在---分隔符號之間。它包含 AWS DevOps 代理程式用來判斷在調查或任務期間何時啟用技能的 name和 description 欄位。

```
---
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 – 技能的唯一識別符。僅使用小寫字母、數字和連字號（最多 64 個字元）。不得以連字號開頭或結尾。

描述 – 詳細說明何時和為何 AWS DevOps 代理程式應該使用此 Skill。 AWS DevOps 代理程式會評估此欄位，以決定技能是否與目前的任務相關。即使指示撰寫良好，模糊或遺漏的描述也可能導致客服人員完全略過技能。

重要 – 從客服人員的角度撰寫描述。包含應觸發技能的特定案例、服務、錯誤類型或症狀。例如，「在調查 Amazon RDS 執行個體的資料庫延遲、連線錯誤或查詢逾時時使用此技能」比「RDS 技能」更有效。

當您在 UI 中建立技能時，系統會自動從您提供的名稱和描述產生前綴。上傳為 zip 檔案的技能必須在 SKILL.md 檔案中包含前綴。

範例：完成技能

下列範例顯示調查 RDS 效能問題的完整、格式良好的技能。它示範了目錄結構、https：//SKILL.md 前綴、可執行的調查程序和補充參考檔案。

目錄結構：

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

https：//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
```

參考/rds-metrics-reference.md：// ：

```
# RDS CloudWatch Metrics Reference


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

建立技能

在建立技能之前，您必須擁有 代理程式空間。如需詳細資訊，請參閱the section called “建立 代理程式空間”。

您可以根據您的工作流程偏好設定和技能複雜性，以兩種方式建立技能：

在 UI 中建立技能

在 AWS DevOps Agent Operator Web 應用程式中建立的技能在單一 SKILL.md 檔案中包含名稱、描述和指示。

若要在 UI 中建立技能：

- 導覽至 Agent Space Operator Web 應用程式中的技能頁面。

- 按一下「新增技能」。
- 從模態中選取「建立技能」。
- 填寫技能表單：
  - 名稱 – 僅限小寫字母、數字和連字號（最多 64 個字元）。不得以連字號開頭或結尾。範例：`rds-throttling-investigation`
  - 描述 – 簡短說明何時使用此技能（建議最少 100 個字元，最多 1，024 個字元）。這有助於客服人員判斷何時啟用技能。
  - 狀態 – 設定為作用中（預設）或非作用中。客服人員不會使用非作用中技能。
  - 客服人員類型 – 選取一個或多個可使用此技能的客服人員類型。依預設會選取一般，讓所有客服人員類型都能使用這項技能。若要鎖定特定客服人員，請取消選取一般，然後從中選擇：隨需、事件分類、事件 RCA、事件緩解或評估。
  - 說明 – Markdown 格式的Step-by-step程序。具體且可行。
- 按一下「建立」以儲存技能。

系統會自動產生具有適當前端結構的 SKILL.md 檔案。

若要編輯在 UI 中建立的技能：

- 導覽至技能清單中的技能，然後按一下技能將其開啟。
- 按一下 Edit (編輯)。
- 修改名稱、描述或指示。
- 按一下儲存以更新技能。

上傳技能

上傳為 zip 檔案的技能包含 SKILL.md 檔案和其他資源，例如參考資料或資產。

技能結構：

```
my-skill.zip
### SKILL.md              # Required: main skill instructions
### references/          # Optional: reference documentation
#   ### architecture.md
#   ### troubleshooting.md
### assets/              # Optional: images, diagrams, data files
    ### topology.png
```

```
    ### metrics.csv
```

SKILL.md 前端要求：

上傳為 zip 檔案的技能必須在 SKILL.md 中使用 name和 description 欄位包含前綴。 AWS DevOps 代理程式會使用這些欄位來判斷何時啟用技能。如需撰寫有效前綴的詳細資訊，請參閱本主題稍早的前綴章節。

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

若要透過 zip 上傳建立技能：

- 依照上述結構，使用技能檔案建立目錄。
- 確保 SKILL.md 包含適當的前綴（名稱和描述）。
- 將目錄壓縮為 .zip 檔案。
- 導覽至 Agent Space Operator Web 應用程式中的技能頁面。
- 按一下「新增技能」。
- 從模態中選取「上傳技能」。
- 拖放您的 .zip 檔案或按一下以瀏覽（僅限 ZIP 檔案，最多 6 MB)。
- 選取一個或多個可使用此技能的代理程式類型（依預設會選取通用，並套用到所有代理程式類型；取消選取以鎖定隨需、事件分類、事件 RCA、事件緩解或評估）。
- 檢閱 zip 檔案需求和驗證結果。
- 按一下「上傳」，將技能新增至您的客服人員空間。

上傳為 zip 檔案之技能的重要限制：

- 目前不支援指令碼 – 包含 `scripts/`目錄中指令碼的技能會在上傳期間遭到拒絕。一旦代理程式可以存取安全編碼環境，就會在未來版本中啟用指令碼執行。
- 大小限制 – zip 檔案大小總計不得超過 6 MB （包括所有檔案）。
- 需要 SKILL.md – zip 檔案必須包含具有有效前端的 SKILL.md 檔案。

命名技能的最佳實務：

使用清晰的描述性名稱，例如「rds-throttling-investigation」，而不是通用名稱。良好的技能名稱會反映其處理的特定案例或服務，讓您更輕鬆地一目了然地識別正確的技能。

管理技能

AWS DevOps Agent 透過 Operator Web 應用程式提供全方位的技能管理功能：

列出技能 – 檢視客服人員空間中的所有技能。技能頁面會顯示技能名稱、作用中或非作用中狀態、建立日期、上次更新日期，以及可用的動作。

檢視技能 – 按一下任何技能以查看其詳細資訊檢視。在 UI 中建立的技能會顯示可編輯的內容，您可以在其中直接在 UI 中修改名稱、描述或指示，然後按一下「儲存」進行更新。上傳為 zip 檔案的技能會顯示檔案樹狀目錄，其中顯示 SKILL.md 和參考/ 和資產/ 等任何其他目錄。按一下樹狀目錄中的檔案，以唯讀模式檢視其內容。

選取技能的客服人員 – 設定哪些客服人員類型可以在建立或編輯時使用每項技能。在客服人員類型下拉式清單中，使用核取方塊選取一或多個客服人員類型：一般（預設 - 適用於所有客服人員類型）、隨需（整合查詢）、事件分類（初始事件評估）、事件 RCA（根本原因分析）、事件緩解（自動化事件回應）或評估（主動建議）。依預設會選取一般，讓所有客服人員類型都能使用這項技能。以特定代理程式為目標的技能可減少內容消耗並改善代理程式焦點。

啟用和停用技能 – 暫時停用技能，而不使用作用中/非作用中切換將其刪除。開啟技能詳細資訊檢視並切換切換到「非作用中」，以防止代理程式載入它以進行新的調查，同時保留所有內容和組態。進行中的調查會繼續使用技能。切換回「作用中」，讓技能立即再次可用。

更新技能 – 根據現有技能的建立方式來修改現有技能。對於在 UI 中建立的技能，請按一下技能詳細資訊檢視中的「編輯」、修改名稱、描述或指示，然後按一下「儲存」進行更新。對於上傳為 zip 檔案的技能，請在本機修改檔案、建立新的 zip 檔案，以及上傳新版本。

刪除技能 – 從客服人員空間永久移除技能。開啟技能清單檢視，按一下更多選項選單 (:)，然後選取「刪除」，檢閱永久刪除的相關警告，輸入要確認的技能名稱，然後按一下「刪除技能」。刪除無法復原。如果嘗試載入已刪除的技能，進行中的調查可能會受到影響。對於上傳為 zip 檔案的技能，請先下

載 zip 檔案，然後再刪除 做為備份。如果您可能需要再次刪除技能，請考慮停用技能，而不是將其刪除。

從 Runbook 遷移

現有的 Runbook 會自動遷移至 Skills，而不需要客戶動作。當您的客服人員空間轉換為技能模型時，所有 Runbook 都會轉換為技能，並顯示在您的技能使用者介面中。遷移後，您可以：

- 檢閱遷移的技能 – 檢查自動遷移是否已正確轉換您的 Runbook。
- 視需要更新 – 直接在 UI 中編輯技能，以精簡指示、更新描述或設定代理程式類型目標。
- 使用參考展開 – 對於可能受益於其他參考資料或架構圖的技能，請使用參考/ 或資產/ 目錄將其重新建立為 zip 上傳技能。
- 建立新的技能：為 Runbook 先前未涵蓋的調查工作流程新增技能。

如果您在自動遷移的技能方面遇到任何問題，或需要遷移後更新方面的協助，請聯絡 AWS Support。

學習的技能

什麼是學到的技能？

學到的技能是 DevOps 代理程式從您的代理程式空間資料產生的結構化知識檔案。每個學到的技能都會編碼 AWS DevOps 代理程式在執行任務時所使用的特定知識類型。在啟動時，有兩個學習到的技能可用：客服人員空間理解和工具使用最佳實務。

客服人員空間了解

Agent Space Understanding 技能 (`understanding-agent-space`) 會分析您連線的雲端帳戶、程式碼儲存庫和遙測整合，以在 Agent Space 中建置資源和關係的映射。

技能會產生主要`SKILL.md`檔案和一組參考檔案。主要檔案包含純語言系統概觀，其中包含關鍵網域概念、部署環境 (AWS 帳戶和區域對、Azure 訂閱和區域等）、容器層級架構圖，顯示邏輯服務如何連線、應用程式的核心請求路徑及其周遊的元件，以及程式碼儲存庫與容器的映射。

每個邏輯容器都會收到一個專用參考檔案，描述其內部元件（運算、資料、傳訊、網路等），其中包含資源類型和實體識別符，例如 ARNs、資料表名稱和佇列 URLs。參考檔案也會擷取可觀測性涵蓋範圍，包括連結至每個元件的警示、儀表板和監視器。它也會將每個元件映射至其相關聯的程式碼儲存庫、套件和infrastructure-as-code定義，提供從原始程式碼到部署資源的完整可追蹤性鏈。

每個關鍵請求路徑都會收到一個專用參考檔案，描述從進入點到每個中繼服務、資料存放區和外部相依性的完整end-to-end請求流程。檔案包含排序流程圖，其中顯示元件之間的操作順序和互動機制，以及每位參與者的責任。它還會為與路徑相關的可觀測性訊號編製目錄：每個躍點的日誌群組模式、關鍵指標（延遲、錯誤率、限流、字符配額）及其警示名稱和維度，以及可在服務和帳戶之間建立關聯的分散式追蹤範圍。

工具使用最佳實務

工具使用最佳實務技能會分析過去的調查工具，用來擷取有效的使用模式、常見失敗模式和參數指導。這有助於 DevOps 代理程式避免已知的陷阱，並以較少的浪費步驟執行調查。技能會產生主要檔案和一組每個工具的參考檔案。主要檔案做為路由索引，列出每個工具及其支援的調查案例，以及對應參考檔案的連結。

每個工具參考檔案最多可包含三個區段：

- 最佳實務 — 從成功工具使用中擷取的調查驅動技術，例如 CloudWatch Logs Insights 查詢範本、環境特定指標命名空間和維度，以及 CloudTrail 事件來源篩選條件。每個項目都圍繞調查案例進行組織，並包含在過去調查中觀察到的具體參數值和範例。
- 常見錯誤 — 重複失敗模式及其修正。每個項目描述特定的錯誤條件，例如查詢無法存取的帳戶或建構格式錯誤的彙總查詢，並提供修正動作，讓代理程式可以避免錯誤或從錯誤中復原，而不會浪費調查步驟。
- 輸出管理 — 容易傳回大型回應的工具呼叫指導。每個項目描述參數變更或處理策略，以減少輸出大小，同時保留診斷值。

當即時基礎設施存取可用時，技能會先針對您的環境驗證模式，再納入它們。確認的模式是以可信度表示，未確認的模式使用謹慎的語言，並排除拒絕的模式。這可讓技能與基礎設施的目前狀態保持一致。

管理學到的技能

更新：DevOps 代理程式會根據您代理程式空間中的活動，自動產生和更新學到的技能。以下說明每個技能的更新時間。

DevOps 代理程式每 30 次調查會產生更新的 工具使用最佳實務技能。

Agent Space Understanding 技能是由學習代理程式產生，在您新增、更新或移除 Agent Space 功能或整合時執行。

若要手動重新產生學習的技能，請選擇運算子應用程式拓撲頁面上的重新產生按鈕，或與客服人員聊天並要求其更新學習的技能。

停用 — 根據預設，學習的技能處於作用中狀態。作用中時，DevOps 代理程式會在每個 DevOps 代理程式任務開始時載入它們。若要停止套用學到的技能，請從運算子應用程式中的技能檢視器停用它。停用技能不會將其刪除。技能會保留，而且可以隨時重新啟用。停用技能時，DevOps 代理程式會在該技能不知情的情況下運作。

拓撲檢視 — Agent Space 的 Web 應用程式中的拓撲頁面使用 Agent Space Understanding Skill，以視覺化方式將您的 Agent Space 環境顯示為邏輯容器和元件。按一下任何容器以查看其元件、資源識別符和遙測。

支援的區域

本主題說明您可以使用 AWS DevOps 代理程式 AWS 的區域。如需 AWS 區域的詳細資訊，請參閱《帳戶管理參考指南》中的指定 AWS 您的帳戶可以使用的區域。 AWS

跨區域資源監控

AWS DevOps 代理程式可以監控和調查位於任何 AWS 區域的 AWS 帳戶中的資源，無論您在哪個支援的區域建立代理程式空間。當您將 AWS 帳戶與 代理程式空間建立關聯時，代理程式會探索並映射該帳戶內所有區域中的資源。這表示您在工作負載執行的每個區域中都不需要 代理程式空間。

根據您偏好的資料落地、與營運團隊的距離或組織需求，選擇支援的 區域。

支援的區域

AWS DevOps 代理程式可在下列 AWS 區域使用。

區域名稱	區域代碼	主控台連結
美國東部 (維吉尼亞北部)	us-east-1	開啟主控台
美國西部 (奧勒岡)	us-west-2	開啟主控台
亞太地區 (悉尼)	ap-southeast-2	開啟主控台
亞太區域 (東京)	ap-northeast-1	開啟主控台
歐洲 (法蘭克福)	eu-central-1	開啟主控台
歐洲 (愛爾蘭)	eu-west-1	開啟主控台

服務端點

區域名稱	區域代碼	Endpoint	通訊協定
美國東部 (維吉尼亞北部)	us-east-1	aidevops.us-east-1.amazonaws.com	HTTPS
美國西部 (奧勒岡)	us-west-2	aidevops.us-west-2.amazonaws.com	HTTPS
亞太地區 (悉尼)	ap-southeast-2	aidevops.ap-southeast-2.amazonaws.com	HTTPS
亞太地區 (東京)	ap-northeast-1	aidevops.ap-northeast-1.amazonaws.com	HTTPS
歐洲 (法蘭克福)	eu-central-1	aidevops.eu-central-1.amazonaws.com	HTTPS
歐洲 (愛爾蘭)	eu-west-1	aidevops.eu-west-1.amazonaws.com	HTTPS

考量事項

- 客服人員空間區域選擇 — 客服人員空間及其資料（調查、

拓撲、建議）會存放在您建立它的區域。選擇符合您資料落地需求的區域。

- 跨區域監控 — 與客服人員相關聯的 AWS 帳戶中的資源

無論這些資源部署在哪個區域，空間都會受到監控。您不需要在工作負載執行的每個區域中建立個別的 Agent Spaces。

- 第三方整合 — 與 CI/CD 提供者的連線 (GitHub、GitLab)、

可觀測性工具 (Dynatrace、Datadog、New Relic、Splunk) 和 MCP 伺服器是依代理程式空間設定，與區域無關。

開始使用 AWS DevOps 代理程式

在本入門指南中，您將建立基本的客服人員空間、設定最少的許可，以及執行您的第一個 AI 型調查。

主題：

建立 代理程式空間

Agent Space 定義了 AWS DevOps Agent 可存取的工具和基礎設施。本指南會逐步引導您建立 Agent Space、設定主要帳戶存取，以及啟用 DevOps Agent Web 應用程式。請參閱「什麼是客服人員空間」以進一步了解客服人員空間概念。

建立 代理程式空間

存取 AWS DevOps 代理程式主控台

1. 登入 AWS 管理主控台
2. 導覽至 AWS DevOps 代理程式主控台

為客服人員空間命名

1. 按一下建立客服人員空間

在客服人員空間詳細資訊區段中，提供：

1. 在名稱欄位中，輸入客服人員空間的名稱
2. （選用）在描述欄位中，新增有關客服人員空間用途的詳細資訊

3. （選用）從客服人員回應語言下拉式清單中，選取客服人員在產生回應、調查結果和調查輸出時所使用的語言。選項包括：印尼文、中文（簡體/PRC)、中文（繁體/台灣）、英文（英國）、法文（法國）、德文（德國）、義大利文（義大利）、日文（日本）、韓文（韓國）、葡萄牙文（巴西）、西班牙文（拉丁美洲）、土耳其文（土耳其）、阿拉伯文（沙烏地阿拉伯）、泰文（泰國）和越南文（越南）。如果未選取語言，代理程式會以輸入的語言回應。

設定主要帳戶存取

在授予此代理程式空間 AWS 資源存取權區段中，您將設定 IAM 角色，以授予代理程式空間對主要 AWS 帳戶的存取權。主要帳戶是您建立 Agent Space AWS 的帳戶。 AWS DevOps Agent 需要 IAM 角色，才能在調查期間探索和存取此帳戶中 AWS 的資源。

選擇角色組態方法。 選取以下其中一個選項：

選項 1：自動建立新的 AWS DevOps 代理程式角色（建議）

此選項會自動建立具有適當許可的角色，讓 AWS DevOps 代理程式調查您帳戶中的資源。

> **Note**
>
> 您必須擁有 IAM 許可才能建立新的角色，才能使用此選項。

1. 選取自動建立新的 AWS DevOps 代理程式角色
2. （選用）更新要建立的客服人員空間角色名稱

選項 2：指派現有角色

當其他管理員先前已建立專門針對 AWS DevOps 代理程式的角色時，請使用此選項。

1. 選取指派現有角色
2. 從下拉式選單中選取具有適當許可的現有角色

選項 3：使用政策範本建立新的 AWS DevOps 代理程式角色

當您需要限制客服人員可在主要帳戶中存取的服務和資源時，請使用此選項。

1. 選取使用政策範本建立新的 AWS DevOps 代理程式角色
2. 依照指示建立新角色的信任政策和內嵌政策。

啟用 Agent Space Web 應用程式

Web 應用程式可讓人員與 AWS DevOps Agent 互動，以進行事件調查和檢閱建議。See AWS DevOps 代理程式主控台架構 【連結】 以進一步了解。啟用時，使用者可以從 AWS 管理主控台透過 IAM 身分驗證連結存取 Agent Space Web 應用程式。

選取以下其中一個選項：

選項 1：自動建立新的 AWS DevOps 代理程式角色 （建議）

此選項會自動建立具有適當許可的角色，以存取 DevOps Agent Web 應用程式。

> **Note**
> 您必須擁有 IAM 許可才能建立新的角色，才能使用此選項。

1. 選取自動建立新的 AWS DevOps 代理程式角色
2. 檢閱將授予角色的許可

選項 2：指派現有角色

當其他管理員先前已建立 運算子角色時，請使用此選項。

1. 選取指派現有角色
2. 從下拉式選單中選取具有適當許可的現有角色

選項 3：使用政策範本建立新的 AWS DevOps 代理程式角色

當您需要自訂 Web 應用程式存取的許可時，請使用此選項。

1. 選取使用政策範本建立新的 AWS DevOps 代理程式角色
2. 依照指示建立新角色的信任政策和內嵌政策。

新增標籤 （選用）

您可以在建立期間將 AWS 標籤新增至 代理程式空間。標籤是索引鍵/值對，可協助您整理和識別資源。每個客服人員空間最多可以新增 50 個標籤。若要新增標籤，請展開建立客服人員空間頁面上的標籤區段，然後按一下新增標籤。

完成客服人員空間建立

填寫所有區段後，按一下建立

驗證您的客服人員空間設定

設定完成後，運算子存取按鈕會出現在客服人員空間詳細資訊頁面上。按一下它會在新標籤中開啟 Web 應用程式，並成功驗證。

後續步驟

設定您的客服人員空間後，請考慮這些後續步驟：

- 如果您的應用程式跨越多個帳戶，請新增次要 AWS 帳戶
- 設定第三方整合，例如可觀測性工具或票證系統
- 設定生產環境的 AWS Identity Center 身分驗證
- 探索您的應用程式資源映射，以協助 AWS DevOps 代理程式了解您的基礎設施

AWS DevOps Agent CLI 入門指南

概觀

透過 AWS DevOps 代理程式，您可以監控和管理 AWS 基礎設施。本指南會逐步引導您使用 AWS 命令列界面 (AWS CLI) 來設定 up AWS DevOps 代理程式。您可以建立 IAM 角色、設定客服人員空間，以及建立 AWS 帳戶關聯。您也可以啟用 運算子應用程式，並選擇性地連接第三方整合。本指南大約需要 20 分鐘才能完成。

AWS DevOps 代理程式適用於六個 AWS 區域：美國東部（維吉尼亞北部）、美國西部（奧勒岡）、亞太區域（雪梨）、亞太區域（東京）、歐洲（法蘭克福）和歐洲（愛爾蘭）。如需支援區域的詳細資訊，請參閱 the section called “支援的區域”。

先決條件

開始前，請確定您有下列項目：

- AWS 已安裝和設定的 CLI 第 2 版
- 對您的 AWS 監控帳戶進行身分驗證

- 建立 AWS Identity and Access Management (IAM) 角色並連接政策的許可
- 用作監控 AWS 帳戶的帳戶
- 熟悉 AWS CLI 和 JSON 語法

在本指南中，將下列預留位置值取代為您自己的預留位置值：

- <MONITORING_ACCOUNT_ID> — 監控（主要）AWS 帳戶的 12 位數帳戶 ID
- <EXTERNAL_ACCOUNT_ID> — 要監控的次要 AWS 帳戶的 12 位數帳戶 ID（用於步驟 4）
- <REGION> — 代理程式空間的區域 AWS 代碼（例如 us-east-1或 eu-central-1）
- <AGENT_SPACE_ID> — create-agent-space命令傳回的代理程式空間識別符

IAM 角色設定

1. 建立 DevOps Agent 空間角色

執行下列命令來建立 IAM 信任政策：

```
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
```

建立 IAM 角色：

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

執行下列命令來儲存角色 ARN：

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

連接 AWS 受管政策：

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

建立並連接內嵌政策，以允許建立 Resource Explorer 服務連結角色：

```
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. 建立運算子應用程式 IAM 角色

執行下列命令來建立 IAM 信任政策：

```
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
```

建立 IAM 角色：

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

執行下列命令來儲存角色 ARN：

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

連接 AWS 受管運算子應用程式政策：

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

此受管政策授予 運算子應用程式存取代理程式空間功能的許可。這些功能包括調查、建議、知識管理、聊天和 AWS 支援整合。政策會使用 `aws:PrincipalTag/AgentSpaceId`條件來限制對特定代理程式空間的存取。如需動作完整清單的詳細資訊，請參閱 the section called “DevOps Agent IAM 許可”。

加入步驟

1. 建立客服人員空間

執行下列命令來建立代理程式空間：

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

或者，指定 `--kms-key-arn`使用客戶受管 AWS KMS 金鑰進行加密。您也可以使用 `--tags` 來新增資源標籤`--locale`，以及設定客服人員回應的語言。

`agentSpaceId` 從回應儲存（位於 `agentSpace.agentSpaceId`)。

若要稍後列出您的代理程式空間，請執行下列命令：

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

2. 關聯 AWS 您的帳戶

關聯 AWS 您的帳戶以開啟拓撲探索。將 `accountType`設定為下列其中一個值：

- `monitor` — 代理程式空間存在的主要帳戶。此帳戶託管代理程式，並用於拓撲探索。
- `source` — 代理程式監控的其他帳戶。當您在步驟 4 中關聯外部帳戶時，請使用此類型。

```
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. 啟用 運算子應用程式

身分驗證流程可以使用 IAM、IAM Identity Center (IDC) 或外部身分提供者 (IdP)。執行下列命令，為您的代理程式空間啟用 運算子應用程式：

```
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>
```

對於 IAM Identity Center 身分驗證，請使用 `--auth-flow idc`並提供 `--idc-instance-arn`。對於外部身分提供者，請使用 `--auth-flow idp`並提供 `--issuer-url`、 `--idp-client-id`和 `--idp-client-secret`。如需詳細資訊，請參閱the section called “設定 IAM Identity Center 身分驗證”及the section called “設定外部身分提供者 (IdP) 身分驗證”。

> 注意：如果您先前為帳戶中的另一個客服人員空間建立 運算子應用程式角色，則可以重複使用該角色 ARN。

4. （選用） 關聯其他來源帳戶

若要使用 AWS DevOps 代理程式監控其他帳戶，請建立 IAM 跨帳戶角色。

在外部帳戶中建立跨帳戶角色

切換到外部帳戶並建立信任政策。MONITORING_ACCOUNT_ID 是主帳戶，託管您在步驟 2 中設定的代理程式空間。此組態可讓 AWS DevOps Agent 服務代表監控帳戶擔任次要來源帳戶中的角色。

執行下列命令來建立信任政策：

```
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
```

建立跨帳戶 IAM 角色：

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

執行下列命令來儲存角色 ARN：

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

連接 AWS 受管政策：

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

連接內嵌政策，以允許在外部帳戶中建立 Resource Explorer 服務連結角色：

```
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
```

關聯外部帳戶

切換回您的監控帳戶，然後執行下列命令來建立外部帳戶的關聯：

```
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.（選用）關聯 GitHub

> 注意：您必須先使用 OAuth 流程透過 AWS DevOps 代理程式主控台註冊 GitHub，才能透過 CLI 建立關聯。

如需透過主控台註冊 GitHub 的說明，請參閱 the section called “連線至 CI/CD 管道”。

列出已註冊的服務：

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

儲存 <SERVICE_ID> for serviceType：github。

在主控台中註冊 GitHub 之後，請執行下列命令來建立 GitHub 儲存庫的關聯：

```
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.（選用）註冊和關聯 ServiceNow

首先，使用 OAuth 登入資料註冊 ServiceNow 服務：

```
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>
```

儲存傳回的 <SERVICE_ID>，然後與 ServiceNow 建立關聯：

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

7.（選用）註冊和關聯 Dynatrace

首先，使用 OAuth 登入資料註冊 Dynatrace 服務：

```
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>
```

儲存傳回的 <SERVICE_ID>，然後與 Dynatrace 建立關聯。資源是選用的。環境會指定要與哪個 Dynatrace 環境建立關聯。

```
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>
```

回應包含用於整合的 Webhook 資訊。您可以使用此 Webhook 來觸發來自 Dynatrace 的調查。如需詳細資訊，請參閱the section called “連接 Dynatrace”。

8.（選用）註冊和關聯 Splunk

首先，使用 BearerToken 登入資料註冊 Splunk 服務。

端點使用下列格式： `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>
```

儲存傳回的 <SERVICE_ID>，然後與 Splunk 建立關聯：

```
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>
```

回應包含用於整合的 Webhook 資訊。您可以使用此 Webhook 來觸發 Splunk 的調查。如需詳細資訊，請參閱the section called “連接 Splunk”。

9.（選用）註冊和關聯新複本

首先，使用 API 金鑰登入資料註冊 New Relic 服務。

區域： US或 EU。

選用欄位：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>
```

儲存傳回的 <SERVICE_ID>，然後關聯新複本：

```
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>
```

回應包含用於整合的 Webhook 資訊。您可以使用此 Webhook 觸發來自 New Relic 的調查。如需詳細資訊，請參閱the section called “連接新複本”。

10.（選用）註冊和關聯 Datadog

您必須先使用 OAuth 流程透過 AWS DevOps Agent 主控台註冊 Datadog，才能透過 CLI 建立關聯。如需詳細資訊，請參閱the section called “連接 DataDog”。

列出已註冊的服務：

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

儲存 <SERVICE_ID> for serviceType：`mcpserverdatadog`。

然後建立 Datadog 的關聯：

```
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>
```

回應包含用於整合的 Webhook 資訊。您可以使用此 Webhook 來觸發 Datadog 的調查。如需詳細資訊，請參閱the section called “連接 DataDog”。

11.（選用）刪除代理程式空間

刪除代理程式空間會移除該代理程式空間的所有關聯、組態和調查資料。此動作無法復原。

若要刪除代理程式空間，請執行下列命令：

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

驗證

若要驗證您的設定，請執行下列命令：

```
# 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>
```

後續步驟

- 若要連接其他整合，請參閱 設定適用於 AWS DevOps 代理程式的功能。
- 若要了解客服人員的技能和功能，請參閱 the section called “DevOps 代理程式技能”。
- 若要了解 運算子 Web 應用程式，請參閱 the section called “什麼是 DevOps Agent Web 應用程式？”。

備註

- 將 <AGENT_SPACE_ID>、<MONITORING_ACCOUNT_ID>、<REGION>、<EXTERNAL_ACCOUNT_ID>等取代為您的實際值。

- 如需支援的區域的清單，請參閱 the section called “支援的區域”。

建立測試環境

本指南提供實作測試，以使用範例架構來驗證 AWS DevOps 代理程式的事件回應功能。如果您想要在連接生產系統之前測試 DevOps Agent，請使用此增補。

先決條件

- AWS 具有管理存取權的帳戶
- 使用 Auto create AWS DevOps Agent 角色流程建立和設定的 DevOps Agent Space

成本與安全性概觀

成本保護

- EC2 測試：免費 (AWS 免費方案） 或 ~$0.02 2 小時
- Lambda 測試：免費 （每月 100 萬個請求免費方案）
- CloudWatch：免費 （包含 10 個警示、基本指標）
- 預期總成本：完成測試為 0.00 - 0.05 美元

這些測試中的安全功能

- 自動終止：內建自動關閉
- 符合免費方案資格：使用最小執行個體類型
- 限制範圍：最小、隔離的測試資源
- 輕鬆清除：移除所有項目的簡單主控台步驟
- 不影響生產：完全獨立的測試環境

設定 AWS 您的帳戶進行測試

> **Important**
>
> 基礎設施資源需要部署在您 AWS 建立 DevOps Agent Space 主要雲端帳戶的帳戶。特定區域並不重要。

1. 登入 AWS 主控台：https：//https://console.aws.amazon.com
2. 確定您在 DevOps Agent Space 所在的相同 AWS 帳戶中工作
3. 您可以針對測試資源使用任何區域

> **Note**
>
> 您的 DevOps 代理程式主要帳戶與您建立的測試環境資源之間的 1：1 映射可簡化測試設定。您可以輕鬆擴展 DevOps Agent Space，以包含次要帳戶並啟用跨帳戶調查。

選擇您的測試

您可以獨立執行測試或同時執行兩者：

測試選項 A：EC2 CPU 容量測試

目的：Verify AWS DevOps 代理程式偵測和調查 EC2 效能問題的能力

預估時間：5 分鐘設定 + 10 分鐘自動執行

難度：完全自動化（不需要手動步驟）

測試選項 B：Lambda 錯誤率測試

目的：Verify AWS DevOps 代理程式偵測和調查 Lambda 函數錯誤的能力

預估時間：10 分鐘設定 + 2 分鐘觸發

難度：非常簡單

測試選項 A：EC2 CPU 容量測試

步驟 1：為 EC2 測試部署 CloudFormation 堆疊

我們將使用 CloudFormation 來建立測試資源，這可讓 AWS DevOps 代理程式正確追蹤和調查這些資源。

1. 導覽至 CloudFormation：
   a. 在 AWS 主控台中搜尋「CloudFormation」，然後按一下 CloudFormation
   b. 按一下使用新資源建立堆疊 >（標準）
2. 上傳範本：
   a. 建立新的本機檔案，名為`AWS-DevOpsAgent-ec2-test.yaml`
   b. 將此 CloudFormation 範本複製並貼到 檔案中：
      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. 在 CloudFormation 主控台中，選取上傳範本檔案

d. 按一下選擇檔案

e. 選取 AWS-DevOpsAgent-ec2-test.yaml 檔案

f. 按一下下一步

3. 設定堆疊：

a. 堆疊名稱：AWS-DevOpsAgent-EC2-Test

b. 參數：

i. MyIP：保留為預設值 0.0.0.0/0（如有需要，您可以稍後加以保護）

c. 按一下下一步

4. 設定堆疊選項：

a. 保留預設值，按一下下一步

5. 檢閱和建立：

a. 檢查 我確認 AWS CloudFormation 可能會建立 IAM 資源

b. 按一下提交

6. 等待完成：

a. 堆疊建立需要 3-5 分鐘

b. 狀態將從 `CREATE_IN_PROGRESS`變更為`CREATE_COMPLETE`

c. 重要：您的 EC2 執行個體現在是 AWS DevOpsAgent 可以追蹤的 CloudFormation 堆疊的一部分！

選用：安全 SSH 存取（僅當您計劃連線到執行個體時）

如果您只想執行自動化測試，請略過此步驟

1. 導覽至 EC2 安全群組：

a. 在 AWS 主控台中，前往 EC2 → 安全群組

b. 尋找`AWS-DevOpsAgent-test-sg`

2. 更新 SSH 規則：

a. 選取安全群組 → 傳入規則索引標籤 → 編輯傳入規則

b. 尋找 SSH 規則（連接埠 22)

c. 將來源從 `0.0.0.0/0`變更為您的 IP：`[YOUR_IP]/32`

d. 從 https://whatismyipaddress.com：// 取得您的 IP

e. 按一下儲存規則

步驟 2：等待自動測試執行

1. 自動測試執行：

- CPU 壓力測試會在執行個體啟動後 5 分鐘自動啟動
- 不需要手動介入 - 只需等待，測試會在背景中完全執行

2. 監控測試

- 執行個體會自動開機並準備測試
- 指令碼將執行 5 分鐘，並產生 >70% 的 CPU 用量
- CloudWatch 警示應在總計 8-10 分鐘內觸發 (5 分鐘延遲 + 3-5 分鐘警示）

3. 選用：手動重新執行（用於其他測試）：
   - 連線至您的執行個體：EC2 主控台 →`AWS-DevOpsAgent-Test-Instance`→ Connect → Session Manager
   - 再次執行壓力測試：`./cpu-stress-test.sh`
   - 非常適合多次測試 AWS DevOpsAgent 的回應

測試選項 B：Lambda 錯誤率測試

步驟 1：為 Lambda 測試部署 CloudFormation 堆疊

1. 導覽至 CloudFormation：
   a. 在 AWS 主控台中，前往 CloudFormation
   b. 按一下建立堆疊 → 使用新資源（標準）
2. 上傳範本：
   a. 建立新的本機檔案，名為`AWS-DevOpsAgent-lambda-test.yaml`
   b. 將此 CloudFormation 範本複製並貼到 檔案中：
      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. 在 CloudFormation 主控台中，選取上傳範本檔案

d. 按一下選擇檔案

e. 選取 AWS-DevOpsAgent-lambda-test.yaml 檔案

f. 按一下下一步

3. 設定堆疊：

a. 堆疊名稱：AWS-DevOpsAgent-Lambda-Test

b. 按一下下一步

4. 設定堆疊選項：

a. 保留預設值，按一下下一步

5. 檢閱和建立：

a. 檢查 我確認 AWS CloudFormation 可能會建立 IAM 資源

b. 按一下提交

6. 等待完成：

a. 堆疊建立需要 2-3 分鐘

b. 狀態將變更為`CREATE_COMPLETE`

步驟 2：觸發 Lambda 錯誤

1. 導覽至 Lambda 主控台：

a. 前往 AWS Lambda 主控台

b. 尋找您的 函數`AWS-DevOpsAgent-test-lambda`

2. 測試 函數：

a. 按一下測試索引標籤

b. 按一下建立新事件

c. 事件名稱：`AWS-DevOpsAgent-test-event`

d. 使用此 JSON 承載：

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

e. 按一下儲存

3. 產生錯誤：

a. 按一下測試按鈕 3 次 （每次等待 10 秒）

b. 每個測試都會產生故意錯誤

c. CloudWatch 警示應在 2-3 分鐘內觸發

d. AWS DevOpsAgent 現在應該能夠使用接下來要設定的運算子應用程式中的調查來偵測警示。

Validate AWS DevOps 代理程式偵測

步驟 1：健全檢查 CloudWatch 警示 （選用）

此步驟用於確保上述測試現在處於警示狀態。

對於 EC2 測試：

- 在 CloudWatch 主控台中，前往警示
- 在開始壓力測試後等待 3-5 分鐘
- 您的警示應顯示為警示狀態
- 如果仍然「確定」：再等待 2-3 分鐘 (CloudWatch 指標可能會延遲）

對於 Lambda 測試：

- 檢查`AWS-DevOpsAgent-Lambda-Error-Test`警示
- 應在執行測試後 2-3 分鐘內顯示警示

步驟 2：開始 a AWS DevOps 代理程式調查

1. 開啟 AWS DevOps AgentSpace
2. 按一下管理員存取權。這將在新視窗中開啟 DevOps Agent Space Web 應用程式
3. 按一下畫面右側的開始調查按鈕
4. 填寫下列表單：
   a. 調查詳細資訊：描述您要執行的調查。包含有關調查目標、要探索的領域或相關資訊的任何詳細資訊。
   b. 調查起點：描述您要從中開始調查的資訊。您可以提及警示、指標、日誌程式碼片段或任何其他項目，以便讓 DevOps 代理程式開始運作。在此情況下，請提供您剛建立的警示摘要。
   c. 事件發生的日期和時間（偏好 ISO 8601)：YYYY-MM-DDTHH：MMZ
   d. 為您的調查命名：範例： `Oncall_investigation_1:2025-10-27`
   e. 事件AWS 的帳戶 ID
   f. 事件發生的區域
   g. Priority - AWS DevOpsAgent 允許進行兩次並行調查。Priority 可讓您定義調查的執行順序。
5. 按一下調查以啟動調查。

6. 按一下儀表板中列出的調查。系統會將您導向至調查詳細資訊畫面，您可以在其中檢視 DevOps 代理程式正在採取的精細步驟。

預期結果

EC2 測試結果：

- 偵測 EC2 CPU 警示
- 識別根本原因：「CPU 壓力測試工作負載」
- 顯示時間軸：壓力測試 → CPU 峰值 → 警示
- 提供監控和擴展的建議

Lambda 測試結果：

- 偵測 Lambda 錯誤率峰值
- 識別根本原因：「刻意測試例外狀況」
- 顯示時間軸：函數調用 → 錯誤 → 警示
- 提供錯誤處理和監控的建議

清除指示

清除測試 A (EC2 測試)

自動清除

- 執行個體會在 2 小時後自動終止 （ 建置在 CloudFormation 範本中 ）

手動清除 （ 立即 ）

1. 刪除 CloudFormation 堆疊：
   a. 前往 CloudFormation 主控台
   b. 選取`AWS-DevOpsAgent-EC2-Test`堆疊
   c. 按一下刪除
   d. 確認刪除
   e. 這會自動刪除所有資源：EC2 執行個體、安全群組、金鑰對和 CloudWatch 警示

清除測試 B (Lambda 測試)

1. 刪除 CloudFormation 堆疊：
   a. 前往 CloudFormation 主控台
   b. 選取`AWS-DevOpsAgent-Lambda-Test`堆疊
   c. 按一下刪除
   d. 確認刪除
   e. 這會自動刪除所有資源：Lambda 函數、IAM 角色和 CloudWatch 警示

疑難排解

常見問題

「無法連線至 EC2 執行個體」

- 檢查安全群組：確定您的 IP 已開啟 SSH （連接埠 22)
- 檢查金鑰許可：執行`chmod 400 AWS-DevOpsAgent-test-key.pem`
- 驗證公有 IP：執行個體必須指派公有 IP
- 等待執行個體：確保執行個體處於「執行中」狀態

「警示未觸發」

- 等待指標：CloudWatch 指標可能需要 2-5 分鐘才會顯示
- 檢查 CPU 負載：SSH 到執行個體並執行 `top`以驗證 CPU >70%
- 驗證壓力測試：執行 `ps aux | grep yes`以查看載入程序是否正在執行
- 延長等待時間：第一個警示觸發有時最多需要 7-8 分鐘

測試驗證

Your AWS DevOp 代理程式測試在以下情況成功：

技術驗證

- 調查準確性：EC2 測試的結果應正確指出警示因 CPU 負載而觸發。Lambda 測試的結果應指出這是刻意失敗。

- 時間軸準確度：顯示的正確事件順序
- 建議品質：提供可行的建議

使用 AWS CDK 開始使用 AWS DevOps 代理程式

概觀

本指南說明如何使用 AWS 雲端開發套件 (AWS CDK) 來建立和部署 AWS DevOps 代理程式資源。AWS CDK 應用程式會透過 AWS CloudFormation 自動建立代理程式空間、 AWS Identity and Access Management (IAM) 角色、運算子應用程式和 AWS 帳戶關聯。

AWS CDK 方法透過將所有必要資源定義為基礎設施作為程式碼，自動執行 CLI 入門指南中所述的手動步驟。

AWS DevOps 代理程式可在下列 6 AWS 區域使用：美國東部 （維吉尼亞北部）、美國西部 （奧勒岡）、亞太區域 （雪梨）、亞太區域 （東京）、歐洲 （法蘭克福） 和歐洲 （愛爾蘭）。如需支援區域的詳細資訊，請參閱 the section called “支援的區域”。

先決條件

開始前，請確定您有下列項目：

- AWS 命令列界面 (AWS CLI) 已安裝並使用適當的登入資料設定
- Node.js 第 18 版或更新版本
- AWS 全域安裝的 CDK 命令列界面 (CLI)。若要安裝 AWS CDK CLI，請執行下列命令：

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

- 一個 AWS 帳戶用於監控 （主要） 帳戶
- （選用） 如果您想要設定跨 AWS 帳戶監控的第二個帳戶

本指南涵蓋的內容

本指南分為兩個部分：

- 第 1 部分 — 使用 運算子應用程式和監控帳戶中的 AWS 關聯部署代理程式空間。完成此部分後，代理程式可以監控該帳戶中的問題。

- 第 2 部分（選用）— 新增服務帳戶的來源 AWS 關聯，並將跨帳戶 IAM 角色部署至該帳戶。此組態可讓客服人員空間跨帳戶監控資源。

已建立資源

第 1 部分：DevOpsAgentStack（監控帳戶）

- IAM 角色 (`DevOpsAgentRole-AgentSpace`) — 由 DevOps Agent 服務擔任以監控帳戶。包括 `AIDevOpsAgentAccessPolicy`受管政策和允許建立 Resource Explorer 服務連結角色的內嵌政策。
- IAM 角色 (`DevOpsAgentRole-WebappAdmin`) — 具有客服人員操作`AIDevOpsOperatorAppAccessPolicy`受管政策的操作員應用程式角色。
- 客服人員空間 (`MyCDKAgentSpace`) — 使用 `AWS::DevOpsAgent::AgentSpace` CloudFormation 資源建立的中央客服人員空間。包括運算子應用程式組態。
- 關聯 (AWS 監視器）— 使用 `AWS::DevOpsAgent::Association` CloudFormation 資源將監控帳戶連結至代理程式空間。
- 關聯 (AWS 來源）—（選用）將服務帳戶連結到代理程式空間以進行跨帳戶監控。

第 2 部分：ServiceStack（服務帳戶，選用）

- IAM 角色 (`DevOpsAgentRole-SecondaryAccount`) — 具有固定名稱的跨帳戶角色。受監控帳戶中的代理程式空間信任。包括 `AIDevOpsAgentAccessPolicy`受管政策和允許建立 Resource Explorer 服務連結角色的內嵌政策。
- Lambda 函數 (`echo-service`) — 回應輸入事件的簡單範例服務。

設定

步驟 1：複製範例儲存庫

執行下列命令來複製儲存庫並變更至專案目錄：

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

步驟 2：安裝相依性

執行下列命令來安裝專案相依性：

```
npm install
```

第 1 部分：部署代理程式空間

在本節中，您會在監控帳戶中建立代理程式空間、IAM 角色、運算子應用程式和 AWS 關聯。

步驟 1：設定監控帳戶 ID

開啟`lib/constants.ts`並設定您的監控帳戶 ID：

下列範例顯示要更新的常數：

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

步驟 2：引導 AWS CDK 環境

如果您尚未在監控帳戶中引導 AWS CDK，請執行下列命令：

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

步驟 3：建置和部署

執行下列命令來建置 TypeScript 程式碼並部署堆疊：

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

步驟 4：記錄堆疊輸出

部署完成後， AWS CDK 會列印堆疊輸出。記錄這些值以供日後使用。

下列範例顯示預期的輸出：

```
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
```

如果您打算完成第 2 部分，請儲存該AgentSpaceArn值。您需要它來設定服務帳戶堆疊。

步驟 5：驗證部署

若要確認已成功建立代理程式空間，請執行下列 CLI AWS 命令：

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

此時，您的代理程式空間會在啟用 運算子應用程式且您的監控帳戶相關聯的情況下部署。代理程式可以監控此帳戶中的問題。

第 2 部分 （選用）：新增跨帳戶監控

在本節中，您會擴展設定，以便您的代理程式空間可以監控第二個 AWS 帳戶 （服務帳戶） 中的資源。這涉及兩個動作：

1. 在 DevOpsAgentStack 中新增指向服務帳戶的來源 AWS 關聯。
2. 使用信任代理程式空間的 IAM 角色，將 ServiceStack 部署至服務帳戶。

> **Important**
>
> 您必須完成第 1 部分，才能繼續。ServiceStack 需要 DevOpsAgentStack 部署輸出AgentSpaceArn中的 。

步驟 1：設定服務帳戶 ID

開啟lib/constants.ts並設定您的服務帳戶 ID：

下列範例顯示要更新的常數：

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

DevOpsAgentStack 使用此帳戶 ID 建立來源 AWS 關聯。如果您在設定此值之前已部署 DevOpsAgentStack，請重新部署 以建立關聯：

執行下列命令以重新部署：

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

步驟 2：設定客服人員空間 ARN

從 DevOpsAgentStack 輸出複製 `AgentSpaceArn`值 （ 第 1 部分，步驟 4)，並在 中設定： `lib/constants.ts`

下列範例顯示要更新的常數：

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

ServiceStack 使用此值來限制次要帳戶角色的信任政策範圍。只有在設定此值時，才會合成 ServiceStack。

步驟 3：引導服務帳戶

如果您尚未在服務帳戶中啟動 AWS CDK，請執行下列命令：

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

步驟 4：部署 ServiceStack

執行下列命令，使用服務帳戶的登入資料來建置和部署 ServiceStack：

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

這會在服務帳戶中建立下列資源：

- 信任監控帳戶中客服人員空間的 IAM 角色 (`DevOpsAgentRole-SecondaryAccount`)

- 以 echo Lambda 函數 (`echo-service`) 做為範例服務

步驟 5：驗證部署

若要確認 Lambda 函數已成功部署，請執行下列命令來測試 echo 服務：

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

疑難排解

本節說明常見問題以及如何解決這些問題。

找不到 CloudFormation 資源類型

- 確認您是在 中部署the section called “支援的區域”。
- 確認您的 AWS CLI 已設定適當的許可。

IAM 角色建立失敗

- 確認您的部署角色具有建立 IAM 角色的許可。
- 檢查信任政策條件是否符合您的帳戶 ID。

跨帳戶部署失敗，並顯示「無法在目標帳戶中擔任角色」

- 每個堆疊都必須使用目標帳戶的登入資料進行部署。使用 `--profile`旗標指定正確的 CLI AWS 設定檔。
- 確認已在目標帳戶中引導 AWS CDK。

IAM 傳播延遲

- IAM 角色變更可能需要幾分鐘的時間才能傳播。如果客服人員空間建立在角色建立後立即失敗，請等待幾分鐘，然後重新部署。

清除

若要移除所有資源，請以相反順序銷毀堆疊。

執行下列命令來銷毀堆疊：

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

警告：此動作會永久刪除您的客服人員空間和所有相關資料。此動作無法復原。在繼續之前，請確定您已備份任何重要資訊。

安全考量

- AWS CDK 應用程式會使用只允許`aidevops.amazonaws.com`服務主體擔任的信任政策來建立 IAM 角色。
- 信任政策包括限制存取特定 AWS 帳戶和客服人員空間 ARN 的條件。
- 所有政策都遵循最低權限原則。根據組織的安全需求檢閱和自訂 IAM 政策。
- 跨帳戶角色 (`DevOpsAgentRole-SecondaryAccount`) 使用固定名稱，範圍為特定的客服人員空間 ARN。

後續步驟

使用 AWS CDK 部署您的 AWS DevOps 代理程式之後：

1. 請參閱 DevOps Agent AWS 使用者指南，了解 DevOps Agent 功能的完整範圍。
2. 考慮將 AWS CDK 部署整合到您的 CI/CD 管道，以進行自動化基礎設施管理。

其他資源

- AWS DevOps 代理程式使用者指南
- GitHub 網站上的範例 CDK 儲存庫
- CLI 入門指南

開始使用 AWS CloudFormation 的 AWS DevOps 代理程式

概觀

本指南說明如何使用 AWS CloudFormation 範本來建立和部署 AWS DevOps 代理程式資源。範本會自動建立代理程式空間、 AWS Identity and Access Management (IAM) 角色、運算子應用程式，以及 AWS 帳戶關聯做為基礎設施做為程式碼。

CloudFormation 方法會透過在宣告式 YAML 範本中定義所有必要資源，自動執行 CLI 入門指南中所述的手動步驟。

AWS DevOps 代理程式可在下列 6 AWS 區域使用：美國東部 （維吉尼亞北部）、美國西部 （奧勒岡）、亞太區域 （雪梨）、亞太區域 （東京）、歐洲 （法蘭克福） 和歐洲 （愛爾蘭）。如需支援區域的詳細資訊，請參閱 the section called “支援的區域”。

先決條件

開始前，請確定您有下列項目：

- AWS 命令列界面 (AWS CLI) 已安裝並使用適當的登入資料設定
- 建立 IAM 角色和 CloudFormation 堆疊的許可
- 一個 AWS 帳戶用於監控 （主要） 帳戶
- （選用） 如果您想要設定跨 AWS 帳戶監控的第二個帳戶

本指南涵蓋的內容

本指南分為兩個部分：

- 第 1 部分 — 使用 運算子應用程式和監控帳戶中的 AWS 關聯部署代理程式空間。完成此部分後，代理程式可以監控該帳戶中的問題。
- 第 2 部分 （選用） — 將跨帳戶 IAM 角色部署到次要帳戶，並新增來源 AWS 關聯。此組態可讓客服人員空間跨帳戶監控資源。

第 1 部分：部署代理程式空間

在本節中，您會建立 CloudFormation 範本，在監控帳戶中佈建代理程式空間、IAM 角色、運算子應用程式和 AWS 關聯。

步驟 1：建立 CloudFormation 範本

將下列範本儲存為 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
```

步驟 2：部署堆疊

執行下列命令來部署堆疊。<REGION> 以 取代 the section called “支援的區域”（例如，us-east-1)。

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

步驟 3：記錄堆疊輸出

部署完成後，請執行下列命令來擷取堆疊輸出。記錄這些值以供日後使用。

```
aws cloudformation describe-stacks \
```

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

下列範例顯示預期的輸出：

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

如果您打算完成第 2 部分，請儲存該AgentSpaceArn值。您需要它來設定跨帳戶角色。

步驟 4：驗證部署

若要驗證已成功建立代理程式空間，請執行下列 AWS CLI 命令：

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

此時，您的代理程式空間會部署並啟用 運算子應用程式，並且與監控帳戶相關聯。代理程式可以監控此帳戶中的問題。

第 2 部分（選用）：新增跨帳戶監控

在本節中，您會擴展設定，讓您的客服人員空間可以監控第二個 AWS 帳戶（服務帳戶）中的資源。這涉及兩個動作：

1. 在信任客服人員空間的服務帳戶中部署 IAM 角色。
2. 在監控帳戶中新增指向服務帳戶的來源 AWS 關聯。

> 注意：您必須先完成第 1 部分，才能繼續。服務帳戶範本需要AgentSpaceArn來自第 1 部分堆疊輸出的 。

步驟 1：建立服務帳戶範本

將下列範本儲存為 devops-agent-service-account.yaml。此範本會在次要帳戶中建立跨帳戶 IAM 角色。

```
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
```

步驟 2：部署服務帳戶堆疊

使用服務帳戶的登入資料，執行下列命令：

```
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>
```

步驟 3：新增來源 AWS 關聯

切換回監控帳戶並建立來源 AWS 關聯。您可以透過建立個別堆疊或更新原始範本來執行此操作。下列範例使用獨立範本。

將下列範本儲存為 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
```

使用監控帳戶登入資料部署關聯堆疊：

```
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>
```

驗證

執行下列 CLI AWS 命令來驗證您的設定：

```
# 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>
```

疑難排解

本節說明常見問題以及如何解決這些問題。

找不到 CloudFormation 資源類型

- 確認您正在 中部署the section called “支援的區域”。
- 確認您的 AWS CLI 已設定適當的許可。

IAM 角色建立失敗

- 確認您的部署登入資料具有使用自訂名稱 () 建立 IAM 角色的許可CAPABILITY_NAMED_IAM。
- 檢查信任政策條件是否符合您的帳戶 ID。

跨帳戶部署失敗

- 每個堆疊都必須使用目標帳戶的登入資料進行部署。使用 --profile旗標指定正確的 CLI AWS 設定檔。
- 確認 AgentSpaceArn 參數符合來自第 1 部分堆疊輸出的確切 ARN。

IAM 傳播延遲

- IAM 角色變更可能需要幾分鐘的時間才能傳播。如果客服人員空間建立在角色建立後立即失敗，請等待幾分鐘，然後重新部署。

清除

若要移除所有資源，請以相反順序刪除堆疊。

警告：此動作會永久刪除您的客服人員空間和所有相關資料。此動作無法復原。在繼續之前，請確定您已備份任何重要資訊。

執行下列命令來刪除堆疊：

```
# 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>
```

後續步驟

使用 AWS CloudFormation 部署您的 AWS DevOps 代理程式之後：

- 若要連接其他整合，請參閱 設定適用於 AWS DevOps 代理程式的功能。
- 若要了解客服人員的技能和功能，請參閱 the section called “DevOps 代理程式技能”。
- 若要了解 運算子 Web 應用程式，請參閱 the section called “什麼是 DevOps Agent Web 應用程式？”。

使用 Terraform 開始使用 AWS DevOps 代理程式

概觀

本指南說明如何使用 Terraform 來建立和部署 AWS DevOps 代理程式資源。Terraform 組態會自動建立代理程式空間、IAM 角色、運算子應用程式和 AWS 帳戶關聯。

Terraform 方法透過將所有必要資源定義為基礎設施作為程式碼，自動執行 CLI 入門指南中所述的手動步驟。

AWS DevOps 代理程式可在下列 6 AWS 區域使用：美國東部（維吉尼亞北部）、美國西部（奧勒岡）、亞太區域（雪梨）、亞太區域（東京）、歐洲（法蘭克福）和歐洲（愛爾蘭）。如需支援區域的詳細資訊，請參閱 the section called “支援的區域”。

先決條件

開始前，請確定您具有下列項目：

- Terraform >= 已安裝 1.0
- AWS 使用適當的登入資料安裝和設定 CLI
- 一個 AWS 帳戶用於監控（主要）帳戶
- （選用）如果您想要設定跨 AWS 帳戶監控的第二個帳戶

本指南涵蓋的內容

本指南分為兩個部分：

- 第 1 部分 — 使用 運算子應用程式和監控帳戶中的 AWS 關聯部署代理程式空間。完成此部分後，客服人員可以監控該帳戶中的問題。
- 第 2 部分（選用）— 新增服務帳戶的來源 AWS 關聯，並將跨帳戶 IAM 角色加上 echo Lambda 部署至該帳戶。這可讓客服人員空間跨帳戶監控資源。

已建立資源

第 1 部分：監控帳戶

- IAM 角色 (DevOpsAgentRole-AgentSpace-*) — 由 DevOps Agent 服務擔任以監控帳戶。包括 AIDevOpsAgentAccessPolicy受管政策和允許建立 Resource Explorer 服務連結角色的內嵌政策。
- IAM 角色 (DevOpsAgentRole-WebappAdmin-*) — 具有客服人員操作AIDevOpsOperatorAppAccessPolicy受管政策的操作員應用程式角色。
- 客服人員空間（可設定的名稱）— 使用 awscc_devopsagent_agent_space 資源建立的中央客服人員空間。包括運算子應用程式組態。
- 關聯 (AWS 監視器）— 使用 awscc_devopsagent_association 資源將監控帳戶連結至客服人員空間。
- 關聯 (AWS 來源）—（選用）將服務帳戶連結到代理程式空間以進行跨帳戶監控。

第 2 部分：服務帳戶（選用）

- IAM 角色 (DevOpsAgentRole-SecondaryAccount-TF) — 具有固定名稱的跨帳戶角色。受監控帳戶中的代理程式空間信任。包括 AIDevOpsAgentAccessPolicy受管政策和允許建立 Resource Explorer 服務連結角色的內嵌政策。
- Lambda 函數 (echo-service-tf) — 回應輸入事件的簡單範例服務。

設定

步驟 1：複製範例儲存庫

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

步驟 2：設定變數

複製範例變數檔案，並針對您的環境進行自訂：

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

terraform.tfvars 使用您的客服人員空間名稱和描述進行編輯：

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

第 1 部分：部署代理程式空間

在本節中，您會在監控帳戶中建立代理程式空間、IAM 角色、運算子應用程式和 AWS 關聯。

步驟 1：使用自動化部署 （ 建議 ）

使用提供的部署指令碼進行簡化設定：

```
./deploy.sh
```

此指令碼會自動：

- 檢查先決條件 (Terraform、 AWS CLI、登入資料)
- 視需要`terraform.tfvars`從範例建立
- 初始化、驗證、規劃和套用 Terraform

或者，如果您偏好手動控制：

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

出現提示`yes`時輸入 以確認部署。

步驟 2：記錄輸出

部署完成後，Terraform 會列印輸出。記錄這些值以供日後使用：

```
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"
```

如果您打算完成第 2 部分，請儲存該agent_space_arn值。您將需要它來設定服務帳戶資源。

步驟 3：驗證部署

執行部署後驗證指令碼：

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

或使用 AWS CLI 來驗證已成功建立代理程式空間：

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

此時，您的代理程式空間會在啟用 運算子應用程式且您的監控帳戶相關聯的情況下部署。代理程式可以監控此帳戶中的問題。

第 2 部分 （選用）：新增跨帳戶監控

在本節中，您會擴展設定，讓客服人員空間可以監控第二個 AWS 帳戶 （服務帳戶） 中的資源。這涉及兩個動作：

1. 新增指向服務帳戶的來源 AWS 關聯。
2. 將跨帳戶 IAM 角色和 echo Lambda 函數部署至服務帳戶。

> **Important**
>
> 您必須完成第 1 部分，才能繼續。服務帳戶資源需要agent_space_arn來自第 1 部分部署輸出的 。

步驟 1：設定服務帳戶 ID

在 中terraform.tfvars，設定您的服務帳戶 ID：

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

步驟 2：設定客服人員空間 ARN

從第 1 部分輸出複製 agent_space_arn值 （ 步驟 2)，並在 中設定terraform.tfvars：

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

服務帳戶資源使用此值來限制次要帳戶角色的信任政策範圍。只有在設定此值時，才會建立這些資源。

步驟 3：設定 `aws.service` 供應商

在 中main.tf，使用服務帳戶的登入資料設定aws.service提供者別名。您可以使用具名設定檔或擔任角色：

使用設定檔：

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

或使用擔任角色：

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

步驟 4：部署

套用更新的組態：

```
terraform apply
```

這會在服務帳戶中建立下列資源：

- 信任監控帳戶中客服人員空間的 IAM 角色 (DevOpsAgentRole-SecondaryAccount-TF)
- 以 echo Lambda 函數 (echo-service-tf) 做為範例服務

它也會在監控帳戶中建立來源 AWS 關聯，以連結服務帳戶。

步驟 5：驗證部署

測試 echo 服務以確認 Lambda 函數已成功部署：

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

疑難排解

IAM 傳播延遲

- 組態包含 IAM 角色建立與客服人員空間建立time_sleep之間的 30 秒。DevOps Agent 服務會在建立 Agent Space 期間驗證運算子角色的信任政策，如果 IAM 尚未完全傳播，則可能會失敗。如果您仍然看到信任政策錯誤，請等待一分鐘，然後terraform apply再次執行 - IAM 角色將已存在，而套用將在停止的位置取得。

許可錯誤

- 確認您的 AWS 登入資料具有建立角色和政策所需的 IAM 許可。
- 檢查信任政策條件是否符合您的帳戶 ID。

跨帳戶部署失敗

- 必須使用服務帳戶的登入資料來設定aws.service提供者。使用具名設定檔或擔任角色區塊。
- 確認該agent_space_arn值符合來自第 1 部分輸出的 ARN。

找不到 Terraform 資源類型

- 確認您有awscc提供者版本 ~> 1.0或更新版本。awscc_devopsagent_agent_space 和資源awscc_devopsagent_association需要 AWS 雲端控制供應商。

清除

若要移除所有資源，如果您已部署第 2 部分，請以相反順序銷毀：

```
./cleanup.sh
```

或手動：

```
terraform destroy
```

警告：這會永久刪除您的客服人員空間和所有相關聯的資料。在繼續之前，請確定您已備份任何重要資訊。

安全考量

- Terraform 組態會建立具有信任政策的 IAM 角色，只允許aidevops.amazonaws.com服務主體擔任這些角色。
- 信任政策包括限制存取特定 AWS 帳戶和客服人員空間 ARN 的條件。
- 所有政策都遵循最低權限原則。根據組織的安全需求檢閱和自訂 IAM 政策。
- 跨帳戶角色 (DevOpsAgentRole-SecondaryAccount-TF) 使用固定名稱，範圍為特定的客服人員空間 ARN。

後續步驟

使用 Terraform 部署您的 AWS DevOps 代理程式之後：

1. 請參閱 DevOps Agent AWS 使用者指南，了解 DevOps Agent 功能的完整範圍。
2. 考慮將 Terraform 部署整合到您的 CI/CD 管道，以進行自動化基礎設施管理。

其他資源

- AWS DevOps 代理程式使用者指南
- 範例 Terraform 儲存庫

- CLI 入門指南

使用 DevOps 代理程式

使用 DevOps 代理程式

AWS DevOps 代理程式會與您的營運團隊合作，涵蓋從偵測到調查、復原和預防的完整事件生命週期。下列主題說明如何使用 DevOps 代理程式來管理此生命週期的每個階段。

自治事件回應

偵測到事件時，無論是透過內建的票證系統整合、監控工具中的 Webhook 還是手動觸發程序，DevOps 代理程式都會自動開始調查。代理程式會分析指標、日誌、追蹤、程式碼變更和部署歷史記錄，以判斷根本原因並提議緩解計畫。如果您需要其他協助，您可以從 DevOps Agent Space Web 應用程式直接向 AWS Support 呈報，該應用程式會自動與支援工程師共用調查內容，因此您不需要重複代理找到的內容。如需詳細資訊，請參閱the section called “自治事件回應”。

隨需 DevOps 任務

在事件生命週期的任何時候，您都可以透過對話式聊天介面與 DevOps Agent 互動。使用自然語言詢問有關 AWS 資源、系統運作狀態、警示狀態和部署歷史記錄的問題。聊天內容感知 – 當您檢視特定調查時，您可以引導客服人員探索特定假設、專注於特定日誌，或更新其根本原因分析。您也可以查詢整個環境的資源組態、錯誤趨勢和調查洞察，而無需在主控台之間導覽。如需詳細資訊，請參閱the section called “隨需 DevOps 任務”。

主動事件預防

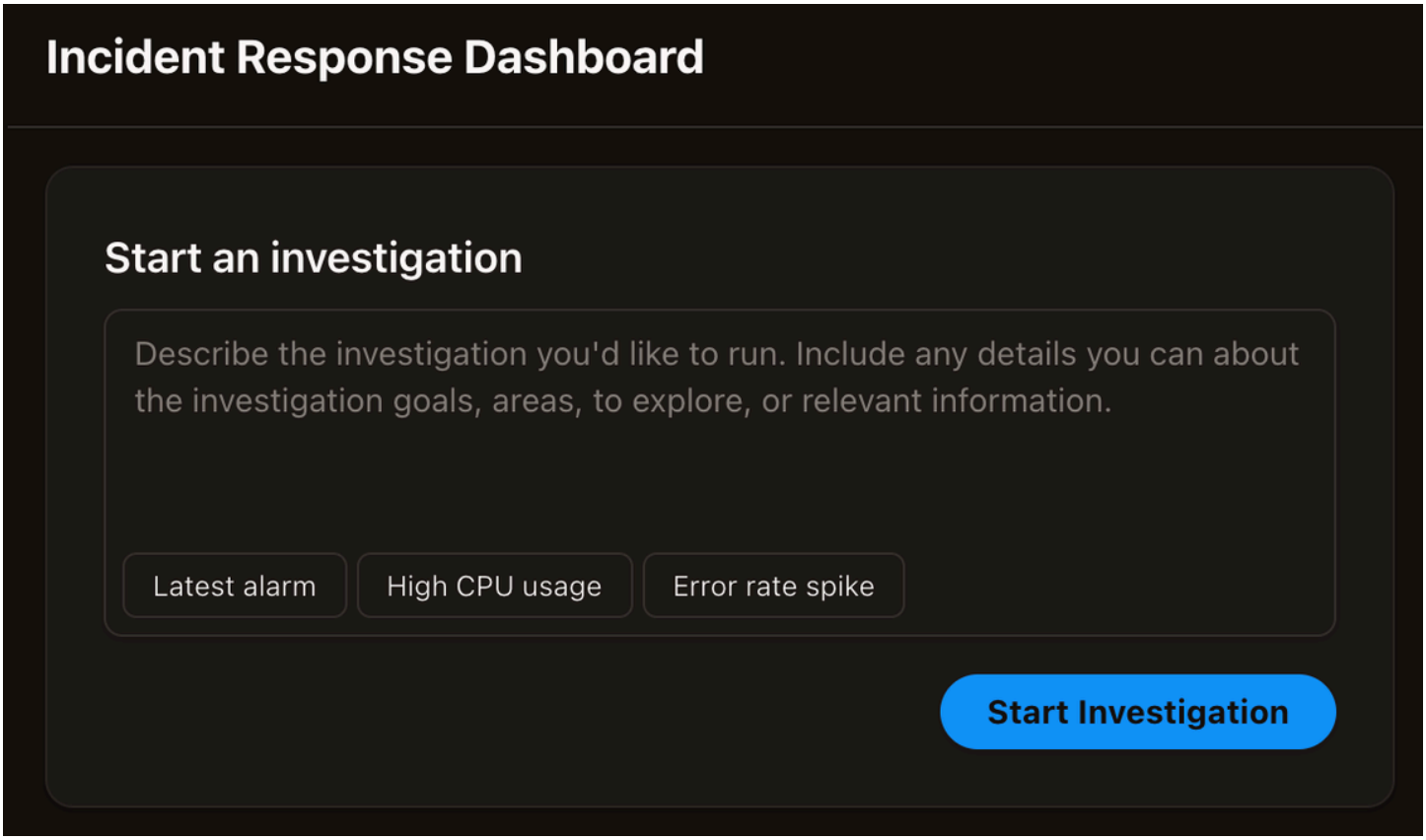
解決事件之後，DevOps 代理程式會分析調查歷史記錄中的模式，以產生防止未來事件並縮短平均偵測時間的建議。建議涵蓋四個領域：可觀測性狀態、測試差距、程式碼變更和基礎設施架構。代理程式每週執行評估，並在新事件發生時更新建議。您可以接受、拒絕或追蹤建議，而且客服人員會從意見回饋中學習，以精簡未來的建議。如需詳細資訊，請參閱the section called “主動事件預防”。

自治事件回應

開始調查

事件回應調查可以透過三種方式之一開始。

- 內建整合 - 您可以使用內建整合，將 DevOps Agent Space 連線至 ServiceNow 等票證系統。連線後，DevOps 代理程式事件回應調查將從支援票證自動觸發，您的 DevOps 代理程式將在原始票證中提供其關鍵調查結果、根本原因分析和緩解計劃的更新。
- Webhook - 您可以使用 Webhook 將事件傳送至 AWS DevOps 代理程式。例如，您可以使用 Webhook 觸發來自 PagerDuty 票證或 Grafana 警示的事件回應調查。
- 手動 - 您可以從任何 DevOps Agent Space Web 應用程式的事件回應索引標籤手動啟動事件回應調查。您可以輸入描述您希望 DevOps 代理程式調查之事件的任意格式文字，它會建立調查計畫、收集調查結果、判斷根本原因，並提供 來產生緩解計畫。您也可以從數個預先設定的起點中進行選擇，以快速開始您的調查： 最新警示以調查您最近觸發的警示，並分析基礎指標和日誌以判斷根本原因， 高 CPU 使用量可調查運算資源中的高 CPU 使用率指標，並識別哪些程序或服務耗用過多的資源， 或 錯誤率激增，透過分析指標來調查應用程式錯誤率最近增加的情況， 應用程式日誌、 並識別失敗的來源。



按一下「開始調查」後，系統會要求您提供一些其他詳細資訊，以協助客服人員專注於其工作。調查對話方塊包含下列欄位：

- 調查詳細資訊 – 預先填入您的描述。您可以編輯此項目來縮小調查範圍。

- 調查起點 – 選擇性地描述代理程式的特定警示、指標、日誌程式碼片段或其他起點。
- 事件發生的日期和時間 – 以 UTC 格式自動填入目前時間。如果事件發生，請調整 。
- 為您的調查命名 – 使用時間戳記自動產生。您可以自訂此項目 （最多 400 個字元）。
- 優先順序 – 從下拉式清單中選取調查優先順序 （預設為中）。

視需要檢閱和調整這些欄位，然後按一下「開始調查...」開始。然後，系統會將您導向至調查詳細資訊頁面，您可以在其中看到 DevOps 代理程式實際運作！

事件分類

分類階段是 AWS DevOps 代理程式事件回應系統的第一階段。當外部事件觸發時，例如來自 Datadog 的警示、來自 ServiceNow 的事件票證，或來自 Dynatrace 的問題時， AWS DevOps 代理程式會自動在幾秒鐘內處理，以判斷是否應獨立調查或連結至現有的調查。

分類階段的主要函數是事件關聯 — 識別相關事件並將其合併為單一調查，以避免重複的工作和資源浪費。當新事件送達時， AWS DevOps 代理程式會在回顧時段 （通常為 20 分鐘） 內與主動調查一起進行分析。使用 AI 技術的分析，它會檢查元件相似性、地理區域和時間模式等因素，以確定事件之間的關係。

AWS DevOps 代理程式會做出以下兩個決策之一：

- 連結 – 將事件與現有調查相關聯，並傳送指導訊息至該調查，其中包含新事件的相關內容。
- 繼續 – 為事件安排新的獨立調查。

檢視分類決策

連結事件時，主要調查會收到轉向訊息，其中包含連結事件的詳細資訊和相互關聯推理。在 your AWS DevOps Agent Space Web 應用程式中，您會看到 LINKED 的狀態，以及解釋事件連結原因的相互關聯推理。主要調查會顯示所有連結事件的清單，讓您可以查看一起調查的相關問題的完整範圍。您的外部票證系統 (ServiceNow、PagerDuty 等） 和通訊管道 (Slack) 將收到通知，告知事件已與相互關聯推理連結。

取消連結事件和自訂相互關聯規則

If AWS DevOps Agent 不正確地關聯事件，您可以透過 AWS DevOps Agent Space Web 應用程式手動取消連結事件。這會將未連結的事件重新排程為獨立調查。您也可以建立包含相互關聯邏輯的 AWS DevOps AWS 代理程式技能，並將其與分類階段建立關聯，藉此提供自訂相互關聯規則來引導 DevOps 代理程式。

尋求人類支援

AWS DevOps 代理程式可以直接與 AWS Support 連線，以簡化您的事件回應程序。當您需要 AWS Support 的其他協助時，您可以從 DevOps Agent Space Web 應用程式建立支援案例，自動與 AWS Support 工程師共用調查內容，以減少解釋問題所需的時間。

運作方式

調查事件時， AWS DevOps Agent 會建置其分析的完整日誌，包括：

- 根本原因調查結果
- 分析的指標、日誌和追蹤
- 程式碼變更和部署歷史記錄已檢閱
- 建議的修復動作
- 事件和系統行為的時間軸

您可以直接從 AWS DevOps Agent Space Web 應用程式將調查呈報給 AWS Support。當您這麼做時， AWS DevOps 代理程式會自動將其調查日誌傳遞給 AWS Support，為支援工程師提供有關調查的完整內容，而不需要您手動收集和解釋詳細資訊。

與 AWS Support 聊天

建立支援案例後，您可以在您的 AWS DevOps Agent Space Web 應用程式中的個別聊天視窗中與 AWS Support 通訊。這可讓您：

- 與 AWS Support 工程師和您的 AWS DevOps 代理程式的調查時間表一起討論您的問題
- 在相同界面中檢視 AWS DevOps 代理程式的自動化分析和 AWS Support 的專家指引
- 視需要無縫分享其他資訊或釐清

聊天體驗可讓您隨時存取您的 AWS DevOps 客服人員調查和 AWS 支援對話，從而加快協作和解決速度。

支援計劃需求

您透過 AWS DevOps 代理程式建立支援案例並與之互動的能力取決於您的 AWS 支援計劃。請參閱 Support Plans 使用者指南，進一步了解您的權利。

注意 基本支援 客戶無法建立技術支援案例，因此無法將 AWS DevOps 客服人員調查呈報 AWS 支援開發人員支援 客戶可以透過 AWS DevOps 客服人員建立案例，但必須造訪AWS 支援中心與支援工程師進行對應，因為開發人員支援不包含聊天支援 所有其他計劃都可以使用 AWS DevOps 客服人員中的整合聊天體驗。如需支援計劃權利的完整詳細資訊，包括回應時間和可用的案例嚴重性，請參閱AWS Support Plans 使用者指南。

與 AWS Support 共用哪些資訊

當您從 AWS DevOps Agent Space Web 應用程式建立支援案例時，會自動與 AWS Support 共用以下資訊：

- 調查時間表： AWS DevOps 代理程式分析的計時記錄
- 資源資訊：受影響的 AWS 資源
- 可觀測性資料：整合監控工具的相關指標、日誌和追蹤
- 最近的變更：程式碼部署、基礎設施變更和組態更新
- 修復嘗試：建議使用 Actions AWS DevOps 代理程式
- 影響評估：事件的範圍和嚴重性

與 AWS Support 共用的所有資料都遵循您現有的 AWS 資料落地和安全性組態。 AWS DevOps 代理程式只會共用與您特定調查相關的資訊，並尊重組織的資料控管政策。

開始使用

若要使用 AWS DevOps 代理程式的 AWS 支援整合：

1. 確保您有作用中的 AWS 支援計劃。
2. 驗證您的 AWS DevOps 代理程式的 IAM 許可包含支援案例建立 (support：CreateCase，support：DescribeCases)。
3. 當 AWS DevOps 代理程式正在調查問題且您需要 AWS 支援協助時，請選擇從您的 DevOps Agent Space Web 應用程式請求人工支援。
4. 檢閱將與 AWS Support 共用的調查摘要。
5. 根據您的支援計劃權利選取適當的案例嚴重性。
6. 提交案例 - AWS DevOps 代理程式會自動包含您的調查日誌。

聊天視窗會自動開啟，讓您可以立即開始與 AWS Support 協作。

主動事件預防

AWS DevOps 代理程式會分析整個事件調查的模式，以提供目標性建議，以持續改善您的營運狀態並防止未來的事件。透過 Operator Web 應用程式中的 Ops 待處理項目頁面存取主動事件預防。

主動事件預防的運作方式

AWS DevOps 代理程式會評估最近的事件調查，以識別持續的改進，以防止未來的事件，並加快平均偵測時間 (MTTD)。代理程式會分析多個事件，以識別可能在未來阻止整個事件類別的建議，專注於最具影響力的建議，以確保這些建議可採取動作。

根據預設，代理程式會每週自動執行評估。如果您想要僅隨需執行評估，可以暫停排程。手動評估隨時可用，這在最近的調查需要快速解決建議的改進時很有用。

客服人員會識別四個類別的改進，如 Ops Backlog 頁面上的建議分類圖表所示：

- 可觀測性 – 增強監控、提醒、記錄和系統可見性的建議，以更快、更準確地偵測問題。
- 基礎設施 – 最佳化資源組態、容量調校和架構彈性的建議。
- 控管 – 強化部署程序、管道改善、測試實務和操作控制的建議。
- 程式碼最佳化 – 改善應用程式程式碼品質、錯誤處理和程式碼彈性的建議。

此分類可協助您了解最需要改善營運的位置，並可讓您根據團隊的重點領域排定建議優先順序。

優勢

- 防止重複發生事件 – 有系統地解決根本原因，而不是重複回應相同類型的問題
- 減少營運管道 – 讓團隊免於重複的消防，專注於創新和策略改進
- 改善系統彈性 – 根據實際事件資料強化您的基礎設施、可觀測性和部署程序
- 從歷史模式學習 - 利用過去事件的洞察，進行具有最大影響的目標改善

客服人員摘要

Web 應用程式操作待處理項目頁面中的客服人員摘要提供最近事件上次評估結果的說明。摘要說明分析的事件調查數量、哪些事件與過去的事件類似，以及使用新資訊建立或更新了哪些建議。

摘要可協助您快速了解客服人員在最近一次評估期間發現的情況，並重點介紹可能對您的營運狀態產生最大影響的最值得注意的建議。

控制評估

您可以控制何時 AWS DevOps 代理程式評估事件並產生建議：

- 手動執行評估 – 按一下操作待處理項目頁面中的立即執行按鈕，立即開始評估。當最近的調查需要快速解決建議的改進時，這很有用。
- 停止作用中的評估 – 按一下 Ops Backlog 頁面中的停止評估按鈕，以停止目前正在進行的評估。

管理建議

AWS DevOps 代理程式會在 Ops Backlog 頁面中提供建議，您可以在其中檢閱和管理這些建議：

- 檢視建議詳細資訊 – 按一下建議以開啟建議詳細資訊頁面，您可以在其中查看建議改進的詳細資訊，包括通知建議的事件、預期的影響和後續步驟。如需程式碼變更的建議，您也可以檢視可供代理程式使用的規格，這些規格可以交給編碼代理程式進行實作。
- 保留 – 按一下「保留」，在待處理項目中保留建議以進行追蹤。這可讓您監控計劃實作和追蹤其進度的改善項目。
- 捨棄 – 按一下「捨棄」以從待處理項目中移除建議。當您捨棄建議時，您可以提供自然語言說明，說明它為何不符合您的需求。客服人員會從此意見回饋中學習，並用它來通知未來的建議，確保隨著時間的推移，它們更符合您的操作優先事項和需求。
- 已實作 – 按一下「已實作」，將建議標記為已完成。這可協助您追蹤已套用的改善項目，並允許代理程式測量其建議隨時間經過的有效性。
- 自動移除 – 如果實作建議不會阻止任何新事件，則大約 6 週後可能會移除未標記為保留或實作的建議。這可確保營運待處理項目頁面專注於最相關的營運挑戰改進。
- 建議更新 – 當發現有較新的事件被建議所阻止時，現有的建議會更新。更新可能會變更建議的優先順序，或根據新洞見精簡建議。

客服人員就緒規格

對於涉及程式碼或組態變更的建議， AWS DevOps 代理程式可以產生適用於代理程式的規格。此規格提供結構化文件，可直接交給編碼代理程式進行實作。

規格包括：

- 問題陳述式 – 問題及其根本原因的摘要
- 解決方案摘要 – 建議方法的高階描述

- 目標儲存庫 – 需要變更的特定儲存庫
- 程式碼變更 – 詳細說明需要變更的內容和原因，以及特定的檔案路徑和實作考量
- 測試需求 – 需要測試哪些案例
- 實作計畫 – 實作變更的分階段方法

代理程式就緒規格透過為編碼代理程式提供進行生產就緒變更所需的內容來加速實作，而不需要工程師進行廣泛的back-and-forth操作。

實作建議

若要將主動事件預防建議的價值最大化，請考慮採取下列做法來採取行動：

- 使用代理程式就緒規格 – 如需程式碼變更的建議，請使用產生的規格，透過將其交給編碼代理程式，或使用它作為手動實作的詳細指南來加速實作。
- 將建議新增至票證待處理項目 – 將建議複製到團隊的票證系統或專案管理工具，以確保與其他工程工作一起排定優先順序。
- 根據影響排定建議優先順序 – 首先專注於處理最常見或嚴重事件類型，或影響關鍵系統的建議。
- 追蹤實作進度 – 監控已實作的建議，並透過觀察類似事件是否隨著時間減少來衡量其有效性。
- 與開發團隊協調 – 與擁有受影響系統的適當團隊分享建議，確保他們擁有實作改善所需的內容和資源。

隨需 DevOps 任務

AWS DevOps Agent On Demand Tasks 是生成式人工智慧 (AI) 支援的對話式助理，可讓營運團隊查詢其應用程式架構、分析系統運作狀態，並使用自然語言存取調查洞見。您可以詢問有關 AWS 資源、系統指標、警示狀態、部署歷史記錄和事件模式的問題。聊天提供以實際基礎設施和操作資料為基礎的立即答案，無需在多個 AWS 主控台或監控工具之間導覽。

Chat 已整合到 DevOps Agent Space Web 應用程式，並根據您正在檢視的頁面提供內容感知回應。界面會維護對話歷史記錄，讓您能夠繼續先前的討論，並建置在先前的查詢上。

任務功能

AWS DevOps Agent On Demand Tasks 提供全方位的功能，可協助您管理和了解基礎設施：

資源查詢 – 詢問 代理程式空間中的 AWS 資源，包括 Lambda 函數、DynamoDB 資料表、EKS 部署、憑證和基礎設施組態。聊天功能可以根據執行時間版本、容量設定或部署狀態等屬性來篩選和分析資源。例如，詢問「有多少 Lambdas 使用 Python 3.8？」 或「我是否有任何即將過期的憑證？」

系統運作狀態分析 – 查詢目前和歷史系統運作狀態指標，包括警示狀態、錯誤率、CPU 使用率和服務可用性。聊天可以產生涵蓋特定時段的運作狀態摘要，並識別系統行為的趨勢。詢問以下問題：「在過去 24 小時內觸發了哪些警示？」 或「過去一小時有任何 5xx 錯誤？」

調查洞察 – 從已完成和進行中的調查存取資訊，包括根本原因分析、探索的假設、檢閱的日誌和解決模式。聊天可以識別常見的事件原因，並根據歷史資料提供建議。查詢「上個月事件最常見的原因是什麼？」 或「已完成調查的平均解決時間為何？」

調查轉向 – 檢視調查詳細資訊頁面時，請指示客服人員專注於特定日誌、探索特定假設或更新根本原因分析，以引導調查。提供轉向輸入，例如「付款服務日誌上的焦點和更新您的 RCA」或「探索 DynamoDB 調節導致問題的假設」。

聊天成品 – 產生結構化報告和文件，例如操作運作狀態摘要、錯誤報告和事件分析。成品會出現在專用面板中，並支援對話中的版本編輯。

建議篩選 – 使用特定條件查詢事件預防建議，例如與特定服務或操作問題相關的建議。聊天說明每個建議的影響和實作考量。例如，「顯示可預防涉及 DynamoDB 之事件的建議」或「哪些建議有助於更快偵測請求延遲問題？」

存取聊天

聊天功能可作為 DevOps Agent Space Web 應用程式左側的持久性面板。左側邊欄包含 + 新增聊天按鈕、導覽至事件、操作待處理項目和拓撲的頁面區段，以及顯示您最近對話的聊天區段。選擇檢視全部，查看您的完整對話歷史記錄。

聊天功能會根據您存取內容的位置提供內容感知回應：

拓撲 – 詢問有關 Agent Space 資源、架構和操作運作狀態的一般問題。聊天功能可完整查看所有連線的帳戶和服務。在此內容中，您可以查詢資源組態、部署歷史記錄、拓撲資訊和可觀測性工具整合。

事件回應 – 檢視事件回應頁面時，請針對客服人員空間的調查趨勢、解決時間和事件模式提出問題。聊天功能可以分析歷史調查資料，以識別常見原因和改善機會。

調查詳細資訊 – 檢視特定調查時，聊天會提供有關該調查的內容感知回應。詢問有關已檢閱的日誌、探索的假設、根本原因結論和緩解計劃。您也可以提供轉向輸入來引導調查焦點。

預防 – 從預防頁面，使用篩選條件查詢建議，了解提出建議的原因，並探索實作方法。聊天可協助您排定優先順序，並了解事件預防建議的影響。

當您在頁面之間切換時，聊天界面仍然可用，但內容會變更以提供目前檢視的相關資訊。當您開始新的對話時，它會在沒有上下文的情況下開始。當您繼續現有的對話時，聊天會維護後續問題的完整對話歷史記錄。

內容感知回應

Chat 會根據您在 DevOps Agent Space Web 應用程式中檢視的頁面調整其回應。此內容感知可確保您接收相關資訊，而不需要指定您要詢問的調查或資源範圍。

檢視調查詳細資訊頁面時，聊天會自動了解您正在詢問有關該特定調查的問題。例如「您查看了哪些日誌？」的問題 或「您探索了哪些假設？」 請參閱目前顯示的調查。當您提供轉向輸入時，聊天會將其套用至主動調查，並在適當時建立新的根本原因版本。

在預防頁面上，聊天了解您對事件預防建議感興趣。查詢會自動篩選和分析客服人員空間內容中的建議。系統會辨識您是否詢問一般建議或特定建議詳細資訊。

從拓撲頁面存取聊天時，聊天可讓您全面了解客服人員空間中的所有資源、指標和歷史資料。您可以詢問任何資源、服務或營運問題，而無需指定調查或建議內容。

這種內容感知不需要重複指定您正在參考的調查、建議或資源範圍，從而建立更自然的對話流程。

管理對話

聊天會維護對話歷史記錄，讓您繼續先前的討論並參考先前的查詢。

建立新的對話 – 按一下聊天面板中的「新工作階段」按鈕，在沒有先前內容的情況下開始新的對話。新的對話不會延續先前聊天中的資訊，可讓您提出不相關的問題，而不會產生混淆。

存取對話歷史記錄 – 按一下「歷史記錄」以檢視客服人員空間中的所有先前對話。對話會依時間順序與時間戳記和預覽文字進行組織。對話歷史記錄會保留 90 天，並在客服人員空間中私有至您的使用者帳戶。

繼續對話 – 從歷史記錄中選取任何對話，從您離開的地方繼續。聊天會維護先前訊息的完整內容，讓您可以提出參考先前對話部分的後續問題。當您在檢視對話時切換頁面時，對話內容仍會保留，但頁面特定內容會根據您目前的位置更新。

請注意，對話歷史記錄會在每個客服人員空間中隔離。無法從其他客服人員空間看見或存取一個客服人員空間中的對話。此隔離可確保敏感資訊根據您的組織界限保持隔離。

產生成品

AWS DevOps Agent 支援聊天成品：客服人員在對話期間產生的結構化、版本控制文件。成品在聊天 UI 中提供專用的互動式面板，用於檢閱和編輯 AI 產生的內容，例如操作報告、錯誤摘要和運作狀態評估。

您可以從 DevOps Agent Space Web 應用程式中的任何頁面請求成品。聊天使用目前的頁面內容來限定成品內容的範圍。

成品的運作方式

當您要求聊天建立或更新內容時，聊天會產生成品 — 通常是格式化的文件 — 並將其顯示在對話旁的成品面板中。

產生 – 傳送自然語言請求以建立報告或文件。例如，詢問「為我的客服人員空間產生每週營運運作狀態報告」或「顯示上週我的 4xx 錯誤報告」。

檢閱 – 成品與對話一起出現在專用面板中。您可以檢閱完整內容，同時繼續與聊天互動。

編輯 – 透過聊天請求對成品進行變更。例如，詢問「在 Lambda 冷啟動上新增區段」或「更新報告以包含上個月的資料」。Chat 會使用您請求的變更來建立新的成品版本。

範例查詢

下列範例示範您可以詢問 Chat 的問題類型。這些範例會依使用案例和內容進行組織。

成品產生查詢

從 DevOps Agent Space Web 應用程式的任何頁面：

- 為我的客服人員空間產生每週營運運作狀態摘要
- 建立上週所有 4xx 錯誤的報告
- 建立過去 30 天的事件摘要報告
- 建立本週付款服務的警示活動摘要
- 產生過去 7 天的部署歷史記錄報告
- 將所有開啟的建議摘要到報告中

資源資訊查詢

從 DevOps Agent Space Web 應用程式的任何頁面：

- 有多少 Lambda 函數正在使用 Python 3.8？
- 我是否有任何即將過期的憑證？
- 列出具有隨需計費的所有 DynamoDB 資料表
- 在生產環境中顯示 EKS 叢集
- 哪些 Lambda 函數在過去 90 天內尚未部署？
- 列出未啟用版本控制的 S3 儲存貯體
- 哪些 RDS 執行個體正在執行資料庫版本 X？

系統運作狀態查詢

從拓撲或事件回應頁面：

- 在過去 24 小時內發出的警示有哪些？
- 過去一小時是否有任何 5xx 錯誤？
- 顯示付款服務的 Lambda 錯誤趨勢
- 我的 ECS 叢集的 CPU 使用率是多少？
- 負載平衡器中是否有任何運作狀態不佳的目標？
- 顯示昨天的 API Gateway 限流事件
- 上週哪些服務的錯誤率最高？
- 提供涵蓋過去 24 小時的整體運作狀態報告

可觀測性工具查詢

從拓撲：

- 列出 Splunk 日誌群組
- 顯示 Prometheus 指標及其警示閾值
- 此服務設定了哪些 Datadog 監視器？
- 列出新複本提醒政策
- 顯示 Dynatrace 儀表板組態

調查洞察查詢

從事件回應頁面：

- 上個月發生的事件最常見的原因是什麼？
- 已完成調查的平均解決時間為何？
- 總結上週的調查及其 RCA
- DynamoDB 調節造成多少個事件？
- 顯示過去季度的調查趨勢
- 哪些服務發生最常見的事件？

調查詳細資訊查詢

從調查詳細資訊頁面：

- 您查看了哪些日誌？
- 您探索了哪些假設？
- 您提議的緩解動作風險有多高？
- 此事件期間的事件時間表為何？
- 您為什麼認為這是根本原因？
- 哪些證據支援您的根本原因分析？
- 調查期間，誰提供了指導？
- 提供此事件調查的摘要

調查轉向查詢

從調查詳細資訊頁面：

- 專注於 14：00-15：00 UTC 之間的付款服務日誌，並更新您的 RCA
- 探索 DynamoDB 調節導致問題的假設
- 檢查 ECS 叢集組態，以查看是否造成警示
- 只檢查過去 2 小時的日誌，而不是整天的日誌
- 在下午 3 點調查錯誤峰值
- 查看 API Gateway 日誌而非 Lambda 日誌

預防建議查詢

從預防頁面：

- 我的前 3 個事件預防建議是什麼？
- 顯示可預防涉及 DynamoDB 之事件的建議
- 哪些建議有助於我更快地偵測請求延遲問題？
- 列出可預防類似事件的可觀測性改善
- 顯示付款服務的基礎設施建議
- 哪些建議對系統彈性的影響最高？

在客服人員空間中啟用聊天

聊天功能適用於所有 DevOps Agent Space Web 應用程式。設定程序取決於您是否擁有新的或現有的客服人員空間。

新的客服人員空間

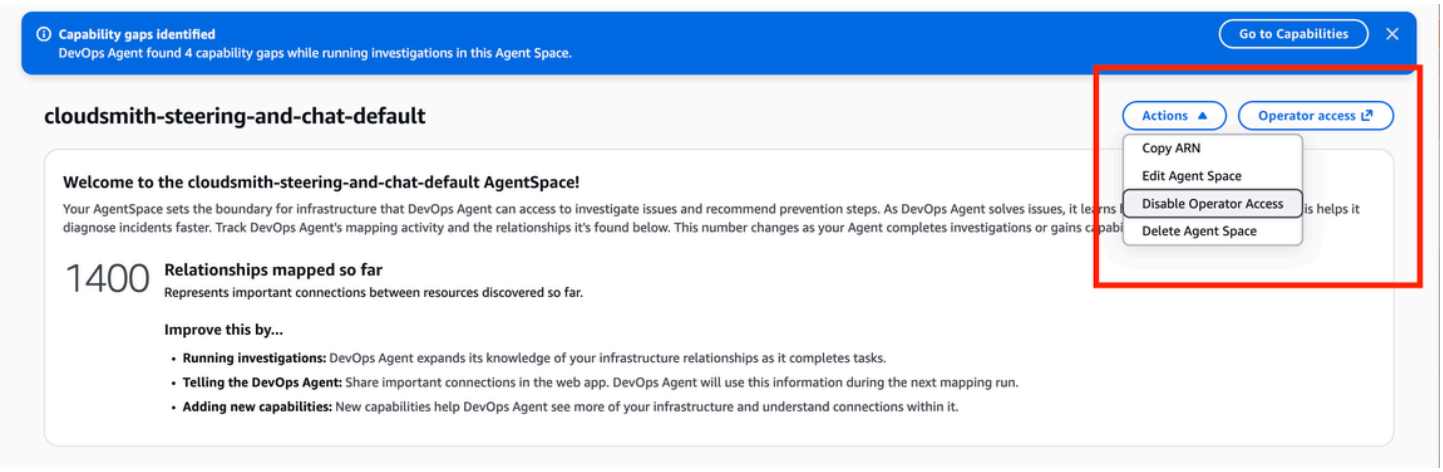
當您建立新的客服人員空間時，聊天功能會自動啟用。不需要額外的組態或 IAM 許可設定。設定 DevOps Agent Space Web 應用程式後，聊天功能可立即做為任何頁面左側的持久性面板使用。

現有的客服人員空間

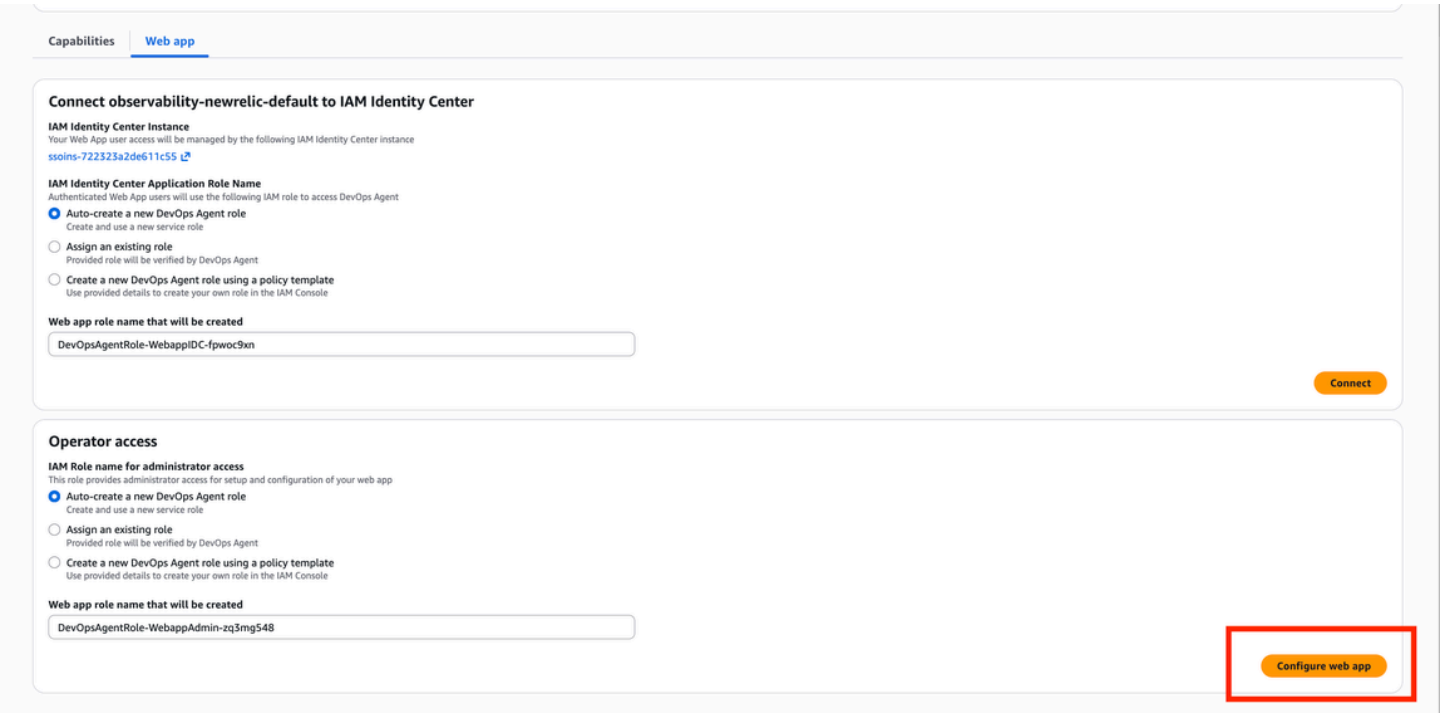
如果您在聊天發佈之前建立了客服人員空間，則必須啟用所需的 IAM 許可。您有兩種選擇：

選項 1：撤銷並重新啟用運算子應用程式存取

導覽至 AWS DevOps 代理程式管理主控台，找到右上角的動作下拉式清單，並停用目前的操作員存取組態。



然後，為操作員存取啟用自動建立選項。



這會自動套用聊天所需的 IAM 許可，以及所有其他目前的運算子許可。

選項 2：手動新增 IAM 許可

將下列 IAM 許可新增至您現有的操作員存取角色：

- `aidevops:ListChats` – 檢視聊天對話歷史記錄
- `aidevops:CreateChat` – 建立新的聊天對話
- `aidevops:SendMessage` – 傳送訊息和接收回應

導覽至 AWS IAM 主控台，找到您的 DevOps Agent Operator 角色，然後將這些許可新增至角色政策。新增許可後，聊天會立即可用。

完成任一選項後，請重新整理 DevOps Agent Space Web 應用程式，聊天面板會顯示在任何頁面的左側。

設定適用於 AWS DevOps 代理程式的功能

AWS DevOps Agent 功能透過將代理程式連接到現有的工具和基礎設施來擴展代理程式的功能。設定這些功能以啟用全方位的事件調查、自動化回應工作流程，以及與 DevOps 生態系統的無縫整合。

下列功能可協助您將 DevOps 代理程式的有效性最大化：

- AWS EKS 存取設定 - 為公有和私有 EKS 環境啟用 Kubernetes 叢集、Pod 日誌和叢集事件的自我檢查
- Azure 整合 - 連接 Azure 訂閱和 Azure DevOps 組織，以調查 Azure 資源並將 Azure DevOps 部署與事件建立關聯
- CI/CD 管道整合 - 連接 GitHub 和 GitLab 管道，以將部署與事件建立關聯，並在調查期間追蹤程式碼變更
- MCP 伺服器連線 - 透過模型內容通訊協定連接外部可觀測性工具和自訂監控系統來擴展調查功能
- 多帳戶 AWS 存取 - 設定次要 AWS 帳戶，以在事件回應期間調查整個組織的資源
- 遙測來源整合 - 連接監控平台，例如 Datadog、Dynatrace、Grafana、New Relic 和 Splunk，以實現全面的可觀測性資料存取
- 票證和聊天整合 - Connect ServiceNow、PagerDuty 和 Slack，以自動化事件回應工作流程並啟用團隊協作
- Webhook 組態 - 允許外部系統透過 HTTP 請求自動觸發 DevOps Agent 調查
- Amazon EventBridge 整合 - 透過將調查和緩解生命週期事件路由到 Amazon EventBridge 目標，將 AWS DevOps 代理程式併入事件驅動型應用程式

您可以根據團隊的特定需求和現有的工具堆疊，獨立設定每個功能。從對事件回應工作流程最關鍵的整合開始，然後視需要擴展至其他功能。

從公開預覽遷移到一般可用性

如果您在公開預覽期間使用 AWS DevOps 代理程式，您必須在 GA 發行之前更新您的 IAM 角色。本指南會逐步解說更新 帳戶中的監控角色和運算子角色。

正在變更的內容

1. 無法再存取預覽期間的隨需聊天歷史記錄
2. 新的 受管政策取代預覽期間可用的政策

3. Agent Spaces 可能有過時的 IAM Identity Center 應用程式存取範圍

來自公開預覽的隨需聊天歷史記錄

GA 版本引進額外的安全措施，以強化聊天歷史記錄的存取控制。由於這些變更，來自公開預覽期間 (2026 年 3 月 30 日之前） 的隨需聊天歷史記錄將無法再存取。在公開預覽期間建立的調查日誌和調查結果不受影響。此變更僅適用於隨需聊天對話。

新的 受管政策

對於 GA， AWS 提供取代預覽時代政策的新受管政策：

角色類型	移除	加
監控	`AIOpsAssistantPolicy` 受管政策	`AIDevOpsAgentAccessPolicy` 受管政策
運算子 (IAM 和 IDC)	內嵌政策	`AIDevOpsOperatorAppAccessPolicy` 受管政策

此外，運算子角色需要更新的信任政策，而 IDC 運算子角色需要新的內嵌政策。

先決條件

- 存取已設定 DevOps 代理程式角色 AWS 的帳戶 （主要和所有次要帳戶）
- 修改角色、政策和信任關係的 IAM 許可
- 您的客服人員空間 ID、 AWS 帳戶 ID 和區域 (DevOps Agent 主控台中可見）

步驟 1：更新監控角色

更新主要帳戶中和每個次要帳戶中的監控角色。這些是在您的客服人員空間的功能索引標籤下設定的主要/次要來源角色 （範例主要/次要角色：`DevOpsAgentRole-AgentSpace-3xj2396z`)。

1. 在 DevOps Agent 主控台中，前往您的 Agent Space，然後選擇功能索引標籤。
2. 尋找主要/次要來源的監控角色 （例如 `DevOpsAgentRole-AgentSpace-3xj2396z`)，然後選擇編輯。

3. 在許可政策下，移除 AIOpsAssistantPolicy AWS 受管政策。

4. 選擇新增許可、連接政策，以及連接AIDevOpsAgentAccessPolicy受管政策。

5. 編輯內嵌政策，並以下列內容取代其內容，以取代您的帳戶 ID：

```
{
    "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. 監控角色的信任政策不需要變更。驗證是否符合下列各項：

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

```
            }
        }
    ]
}
```

- 為每個次要帳戶中的監控角色重複步驟 2-6。

步驟 2：更新運算子角色 (IAM)

1. 在 DevOps Agent 主控台中，選擇存取索引標籤並尋找運算子角色。
2. 在 IAM 主控台中，從運算子角色移除現有的內嵌政策。
3. 選擇新增許可、連接政策，以及連接AIDevOpsOperatorAppAccessPolicy受管政策。
4. 選擇信任關係索引標籤，然後選擇編輯信任政策。將信任政策取代為下列項目，取代您的帳戶 ID、區域和客服人員空間 ID：

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

步驟 3：更新運算子角色 (IDC)

如果您使用 IAM Identity Center 搭配 DevOps Agent，請更新每個 IDC 運算子角色。

1. 在 IAM 主控台中，前往角色並搜尋 `WebappIDC`以尋找您的 DevOps 代理程式 IDC 角色 （例如 `DevOpsAgentRole-WebappIDC-<id>`)。
2. 對於每個 IDC 角色：

a. 移除現有的內嵌政策。

b. 選擇新增許可、連接政策，以及連接`AIDevOpsOperatorAppAccessPolicy`受管政策。

c. 選擇信任關係索引標籤，然後選擇編輯信任政策。將信任政策取代為下列項目，取代您的帳戶 ID、區域和客服人員空間 ID：

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

```
                "StringEquals": {
                    "aws:SourceAccount": "<account-id>"
                },
                "ArnEquals": {
                    "aws:SourceArn": "arn:aws:aidevops:<region>:<account-
id>:agentspace/<agentspace-id>"
                },
                "ForAllValues:ArnEquals": {
                    "sts:RequestContextProviders": [
                        "arn:aws:iam::aws:contextProvider/IdentityCenter"
                    ]
                },
                "Null": {
                    "sts:RequestContextProviders": "false"
                }
            }
        }
    ]
}
```

d. 使用下列許可建立新的內嵌政策，以取代您的帳戶 ID：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowDevOpsAgentSSOAccess",
            "Effect": "Allow",
            "Action": [
                "sso:ListInstances",
                "sso:DescribeInstance"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowDevOpsAgentIDCUserAccess",
            "Effect": "Allow",
            "Action": "identitystore:DescribeUser",
            "Resource": [
                "arn:aws:identitystore::<account-id>:identitystore/*",
                "arn:aws:identitystore:::user/*"
            ]
        }
```

```
    ]
}
```

重新連線 IAM Identity Center （ 如適用 ）

在公開預覽期間建立的 代理程式空間，可能有已設定過時存取範圍的 IAM Identity Center 應用程式。對於 GA，正確的範圍為 **aidevops:read_write**。如果您的 IAM Identity Center 應用程式有先前的範圍 (**awsaidevops:read_write**)，您必須中斷連線並重新連線 IAM Identity Center。

如何檢查您的 IAM Identity Center 應用程式範圍

執行下列 AWS CLI 命令來檢查 IAM Identity Center 應用程式上的範圍。您可以在應用程式下的 IAM Identity Center 主控台中找到應用程式 ARN。

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

輸出應會顯示正確的範圍 **aidevops:read_write**：

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

如果範圍顯示 **awsaidevops:read_write**，則會過期。請依照下列步驟進行更新。

如何重新連線 IAM Identity Center

AWS 受管 IAM Identity Center 應用程式的存取範圍無法直接更新。您必須中斷連線並重新連線：

1. 在 AWS DevOps 代理程式主控台中，前往您的代理程式空間，然後選擇存取索引標籤。
2. 選擇 IAM Identity Center 組態旁的中斷連線。
3. 確認中斷連線。
4. 選擇連線以再次設定 IAM Identity Center。服務會以正確的範圍建立新的 IAM Identity Center 應用程式。

5. 在 IAM Identity Center 主控台中將使用者和群組重新指派給新應用程式。

> **Important**
>
> 中斷連線會移除與 IAM Identity Center 使用者帳戶相關聯的個別使用者聊天和成品歷史記錄。重新連線後，使用者將需要再次登入。

驗證

完成所有步驟後：

1. 返回 DevOps Agent 主控台，並確認 Agent Space Access 索引標籤上沒有出現許可錯誤。
2. 測試 運算子 Web 應用程式，以確認其正確載入和運作。
3. 如果您使用 IDC，請確認使用者可以驗證和存取運算子體驗。

疑難排解

遷移後許可遭拒錯誤

- 確認 `AIOpsAssistantPolicy` 已移除`AIDevOpsAgentAccessPolicy`並連接至監控角色。
- 確認舊的內嵌政策已移除`AIDevOpsOperatorAppAccessPolicy`並連接至運算子角色。
- 檢查運算子信任政策是否包含 `sts:TagSession`。
- 確認您已將所有預留位置值 (`<account-id>`、`<region>`、`<agentspace-id>`) 取代為實際值。

次要帳戶無法運作

- 每個次要帳戶的監控角色都必須獨立更新。登入每個帳戶並重複步驟 1。

IDC 身分驗證失敗

- 驗證 IDC 信任政策同時包含 `sts:AssumeRole`/`sts:TagSession` 陳述式和 `TrustedIdentityPropagation`陳述式。
- 使用 `sso:ListInstances`、 `sso:DescribeInstance`和 確認內嵌政策`identitystore:DescribeUser`已建立。

遷移後缺少隨需聊天歷史記錄

- 在 GA 發行之後，無法存取來自公開預覽期間的隨需聊天歷史記錄。這是由於 GA 中引入的增強型安全措施而預期的行為。來自公開預覽的調查日誌和調查結果不受影響。

AWS EKS 存取設定

您可以針對公有和私有叢集執行唯讀`kubectl`命令，讓 AWS DevOps 代理程式調查 Amazon EKS 叢集中的問題。您可以將任意數量的 EKS 叢集連接到相同的代理程式空間。

連線後，代理程式可以協助診斷叢集中的操作問題：描述資源、擷取 Pod 日誌、檢查叢集事件、檢查節點運作狀態等。代理程式無法建立、修改或刪除叢集中的任何資源。

先決條件

在設定 EKS 存取之前，請確定 EKS 叢集的身分驗證模式包含 EKS API。您可以在 Amazon EKS 主控台的存取索引標籤上檢查此問題。如果模式不包含 EKS API，請在繼續之前選取執行的模式。

設定

對於您要為其建立存取項目的每個叢集，這些步驟需要從 Amazon EKS 主控台完成。您可以在 Agent Space（請參閱 the section called “建立 代理程式空間”) 的功能 > 雲端 > 主要來源 > 編輯下找到您的 IAM 角色 ARN。

1. 前往存取索引標籤。如果身分驗證模式已顯示 EKS API，您可以新增存取項目。否則，請選取包含 EKS API 的模式。
2. 從存取索引標籤中，建立新的 IAM 存取項目。複製您的主要雲端來源 IAM 角色 ARN，並將其輸入為存取項目的 IAM 主體。按一下 Next (下一步)。
3. 選取 AWS Managed AmazonAIOpsAssistantPolicy 存取政策，然後選取存取範圍的叢集。（或者，如果您希望代理程式只存取特定命名空間，請選取所需的 Kubernetes 命名空間)。按一下新增政策，然後按一下下一步。
4. 檢閱變更並確認已選擇正確的存取項目政策和 IAM 角色，然後按一下「建立」來建立您的存取項目。

若要驗證 EKS 存取是否已正確設定，請導覽至運算子應用程式並開始新的調查，向客服人員詢問有關叢集的問題，例如「列出預設命名空間中的所有 Pod」或「顯示我叢集中的最近事件」。

疑難排解

如果代理程式無法連接您的叢集，請確認存取項目使用設定對話方塊中顯示的正確 IAM 角色 ARN，且已連接 AmazonAIOpsAssistantPolicy 存取政策。

連接 Azure

Azure 整合可讓 AWS DevOps 代理程式調查 Azure 環境中的資源，並將 Azure DevOps 管道部署與操作事件建立關聯。透過連接 Azure，代理程式可以獲得 Azure 基礎設施的可見性，並可以跨 AWS 和 Azure 資源執行根本原因分析。

Azure 整合包含兩個獨立功能：

- Azure 資源 – 可讓代理程式探索和調查 Azure 雲端資源，例如虛擬機器、Azure Kubernetes Service (AKS) 叢集、資料庫和網路元件。代理程式會使用 Azure Resource Graph 在事件調查期間查詢您的資源。
- Azure DevOps – 讓代理程式存取 Azure DevOps 儲存庫和管道執行歷史記錄。代理程式可以將程式碼變更和部署與事件建立關聯，以協助識別潛在的根本原因。

每個功能都會在 AWS 帳戶層級註冊，然後可以與個別客服人員空間建立關聯。

註冊方法

AWS DevOps Agent 支援兩種連線至 Azure 的方法：

- 管理員同意 – 簡化的以同意為基礎的流程，您可以在其中授權 Azure 租用戶中的 AWS DevOps Agent Entra 應用程式。在 主控台中，這會顯示為管理員同意選項。此方法需要使用具有在 Microsoft Entra ID 中執行管理員同意許可的帳戶登入。
- 應用程式註冊 – 一種自我管理的方法，可讓您使用傳出聯合身分，使用聯合身分憑證建立自己的 Entra 應用程式。在 主控台中，這會顯示為應用程式註冊選項。當您需要對應用程式組態進行更多控制，或管理員同意許可不可用時，此方法很適合。

這兩種方法都提供相同的功能。您可以在同一個 AWS 帳戶中使用一種或兩種方法。

已知限制

- 管理員同意：每個 Azure 租用戶一個 AWS 帳戶 – 每個 Azure 租用戶一次只能與一個 AWS 帳戶 AWS DevOps 建立關聯。若要將相同的租用戶與不同的 AWS 帳戶建立關聯，您必須先取消註冊現有的註冊。
- 應用程式註冊：每個註冊的唯一應用程式 – 每個應用程式註冊都必須使用不同的應用程式（用戶端 ID)。您無法使用相同的用戶端 ID 註冊多個組態。
- Azure DevOps：原始程式碼存取 – Azure DevOps 整合提供管道執行歷史記錄的存取，無論原始程式碼託管在何處。不過，若要存取實際來源碼，儲存庫必須透過支援的來源提供者（例如）分別連線the section called “連接 GitHub”。Bitbucket 中託管的原始程式碼無法透過 Azure DevOps 整合直接存取。

主題

- the section called “連接 Azure 資源”
- the section called “連接 Azure DevOps”

連接 Azure 資源

Azure 資源整合可讓 AWS DevOps 代理程式在事件調查期間探索和調查 Azure 訂閱中的資源。代理程式使用 Azure Resource Graph 進行資源探索，並可跨 Azure 環境存取指標、日誌和組態資料。

此整合遵循兩個步驟：在 AWS 帳戶層級註冊 Azure，然後將特定 Azure 訂閱與個別客服人員空間建立關聯。

先決條件

連接 Azure 資源之前，請確定您已：

- 存取 AWS DevOps 代理程式主控台
- 可存取目標訂閱的 Azure 帳戶
- 針對管理員同意方法：具有在 Microsoft Entra ID 中執行管理員同意許可的帳戶
- 對於應用程式註冊方法：具有設定聯合身分憑證許可的 Entra 應用程式，以及在 AWS 您的帳戶中啟用傳出身分聯合

注意：您也可以從 代理程式空間內開始註冊。導覽至次要來源，按一下新增，然後選取 Azure。如果 Azure Cloud 尚未註冊，主控台會先引導您完成註冊。

透過管理員同意註冊 Azure 資源

管理員同意方法使用以同意為基礎的流程搭配 AWS DevOps 代理程式受管應用程式。

步驟 1：開始註冊

1. 登入 AWS 管理主控台並導覽至 AWS DevOps 代理程式主控台
2. 前往功能提供者頁面
3. 找到 Azure 雲端區段，然後按一下註冊
4. 選取管理員同意註冊方法

步驟 2：完成管理員同意

1. 檢閱請求的許可
2. 按一下以繼續 - 系統會將您重新導向至 Microsoft Entra 管理員同意頁面
3. 使用具有執行管理員同意許可的使用者主體帳戶登入
4. 檢閱並授予對 AWS DevOps Agent 應用程式的同意

步驟 3：完成使用者授權

1. 管理員同意後，系統會提示您輸入使用者授權，以驗證您的身分為授權租用戶的成員
2. 使用屬於相同 Azure 租用戶的帳戶登入
3. 授權之後，系統會將您重新導向回成功狀態的 AWS DevOps 代理程式主控台

步驟 4：指派角色

請參閱以下指派 Azure 角色。選取成員時搜尋 AWS DevOps 代理程式。

透過應用程式註冊註冊 Azure 資源

應用程式註冊方法使用您自己的 Entra 應用程式搭配聯合身分憑證。

步驟 1：開始註冊

1. 在 AWS DevOps 代理程式主控台中，前往功能提供者頁面

2. 找到 Azure 雲端區段，然後按一下註冊
3. 選取應用程式註冊方法

步驟 2：建立和設定您的 Entra 應用程式

遵循主控台中顯示的指示，以：

1. 在 AWS 帳戶中啟用傳出聯合身分 （在 IAM 主控台中，前往帳戶設定 → 傳出聯合身分)
2. 在 Microsoft Entra ID 中建立 Entra 應用程式，或使用現有的應用程式
3. 在應用程式上設定聯合身分憑證

步驟 3：提供註冊詳細資訊

使用下列項目填寫註冊表單：

- 租用戶 ID – Azure 租用戶識別符
- 租用戶名稱 – 租用戶的顯示名稱
- 用戶端 ID – 您建立之 Entra 應用程式的應用程式 （用戶端） ID
- 對象 – 聯合登入資料的對象識別符

步驟 4：建立 IAM 角色

當您透過主控台提交註冊時，會自動建立 IAM 角色。它允許 AWS DevOps 代理程式擔任登入資料並叫用 `sts:GetWebIdentityToken`。

步驟 5：指派角色

請參閱以下指派 Azure 角色。搜尋您在選取成員時建立的 Entra 應用程式。

步驟 6：完成註冊

1. 在 AWS DevOps 代理程式主控台中確認組態
2. 按一下提交以完成註冊

指派 Azure 角色

註冊後，授予應用程式對 Azure 訂閱的讀取存取權。管理員同意和應用程式註冊方法的此步驟相同。

1. 在 Azure 入口網站中，導覽至您的目標訂閱
2. 前往存取控制 (IAM)
3. 按一下新增 > 新增角色指派
4. 選取讀取器角色，然後按一下下一步
5. 按一下選取成員，搜尋應用程式 (AWS DevOps Agent for Admin Consent 或您自己的 Entra 應用程式用於應用程式註冊）
6. 選取應用程式，然後按一下檢閱 + 指派
7. （選用） 若要讓代理程式存取 Azure Kubernetes Service (AKS) 叢集，請完成下列 AKS 存取設定。

> 安全需求：服務主體必須僅指派讀取器角色（以及選擇性地指派下列 AKS 唯讀角色）。Reader 角色可做為安全界限，將代理程式限制為唯讀操作，並限制間接提示注入攻擊的影響。指派具有寫入或動作許可的角色會大幅增加提示注入的爆量半徑，並可能導致 Azure 資源遭到入侵。 AWS DevOps 代理程式只會執行讀取操作。代理程式不會修改、建立或刪除 Azure 資源。

AKS 存取設定 （選用）

步驟 1：Azure Resource Manager (ARM) 層級存取

將 Azure Kubernetes 服務叢集使用者角色指派給應用程式。

在 Azure 入口網站中，前往訂閱 → 選取訂閱 → 存取控制 (IAM) → 新增角色指派 → 選取 Azure Kubernetes 服務叢集使用者角色 → 指派給應用程式 (AWS DevOps Agent for Admin Consent 或您自己的 Entra 應用程式用於應用程式註冊）。

這涵蓋訂閱中的所有 AKS 叢集。若要限定特定叢集的範圍，請改為在資源群組或個別叢集層級指派 。

步驟 2：Kubernetes API 存取

根據叢集的身分驗證組態選擇一個選項：

選項 A：Kubernetes 的 Azure 角色型存取控制 (RBAC)（建議）

1. 如果尚未啟用，請在叢集上啟用 Azure RBAC：Azure 入口網站 → AKS 叢集 → 設定 → 安全組態 → 身分驗證和授權 → 選取 Azure RBAC
2. 指派唯讀角色：Azure 入口網站 → 訂閱 → 選取訂閱 → 存取控制 (IAM) → 新增角色指派 → 選取 Azure Kubernetes Service RBAC 讀取器 → 指派給應用程式

這涵蓋訂閱中的所有 AKS 叢集。

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

如果您的叢集已使用預設的 Azure AD 身分驗證組態，且您不想啟用 Azure RBAC，請使用此選項。這需要每個叢集kubectl的設定。

1. 將下列資訊清單儲存為 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> 以服務主體的物件 ID 取代 。若要尋找它：Azure Portal → Entra ID → 企業應用程式 → 搜尋應用程式名稱 (AWS DevOps Agent for Admin Consent 或您自己的 Entra 應用程式用於應用程式註冊）。
2. 套用至每個叢集：

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

> 注意：僅支援使用本機帳戶的叢集（不含 Azure AD)。建議您在叢集上啟用 Azure AD 整合，以使用此功能。

最低權限自訂角色（選用）

若要加強存取控制，您可以建立自訂 Azure 角色，範圍僅限於資源提供者 AWS DevOps 代理程式使用的 ，而非廣泛的讀取器角色：

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

將訂閱與客服人員空間建立關聯

在帳戶層級註冊 Azure 之後，請將特定訂閱與您的 代理程式空間建立關聯：

1. 在 AWS DevOps 代理程式主控台中，選取您的代理程式空間
2. 前往功能索引標籤
3. 在次要來源區段中，按一下新增
4. 選取 Azure
5. 提供您要關聯的 Azure 訂閱的訂閱 ID
6. 按一下新增以完成關聯

您可以將多個訂閱與相同的代理程式空間建立關聯，讓代理程式在您的 Azure 環境中可見性。

管理 Azure 資源連線

- 檢視連線訂閱 – 在功能索引標籤中，次要來源區段會列出所有連線的 Azure 訂閱。
- 移除訂閱 – 若要中斷訂閱與客服人員空間的連線，請在次要來源清單中選取訂閱，然後按一下移除。這不會影響帳戶層級註冊。
- 移除註冊 – 若要完全移除 Azure 雲端註冊，請前往功能提供者頁面並刪除註冊。必須先移除所有 Agent Space 關聯。

連接 Azure DevOps

Azure DevOps 整合可讓 AWS DevOps Agent 存取 Azure DevOps 組織中的儲存庫和管道執行歷史記錄。代理程式可以將程式碼變更和部署與操作事件建立關聯，以協助識別潛在的根本原因。

> 注意：Azure DevOps 管道可以使用來自 Azure Repos、GitHub 或 Bitbucket 的原始程式碼。無論來源提供者為何，Azure DevOps 整合都可以存取管道執行歷史記錄。不過，若要在調查期間存取實際的原始程式碼，儲存庫必須透過支援的整合分別連線，例如 the section called “連接 GitHub”。Bitbucket 中的原始程式碼無法透過此整合直接存取。

此整合遵循兩個步驟：在 AWS 帳戶層級註冊 Azure DevOps，然後將特定專案與個別客服人員空間建立關聯。

先決條件

連接 Azure DevOps 之前，請確定您擁有：

- 存取 AWS DevOps 代理程式主控台
- 至少有一個專案包含儲存庫和管道歷史記錄的 Azure DevOps 組織
- 將使用者新增至 Azure DevOps 組織的許可
- 針對管理員同意方法：具有在 Microsoft Entra ID 中執行管理員同意許可的帳戶
- 對於應用程式註冊方法：具有設定聯合身分憑證許可的 Entra 應用程式，以及在 AWS 您的帳戶中啟用傳出身分聯合

> 注意：您也可以從 代理程式空間內開始註冊。導覽至管道區段，按一下新增，然後選取 Azure DevOps。如果 Azure DevOps 尚未註冊，主控台會先引導您完成註冊。

透過管理員同意註冊 Azure DevOps

管理員同意方法使用以同意為基礎的流程搭配 AWS DevOps 代理程式受管應用程式。

步驟 1：開始註冊

1. 登入 AWS 管理主控台並導覽至 AWS DevOps 代理程式主控台
2. 前往功能提供者頁面
3. 找到 Azure DevOps 區段，然後按一下註冊
4. 出現提示時輸入您的 Azure DevOps 組織名稱

步驟 2：完成管理員同意

1. 按一下以繼續 - 系統會將您重新導向至 Microsoft Entra 管理員同意頁面
2. 使用具有執行管理員同意許可的使用者主體帳戶登入
3. 檢閱並授予對 AWS DevOps Agent 應用程式的同意

步驟 3：完成使用者授權

1. 管理員同意後，系統會提示您輸入使用者授權，以驗證您的身分為授權租用戶的成員
2. 使用屬於相同 Azure 租用戶的帳戶登入
3. 授權之後，系統會將您重新導向回成功狀態的 AWS DevOps 代理程式主控台

步驟 4：在 Azure DevOps 中授予存取權

請參閱下方的在 Azure DevOps 中授予存取權。在新增使用者時搜尋 AWS DevOps 代理程式。

透過應用程式註冊註冊 Azure DevOps

應用程式註冊是在 Azure 資源和 Azure DevOps 之間共用。如果您已完成 Azure 資源的應用程式註冊，可以跳到在 Azure DevOps 中授予存取權。

步驟 1：啟動 ADO 應用程式註冊

1. 在 AWS DevOps 代理程式主控台中，前往功能提供者頁面
2. 找到 Azure 雲端區段，然後按一下註冊
3. 選取應用程式註冊方法

步驟 2：建立和設定您的 Entra 應用程式

遵循主控台中顯示的指示，以：

1. 在 AWS 帳戶中啟用傳出聯合身分 （ 在 IAM 主控台中，前往帳戶設定 → 傳出聯合身分)
2. 在 Microsoft Entra ID 中建立 Entra 應用程式，或使用現有的應用程式
3. 在應用程式上設定聯合身分憑證

步驟 3：提供註冊詳細資訊

使用下列項目填寫註冊表單：

- 租用戶 ID – Azure 租用戶識別符
- 租用戶名稱 – 租用戶的顯示名稱
- 用戶端 ID – Entra 應用程式的應用程式 （ 用戶端 ） ID
- 對象 – 聯合登入資料的對象識別符

步驟 4：建立 IAM 角色

當您透過主控台提交註冊時，會自動建立 IAM 角色。它允許 AWS DevOps 代理程式擔任登入資料並叫用 `sts:GetWebIdentityToken`。

步驟 5：完成註冊

1. 在 AWS DevOps 代理程式主控台中確認組態
2. 按一下提交以完成註冊

步驟 6：在 Azure DevOps 中授予存取權

請參閱下方的在 Azure DevOps 中授予存取權。在新增使用者時，搜尋您在應用程式註冊期間建立的 Entra 應用程式。

在 Azure DevOps 中授予存取權

註冊後，將應用程式存取權授予 Azure DevOps 組織。管理員同意和應用程式註冊方法的此步驟相同。

1. 在 Azure DevOps 中，前往組織設定 > 使用者 > 新增使用者
2. 搜尋應用程式 (AWS DevOps Agent for Admin Consent 或您自己的 Entra 應用程式用於應用程式註冊）
3. 將存取層級設定為基本
4. 在新增至專案下，選取您希望客服人員存取的專案
5. 在 Azure DevOps 群組下，選取專案讀取器
6. 按一下新增以完成

> 安全需求：僅指派 Project Readers 群組。唯讀存取可做為安全界限，將代理程式限制為唯讀操作，並限制間接提示注入攻擊的影響。指派具有寫入或動作許可的群組會大幅增加提示注入的爆量半徑，並可能導致 Azure DevOps 資源遭到入侵。

將專案與客服人員空間建立關聯

在帳戶層級註冊 Azure DevOps 之後，請將特定專案與您的 代理程式空間建立關聯：

1. 在 AWS DevOps 代理程式主控台中，選取您的代理程式空間
2. 前往功能索引標籤
3. 在管道區段中，按一下新增
4. 從可用提供者清單中選取 Azure DevOps
5. 從可用專案的下拉式清單中選取專案
6. 按一下新增以完成關聯

管理 Azure DevOps 連線

- 檢視連線的專案 – 在功能索引標籤中，管道區段會列出所有連線的 Azure DevOps 專案。

- 移除專案 – 若要中斷專案與客服人員空間的連線，請在管道區段中選取專案，然後按一下移除。
- 移除註冊 – 若要完全移除 Azure DevOps 註冊，請前往功能提供者頁面並刪除註冊。必須先移除所有 Agent Space 關聯。

連線至 CI/CD 管道

CI/CD 管道整合可讓 AWS DevOps 代理程式監控部署，並在調查期間將程式碼變更與操作事件建立關聯。透過連接您的 CI/CD 供應商，代理程式可以追蹤部署事件並將其與 AWS 資源建立關聯，以協助識別事件回應期間的潛在根本原因。

AWS DevOps Agent 透過兩步驟程序支援與熱門 CI/CD 平台整合：

1. 帳戶層級註冊 – 在 AWS 帳戶層級註冊您的 CI/CD 供應商一次
2. 客服人員空間連線 – 根據您的組織需求，將特定專案或儲存庫連線至個別客服人員空間

此方法可讓您跨多個客服人員空間共用 CI/CD 提供者註冊，同時維持每個空間監控哪些專案的精細控制。

支援的 CI/CD 供應商

AWS DevOps Agent 支援下列 CI/CD 平台：

- GitHub – 使用 AWS DevOps Agent GitHub 應用程式從 GitHub.com 連接儲存庫。 GitHub
- GitLab – 從 GitLab.com,受管 GitLab 執行個體或可公開存取的自我託管 GitLab 部署連接專案。

主題

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

連接 GitHub

GitHub 整合可讓 AWS DevOps 代理程式在事件調查期間存取程式碼儲存庫並接收部署事件。此整合遵循兩個步驟：GitHub 的帳戶層級註冊，接著將特定儲存庫連線至個別 Agent Spaces。

AWS DevOps Agent 同時支援 GitHub.com (SaaS) 和 GitHub Enterprise Server （ 自我託管 ） 執行個體。

先決條件

連接 GitHub 之前，請確定您有：

- 存取 AWS DevOps 代理程式管理員主控台
- 具有管理員許可的 GitHub 使用者帳戶或組織
- 在您的帳戶或組織中安裝 GitHub 應用程式的授權

對於 GitHub Enterprise Server，您也需要：

- 透過 HTTPS 存取的 GitHub Enterprise Server 執行個體 (3.x 版或更新版本）
- GitHub Enterprise Server 執行個體的 HTTPS URL（例如 `https://github.example.com`)
- （選用）私有連線，如果您的 GitHub Enterprise Server 執行個體無法公開存取

註冊 GitHub（帳戶層級）

GitHub 已在 AWS 帳戶層級註冊，並與該帳戶中的所有客服人員空間共用。每個 AWS 帳戶只需要註冊一次 GitHub。

步驟 1：導覽至管道供應商

1. 登入 AWS 管理主控台
2. 導覽至 AWS DevOps 代理程式主控台
3. 前往功能索引標籤
4. 在管道區段中，按一下新增
5. 從可用提供者清單中選取 GitHub

如果 GitHub 尚未註冊，系統會提示您先註冊。

步驟 2：選擇連線類型

在「註冊 GitHub 帳戶/組織」畫面上，選取您是以使用者或組織身分連線：

- 使用者 – 具有使用者名稱和設定檔的個人 GitHub 帳戶
- Organization – 共用的 GitHub 帳戶，可讓多個人員一次跨多個專案協作

如果您要連線至 GitHub Enterprise Server 執行個體，請勾選使用 GitHub Enterprise Server 核取方塊，然後輸入執行個體的 HTTPS URL（例如 `https://github.example.com`)。

如果您的 GitHub Enterprise Server 執行個體無法公開存取，您可以選擇設定私有連線，以允許 AWS DevOps 代理程式安全地連線到您的執行個體。如需詳細資訊，請參閱the section called “連線至私有託管工具”。

> **Note**
>
> 請勿在 URL 中包含 `/api/v3`或任何結尾路徑 — 請僅輸入基本 URL。

步驟 3：設定 GitHub 應用程式

按一下提交以開始應用程式設定程序。後續步驟會根據您是否連線到 GitHub.com 或 GitHub Enterprise Server 而有所不同。

對於 GitHub.com

1. 您將重新導向至 GitHub 以安裝 AWS DevOps Agent GitHub 應用程式。
2. 選取要安裝應用程式的帳戶或組織。
3. 應用程式允許 AWS DevOps Agent 從連線的儲存庫接收事件，包括部署事件。

對於 GitHub Enterprise Server

GitHub Enterprise Server 使用 GitHub 應用程式資訊清單流程，自動在您的執行個體上設定新的 GitHub 應用程式。這包括 GitHub Enterprise Server 執行個體的兩個重新導向。

1. 您的瀏覽器將重新導向至 GitHub Enterprise Server 執行個體的「建立 GitHub 應用程式」頁面。
2. 您將看到應用程式名稱已預先填入。您可以視需要變更名稱。按一下建立 GitHub 應用程式。
3. 系統會將您重新導向回 AWS DevOps 代理程式，該代理程式會交換資訊清單程式碼以取得應用程式登入資料。

步驟 4：選取儲存庫並完成安裝

1. 您將看到 GitHub 應用程式的安裝與授權頁面。
2. 選取要允許應用程式存取的儲存庫：
   - 所有儲存庫 – 授予所有目前和未來儲存庫的存取權

- 僅選取儲存庫 – 從您的帳戶或組織選擇特定儲存庫

3. 按一下安裝和授權。
4. 系統會將您重新導向回 AWS DevOps 代理程式主控台，其中 GitHub 會在帳戶層級顯示為已註冊。

將儲存庫連線至 代理程式空間

在帳戶層級註冊 GitHub 之後，您可以將特定儲存庫連線至個別客服人員空間：

1. 在 AWS DevOps 代理程式主控台中，選取您的代理程式空間
2. 前往功能索引標籤
3. 在管道區段中，按一下新增
4. 從可用提供者清單中選取 GitHub
5. 選取與此客服人員空間相關的儲存庫子集
6. 按一下新增以完成連線

您可以根據您的組織需求，將不同的儲存庫集連接到不同的 Agent Spaces。

了解 GitHub 應用程式

The AWS DevOps Agent GitHub 應用程式：

- 請求唯讀存取您的儲存庫
- 接收部署事件和其他儲存庫事件
- 允許 AWS DevOps 代理程式將程式碼變更與操作事件建立關聯
- 您可以隨時透過 GitHub 設定解除安裝

對於 GitHub Enterprise Server，GitHub 應用程式會在註冊期間自動在您的執行個體上建立。您可以透過設定 > 應用程式 > 已安裝的 GitHub 應用程式來管理應用程式的儲存庫存取權或將其解除安裝。若要完全刪除應用程式定義，請前往設定 > 開發人員設定 > GitHub 應用程式。

管理 GitHub 連線

- 更新儲存庫存取 – 若要變更 GitHub 應用程式可存取的儲存庫，請前往您的 GitHub 帳戶或組織設定（或 GitHub Enterprise Server 執行個體設定）、導覽至已安裝的 GitHub 應用程式，以及修改 AWS DevOps Agent 應用程式組態。

- 檢視連線的儲存庫 – 在 AWS DevOps 代理程式主控台中，選取您的代理程式空間，然後前往功能索引標籤，在管道區段中檢視連線的儲存庫。
- 移除 GitHub 連線 – 若要中斷 GitHub 與客服人員空間的連線，請在管道區段中選取連線，然後按一下移除。若要完全解除安裝 GitHub 應用程式，請從 GitHub 帳戶或組織設定解除安裝它。對於 GitHub Enterprise Server，由於 GitHub 應用程式是在註冊期間直接在您的執行個體上建立，因此您可以選擇完全清除應用程式，方法是執行下列兩項操作：
  - 解除安裝應用程式 – 前往設定 > 應用程式 > 安裝的 GitHub 應用程式，按一下應用程式上的設定，然後解除安裝應用程式。
  - 刪除應用程式 – 前往設定 > 開發人員設定 > GitHub 應用程式、選取應用程式、前往進階索引標籤，然後選擇刪除 GitHub 應用程式。警告：刪除 GitHub 應用程式是永久的，無法復原。如果您刪除它，則需要從 AWS DevOps 代理程式主控台的開頭重新註冊 GitHub Enterprise Server，以建立新的應用程式。

連接 GitLab

GitLab 整合可讓 AWS DevOps 代理程式從 GitLab 管道監控部署，以在事件回應期間通知因果調查。此整合遵循兩個步驟：GitLab 的帳戶層級註冊，接著將特定專案連線至個別 Agent Spaces。

註冊 GitLab （帳戶層級）

GitLab 已在 AWS 帳戶層級註冊，並與該帳戶中的所有客服人員空間共用。個別客服人員空間接著可以選擇哪些特定專案適用於其客服人員空間。

步驟 1：導覽至管道供應商

1. 登入 AWS 管理主控台
2. 導覽至 AWS DevOps 代理程式主控台
3. 前往能力提供者頁面 （可從側邊導覽存取）
4. 在管道下的可用供應商區段中尋找 GitLab，然後按一下註冊

步驟 2：設定 GitLab 連線

在 GitLab 註冊頁面上，設定下列項目：

連線類型 – 選取您是以個人或群組身分連線：

- 個人 （預設） – 具有使用者名稱和設定檔的個別 GitLab 使用者帳戶

- 群組 – 在 GitLab 中，您可以使用群組來同時管理一或多個相關專案

GitLab 執行個體類型 – 選擇您要連接的 GitLab 執行個體類型：

- GitLab.com （預設） – 公有 GitLab 服務
- 可公開存取的自我託管 GitLab – 檢查使用 GitLab 自我託管端點方塊，並將 URL 提供給 GitLab 執行個體

> **Note**
>
> 目前僅支援可公開存取的 GitLab 執行個體。

存取字符 – 提供 GitLab 個人存取字符：

1. 在單獨的瀏覽器索引標籤中，登入您的 GitLab 帳戶
2. 導覽至您的使用者設定，然後選取存取字符
3. 使用下列許可建立新的個人存取字符：
   - `read_repository` – 存取儲存庫內容時需要
   - `read_virtual_registry` – 存取虛擬登錄檔資訊時需要
   - `read_registry` – 存取登錄檔資訊時需要
   - `api` – 讀取和寫入 API 存取時需要
   - `self_rotate` - 輪換權杖時需要。 AWS DevOps 代理程式目前不支援此功能，但稍後會支援此功能。立即新增 可避免未來需要建立新的權杖。
4. 將字符過期設定為從目前日期起算最多 365 天
5. 複製產生的字符
6. 返回 AWS DevOps 代理程式主控台
7. 將字符貼到「存取字符」欄位中

步驟 3：完成註冊

（選用） 標籤 – 將 AWS 標籤新增至 GitLab 註冊以供組織使用。

按一下下一步以檢閱您的組態，然後按一下提交以完成 GitLab 註冊程序。系統會驗證您的存取權杖並建立連線。

將專案連線至 代理程式空間

在帳戶層級註冊 GitLab 之後，您可以將特定專案連線到個別的 Agent Spaces：

1. 在 AWS DevOps 代理程式主控台中，選取您的代理程式空間
2. 前往功能索引標籤
3. 在管道區段中，按一下新增
4. 從可用提供者清單中選取 GitLab
5. 選取與您的客服人員空間相關的 GitLab 專案
6. 按一下儲存

AWS DevOps 代理程式將監控這些專案的 GitLab Pipelines 部署，以通知因果調查。

管理 GitLab 連線

- 更新存取權杖 – 如果您的存取權杖過期或需要更新，您可以在帳戶層級修改 GitLab 註冊，以在 AWS DevOps 代理程式主控台中更新它。
- 檢視連線的專案 – 在 AWS DevOps 代理程式主控台中，選取您的代理程式空間，然後前往功能索引標籤，在管道區段中檢視連線的專案。
- 移除 GitLab 連線 – 若要中斷 GitLab 專案與客服人員空間的連線，請在管道區段中選取連線，然後按一下移除。若要完全移除 GitLab 註冊，請先將其從所有 Agent Spaces 中移除，然後在帳戶層級刪除註冊。

連接 MCP 伺服器

模型內容通訊協定 (MCP) 伺服器透過提供來自外部可觀測性工具、自訂監控系統和操作資料來源的資料存取權，來擴展 AWS DevOps 代理程式的調查功能。本指南說明如何將 MCP 伺服器連線至 AWS DevOps Agent。

要求

連接 MCP 伺服器之前，請確定您的伺服器符合下列要求：

- 可串流 HTTP 傳輸通訊協定 – 僅支援實作可串流 HTTP 傳輸通訊協定的 MCP 伺服器。
- 身分驗證支援 – MCP 伺服器必須支援 OAuth 2.0 身分驗證流程或 API 金鑰/金鑰型身分驗證。

安全考量

將 MCP 伺服器連線至 AWS DevOps 代理程式時，請考慮下列安全層面：

- 工具允許清單 – 您應該只允許列出 Agent Space 所需的特定工具，而不是從 MCP 伺服器公開所有工具。如需如何允許每個客服人員空間的清單工具，請參閱在客服人員空間中設定 MCP 工具。

請注意，任何 MCP 工具的工具長度上限為 64。

- 提示注入風險 – 自訂 MCP 伺服器可能會帶來提示注入攻擊的額外風險。如需詳細資訊，請參閱提示注入保護： AWS DevOps 代理程式安全。
- 唯讀工具和存取 – 僅允許列出唯讀 MCP 工具，並確保身分驗證登入資料只允許唯讀存取。

AWS DevOps 代理程式安全性 如需提示注入和共同責任模型的詳細資訊，請參閱 。

> **Note**
>
> 如果您的 MCP 伺服器位於私有網路上，請參閱 the section called “連線至私有託管工具”

註冊 MCP 伺服器 （帳戶層級）

MCP 伺服器會在 AWS 帳戶層級註冊，並在該帳戶中的所有客服人員空間之間共用。個別客服人員空間接著可以從每個 MCP 伺服器選擇他們所需的特定工具。

步驟 1：MCP 伺服器詳細資訊

1. 登入 AWS 管理主控台
2. 導覽至 AWS DevOps 代理程式主控台
3. 前往能力提供者頁面 （可從側邊導覽存取）
4. 在可用供應商區段中尋找 MCP 伺服器，然後按一下註冊
5. 在 MCP 伺服器詳細資訊頁面上，輸入下列資訊：
   - 名稱 – 輸入 MCP 伺服器的描述性名稱
   - 端點 URL – 輸入 MCP 伺服器端點的完整 HTTPS URL
   - 描述 （選用） – 新增描述以協助識別伺服器的目的

- 啟用動態用戶端註冊 – 如果您想要允許 AWS DevOps 代理程式自動向 MCP 伺服器的授權伺服器註冊，請選取此核取方塊

6. 按一下下一步

> **Note**
>
> MCP 伺服器端點 URL 會顯示在您帳戶中的 AWS CloudTrail 日誌中。

步驟 2：授權流程

選取 MCP 伺服器的身分驗證方法：

OAuth 用戶端登入資料 – 如果您的 MCP 伺服器使用 OAuth 用戶端登入資料流程：

1. 選取 OAuth 用戶端登入資料
2. 按一下下一步

OAuth 3LO （三引號 OAuth) – 如果您的 MCP 伺服器使用 OAuth 3LO 進行身分驗證：

1. 選取 OAuth 3LO
2. 按一下下一步

API 金鑰 – 如果您的 MCP 伺服器使用 API 金鑰身分驗證：

1. 選取 API 金鑰
2. 按一下下一步

步驟 3：授權組態

根據選取的身分驗證方法設定其他授權參數：

對於 OAuth 用戶端登入資料：

1. 用戶端 ID – 輸入 OAuth 用戶端的用戶端 ID
2. 用戶端秘密 – 輸入 OAuth 用戶端的用戶端秘密
3. Exchange URL – 輸入 OAuth 權杖交換端點 URL

4. Exchange 參數 – 輸入 OAuth 權杖交換參數以使用 服務進行驗證
5. 新增範圍 – 新增身分驗證的 OAuth 範圍
6. 按一下下一步

對於 OAuth 3LO：

1. 用戶端 ID – 輸入 OAuth 用戶端的用戶端 ID
2. 用戶端秘密 – 如果您的 OAuth 用戶端需要，請輸入 OAuth 用戶端的用戶端秘密
3. Exchange URL – 輸入 OAuth 權杖交換端點 URL
4. 授權 URL - 輸入 OAuth 授權端點 URL
5. 程式碼挑戰支援 - 如果您的 OAuth 用戶端支援程式碼挑戰，請選取此核取方塊
6. 新增範圍 – 新增身分驗證的 OAuth 範圍
7. 按一下下一步

針對 API 金鑰：

1. 輸入 API 金鑰名稱
2. 輸入將在請求中包含 API 金鑰的標頭名稱
3. 輸入您的 API 金鑰值
4. 按一下下一步

步驟 4：檢閱並提交

1. 檢閱所有 MCP 伺服器組態詳細資訊
2. 按一下提交以完成註冊
3. AWS DevOps 代理程式將驗證與您的 MCP 伺服器的連線
4. 成功驗證後，您的 MCP 伺服器將在帳戶層級註冊

在客服人員空間中設定 MCP 工具

在帳戶層級註冊 MCP 伺服器之後，您可以設定該伺服器的哪些工具可供特定 Agent Spaces 使用：

1. 在 AWS DevOps 代理程式主控台中，選取您的代理程式空間

2. 前往功能索引標籤
3. 在 MCP 伺服器區段中，按一下新增
4. 選取您要連線至此客服人員空間的已註冊 MCP 伺服器
5. 設定 Agent Space 應可從此 MCP 伺服器使用哪些工具：
   - 允許所有工具 – 讓 MCP 伺服器中的所有工具都可用
   - 選取特定工具 – 可讓您選擇要允許清單的工具
6. 按一下新增，將 MCP 伺服器連線至您的客服人員空間

AWS DevOps 代理程式現在可以在此代理程式空間的調查期間，使用 MCP 伺服器的允許清單工具。

管理 MCP 伺服器連線

更新身分驗證憑證 – 如果您的身分驗證憑證需要更新，您將需要重新註冊 MCP 伺服器。導覽至 AWS DevOps 代理程式主控台中的功能提供者頁面，尋找您的 MCP 伺服器，移除任何作用中的關聯，然後按一下取消註冊。接著，使用新的身分驗證登入資料註冊您的 MCP 伺服器，然後重新建立與 Agent Space 的任何必要關聯。

檢視連線的 MCP 伺服器 – 若要查看所有連線至客服人員空間的 MCP 伺服器，請選取您的客服人員空間，前往功能索引標籤，然後檢查 MCP 伺服器區段。您也可以在這裡更新選取的工具。

移除 MCP 伺服器連線 – 若要中斷 MCP 伺服器與客服人員空間的連線，請在 MCP 伺服器區段中選取伺服器，然後按一下移除。若要完全刪除 MCP 伺服器註冊，請先將其從所有客服人員空間中移除，然後刪除帳戶層級註冊。

相關主題

- 安全 in AWS DevOps 代理程式
- 設定 代理程式空間
- 提示注入保護

連接多個 AWS 帳戶

次要 AWS 帳戶可讓 AWS DevOps 代理程式調查組織中多個 AWS 帳戶的資源。當您的應用程式跨越多個帳戶時，新增次要帳戶可確保客服人員在事件調查期間可查看所有相關資源。對構成應用程式的帳戶和資源的更多存取可確保更高的調查準確性。

先決條件

新增次要 AWS 帳戶之前，請確定您有：

- 存取主要帳戶中的 AWS DevOps 代理程式主控台
- 次要 AWS 帳戶的管理存取權
- 在次要帳戶中建立角色的 IAM 許可

新增次要 AWS 帳戶

除了下列步驟之外，您還可以使用 the section called “AWS DevOps Agent CLI 入門指南”以程式設計方式新增次要帳戶。

步驟 1：啟動次要帳戶組態

1. 登入 AWS 管理主控台並導覽至 AWS DevOps 代理程式主控台
2. 選取您的客服人員空間
3. 前往功能索引標籤
4. 在雲端區段中，尋找次要來源子區段
5. 按一下新增

步驟 2：指定角色名稱

1. 在為您的角色命名欄位中，輸入您將在次要帳戶中建立的角色名稱
2. 請注意此名稱 - 在次要帳戶中建立角色時，您將再次使用它
3. 複製主控台中提供的信任政策，並將其儲存在暫存空間中

步驟 3：在次要帳戶中建立角色

1. 開啟新的瀏覽器索引標籤，並登入次要 AWS 帳戶中的 IAM 主控台
2. 導覽至 IAM >角色 > 建立角色
3. 選取自訂信任政策
4. 貼上您從步驟 2 複製的信任政策
5. 按一下下一步

步驟 4：連接 AWS 受管政策

1. 在許可政策區段中，搜尋 AIOpsAssistantPolicy
2. 選取 AIOpsAssistantPolicy 受管政策旁的核取方塊
3. 按一下下一步

步驟 5：命名並建立角色

1. 在角色名稱欄位中，輸入您在步驟 2 中提供的相同角色名稱
2. （選用）新增描述以協助識別角色的目的
3. 檢閱信任政策和連接的許可
4. 按一下建立角色

步驟 6：連接內嵌政策

1. 在 IAM 主控台中，尋找並選取您剛建立的角色
2. 前往許可索引標籤
3. 按一下新增許可 > 建立內嵌政策
4. 切換到 JSON 索引標籤
5. 貼上您在步驟 2 中儲存的政策
6. 將政策貼到 IAM 主控台的 JSON 編輯器
7. 按一下下一步
8. 提供內嵌政策的名稱（例如「DevOpsAgentInlinePolicy」）
9. 按一下建立政策

步驟 7：完成組態

1. 返回主要帳戶中的 AWS DevOps 代理程式主控台
2. 按一下下一步以完成次要帳戶組態
3. 確認連線狀態顯示為作用中

了解必要的政策

AWS DevOps Agent 需要三個政策元件，才能存取次要帳戶中的資源：

- 信任政策 – 允許主要帳戶中的 AWS DevOps 代理程式擔任次要帳戶中的角色。這會建立帳戶之間的信任關係。
- AIOpsAssistantPolicy (AWS 受管政策） – 提供 AWS DevOps 代理程式調查次要帳戶中資源所需的核心唯讀許可。此政策由 維護 AWS ，並在新增新功能時更新。
- 內嵌政策 – 提供代理程式空間組態特定的其他許可。此政策會根據您的客服人員空間設定產生，並可能包含特定整合或功能的許可。

在主要帳戶中， AWS DevOps 代理程式 IAM 角色必須能夠擔任在次要帳戶中建立的角色。

管理次要帳戶

- 檢視連線帳戶 – 在功能索引標籤中，次要來源子區段會列出所有連線的次要帳戶及其連線狀態。
- 更新 IAM 角色 – 如果您需要修改許可，請更新連接到次要帳戶中角色的內嵌政策。變更會立即生效。
- 移除次要帳戶 – 若要中斷連接次要帳戶，請在次要來源清單中選取該帳戶，然後按一下移除。這不會刪除次要帳戶中的 IAM 角色。

連接遙測來源

AWS DevOps Agent 提供三種連線至遙測來源的方式。

內建的雙向整合

目前， AWS DevOps Agent 支援具有內建雙向整合的 Dynatrace 使用者，可啟用下列項目：

- 拓撲資源映射 - AWS DevOps 代理程式會透過 a AWS DevOps 代理程式託管的 Dynatrace MCP 伺服器，使用可用的實體和關係來增強您的 DevOps 代理程式空間拓撲。
- 自動化調查觸發 - 可以設定 Dynatrace 工作流程來觸發來自 Dynatrace 問題的事件解決調查。
- 遙測自我檢查 - AWS DevOps 代理程式可以透過 AWS DevOps 代理程式託管的 Dynatrace MCP 伺服器調查問題時，可以自我檢查 Dynatrace 遙測。
- 狀態更新 - AWS DevOps 代理程式會將金鑰調查結果、根本原因分析和產生的緩解計畫發佈至 Dynatrace 使用者介面。

若要了解雙向整合，請參閱

- the section called “連接 Dynatrace”

內建的單向整合

目前， AWS DevOps Agent 支援具有內建的單向整合的 AWS CloudWatch、Datadog、Grafana、New Relic 和 Splunk 使用者。

安全最佳實務：設定內建單向整合的登入資料時，我們建議將 API 金鑰和權杖範圍限定為唯讀存取。AWS DevOps 代理程式只會使用這些登入資料進行遙測自我檢查，不需要對遙測供應商進行寫入存取。

AWS CloudWatch 內建的單向整合不需要額外設定，並啟用下列項目：

- 拓撲資源映射 - AWS DevOps 代理程式將透過您設定的主要和次要 AWS 雲端帳戶，使用可用的實體和關係來增強您的 DevOps 代理程式空間拓撲。
- 遙測自我檢查 - AWS DevOps 代理程式可以透過主要和次要 AWS 雲端帳戶組態期間提供的 IAM 角色 (IAM) 調查問題時，可以自我檢查 AWS CloudWatch 遙測。

Datadog、Grafana、New Relic 和 Splunk 內建的 1 種方式整合需要設定並啟用下列項目：

- 自動化調查觸發 - 可以設定 Datadog、Grafana、New Relic 和 Splunk 事件來觸發透過 AWS DevOps 代理程式 Webhook 的 AWS DevOps 代理程式事件解決調查。
- 遙測自我檢查 - AWS DevOps 代理程式可以在透過每個供應商的遠端 MCP 伺服器調查問題時，自我檢查 Datadog、Grafana、New Relic 和 Splunk 遙測。

若要了解單向整合，請參閱以下內容：

- the section called “連接 DataDog”
- the section called “連接 Grafana”
- the section called “連接新複本”
- the section called “連接 Splunk”

Bring-your-own遙測來源

對於任何其他遙測來源，包括 Prometheus 指標，您可以利用 AWS DevOps 代理程式對 Webhook 和 MCP 伺服器整合的支援。

若要了解bring-your-own整合，請參閱以下內容

- the section called “透過 Webhook 叫用 DevOps 代理程式”
- the section called “連接 MCP 伺服器”

連接 Dynatrace

內建的雙向整合

目前， AWS DevOps Agent 支援具有內建雙向整合的 Dynatrace 使用者，可啟用下列項目：

- 拓撲資源映射 - AWS DevOps 代理程式將使用 Dynatrace 環境中可用的實體和關係來增強您的 DevOps 代理程式空間拓撲。
- 自動化調查觸發 - 可以設定 Dynatrace 工作流程來觸發來自 Dynatrace 問題的事件解決調查。
- 遙測自我檢查 - AWS DevOps 代理程式可以透過 AWS DevOps 代理程式託管的 Dynatrace MCP 伺服器調查問題時，可以自我檢查 Dynatrace 遙測。
- 狀態更新 - AWS DevOps 代理程式會將金鑰調查結果、根本原因分析和產生的緩解計畫發佈至 Dynatrace 使用者介面。

上線

加入程序

加入您的 Dynatrace 可觀測性系統包含三個階段：

1. Connect - 透過設定帳戶存取登入資料與您可能需要的所有環境來建立與 Dynatrace 的連線
2. 啟用 - 在具有特定 Dynatrace 環境的特定代理程式空間中啟用 Dynatrace
3. 設定您的 Dynatrace 環境 - 下載工作流程和儀表板，並匯入 Dynatrace，記下 Webhook 詳細資訊，以觸發指定客服人員空間中的調查

步驟 1：連線

建立與 Dynatrace 環境的連線

Configuration

1. 前往能力提供者頁面（可從側邊導覽存取）
2. 在遙測下的可用提供者區段中尋找 Dynatrace，然後按一下註冊
3. 在 Dynatrace 中建立具有詳細許可的 OAuth 用戶端。
   a. 請參閱 Dynatrace 文件
   b. 準備就緒後，按下一步
   c. 您可以針對您可能擁有的每個 DevOps Agent 空間，將多個 Dynatrace 環境和更新範圍連接到特定環境。
4. 從 OAuth 用戶端設定輸入您的 Dynatrace 詳細資訊：
   - 用戶端名稱
   - 用戶端 ID
   - 用戶端秘密
   - 帳戶 URN
5. 按一下下一步
6. 檢閱並新增

步驟 2：啟用

在特定客服人員空間中啟用 Dynatrace，並設定適當的範圍

Configuration

1. 從客服人員空間頁面，選取客服人員空間並按下檢視詳細資訊
2. 選取功能索引標籤
3. 找到遙測區段，按下新增
4. 您會注意到 Dynatrace 狀態為「已註冊」。按一下新增以將此新增至您的客服人員空間
5. Dynatrace 環境 ID - 提供您要與此 DevOps 代理程式空間建立關聯的 Dynatrace 環境 ID。
6. 輸入一或多個 Dynatrace IDs - 這些可協助 DevOps 代理程式探索您最重要的資源，範例可能是服務或應用程式。如果您不確定可以按移除。
7. 檢閱並按下儲存
8. 複製 Webhook URL 和 Webhook 秘密。請參閱 Dynatrace 文件，將這些登入資料新增至 Dynatrace。

步驟 3：設定您的 Dynatrace 環境

若要完成 Dynatrace 設定，您需要在 Dynatrace 環境中執行特定設定步驟。遵循 Dynatrace 文件中的指示。

支援的事件結構描述

AWS DevOps Agent 使用 Webhook 支援來自 Dynatrace 的兩種事件類型。支援的事件結構描述記錄如下：

事件事件

事件會用來觸發調查。事件結構描述為：

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

緩解事件

緩解事件用於觸發產生緩解報告以進行後續步驟的調查。事件結構描述為：

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

移除

遙測來源在客服人員空間層級和帳戶層級的兩個層級上連接。若要完全移除它，您必須先從使用它的所有代理程式空間中移除它，然後可以取消註冊。

步驟 1：從客服人員空間移除

1. 從客服人員空間頁面，選取客服人員空間並按下檢視詳細資訊
2. 選取功能索引標籤
3. 向下捲動至遙測區段
4. 選取 Dynatrace
5. 按移除

步驟 2：從帳戶取消註冊

1. 前往能力提供者頁面（可從側邊導覽存取）
2. 捲動至目前註冊的區段。
3. 檢查客服人員空間計數為零（如果不是在其他客服人員空間中重複上述步驟 1)
4. 按下 Dynatrace 旁的取消註冊

連接 DataDog

內建的單向整合

目前， AWS DevOps Agent 支援具有內建的單向整合的 Datadog 使用者，啟用下列項目：

- 自動化調查觸發 - 資料狗事件可設定為觸發透過 AWS DevOps 代理程式 Webhook 的 AWS DevOps 代理程式事件解決調查。
- 遙測自我檢查 - AWS DevOps 代理程式可以在透過每個提供者的遠端 MCP 伺服器調查問題時自我檢查資料狗遙測。

上線

步驟 1：連線

使用帳戶存取憑證建立與 Datadog 遠端 MCP 端點的連線

Configuration

1. 前往能力提供者頁面（可從側邊導覽存取）
2. 在遙測下的可用提供者區段中尋找 Datadog，然後按一下註冊

3. 輸入您的 Datadog MCP 伺服器詳細資訊：
   - 伺服器名稱 - 唯一識別符（例如 my-datadog-server)
   - 端點 URL - Datadog MCP 伺服器端點。端點 URL 會根據 Datadog 網站而有所不同。請參閱下面的 Datadog 網站端點表格。
   - 描述 - 選用的伺服器描述
4. 按一下下一步
5. 檢閱並提交

Datadog 網站端點

MCP 端點 URL 會根據 Datadog 網站而有所不同。若要識別您的網站，請在登入 Datadog 時檢查瀏覽器中的 URL，或參閱存取 Datadog 網站。

Datadog 網站	網站網域	MCP 端點 URL
US1（預設）	`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/un`

Datadog 網站	網站網域	MCP 端點 URL
		`stable/mcp-server/mcp`
AP2	`ap2.datadoghq.com`	`https://mcp.ap2.datadoghq.com/api/unstable/mcp-server/mcp`

Authorization

透過下列方式完成 OAuth 授權：

- 在 Datadog OAuth 頁面上以您的使用者身分授權
- 如果未登入，請按一下允許、登入，然後授權

設定完成後，Datadog 即可在所有客服人員空間中使用。

步驟 2：啟用

在特定客服人員空間中啟用 DataDog，並設定適當的範圍

Configuration

1. 從客服人員空間頁面，選取客服人員空間並按下檢視詳細資訊（如果您尚未建立客服人員空間，請參閱 the section called “建立 代理程式空間”）
2. 選取功能索引標籤
3. 向下捲動至遙測區段
4. 按下新增
5. 選取 Datadog
6. 下一頁
7. 檢閱並按下儲存
8. 複製 Webhook URL 和 API 金鑰

步驟 3：設定 Webhook

使用 Webhook URL 和 API 金鑰，您可以設定 Datadog 傳送事件以觸發調查，例如從警示。

為了確保 DevOps 代理程式可以使用傳送的事件，請確定傳輸至 Webhook 的資料符合下列指定的資料結構描述。DevOps 代理程式可能會忽略不符合此結構描述的事件。

設定 方法和標頭

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

以 JSON 字串傳送內文。

```
{
    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;
}
```

使用 Datadog https://docs.datadoghq.com/integrations/webhooks/ 傳送 Webhook （請注意，選取不授權，而是使用自訂標頭選項）。

進一步了解：Datadog 遠端 MCP 伺服器

移除

遙測來源在客服人員空間層級和帳戶層級的兩個層級上連接。若要完全移除它，您必須先從使用它的所有代理程式空間中移除它，然後可以取消註冊。

步驟 1：從客服人員空間移除

1. 從客服人員空間頁面，選取客服人員空間並按下檢視詳細資訊

2. 選取功能索引標籤
3. 向下捲動至遙測區段
4. 選取 Datadog
5. 按移除

步驟 2：從帳戶取消註冊

1. 前往能力提供者頁面（可從側邊導覽存取）
2. 捲動至目前註冊的區段。
3. 檢查客服人員空間計數為零（如果不是在其他客服人員空間中重複上述步驟 1)
4. 按下 Datadog 旁的取消註冊

連接 Grafana

Grafana 整合可讓 AWS DevOps 代理程式在事件調查期間查詢 Grafana 執行個體中的指標、儀表板和提醒資料。此整合遵循兩個步驟：Grafana 的帳戶層級註冊，然後將其連接到個別客服人員空間。

為了提高安全性，Grafana 整合僅啟用唯讀工具。寫入工具已停用且無法啟用。這表示代理程式可以從 Grafana 執行個體查詢和讀取資料，但無法建立、修改或刪除任何 Grafana 資源，例如儀表板、提醒或註釋。如需詳細資訊，請參閱 Security in AWS DevOps Agent。

Grafana 要求

連接 Grafana 之前，請確定您有：

- Grafana 9.0 版或更新版本。由於缺少 API 端點，某些功能，特別是與資料來源相關的操作，可能無法正確使用舊版。
- 可透過 HTTPS 存取的 Grafana 執行個體。同時支援公有和私有網路端點。透過私有網路連線，您的 Grafana 執行個體可以在沒有公有網際網路存取的 VPC 中託管。如需詳細資訊，請參閱the section called “連線至私有託管工具”。
- 具有適當讀取許可之存取權杖的 Grafana 服務帳戶

註冊 Grafana（帳戶層級）

Grafana 已在 AWS 帳戶層級註冊，並在該帳戶中的所有客服人員空間之間共用。

步驟 1：設定 Grafana

1. 登入 AWS 管理主控台
2. 導覽至 AWS DevOps 代理程式主控台
3. 前往能力提供者頁面（可從側邊導覽存取）
4. 在遙測下的可用提供者區段中尋找 Grafana，然後按一下註冊
5. 在設定 Grafana 頁面上，輸入下列資訊：
   - 服務名稱（必要）– 使用英數字元、連字號和底線輸入 Grafana 伺服器的描述性名稱。例如 `my-grafana-server`。
   - Grafana URL（必要）– 輸入 Grafana 執行個體的完整 HTTPS URL。例如 `https://myinstance.grafana.net`。
   - 服務帳戶存取字符（必要）– 輸入 Grafana 服務帳戶存取字符。字符通常以 開頭`glsa_`。若要建立服務帳戶字符，請導覽至您的 Grafana 執行個體，前往管理 > 服務帳戶，使用檢視器角色建立服務帳戶，並產生字符。
   - 描述（選用）– 新增描述以協助識別伺服器的目的。例如 `Production Grafana server for monitoring`。
6. （選用）將 AWS 標籤新增至註冊以供組織使用。
7. 按一下下一步

步驟 2：檢閱並提交 Grafana 註冊

1. 檢閱所有 Grafana 組態詳細資訊
2. 按一下提交以完成註冊
3. 成功註冊後，Grafana 會出現在能力提供者頁面的目前註冊區段中

將 Grafana 新增至代理程式空間

在帳戶層級註冊 Grafana 之後，您可以將其連線至個別客服人員空間：

1. 在 AWS DevOps 代理程式主控台中，選取您的代理程式空間
2. 前往功能索引標籤
3. 在遙測區段中，按一下新增
4. 從可用供應商清單中選取 Grafana

5. 按一下儲存

設定 Grafana 警示 Webhook

您可以將 Grafana 設定為在透過 Grafana 聯絡點傳送 Webhook 來提醒火災時自動觸發 AWS DevOps 代理程式調查。如需 Webhook 身分驗證方法和憑證管理的詳細資訊，請參閱 the section called “透過 Webhook 叫用 DevOps 代理程式”。

步驟 1：建立自訂通知範本

在您的 Grafana 執行個體中，導覽至警示 > 聯絡點 > 通知範本，並使用下列內容建立新的範本：

```
{{ 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 }}
```

此範本會將 Grafana 警示格式化為 AWS DevOps 代理程式預期的 Webhook 承載結構。它會將提醒標籤、註釋和狀態映射到適當的欄位中，並包含所有提醒標籤作為中繼資料。

> 注意：此範本只會處理群組中的第一個提醒。根據預設，Grafana 會將多個射擊警示分組為單一通知。若要確保個別傳送每個提醒，請將您的通知政策設定為依 分組`alertname`。此外，此範本不會逸出標籤值或註釋中的特殊 JSON 字元。確保提醒標籤和`summary`註釋不包含雙引號或換行等字元，這會產生無效的 JSON。

步驟 2：建立 Webhook 聯絡點

1. 在 Grafana 中，導覽至警示 > 聯絡點，然後按一下新增聯絡點
2. 選取 Webhook 做為整合類型
3. 將 URL 設定為 your AWS DevOps Agent Webhook 端點
4. 在選用 Webhook 設定下，根據您的 Webhook 類型設定身分驗證標頭。如需詳細資訊，請參閱 Webhook 身分驗證方法。
5. 設定訊息欄位以使用您的自訂範本：`{{ template "devops-agent-payload" . }}`
6. 按一下儲存聯絡點

步驟 3：將聯絡點指派給通知政策

1. 導覽至警示 > 通知政策
2. 編輯現有政策或建立新的政策
3. 將聯絡點設定為您建立的 Webhook 聯絡點
4. 按一下儲存政策

當相符的提醒觸發時，Grafana 會將格式化的承載傳送至 AWS DevOps 代理程式，自動開始調查。

限制

- ClickHouse 資料來源工具 – 目前不支援 ClickHouse 資料來源工具。
- 主動預防事件 – 目前the section called “主動事件預防”不使用 Grafana 工具。未來版本已規劃支援。

Amazon Managed Grafana 考量事項

如果您使用的是 Amazon Managed Grafana (AMG)，請注意下列限制：

- 不支援 Webhook 聯絡點 – AMG 目前在其提醒組態中不支援 Webhook 聯絡點。您無法使用 AMG 將警示 Webhook 直接傳送到 AWS DevOps 代理程式。如需詳細資訊，請參閱 Amazon Managed Grafana 中的警示聯絡點。
- 服務帳戶字符過期 – AMG 服務帳戶字符的過期天數上限為 30 天。您將需要輪換權杖，並在權杖過期之前更新 Grafana 註冊 in AWS DevOps 代理程式。如需如何更新登入資料，請參閱管理 Grafana 連線。如需 AMG 字符限制的詳細資訊，請參閱 Amazon Managed Grafana 中的服務帳戶。

管理 Grafana 連線

- 更新登入資料 – 如果您的服務帳戶字符過期或需要更新，請從能力提供者頁面取消註冊 Grafana，並使用新字符重新註冊。
- 檢視連線的執行個體 – 在 AWS DevOps 代理程式主控台中，選取您的代理程式空間，然後前往功能索引標籤以檢視連線的遙測來源。
- 移除 Grafana – 若要中斷 Grafana 與客服人員空間的連線，請在遙測區段中選取它，然後按一下移除。若要完全移除註冊，請先將其從所有客服人員空間中移除，然後從能力提供者頁面取消註冊。

連接新複本

內建的單向整合

目前， AWS DevOps Agent 支援具有內建的單向整合的新 Relic 使用者，啟用下列項目：

- 自動化調查觸發 - 新複本事件可設定為透過 AWS DevOps 代理程式 Webhook 觸發 AWS DevOps 代理程式事件解決調查。
- 遙測自我檢查 - AWS DevOps 代理程式可以在透過每個提供者的遠端 MCP 伺服器調查問題時自我檢查新舊式遙測。

上線

步驟 1：連線

使用帳戶存取憑證建立與 New Relic 遠端 MCP 端點的連線

請在新複本中使用完整平台使用者 （非基本/核心） 來啟用新複本 MCP 工具。

Configuration

1. 前往能力提供者頁面 （可從側邊導覽存取）
2. 在遙測下的可用提供者區段中尋找新複本，然後按一下註冊
3. 依照指示取得您的新複本 API 金鑰
4. 輸入您的 New Relic MCP 伺服器 API 金鑰詳細資訊：
   - 帳戶 ID：輸入上述取得的新複本帳戶 ID
   - API 金鑰：輸入上述取得的 API 金鑰

- 根據新複本帳戶所在的位置選取美國或歐洲區域。

5. 按一下新增

步驟 2：啟用

在特定客服人員空間中啟用新複本，並設定適當的範圍

Configuration

1. 從客服人員空間頁面，選取客服人員空間並按下檢視詳細資訊（如果您尚未建立客服人員空間，請參閱 the section called “建立 代理程式空間”）
2. 選取功能索引標籤
3. 向下捲動至遙測區段
4. 按下新增
5. 選取新複本
6. 下一頁
7. 檢閱並按下儲存
8. 複製 Webhook URL 和 API 金鑰

步驟 3：設定 Webhook

使用 Webhook URL 和 API 金鑰，您可以設定 New Relic 傳送事件以觸發調查，例如從警示。如需設定 Webhook 的詳細資訊，請參閱變更追蹤 Webhook。

為了確保 DevOps 代理程式可以使用傳送的事件，請確定傳輸至 Webhook 的資料符合下列指定的資料結構描述。DevOps 代理程式可能會忽略不符合此結構描述的事件。

設定 方法和 標頭

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

以 JSON 字串傳送內文。

```
{
    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;
}
```

使用 New Relic https://newrelic.com/instant-observability/webhook-notifications 傳送 Webhook。您可以為授權類型選取承載字符，或選取無授權，並改為新增 `Authorization: Bearer <Token>` 做為自訂標頭。

進一步了解：https：//https://docs.newrelic.com/docs/agentic-ai/mcp/overview/

移除

遙測來源在客服人員空間層級和帳戶層級的兩個層級上連接。若要完全移除它，您必須先從使用它的所有代理程式空間中移除它，然後可以取消註冊。

步驟 1：從客服人員空間移除

1. 從客服人員空間頁面，選取客服人員空間並按下檢視詳細資訊
2. 選取功能索引標籤
3. 向下捲動至遙測區段
4. 選取新複本
5. 按移除

步驟 2：從帳戶取消註冊

1. 前往能力提供者頁面（可從側邊導覽存取）
2. 捲動至目前註冊的區段。
3. 檢查客服人員空間計數為零（如果不是在其他客服人員空間中重複上述步驟 1)
4. 按下新複本旁的取消註冊

連接 Splunk

內建的單向整合

目前， AWS DevOps Agent 支援具有內建單向整合的 Splunk 使用者，啟用下列項目：

- 自動化調查觸發 - Splunk 事件可設定為透過 AWS DevOps 代理程式 Webhook 觸發 AWS DevOps 代理程式事件解決調查。
- 遙測自我檢查 - AWS DevOps 代理程式可以透過每個供應商的遠端 MCP 伺服器調查問題，來自我檢查 Splunk 遙測。

先決條件

取得 Splunk API 字符

您需要 MCP URL 和字符才能連接 Splunk。

Splunk 管理員步驟

您的 Splunk 管理員需要執行下列步驟：

- 啟用 REST API 存取
- 在 部署上啟用權杖身分驗證。
- 建立新的角色 'mcp_user'，新角色不需要有任何功能。
- 會將角色 'mcp_user' 指派給部署上獲授權使用 MCP 伺服器的任何使用者。
- 如果使用者沒有建立字符本身的許可，請為對象為 'mcp' 的授權使用者建立字符，並設定適當的過期時間。

Splunk 使用者步驟

Splunk 使用者需要執行下列步驟：

- 向 Splunk 管理員取得適當的權杖，如果他們具有 許可，則自行建立權杖。字符的對象必須是 'mcp'。

上線

步驟 1：連線

使用帳戶存取憑證建立與 Splunk 遠端 MCP 端點的連線

Configuration

1. 前往能力提供者頁面（可從側邊導覽存取）
2. 在遙測下的可用供應商區段中尋找 Splunk，然後按一下註冊
3. 輸入您的 Splunk MCP 伺服器詳細資訊：
   - 伺服器名稱 - 唯一識別符（例如 my-splunk-server)
   - 端點 URL - Splunk MCP 伺服器端點：

```
https://<YOUR_SPLUNK_DEPLOYMENT_NAME>.api.scs.splunk.com/
<YOUR_SPLUNK_DEPLOYMENT_NAME>/mcp/v1/
```

- 描述 - 選用的伺服器描述
- 權杖名稱 - 用於身分驗證的承載權杖名稱： `my-splunk-token`
- 權杖值 身分驗證的承載權杖值

步驟 2：啟用

在特定客服人員空間中啟用 Splunk，並設定適當的範圍

Configuration

1. 從客服人員空間頁面，選取客服人員空間並按下檢視詳細資訊（如果您尚未建立客服人員空間，請參閱 the section called “建立 代理程式空間”)
2. 選取功能索引標籤
3. 向下捲動至遙測區段
4. 按下新增
5. 選取 Splunk
6. 下一頁
7. 檢閱並按下儲存

8. 複製 Webhook URL 和 API 金鑰

步驟 3：設定 Webhook

使用 Webhook URL 和 API 金鑰，您可以設定 Splunk 傳送事件以觸發調查，例如從警示。

為了確保 DevOps 代理程式可以使用傳送的事件，請確定傳輸至 Webhook 的資料符合下列指定的資料結構描述。DevOps 代理程式可能會忽略不符合此結構描述的事件。

設定 方法和標頭

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

以 JSON 字串傳送內文。

```
{
    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;
}
```

使用 Splunk https://help.splunk.com/en/splunk-enterprise/alert-and-respond/alerting-manual/9.4/configure-alert-actions/use-a-webhook-alert-action 傳送 Webhook （請注意，選取不授權，而是使用自訂標頭選項）

進一步了解：

- Splunk 的 MCP 伺服器文件：https：//https://help.splunk.com/en/splunk-cloud-platform/mcp-server-for-splunk-platform/about-mcp-server-for-splunk-platform

- Splunk Cloud Platform REST API 的存取需求和限制：https：//https://docs.splunk.com/Documentation/SplunkCloud/latest/RESTTUT/RESTandCloud
- 在 Splunk Cloud Platform 中管理身分驗證字符：https：//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
- 使用 Splunk Web 建立和管理角色：https：//https://docs.splunk.com/Documentation/SplunkCloud/latest/Security/Addandeditroles

移除

遙測來源在客服人員空間層級和帳戶層級的兩個層級上連接。若要完全移除它，您必須先將其從使用它的所有代理程式空間中移除，然後才能取消註冊。

步驟 1：從客服人員空間移除

1. 從客服人員空間頁面，選取客服人員空間並按下檢視詳細資訊
2. 選取功能索引標籤
3. 向下捲動至遙測區段
4. 選取 Splunk
5. 按移除

步驟 2：從帳戶取消註冊

1. 前往能力提供者頁面（可從側邊導覽存取）
2. 捲動至目前註冊的區段。
3. 檢查客服人員空間計數為零（如果不是在其他客服人員空間中重複上述步驟 1)
4. 按下 Splunk 旁的取消註冊

連線至票證和聊天

AWS DevOps 代理程式旨在透過參與團隊現有的溝通管道，成為您團隊的成員。您可以將 DevOps Agent 連線至您的票證和警示系統，例如 ServiceNow 和 PagerDuty，以自動從事件票證啟動調查，加速現有工作流程中的事件回應，以減少預期復原時間 (MTTR)。您也可以將 DevOps 代理程式連線至 Slack 等團隊協作系統，以在聊天頻道中接收來自 DevOps 代理程式的活動摘要。

若要了解如何連接票證和聊天整合，請參閱以下內容：

- the section called “連接 PagerDuty”
- the section called “連接 ServiceNow”
- the section called “連接 Slack”

連接 PagerDuty

PagerDuty 整合可讓 AWS DevOps 代理程式在事件調查和自動回應期間，從您的 PagerDuty 帳戶存取和更新事件資料、通話中排程和服務資訊。此整合使用 OAuth 2.0 進行安全身分驗證。

> **Important**
>
> AWS DevOps 代理程式僅支援較新的 PagerDuty OAuth 2.0（範圍 OAuth)。不支援具有重新導向 uri 的舊版 PagerDuty OAuth。

PagerDuty 需求

在連接 PagerDuty 之前，請確定您已：

- 具有您的 OAuth 用戶端 ID 和用戶端秘密的 PagerDuty 帳戶
- 您的 PagerDuty 帳戶子網域（例如，如果您的 PagerDuty URL 為 `https://your-company.pagerduty.com`，則子網域為 `your-company`)

註冊 PagerDuty

PagerDuty 已在 AWS 帳戶層級註冊，並在該帳戶中的所有客服人員空間之間共用。

步驟 1：在 PagerDuty 中設定存取權

1. 登入 AWS 管理主控台
2. 導覽至 AWS DevOps 代理程式主控台
3. 前往能力提供者頁面（可從側邊導覽存取）
4. 在通訊下的可用供應商區段中尋找 PagerDuty，然後按一下註冊
5. 遵循在 PagerDuty 中設定存取權頁面上的引導設定：

檢查您的服務區域和子網域：

- 帳戶範圍 – 選取您的 PagerDuty 區域 (美國或歐洲)，然後輸入您的 PagerDuty 子網域。例如，如果您的 PagerDuty URL 為 `https://your-company.pagerduty.com`，請輸入 `your-company`。

在 PagerDuty 中建立新的應用程式：

- 在單獨的瀏覽器索引標籤中，登入 PagerDuty 並導覽至整合 > 應用程式註冊
- 使用 OAuth 2.0 範圍 OAuth 建立新的應用程式
- 在許可下，授予下列最低必要範圍：`incidents.write`、 `incidents.read`和 `services.read`
- 啟用事件整合以允許 AWS DevOps Agent 和 PagerDuty 之間的雙向通訊

設定 OAuth 登入資料：

- 許可範圍 – 所需的最低範圍為：`incidents.read`、`incidents.write`、 `services.read`
- 用戶端名稱 – 輸入 OAuth 用戶端的描述性名稱
- 用戶端 ID – 輸入來自 PagerDuty 應用程式註冊的 OAuth 用戶端 ID
- 用戶端秘密 – 輸入來自 PagerDuty 應用程式註冊的 OAuth 用戶端秘密

步驟 2：檢閱並提交 PagerDuty 註冊

1. 檢閱所有 PagerDuty 組態詳細資訊
2. 按一下提交以完成註冊
3. 成功註冊後，PagerDuty 會出現在能力提供者頁面的目前註冊區段中

將 PagerDuty 新增至客服人員空間

在帳戶層級註冊 PagerDuty 之後，您可以將其連線至個別客服人員空間：

1. 在 AWS DevOps 代理程式主控台中，選取您的代理程式空間
2. 前往功能索引標籤
3. 在通訊區段中，按一下新增
4. 從可用提供者清單中選取 PagerDuty
5. 按一下儲存

管理 PagerDuty 連線

- 更新登入資料 – 如果您的 OAuth 登入資料需要更新，請從能力提供者頁面取消註冊 PagerDuty，並使用新的登入資料重新註冊。
- 檢視連線 – 在 AWS DevOps 代理程式主控台中，選取您的代理程式空間，然後前往功能索引標籤以檢視連線的通訊整合。
- 移除 PagerDuty – 若要中斷 PagerDuty 與客服人員空間的連線，請在通訊區段中選取它，然後按一下移除。若要完全移除註冊，請先將其從所有客服人員空間中移除，然後從能力提供者頁面取消註冊。

Webhook 支援

AWS DevOps 代理程式僅支援 PagerDuty V3 Webhook。不支援較舊的 Webhook 版本。

如需 PagerDuty V3 Webhook 訂閱的詳細資訊，請參閱 PagerDuty 開發人員文件中的 Webhooks 概觀。

連接 ServiceNow

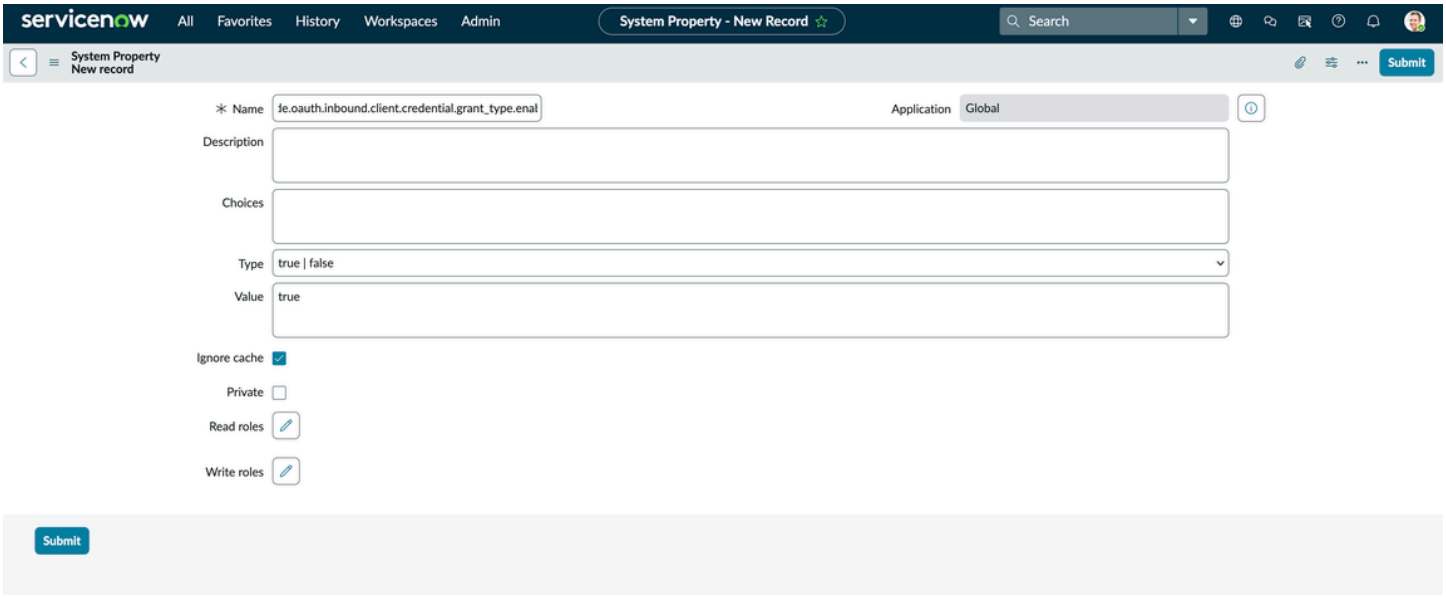
本教學課程會逐步引導您將 ServiceNow 執行個體連線至 AWS DevOps 代理程式，讓它在建立票證時自動啟動事件回應調查，並將其金鑰調查結果發佈至原始票證。它還包含如何設定 ServiceNow 執行個體僅將特定票證傳送至 DevOps 代理程式空間，以及如何協調多個 DevOps 代理程式空間之間的票證路由的範例。

初始設定

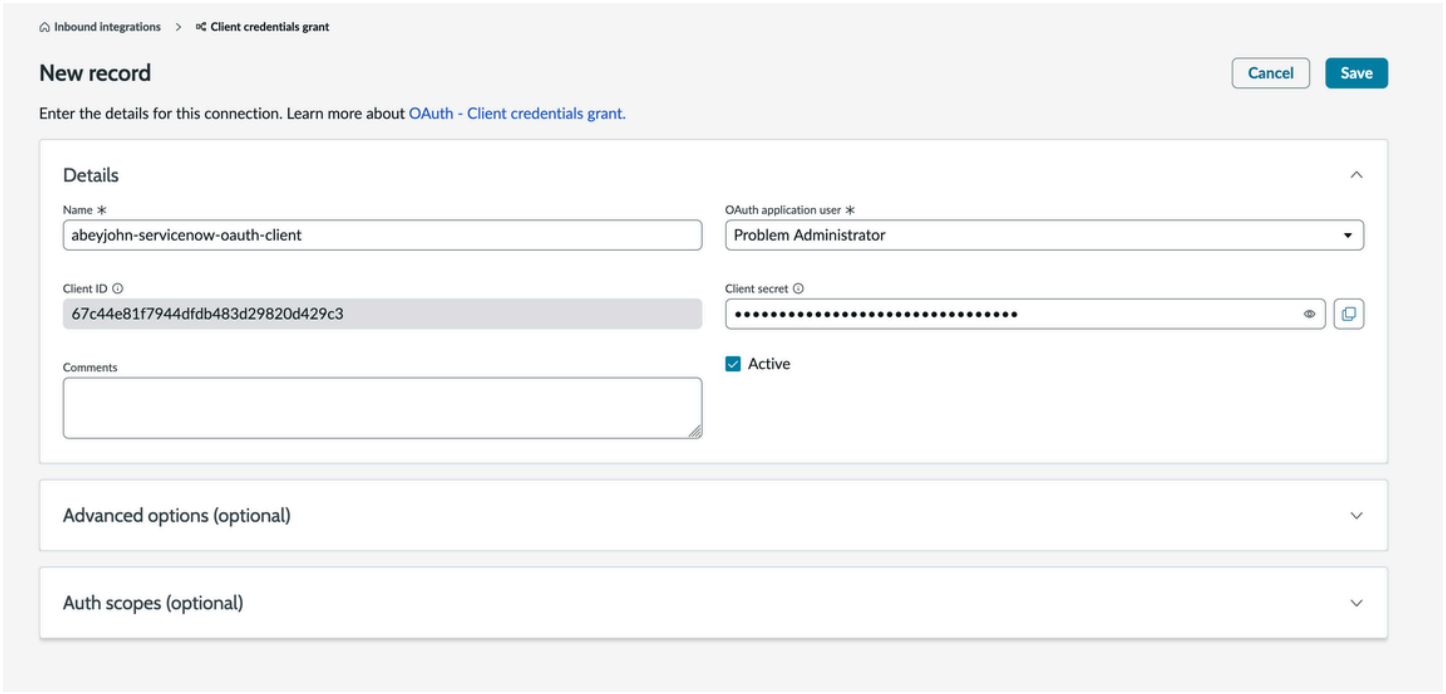
第一步是在 ServiceNow 中建立 OAuth 應用程式用戶端， AWS DevOps 可用來存取您的 ServiceNow 執行個體。

建立 ServiceNow OAuth 應用程式用戶端

1. 啟用執行個體的用戶端登入資料系統屬性
   a. `sys_properties.list` 在篩選條件搜尋方塊中搜尋，然後按 Enter 鍵 （它不會顯示 選項，但按 Enter 鍵有效）
   b. 選擇新增
   c. 將名稱新增為 ，`glide.oauth.inbound.client.credential.grant_type.enabled`並將值新增為 true，類型為 true | false



1. 從篩選條件搜尋方塊中導覽至系統 OAuth > 應用程式登錄檔
2. 選擇「新」>「新傳入整合體驗」>「新整合」>「OAuth - 用戶端憑證授予」
3. 選擇名稱並將 OAuth 應用程式使用者設定為「問題管理員」，然後按一下「儲存」

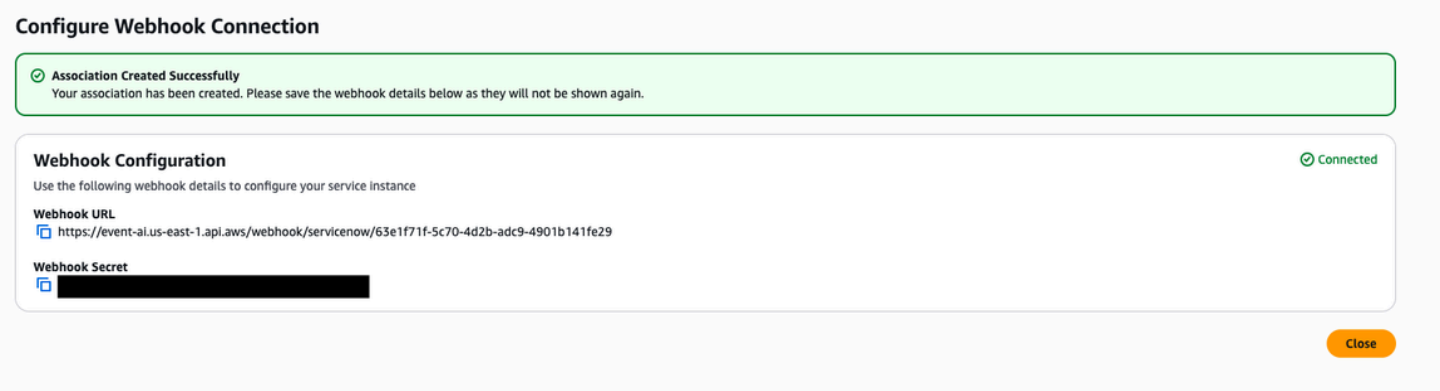


將您的 ServiceNow OAuth 用戶端連線至 AWS DevOps 代理程式

1. 您可以從兩個位置開始此程序。首先，導覽至能力提供者頁面並在通訊下尋找 ServiceNow，然後按一下註冊。或者，您可以選擇您可能已建立的任何 DevOps 代理程式空間，並導覽至功能 → 通訊 → 新增 → ServiceNow，然後按一下註冊。
2. 接著，授權 DevOps Agent 使用您剛建立的 OAuth 應用程式用戶端存取您的 ServiceNow 執行個體。

- 遵循後續步驟，並儲存 Webhook 的產生資訊

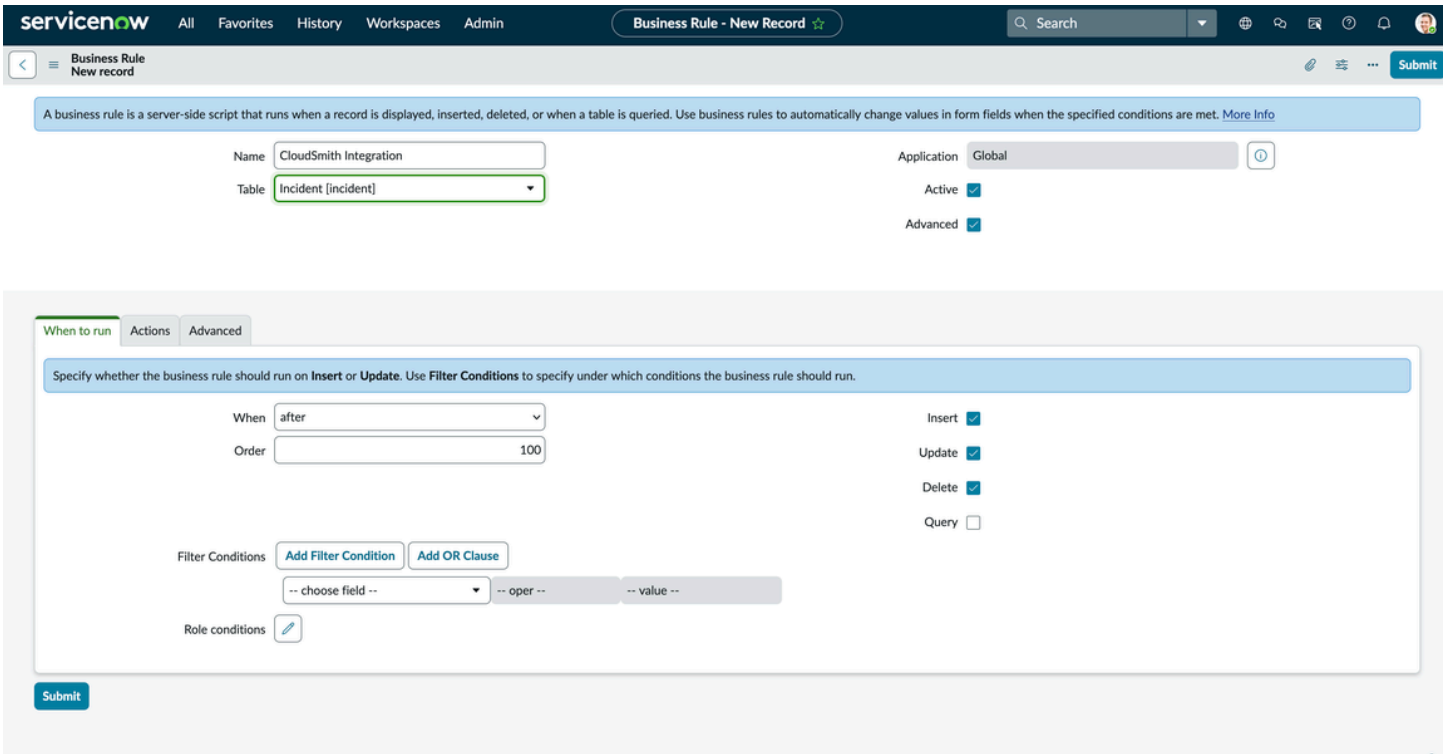
> **Important**
>
> 您不會再看到此資訊



設定您的 ServiceNow 業務規則

建立連線後，您需要在 ServiceNow 中設定商業規則，將票證傳送至 DevOps Agent Space(s)。

1. 導覽至活動訂閱 → 管理 → 業務規則，然後按一下新增。
2. 將「資料表」欄位設定為「事件 【事件】」，勾選「進階」方塊，並將規則設定為在插入、更新和刪除之後執行。



1. 導覽至「進階」索引標籤，並新增下列 Webhook 指令碼，在指示的位置插入您的 Webhook 秘密和 URL，然後按一下提交。

```
(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 {
```

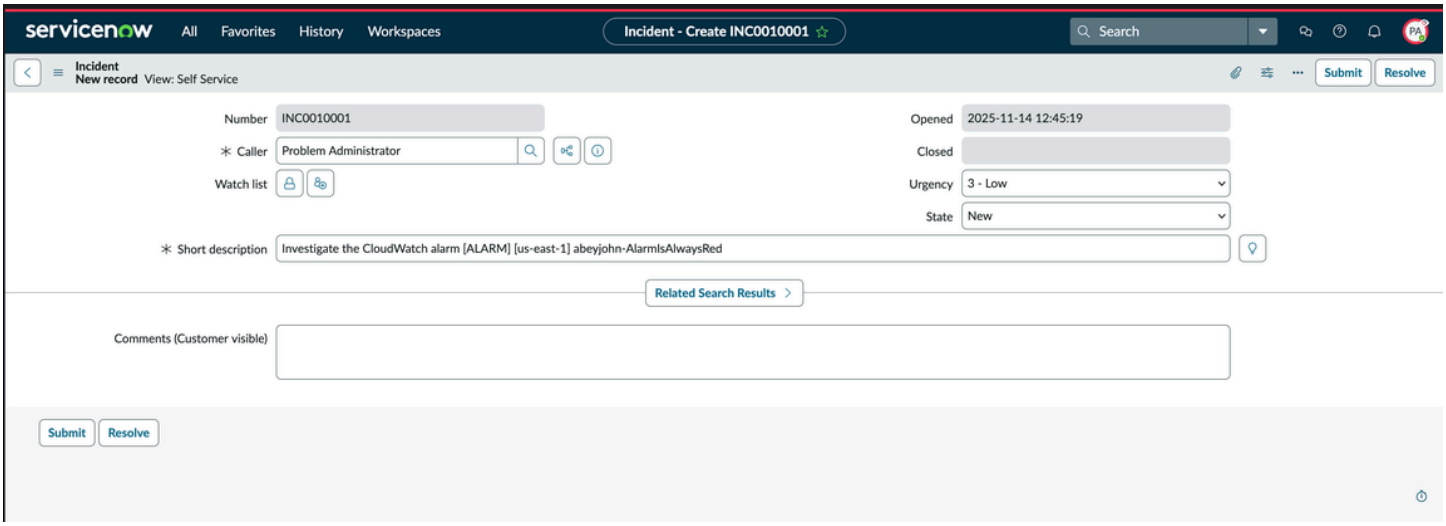
```
            var mac = new GlideCertificateEncryption();
            var signature = mac.generateMac(secret, "HmacSHA256", payloadString);
            return signature;
        } catch (e) {
            gs.error('HMAC generation failed: ' + e);
            return null;
        }
    }

    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);
```

如果您選擇從功能提供者頁面註冊 ServiceNow 連線，您現在需要導覽至要調查 ServiceNow 事件票證的 DevOps 代理程式空間，選取功能 → 通訊，然後在功能提供者頁面註冊您註冊的 ServiceNow 執行

個體。現在，應該設定所有內容，而且呼叫者設定為「問題管理員」的所有事件（模擬您給予 AWS DevOps OAuth 用戶端的許可）都會在設定的 DevOps 代理程式空間中觸發事件回應調查。您可以在 ServiceNow 中建立新的事件，並將事件的來電者欄位設定為「問題管理員」，以測試這一點。



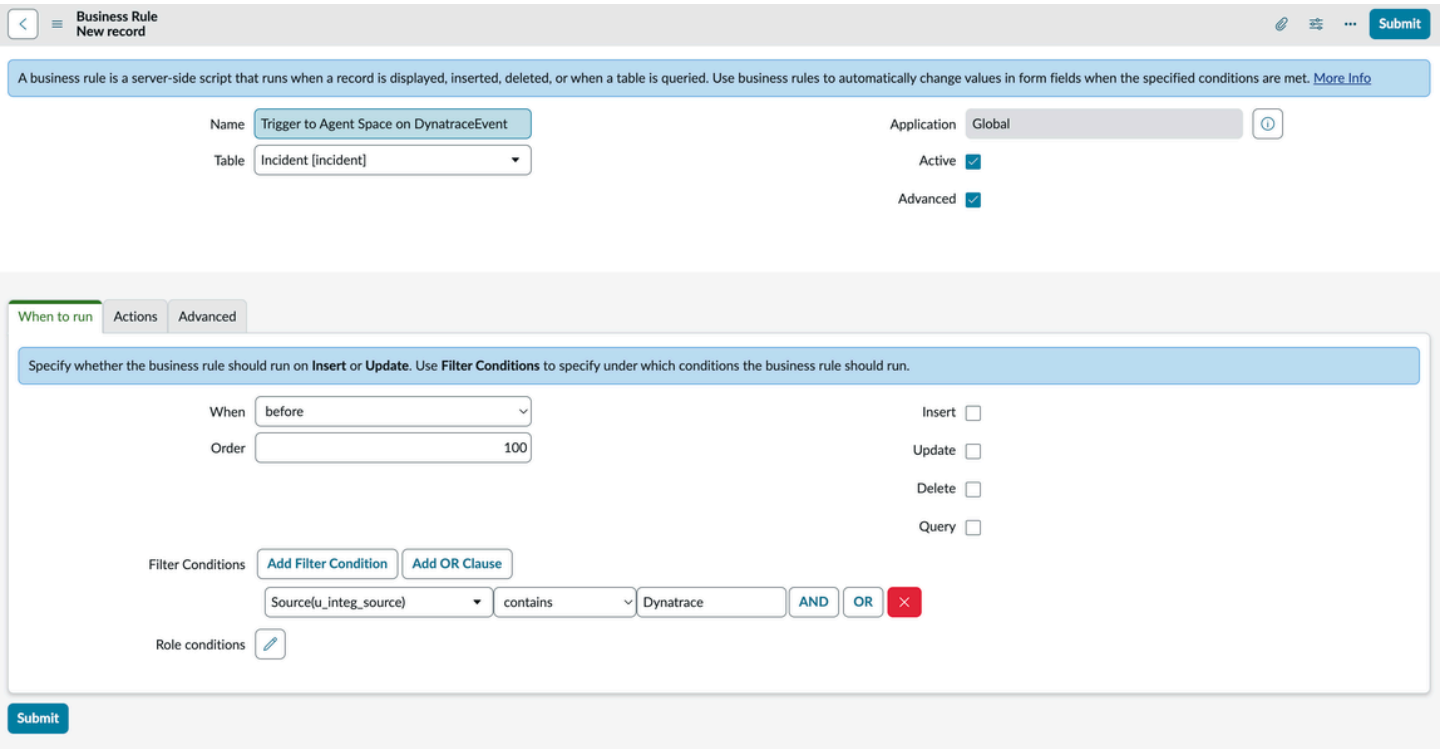
ServiceNow 票證更新

在所有觸發的事件回應調查期間，DevOps 代理程式會在原始票證中提供其關鍵調查結果、根本原因分析和緩解計劃的更新。客服人員調查結果會張貼到事件的註解，我們目前只會張貼類型為 `finding`、`cause`、、 `investigation_summary` `mitigation_summary` 和調查狀態更新的客服人員記錄（例如 `AWS DevOps Agent started/finished its investigation`)。

票證路由和協同運作範例

案例：篩選哪些事件會傳送到 DevOps 代理程式空間

這是簡單的案例，但在 ServiceNow 中需要一些組態，才能在 ServiceNow 中建立欄位來追蹤事件來源。基於此範例的目的，請使用 SNOW 表單建置器建立新的來源 (u_source) 欄位。這將允許追蹤事件來源，並使用它將請求從特定來源路由到 DevOps 代理程式空間。路由的完成方式是建立 Service Now Business Rule，並在何時執行索引標籤中設定「何時」觸發條件和「篩選條件」。在此範例中，篩選條件的設定如下：

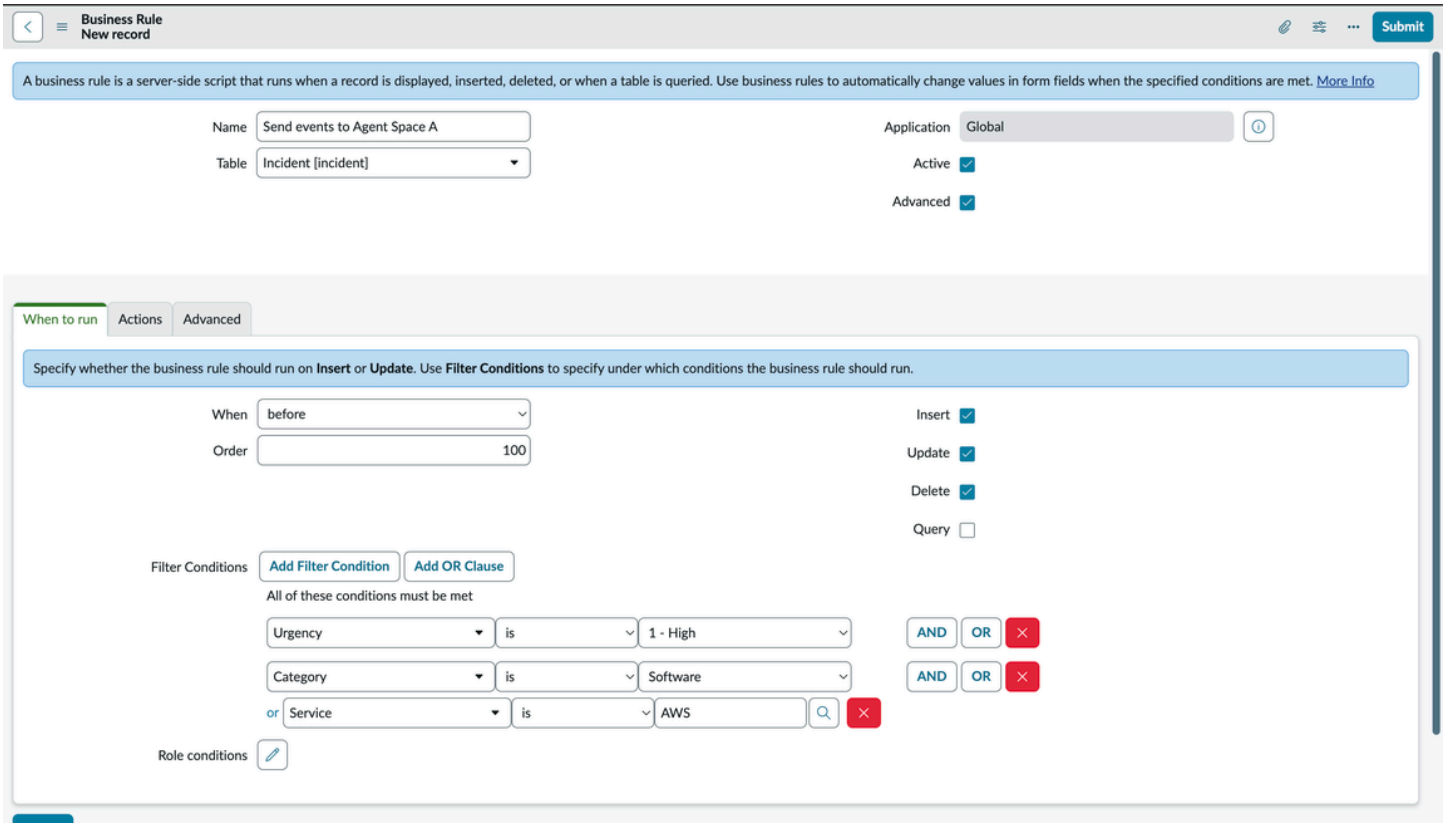


案例：跨多個 DevOps 代理程式空間路由事件

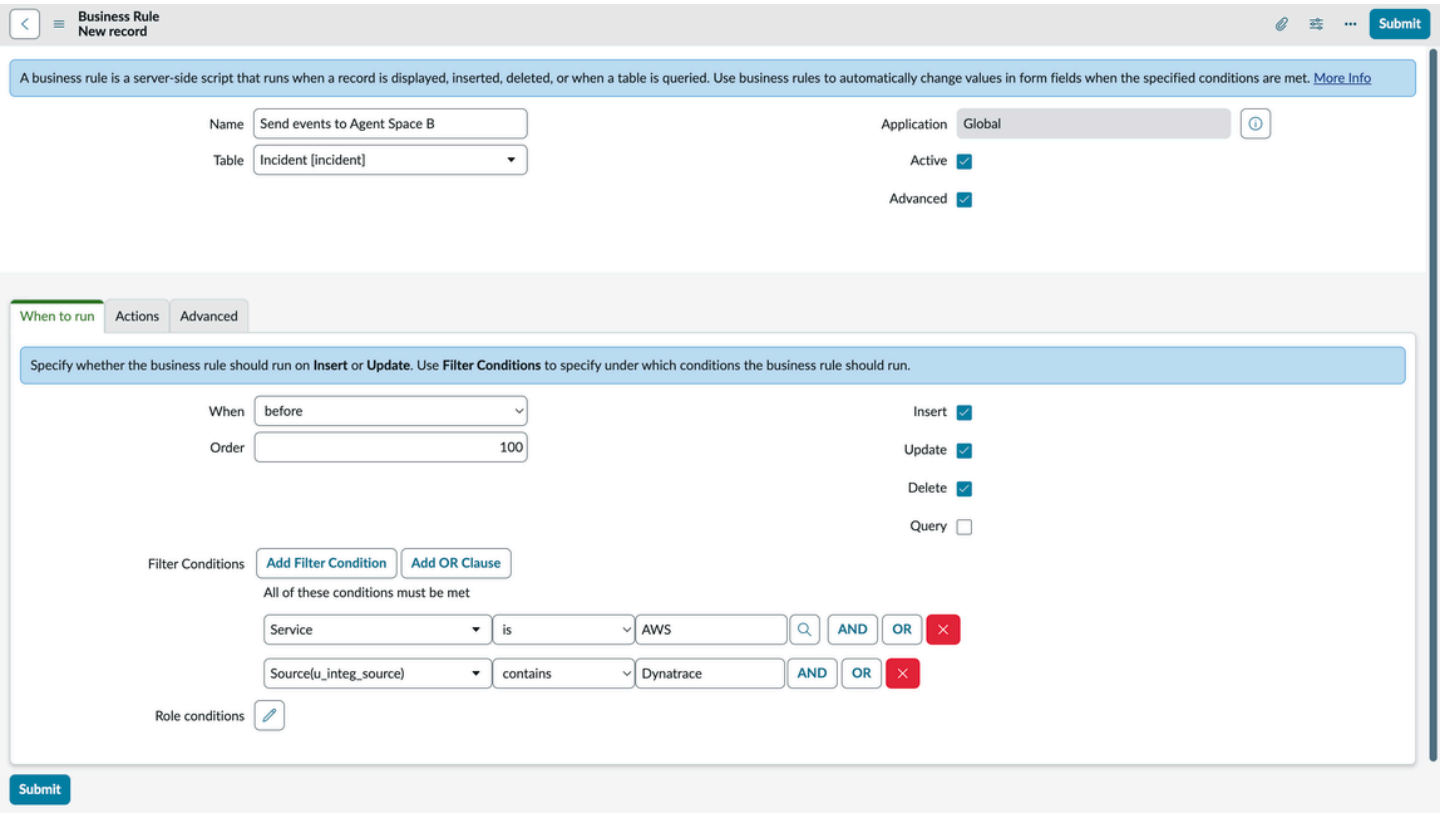
此範例示範如何在緊急程度為 1、類別為 `Software` 或服務為 時觸發 DevOps Agent Space B 中的調查`AWS`，以及在服務為 `AWS`且來源為 時觸發 DevOps Agent Space A 中的調查`Dynatrace`。

此案例可以透過兩種方式完成。Webhook 指令碼本身可以更新為包含此商業邏輯。在此案例中，我們將示範如何使用 ServiceNow 業務規則來完成它，以實現透明度並簡化偵錯。路由是透過建立兩個 Service Now Business 規則來完成。

- 在 ServiceNow for DevOps Agent Space A 中建立業務規則，並使用 條件建置器建立條件，只根據我們指定的條件傳送事件。



- 接著，在 ServiceNow for AgentSpace B 中建立另一個業務規則，只有在服務為 AWS 且來源為 Dynatrace 時，才會觸發業務規則。



現在，當您建立符合指定條件的新事件時，它會觸發對 DevOps Agent Space A 或 DevOps Agent Space B 的調查，為您提供對事件路由的精細控制。

連接 Slack

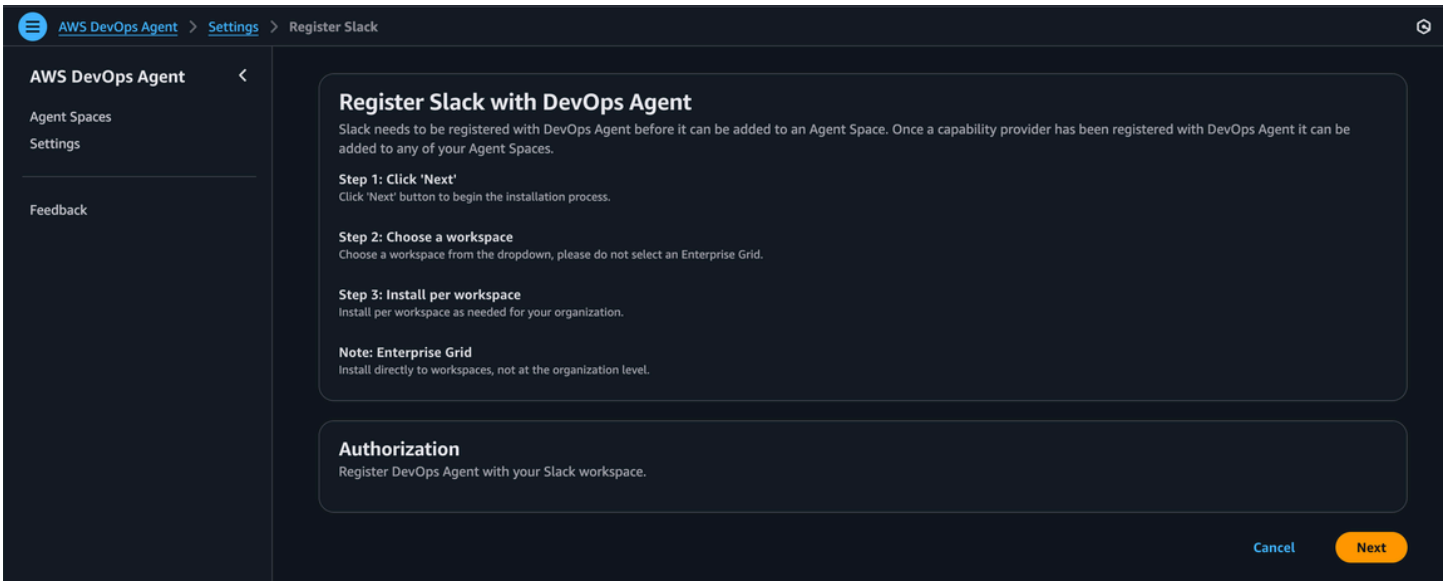
您可以設定 AWS DevOps 代理程式，透過事件回應調查金鑰調查結果、根本原因分析和產生的緩解計劃來更新您選取的 Slack 頻道。

開始之前

Slack 需要先向 DevOps Agent 註冊，才能新增至 Agent Space。若要將 AWS DevOps 代理程式與 Slack 整合，您必須符合下列要求：

- 可存取 Slack 工作區，並能夠安裝和授權第三方應用程式
- 已識別您希望 AWS DevOps 代理程式傳送通知的 Slack 管道

註冊與 AWS DevOps 代理程式的 Slack 整合



1. 從 AWS DevOps 代理程式主控台的功能提供者頁面，在通訊下的可用提供者區段中尋找 Slack，然後按一下註冊。
2. 選擇註冊按鈕。
3. 系統會將您重新導向至 Slack，為您的工作區授權 AWS DevOps 代理程式應用程式。
4. 在 Slack 授權頁面上，將 直接安裝到工作區，而不是組織層級。
5. 從下拉式清單中選擇工作區。請勿選取企業網格。
6. 視需要為您的組織在每個工作區安裝 。
7. 檢閱請求的範圍，然後按一下允許以授權整合。
8. 授權之後，您將返回 AWS DevOps 代理程式主控台。

將 Slack 與您的 DevOps Agent Space(s) 建立關聯

在 DevOps Agent Space (DevOps Agent Space) 中註冊 Slack 之後，您可以將其與 DevOps Agent Space 建立關聯：

1. 從已設定 AgentSpace 的功能索引標籤中，導覽至通訊 > Slack。
2. 選取新增 Slack
3. 輸入頻道 ID
4. 選擇建立以完成 Slack 組態。

> **Note**
>
> 代理程式的機器人使用者必須新增至私有通道，才能發佈訊息。

> **Important**
>
> 解除安裝 Slack 應用程式可能會導致無法重新安裝 Slack 應用程式。請避免解除安裝 Slack 應用程式。

透過 Webhook 叫用 DevOps 代理程式

Webhook 可讓外部系統自動觸發 AWS DevOps 代理程式調查。這可整合票證系統、監控工具，以及可在事件發生時傳送 HTTP 請求的其他平台。

先決條件

在設定 Webhook 存取之前，請確定您有：

- 在 AWS DevOps 代理程式中設定的代理程式空間
- 存取 AWS DevOps 代理程式主控台
- 將傳送 Webhook 請求的外部系統

Webhook 類型

AWS DevOps 代理程式支援下列類型的 Webhook：

- 整合特定的 Webhook – 當您設定第三方整合時自動產生，例如 Dynatrace、Splunk、Datadog、New Relic、ServiceNow 或 Slack。這些 Webhook 與特定整合相關聯，並使用整合類型決定的身分驗證方法
- 一般 Webhook – 可以手動建立，以觸發來自特定整合未涵蓋之任何來源的調查。一般 Webhook 目前使用 HMAC 身分驗證 （目前無法使用承載字符）。
- Grafana 提醒 Webhook – Grafana 可以透過 Webhook 聯絡點將提醒通知直接傳送至 AWS DevOps 代理程式。如需包含自訂通知範本的設定說明，請參閱連接 Grafana。

Webhook 身分驗證方法

Webhook 的身分驗證方法取決於其與哪個整合相關聯：

HMAC 身分驗證 – 用於：

- Dynatrace 整合 Webhook
- 一般 Webhook （未連結至特定的第三方整合）

承載字符身分驗證 – 用於：

- Splunk 整合 Webhook
- Datadog 整合 Webhook
- 新的 Relic 整合 Webhook
- ServiceNow 整合 Webhook
- Slack 整合 Webhook

設定 Webhook 存取

步驟 1：導覽至 Webhook 組態

1. 登入 AWS 管理主控台並導覽至 AWS DevOps 代理程式主控台
2. 選取您的客服人員空間
3. 前往功能索引標籤
4. 在 Webhook 區段中，按一下設定

步驟 2：產生 Webhook 登入資料

對於整合特定的 Webhook：

當您完成第三方整合的組態時，Webhook 會自動產生。Webhook 端點 URL 和登入資料會在整合設定程序結束時提供。

對於一般 Webhook：

1. 按一下產生 Webhook

2. 系統會產生 HMAC 金鑰對
3. 安全地存放產生的金鑰和秘密 - 您將無法再次擷取它們
4. 複製提供的 Webhook 端點 URL

步驟 3：設定外部系統

使用 Webhook 端點 URL 和登入資料，將外部系統設定為將請求傳送至 AWS DevOps 代理程式。特定組態步驟取決於您的外部系統。

管理 Webhook 登入資料

移除登入資料 – 若要刪除 Webhook 登入資料，請前往 Webhook 組態區段，然後按一下移除。移除登入資料後，Webhook 端點將不再接受請求，直到您產生新的登入資料為止。

重新產生登入資料 – 若要產生新的登入資料，請先移除現有的登入資料，然後產生新的金鑰對或字符。

使用 Webhook

Webhook 請求格式

若要觸發調查，您的外部系統應將 HTTP POST 請求傳送至 Webhook 端點 URL。

對於第 1 版 (HMAC 身分驗證）：

標頭：

- `Content-Type: application/json`
- `x-amzn-event-signature: <HMAC signature>`
- `x-amzn-event-timestamp: <+%Y-%m-%dT%H:%M:%S.000Z>`

HMAC 簽章是透過使用 SHA-256 使用您的私密金鑰簽署請求內文來產生。

對於第 2 版（承載字符身分驗證）：

標頭：

- `Content-Type: application/json`
- `Authorization: Bearer <your-token>`

請求內文：

請求內文應包含事件的相關資訊：

```
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"
    }
  }
}
```

範例程式碼

第 1 版 (HMAC 身分驗證） - 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);
});
```

第 1 版 (HMAC 身分驗證） - 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"
```

第 2 版（承載字符身分驗證）- 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),
  });
}
```

第 2 版（承載字符身分驗證）- 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"
```

故障診斷 Webhook

如果您沒有收到 200

收到 200 和 Webhook 等訊息表示已通過身分驗證，且訊息已排入佇列，供系統驗證和處理。如果您沒有得到 200，但 4xx 很可能存在身分驗證或標頭錯誤。嘗試使用 curl 選項手動傳送，以協助偵錯身分驗證。

如果您收到 200，但調查未開始

可能的原因是承載格式錯誤。

1. 檢查時間戳記和事件 ID 是否已更新且是唯一的。重複的訊息會刪除重複訊息。
2. 檢查訊息是否為有效的 JSON
3. 檢查格式是否正確

如果您收到 200 且調查立即取消

您很可能已達到當月的限制。如果適用，請與您的 AWS 聯絡人討論，要求變更費率限制。

相關主題

- the section called “建立 代理程式空間”
- the section called “什麼是 DevOps Agent Web 應用程式？”
- the section called “DevOps Agent IAM 許可”

將 AWS DevOps 代理程式與 Amazon EventBridge 整合

您可以使用調查和緩解生命週期期間發生的事件，將 AWS DevOps 代理程式與您的事件驅動型應用程式整合。 AWS DevOps 代理程式會在調查或緩解狀態變更時將事件傳送至 Amazon EventBridge。然後，您可以建立 EventBridge 規則，根據這些事件採取動作。

例如，您可以建立執行下列動作的規則：

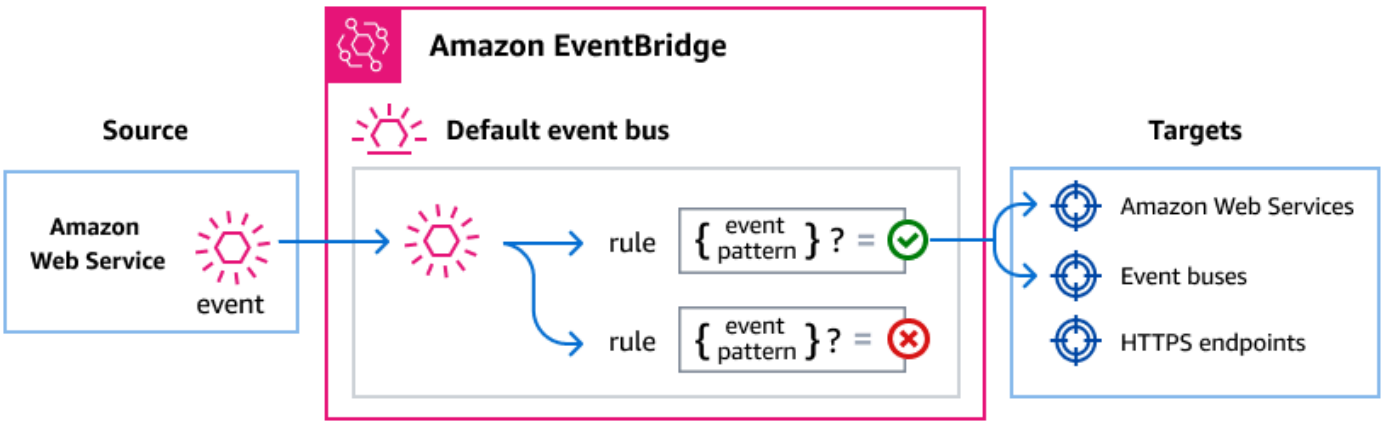
- 調查完成時，叫用 AWS Lambda 函數來處理調查結果。

- 調查失敗或逾時時傳送 Amazon SNS 通知。
- 在建立新的調查時更新票證系統。
- 在緩解動作完成時啟動 AWS Step Functions 工作流程。

EventBridge 如何路由 AWS DevOps 代理程式事件

AWS DevOps Agent 會將事件傳送至 EventBridge 預設事件匯流排。然後EventBridge 會根據您建立的規則評估事件。當事件符合規則的事件模式時，EventBridge 會將事件傳送至指定的目標。

下圖顯示 EventBridge 如何路由 AWS DevOps 代理程式事件。



1. 當調查或緩解生命週期狀態變更時，AWS DevOps 代理程式會將事件傳送至 EventBridge 預設事件匯流排。
2. EventBridge 會根據您建立的規則評估事件。
3. 如果事件符合規則的事件模式，EventBridge 會將事件傳送至規則中指定的目標。

AWS DevOps 代理程式事件

AWS DevOps 代理程式會將下列事件傳送至 EventBridge。所有事件都使用來源 `aws.aidevops`。

支援的調查事件

詳細資訊類型	Description
`Investigation Created`	已在客服人員空間中建立調查。

詳細資訊類型	Description
`Investigation Priority Updated`	調查的優先順序已變更。
`Investigation In Progress`	調查已開始主動分析。
`Investigation Completed`	調查結果已成功完成調查。
`Investigation Failed`	調查發生錯誤，無法完成。
`Investigation Timed Out`	調查超過允許的持續時間上限。
`Investigation Cancelled`	調查在完成之前已取消。
`Investigation Pending Triage`	在主動分析開始之前，調查正在等待分類。
`Investigation Linked`	調查已連結至相關事件或票證。

支援的緩解事件

詳細資訊類型	Description
`Mitigation In Progress`	已啟動緩解動作。
`Mitigation Completed`	已成功完成緩解動作。
`Mitigation Failed`	緩解動作發生錯誤，無法完成。
`Mitigation Timed Out`	緩解動作超過允許的持續時間上限。
`Mitigation Cancelled`	緩解動作已在完成之前取消。

如需詳細的欄位描述和範例事件，請參閱 the section called “AWS DevOps Agent 事件詳細資訊參考”。

建立符合 AWS DevOps 代理程式事件的事件模式

EventBridge 規則使用事件模式來選取事件並將其路由至目標。事件模式與其處理的事件結構相符。您可以建立事件模式，根據事件欄位篩選 AWS DevOps 代理程式事件。

下列範例顯示常見使用案例的事件模式。

比對 all AWS DevOps 代理程式事件

下列事件模式符合來自 AWS DevOps Agent 的所有事件。

```
{
  "source": ["aws.aidevops"]
}
```

僅比對調查事件

下列事件模式使用字首比對來僅選取調查生命週期事件。

```
{
  "source": ["aws.aidevops"],
  "detail-type": [{"prefix": "Investigation"}]
}
```

僅比對完成和失敗事件

下列事件模式符合已完成或失敗的調查和緩解措施的事件。

```
{
  "source": ["aws.aidevops"],
  "detail-type": [
    "Investigation Completed",
    "Investigation Failed",
    "Mitigation Completed",
    "Mitigation Failed"
  ]
}
```

比對特定客服人員空間的事件

下列事件模式符合來自特定客服人員空間的事件。

```
{
  "source": ["aws.aidevops"],
  "detail": {
    "metadata": {
      "agent_space_id": ["your-agent-space-id"]
```

```
    }
  }
}
```

如需有關事件模式的相關資訊，請參閱《Amazon EventBridge 使用者指南》中的 Amazon EventBridge 事件規則。

Amazon EventBridge 許可

AWS DevOps Agent 不需要額外的許可，即可將事件交付至 EventBridge。事件會自動傳送至預設事件匯流排。

根據您為 EventBridge 規則設定的目標，您可能需要新增特定許可。如需目標所需許可的詳細資訊，請參閱《Amazon EventBridge 使用者指南》中的使用 Amazon EventBridge 的資源型政策。EventBridge

額外 EventBridge 資源

如需 EventBridge 概念和組態的詳細資訊，請參閱《Amazon EventBridge 使用者指南》中的下列主題：

- EventBridge 事件匯流排
- EventBridge 事件
- EventBridge 事件模式
- EventBridge 規則
- EventBridge 目標

AWS DevOps Agent 事件詳細資訊參考

來自 AWS 服務的事件具有常見的中繼資料欄位，包括 `source`、`detail-type`、`region`、`account`和 `time`。這些事件也包含具有服務特定資料`detail`的欄位。For AWS DevOps 代理程式事件， 一律`source`為 `aws.aidevops`，而 會`detail-type`識別特定事件。

調查事件

下列`detail-type`值可識別調查事件：

- `Investigation Created`

- Investigation Priority Updated
- Investigation In Progress
- Investigation Completed
- Investigation Failed
- Investigation Timed Out
- Investigation Cancelled
- Investigation Pending Triage
- Investigation Linked

以下包含 source和 detail-type 欄位，因為其中包含 AWS DevOps 代理程式事件的特定值。如需所有事件中包含的其他中繼資料欄位的定義，請參閱《Amazon EventBridge 事件參考》中的事件結構。

以下是調查事件的 JSON 結構。

```
{
  . . .,
  "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 識別事件的類型。對於調查事件，這是先前列出的其中一個事件名稱。

source 識別產生事件的服務。對於 AWS DevOps 代理程式事件，此值為 aws.aidevops。

detail 包含事件特定資料的 JSON 物件。detail 物件包含下列欄位：

- version（字串）– 事件詳細資訊的結構描述版本。目前為 1.0.0。
- metadata.agent_space_id（字串）– 事件來源的代理程式空間的唯一識別符。
- metadata.task_id（字串）– 任務的唯一識別符。
- metadata.execution_id（字串）– 執行執行的唯一識別符。將執行指派給調查時出現。
- data.task_type（字串）– 任務的類型。值：INVESTIGATION。
- data.priority（字串）– 優先順序層級。值：CRITICAL、HIGH、MEDIUM、LOW、MINIMAL。
- data.status（字串）– 目前狀態。
  值：PENDING_START、IN_PROGRESS、COMPLETED、FAILED、TIMED_OUT、CANCELLED、PENDING_
- data.created_at（字串）– 建立任務時的 ISO 8601 時間戳記。
- data.updated_at（字串）– 任務上次更新時的 ISO 8601 時間戳記。
- data.summary_record_id（字串）– 包含調查結果的摘要記錄識別符。在針對已完成的調查產生摘要時包含。您可以使用此識別符，透過 AWS DevOps 代理程式 API 擷取摘要內容，以查詢記錄類型為 的日誌記錄investigation_summary_md。

範例：調查已完成事件

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

範例：調查失敗事件

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

緩解事件

下列detail-type值可識別緩解事件：

- Mitigation In Progress
- Mitigation Completed
- Mitigation Failed
- Mitigation Timed Out
- Mitigation Cancelled

以下包含 source和 detail-type 欄位，因為其中包含 AWS DevOps 代理程式事件的特定值。如需所有事件中包含的其他中繼資料欄位的定義，請參閱《Amazon EventBridge 事件參考》中的事件結構。

以下是緩解事件的 JSON 結構。

```
{
  . . .,
  "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 識別事件的類型。對於緩解事件，這是先前列出的其中一個事件名稱。

source 識別產生事件的服務。對於 AWS DevOps 代理程式事件，此值為 aws.aidevops。

detail 包含事件特定資料的 JSON 物件。detail 物件包含下列欄位：

- version（字串）– 事件詳細資訊的結構描述版本。目前為 1.0.0。
- metadata.agent_space_id（字串）– 事件來源的代理程式空間的唯一識別符。
- metadata.task_id（字串）– 任務的唯一識別符。
- metadata.execution_id（字串）– 執行執行的唯一識別符。當執行已指派給緩解措施時出現。
- data.task_type（字串）– 任務的類型。值：INVESTIGATION。
- data.priority（字串）– 優先順序層級。值：CRITICAL、HIGH、MEDIUM、LOW、MINIMAL。
- data.status（字串）– 目前狀態。
  值：IN_PROGRESS、COMPLETED、FAILED、TIMED_OUT、CANCELLED。
- data.created_at（字串）– 建立任務時的 ISO 8601 時間戳記。
- data.updated_at（字串）– 任務上次更新時的 ISO 8601 時間戳記。
- data.summary_record_id（字串）– 包含緩解問題清單之摘要記錄的識別符。在針對已完成的緩解產生摘要時包含。您可以使用此識別符查詢記錄類型為 的日誌記錄，透過 AWS DevOps 代理程式 API 擷取摘要內容mitigation_summary_md。

範例：緩解已完成事件

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

範例：緩解失敗事件

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

已取代的日誌和指標

您可以使用 vended Amazon CloudWatch 指標和日誌來監控代理程式空間和服務操作。本主題說明 AWS DevOps 代理程式自動發佈至您帳戶的 CloudWatch 指標，以及您可以設定以交付至您偏好目的地的付費日誌。

已取代的 CloudWatch 指標

AWS DevOps 代理程式會自動將指標發佈至您帳戶中的 Amazon CloudWatch。這些指標無需任何組態即可使用。您可以使用它們來監控用量、追蹤操作活動，以及建立警示。

服務連結角色

若要在您的帳戶中發佈此服務的 Amazon CloudWatch 指標， AWS DevOps 代理程式會自動為您建立服務連結角色 AWSServiceRoleForAIDevOps 服務連結角色。如果叫用 API 的 IAM 角色沒有適當的許可，資源建立將會失敗，並顯示 InvalidParameterException。

> **Important**
>
> 在 2026 年 3 月 13 日之前建立 AgentSpace 的客戶將需要手動建立 AWSServiceRoleForAIDevOps 服務連結角色，才能在其帳戶中發佈 AWS DevOps Agent 的 CloudWatch 指標。

手動建立服務連結角色 （適用於現有客戶）

執行以下任意一項：

- 在 IAM 主控台中，在 AWS DevOps Agent 服務下建立 AWSServiceRoleForAIDevOps 角色。
- 從 AWS CLI 執行下列命令：

```
aws iam create-service-linked-role --aws-service-name aidevops.amazonaws.com
```

命名空間

所有指標都會在 AWS/AIDevOps 命名空間下發佈。

維度

所有指標都包含下列維度。

維度	Description
`AgentSpaceUUID`	代理程式空間的唯一識別符。若要彙總帳戶中所有代理程式空間的指標，請使用 CloudWatch 數學表達式或省略維度篩選條件。

指標參考

指標名稱	Description	單位	發佈頻率	有用的統計資料
ConsumedChatRequests	客服人員空間耗用的聊天請求數量。若要取得您帳戶的總計數，請跨所有`AgentSpaceUUID` 維度使用 SUM 統計資料。	計數	每 5 分鐘	總和，平均
ConsumedInvestigationTime	在代理程式空間中執行調查所花費的時間。	秒鐘	每 5 分鐘	總和、平均值、最大值
ConsumedEvaluationTime	在代理程式空間中執行評估所花費的時間。	秒鐘	每 5 分鐘	總和、平均值、最大值
TopologyCompletionCount	拓撲處理完成次數。 AWS DevOps 代理程式會在拓撲處理完成時發出此指標，無論是從加	計數	事件驅動（每次完成時發出）	Sum、SampleCount

指標名稱	Description	單位	發佈頻率	有用的統計資料
	入期間的初始建立、手動更新或排定的每日重新整理。			

在 CloudWatch 主控台中檢視指標

1. 開啟 CloudWatch 主控台。
2. 在導覽窗格中，選擇 Metrics (指標)，然後選擇 All metrics (所有指標)。
3. 選擇 AWS/AIDevOps 命名空間。
4. 選擇 By AgentSpace 以檢視代理程式空間的指標。

> **Note**
>
> 您可以在這些指標上建立 CloudWatch 警示，以便在用量超過閾值時接收通知。例如，在 上建立警示`ConsumedChatRequests`以監控聊天請求耗用量。

先決條件

設定日誌交付之前，請確定您有下列項目：

- 可存取 AWS DevOps Agent 主控台的作用中 AWS 帳戶
- 具有 CloudWatch Logs 交付 APIsIAM 主體
- （選用）如果您打算使用 Amazon S3 儲存貯體或 Amazon Data Firehose 交付串流做為日誌目的地

付費日誌

AWS DevOps 代理程式支援自動提供的日誌，可讓您查看代理程式空間和服務註冊處理的事件。已取代的日誌使用 Amazon CloudWatch Logs 基礎設施，將日誌交付至您偏好的目的地。

若要使用付費日誌，您必須設定交付目的地。支援下列目的地：

- Amazon CloudWatch Logs – 您帳戶中的日誌群組

- Amazon S3 – 您帳戶中的 S3 儲存貯體
- Amazon Data Firehose – 帳戶中的 Firehose 交付串流

支援的日誌類型

支援單一日誌類型：APPLICATION_LOGS。此日誌類型涵蓋服務發出的所有操作事件。

日誌事件類型

下表摘要說明 AWS DevOps 代理程式記錄的事件。

事件	Description	日誌層級
收到的客服人員來電事件	代理程式是由整合來源觸發，並接收傳入事件（例如 PagerDuty 事件）。	INFO
代理程式傳入事件已捨棄	傳入事件在客服人員處理之前遭到捨棄。日誌包含原因（例如，資料格式不正確）。	待定
客服人員傳出通訊失敗	與第三方整合的傳出通訊失敗。日誌包含任務 ID 和目的地識別符（例如，身分驗證錯誤）。	待定
拓撲建立已排入佇列	拓撲建立任務已排入佇列進行處理。	INFO
拓撲建立已開始	拓撲建立任務開始處理。	INFO
拓撲建立完成	拓撲建立任務已完成處理。此事件適用於初始建立、更新和每日重新整理。	INFO
資源探索失敗	建立拓撲期間的資源探索遇到失敗。	ERROR
服務註冊失敗	服務註冊遇到無法復原的失敗	ERROR

事件	Description	日誌層級
Webhook 驗證失敗	當 Devops 代理程式收到的 Webhook 不符合預期的結構描述時	ERROR
關聯驗證狀態更新	當客服人員空間關聯（典型的主要/次要帳戶）時，驗證狀態會從有效變更為無效，反之亦然（例如，由格式不正確的角色所造成，且服務無法擔任）。	錯誤/資訊

許可

AWS DevOps 代理程式使用 CloudWatch 提供的日誌 (V2 許可） 來交付日誌。若要設定日誌交付，設定交付的 IAM 角色必須具有下列許可：

- `aidevops:AllowVendedLogDeliveryForResource` – 允許代理程式空間資源的日誌交付時需要。
- CloudWatch Logs 交付 APIs許可 (`logs:PutDeliverySource`、`logs:CreateDelivery`、`logs:PutDeliveryDestination`和相關操作）。
- 您所選交付目的地的特定許可。

如需每個目的地類型所需的完整 IAM 政策，請參閱《Amazon CloudWatch Logs 使用者指南》中的下列主題：

- 傳送至 CloudWatch Logs 的日誌
- 傳送至 Amazon S3 的日誌
- 傳送至 Firehose 的日誌

設定日誌交付（主控台）

AWS DevOps Agent 在 AWS 管理主控台中提供兩個位置來設定日誌交付：

- 服務註冊設定頁面 – 設定服務層級事件的日誌交付。這些日誌使用服務 ARN (`arn:aws:aidevops:<region>:<account-id>:service/<account-id>`) 做為資源。
- 客服人員空間頁面 – 為個別客服人員空間特定的事件設定日誌交付。這些日誌使用客服人員空間 ARN (`arn:aws:aidevops:<region>:<account-id>:agentspace/<agent-space-id>`) 做為資源。

設定服務註冊的日誌交付

1. 在 AWS 管理主控台中開啟 AWS DevOps 代理程式主控台。
2. 在導覽窗格中，選擇設定。
3. 在功能提供者 > 日誌索引標籤中，選擇設定。
4. 針對目的地類型，選擇下列其中一項：
5. CloudWatch Logs – 選取或建立日誌群組。
6. Amazon S3 – 輸入 S3 儲存貯體 ARN。
7. Amazon Data Firehose – 選取或建立 Firehose 交付串流。
8. 對於其他設定 – 選用，您可以指定下列選項：
   a. 針對欄位選擇，選取您要交付至目的地的日誌欄位名稱。您可以選取存取日誌欄位和即時存取日誌欄位的子集。
   b. (僅限 Amazon S3) 對於分割，指定分割日誌檔案資料的路徑。
   c. (僅限 Amazon S3) 對於 Hive 相容檔案格式，您可以選取核取方塊以使用 Hive 相容 S3 路徑。這有助於簡化將新資料載入 Hive 相容工具的過程。
   d. 針對輸出格式，請指定您偏好的格式。
   e. 對於欄位分隔符號，指定如何分隔日誌欄位。
9. 選擇儲存。
10. 確認交付狀態顯示作用中。

設定客服人員空間的日誌交付

1. 在 AWS 管理主控台中開啟 AWS DevOps 代理程式主控台。
2. 選擇您要設定的代理程式空間。
3. 在組態索引標籤中，選擇設定。
4. 針對目的地類型，選擇下列其中一項：
5. CloudWatch Logs – 選取或建立日誌群組。

6. Amazon S3 – 輸入 S3 儲存貯體 ARN。
7. Amazon Data Firehose – 選取或建立 Firehose 交付串流。
8. 對於其他設定 – *選用 *，您可以指定下列選項：
   a. 針對欄位選擇，選取您要交付至目的地的日誌欄位名稱。您可以選取存取日誌欄位和即時存取日誌欄位的子集。
   b. (僅限 Amazon S3) 對於分割，指定分割日誌檔案資料的路徑。
   c. (僅限 Amazon S3) 對於 Hive 相容檔案格式，您可以選取核取方塊以使用 Hive 相容 S3 路徑。這有助於簡化將新資料載入 Hive 相容工具的過程。
   d. 針對輸出格式，請指定您偏好的格式。
   e. 對於欄位分隔符號，指定如何分隔日誌欄位。
9. 選擇儲存。
10確認交付狀態顯示作用中。

設定日誌交付 (CloudWatch API)

您也可以使用 CloudWatch Logs API，以程式設計方式設定日誌交付。工作日誌交付包含三個元素：

- DeliverySource – 代表產生日誌的 AWS DevOps 代理程式空間資源。
- DeliveryDestination – 代表寫入日誌的目的地。
- 交付 – 將交付來源連接到交付目的地。

步驟 1：建立交付來源

使用 PutDeliverySource 操作來建立交付來源。傳遞 your AWS DevOps 代理程式空間資源的 ARN，並將 指定APPLICATION_LOGS為日誌類型。

下列範例會建立客服人員空間的交付來源：

```
{
    "name": "my-agent-space-delivery-source",
    "resourceArn": "arn:aws:aidevops:us-east-1:123456789012:agentspace/my-agent-space-
id",
    "logType": "APPLICATION_LOGS"
}
```

下列範例會建立服務的交付來源：

```
{
    "name": "my-service-delivery-source",
    "resourceArn": "arn:aws:aidevops:us-east-1:123456789012:service",
    "logType": "APPLICATION_LOGS"
}
```

步驟 2：建立交付目的地

使用 PutDeliveryDestination 操作來設定日誌存放的位置。您可以選擇 Amazon CloudWatch Logs、Amazon S3 或 Amazon Data Firehose。

下列範例會建立 CloudWatch Logs 目的地：

```
{
    "name": "my-cwl-destination",
    "deliveryDestinationConfiguration": {
        "destinationResourceArn": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/
aidevops/my-agent-space"
    },
    "outputFormat": "json"
}
```

下列範例會建立 Amazon S3 目的地：

```
{
    "name": "my-s3-destination",
    "deliveryDestinationConfiguration": {
        "destinationResourceArn": "arn:aws:s3:::my-aidevops-logs-bucket"
    },
    "outputFormat": "json"
}
```

下列範例會建立 Amazon Data Firehose 目的地：

```
{
    "name": "my-firehose-destination",
    "deliveryDestinationConfiguration": {
        "destinationResourceArn": "arn:aws:firehose:us-
east-1:123456789012:deliverystream/my-aidevops-log-stream"
    },
    "outputFormat": "json"
```

```
}
```

> **Note**
>
> 如果您跨帳戶交付日誌，則必須在目的地帳戶中使用 PutDeliveryDestinationPolicy 來授權交付。

如果您想要使用 CloudFormation，您可以使用下列項目：

- Delivery
- DeliveryDestination
- DeliverySource

ResourceArn 是 AgentSpaceArn，且 LogType 必須是 APPLICATION_LOGS 作為支援的日誌類型。

步驟 3：建立交付

使用 CreateDelivery 操作將交付來源連結至交付目的地。

```
{
    "deliverySourceName": "my-agent-space-delivery-source",
    "deliveryDestinationArn": "arn:aws:logs:us-east-1:123456789012:delivery-
destination:my-cwl-destination"
}
```

AWS CloudFormation

您也可以使用 AWS CloudFormation 搭配下列資源來設定日誌交付：

- AWS：：Logs：：DeliverySource
- AWS：：Logs：：DeliveryDestination
- AWS：：Logs：：Delivery

將 ResourceArn設定為 AWS DevOps 代理程式空間或服務 ARN，並將 LogType設定為 APPLICATION_LOGS。

日誌結構描述參考

AWS DevOps 代理程式在所有事件類型中使用共用日誌結構描述。不是每個日誌事件都會使用每個欄位。

下表說明日誌結構描述中的欄位。

欄位	Type	說明
event_timestamp	Long	Unix 事件發生時的時間戳記
resource_arn	String	產生事件之資源的 ARN
optional_account_id	String	AWS 與日誌相關聯的 帳戶 ID。
optional_level	String	日誌層級：INFO、WARN、ERROR
optional_agent_space_id	String	代理程式空間的識別符。
optional_association_id	String	日誌的關聯識別符。
optional_status	String	拓撲操作的狀態。
optional_webhook_id	String	Webhook 識別符。
optional_mcp_endpoint_url	String	MCP 伺服器端點 URL
optional_service_type	String	服務的類型：DYNATRACE 、DATADOG、GITHUB、SLACK、SERVICENOW 。
optional_service_endpoint_url	String	第三方整合的端點 URL。
optional_service_id	String	來源的識別符。
request_id	String	請求識別符以關聯 AWS CloudTrail 或支援票證。
optional_operation	String	已執行操作的名稱。

欄位	Type	說明
optional_task_type	String	客服人員待處理項目任務類型： `INVESTIGATION` 或 `EVALUATION`
optional_task_id	String	客服人員待處理任務 IDAgent 待處理任務識別符。
optional_reference	String	客服人員任務的參考（例如 Jira 票證）。
optional_error_type	String	錯誤類型
optional_error_message	String	操作失敗時的錯誤描述。
optional_details	字串 (JSON)	包含操作參數和結果的服務特定事件承載。

管理和停用日誌交付

您可以隨時從 AWS 管理主控台中的 AWS DevOps 代理程式主控台或使用 CloudWatch Logs API 來修改或移除日誌交付。

管理日誌交付（主控台）

1. 在 AWS 管理主控台中開啟 AWS DevOps 代理程式主控台。
2. 導覽至設定頁面（適用於服務層級日誌）或特定客服人員空間頁面（適用於客服人員空間層級日誌）。
3. 在組態索引標籤（適用於客服人員空間層級日誌）或功能提供者 > 日誌索引標籤（適用於服務層級日誌）中，選擇要修改的交付。
4. 視需要更新組態，然後選擇儲存。

> 注意：您無法變更現有交付的目的地類型。若要變更目的地類型，請刪除目前的交付並建立新的交付。

停用日誌交付（主控台）

1. 在 AWS 管理主控台中開啟 AWS DevOps 代理程式主控台。
2. 導覽至設定頁面（適用於服務層級日誌）或特定客服人員空間頁面（適用於客服人員空間層級日誌）。
3. 在組態索引標籤（適用於客服人員空間層級日誌）或功能提供者 > 日誌索引標籤（適用於服務層級日誌）中，選取要移除的交付。
4. 選擇刪除並確認。

停用日誌交付 (API)

若要使用 API 移除日誌交付，請依下列順序刪除資源：

1. 使用 DeleteDelivery 刪除交付。
2. 使用 DeleteDeliverySource 刪除交付來源。
3. （選用）如果不再需要交付目的地，請使用 DeleteDeliveryDestination 將其刪除。

> **Important**
>
> 您需負責在刪除產生日誌的客服人員空間資源後移除日誌交付資源（例如，刪除客服人員空間後）。如果您不移除這些資源，孤立的交付組態可能會保留。

定價

啟用付費日誌時 AWS DevOps 代理程式不會收取費用。不過，您可能需要支付交付、擷取、儲存或存取的費用，視您選取的日誌交付目的地而定。如需定價詳細資訊，請參閱 Amazon CloudWatch 定價中日誌索引標籤上的已修訂日誌。

如需目的地特定的定價，請參閱下列內容：

- Amazon CloudWatch Logs 定價
- Amazon S3 定價
- Amazon Data Firehose 定價

連線至私有託管工具

私有連線概觀

AWS DevOps 代理程式可以使用自訂模型內容通訊協定 (MCP) 工具和其他整合進行擴充，這些工具可讓代理程式存取內部系統，例如私有套件登錄檔、自我託管可觀測性平台、內部文件 APIs 和來源控制執行個體 （請參閱：設定適用於 AWS DevOps 代理程式的功能)。這些服務通常在 Amazon Virtual Private Cloud (Amazon VPC) 內部執行，但有限制或沒有公有網際網路存取，這表示預設情況下，AWS DevOps 代理程式無法與其連線。

適用於 AWS DevOps Agent 的私有連線可讓您安全地將 Agent Space 連線至 VPC 中執行的服務，而不會將其公開至公有網際網路。私有連線可與需要連線到私有端點的任何整合搭配使用，包括 MCP 伺服器、自我託管 Grafana 或 Splunk 執行個體，以及 GitHub Enterprise Server 和 GitLab 自我管理等來源控制系統。

> **Note**
>
> 如果您的私有託管工具從 VPC 內向 AWS DevOps 代理程式發出傳出請求，也可以使用 VPC 端點來保護此流量，使其保持在 AWS 網路內。例如，這可與透過 Webhook 事件觸發 DevOps 代理程式的工具搭配使用 （請參閱：the section called “透過 Webhook 叫用 DevOps 代理程式”)。如需詳細資訊，請參閱the section called “VPC 端點 (AWS PrivateLink)”。

私有連線的運作方式

私有連線會在 AWS DevOps Agent 與 VPC 中的目標資源之間建立安全的網路路徑。在幕後， AWS DevOps 代理程式會使用 Amazon VPC Lattice 來建立此安全的私有連線路徑。VPC Lattice 是一種應用程式聯網服務，可讓您跨 VPCs、帳戶和運算類型連線、保護和監控應用程式之間的通訊，而無需管理基礎網路基礎設施。

當您建立私有連線時，會發生下列情況：

- 您可以提供具有目標服務網路連線能力的 VPC、子網路和 （選用） 安全群組。
- AWS DevOps Agent 會建立服務管理的資源閘道，並在您指定的子網路中佈建其彈性網路介面 (ENIs)。
- 代理程式會使用 資源閘道，透過私有網路路徑將流量路由到目標服務的 IP 地址或 DNS 名稱。

資源閘道由 AWS DevOps Agent 完全管理，並顯示為帳戶中的唯讀資源（名為 `aidevops-{your-private-connection-name}`）。您不需要設定或維護它。在您的 VPC 中建立的唯一資源是您指定的子網路中的 ENIs。這些 ENIs做為私有流量的進入點，完全由 服務管理。他們不接受來自網際網路的傳入連線，而且您可以透過自己的安全群組保留對流量的完全控制。

安全

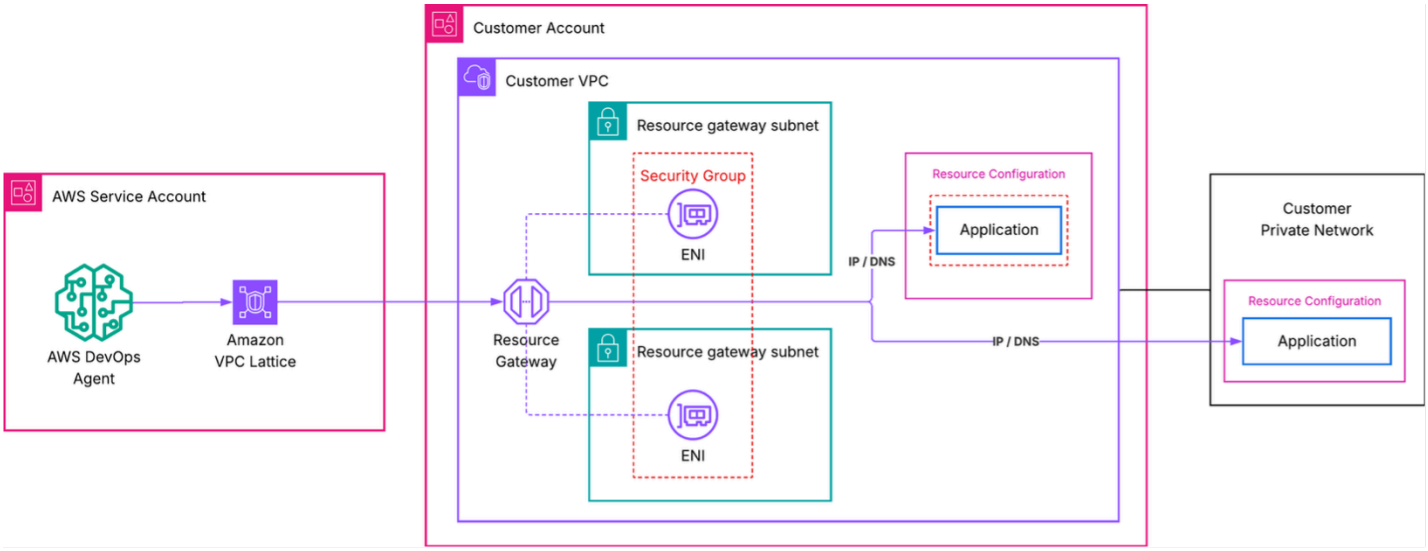
私有連線的設計具有多層安全性：

- 不公開網際網路 – AWS DevOps 代理程式與目標服務之間的所有流量都會保留在 AWS 網路上。您的服務永遠不需要公有 IP 地址或網際網路閘道。
- 服務控制的資源閘道 – 服務管理的資源閘道在您的帳戶中為唯讀。它只能由 AWS DevOps Agent 使用，其他服務或主體都無法透過它路由流量。您可以在 AWS CloudTrail 日誌中驗證這一點，這會記錄所有 VPC Lattice API 呼叫。
- 您的安全群組、規則 – 您可以透過您擁有和管理的安全群組來控制 ENIs的傳入和傳出流量。如果您未指定安全群組， AWS DevOps Agent 會建立預設安全群組，範圍限定於您定義的連接埠。
- 具有最低權限的服務連結角色 – AWS DevOps Agent 使用服務連結角色來僅建立必要的 VPC Lattice 和 Amazon EC2 資源。此角色的範圍是標記 的資源`AWSAIDevOpsManaged`，且無法存取您帳戶中的任何其他資源。

> **Note**
>
> 如果您的組織具有限制 VPC Lattice API 動作的服務控制政策 (SCPs)，則會透過服務連結角色建立服務受管資源閘道。確保您的 SCPs服務連結角色的必要動作。

Architecture

下圖顯示私有連線的網路路徑。



在此架構中：

- AWS DevOps 代理程式會對目標服務啟動請求。
- Amazon VPC Lattice 透過 VPC 中的服務受管資源閘道路由請求。如需使用您自己的 VPC Lattice 資源的進階設定，請參閱使用現有 VPC Lattice 資源的進階設定。
- VPC 中的 ENI 會收到流量，並將其轉送至目標服務的 IP 地址或 DNS 名稱。
- 您的安全群組會管理透過 ENIs 允許哪些流量。
- 從目標服務的角度來看，請求源自 VPC 內 ENIs的私有 IP 地址。

建立私有連線

您可以使用 AWS 管理主控台或 CLI AWS 建立私有連線。

> **Note**
>
> VPC Lattice 不支援下列可用區域：use1-az3、usw1-az2、apne1-az3、apne2-az2、euc1-az2、euw1-az4cac1-az3、、ilc1-az2。

先決條件

建立私有連線之前，請確認您有下列項目：

- 作用中的客服人員空間 – 您需要帳戶中現有的客服人員空間。如果您沒有帳戶，請參閱 開始使用 AWS DevOps 代理程式。

- 可私有連線的目標服務 – 您的 MCP 伺服器、可觀測性平台或其他服務必須可從部署資源閘道的 VPC 存取已知的私有 IP 地址或 DNS 名稱。該服務可以在相同的 VPC、對等 VPC 或內部部署中執行，只要它可從資源閘道子網路路由。服務必須在您在建立連線時指定的連接埠上，提供最低 TLS 版本為 1.2 的 HTTPS 流量。
- VPC 中的子網路 – 識別要建立 ENIs的 1-20 個子網路。建議您選取多個可用區域中的子網路，以獲得高可用性。這些子網路必須具有與目標服務的網路連線能力。VPC Lattice 可以使用每個可用區域一個子網路。
- （選用） 安全群組 – 如果您想要使用特定規則控制流量，請準備最多五個安全群組 IDs以連接到 ENIs。如果您省略安全群組， AWS DevOps Agent 會建立預設安全群組。

私有連線是帳戶層級的資源。建立私有連線後，您可以在需要連接相同主機的多個整合和客服人員空間中重複使用該連線。

使用主控台建立私有連線

1. 開啟 AWS DevOps Agent 主控台。
2. 在導覽窗格中，選擇能力提供者，然後選擇私有連線。
3. 選擇建立新的連線。
4. 針對名稱，輸入連線的描述性名稱，例如 `my-mcp-tool-connection`。
5. 針對 VPC，選取將部署資源閘道 ENIs VPC。
6. 針對子網路，選取一或多個子網路 （最多 20 個）。我們建議在至少兩個可用區域中選擇子網路。
7. 針對 IP 地址類型，選取目標服務的 IP 地址類型 (`IPv4`、 `IPv6`或 `DualStack`)。
8. （選用） 對於 IPv4 地址數量，如果您為 IP 地址類型選取 IPv4 或 Dualstack，則可以為資源閘道輸入每個 ENI 的 IPv4 地址數量。預設為每個 ENI 16 個 IPv4 地址。
9. （選用） 對於安全群組，選取現有的安全群組 （最多 5 個），以限制允許哪些流量到達您的目標服務。如果您未選取任何 ，則會建立預設安全群組。
10. （選用） 對於連接埠範圍，指定目標應用程式接聽的 TCP 連接埠 （例如 `443`或 `8080-8090`)。您最多可以指定 11 個連接埠範圍。
11. 針對主機地址，輸入目標服務的 IP 地址或 DNS 名稱 （例如 `mcp.internal.example.com`或 `10.0.1.50`)。該服務必須可從選取的 VPC 連線。如果您選擇 DNS 名稱，則必須可從選取的 VPC 解析。
12. （選用） 對於憑證公有金鑰，如果您指定的主機地址使用私有憑證授權單位發行的 TLS 憑證，請輸入憑證的 PEM 編碼公有金鑰。這可讓 AWS DevOps 代理程式信任與目標服務的 TLS 連線。

13選擇建立連線。

連線狀態會變更為建立進行中。此程序最多可能需要 10 分鐘。當狀態變更為作用中時，表示網路路徑已就緒。

如果狀態變更為建立失敗，請確認下列事項：

- 您指定的子網路具有可用的 IP 地址。
- 您的帳戶尚未達到 VPC Lattice 服務配額。
- 沒有任何限制性 IAM 政策阻止服務連結角色建立資源。

> **Note**
>
> 這些步驟也可以在功能提供者註冊`Create a new private connection`期間選取 來執行。如需詳細資訊，請參閱使用與功能提供者的私有連線。

使用 CLI AWS 建立私有連線

執行下列命令來建立私有連線。將預留位置值取代為您自己的值。

```
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"]
        }
    }'
```

回應包含連線名稱和 的狀態`CREATE_IN_PROGRESS`：

```
{
    "name": "my-mcp-tool-connection",
    "status": "CREATE_IN_PROGRESS",
    "resourceGatewayId": "rgw-0123456789abcdef0",
    "hostAddress": "mcp.internal.example.com",
    "vpcId": "vpc-0123456789abcdef0"
}
```

若要檢查連線狀態，請使用 `describe-private-connection`命令：

```
aws devops-agent describe-private-connection \
    --name my-mcp-tool-connection
```

當狀態為 時`ACTIVE`，您的私有連線即可使用。

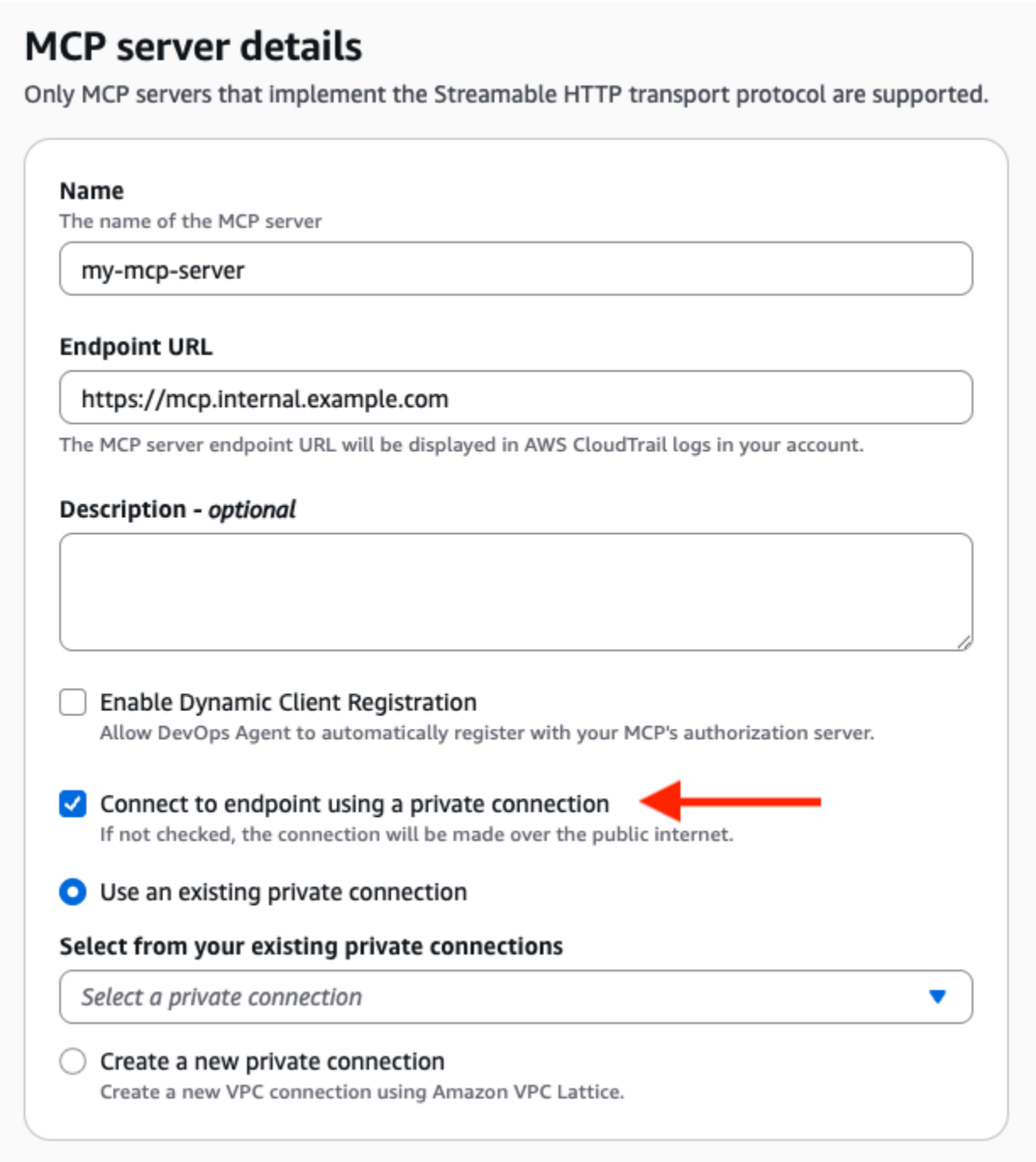
搭配功能提供者使用私有連線

若要使用私有連線，您可以在功能提供者註冊期間連結至該連線。可搭配私有連線使用的支援功能包括：`GitHub`、`MCP Server`、 `GitLab`和 `Grafana`。您可以使用 AWS 管理主控台或 CLI AWS 執行此步驟。

> **Note**
>
> 註冊功能提供者時， AWS DevOps 代理程式會驗證端點是否可連線並回應。在完成註冊之前，請確定您的目標服務正在執行並接受連線。

使用主控台與功能提供者使用私有連線

在 AWS DevOps 代理程式主控台中，可透過選取「使用私有連線連線至端點」選項，在註冊期間將私有連線連結至功能。



1. 開啟 AWS DevOps 代理程式主控台並導覽至您的代理程式空間。
2. 在功能提供者區段中，選擇註冊。
3. 選取註冊您要搭配私有連線使用的功能類型。
4. 在註冊詳細資訊檢視中，輸入您要使用私有連線 （例如，) 連線到的端點 URL`https://mcp.internal.example.com`。
5. 選取使用私有連線連線至端點。

6. 選取對應至您要連線之端點 URL 的現有私有連線，或選取建立新的私有連線以建立一個。
7. 完成能力提供者的註冊程序。

使用 AWS CLI 搭配功能提供者使用私有連線

您可以透過包含 `private-connection-name`引數，向私有連線註冊 功能。以下是使用`my-mcp-tool-connection`私有連線向 API 金鑰授權註冊 MCP 伺服器的範例。將預留位置值取代為您自己的值。

```
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
```

驗證私有連線

私有連線達到作用中狀態且功能提供者已使用後，請確認 AWS DevOps 代理程式可以到達您的目標服務：

1. 開啟 AWS DevOps 代理程式主控台並導覽至您的代理程式空間。
2. 啟動新的聊天工作階段。
3. 叫用使用私有連線所支援整合的命令。例如，如果您的 MCP 工具提供內部知識庫的存取權，請向客服人員詢問需要該知識庫的問題。
4. 確認代理程式傳回私有服務的結果。

如果連線失敗，請檢查下列項目：

- VPC Lattice 限制 - 確認您未達到任何資源閘道或其他 VPC Lattice 配額限制
- 安全群組規則 – 確認連接至 ENIs的安全群組允許您服務接聽之連接埠上的傳出流量。同時驗證服務的安全群組是否允許目標連接埠上的傳入流量。流量來自 VPC CIDR 範圍內的 VPC Lattice 資料平面 IPs。您可以使用參考安全群組（允許 ENI 安全群組做為來源）或允許從 VPC CIDR 傳入。
- 子網路連線 – 確認您選取的子網路可以將流量路由到您的服務。如果服務在不同子網路中執行，請確認路由表允許它們之間的流量。
- 服務可用性 – 確認您的服務正在預期連接埠上執行並接受連線。
- 不支援的可用區域 - 確認您的子網路位於支援的可用區域。執行`aws ec2 describe-subnets --subnet-ids <your-subnet-ids> --query 'Subnets[*].[SubnetId,AvailabilityZoneId]'`並檢查上述不支援的可用區域。

刪除私有連線

您可以使用 AWS 管理主控台或 CLI AWS 刪除未使用的私有連線。

使用主控台刪除私有連線

1. 開啟 AWS DevOps Agent 主控台。
2. 在導覽窗格中，選擇能力提供者，然後選擇私有連線。
3. 選取您要刪除之私有連線的動作功能表，然後選取移除。

私有連線的狀態將顯示為「移除連線」，而 AWS DevOps 代理程式會從 VPC 中移除受管資源閘道和 ENIs。刪除完成後，連線不會再出現在私有連線清單中。

使用 CLI AWS 刪除私有連線

```
aws devops-agent delete-private-connection \
    --name my-mcp-tool-connection
```

回應會傳回 `DELETE_IN_PROGRESS`. AWS DevOps 代理程式的狀態，將受管資源閘道和 ENIs 從 VPC 中移除。刪除完成後，連線不會再出現在私有連線清單中。

使用現有 VPC Lattice 資源的進階設定

如果您的組織已使用 Amazon VPC Lattice 並管理您自己的資源組態，您可以在自我管理模式中建立私有連線。您可以提供指向目標服務的現有資源組態的 Amazon Resource Name (ARN)，而不是讓 AWS DevOps 代理程式為您建立資源閘道。

當您：

- 想要完全控制資源閘道和資源組態生命週期。
- 需要在多個 AWS 帳戶或服務之間共用資源組態。
- 需要 VPC Lattice 存取日誌，以進行詳細的流量監控。
- 執行hub-and-spoke網路架構。

若要使用 CLI AWS 建立自我管理的私有連線：

```
aws devops-agent create-private-connection \
    --name my-advanced-connection \
    --mode '{
        "selfManaged": {
            "resourceConfigurationId": "arn:aws:vpc-lattice:us-
east-1:123456789012:resourceconfiguration/rcfg-0123456789abcdef0"
        }
    }'
```

如需設定 VPC Lattice 資源閘道和資源組態的詳細資訊，請參閱 Amazon VPC Lattice 使用者指南。

相關主題

- the section called “VPC 端點 (AWS PrivateLink)”
- the section called “連接 MCP 伺服器”
- 設定適用於 AWS DevOps 代理程式的功能
- AWS DevOps 代理程式安全性
- the section called “DevOps Agent IAM 許可”

AWS DevOps 代理程式安全性

本文件提供有關安全考量、資料保護、存取控制和 AWS DevOps Agent 合規功能的資訊。使用此資訊來了解 AWS DevOps Agent 如何設計來滿足您的安全和合規要求。

多層安全性

AWS DevOps Agent 會在多層實作安全性。即使將更廣泛的許可授予代理程式的 IAM 角色，代理程式也會強制執行自己的內部存取控制，以限制其動作的範圍。例如，如果客戶將完整的 Amazon S3 存取 IAM 政策新增至客服人員的 IAM 角色， AWS DevOps 代理程式將確保只讀取AWSLogs字首之後的日誌以進行故障診斷。

在設定 AWS DevOps 代理程式的 IAM 許可，以及在多層實作安全性時，我們建議遵循最低權限原則。深度防禦可確保單一錯誤設定不會危及環境的安全性。

客服人員空間

Agent Spaces 做為主要安全界限 in AWS DevOps Agent。每個客服人員空間：

- 使用自己的組態和許可獨立操作
- 定義客服人員可以存取 AWS 的帳戶和資源
- 建立與第三方平台的連線

Agent Spaces 會維持嚴格的隔離，以確保安全性，並防止不同環境或團隊的意外存取。

區域處理和資料流程

AWS DevOps 代理程式使用區域處理功能在全球營運。代理程式會從設定之代理程式空間內授予存取權的所有 AWS 帳戶 AWS 的區域擷取操作資料。此多區域跨帳戶資料收集可確保完整的事件分析，同時遵守推論處理的地理界限。

Amazon Bedrock 用量和跨區域推論

AWS DevOps 代理程式會自動選取您地理位置內的最佳區域，以處理您的推論請求。這可將可用的運算資源、模型可用性最大化，並提供最佳客戶體驗。您的資料只會儲存在建立 Agent Space 的區域，不過，輸入提示和輸出結果可能會在該區域之外進行處理，如下列清單所述。所有資料都會透過 Amazon 的安全網路進行加密傳輸。

AWS DevOps 代理程式會將您的推論請求安全地路由到發出請求的地理區域內的可用運算資源，如下所示：

- 源自歐盟的推論請求將在歐盟內處理。
- 來自美國的推論請求將在美國境內處理。
- 來自澳洲的推論請求將在澳洲處理。
- 來自日本的推論請求將在日本內處理。
- 如果推論請求源自未列出的區域，則預設會在美國境內進行處理。
- DevOps 代理程式和 Bedrock 不受服務控制政策 (SCPs) 或 Control Tower 中的客戶政策影響，這些政策會將客戶內容限制在特定區域
- Bedrock 可能會使用地理中原始區域以外的區域來執行無狀態推論，以最佳化效能和可用性

身分與存取管理

身分驗證方法

AWS DevOps Agent 提供兩種身分驗證方法來登入 AWS DevOps Agent Space Web 應用程式：

- AWS Identity Center 整合 – 主要身分驗證方法使用 OAuth 2.0 搭配僅使用 HTTP 的 Cookie 的工作階段型身分驗證。 AWS Identity Center 可以透過標準 OIDC 和 SAML 通訊協定與外部身分提供者聯合，包括 Okta、Ping Identity 和 Microsoft Entra ID 等提供者。此方法支援透過您的身分提供者進行多重驗證。 AWS Identity Center 預設為工作階段持續時間最長 12 小時，並可設定為所需的持續時間。
- IAM 身分驗證連結 – 一種替代方法可讓您使用衍生自現有 AWS 管理主控台工作階段的 JWT 型權杖，從 AWS 管理主控台直接存取 Web 應用程式。此選項有助於在實作完整的 Identity Center 整合之前評估 AWS DevOps 代理程式，以及在透過 Identity Center 型身分驗證無法存取 AWS DevOps 代理程式 Web 應用程式時取得管理存取權。工作階段限制為 10 分鐘。

IAM 角色

AWS DevOps Agent 使用 IAM 角色來定義存取許可：

- 主要帳戶角色 – 授予代理程式存取您建立代理程式空間之 AWS 帳戶中的資源，以及存取次要帳戶角色。
- 次要帳戶角色 – 授予代理程式對連接到代理程式空間之其他 AWS 帳戶中資源的存取權。

- Web 應用程式角色 – 授予使用者存取 Web 應用程式中 AWS 的 DevOps 代理程式調查資料和調查結果的權限。

這些角色應根據最低權限原則進行設定，只授予調查所需的唯讀許可。

資料保護

資料加密

AWS DevOps 代理程式會加密所有客戶資料：

- 靜態加密 – 所有資料都會使用 AWS受管金鑰加密。
- 傳輸中加密 – 所有擷取的日誌、指標、知識項目、票證中繼資料和其他資料都會在客服人員的私有網路和外部網路內傳輸時加密。

資料儲存和保留

資料會存放在建立代理程式空間的區域，而推論處理可能會在您的地理位置內進行，如上述 Amazon Bedrock 用量一節所述。

個人身分識別資訊 (PII)

在彙總調查、建議評估或聊天回應期間收集的資料時，AWS DevOps 代理程式不會篩選 PII 資訊。建議先修訂 PII 資料，再儲存在可觀測性日誌中。

客服人員日誌和稽核記錄

客服人員日誌

事件調查和預防功能都會維護詳細的日誌：

- 記錄採取的每個推理步驟和動作
- 建立客服人員決策程序的完整透明度
- 客服人員一旦記錄就無法修改，將攻擊降至最低，例如提示注入隱藏重要動作
- 包含調查頁面中的所有聊天訊息

AWS CloudTrail 整合

託管 AWS 帳戶中的 AWS CloudTrail 會自動擷取 All AWS DevOps 代理程式 API 呼叫。使用 CloudTrail 收集的資訊，您可以判斷：

- 向客服人員提出的請求
- 提出請求的 IP 位址
- 提出要求的人員
- 所提出的時間

提示注入保護

當攻擊者將惡意指示嵌入外部資料時，會發生提示注入攻擊。 例如網頁或文件 生成式 AI 系統稍後將處理的 。 AWS DevOps 代理程式會原生使用許多資料來源作為其正常操作的一部分， 包括日誌、 資源標籤、 和其他操作資料。 AWS DevOps 代理程式會透過以下保護措施，防範即時注入攻擊， 但請務必確保所有連線的資料來源和使用者對這些資料來源的存取都受到信任。如需詳細資訊，請參閱共同責任模型一節。

提示注入防護：

- 有限的寫入功能 – 代理程式可用的工具無法變更資源，但開立票證和支援案例除外。這可防止惡意指示修改您的基礎設施或應用程式。
- 帳戶界限強制執行 – AWS DevOps 代理程式只能在主要和連線次要 AWS 帳戶中指派給代理程式的角色允許的界限內操作。代理程式無法存取或修改超出其設定範圍的資源。
- AI 安全保護 – AWS DevOps Agent 使用具有 AI 安全層級 3 (ASL-3) 保護的模型。這些保護包括分類器，可在影響代理程式行為之前偵測和防止提示注入攻擊。
- 不可避免的稽核線索 – 代理程式日誌會記錄所採取的每個推理步驟和動作。一旦記錄日誌項目，代理程式就無法修改日誌項目，防止提示注入攻擊隱藏惡意動作。

雖然 AWS DevOps 代理程式提供多層保護，防止快速注入攻擊，但某些組態可能會增加風險：

- 自訂 MCP 伺服器工具 – bring-your-own MCP 功能可讓您將自訂工具引入代理程式，這可能會帶來額外的快速注入機會。自訂工具可能不具有與 native AWS DevOps 代理程式工具相同的安全控制，惡意指示可能會以非預期的方式利用這些工具。如需詳細資訊，請參閱共同責任模型一節。

- 授權使用者攻擊 – 獲授權在 AWS 帳戶界限或連線工具內操作的使用者嘗試對代理程式進行攻擊的機率較高。這些使用者可以修改代理程式使用的資料來源，例如日誌或資源標籤，以便更輕鬆地嵌入代理程式將處理的惡意指示。

若要降低這些風險：

1. 在 Agent Spaces 中部署自訂 MCP 伺服器之前，請仔細檢閱和測試這些伺服器。
   a. 確保只允許他們執行唯讀動作
   b. 確認 MCP 伺服器存取的外部工具使用者是受信任的實體，與 MCP 連接的 as AWS DevOps 代理程式依賴於這些工具使用者與 AWS DevOps 代理程式之間建立的隱含信任關係
2. 在授予使用者存取提供資料給代理程式的系統時，套用最低權限原則
3. 定期稽核哪些 MCP 伺服器連接到您的 代理程式空間
4. 由於從允許清單 URLs 擷取的任何內容都可能嘗試操縱代理程式的行為，因此只在您的允許清單中包含信任的來源。

整合安全性

AWS DevOps 代理程式支援多種整合類型，每個類型都有自己的安全模型：

- 原生雙向整合 – 內建整合，可將資料傳送至代理程式並從代理程式接收更新。這會使用廠商的身分驗證方法
- MCP 伺服器 – 利用 OAuth 2.0 身分驗證流程和 API 金鑰與外部系統安全通訊的遠端模型內容通訊協定伺服器。
- Webhook 觸發條件 – 調查來自遠端服務的觸發條件，例如票證或可觀測性系統。Webhooks 使用雜湊型訊息驗證碼 (HMAC) 來確保安全。
- 傳出通訊 – Slack 和票證系統等整合會收到客服人員的更新，但尚不支援雙向通訊。

註冊供應商

某些外部工具會在帳戶層級進行身分驗證，並在帳戶中的所有客服人員空間之間共用。當您註冊這些工具時，您可以在帳戶層級驗證一次，然後每個 Agent Space 都可以連線到該已註冊連線中的特定資源。

下列工具使用帳戶層級註冊：

- GitHub – 使用 OAuth 流程進行身分驗證。在帳戶層級註冊 GitHub 之後，每個 Agent Space 都可以連線到 GitHub 組織內的特定儲存庫。
- Dynatrace – 使用 OAuth 字符身分驗證。在帳戶層級註冊 Dynatrace 之後，每個客服人員空間都可以連線到特定的 Dynatrace 環境或監控組態。
- Slack – 使用 OAuth 字符身分驗證。在帳戶層級註冊 Slack 之後，每個客服人員空間都可以連線到特定的 Slack 頻道頻道。
- Datadog – 使用 MCP 搭配 OAuth 流程進行身分驗證。在帳戶層級註冊 Datadog 之後，每個 Agent Space 都可以連線到特定的 Datadog 監控資源。
- 新複本 – 使用 API 金鑰身分驗證。在帳戶層級註冊 New Relic 之後，每個代理程式空間都可以連線到特定的 New Relic 監控組態。
- Splunk – 使用承載字符身分驗證。在帳戶層級註冊 Splunk 之後，每個客服人員空間都可以連線到特定的 Splunk 資料來源。
- GitLab – 使用存取權杖身分驗證。在帳戶層級註冊 GitLab 之後，每個 Agent Space 都可以連線到特定的 GitLab 儲存庫。
- ServiceNow – 使用 OAuth 用戶端金鑰/金鑰身分驗證。在帳戶層級註冊 ServiceNow 之後，每個客服人員空間都可以連線到特定的 ServiceNow 執行個體或票證佇列。
- 一般可公開存取的遠端 MCP 伺服器 – 使用 OAuth 流程進行身分驗證。在帳戶層級註冊遠端 MCP 伺服器之後，每個 Agent Space 都可以連線到該伺服器公開的特定資源。

網路連線

AWS DevOps Agent 會連線至您的第三方系統和遠端 MCP 伺服器，以執行調查和其他操作。

從 AWS DevOps 代理程式到系統的傳入流量

AWS DevOps 代理程式會啟動第三方系統和遠端 MCP 伺服器的傳出連線，這些伺服器會以傳入流量的形式抵達您的基礎設施。保護此流量的方式取決於您的工具託管方式：

- 私有託管工具 – 如果您的工具可從 AWS VPC 內存取，您可以使用 AWS DevOps 代理程式私有連線，讓流量與 AWS 網路和公有網際網路隔離。如需詳細資訊，請參閱the section called “連線至私有託管工具”。
- 公有託管工具 – 如果您的工具可透過公有網際網路連線，並使用 IP 允許清單或防火牆規則，您必須允許來自下列 AWS DevOps 代理程式來源 IP 地址的傳入流量：
  - 亞太地區 (雪梨) (ap-southeast-2)

    - 13.237.95.197
    - 13.238.84.102
- 亞太區域 (東京) (ap-northeast-1)
    - 13.192.12.233
    - 35.74.181.230
    - 57.183.50.158
- 歐洲 (法蘭克福) (eu-central-1)
    - 18.158.110.140
    - 52.57.96.160
    - 52.59.55.56
- 歐洲 (愛爾蘭) (eu-west-1)
    - 34.251.85.24
    - 52.30.157.157
    - 52.51.192.222
- 美國東部 (維吉尼亞北部) (us-east-1)
    - 34.228.181.128
    - 44.219.176.187
    - 54.226.244.221
- 美國西部 (奧勒岡) (us-west-2)
    - 34.212.16.133
    - 52.89.67.212
    - 54.187.135.61

從 VPC 到 AWS DevOps 代理程式的傳出流量

對於從 AWS VPC 到 AWS DevOps 代理程式的傳出流量（例如，使用 the section called “透過 Webhook 叫用 DevOps 代理程式”），您可以使用 VPC 端點將此網路流量與 AWS 網路隔離。如需詳細資訊，請參閱the section called “VPC 端點 (AWS PrivateLink)”。

共同責任模型

AWS 責任

AWS 負責：

- 維護代理程式所擷取資料的安全性
- 保護可供 代理程式使用的原生工具
- 保護執行 AWS DevOps Agent 的基礎設施

客户責任

客戶負責：

- 管理使用者對客服人員空間的存取
- 限制提供輸入給代理程式之外部系統的受信任使用者存取，例如產生日誌、CloudTrail 事件、票證等的服務和資源 – 可用於嘗試惡意提示注入。
- 確保所有連線的資料來源都具有不可能被用來嘗試快速注入攻擊的信任資料
- 確保bring-your-own MCP 伺服器整合可安全運作
- 確保指派給客服人員的 IAM 角色具有適當的範圍
- 在可觀測性日誌和其他代理程式資料來源中存放之前，先修訂 PII 資料
- 遵循建議的做法，將唯讀許可授予連線的資料來源，包括bring-your-own MCP 伺服器

資料用量

AWS 不會使用來自整合資料來源的客服人員資料、聊天訊息或資料來訓練模型或改善產品。 AWS DevOps Agent Space 使用客戶產品內意見回饋來改善客服人員的回應和調查，但不 AWS 用它來改善服務本身。

合規

在預覽版中， AWS DevOps 代理程式不符合標準，包括 SOC 2、PCI-DSS、ISO 27001 或 FedRAMP。 AWS 稍後將宣佈提供哪些合規認證。

DevOps Agent IAM 許可

AWS DevOps Agent 使用服務特定的 AWS Identity and Access Management (IAM) 動作來控制對其功能和功能的存取。這些動作決定使用者可以在 AWS DevOps Agent 主控台和 Operator Web 應用程式中執行的動作。這與代理程式本身用來調查 資源 AWS 的服務 API 許可不同。

如需限制客服人員存取的詳細資訊，請參閱在 AWS 帳戶中限制客服人員存取。

客服人員空間管理動作

這些動作控制對 Agent Space 組態和管理的存取：

- aidevops：GetAgentSpace – 允許使用者檢視客服人員空間的詳細資訊，包括其組態、狀態和相關聯的帳戶。使用者需要此許可才能在 AWS 管理主控台中存取 代理程式空間。
- aidevops：GetAssociation – 允許使用者檢視特定帳戶關聯的詳細資訊，包括 IAM 角色組態和連線狀態。
- aidevops：ListAssociations – 允許使用者列出為客服人員空間設定的所有 AWS 帳戶關聯，包括主要和次要帳戶。

調查和執行動作

這些動作控制對事件調查功能的存取：

- aidevops：ListExecutions – 允許使用者檢視執行中繼資料，包括 ID、狀態等，以進行與任務相關聯的調查、緩解、評估和聊天對話。
- aidevops：ListJournalRecords – 允許使用者存取詳細日誌，其中顯示客服人員在調查、緩解、評估和聊天對話期間所諮詢的推理步驟、採取的動作和資料來源。這有助於了解客服人員如何得出結論。

聊天管理動作

Chat 需要下列 IAM 許可才能運作：

- aidevops：ListChats – 允許使用者列出和存取聊天對話歷史記錄。
- aidevops：CreateChat – 允許使用者建立新的聊天對話。
- aidevops：SendMessage – 允許使用者提交查詢並接收串流回應。

拓撲和探索動作

這些動作控制對應用程式資源映射功能的存取：

- aidevops：DiscoverTopology – 允許使用者觸發代理程式空間的拓撲探索和映射。此動作會啟動掃描 AWS 帳戶和建置應用程式資源拓撲的程序。

預防和建議動作

這些動作控制對預防功能的存取：

- aidevops：ListGoals – 允許使用者根據最近的事件模式，檢視客服人員正在努力的預防目標。
- aidevops：ListRecommendations – 允許使用者檢視預防功能產生的所有建議，包括其優先順序和類別。
- aidevops：GetRecommendation – 允許使用者檢視特定建議的詳細資訊，包括其會阻止的事件和實作指引。

待處理項目任務管理動作

這些動作可控制將建議管理為待處理任務的能力：

- aidevops：CreateBacklogTask – 允許使用者建立事件調查或預防評估任務。
- aidevops：UpdateBacklogTask – 允許使用者核准緩解計劃或取消主動調查或評估。
- aidevops：GetBacklogTask – 允許使用者擷取特定任務的詳細資訊。
- aidevops：ListBacklogTasks – 允許使用者列出代理程式空間的任務，依任務類型、狀態、優先順序或建立時間篩選。

知識管理動作

這些動作可控制新增和管理代理程式在調查期間可以使用的自訂知識的能力：

- aidevops：CreateKnowledgeItem – 允許使用者新增自訂知識項目，例如技能、故障診斷指南或客服人員應參考的應用程式特定資訊。
- aidevops：ListKnowledgeItems – 允許使用者檢視為客服人員空間設定的所有知識項目。
- aidevops：GetKnowledgeItem – 允許使用者擷取特定知識項目的詳細資訊。
- aidevops：UpdateKnowledgeItem – 允許使用者修改現有知識項目，以保持資訊為最新狀態。

- aidevops：DeleteKnowledgeItem – 允許使用者移除不再相關的知識項目。

AWS 支援整合動作

這些動作會控制與 AWS 支援案例的整合：

- aidevops：InitiateChatForCase – 允許使用者直接從調查中使用 AWS Support 啟動聊天工作階段，並自動提供有關事件的內容。
- aidevops：EndChatForCase – 允許使用者結束作用中的 AWS 支援案例聊天工作階段。
- aidevops：DescribeSupportLevel – 允許使用者檢查帳戶的 AWS 支援計劃層級，以判斷可用的支援選項。

用量和監控動作

這些動作會控制對用量資訊的存取：

- aidevops：GetAccountUsage – 允許使用者檢視調查時數、預防評估時數和聊天請求的 AWS DevOps 代理程式每月配額，以及當月的用量。

常見的 IAM 政策範例

管理員政策

此政策授予 all AWS DevOps Agent 功能的完整存取權：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "aidevops:*",
      "Resource": "*"
    }
  ]
}
```

運算子政策

此政策授予沒有管理功能的調查和預防功能的存取權：

```
{
  "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": "*"
    }
  ]
}
```

唯讀政策

此政策授予僅檢視的調查和建議存取權：

```
{
  "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": "*"
    }
  ]
}
```

使用適用於 AWS DevOps 代理程式的服務連結角色

AWS DevOps 代理程式使用 AWS Identity and Access Management (IAM) 服務連結角色。服務連結角色是直接連結至 AWS DevOps 代理程式的唯一 IAM 角色類型。服務連結角色由 AWS DevOps Agent 預先定義，並包含該服務代表您呼叫其他 AWS 服務所需的所有許可。

服務連結角色許可

`AWSServiceRoleForAIDevOps` 服務連結角色信任 `aidevops.amazonaws.com` 服務委託人來擔任角色。

角色使用`AWSServiceRoleForAIDevOpsPolicy`具有下列許可的 受管政策：

- `cloudwatch:PutMetricData` – 將用量指標發佈至 `AWS/AIDevOps` CloudWatch 命名空間。受`cloudwatch:namespace`條件限制，只允許`AWS/AIDevOps`命名空間。
- `vpc-lattice:CreateResourceGateway` – 建立私有連線的 VPC Lattice 資源閘道。受 `aws:RequestTag/AWSAIDevOpsManaged`條件限制，因此服務只能建立帶有`AWSAIDevOpsManaged`標籤的資源閘道。

- vpc-lattice:TagResource – 標記 VPC Lattice 資源閘道。由 aws:RequestTag/AWSAIDevOpsManaged條件範圍。
- vpc-lattice:DeleteResourceGateway – 刪除 VPC Lattice 資源閘道。受 aws:ResourceTag/AWSAIDevOpsManaged條件限制，因此服務只能刪除其建立的資源閘道。
- vpc-lattice:GetResourceGateway – 擷取 VPC Lattice 資源閘道的相關資訊。受 aws:ResourceTag/AWSAIDevOpsManaged條件限制，因此服務只能讀取其建立的資源閘道。
- ec2:DescribeVpcs、ec2:DescribeSubnets、 ec2:DescribeSecurityGroups – 擷取設定資源閘道所需的 VPC 聯網資源相關資訊。這些唯讀動作適用於所有 VPC 資源，因為 EC2 API 不支援描述呼叫的資源層級許可。
- iam:CreateServiceLinkedRole – 建立資源閘道操作所需的 VPC Lattice 服務連結角色。此許可僅限於vpc-lattice.amazonaws.com服務主體，不能用於為任何其他服務建立服務連結角色。

建立 服務連結角色

您不需要手動建立 AWSServiceRoleForAIDevOps 服務連結角色。當您開始使用 AWS DevOps Agent 時，服務會為您建立服務連結角色。

若要允許服務代表您建立角色，您必須擁有 iam:CreateServiceLinkedRole許可。建議使用 iam:AWSServiceName條件來限定此許可aidevops.amazonaws.com，以便 遵循最低權限原則。如需更多資訊，請參閱服務連結角色許可權限。

編輯 服務連結角色

您不能編輯 AWSServiceRoleForAIDevOps 服務連結角色。建立角色之後，您無法變更角色的名稱，因為各種實體可能會依名稱參考角色。然而，您可使用 IAM 來編輯角色描述。如需詳細資訊，請參閱編輯服務連結角色。

刪除 服務連結角色

如果您不再需要使用 AWS DevOps Agent，我們建議您刪除AWSServiceRoleForAIDevOps服務連結角色。您必須先移除在客服人員空間中設定的任何私有連線，才能刪除角色。刪除服務連結角色不會自動移除先前由服務建立AWSAIDevOpsManaged、以 標記的 VPC Lattice 資源閘道。如果不再需要這些資源閘道，您應該手動將其刪除。如需詳細資訊，請參閱刪除服務連結角色。

AWS 適用於 AWS DevOps 代理程式的受管政策

AWS 透過提供由 建立和管理的獨立 IAM 政策，解決許多常見的使用案例 AWS。這些 AWS 受管政策會授予常見使用案例的必要許可，讓您不必調查需要哪些許可。如需詳細資訊，請參閱《_IAM 使用者指南》中的 AWS 受管政策。

下列 AWS 受管政策是 AWS DevOps Agent 特有的，您可以連接到您帳戶中的使用者。

AIDevOpsAgentReadOnlyAccess

透過 AWS 管理主控台提供 Amazon DevOps 代理程式的唯讀存取權

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "AIDevOpsAgentReadOnlyAccess",
   "Effect": "Allow",
   "Action": [
    "aidevops:Get*",
    "aidevops:List*",
    "aidevops:SearchServiceAccessibleResource"
   ],
   "Resource": "*"
  }
 ]
}
```

AIDevOpsAgentFullAccess

透過 AWS 管理主控台提供 Amazon DevOps 代理程式的完整存取權

```
{
 "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

提供對 代理程式空間使用 AWS DevOps 運算子 Web 應用程式的存取權。

```
{
 "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

提供 AWS DevOps 代理程式執行調查和分析客戶 AWS 資源所需的許可。

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

限制 AWS 帳戶中的客服人員存取

AWS DevOps Agent 使用 IAM 角色在事件調查和預防性評估期間探索和描述 AWS 資源。您可以透過設定連接到這些角色的 IAM 政策來控制代理程式的存取層級。應用程式拓撲不會顯示代理程式可存取的所有內容，IAM 政策是真正限制代理程式可存取 AWS 的服務 APIs和資源的唯一方法。

了解 AWS 適用於 DevOps 代理程式的 IAM 角色

AWS DevOps Agent 使用 IAM 角色來存取兩種帳戶中的資源：

- 主要帳戶角色 – 授予代理程式存取您建立代理程式空間之 AWS 帳戶中的資源。
- 次要帳戶角色 – 授予代理程式存取您連接至代理程式空間之其他 AWS 帳戶中的資源。

對於任一類型的帳戶，您可以限制代理程式可以存取 AWS 的服務、限制對這些服務中特定資源的存取，以及控制代理程式可以操作的區域。

選擇您的資源界限

限制資源存取時，您需要包含足夠的許可，讓代理程式成功調查應用程式事件。其中包含：

- 客服人員應監控和調查範圍內應用程式的所有資源
- 這些應用程式依賴的所有支援基礎設施

支援基礎設施可能包括：

- 網路元件 VPCs、子網路、負載平衡器、API 閘道）
- 資料存放區 （資料庫、快取、物件儲存）
- 運算資源 (EC2 執行個體、Lambda 函數、容器）
- 監控和記錄服務 (CloudWatch、CloudTrail)
- 了解許可所需的身分和存取管理資源

如果您限制存取太窄，代理程式可能無法識別源自於您定義界限外之支援基礎設施的根本原因。

限制服務存取

您可以修改連接至客服人員角色的 IAM 政策，以限制客服人員可存取 AWS 的服務。建立自訂政策時，請遵循下列最佳實務：

- 授予唯讀許可 – 代理程式需要在調查期間讀取資源組態、指標和日誌。避免授予允許代理程式修改或刪除資源的許可。
- 限制為必要服務 – 僅包含包含與您應用程式相關資源 AWS 的服務。例如，如果您的應用程式不使用 Amazon RDS，請勿在政策中包含 RDS 許可。
- 使用特定動作而非萬用字元 – 指定個別動作，例如 `cloudwatch:GetMetricData`或 ，而非授予`service:*`許可`ec2:DescribeInstances`。

限制為特定 服務的政策範例：

```
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": "*"
    }
  ]
}
```

限制資源存取

若要將代理程式限制在服務中的特定資源，請在 IAM 政策中使用資源層級許可。這可讓您僅將存取權授予符合特定模式的資源。

使用資源 ARN 模式：

```
{
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lambda:GetFunction",
        "lambda:GetFunctionConfiguration"
      ],
      "Resource": "arn:aws:lambda:*:*:function:production-*"
    }
  ]
}
```

此範例限制代理程式只能存取名稱開頭為 "production-" 的 Lambda 函數。

使用標籤型限制：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeInstanceStatus"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Environment": "production"
        }
      }
    }
  ]
}
```

此範例限制代理程式只能存取以 標記的 EC2 執行個體`Environment=production`。

限制區域存取

若要限制代理程式可存取 AWS 的區域，請在 IAM 政策中使用 `aws:RequestedRegion`條件索引鍵：

```
{
```

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

此範例限制代理程式只能存取 us-east-1 和 us-west-2 區域中的資源。

建立自訂 IAM 政策

當您建立客服人員空間或新增次要帳戶時，您可以選擇使用政策範本建立自訂 IAM 角色。這可讓您實作最低權限原則。

建立 代理程式空間時

從 AWS 管理主控台中的 DevOps Agent 主控台...

- 選擇使用政策文件建立新的 DevOps 代理程式角色，並依照指示操作

編輯客服人員空間時

從 AWS 管理主控台中的 DevOps Agent 主控台...

- 選取功能索引標籤
- 從雲端區段中選取您要編輯的次要帳戶，然後按一下編輯
- 選擇使用範本建立新的 DevOps 代理程式政策，並依照指示操作

自訂政策最佳實務

- 授予唯讀許可 – 避免允許資源修改或刪除的許可
- 盡可能使用資源層級許可 – 限制使用 ARN 模式或標籤存取特定資源
- 定期檢閱和稽核許可 – 定期檢閱代理程式的 IAM 政策，以確保其仍符合您的安全需求

設定 IAM Identity Center 身分驗證

IAM Identity Center 身分驗證提供集中式方法來管理使用者對 AWS DevOps Agent Space Web 應用程式的存取。本指南說明如何設定 IAM Identity Center 身分驗證和管理使用者。

先決條件

設定 IAM Identity Center 身分驗證之前，請確定您已：

- 在您的組織或帳戶中啟用 IAM Identity Center
- 管理員許可 in AWS DevOps 代理程式
- 已設定或準備建立的客服人員空間

身分驗證選項

AWS DevOps Agent 提供兩種存取 Agent Space Web 應用程式的身分驗證方法：

IAM Identity Center 身分驗證 – 建議用於生產環境。提供集中式使用者管理、與外部身分提供者整合，以及長達 12 小時的工作階段。

管理員存取 (IAM 身分驗證） – 在初始設定和組態期間為管理員提供快速存取。工作階段限制為 30 分鐘。

在客服人員空間建立期間設定 IAM Identity Center

建立客服人員空間時，您可以在存取索引標籤上設定 IAM Identity Center 身分驗證：

步驟 1：導覽至 Web 應用程式組態

1. 設定客服人員空間詳細資訊和 AWS 帳戶存取權後，請繼續前往存取索引標籤
2. 您將看到兩個部分：「連接 IAM Identity Center」和「管理存取」

步驟 2：設定 IAM Identity Center 整合

在將 【客服人員空間】 連線至 IAM Identity Center 區段中：

1. 驗證 IAM Identity Center 執行個體 – 主控台會顯示哪個 Identity Center 執行個體將管理 Web App 使用者存取權（例如 `ssoins-7223a9580931edbe`)。系統會自動預先填入您最接近的 IAM Identity Center 執行個體。
2. 選取 IAM Identity Center 應用程式角色名稱選項 – 選擇三個選項之一：

自動建立新的 DevOps Agent 角色（建議）：

- 系統會自動建立具有適當許可的新服務角色
- 這是最簡單的選項，適用於大多數使用案例

指派現有角色：

- 使用您已建立的現有 IAM 角色
- 系統會驗證角色具有必要的許可
- 如果您的組織具有預先建立的 AWS DevOps Agent 角色，請選擇此選項

使用政策範本建立新的 DevOps Agent 角色：

- 使用提供的政策詳細資訊，在 IAM 主控台中建立您自己的自訂角色
- 如果您需要自訂角色許可，請選擇此選項

按一下連線後，系統會自動：

- 建立或設定指定的 IAM 角色
- 為您的客服人員空間設定 IAM Identity Center 應用程式
- 在 IAM Identity Center 和 Agent Space Web 應用程式之間建立信任關係
- 設定 OAuth 2.0 身分驗證流程以安全存取使用者

替代方案：使用管理員存取

如果您想要立即存取 Agent Space Web 應用程式，而不設定 IAM Identity Center：

1. 在管理員存取區段中，記下提供管理員存取權的 IAM 角色 ARN （例如 `arn:aws:iam::440491339484:role/service-role/DevOpsAgentRole-WebappAdmin-15ppoc42`)
2. 按一下藍色管理員存取按鈕，以使用 IAM 身分驗證啟動 Agent Space Web 應用程式
3. 使用此方法的工作階段限制為 30 分鐘

> **Note**
>
> 管理員存取權適用於初始設定和組態。對於生產用途和持續操作，設定 IAM Identity Center 身分驗證。

新增使用者和群組

設定 IAM Identity Center 身分驗證後，您需要授予特定使用者和群組對 Agent Space Web 應用程式的存取權：

步驟 1：存取使用者管理

1. 在 AWS DevOps 代理程式主控台中，選取您的代理程式空間
2. 前往存取索引標籤
3. 在使用者存取下，按一下管理使用者和群組

步驟 2：新增使用者或群組

1. 選擇新增使用者或群組
2. 在 IAM Identity Center 目錄中搜尋使用者或群組
3. 選取您要新增的使用者或群組旁的核取方塊
4. 按一下新增以授予他們存取權

選取的使用者現在可以使用其 IAM Identity Center 憑證存取 Agent Space Web 應用程式。

使用外部身分提供者

如果您使用外部身分提供者 （例如 Okta、Microsoft Entra ID 或 Ping Identity) 搭配 IAM Identity Center：

- 使用者和群組會從外部身分提供者同步到 IAM Identity Center
- 當您將使用者和群組新增至 Agent Space Web 應用程式時，您會從同步目錄選取
- 使用者屬性和群組成員資格由外部身分提供者維護
- 同步後，您的身分提供者的變更會自動反映在 IAM Identity Center 中

使用者如何存取 Agent Space Web 應用程式

將使用者新增至 代理程式空間之後：

1. 與授權使用者共用 Agent Space Web 應用程式 URL
2. 當使用者導覽至 URL 時，他們會重新導向至 IAM Identity Center 登入頁面
3. 輸入登入資料（如果已設定完成 MFA) 之後，系統會將其重新導向回 Agent Space Web 應用程式
4. 根據預設，其工作階段的有效期為 8 小時（可由 Identity Center 管理員設定）

管理使用者存取

您可以隨時更新使用者存取權：

新增更多使用者或群組：

- 依照上述相同步驟新增其他使用者或群組

移除存取權：

1. 在使用者存取區段中，尋找要移除的使用者或群組
2. 按一下其名稱旁的移除按鈕
3. 確認移除

移除的使用者將立即失去存取權，但作用中的工作階段可能會繼續，直到過期為止。

工作階段管理

Agent Space Web 應用程式的 IAM Identity Center 工作階段具有下列特性：

- 預設工作階段持續時間 – 8 小時
- 工作階段安全性 – 用於增強保護的僅限 HTTP Cookie

- 多重要素驗證 – 在 IAM Identity Center 中設定時支援
- API 登入資料 – 短期 (15 分鐘） SigV4 登入資料會針對 API 呼叫發出並自動續約

若要設定工作階段持續時間：

1. 導覽至 IAM Identity Center 主控台
2. 前往設定 > 身分驗證
3. 在工作階段持續時間下，設定您偏好的持續時間 （從 1 小時到 12 小時）
4. 選擇 Save changes (儲存變更)

中斷連接 Identity Center

1. 在客服人員空間的主控台中，按一下右上角的動作，然後選取從 IAM Identity Center 中斷連線
2. 在確認對話方塊中確認

設定外部身分提供者 (IdP) 身分驗證

外部身分提供者 (IdP) 身分驗證可讓您的組織使用現有的 OIDC 相容身分提供者，例如 Okta 或 Microsoft Entra ID，來管理使用者對 AWS DevOps Agent Space Web 應用程式的存取。使用者直接透過 IdP 使用其公司登入資料登入，而不需要 AWS IAM Identity Center。

先決條件

在設定外部 IdP 身分驗證之前，請確定您已：

- OIDC 相容身分提供者 (Okta 或 Microsoft Entra ID)
- 身分提供者的管理員存取權
- 存取 AWS DevOps Agent 主控台的管理員許可
- 已設定或準備建立的客服人員空間

運作方式

當您設定外部 IdP 身分驗證時：

- 使用者導覽至 Agent Space Web 應用程式 URL

- 它們會重新導向至您的身分提供者的登入頁面
- 使用其公司登入資料進行驗證後，它們會重新導向回 Web 應用程式
- Web 應用程式會交換身分驗證字符，以取得範圍為客服人員空間的短期 AWS 憑證

工作階段的有效期最長為 8 小時。登入資料會使用 OIDC 重新整理權杖自動重新整理，而不需要使用者重新驗證。

設定外部 IdP 身分驗證

步驟 1：在您的身分提供者中註冊應用程式

選擇您的身分提供者，並遵循對應的設定指示。

選項 A：Okta

1. 在 Okta 管理員主控台中，導覽至應用程式 > 應用程式，然後選擇建立應用程式整合
2. 選取 OIDC - OpenID Connect 做為登入方法，然後選取 Web 應用程式做為應用程式類型。選擇下一步
3. 設定應用程式的描述性名稱（例如 `AWS DevOps Agent`）
4. 在授予類型下，確保已檢查下列項目：
   - 授權碼（預設）
   - 重新整理權杖 — 這是工作階段重新整理的必要項目。如果未啟用，使用者將無法維護工作階段。

> **Note**
>
> 根據預設，Okta 不會啟用重新整理權杖授予類型。您必須明確啟用它。

1. 將登入重新導向 URIs 保留為目前預設值 - 您將在設定客服人員空間後進行更新
2. 在指派下，指派應具有存取權的使用者或群組
3. 選擇儲存
4. 在應用程式的一般索引標籤上，記下下列值：
   - 用戶端 ID
   - 用戶端秘密 — 選擇複製以安全地儲存此值
5. 請注意您的 Okta 網域 — 這是您的發行者 URL（例如 `https://dev-12345678.okta.com`）。

> **Note**
>
> 在登入索引標籤上，確認發行者設定為 Okta URL （非動態）。這可確保穩定的發行者 URL。

> **Note**
>
> 請勿將群組宣告新增至授權伺服器的宣告索引標籤中的 ID 字符。 AWS DevOps 代理程式不會使用來自 IdP 的群組成員資格。

選項 B：Microsoft Entra ID

1. 在 Azure 入口網站中，導覽至 Microsoft Entra ID > 應用程式註冊 > 新註冊
2. 設定描述性名稱（例如 `AWS DevOps Agent`)
3. 在支援的 帳戶類型下，選取適合您組織的選項 （通常僅適用於此組織目錄中的帳戶)
4. 現在將重新導向 URI 保留空白。選擇註冊
5. 在應用程式概觀頁面上，記下下列值：
   - 應用程式 （用戶端） ID — 在設定客服人員空間時用作用戶端 ID
   - 目錄 （租戶） ID — 用來建構發行者 URL
6. 導覽至憑證和秘密 > 新用戶端秘密
   - 設定描述和過期期間
   - 選擇新增並立即複製秘密值 — 不會再次顯示
7. Entra ID 的發行者 URL 遵循此格式。`{tenant-id}` 將 取代為步驟 5 中的目錄 （租戶） ID：
   - `https://login.microsoftonline.com/{tenant-id}/v2.0`

> **Note**
>
> 請勿在權杖組態 中啟用群組選用宣告。 AWS DevOps 代理程式不會使用來自 IdP 的群組成員資格。

步驟 2：使用 IdP 身分驗證啟用 Operator App

1. 在 AWS DevOps 代理程式主控台中，選取您的代理程式空間

2. 前往存取索引標籤
3. 在使用者存取下，選擇外部身分提供者
4. 在組態表單中，設定下列項目：
   - 身分提供者 — 選取您的身分提供者 (Okta 或 Microsoft Entra ID)
   - 發行者 URL — 來自您的身分提供者的 OIDC 發行者 URL
   - 用戶端 ID — 您建立之 OIDC 應用程式的用戶端 ID
   - 用戶端秘密 — 來自 OIDC 應用程式的用戶端秘密
5. 在身分提供者應用程式角色名稱下，選擇三個選項之一：
   - 自動建立新的 DevOps Agent 角色 （建議） — 建立具有適當許可的新服務角色
   - 指派現有角色 — 使用您已建立的現有 IAM 角色
   - 使用政策範本建立新的 DevOps 代理程式角色 — 使用提供的詳細資訊在 IAM 主控台中建立您自己的角色
6. 檢閱表單底部顯示的回呼 URL 警告提醒。複製此 URL — 您需要將其新增至身分提供者允許的重新導向 URIs使用者才能登入。
7. 選擇連線

選擇連線後，主控台會顯示外部身分提供者組態，其中包含下列詳細資訊：

- 提供者 — 您選取的身分提供者
- 發行者 URL — 設定的 OIDC 發行者 URL
- 用戶端 ID — 設定的用戶端 ID
- IAM 角色 ARN — 用於使用者存取的 IAM 角色
- 回呼 URL — 在身分提供者中將此 URL 設定為允許的重新導向 URI
- 登入 URL — 使用此 URL 透過您的身分提供者存取 Web 應用程式

步驟 3：將回呼 URL 新增至您的身分提供者

Okta

1. 在 Okta 管理員主控台中，導覽至應用程式的一般索引標籤
2. 在登入下，選擇編輯
3. 新增回呼 URL 做為登入重新導向 URI：

- `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/callback`

4. （選用）設定起始登入 URI 以從 Okta 儀表板啟用 IdP 起始的登入：
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/login`
5. （建議）新增登出重新導向 URI，以在登出後將使用者重新導向回 Web 應用程式。如果沒有這樣做，使用者在登出時可能會看到錯誤頁面：
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome`
6. 選擇儲存

Microsoft Entra ID

1. 在 Azure 入口網站中，導覽至應用程式的身分驗證頁面
2. 在平台組態下，選擇新增平台 > Web
3. 輸入回呼 URL 做為重新導向 URI：
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/callback`
4. （選用）新增登出重新導向 URI，以在登出後將使用者重新導向回 Web 應用程式：
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome`
5. 選擇設定

步驟 4：驗證組態

1. 導覽至主控台中顯示的登入 URL：
   - `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/idp/login`
2. 您應該重新導向至身分提供者的登入頁面
3. 使用您的公司登入資料登入
4. 身分驗證成功後，系統會將您重新導向回 Agent Space Web 應用程式

更新 IdP 組態

您可以在不中斷連線的情況下輪換用戶端秘密：

1. 在 AWS DevOps 代理程式主控台中，選取您的代理程式空間
2. 前往存取索引標籤

3. 在外部身分提供者組態下，選擇輪換用戶端秘密
4. 輸入新的用戶端秘密
5. 選擇儲存

若要變更任何其他 IdP 組態欄位（例如發行者 URL、用戶端 ID 或身分提供者），您必須中斷現有 IdP 的連線並設定新的 IdP。

使用者如何存取 Agent Space Web 應用程式

設定外部 IdP 身分驗證之後：

- 與授權使用者共用 Agent Space Web 應用程式 URL
- 當使用者導覽至 URL 時，他們會重新導向至您的身分提供者的登入頁面
- 輸入登入資料（以及完成由 IdP 設定的 MFA) 之後，系統會將其重新導向回 Agent Space Web 應用程式
- 工作階段自動重新整理 — 如需詳細資訊，請參閱工作階段管理

工作階段管理

Agent Space Web 應用程式的外部 IdP 工作階段具有下列特性：

- 工作階段持續時間 — 瀏覽器工作階段最多持續 8 小時。這無法設定 in AWS DevOps 代理程式。如果您的 IdP 工作階段存留期超過 8 小時，使用者可能會在下一次造訪時自動重新驗證，而無需輸入登入資料。根據組織的安全需求，設定 IdP 的工作階段和字符生命週期。
- 登入資料重新整理 — 工作階段會使用 OIDC 重新整理權杖自動重新整理，而不需要使用者重新驗證
- 多重要素驗證 — 在您的身分提供者中設定時支援。IdP 會在登入期間處理 MFA — 在 AWS DevOps 代理程式中不需要額外的組態

登出行為

當使用者在 Web 應用程式中按一下登出時：

1. 所有工作階段 Cookie 都會立即清除
2. 使用者會重新導向至身分提供者的 OIDC 登出端點，以終止 SSO 工作階段
3. 如果已設定登出重新導向 URI，則會將使用者重新導向回 Web 應用程式歡迎頁面

撤銷使用者存取權

若要立即撤銷使用者的存取權，您可以直接在身分提供者的管理員入口網站中撤銷其工作階段：

- Okta — 在 Okta 管理員主控台中，導覽至目錄 > 人員，選取使用者，選擇更多動作 > 清除使用者工作階段
- Microsoft Entra ID — 在 Azure 入口網站中，導覽至使用者、選取使用者，然後選擇撤銷工作階段

安全考量

用戶端秘密儲存 — 如果您在建立客服人員空間時提供用戶端秘密，或在其他情況下提供服務擁有的金鑰，則您在設定期間提供的用戶端秘密會使用客戶受管 KMS 金鑰進行加密。它永遠不會在 API 回應中傳回，或在初始組態後顯示在主控台中。

用戶端秘密輪換 — Entra 用戶端秘密具有可設定的過期時間。設定提醒，使用 AWS DevOps 代理程式主控台中的輪換用戶端秘密選項，在秘密過期之前輪換秘密。如果秘密過期，使用者將無法登入，直到輪換為止。

權杖生命週期管理 — 身分提供者發出的權杖生命週期（存取權杖、重新整理權杖）由 IdP 的組態控制。我們建議您在 IdP 中設定適當的字符生命週期：

- Okta — 在安全性 > API > 授權伺服器 > 存取政策下設定字符生命週期
- Microsoft Entra ID — 使用權杖生命週期政策設定權杖生命週期

群組宣告 — 請勿在身分提供者的字符組態中啟用群組宣告。 AWS DevOps 代理程式目前不使用來自 IdP 的群組成員資格。

使用者識別符 — AWS DevOps 代理程式使用提供者特定的宣告來唯一識別使用者：

- Okta — 使用來自 ID 字符的`sub`宣告
- Microsoft Entra ID — 使用 ID 字符中的 `oid`（物件識別符）宣告

這些識別符不可變，並出現在 CloudTrail 日誌中以供稽核之用。

中斷連接外部 IdP

1. 在 AWS DevOps 代理程式主控台中，選取您的代理程式空間

2. 前往存取索引標籤
3. 在使用者存取下，選擇中斷連線
4. 檢閱確認對話方塊中列出的影響並確認

中斷連線將：

- 從客服人員空間移除 IdP 組態
- 防止使用者透過外部身分提供者登入
- 移除與 IdP 使用者帳戶相關聯的個別聊天和成品歷史記錄

作用中的使用者工作階段會繼續，直到過期或下一次登入資料重新整理失敗為止。

疑難排解

- 重新導向至 IdP 失敗 — 驗證發行者 URL 符合您 IdP 的 OIDC 探索端點。對於 Okta，請確保在登入索引標籤上將發行者設定為 Okta URL （非動態)。對於 Entra，請使用格式 `https://login.microsoftonline.com/{tenant-id}/v2.0`。
- 存取遭拒或政策錯誤 (Okta) — 確認使用者或其群組已指派給指派下的應用程式。檢查登入 > 登入政策規則。
- 登入後的 IdP 組態錯誤 — 您的身分提供者未傳回重新整理字符。確保已啟用`offline_access`範圍和重新整理字符授予類型：
  - Okta — 前往您應用程式的一般索引標籤，並在授予類型下啟用重新整理權杖核取方塊
  - Entra — 前往 API 許可，並確保`offline_access`在委派許可下列出
- 身分驗證成功，但 Web 應用程式顯示錯誤 — 驗證 IdP 中的重新導向 URI 完全符合 AWS DevOps 代理程式主控台中顯示的回呼 URL。
- 身分驗證失敗 — 如果已在 IdP 中啟用群組選用宣告，請將其停用。 AWS DevOps 代理程式不會使用群組宣告。
- IdP 身分驗證後登入失敗 — 對於 Entra，驗證`requestedAccessTokenVersion`未在`null`應用程式資訊清單中設定為 。對於 Okta，驗證發行者 URL 是否正確。
- 按一下登出 (Okta) 之後的錯誤頁面 — 如果您在登出後看到`post_logout_redirect_uri`錯誤，請在 Okta 應用程式的一般索引標籤中新增 `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome`做為登出重新導向 URI。

- 使用者在登出後會停留在身分提供者頁面 (Entra) — 若要在登出後將使用者重新導向回 Web 應用程式，請在 Entra 應用程式的身分驗證頁面中新增 `https://{agentSpaceId}.aidevops.global.app.aws/authorizer/welcome`作為重新導向 URI。

for AWS DevOps 代理程式的靜態加密

AWS DevOps 代理程式會加密所有靜態客戶資料。根據預設， AWS DevOps 代理程式會使用 AWS 擁有的金鑰自動加密您的資料，無需額外費用。您無法檢視、管理或稽核 AWS 擁有金鑰的使用。不過，您不需要採取任何動作來保護這些金鑰。您的資料會自動受到保護。

您可以選擇使用您在 AWS Key Management Service (AWS KMS) 中建立、擁有和管理的對稱客戶受管金鑰來加密資料。由於您可以完全控制此加密層，因此您可以執行下列任務：

- 建立和維護金鑰政策
- 啟用和停用金鑰政策
- 輪換金鑰密碼編譯資料
- 新增 標籤
- 建立金鑰別名
- 安排金鑰供刪除

如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的客戶受管金鑰。

> **Note**
>
> AWS DevOps 代理程式會使用 AWS 擁有的金鑰自動啟用靜態加密，以免費保護客戶資料。當您使用客戶受管金鑰時，需支付標準 AWS KMS 費用。如需定價的詳細資訊，請參閱 AWS Key Management Service 定價。

客戶自管金鑰

客戶受管金鑰是您建立、擁有和管理之 AWS 帳戶中的 KMS 金鑰。您可以完全控制這些 KMS 金鑰，包括建立和維護其金鑰政策。

當您設定客戶受管金鑰時， AWS DevOps 代理程式會使用它來保護敏感資源資料。 AWS DevOps 代理程式會使用信封加密搭配 AWS Encryption SDK 階層式 keyring。您的 KMS 金鑰會用來產生分支金鑰，進而保護您的資料。

您可以在建立下列資源時指定客戶受管金鑰：

- Agent Space — 加密從 DevOps Agent Web 應用程式建立的與調查、技能和聊天相關的 Agent Space 詳細資訊和內容。
- 服務 — 加密靜態的第三方服務憑證。

若要設定客戶受管金鑰 in AWS DevOps 代理程式，請遵循下列步驟。

步驟 1：建立客戶受管金鑰

您可以使用 AWS KMS 主控台或 AWS KMS API 來建立對稱客戶受管金鑰。金鑰必須符合下列要求：

屬性	需求
Key type	對稱
金鑰規格	SYMMETRIC_DEFAULT
金鑰用途	ENCRYPT_DECRYPT

> **Note**
>
> AWS DevOps 代理程式僅支援具有金鑰規格和ENCRYPT_DECRYPT金鑰用量的對稱加密 KMS SYMMETRIC_DEFAULT金鑰。目前不支援多區域金鑰和非對稱金鑰。

如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的建立對稱客戶受管金鑰。

步驟 2：設定金鑰政策

金鑰政策會控制客戶受管金鑰的存取權限。每個客戶受管金鑰都必須只有一個金鑰政策，其中包含決定誰可以使用金鑰及其使用方式的陳述式。

您的金鑰政策必須同時授予呼叫委託人（您的 IAM 身分）和 AWS DevOps 代理程式服務的許可。AWS DevOps 代理程式會使用兩組登入資料存取您的金鑰：

1. 您的呼叫者登入資料 — 用於所有同步操作，包括金鑰驗證、資源建立時的加密，以及傳回直接回應給呼叫者的任何 API 呼叫。

2. AWS DevOps Agent 服務主體 — 用於在背景執行的非同步操作，例如操作調查、事件分析、事件關聯和根本原因分析產生。

下表列出必要的 KMS 動作：

KMS 動作	Description
`kms:DescribeKey`	在資源建立時間驗證金鑰組態
`kms:GenerateDataKey`	產生用於信封加密的資料加密金鑰
`kms:Decrypt`	解密資料
`kms:Encrypt`	加密資料
`kms:ReEncrypt`	在相同或不同的金鑰下重新加密資料

AWS DevOps 代理程式會在組態時間使用乾執行操作驗證所有這些許可。如果缺少任何許可，請求會失敗並出現例外狀況。

金鑰政策範例如下。將預留位置值取代為您自己的值。

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

此政策包含下列陳述式：

- AllowKeyAdministration — 授予帳戶根對金鑰的完整管理存取權。將 取代111122223333為 AWS 您的帳戶 ID。
- AllowCallerAccessViaService — 授予您的 IAM 主體所有 synchronous AWS DevOps 代理程式操作所需的 KMS 許可。這包括資源建立時的金鑰驗證，以及針對傳回直接回應給發起人的任何 API 呼叫進行加密和解密操作。kms:ViaService 條件可確保您只能透過 AWS DevOps Agent 服務使用金鑰。111122223333 將 取代為您的 AWS 帳戶 ID，並將 us-east-1取代為您的 AWS 區域。
- AllowDevOpsAgentServiceAccessForAgentSpace / AllowDevOpsAgentServiceAccessForService — 授予aidevops.amazonaws.com服務委託人非同步操作所需的 KMS 許可。 AWS DevOps 代理程式在執行操作調查、分析事件、關聯服務之間的事件，以及產生根本原因分析等背景操作時，使用此服務委託人來加密和解密您的資料。如果沒有此存取權， AWS DevOps 代理程式就無法讀取代表您執行調查所需的加密資料。aws:SourceArn 條件會限制存取來自 your AWS DevOps 代理程式資源的請求，且kms:EncryptionContext條件可確保加密內容符合您的資源 ARNs。111122223333 將 取代為您的 AWS 帳戶 ID，並將 us-east-1取代為您的 AWS 區域。

如需金鑰政策的詳細資訊，請參閱 Key Management Service 開發人員指南中的 AWS KMS 中的金鑰政策。 AWS

步驟 3：在建立資源時指定金鑰

建立金鑰並設定金鑰政策之後，您可以在建立 AWS DevOps 代理程式資源時指定金鑰。

主控台

若要在主控台中建立客服人員空間時設定客戶受管金鑰：

1. 開啟 AWS DevOps Agent 主控台。
2. 選擇建立客服人員空間或註冊服務。
3. 輸入客服人員空間詳細資訊（名稱、描述和 IAM 角色）。
4. 展開進階組態區段。
5. 在加密金鑰類型下，選取客戶受管金鑰。
6. 從下拉式清單中選擇 KMS 金鑰，或輸入 KMS 金鑰 ARN。
7. 檢閱金鑰政策可擴展區段中顯示的金鑰政策。請確定您已將此政策連接至 KMS 金鑰。您可以使用複製按鈕來複製政策。
8. 完成剩餘的組態，然後選擇建立。

> **Note**
>
> 如果您在下拉式清單中看不到 KMS 金鑰，請確認金鑰符合步驟 1 中的要求，而且您具有 `kms:ListKeys` 和 `kms:DescribeKey`許可。

API

使用客戶受管金鑰建立客服人員空間

在建立代理程式空間時指定 `kmsKeyArn` 參數。值必須是完整的 KMS 金鑰 ARN。

```
{
  "name": "my-agent-space",
  "description": "An encrypted agent space",
  "kmsKeyArn": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

使用客戶受管金鑰註冊服務

註冊服務時指定 kmsKeyArn 參數。值必須是完整的 KMS 金鑰 ARN。所有服務類型都支援此參數，包括 Dynatrace、ServiceNow、PagerDuty、GitLab、GitHub 和 MCP Servers。

```
{
  "service": "dynatrace",
  "kmsKeyArn": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "serviceDetails": { ... }
}
```

> **Note**
>
> 您必須在資源建立時指定客戶受管金鑰。您無法新增或變更現有資源的客戶受管金鑰。

AWS DevOps 代理程式加密內容

加密內容是一組非秘密金鑰/值對，其中包含有關資料的其他內容資訊。 AWS KMS 使用加密內容作為額外的已驗證資料，以支援已驗證的加密。當您在加密資料的請求中包含加密內容時， AWS KMS 會將加密內容繫結至加密的資料。若要解密資料，您必須在請求中包含相同的加密內容。

AWS DevOps 代理程式會在所有密碼編譯操作上使用下列加密內容：

```
{
  "aws-crypto-ec:aws:aidevops:arn": "arn:aws:aidevops:{region}:{accountId}:
{resourceType}/{resourceId}"
}
```

加密內容值是正在加密之 AWS DevOps 代理程式資源的 ARN。您可以在金鑰政策條件和 in AWS CloudTrail 日誌中使用此加密內容，以稽核金鑰的使用方式。

金鑰管理

如果您停用或排程刪除 KMS 金鑰， AWS DevOps 代理程式無法解密您的資料。這會導致讀取加密資料的操作AccessDeniedException發生錯誤。

> **Important**
>
> 如果您選擇使用客戶受管金鑰，您必須負責管理金鑰及其許可。如果金鑰已停用或刪除，或 if AWS DevOps Agent 失去使用金鑰的許可，您會失去加密資料的存取權。

下表說明常見的失敗案例：

Action	影響
金鑰政策許可已撤銷	`AccessDeniedException` 加密和解密操作
KMS 金鑰已停用	`DisabledException` 加密和解密操作
KMS 金鑰已排定刪除	`KMSInvalidStateException` 加密和解密操作
KMS 金鑰已刪除	永久資料遺失 - 無法復原加密的資料

在停用或刪除金鑰之前：

1. 確認沒有 Active AWS DevOps 代理程式資源依賴 金鑰。
2. 考慮先停用金鑰，以在排程刪除之前測試影響。
3. AWS KMS 會在刪除金鑰之前強制執行最短等待期間，讓您在需要時有時間取消。

> 注意：： AWS DevOps 代理程式不會在新金鑰下自動重新加密資料。如果您需要輪換到新的客戶受管金鑰，您必須使用新金鑰建立新的資源。

監控加密金鑰

當您使用客戶受管金鑰搭配 AWS DevOps 代理程式時，您可以使用 AWS CloudTrail 來追蹤 AWS DevOps 代理程式傳送至 AWS KMS 的請求。

您可以透過以下方式篩選 CloudTrail 事件：

- 事件來源 — `kms.amazonaws.com`
- 加密內容金鑰 — `aws-crypto-ec:aws:aidevops:arn`

- 金鑰 ARN — 請求參數中的客戶受管金鑰 ARN

如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的使用 AWS CloudTrail 記錄 AWS KMS API 呼叫。

VPC 端點 (AWS PrivateLink)

您可以使用 AWS PrivateLink 在 VPC 和 AWS DevOps 代理程式之間建立私有連線。您可以像在 VPC 中一樣存取 AWS DevOps 代理程式，無需使用網際網路閘道、NAT 裝置、VPN 連接或 Direct Connect 連接。VPC 中的執行個體不需要公有 IP 地址即可存取 AWS DevOps 代理程式。

您可以透過建立由 AWS PrivateLink 提供支援的介面端點來建立此私有連線。我們會在您為介面端點啟用的每個子網中建立端點網路介面。這些是請求者管理的網路介面，可做為目的地為 for AWS DevOps 代理程式之流量的進入點。

如需詳細資訊，請參閱《 AWS PrivateLink 指南》中的透過 PrivateLink 存取 AWS 服務。 AWS PrivateLink

AWS DevOps Agent VPC 端點的考量事項

在您設定 AWS DevOps 代理程式的介面端點之前，請檢閱 AWS PrivateLink 指南中的考量事項。

AWS DevOps Agent 支援透過下列 VPC 端點進行 API 呼叫。

Category	端點尾碼
AWS DevOps 代理程式控制平面 API 動作	`aidevops`
AWS DevOps 代理程式執行期操作	`aidevops-dataplane`
AWS DevOps Agent Webhook 事件	`event-ai`

建立適用於 AWS DevOps Agent 的介面端點

您可以使用 Amazon VPC 主控台或 AWS 命令列界面 (AWS CLI) 建立介面端點 for AWS DevOps 代理程式。如需詳細資訊，請參閱 AWS PrivateLink 指南中的建立介面端點。

使用下列服務名稱建立適用於 AWS DevOps Agent 的介面端點：

- com.amazonaws.{region}.aidevops
- com.amazonaws.{region}.aidevops-dataplane
- com.amazonaws.{region}.event-ai

建立端點後，您可以選擇啟用私有 DNS 主機名稱。當您建立 VPC 端點時，在 VPC 主控台中選取 Enable Private DNS Name (啟用私有 DNS 名稱) 來啟用此設計。

如果您為介面端點啟用私有 DNS，您可以使用其預設的區域 DNS 名稱向 AWS DevOps 代理程式提出 API 請求。下列範例顯示預設區域 DNS 名稱的格式。

- aidevops.{region}.api.aws
- aidevops-dataplane.{region}.amazonaws.com
- event-ai.{region}.api.aws

為您的介面端點建立端點政策

端點政策為 IAM 資源，您可將其連接至介面端點。預設端點政策允許透過介面端點完整存取 AWS DevOps 代理程式。若要控制 VPC 中允許的 to AWS DevOps 代理程式存取，請將自訂端點政策連接至介面端點。

端點政策會指定以下資訊：

- 可執行動作的主體 (AWS 帳戶、IAM 使用者和 IAM 角色）。
- 可執行的動作。
- 可供執行動作的資源。

如需詳細資訊，請參閱《 AWS PrivateLink 指南》中的使用端點政策控制對 服務的存取。

配額

AWS DevOps 代理程式配額包括代理程式空間數、並行調查等。您可以為某些配額請求增加，但並非所有配額都可以增加。這些增加不會立即授予，因此增加可能需要幾個小時到幾天才會生效。除非另有說明，否則每個配額都是區域特定的。

下表說明 AWS DevOps Agent 的配額。

名稱	預設	可調整	Description
每個區域的每個帳戶的代理程式空間	10	是	您可以在每個 AWS 區域中為每個帳戶建立的客服人員空間數目上限。
每個客服人員空間的並行調查	3	是	可在單一代理程式空間中同時執行的事件解決調查數量上限。
每個代理程式空間的並行評估	1	否	在單一代理程式空間中可同時執行的事件預防評估數量上限。
每個客服人員空間的並行隨需調用	10	是	可在單一代理程式空間中同時執行的隨需 DevOps 呼叫數量上限。

請求提高配額

您可以使用下列其中一個選項來請求提高配額：

- 從 AWS 管理主控台 – 開啟 Service Quotas 主控台。在導覽窗格中，選擇 AWS services (AWS 服務)。選取 DevOps 代理程式，選取配額，然後依照指示請求增加配額。如需詳細資訊，請參閱「Service Quotas 使用者指南」中的請求提高配額。
- 從 AWS CLI – 使用 request-service-quota-increase AWS CLI 命令。如需詳細資訊，請參閱「Service Quotas 使用者指南」中的請求提高配額。

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。