aws

使用者指南

Amazon CloudWatch 日誌

Amazon CloudWatch 日誌: 使用者指南

Table of Contents

什麼是 Amazon CloudWatch 日誌？

您可以使用 Amazon CloudWatch 日誌從 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體、Route 53 和其他來源監控 AWS CloudTrail、存放和存取日誌檔。

CloudWatch Logs 可讓您將所使用之所有系統、應用程式和 AWS 服務的記錄集中在單一、可高度擴充的服務中。然後，您可以輕鬆地查看它們，搜索特定的錯誤代碼或模式，根據特定字段對其進行過濾，或將其安全存檔以供 future 分析。 CloudWatch 記錄可讓您查看所有記錄檔，無論其來源為何，都是依時間排序的單一且一致的事件流程。

CloudWatch 記錄檔也支援使用強大的查詢語言查詢記錄、稽核和遮罩記錄檔中的敏感資料，以及使用篩選器或內嵌記錄格式從記錄檔產生指標。

CloudWatch 記錄檔支援兩種記錄類別。CloudWatch 記錄標準記錄類別中的記錄群組支援所有 CloudWatch 記錄檔功能。Logs 不常存取CloudWatch 記錄類別中的記錄群組會產生較低的擷取費用，並支援標準類別功能的子集。如需詳細資訊，請參閱 日誌類。

功能

- 兩個記錄類別提供彈性 — CloudWatch 記錄提供兩個記錄類別，因此您可以針對不常存取的記錄提供符合成本效益的選項。對於需要即時監控或其他功能的記錄，您也可以使用功能完整的選項。如需詳細資訊，請參閱 日誌類。
- 查詢您的日誌資料 — 您可以使用 CloudWatch 日誌見解以互動方式搜尋和分析您的日誌資料。您可以執行查詢，協助您更有效率且更有效地回應作業問題。 CloudWatch Logs Insights 包含專門建置的查詢語言，其中包含一些簡單但功能強大的命令。我們會提供範例查詢、命令描述、查詢自動完成及日誌欄位探索，以協助您開始使用。包含數種 AWS 服務記錄類型的範例查詢。若要開始使用，請參閱使用日誌見解分析 CloudWatch 日誌資料。
- 使用 Live Tail 偵測和偵錯 – 您可以使用 Live Tail 在擷取時檢視新日誌事件的串流清單，快速進行事件疑難排解。可以近乎即時地檢視、篩選和反白顯示擷取的日誌，協助您快速偵測並解決問題。可以根據指定的詞彙篩選日誌，並反白顯示包含指定詞彙的日誌，以協助您快速找到查詢內容。如需詳細資訊，請參閱 使用 Live Tail 以近乎即時的方式檢視日誌。
- 監控來自 Amazon EC2 執行個體的日誌 — 您可以使用CloudWatch 日誌資料來監控應用程式和系統。例如， CloudWatch 記錄檔可以追蹤應用程式記錄檔中發生的錯誤數目，並在錯誤率超過您指定的閾值時傳送通知給您。 CloudWatch 日誌使用您的日誌數據進行監視；因此，不需要更改代碼。例如，您可以監視應用程式記錄中的特定常值術語 (例如 "NullReferenceException「)，或計算在記錄資料中特定位置的文字項出現次數 (例如 Apache 存取記錄中的「404」狀態碼)。找到您要搜尋的字

詞時，「 CloudWatch 記錄」會將資料報告至您指定的 CloudWatch 量度。日誌資料會在移轉和靜態時加密。若要開始使用，請參閱開始使用 CloudWatch 記錄。

- 監視 AWS CloudTrail 記錄的事件 — 您可以在中建立警示 CloudWatch 並接收擷取的特定 API 活動的通知， CloudTrail 並使用通知執行疑難排解。若要開始AWS CloudTrail 使用，請參閱使用指南中的〈將 CloudTrail 事件傳送至CloudWatch 記錄檔〉。
- 稽核並遮罩敏感資料 – 如果您的日誌中含有敏感資料，則可以利用資料保護政策協助您保護資料。這些政策可讓您稽核並遮罩敏感資料。如果您啟用資料保護功能，則系統會按預設遮罩與您選用之資料識別符相符的敏感資料。如需詳細資訊，請參閱 使用遮罩功能協助保護敏感日誌資料。
- 日誌保留 - 根據預設，日誌將無限期保留且永遠不會過期。您可以調整每個日誌群組的保留政策，維持無限期保留，或選擇保留期間為 1 天至 10 年。
- 封存記錄資料 — 您可以使用 CloudWatch 防護記錄將記錄資料儲存在高耐用性的儲存裝置中。Lo CloudWatch gs 代理程式可讓您輕鬆快速地將旋轉和非輪換的記錄資料從主機傳送到記錄服務中。然後，您可以在需要時存取原始日誌資料。
- 記錄路由 53 DNS 查詢 — 您可以使用 CloudWatch 記錄檔來記錄路由 53 接收之 DNS 查詢的相關資訊。如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的記錄 DNS 查詢。

相關 AWS 服務

下列服務與 CloudWatch 記錄檔搭配使用：

- AWS CloudTrail是一項 Web 服務，可讓您監視對您帳戶的 CloudWatch 記錄 API 進行的呼叫，包括 AWS Management Console、 AWS Command Line Interface (AWS CLI) 和其他服務所發出的呼叫。開啟 CloudTrail 記錄時，會 CloudTrail擷取帳戶中的 API 呼叫，並將日誌檔傳送到您指定的 Amazon S3 儲存貯體。每個日誌檔案可以包含一個或多個記錄，取決於必須執行多少動作來滿足請求。如需有關的詳細資訊 AWS CloudTrail，請參閱什麼是 AWS CloudTrail？ 在《AWS CloudTrail 使用者指南》中。如需 CloudWatch 寫入 CloudTrail 記錄檔之資料類型的範例，請參閱記錄 CloudWatch 日誌 API 和控制台操作 AWS CloudTrail。
- AWS Identity and Access Management (IAM) 是一種 Web 服務，可協助您安全地控制使用者對 AWS 資源的存取。使用 IAM 控制誰可以使用您的 AWS 資源 (身分驗證)，以及他們可以透過何種方式使用哪些資源 (授權)。如需詳細資訊，請參閱《IAM 使用者指南》中的什麼是 IAM？。
- Amazon Kinesis Data Streams 這個 Web 服務可讓您快速且持續擷取和彙總資料。使用的資料類型包括 IT 基礎架構日誌資料、應用程式日誌、社交媒體、市場資料摘要和 web 點擊流資料。因為資料擷取和處理的回應時間是即時的，所以處理通常是輕量的。如需詳細資訊，請參閱《Amazon Kinesis Data Streams 開發人員指南》中的什麼是 Amazon Kinesis Data Streams？。

- AWS Lambda 是可讓您建置應用程式來快速回應新訊息的 web 服務。上傳您的應用程式碼當作 Lambda 函數，Lambda 會在高可用性運算基礎設施上執行您的程式碼，並完全負責管理運算資源，包括伺服器和作業系統維護、容量佈建與自動擴展、程式碼與安全性修補程式部署，以及程式碼監控和記錄。您唯一需要做的就是以 Lambda 支援的其中一種語言提供您的程式碼。如需詳細資訊，請參閱什麼是 AWS Lambda？ 在AWS Lambda 開發人員指南中。

定價

註冊後 AWS，您可以使用免費方案免費開始使用 CloudWatch AWS Logs。

標準費率適用於其他服務使用 CloudWatch 日誌存放的日誌 (例如，Amazon VPC 流量日誌和 Lambda 日誌)。

如需有關定價的詳細資訊，請參閱 Amazon CloudWatch 定價。

有關如何分析 CloudWatch 日誌的成本和用量的詳細資訊 CloudWatch，以及如何降低成本的最佳實務，請參閱CloudWatch 計費和成本。

Amazon CloudWatch 日誌概念

以下說明了您瞭解和使用 CloudWatch 記錄檔的重要術語和概念。

日誌類

CloudWatch 記錄檔提供兩種類別的記錄群組。對於需要即時監控的記錄或您經常存取的記錄，Standard 記錄類別是功能完整的選項。不常存取記錄檔類別是較低成本的選項，存取頻率較低的記錄檔。它支援標準記錄類別功能的子集。

日誌事件

日誌事件是由應用程式或正被監控的資源來記錄的一些活動。 CloudWatch Logs 瞭解的記錄事件記錄包含兩個屬性：事件發生時間的時間戳記和原始事件訊息。事件訊息必須為 UTF-8 編碼。

日誌串流

日誌串流是共享相同來源的一系列日誌事件。更具體地說，日誌串流通常旨在表示來自正在監視的應用程式執行個體或被監控的資源。例如，日誌串流可能與特定主機上的 Apache 存取日誌相關聯。當您不再需要日誌流時，可以使用 aws logs delete-log-stream 命令將其刪除。

日誌群組

日誌群組定義了共享相同保留、監控和存取控制設定的日誌串流群組。每個日誌串流必須屬於一個日誌群組。例如，如果您為每個主機的 Apache 存取日誌提供單獨的日誌串流，您可以將這些日誌串流分到一個名為 `MyWebsite.com/Apache/access_log` 的日誌群組。

可以屬於一個日誌群組的日誌串流數量並沒有限制。

指標篩選條件

您可以使用指標篩選器從擷取的事件擷取量度觀測值，並將其轉換為指標中的資料點。CloudWatch 指標篩選條件指派給日誌群組，並且指派給日誌群組的所有篩選條件都會套用於其日誌串流。

保留設定

保留設定可用來指定記錄事件在CloudWatch 記錄檔中保留的時間長度。過期的日誌事件會自動刪除。就像指標篩選條件一樣，保留設定也會指派給日誌群組，並將指派給日誌群組的保留套用於其日誌串流。

Amazon CloudWatch Logs 帳單與成本

如需有關如何分析 CloudWatch Logs 和 CloudWatch 的成本和用量，以及降低成本的最佳實務的詳細資訊，請參閱 CloudWatch 帳單與成本。

如需定價的詳細資訊，請參閱 Amazon CloudWatch 定價。

當您註冊 AWS 時，您可以透過 AWS 免費方案免費開始使用 CloudWatch Logs。

其他使用 CloudWatch Logs 服務儲存的日誌 (例如，Amazon VPC 流程日誌和 Lambda 日誌) 採用標準費率。

日誌類

CloudWatch 記錄檔提供兩種類別的記錄群組：

- 對於需要即時監控或經常存取的記CloudWatch 錄檔，Logs Standard 記錄類別是功能完整的選項。
- Lo CloudWatch gs 不常存取記錄類別是新的記錄類別，可用來以符合成本效益的方式整合記錄。此記錄類別提供 CloudWatch 記錄功能子集，包括受管理擷取、儲存、跨帳戶記錄分析和加密，且每 GB 擷取價格較低。不常存取記錄類別非常適合在不常存取的記錄檔上進行臨機操作查詢和 after-the-fact 鑑識分析。

> **Note**
>
> 費用方面，標準和不常存取記錄類別僅在擷取成本方面有所不同。每個 CloudWatch 記錄類別的儲存費用和記錄見解費用都相同。

如需 CloudWatch 日誌定價的詳細資訊，請參閱 Amazon CloudWatch 定價。

> **Important**
>
> 建立記錄群組之後，就無法變更其記錄類別。

支援的功能

下表列出了每個日誌類別的功能。

	標準	不常存取	
完全受控的記錄擷取和儲存	✓	✓	
跨帳戶功能	✓	✓	
使用加密 AWS KMS	✓	✓	
CloudWatch 日誌見解查詢命令	✓	✓（大多數命令-見記錄	

	標準	不常存取	
		類別中支援的命令。）	
CloudWatch 日誌見解發現欄位	✓		
自然語言查詢協助	✓		
CloudWatch 記錄異常偵測	✓		
與之前的時間範圍比較	✓		
訂閱過濾器	✓		
匯出至 Amazon S3	✓		
GetLogEvents和 FilterLogEventsAPI 操作	✓	不支援。使用 CloudWatch Logs Insights 來檢視「不常存取」記錄檔類別中儲存在記錄群組中的記錄事件。	
度量篩選器	✓		
容器洞見記錄擷取	✓		
Lambda 洞察日誌擷取	✓		
具有遮罩功能的敏感資料	✓		
內嵌量度格式	✓		

開始使用 CloudWatch 記錄

若要將 Amazon EC2 執行個體和現場部署伺服器的日誌收集到 CloudWatch 日誌中，請使用統一的 CloudWatch 代理程式。它可以讓您使用一個代理程式收集日誌及進階指標。它提供跨作業系統的支援，包括執行 Windows Server 的伺服器。此代理程式也提供更佳的效能。

如果您使用統一的 CloudWatch 代理程式來收集 CloudWatch 指標，它會啟用其他系統指標的收集，以提供客體內的可見度。它也支援使用 `StatsD` 或 `collectd` 收集自訂指標。

如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南中的安裝 CloudWatch 代理程式。

舊版 CloudWatch Logs 代理程式 (僅支援從執行 Linux 的伺服器收集記錄檔) 已取代且不再受支援。如需從舊版 CloudWatch Logs 代理程式移轉至整合代理程式的相關資訊，請參閱使用精靈建立 CloudWatch 代理程式組態檔。

目錄

必要條件

要使用 Amazon CloudWatch 日誌，您需要一個 AWS 帳戶。您的 AWS 帳戶可讓您使用服務 (例如 Amazon EC2) 來產生可在 CloudWatch 主控台中檢視的基於 Web 的界面的日誌。此外，您可以安裝和配置 AWS Command Line Interface （ AWS CLI ）。

註冊一個 AWS 帳戶

如果您沒有 AWS 帳戶，請完成以下步驟來建立一個。

若要註冊成為 AWS 帳戶

1. 開啟 https://portal.aws.amazon.com/billing/signup。
2. 請遵循線上指示進行。

   部分註冊程序需接收來電，並在電話鍵盤輸入驗證碼。

當您註冊一個時 AWS 帳戶，將創建AWS 帳戶根使用者一個。根使用者有權存取該帳戶中的所有 AWS 服務 和資源。安全性最佳做法是將管理存取權指派給使用者，並僅使用 root 使用者來執行需要 root 使用者存取權的工作。

AWS 註冊過程完成後，會向您發送確認電子郵件。您可以隨時登錄 https://aws.amazon.com/ 並選擇我的帳戶，以檢視您目前的帳戶活動並管理帳戶。

建立具有管理權限的使用者

註冊後，請保護您的 AWS 帳戶 AWS 帳戶根使用者 AWS IAM Identity Center、啟用和建立系統管理使用者，這樣您就不會將 root 使用者用於日常工作。

保護您的 AWS 帳戶根使用者

1. 選擇 Root 使用者並輸入您的 AWS 帳戶 電子郵件地址，以帳戶擁有者身分登入。AWS Management Console在下一頁中，輸入您的密碼。

   如需使用根使用者登入的說明，請參閱 AWS 登入 使用者指南中的以根使用者身分登入。
2. 若要在您的根使用者帳戶上啟用多重要素驗證 (MFA)。

   如需指示，請參閱《IAM 使用者指南》中的為 AWS 帳戶 根使用者啟用虛擬 MFA 裝置 (主控台)。

建立具有管理權限的使用者

1. 啟用 IAM Identity Center。

   如需指示，請參閱 AWS IAM Identity Center 使用者指南中的啟用 AWS IAM Identity Center。
2. 在 IAM 身分中心中，將管理存取權授予使用者。

   若要取得有關使用 IAM Identity Center 目錄 做為身分識別來源的自學課程，請參閱《使用指南》IAM Identity Center 目錄中的「以預設值設定使用AWS IAM Identity Center 者存取」。

以具有管理權限的使用者身分登入

- 若要使用您的 IAM Identity Center 使用者簽署，請使用建立 IAM Identity Center 使用者時傳送至您電子郵件地址的簽署 URL。

如需使用 IAM 身分中心使用者登入的說明，請參閱使用AWS 登入 者指南中的登入 AWS 存取入口網站。

指派存取權給其他使用者

1. 在 IAM 身分中心中，建立遵循套用最低權限許可的最佳做法的權限集。

   如需指示，請參閱《AWS IAM Identity Center 使用指南》中的「建立權限集」。
2. 將使用者指派給群組，然後將單一登入存取權指派給群組。

   如需指示，請參閱《AWS IAM Identity Center 使用指南》中的「新增群組」。

設定命令列介面

您可以使用執 AWS CLI 行 CloudWatch 記錄作業。

若要取得有關如何安裝和規劃的資訊 AWS CLI，請參閱《使用指南》中的〈AWS Command Line Interface 使用指 AWS 令行介面進行設置〉。

使用統一 CloudWatch 代理程式開始使用 CloudWatch 記錄

如需使用統一 CloudWatch 代理程式開始使用 CloudWatch 日誌的詳細資訊，請參閱 Amazon 使用者指南中的使用 CloudWatch 代理程式從 Amazon EC2 執行個體和現場部署伺服器收集 CloudWatch 指標和日誌。您完成此區段中所列的步驟以安裝、設定和啟動代理程式。如果您不使用代理程式也收集 CloudWatch 測量結果，則可以忽略任何參考測量結果的段落。

如果您目前使用較舊的 CloudWatch Logs 代理程式，而且想要使用新的整合代理程式移轉至，建議您使用新代理程式套件中包含的精靈。此精靈可以讀取您目前的 CloudWatch Logs 代理程式組態檔，並設定 CloudWatch代理程式以收集相同的記錄檔。如需有關精靈的詳細資訊，請參閱 Amazon CloudWatch 使用者指南中的使用精靈建立 CloudWatch 代理程式組態檔案。

使用先前的 CloudWatch 代理程式開始使用 CloudWatch 記錄

> **Important**
>
> CloudWatch 包括一個統一的 CloudWatch 代理程式，可從 EC2 執行個體和現場部署伺服器收集日誌和指標。舊版僅限日誌的代理程式已作廢且不再予以支援。

如需從舊版僅記錄代理程式移轉至整合代理程式的相關資訊，請參閱使用精靈建立 CloudWatch 代理程式組態檔。
本節的其餘部分將說明對仍在使用該代理程式的客戶使用舊版 CloudWatch Logs 代理程式。

使用 CloudWatch 日誌代理程式，您可以從執行 Linux 或 Windows 伺服器的 Amazon EC2 執行個體發佈日誌資料，以及從中發佈記錄的事件 AWS CloudTrail。我們建議您改用 CloudWatch 統一的代理程式來發佈您的記錄資料。如需有關新代理程式的詳細資訊，請參閱 Amazon 使用者指南中的使用 CloudWatch 代理程式從 Amazon EC2 執行個體和現場部署伺服器收集 CloudWatch 指標和日誌。

目錄

CloudWatch 記錄檔代理程式

記 CloudWatch 錄檔代理程式需要 Python 2.7、3.0 或 3.3 版，以及下列任何一個版本的 Linux：

- Amazon Linux 2014.03.02 版或更新版本。不支援 Amazon Linux 2
- Ubuntu Server 12.04、14.04 或 16.04 版
- CentOS 版本 6、6.3、6.4、6.5 或 7.0
- Red Hat Enterprise Linux (RHEL) 版本 6.5 或 7.0
- Debian 8.0

快速入門：在執行中的 EC2 Linux 執行個體上安裝和設定 CloudWatch 日誌代理程式

Important

較舊的記錄代理程式已取代。 CloudWatch 包括一個統一的代理程式，可從 EC2 執行個體和現場部署伺服器收集日誌和指標。如需詳細資訊，請參閱 開始使用 CloudWatch 記錄。
如需從舊版 CloudWatch Logs 代理程式移轉至整合代理程式的相關資訊，請參閱使用精靈建立 CloudWatch 代理程式組態檔。
舊版 Logs 代理程式只支援 2.6 到 3.5 版的 Python。此外，較舊的 CloudWatch 記錄代理程式不支援執行個體中繼資料服務版本 2 (IMDSv2)。如果您的伺服器使用 IMDSv2，您必須使用較新的統一代理程式，而不是舊版 CloudWatch Logs 代理程式。
本節的其餘部分將說明對仍在使用該代理程式的客戶使用舊版 CloudWatch Logs 代理程式。

Tip

CloudWatch 包含新的統一代理程式，可從 EC2 執行個體和現場部署伺服器收集日誌和指標。如果您尚未使用較舊的 CloudWatch Logs 代理程式，建議您使用較新的整合 CloudWatch代理程式。如需詳細資訊，請參閱 開始使用 CloudWatch 記錄。
此外，舊版的代理程式不支援執行個體中繼資料服務第 2 版 (IMDSv2)。如果您的伺服器使用 IMDSv2，您必須使用較新的統一代理程式，而不是舊版 CloudWatch Logs 代理程式。
本節的其餘部分將說明舊版 CloudWatch Logs 代理程式的使用。

在執行中的 EC2 Linux 執行個體上設定較舊的 CloudWatch 記錄代理程式

您可以在現有 EC2 執行個體上使用 CloudWatch Logs 代理程式安裝程式來安裝和設定 CloudWatch 日誌代理程式。安裝完成後，日誌會自動從執行個體提供到您在安裝代理程式時所建立的日誌串流。代理程式可確認它已啟動並保持執行，直到您停用為止。

除了使用代理程式之外，您也可以使用 CloudWatch 記錄 SDK 或記錄 API 來發佈 CloudWatch 記錄檔資料。 AWS CLI最 AWS CLI 適合在指令行或透過指令碼發佈資料。 CloudWatch 日誌 SDK 最適合直接從應用程序發布日誌數據或構建自己的日誌發布應用程序。

步驟 1：針對記錄設定 IAM 角色或使用 CloudWatch 者

CloudWatch 日誌代理程式支援 IAM 角色和使用者。如果您的執行個體已有相關聯的 IAM 角色，請務必包含以下的 IAM 政策。如果您尚無指派給執行個體的 IAM 角色，則可以在後續步驟中使用 IAM 憑證，或將 IAM 角色指派給該執行個體。如需詳細資訊，請參閱將 IAM 角色連接到執行個體。

為記錄設定 IAM 角色或使用 CloudWatch 者

1. 在以下網址開啟 IAM 主控台：https://console.aws.amazon.com/iam/。
2. 在導覽窗格中，選擇角色。
3. 選取該角色名稱 (不選取該名稱旁的核取方塊) 以選擇角色。
4. 選擇 Attach Policies (連接政策)、Create Policy (建立政策)。

   新的瀏覽器標籤或視窗隨即開啟。
5. 選擇 JSON 標籤，並輸入下列 JSON 政策文件。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
    ],
      "Resource": [
        "*"
    ]
  }
 ]
}
```

6. 完成時，選擇 Review policy (檢閱政策)。Policy Validator (政策檢查工具) 會回報任何語法錯誤。
7. 在 Review Policy (檢閱政策) 頁面上，為您正在建立的政策輸入 Name (名稱) 與 Description (描述) (選用)。檢閱政策 Summary (摘要) 來查看您的政策所授予的許可。然後選擇 Create policy (建立政策) 來儲存您的工作。
8. 關閉瀏覽器標籤或視窗，並返回您角色的 Add permissions (新增許可) 頁面。選擇 Refresh (重新整理)，然後選擇新政策以連接到您的角色。

9. 選擇 Attach Policy (連接政策)。

步驟 2：在現有的 Amazon EC2 執行個體上安裝和設定 CloudWatch 日誌

根據您的 Amazon EC2 實例是運行亞馬遜 Linux，Ubuntu，CentOS 還是紅帽，安裝 CloudWatch 日誌代理程序的過程有所不同。使用適用於您執行個體上 Linux 版本的步驟。

在現有的 Amazon Linux 執行個體上安裝和設定 CloudWatch 日誌

從 Amazon Linux AMI 2014.09 開始， CloudWatch 日誌代理程式可透過安裝 awslog 套件作為 RPM 安裝使用。舊版的 Amazon Linux 可以使用 `sudo yum update -y` 命令更新其執行個體，以存取 awslogs 套件。將 awslogs 套件安裝為 RPM，而不是使用 Logs 安裝程式，您的執行個體會接收定期的套件更新和修補程式，而 AWS 不必手動重新安裝 CloudWatch Logs 代理程式。 CloudWatch

> **Warning**
>
> 如果您先前使用 Python 指令碼來安裝代理程式，請勿使用 RPM 安裝方法更新 CloudWatch 記錄代理程式。這麼做可能會造成設定問題，導致 Lo CloudWatch gs 代理程式無法將記錄檔傳送至 CloudWatch。

1. 連線至 Amazon Linux 執行個體。如需詳細資訊，請參閱 Amazon EC2 使用者指南中的 Connect 到您的執行個體。

   如需有關連線問題的詳細資訊，請參閱 Amazon EC2 使用者指南中的連線至執行個體疑難排解。
2. 更新 Amazon Linux 執行個體，以取得套件儲存庫中的最新變更。

   ```
   sudo yum update -y
   ```
3. 安裝 `awslogs` 套裝服務。這是在 Amazon Linux 執行個體上安裝 awslogs 的建議方法。

   ```
   sudo yum install -y awslogs
   ```
4. 編輯 `/etc/awslogs/awslogs.conf` 檔案來設定要追蹤的日誌。如需編輯此檔案的詳細資訊，請參閱 CloudWatch 記錄用戶端參考。
5. 根據預設，`/etc/awslogs/awscli.conf` 會指向 us-east-1 區域。若要將您的日誌推送至不同的區域，請編輯 `awscli.conf` 檔案並指定該區域。
6. 啟動 `awslogs` 服務。

```
sudo service awslogs start
```

如果您執行的是 Amazon Linux 2，請使用下列命令來啟動 awslogs 服務。

```
sudo systemctl start awslogsd
```

7. (選用) 針對啟動服務時記錄的錯誤，核取 /var/log/awslogs.log 檔案。
8. (選用) 執行以下命令以在每次系統啟動時啟動 awslogs 服務。

```
sudo chkconfig awslogs on
```

如果您執行的是 Amazon Linux 2，請使用在每個系統啟動時的下列命令來啟動服務。

```
sudo systemctl enable awslogsd.service
```

9. 代理程式執行一段時間後，您應該會在 CloudWatch主控台中看到新建立的記錄群組和記錄資料流。

   如需詳細資訊，請參閱 檢視傳送至 CloudWatch 記錄的記錄檔資料。

在現有的 Ubuntu 伺服器、CentOS 或紅帽執行個體上安裝及配置 CloudWatch 記錄

如果您使用的是執行 Ubuntu 伺服器、CentOS 或 Red Hat 的 AMI，請使用下列程序在您的執行個體上手動安裝 CloudWatch 記錄代理程式。

1. 連線至 EC2 執行個體。如需詳細資訊，請參閱 Amazon EC2 使用者指南中的 Connect 到您的執行個體。

   如需有關連線問題的詳細資訊，請參閱 Amazon EC2 使用者指南中的連線至執行個體疑難排解。
2. 使用兩個選項之一執行 CloudWatch Logs 代理程式安裝程式。您可以從網際網路直接執行或下載檔案並獨立執行。

> **Note**
>
> 如果您執行的是 CentOS 6.x、Red Hat 6.x 或 Ubuntu 12.04，請使用下載並執行獨立安裝程式的步驟。這些系統不支援直接從網際網路安裝 CloudWatch Logs 代理程式。

> **Note**
>
> 在 Ubuntu 上執行 `apt-get update`，再執行以下命令。

若要從網際網路直接執行，請使用下列命令並依照提示：

```
curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-setup.py -O
```

```
sudo python ./awslogs-agent-setup.py --region us-east-1
```

如果上述命令不適用，請嘗試：

```
sudo python3 ./awslogs-agent-setup.py --region us-east-1
```

若要下載並單獨執行，請使用下列命令並依照提示：

```
curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-setup.py -O
```

```
curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/AgentDependencies.tar.gz -O
```

```
tar xvf AgentDependencies.tar.gz -C /tmp/
```

```
sudo python ./awslogs-agent-setup.py --region us-east-1 --dependency-path /tmp/AgentDependencies
```

您可以藉由指定 us-east-1、us-west-2 部 -1、美國西部 -2、eu-central-1 南 -1、AP-東南 -2、AP-東南 -1、ap-northeast-1 或 sa-east-1 區域來安裝 CloudWatch 記錄代理程式。

> **Note**
>
> 如需 `awslogs-agent-setup` 目前版本與版本歷史記錄的更多資訊，請參閱 CHANGELOG.txt。

Lo CloudWatch gs 代理程式安裝程式在安裝期間需要特定資訊 開始之前，您需要知道要監控的日誌檔及其時間戳記格式。您也應備妥下列資訊。

項目	描述
AWS 存取金鑰識別碼	如果使用 IAM 角色，請按 Enter 鍵。否則，請輸入您的 AWS 存取金鑰 ID。
AWS 秘密訪問密鑰	如果使用 IAM 角色，請按 Enter 鍵。否則，請輸入您的 AWS 密鑰訪問密鑰。
預設區域名稱	按 Enter。預設為 us-east-2。這可以設為 us-east-1、us-west-1、us-west-2、ap-south-1、ap-northeast-2、ap-southeast-1、ap-southeast-2、ap-northeast-1、eu-central-1、eu-west-1 或 sa-east-1。
預設輸出格式	保留空白並按 Enter 鍵。
要上傳的日誌檔路徑	包含要傳送之日誌資料的檔案位置。安裝程式會為您建議路徑。
Destination Log Group 名稱	您的日誌群組名稱。安裝程式會為您建議日誌群組名稱。
Destination Log Stream 名稱	在預設情況下，此為主機名稱。安裝程式會為您建議主機名稱。
時間戳記格式	指定在指定日誌檔中的時間戳記格式。選擇自訂以指定您自己的格式。
起始地位	如何上傳資料。將此設為 start_of_file 以上傳資料檔案中的所有項目。設定為 end_of_file 以只上傳新附加的資料。

在您完成這些步驟後，安裝程式會要求設定另一個日誌檔。您可以對每個日誌檔以您想要的次數多次執行程序。如果您已經沒有任何要監控的日誌檔，請在安裝程式提示您設定另一個日誌時選擇 N (否)。如需代理程式組態檔案中設定的詳細資訊，請參閱 CloudWatch 記錄用戶端參考。

> **Note**
>
> 不支援設定多個日誌來源，將資料傳送到單個日誌串流。

3. 代理程式執行一段時間後，您應該會在 CloudWatch主控台中看到新建立的記錄群組和記錄資料流。

   如需詳細資訊，請參閱 檢視傳送至 CloudWatch 記錄的記錄檔資料。

快速入門：啟動時在 EC2 Linux 執行個體上安裝和設定 CloudWatch 日誌代理程式

> **Tip**
>
> 本節中討論的舊版 CloudWatch Logs 代理程式位於取代的路徑上。強烈建議您改用可同時收集記錄檔和指標的新整合 CloudWatch 代理程式。此外，較舊的 CloudWatch 日誌代理程式需要 Python 3.3 或更早版本，而這些版本預設不會安裝在新的 EC2 執行個體上。如需有關整合 CloudWatch 代理程式的詳細資訊，請參閱安裝 CloudWatch 代理程式。
> 本節的其餘部分將說明舊版 CloudWatch Logs 代理程式的使用。

啟動時在 EC2 Linux 執行個體上安裝較舊的 CloudWatch 日誌代理程式

您可以使用 Amazon EC2 使用者資料 (Amazon EC2 的一項功能)，可在啟動時將參數資訊傳遞至執行個體，以便在該執行個體上安裝和設定 CloudWatch 日誌代理程式。若要將 CloudWatch 日誌代理程式安裝和組態資訊傳遞給 Amazon EC2，您可以在網路位置 (例如 Amazon S3 儲存貯體) 提供組態檔案。

不支援設定多個日誌來源，將資料傳送到單個日誌串流。

先決條件

建立代理程式設定檔，其會描述您的所有日誌群組和日誌串流。這是一個文字檔案，其會描述要監控的日誌檔以及日誌群組和將要它們上傳至其中的日誌串流。代理程式會消耗此組態檔案，並開始監控和上傳所述的所有日誌檔。如需代理程式組態檔案中設定的詳細資訊，請參閱 CloudWatch 記錄用戶端參考。

以下是適用於 Amazon Linux 2 的代理程式組態檔案範例

```
[general]
state_file = /var/lib/awslogs/state/agent-state

[/var/log/messages]
file = /var/log/messages
log_group_name = /var/log/messages
log_stream_name = {instance_id}
datetime_format = %b %d %H:%M:%S
```

以下是適用於 Ubuntu 的範例代理程式組態檔案。

```
[general]
state_file = /var/awslogs/state/agent-state

[/var/log/syslog]
file = /var/log/syslog
log_group_name = /var/log/syslog
log_stream_name = {instance_id}
datetime_format = %b %d %H:%M:%S
```

設定 IAM 角色

1. 在以下網址開啟 IAM 主控台：https://console.aws.amazon.com/iam/。
2. 在導覽窗格中選擇 Policies (政策)、Create Policy (建立政策)。
3. 在 Create Policy (建立政策) 頁面上，針對 Create Your Own Policy (建立您自己的政策)，選擇 Select (選取)。如需有關建立自訂政策的詳細資訊，請參閱 Amazon EC2 使用者指南中的適用於 Amazon EC2 的 IAM 政策。
4. 在 Review Policy (檢閱政策) 頁面上的 Policy Name (政策名稱) 中，輸入該政策名稱。
5. 在 Policy Document (政策文件) 中，貼上以下政策：

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:DescribeLogStreams"
            ],
            "Resource": [
                "arn:aws:logs:*:*:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::myawsbucket/*"
            ]
        }
    ]
}
```

6. 選擇建立政策。
7. 在導覽窗格中，選擇 Roles (角色)、Create New Role (建立新角色)。
8. 在 Set Role Name (設定角色名稱) 頁面上，輸入該角色的名稱，然後選擇 Next Step (下一步)。
9. 在 Select Role Type (選取角色類型) 頁面上，選擇 Amazon EC2 旁的 Select (選取)。
10. 在 Attach Policy (連接政策) 頁面上的表格標頭中，選擇 Policy Type (政策類型)、Customer Managed (客戶受管理)。
11. 選取您建立的 IAM 政策，然後選擇 Next Step (下一步)。
12. 選擇建立角色。

    如需使用者和政策的詳細資訊，請參閱《IAM 使用者指南》中的 IAM 使用者和群組以及管理 IAM 政策。

啟動新執行個體並啟用 CloudWatch 記錄

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 選擇 Launch Instance (啟動執行個體)。

如需詳細資訊，請參閱 Amazon EC2 使用者指南中的啟動執行個體。

3. 在 Step 1: Choose an Amazon Machine Image (AMI) (步驟 1：選擇 Amazon Machine Image (AMI)) 頁面，選擇要啟動的 Linux 執行個體類型，然後在 Step 2: Choose an Instance Type (步驟 2：選擇執行個體類型) 頁面，選擇 Next: Configure Instance Details (下一步：設定執行個體詳細資料)。

   請確定 cloud-init 包含在您的 Amazon Machine Image (AMI) 中。Amazon Linux AMI 和 Ubuntu 和 RHEL 的 AMI 已經包含了雲初始化，但 CentOS 和其他 AMI 可能不會。 AWS Marketplace

4. 在 Step 3: Configure Instance Details (步驟 3：設定執行個體詳細資訊) 頁面上，針對 IAM role (IAM 角色)，選取您建立的 IAM 角色。

5. 在 Advanced Details (進階詳細資訊)，針對 User data (使用者資料) 將以下指令碼貼到方塊中。然後，透過將 -c 選項的值變更為代理程式組態檔案的位置來更新該指令碼：

   ```
   #!/bin/bash
   curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-
   setup.py -O
   chmod +x ./awslogs-agent-setup.py
   ./awslogs-agent-setup.py -n -r us-east-1 -c s3://DOC-EXAMPLE-BUCKET1/my-config-file
   ```

6. 對執行個體進行其他變更、檢閱啟動設定，然後選擇 Launch (啟動)。

7. 代理程式執行一段時間後，您應該會在 CloudWatch 主控台中看到新建立的記錄群組和記錄資料流。

   如需詳細資訊，請參閱 檢視傳送至 CloudWatch 記錄的記錄檔資料。

快速入門：讓執行 Windows 伺服器 2016 的 Amazon EC2 執行個體能夠使用日誌代理程式將 CloudWatch 日誌傳送到 CloudWatch 日誌

> **Tip**
>
> CloudWatch 包含新的統一代理程式，可從 EC2 執行個體和現場部署伺服器收集日誌和指標。我們建議您使用較新的整合 CloudWatch代理程式。如需詳細資訊，請參閱 開始使用 CloudWatch 記錄。
> 本節的其餘部分將說明舊版 CloudWatch Logs 代理程式的使用。

讓執行 Windows 伺服器 2016 的 Amazon EC2 執行個體能夠使用舊版日誌代理程式將 CloudWatch 日誌傳送到 CloudWatch 日誌

您可以使用多種方法來啟用執行 Windows 伺服器 2016 的執行個體，將記錄檔傳送至 CloudWatch 記錄檔。本節中的步驟使用 Systems Manager Run Command。如需其他可能方法的相關資訊，請參閱將日誌、事件和效能計數器傳送至 Amazon CloudWatch。

步驟

- 下載範例組態檔案
- 設定下列項目的 JSON 檔案 CloudWatch
- 建立 Systems Manager 的 IAM 角色
- 驗證 Systems Manager 先決條件
- 驗證網際網路存取
- 使用 Systems Manager 運行命令啟用 CloudWatch 日誌

下載範例組態檔案

將以下範例檔案下載到您的電腦：`AWS.EC2.Windows.CloudWatch.json`。

設定下列項目的 JSON 檔案 CloudWatch

您可以在組態檔中指定您的選擇，以決定要傳送到 CloudWatch 哪些記錄。建立此檔案並指定選擇的程序。需要 30 分鐘或更久的時間完成。一旦完成此工作，您就可以在所有的執行個體上重複使用該組態檔案。

步驟

- 步驟 1：啟用 CloudWatch 記錄
- 步驟 2：設定 CloudWatch
- 步驟 3：設定要傳送的資料
- 步驟 4：設定流程控制
- 步驟 5：儲存 JSON 內容

步驟 1：啟用 CloudWatch 記錄

在 JSON 檔案頂部，針對 `IsEnabled` 將「false」變更為「true」。

```
"IsEnabled": true,
```

步驟 2：設定 CloudWatch

指定憑證、區域、日誌群組名稱和日誌串流命名空間。這可讓執行個體將記錄資料傳送至 CloudWatch 記錄檔。若要將相同的記錄資料傳送至不同的位置，您可以新增具有唯一 ID 的其他區段 (例如，"CloudWatchLogs2" 和 CloudWatchLogs 3")，以及每個 ID 的不同區域。

設定將記錄檔資料傳送至 CloudWatch 記錄檔的設定

1. 在 JSON 檔案中，找到 CloudWatchLogs 區段。

```
{
    "Id": "CloudWatchLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "AccessKey": "",
        "SecretKey": "",
        "Region": "us-east-1",
        "LogGroup": "Default-Log-Group",
        "LogStream": "{instance_id}"
    }
},
```

2. 將 AccessKey 和 SecretKey 欄位保留空白。使用 IAM 角色設定憑證。
3. 針對 Region，輸入日誌資料要送往的區域 (例如，us-east-2)。
4. 針對 LogGroup，輸入您日誌群組的名稱。此名稱會出現在主 CloudWatch控台的「記錄群組」畫面上。
5. 針對 LogStream，輸入目的地日誌串流。此名稱會出現在 CloudWatch 主控台的「記錄群組 > 串流」畫面上。

   若使用 {instance_id} (預設值)，日誌串流名稱則為此執行個體的執行個體 ID。

   如果您指定的記錄資料流名稱不存在， CloudWatch Logs 會自動為您建立該名稱。您可以使用常值字串、預先定義的變數 {instance_id}、{hostname} 和 {ip_address}，或是組合這些項目，來定義日誌串流名稱。

步驟 3：設定要傳送的資料

您可以將事件記錄檔資料、Windows 事件追蹤 (ETW) 資料，以及其他記錄檔資料傳送至 CloudWatch 記錄檔。

將 Windows 應用程式事件記錄檔資料傳送至 CloudWatch 記錄檔

1. 在 JSON 檔案中，找到 `ApplicationEventLog` 區段。

```
{
    "Id": "ApplicationEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Application",
        "Levels": "1"
    }
},
```

2. 針對 `Levels`，指定要上傳訊息的類型。您可以指定下列其中一個值：

   - **1** - 僅上傳錯誤訊息。
   - **2** - 僅上傳警告訊息。
   - **4** - 僅上傳資訊訊息。

   您可以將值組合，以包含多個類型訊息。例如，值為 **3** 會上傳錯誤訊息 (**1**) 和警告訊息 (**2**)。值為 **7** 會上傳錯誤訊息 (**1**)、警告訊息 (**2**) 和資訊訊息 (**4**)。

將安全性記錄檔資料傳送至 CloudWatch 記錄

1. 在 JSON 檔案中，找到 `SecurityEventLog` 區段。

```
{
    "Id": "SecurityEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Security",
        "Levels": "7"
    }
```

```
},
```

2. 針對 Levels，輸入 **7** 以上傳所有訊息。

將系統事件記錄檔資料傳送至 CloudWatch 記錄

1. 在 JSON 檔案中，找到 SystemEventLog 區段。

```
{
    "Id": "SystemEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "System",
        "Levels": "7"
    }
},
```

2. 針對 Levels，指定要上傳訊息的類型。您可以指定下列其中一個值：

   - **1** - 僅上傳錯誤訊息。
   - **2** - 僅上傳警告訊息。
   - **4** - 僅上傳資訊訊息。

   您可以將值組合，以包含多個類型訊息。例如，值為 **3** 會上傳錯誤訊息 (**1**) 和警告訊息 (**2**)。值為 **7** 會上傳錯誤訊息 (**1**)、警告訊息 (**2**) 和資訊訊息 (**4**)。

將其他類型的事件記錄檔資料傳送至 CloudWatch 記錄

1. 在 JSON 檔案中新增區段。每個區段必須有唯一的 Id。

```
{
    "Id": "Id-name",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Log-name",
        "Levels": "7"
    }
```

```
},
```

2. 針對 Id，輸入名稱代表要上傳的日誌 (例如，**WindowsBackup**)。
3. 針對 LogName，輸入要上傳日誌的名稱。您可以搜尋日誌的名稱，如下所示。

   a. 開啟事件檢視器。

   b. 在導覽窗格中，選擇 Applications and Services Logs (應用程式與服務日誌)。

   c. 導覽至日誌，然後選擇 Actions (動作)、Properties (屬性)。
4. 針對 Levels，指定要上傳訊息的類型。您可以指定下列其中一個值：

   - **1** - 僅上傳錯誤訊息。
   - **2** - 僅上傳警告訊息。
   - **4** - 僅上傳資訊訊息。

   您可以將值組合，以包含多個類型訊息。例如，值為 **3** 會上傳錯誤訊息 (**1**) 和警告訊息 (**2**)。值為 **7** 會上傳錯誤訊息 (**1**)、警告訊息 (**2**) 和資訊訊息 (**4**)。

將 Windows 資料的事件追蹤傳送至 CloudWatch 記錄檔

ETW (Windows 的事件追蹤功能) 提供有效率和詳細的記錄機制，應用程式可將日誌寫入其中。每個 ETW 皆透過工作階段管理員控制，該管理程式可啟動和停止記錄工作階段。每個工作階段都有提供者和一個或多個使用者。

1. 在 JSON 檔案中，找到 ETW 區段。

```
{
    "Id": "ETW",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Microsoft-Windows-WinINet/Analytic",
        "Levels": "7"
    }
},
```

2. 針對 LogName，輸入要上傳日誌的名稱。
3. 針對 Levels，指定要上傳訊息的類型。您可以指定下列其中一個值：

- **1** - 僅上傳錯誤訊息。
- **2** - 僅上傳警告訊息。
- **4** - 僅上傳資訊訊息。

您可以將值組合，以包含多個類型訊息。例如，值為 **3** 會上傳錯誤訊息 (**1**) 和警告訊息 (**2**)。值為 **7** 會上傳錯誤訊息 (**1**)、警告訊息 (**2**) 和資訊訊息 (**4**)。

將自訂記錄檔 (任何以文字為基礎的記錄檔) 傳送至 CloudWatch 記錄

1. 在 JSON 檔案中，找到 `CustomLogs` 區段。

```
{
    "Id": "CustomLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\CustomLogs\\",
        "TimestampFormat": "MM/dd/yyyy HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "Local",
        "LineCount": "5"
    }
},
```

2. 針對 `LogDirectoryPath`，輸入在執行個體上存放日誌的路徑。
3. 針對 `TimestampFormat`，輸入要使用的時間戳記格式。如需關於支援之值的詳細資訊，請參閱 MSDN 上的自訂日期與時間格式字串主題。

> **Important**
>
> 您的原始日誌檔在每個日誌行的開始都必須有時間戳記，且在時間戳記後必須接著一個空格。

4. 針對 `Encoding`，輸入要使用的檔案編碼 (例如，UTF-8)。如需支援值的清單，請參閱 MSDN 上的 Encoding 類別主題。

> **Note**
>
> 使用編碼名稱，而非顯示名稱。

5. (選用) 針對 `Filter`，輸入日誌名稱的前綴。將此參數留白，以監控所有檔案。如需有關支援值的詳細資訊，請參閱 MSDN 上的FileSystemWatcherFilter 屬性主題。
6. (選用) 針對 `CultureName`，輸入要記錄時間戳記的地區設定。若 `CultureName` 留空，其會預設為與 Windows 執行個體目前使用的相同地區設定。如需詳細資訊，請參閱 MSDN 中Language tag產品行為主題中表格的 欄。

> **Note**
>
> 不支援 `div`、`div-MV`、`hu` 和 `hu-HU` 值。

7. (選用) 針對 `TimeZoneKind`，請輸入 `Local` 或 `UTC`。您可以設定此以在日誌時間戳記未包含任何時區資訊時提供時區資訊。如果此參數保留空白，而且您的時間戳記不包含時區資訊，則 CloudWatch 記錄會預設為當地時區。如果您的時間戳記已包含時區資訊，則會忽略此參數。
8. (選用) 針對 `LineCount`，輸入標頭中的行數，以辨識日誌檔案。例如，IIS 日誌檔幾乎都具有相同的標頭。您可以輸入 **5**，這會讀取日誌檔標頭的前三行來進行辨識。在 IIS 日誌檔中，第三行是日期和時間戳記，但不保證時間戳記在日誌檔間總是不同的。因此，我們建議包括至少一行實際的日誌資料，做為用來唯一辨識日誌檔的指紋。

若要將 IIS 記錄檔資料傳送至 CloudWatch 記錄

1. 在 JSON 檔案中，找到 `IISLog` 區段。

```
{
    "Id": "IISLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\inetpub\\logs\\LogFiles\\W3SVC1",
        "TimestampFormat": "yyyy-MM-dd HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "UTC",
        "LineCount": "5"
```

```
    }
},
```

2. 針對 LogDirectoryPath，輸入針對個別網站儲存 IIS 日誌的資料夾 (例如，C:\inetpub\logs\LogFiles\W3SVC*n*)。

> **Note**
>
> 只支援 W3C 日誌格式。不支援 IIS、NCSA 和自訂格式。

3. 針對 TimestampFormat，輸入要使用的時間戳記格式。如需關於支援之值的詳細資訊，請參閱 MSDN 上的自訂日期與時間格式字串主題。
4. 針對 Encoding，輸入要使用的檔案編碼 (例如，UTF-8)。如需關於支援之值的詳細資訊，請參閱 MSDN 上的編碼類別主題。

> **Note**
>
> 使用編碼名稱，而非顯示名稱。

5. (選用) 針對 Filter，輸入日誌名稱的前綴。將此參數留白，以監控所有檔案。如需有關支援值的詳細資訊，請參閱 MSDN 上的FileSystemWatcherFilter 屬性主題。
6. (選用) 針對 CultureName，輸入要記錄時間戳記的地區設定。若 CultureName 留空，其會預設為與 Windows 執行個體目前使用的相同地區設定。如需支援之值的詳細資訊，請參閱 MSDN 中Language tag產品行為主題中表格的 欄。

> **Note**
>
> 不支援 div、div-MV、hu 和 hu-HU 值。

7. (選用) 針對 TimeZoneKind，輸入 Local 或 UTC。您可以設定此以在日誌時間戳記未包含任何時區資訊時提供時區資訊。如果此參數保留空白，而且您的時間戳記不包含時區資訊，則 CloudWatch 記錄會預設為當地時區。如果您的時間戳記已包含時區資訊，則會忽略此參數。
8. (選用) 針對 LineCount，輸入標頭中的行數，以辨識日誌檔案。例如，IIS 日誌檔幾乎都具有相同的標頭。您可以輸入 **5**，這會讀取日誌檔標頭的前五行來進行辨識。在 IIS 日誌檔中，第三行是日期和時間戳記，但不保證時間戳記在日誌檔間總是不同的。因此，我們建議包括至少一行實際的日誌資料，以唯一辨識日誌檔的指紋。

步驟 4：設定流程控制

每個資料類型必須擁有在 Flows 區段中的相對應目的地。例如，若要將自訂記錄檔、ETW 記錄和系統記錄檔傳送至 CloudWatch 記錄，請新增(CustomLogs,ETW,SystemEventLog),CloudWatchLogs至Flows區段。

> **Warning**
>
> 加入無效的步驟會阻礙流程。例如，如果新增磁碟指標步驟，但執行個體沒有磁碟，則流程中的所有步驟都會遭到封鎖。

您可以將相同的日誌檔傳送到多個目的地。例如，如果要將應用程式日誌傳送到您在 CloudWatchLogs 區段中所定義的兩個不同目的地，請在 Flows 區段中加入 ApplicationEventLog,(CloudWatchLogs,CloudWatchLogs2)。

設定流程控制

1. 在 AWS.EC2.Windows.CloudWatch.json 檔案中，找到 Flows 區段。

```
"Flows": {
    "Flows": [
      "PerformanceCounter,CloudWatch",
      "(PerformanceCounter,PerformanceCounter2), CloudWatch2",
      "(CustomLogs, ETW, SystemEventLog),CloudWatchLogs",
      "CustomLogs, CloudWatchLogs2",
      "ApplicationEventLog,(CloudWatchLogs, CloudWatchLogs2)"
    ]
}
```

2. 針對 Flows，新增要上傳的每個資料類型 (例如，ApplicationEventLog) 及其目的地 (例如，CloudWatchLogs)。

步驟 5：儲存 JSON 內容

您現在已經完成編輯 JSON 檔案。進行儲存，並將檔案內容貼到另一個視窗中的文字編輯器。您將需要此程序在後續步驟中的檔案內容。

建立 Systems Manager 的 IAM 角色

當您使用 Systems Manager Run Command 時，需要執行個體憑證的 IAM 角色。此角色可讓 Systems Manager 在執行個體上執行動作。如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的設定 Systems Manager 的安全性角色。有關如何將 IAM 角色附加到現有執行個體的詳細資訊，請參閱 Amazon EC2 使用者指南中的將 IAM 角色附加到執行個體。

驗證 Systems Manager 先決條件

在您使用 Systems Manager 執行命令來設定與 CloudWatch 記錄檔的整合之前，請先確認您的執行個體符合最低需求。如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 Systems Manager 先決條件。

驗證網際網路存取

您的 Amazon EC2 Windows 伺服器執行個體和受管執行個體必須具有輸出網際網路存取權，才能將日誌和事件資料傳送至 CloudWatch。如需有關如何設定網際網路存取的詳細資訊，請參閱《Amazon VPC 使用者指南》中的網際網路閘道。

使用 Systems Manager 運行命令啟用 CloudWatch 日誌

Run Command (執行指令) 可讓您隨需管理執行個體的組態。您指定 Systems Manager 文件、指定參數，然後在一或多個執行個體上執行命令。執行個體上的 SSM Agent 會依指定來處理命令和設定執行個體。

使用執行命令設定與 CloudWatch 記錄的整合

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 開啟位於 https://console.aws.amazon.com/systems-manager/ 的 SSM 主控台。
3. 在導覽窗格中，選擇 執行命令。
4. 選擇 Run a command (執行指令)。
5. 對於命令文件，請選擇 AWS-ConfigureCloudWatch。
6. 對於 Target 執行個體，請選擇要與 CloudWatch 記錄整合的執行個體。如果執行個體未出現於此清單中，可能是無法針對 Run Command 進行設定。如需詳細資訊，請參閱 Amazon EC2 使用者指南中的 Systems Manager 先決條件。
7. 針對 Status (狀態)，請選擇 Enabled (啟用)。
8. 對於 Properties (屬性)，複製並貼上您在之前任務建立的 JSON 內容。
9. 完成填寫剩下的選填欄位，然後選擇 Run (執行)。

使用下列程序，在 Amazon EC2 主控台檢視命令執行的結果。

在主控台中檢視指令輸出

1. 選取指令。
2. 選擇 Output (輸出) 索引標籤。
3. 選擇 View Output (檢視輸出)。此指令輸出頁面會顯示指令執行的結果。

快速入門：啟用執行 Windows 伺服器 2012 和視窗伺服器 2008 的 Amazon EC2 執行個體，將日誌傳送到 CloudWatch 日誌

> **Tip**
>
> CloudWatch 包含新的統一代理程式，可從 EC2 執行個體和現場部署伺服器收集日誌和指標。我們建議您使用較新的整合 CloudWatch代理程式。如需詳細資訊，請參閱 開始使用 CloudWatch 記錄。
> 本節的其餘部分將說明舊版 CloudWatch Logs 代理程式的使用。

讓執行視窗伺服器 2012 和視窗伺服器 2008 的 Amazon EC2 執行個體能夠將日誌傳送到 CloudWatch 日誌

請使用下列步驟來啟用執行 Windows 伺服器 2012 和視窗伺服器 2008 的執行個體，將記錄檔傳送至 CloudWatch 記錄檔。

下載範例組態檔案

將以下範例 JSON 檔案下載到您的電腦：`AWS.EC2.Windows.CloudWatch.json`。在以下步驟進行編輯。

設定下列項目的 JSON 檔案 CloudWatch

您可以在 JSON 組態檔中指定您的選擇，以決定要傳送到 CloudWatch 哪些記錄。建立此檔案並指定選擇的程序。需要 30 分鐘或更久的時間完成。一旦完成此工作，您就可以在所有的執行個體上重複使用該組態檔案。

步驟

- 步驟 1：啟用 CloudWatch 記錄

- 步驟 2：設定 CloudWatch
- 步驟 3：設定要傳送的資料
- 步驟 4：設定流程控制

步驟 1：啟用 CloudWatch 記錄

在 JSON 檔案頂部，針對 IsEnabled 將「false」變更為「true」。

```
"IsEnabled": true,
```

步驟 2：設定 CloudWatch

指定憑證、區域、日誌群組名稱和日誌串流命名空間。這可讓執行個體將記錄資料傳送至 CloudWatch 記錄檔。若要將相同的記錄資料傳送至不同的位置，您可以新增具有唯一 ID 的其他區段 (例如，"CloudWatchLogs2" 和 CloudWatchLogs 3")，以及每個 ID 的不同區域。

設定將記錄檔資料傳送至 CloudWatch 記錄檔的設定

1. 在 JSON 檔案中，找到 CloudWatchLogs 區段。

```
{
    "Id": "CloudWatchLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "AccessKey": "",
        "SecretKey": "",
        "Region": "us-east-1",
        "LogGroup": "Default-Log-Group",
        "LogStream": "{instance_id}"
    }
},
```

2. 將 AccessKey 和 SecretKey 欄位保留空白。使用 IAM 角色設定憑證。
3. 針對 Region，輸入日誌資料要送往的區域 (例如，us-east-2)。
4. 針對 LogGroup，輸入您日誌群組的名稱。此名稱會出現在主 CloudWatch控台的「記錄群組」畫面上。
5. 針對 LogStream，輸入目的地日誌串流。此名稱會出現在 CloudWatch 主控台的「記錄群組 > 串流」畫面上。

若使用 {instance_id} (預設值)，日誌串流名稱則為此執行個體的執行個體 ID。

如果您指定的記錄資料流名稱不存在， CloudWatch Logs 會自動為您建立該名稱。您可以使用常值字串、預先定義的變數 {instance_id}、{hostname} 和 {ip_address}，或是組合這些項目，來定義日誌串流名稱。

步驟 3：設定要傳送的資料

您可以將事件記錄檔資料、Windows 事件追蹤 (ETW) 資料，以及其他記錄檔資料傳送至 CloudWatch 記錄檔。

將 Windows 應用程式事件記錄檔資料傳送至 CloudWatch 記錄檔

1. 在 JSON 檔案中，找到 ApplicationEventLog 區段。

```
{
    "Id": "ApplicationEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Application",
        "Levels": "1"
    }
},
```

2. 針對 Levels，指定要上傳訊息的類型。您可以指定下列其中一個值：

   - **1** - 僅上傳錯誤訊息。
   - **2** - 僅上傳警告訊息。
   - **4** - 僅上傳資訊訊息。

   您可以將值組合，以包含多個類型訊息。例如，值為 **3** 會上傳錯誤訊息 (**1**) 和警告訊息 (**2**)。值為 **7** 會上傳錯誤訊息 (**1**)、警告訊息 (**2**) 和資訊訊息 (**4**)。

將安全性記錄檔資料傳送至 CloudWatch 記錄

1. 在 JSON 檔案中，找到 SecurityEventLog 區段。

```
{
    "Id": "SecurityEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Security",
        "Levels": "7"
    }
},
```

2. 針對 Levels，輸入 **7** 以上傳所有訊息。

將系統事件記錄檔資料傳送至 CloudWatch 記錄

1. 在 JSON 檔案中，找到 SystemEventLog 區段。

```
{
    "Id": "SystemEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "System",
        "Levels": "7"
    }
},
```

2. 針對 Levels，指定要上傳訊息的類型。您可以指定下列其中一個值：

   - **1** - 僅上傳錯誤訊息。
   - **2** - 僅上傳警告訊息。
   - **4** - 僅上傳資訊訊息。

   您可以將值組合，以包含多個類型訊息。例如，值為 **3** 會上傳錯誤訊息 (**1**) 和警告訊息 (**2**)。值為 **7** 會上傳錯誤訊息 (**1**)、警告訊息 (**2**) 和資訊訊息 (**4**)。

將其他類型的事件記錄檔資料傳送至 CloudWatch 記錄

1. 在 JSON 檔案中新增區段。每個區段必須有唯一的 Id。

```
{
    "Id": "Id-name",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Log-name",
        "Levels": "7"
    }
},
```

2. 針對 Id，輸入名稱代表要上傳的日誌 (例如，**WindowsBackup**)。
3. 針對 LogName，輸入要上傳日誌的名稱。您可以搜尋日誌的名稱，如下所示。

   a. 開啟事件檢視器。

   b. 在導覽窗格中，選擇 Applications and Services Logs (應用程式與服務日誌)。

   c. 導覽至日誌，然後選擇 Actions (動作)、Properties (屬性)。
4. 針對 Levels，指定要上傳訊息的類型。您可以指定下列其中一個值：

   - **1** - 僅上傳錯誤訊息。
   - **2** - 僅上傳警告訊息。
   - **4** - 僅上傳資訊訊息。

   您可以將值組合，以包含多個類型訊息。例如，值為 **3** 會上傳錯誤訊息 (**1**) 和警告訊息 (**2**)。值為 **7** 會上傳錯誤訊息 (**1**)、警告訊息 (**2**) 和資訊訊息 (**4**)。

將 Windows 資料的事件追蹤傳送至 CloudWatch 記錄檔

ETW (Windows 的事件追蹤功能) 提供有效率和詳細的記錄機制，應用程式可將日誌寫入其中。每個 ETW 皆透過工作階段管理員控制，該管理程式可啟動和停止記錄工作階段。每個工作階段都有提供者和一個或多個使用者。

1. 在 JSON 檔案中，找到 ETW 區段。

```
{
    "Id": "ETW",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
```

```
        "LogName": "Microsoft-Windows-WinINet/Analytic",
        "Levels": "7"
    }
},
```

2. 針對 LogName，輸入要上傳日誌的名稱。
3. 針對 Levels，指定要上傳訊息的類型。您可以指定下列其中一個值：

   - **1** - 僅上傳錯誤訊息。
   - **2** - 僅上傳警告訊息。
   - **4** - 僅上傳資訊訊息。

   您可以將值組合，以包含多個類型訊息。例如，值為 **3** 會上傳錯誤訊息 (**1**) 和警告訊息 (**2**)。值為 **7** 會上傳錯誤訊息 (**1**)、警告訊息 (**2**) 和資訊訊息 (**4**)。

將自訂記錄檔 (任何以文字為基礎的記錄檔) 傳送至 CloudWatch 記錄

1. 在 JSON 檔案中，找到 CustomLogs 區段。

```
{
    "Id": "CustomLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\CustomLogs\\",
        "TimestampFormat": "MM/dd/yyyy HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "Local",
        "LineCount": "5"
    }
},
```

2. 針對 LogDirectoryPath，輸入在執行個體上存放日誌的路徑。
3. 針對 TimestampFormat，輸入要使用的時間戳記格式。如需關於支援之值的詳細資訊，請參閱 MSDN 上的自訂日期與時間格式字串主題。

> **Important**
>
> 您的原始日誌檔在每個日誌行的開始都必須有時間戳記，且在時間戳記後必須接著一個空格。

4. 針對 Encoding，輸入要使用的檔案編碼 (例如，UTF-8)。如需關於支援之值的詳細資訊，請參閱 MSDN 上的編碼類別主題。

> **Note**
>
> 使用編碼名稱，而非顯示名稱。

5. (選用) 針對 Filter，輸入日誌名稱的前綴。將此參數留白，以監控所有檔案。如需有關支援值的詳細資訊，請參閱 MSDN 上的FileSystemWatcherFilter 屬性主題。
6. (選用) 針對 CultureName，輸入要記錄時間戳記的地區設定。若 CultureName 留空，其會預設為與 Windows 執行個體目前使用的相同地區設定。如需支援之值的詳細資訊，請參閱 MSDN 中Language tag產品行為主題中表格的 欄。

> **Note**
>
> 不支援 div、div-MV、hu 和 hu-HU 值。

7. (選用) 針對 TimeZoneKind，請輸入 Local 或 UTC。您可以設定此以在日誌時間戳記未包含任何時區資訊時提供時區資訊。如果此參數保留空白，而且您的時間戳記不包含時區資訊，則 CloudWatch 記錄會預設為當地時區。如果您的時間戳記已包含時區資訊，則會忽略此參數。
8. (選用) 針對 LineCount，輸入標頭中的行數，以辨識日誌檔案。例如，IIS 日誌檔幾乎都具有相同的標頭。您可以輸入 **5**，這會讀取日誌檔標頭的前三行來進行辨識。在 IIS 日誌檔中，第三行是日期和時間戳記，但不保證時間戳記在日誌檔間總是不同的。因此，我們建議包括至少一行實際的日誌資料，做為用來唯一辨識日誌檔的指紋。

若要將 IIS 記錄檔資料傳送至 CloudWatch 記錄

1. 在 JSON 檔案中，找到 IISLog 區段。

```
{
    "Id": "IISLogs",
```

```
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\inetpub\\logs\\LogFiles\\W3SVC1",
        "TimestampFormat": "yyyy-MM-dd HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "UTC",
        "LineCount": "5"
    }
},
```

2. 針對 LogDirectoryPath，輸入針對個別網站儲存 IIS 日誌的資料夾 (例如，C:\inetpub\logs\LogFiles\W3SVC*n*)。

> **Note**
>
> 只支援 W3C 日誌格式。不支援 IIS、NCSA 和自訂格式。

3. 針對 TimestampFormat，輸入要使用的時間戳記格式。如需關於支援之值的詳細資訊，請參閱 MSDN 上的自訂日期與時間格式字串主題。
4. 針對 Encoding，輸入要使用的檔案編碼 (例如，UTF-8)。如需關於支援之值的詳細資訊，請參閱 MSDN 上的編碼類別主題。

> **Note**
>
> 使用編碼名稱，而非顯示名稱。

5. (選用) 針對 Filter，輸入日誌名稱的前綴。將此參數留白，以監控所有檔案。如需有關支援值的詳細資訊，請參閱 MSDN 上的FileSystemWatcherFilter 屬性主題。
6. (選用) 針對 CultureName，輸入要記錄時間戳記的地區設定。若 CultureName 留空，其會預設為與 Windows 執行個體目前使用的相同地區設定。如需支援之值的詳細資訊，請參閱 MSDN 中Language tag產品行為主題中表格的 欄。

> **Note**
>
> 不支援 div、div-MV、hu 和 hu-HU 值。

7. (選用) 針對 TimeZoneKind，輸入 Local 或 UTC。您可以設定此以在日誌時間戳記未包含任何時區資訊時提供時區資訊。如果此參數保留空白，而且您的時間戳記不包含時區資訊，則 CloudWatch 記錄會預設為當地時區。如果您的時間戳記已包含時區資訊，則會忽略此參數。
8. (選用) 針對 LineCount，輸入標頭中的行數，以辨識日誌檔案。例如，IIS 日誌檔幾乎都具有相同的標頭。您可以輸入 **5**，這會讀取日誌檔標頭的前五行來進行辨識。在 IIS 日誌檔中，第三行是日期和時間戳記，但不保證時間戳記在日誌檔間總是不同的。因此，我們建議包括至少一行實際的日誌資料，以唯一辨識日誌檔的指紋。

步驟 4：設定流程控制

每個資料類型必須擁有在 Flows 區段中的相對應目的地。例如，若要將自訂記錄檔、ETW 記錄和系統記錄檔傳送至 CloudWatch 記錄，請新增(CustomLogs,ETW,SystemEventLog),CloudWatchLogs至Flows區段。

> **Warning**
>
> 加入無效的步驟會阻礙流程。例如，如果新增磁碟指標步驟，但執行個體沒有磁碟，則流程中的所有步驟都會遭到封鎖。

您可以將相同的日誌檔傳送到多個目的地。例如，如果要將應用程式日誌傳送到您在 CloudWatchLogs 區段中所定義的兩個不同目的地，請在 Flows 區段中加入 ApplicationEventLog,(CloudWatchLogs,CloudWatchLogs2)。

設定流程控制

1. 在 AWS.EC2.Windows.CloudWatch.json 檔案中，找到 Flows 區段。

```
"Flows": {
    "Flows": [
      "PerformanceCounter,CloudWatch",
      "(PerformanceCounter,PerformanceCounter2), CloudWatch2",
      "(CustomLogs, ETW, SystemEventLog),CloudWatchLogs",
      "CustomLogs, CloudWatchLogs2",
      "ApplicationEventLog,(CloudWatchLogs, CloudWatchLogs2)"
    ]
}
```

2. 針對 Flows，新增要上傳的每個資料類型 (例如，ApplicationEventLog) 及其目的地 (例如，CloudWatchLogs)。

您現在已經完成編輯 JSON 檔案。在後續步驟中會使用到此檔案。

啟動代理程式

若要啟用執行視窗伺服器 2012 或 Windows 伺服器 2008 的 Amazon EC2 執行個體將日誌傳送到 CloudWatch 日誌，請使用 EC2Config 服務 (. EC2Config.exe) 您的執行個體應該擁有 EC2Config 4.0 或更新版本，而且您可以使用此程序。如需使用舊版 EC2Config 的詳細資訊，請參閱 Amazon EC2 使用者指南 CloudWatch中的使用 EC2Config 3.x 或更早版本進行設定

若要 CloudWatch 使用 EC2Config 4.x 進行配置

1. 對您稍早在此程序中編輯的 AWS.EC2.Windows.CloudWatch.json 檔案檢查編碼。只支援無 BOM 的 UTF-8 編碼。接著，在 Windows Server 2008 – 2012 R2 執行個體的以下資料夾儲存檔案：C:\Program Files\Amazon\SSM\Plugins\awsCloudWatch\。
2. 使用 Windows 服務控制台或使用下列 PowerShell 命令來啟動或重新啟動 SSM 代理程式 (AmazonSSMAgent.exe)：

```
PS C:\> Restart-Service AmazonSSMAgent
```

SSM 代理程式重新啟動之後，它會偵測組態檔並設定執行個體以進行整合。 CloudWatch 如果變更本機組態檔案中的參數與設定，您需要重新啟動 SSM Agent 來反映變更。若要停用執行個體的 CloudWatch 整合，請在組態檔案中變更IsEnabledfalse並儲存您的變更。

快速入門：使用 AWS OpsWorks 和 Chef 安裝 CloudWatch 日誌代理

您可以使用 AWS OpsWorks and Chef (協力廠商系統和雲端基礎結構自動化工具) 來安裝 CloudWatch 記錄代理程式並建立記錄資料流。Chef 使用「配方」(您寫入在您的電腦上安裝和設定軟體的配方)，以及「說明書」(此為配方的集合) 來執行其組態和政策分發任務。如需詳細資訊，請參閱 Chef。

以下 Chef 配方範例說明如何監控在每個 EC2 執行個體上的一個日誌檔。該配方使用堆疊名稱做為日誌群組和執行個體的主機名稱做為日誌串流名稱。為了監控多個日誌檔，您需要擴展配方來建立多個日誌群組和日誌串流。

步驟 1：建立自訂配方

創建一個存儲庫來存儲您的食譜。 AWS OpsWorks 支持 Git 和顛覆，或者您可以將存檔存儲在 Amazon S3 中。《AWS OpsWorks 使用者指南》中的逐步指南儲存庫描述逐步指南儲存庫的結構。以下範例假設說明書名為 `logs`。安裝 .rb 方案會安裝 CloudWatch 記錄代理程式。您也可以下載食譜示例（CloudWatchLogs-Cookbooks.zip）。

建立名為 metadata.rb 的檔案，其中包含以下程式碼：

```
#metadata.rb

name             'logs'
version          '0.0.1'
```

創建日 CloudWatch 誌配置文件：

```
#config.rb

template "/tmp/cwlogs.cfg" do
  cookbook "logs"
  source "cwlogs.cfg.erb"
  owner "root"
  group "root"
  mode 0644
end
```

下載並安裝 CloudWatch 記錄代理程式：

```
# install.rb

directory "/opt/aws/cloudwatch" do
  recursive true
end

remote_file "/opt/aws/cloudwatch/awslogs-agent-setup.py" do
  source "https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-
setup.py"
  mode "0755"
end

  execute "Install CloudWatch Logs agent" do
  command "/opt/aws/cloudwatch/awslogs-agent-setup.py -n -r region -c /tmp/cwlogs.cfg"
```

```
  not_if { system "pgrep -f aws-logs-agent-setup" }
end
```

> **Note**
>
> 在上述範例中，將 *region* 換成以下其中一個：us-east-1、us-west-1、us-west-2、ap-south-1、ap-northeast-2、ap-southeast-1、ap-southeast-2、ap-northeast-1、eu-central-1、eu-west-1 或 sa-east-1。

如果安裝代理程式失敗，檢查以確保已安裝 python-dev 套件。如果沒有，請使用下列命令，然後重試代理程式安裝：

```
sudo apt-get -y install python-dev
```

這個配方使用 cwlogs.cfg.erb 範本檔案，您可以對其修改以指定各種屬性 (例如要記錄哪些檔案)。如需這些屬性的相關資訊，請參閱 CloudWatch 記錄用戶端參考。

```
[general]
# Path to the AWSLogs agent's state file. Agent uses this file to maintain
# client side state across its executions.
state_file = /var/awslogs/state/agent-state


## Each log file is defined in its own section. The section name doesn't
## matter as long as its unique within this file.
#
#[kern.log]
#
## Path of log file for the agent to monitor and upload.
#
#file = /var/log/kern.log
#
## Name of the destination log group.
#
#log_group_name = kern.log
#
## Name of the destination log stream.
#
#log_stream_name = {instance_id}
#
```

```
## Format specifier for timestamp parsing.
#
#datetime_format = %b %d %H:%M:%S
#
#

[<%= node[:opsworks][:stack][:name] %>]
datetime_format = [%Y-%m-%d %H:%M:%S]
log_group_name = <%= node[:opsworks][:stack][:name].gsub(' ','_') %>
file = <%= node[:cwlogs][:logfile] %>
log_stream_name = <%= node[:opsworks][:instance][:hostname] %>
```

該範本會參照堆疊組態和部署 JSON 中的對應屬性以取得堆疊名稱和主機名稱。指定要記錄檔案的此屬性定義在 cwlogs 說明書的 default.rb 屬性檔案 (logs/attributes/default.rb) 中。

```
default[:cwlogs][:logfile] = '/var/log/aws/opsworks/opsworks-agent.statistics.log'
```

步驟 2：建立 AWS OpsWorks 堆疊

1. 請在以下位置開啟 AWS OpsWorks 主控台。 https://console.aws.amazon.com/opsworks/
2. 在OpsWorks 儀表板上，選擇 [新增堆疊] 以建立 AWS OpsWorks 堆疊。
3. 在 Add stack (新增堆疊) 畫面中，選擇 Chef 11 stack (Chef 11 堆疊)。
4. 在 Stack name (堆疊名稱) 中，輸入名稱。
5. 對於 Use custom Chef Cookbooks (使用自訂 Chef 說明書)，選擇 Yes (是)。
6. 對於 Repository type (儲存庫類型)，選取您要使用的儲存庫類型。如果您使用的是上述範例，請選擇 Http Archive (Http 封存)。
7. 對於 Repository URL (儲存庫 URL)，請輸入您要將在先前步驟建立的說明書儲存在其中的儲存庫。如果您使用的是上述範例，請輸入 **`https://s3.amazonaws.com/aws-cloudwatch/downloads/CloudWatchLogs-Cookbooks.zip`**。
8. 選擇 Add Stack (新增推疊) 以建立堆疊。

步驟 3：擴展 IAM 角色

若要搭配 AWS OpsWorks 執行個體使用 CloudWatch Logs，您需要擴充執行個體使用的 IAM 角色。

1. 在以下網址開啟 IAM 主控台：https://console.aws.amazon.com/iam/。
2. 在導覽窗格中選擇 Policies (政策)、Create Policy (建立政策)。

3. 在 Create Policy (建立政策) 頁面上，請在 Create Your Own Policy (建立自己的政策) 下選擇 Select (選取)。如需有關建立自訂政策的詳細資訊，請參閱 Amazon EC2 使用者指南中的適用於 Amazon EC2 的 IAM 政策。
4. 在 Review Policy (檢閱政策) 頁面上的 Policy Name (政策名稱) 中，輸入該政策名稱。
5. 在 Policy Document (政策文件) 中，貼上以下政策：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
    ],
    "Resource": [
      "arn:aws:logs:*:*:*"
    ]
  }
 ]
}
```

6. 選擇建立政策。
7. 在瀏覽窗格中，選擇 [角色]，然後在內容窗格中，針對 [角色名稱] 選取 AWS OpsWorks 堆疊所使用的執行個體角色名稱。您可以在堆疊設定中找到您的堆疊使用的執行個體角色名稱 (預設值為 `aws-opsworks-ec2-role`)。

> **Note**
>
> 選擇角色名稱 (非核取方塊)。

8. 在 Permissions(許可) 索引標籤的 Managed Policies (受管政策) 中，選擇 Attach Policy (連接政策)。
9. 在 Attach Policy (連接政策) 頁面，在表格標頭中 (Filter (篩選條件) 和 Search (搜尋) 旁)，請選擇 Policy Type (政策類型)、Customer Managed Policies (客戶受管政策)。
10. 針對 Customer Managed Policies (客戶受管政策)，選取您在上方建立的 IAM 政策，然後選擇 Attach Policy (連接政策)。

如需使用者和政策的詳細資訊，請參閱《IAM 使用者指南》中的 IAM 使用者和群組以及管理 IAM 政策。

步驟 4：新增層

1. 請在以下位置開啟 AWS OpsWorks 主控台。 https://console.aws.amazon.com/opsworks/
2. 在導覽視窗中，選擇 圖層。
3. 在內容窗格中選取 layer，然後選擇 Add layer (新增 layer)。
4. 在OpsWorks標籤上，對於「圖層類型」，選擇「自訂」。
5. 對於 Name (名稱) 和 Short name (短名稱) 欄位中，輸入層的長短名稱，然後選擇 Add layer (新增層)。
6. 在 [配方] 索引標籤的 [自訂廚師方法] 底下，有數個標題 (設定、設定、部署、取消部署和關閉)，這些標題與 AWS OpsWorks 生命週期事件相對應。 AWS OpsWorks 在執行個體生命週期中的這些關鍵點觸發這些事件，這些事件會執行相關的配方。

   > **Note**
   >
   > 如果沒有顯示上述標題，在 Custom Chef Recipes (自訂 Chef 配方) 下，請選擇 edit (編輯)。

7. 在 Setup (設定) 旁輸入 logs::config, logs::install，選擇 + 以將其新增到清單，然後選擇 Save (儲存)。

   AWS OpsWorks 在執行個體啟動之後，立即在此層中的每個新執行個體上執行此配方。

步驟 5：新增執行個體

該層只控制如何設定執行個體。您現在需要將一些執行個體新增到層，並啟動它們。

1. 請在以下位置開啟 AWS OpsWorks 主控台。 https://console.aws.amazon.com/opsworks/
2. 在導覽窗格中，選擇 Instances (執行個體)，然後選擇在層之下的 + Instance (+ 執行個體)。
3. 接受預設的設定，然後選擇 Add Instance (新增執行個體) 以將執行個體新增到層。
4. 在資料列的 Actions (動作) 欄，按一下 start (開始) 以啟動執行個體。

AWS OpsWorks 啟動新的 EC2 執行個體並設定 CloudWatch 日誌。該執行個體的狀態會在準備好時變更為線上。

步驟 6：檢視您的日誌

代理程式執行一段時間後，您應該會在 CloudWatch 主控台中看到新建立的記錄群組和記錄資料流。

如需詳細資訊，請參閱 檢視傳送至 CloudWatch 記錄的記錄檔資料。

報告 CloudWatch 記錄檔代理程式狀態

使用下列程序來報告 EC2 執行個體上 CloudWatch 日誌代理程式的狀態。

報告代理程式的狀態

1. 連線至 EC2 執行個體。如需詳細資訊，請參閱 Amazon EC2 使用者指南中的 Connect 到您的執行個體。

   如需有關連線問題的詳細資訊，請參閱 Amazon EC2 使用者指南中的連線至執行個體疑難排解
2. 在命令提示，請輸入下列命令：

   ```
   sudo service awslogs status
   ```

   如果您執行的是 Amazon Linux 2，請輸入下列命令：

   ```
   sudo service awslogsd status
   ```
3. 檢查 /var/log/awslogs.log 檔案是否有任何錯誤、警告或 CloudWatch 記錄代理程式的問題。

啟動 CloudWatch 記錄代理程式

如果 EC2 執行個體上的 CloudWatch Logs 代理程式在安裝後未自動啟動，或者您停止了代理程式，則可以使用下列程序啟動代理程式。

啟動代理程式

1. 連線至 EC2 執行個體。如需詳細資訊，請參閱 Amazon EC2 使用者指南中的 Connect 到您的執行個體。

如需有關連線問題的詳細資訊，請參閱 Amazon EC2 使用者指南中的連線至執行個體疑難排解。

2. 在命令提示，請輸入下列命令：

```
sudo service awslogs start
```

如果您執行的是 Amazon Linux 2，請輸入下列命令：

```
sudo service awslogsd start
```

停止 CloudWatch 記錄檔代理程式

使用下列程序停止 EC2 執行個體上的 CloudWatch 記錄代理程式。

停止代理程式

1. 連線至 EC2 執行個體。如需詳細資訊，請參閱 Amazon EC2 使用者指南中的 Connect 到您的執行個體。

如需有關連線問題的詳細資訊，請參閱 Amazon EC2 使用者指南中的連線至執行個體疑難排解。

2. 在命令提示，請輸入下列命令：

```
sudo service awslogs stop
```

如果您執行的是 Amazon Linux 2，請輸入下列命令：

```
sudo service awslogsd stop
```

快速入門：用 AWS CloudFormation 來開始使用 CloudWatch 記錄

AWS CloudFormation 可讓您描述和佈建 JSON 格式的 AWS 資源。這種方法的優點包括能夠以單一單元的形式管理 AWS 資源集合，並輕鬆地跨區域複製資 AWS 源。

AWS 使用佈建時 AWS CloudFormation，您會建立描述要使用之 AWS 資源的範本。以下範例是一種範本程式碼片段，其會建立一個日誌群組和指標篩選條件以計算 404 發生次數並將此計數傳送至日誌群組。

```
"WebServerLogGroup": {
    "Type": "AWS::Logs::LogGroup",
    "Properties": {
        "RetentionInDays": 7
    }
},

"404MetricFilter": {
    "Type": "AWS::Logs::MetricFilter",
    "Properties": {
        "LogGroupName": {
            "Ref": "WebServerLogGroup"
        },
        "FilterPattern": "[ip, identity, user_id, timestamp, request, status_code =
 404, size, ...]",
        "MetricTransformations": [
            {
                "MetricValue": "1",
                "MetricNamespace": "test/404s",
                "MetricName": "test404Count"
            }
        ]
    }
}
```

這是一個基本的範例。您可以 CloudWatch 使用 AWS CloudFormation. 如需範本範例的詳細資訊，請參閱AWS CloudFormation 使用者指南中的 Amazon CloudWatch 日誌範本程式碼片段。如需有關入門的詳細資訊，請參閱《AWS CloudFormation 使用者指南》中的 AWS CloudFormation入門。

搭配 AWS SDK 使用 CloudWatch 記錄檔

AWS 軟件開發套件（SDK）可用於許多流行的編程語言。每個 SDK 都提供 API、程式碼範例和說明文件，讓開發人員能夠更輕鬆地以偏好的語言建置應用程式。

SDK 文件	代碼範例
AWS SDK for C++	AWS SDK for C++ 程式碼範例
AWS CLI	AWS CLI 程式碼範例
AWS SDK for Go	AWS SDK for Go 程式碼範例
AWS SDK for Java	AWS SDK for Java 程式碼範例
AWS SDK for JavaScript	AWS SDK for JavaScript 程式碼範例
適用於 Kotlin 的 AWS SDK	適用於 Kotlin 的 AWS SDK 程式碼範例
AWS SDK for .NET	AWS SDK for .NET 程式碼範例
AWS SDK for PHP	AWS SDK for PHP 程式碼範例
AWS Tools for PowerShell	PowerShell 程式碼範例工具
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) 程式碼範例
AWS SDK for Ruby	AWS SDK for Ruby 程式碼範例
適用於 Rust 的 AWS SDK	適用於 Rust 的 AWS SDK 程式碼範例
適用於 SAP ABAP 的 AWS SDK	適用於 SAP ABAP 的 AWS SDK 程式碼範例
適用於 Swift 的 AWS SDK	適用於 Swift 的 AWS SDK 程式碼範例

如需 CloudWatch 記錄檔的特定範例，請參閱使用 AWS SDK 的 CloudWatch 記錄檔的程式碼範例。

可用性範例

找不到所需的內容嗎？ 請使用本頁面底部的提供意見回饋連結申請程式碼範例。

使用日誌見解分析 CloudWatch 日誌資料

您可以使用 CloudWatch 日誌深入解析，以互動方式搜尋和分析 Amazon CloudWatch Logs 中的日誌資料。您可以執行查詢，協助您有效率地回應操作問題。如果發生問題，您可以使用 CloudWatch 日誌深入解析來識別潛在原因並驗證已部署的修正程式。

CloudWatch Logs Insights 包含專門建置的查詢語言，其中包含一些簡單但功能強大的命令。CloudWatch Logs Insights 提供範例查詢、命令說明、查詢自動完成和記錄欄位探索，協助您開始使用。附有適用於多種 AWS 服務日誌的範例查詢。

CloudWatch 日誌洞見會自動探索來自 Amazon Route 53、、和 Amazon VPC 等 AWS 服務的日誌中的欄位 AWS Lambda AWS CloudTrail，以及任何以 JSON 形式發出日誌事件的應用程式或自訂日誌。

您可以使用 CloudWatch 記錄深入分析來搜尋 2018 年 11 月 5 日或更新版本傳送至 CloudWatch 記錄檔的記錄檔資料。

> **Important**
>
> CloudWatch 日誌見解無法存取日誌事件的時間戳記，這些時間戳記會在記錄群組的建立時間之前。

您也可以使用自然語言建立 CloudWatch 日誌見解查詢。因此，請提出問題或描述您正在尋找的資料。此 AI 輔助功能會根據您的提示產生查詢，並提供查詢運作方式的 line-by-line 說明。如需詳細資訊，請參閱使用自然語言產生和更新 CloudWatch 記錄見解查詢。

如果您已登入設定為 CloudWatch 跨帳戶觀察性監視帳戶的帳戶，則可以針對連結至此監視帳戶的來源帳戶中的 CloudWatch 記錄群組執行 Logs Insights 查詢。您可以查詢位於不同帳戶中的多個日誌群組。如需詳細資訊，請參閱CloudWatch 跨帳戶可觀察性。

單一請求最多可查詢 50 個日誌群組。如果查詢尚未完成，則會在 60 分鐘後逾時。查詢結果可以保留 7 天。

您可以儲存已建立的查詢。這麼做可幫助您在需要時執行複雜的查詢，而不需要在每次執行時都重新建立查詢。

CloudWatch 記錄見解查詢會根據查詢的資料量產生費用。如需詳細資訊，請參閱 Amazon CloudWatch 定價。

> **Important**
>
> 如果您的網路安全性團隊不允許使用 Web 通訊端，您目前無法存取 CloudWatch 主控台的「CloudWatch 記錄見解」部分。您可以使用 API 使用 CloudWatch 日誌見解查詢功能。如需詳細資訊，請參閱 Amazon CloudWatch 日誌 API 參考StartQuery中的。

目錄

記錄類別中支援的命令

標準 CloudWatch 記錄類別中的記錄群組支援所有記錄見解查詢命令。「不常存取」記錄檔類別中的記錄群組支援除`pattern`、`diff`和以外的所有查詢命令。`unmask`

開始使用：查詢教學課程

下列各節包含範例查詢教學課程，可協助您開始使用 CloudWatch 日誌深入解析。

主題

- 教學課程：執行查詢以產生依日誌欄位分組的視覺效果
- 教學課程：執行查詢來產生時間序列視覺化

教學課程：執行和修改範例查詢

下列教學課程可協助您開始使用 CloudWatch 日誌深入解析。您將會執行範例查詢，然後了解如何修改和重新執行它。

若要執行查詢，您必須已經在記錄檔中儲存了 CloudWatch 記錄檔。如果您已經在使用 CloudWatch 記錄檔，且已設定記錄群組和記錄串流，就可以開始使用了。如果您使用 Amazon Route 53 或 Amazon VPC 等服務 AWS CloudTrail，並且已經從這些服務設定了記錄檔以移至日誌，則您可能也已經擁有 CloudWatch 日誌。如需將記錄檔傳送至 CloudWatch 記錄檔的詳細資訊，請參閱開始使用 CloudWatch 記錄。

CloudWatch Logs Insights 中的查詢會傳回記錄事件中的一組欄位，或是數學彙總的結果，或是針對記錄事件執行的其他作業。本教學課程示範的查詢會傳回日誌事件清單。

執行範例查詢

若要執行 CloudWatch 日誌見解範例查詢

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇 Logs (日誌)，然後選擇 Logs Insights (日誌洞察)。

   在 Logs Insights (日誌洞察) 頁面上，查詢編輯器包含會傳回 20 筆最新日誌事件的預設查詢。
3. 在 Select log group(s) (選取日誌群組) 下拉式選單中，選擇一個或多個要查詢的日誌群組。

   如果這是 CloudWatch 跨帳戶觀察性的監視帳戶，您可以在來源帳戶和監視帳戶中選取記錄群組。單一查詢可以一次查詢來自不同帳戶的日誌。

   您可以依日誌群組名稱、帳戶 ID 或帳戶標籤來篩選日誌群組。

   當您在 [標準] 記錄類別中選取記錄群組時， CloudWatch Logs Insights 會自動偵測群組中的資料欄位。若要查看探索的欄位，請選取頁面右上方附近的 Fields (欄位) 選單。

   > **Note**
   >
   > 只有標準記錄類別中的記錄群組才支援探查到的欄位。如需記錄類別的詳細資訊，請參閱日誌類。

4. (選用) 使用時間間隔選擇器，選取您要查詢的時間段。

   您可以選擇 5 分鐘到 30 分鐘的間隔；1 小時、3 小時和 12 小時的間隔；或是自訂的時間範圍。

5. 選擇 Run (執行) 以檢視結果。

   在本教學中，結果包括 20 筆最近新增的日誌事件。

   CloudWatch 記錄檔會顯示一段時間內記錄群組中記錄事件的長條圖。長條圖不僅會顯示表格中的事件，還會顯示日誌群組中符合您的查詢和時間範圍的事件分佈。

6. 若要查看傳回日誌事件的所有欄位，請選擇編號事件左側的三角形下拉圖示。

修改範例查詢

在此教學課程中，您將修改範例查詢來顯示 50 個最新的日誌事件。

如果您尚未執行上一個教學課程，請現在這樣做。此教學課程會從上一個教學課程的結尾處開始。

> **Note**
>
> CloudWatch 日誌見解提供的一些示例查詢使用`head`或`tail`命令而不是`limit`。這些命令已被取代，並換成 `limit`。在您編寫的所有查詢中使用 `limit`，而不是 `head` 或 `tail`。

若要修改 CloudWatch 日誌見解範例查詢

1. 在查詢編輯器中，將 20 變更為 50，然後選擇 Run (執行)。

   新查詢的結果隨即出現。假設日誌群組中有足夠的資料在預設時間範圍內，則現在會列出 50 個日誌事件。

2. (選用) 您可以儲存已建立的查詢。若要儲存此查詢，請選擇 Save (儲存)。如需詳細資訊，請參閱[儲存並重新執行 CloudWatch 日誌見解查詢](#)。

將篩選條件命令新增到範例查詢

本教學課程說明如何在查詢編輯器對查詢進行更強大的變更。在此教學課程中，您將根據已擷取的日誌事件中的欄位，以篩選前一個查詢的結果。

如果您尚未執行先前的教學課程，請現在這樣做。此教學課程會從上一個教學課程的結尾處開始。

將篩選條件命令新增到前一個查詢

1. 決定要篩選的欄位。若要查看過去 15 分鐘內， CloudWatch Logs 在所選記錄群組中所包含的記錄事件中偵測到的最常見欄位，以及每個欄位出現的記錄事件百分比，請選取頁面右側的 [欄位]。

   若要查看特定日誌事件中包含的欄位，請選擇該列左側的圖示。

   awsRegion 欄位可能出現在您的日誌事件中，這取決於日誌中有哪些事件。在本教學剩下的部分，我們將使用 awsRegion 作為篩選條件欄位，但如果沒有該欄位，您可以使用不同的欄位。
2. 在查詢編輯器方塊中，將游標移到 50 後面，然後按 Enter。
3. 在新的一行上，首先輸入 | (垂直線字元) 和空格。 CloudWatch 日誌見解查詢中的命令必須以管道字元分隔。
4. 輸入 **`filter awsRegion="us-east-1"`**。
5. 選擇執行。

   查詢會再次執行，現在會顯示符合新篩選條件的 50 個最新結果。

   如果您篩選不同的欄位，且得到錯誤結果，則可能需要逸出欄位名稱。如果欄位名稱包含非英數字元，您必須在欄位名稱前後加上反引號字元 (`) (例如，**`` `error-code`="102" ``**)。

   您必須將反引號字元用於包含非英數字元的欄位名稱 ，而不是用於值。值一律包含在引號 (") 中。

CloudWatch Logs Insights 包含強大的查詢功能，包括數個命令和對規則運算式、數學和統計運算的支援。如需詳細資訊，請參閱 CloudWatch 日誌見解查詢語法。

教學課程：使用彙總函數執行查詢

您可以在 `stats` 命令中使用彙總函式，也可以作為其他函式的引數。在本教學中，您會了解如何執行查詢命令，以計算包含指定欄位的日誌事件數量。查詢命令會回傳按照指定欄位的一個或多個值分組的總數。如需彙總函數的詳細資訊，請參閱 Amazon CloudWatch Logs 使用者指南中支援的操作和函數。

使用彙總函式執行查詢

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇 Logs (日誌)，然後選擇 Logs Insights (日誌洞察)。
3. 在 Select log group(s) (選取日誌群組) 下拉式選單中，選擇一個或多個要查詢的日誌群組。

如果這是 CloudWatch 跨帳戶觀察性的監視帳戶，您可以在來源帳戶和監視帳戶中選取記錄群組。單一查詢可以一次查詢來自不同帳戶的日誌。

您可以依日誌群組名稱、帳戶 ID 或帳戶標籤來篩選日誌群組。

當您選取記錄群組時，如果 CloudWatch 記錄群組是標準類別記錄群組，則 Logs Insights 會自動偵測記錄群組中的資料欄位。若要查看探索的欄位，請選取頁面右上方附近的 Fields (欄位) 選單。

4. 刪除查詢編輯器中的預設查詢，然後輸入下列命令：

```
stats count(*) by fieldName
```

5. 將 *fieldName* 替換為 Fields (欄位) 選單中探索到的選單欄位。

[欄位] 功能表位於頁面右上方，會顯示 CloudWatch 記錄檔群組中偵測到的所有探索到的欄位。

6. 選擇 Run (執行) 以檢視查詢結果。

查詢結果會顯示日誌群組中與查詢命令相符的記錄筆數，以及按照指定欄位的一個或多個值分組的總數。

教學課程：執行查詢以產生依日誌欄位分組的視覺效果

當您執行的查詢使用 `stats` 函數，依日誌項目中一或多個欄位的值來分組傳回的結果時，您可以透過長條圖、圓餅圖、折線圖或堆疊區域圖來檢視結果。這可協助您更有效率地將日誌中的趨勢視覺化。

執行查詢來產生視覺效果

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇 Logs (日誌)，然後選擇 Logs Insights (日誌洞察)。
3. 在 Select log group(s) (選取日誌群組) 下拉式選單中，選擇一個或多個要查詢的日誌群組。

如果這是 CloudWatch 跨帳戶觀察性的監視帳戶，您可以在來源帳戶和監視帳戶中選取記錄群組。單一查詢可以一次查詢來自不同帳戶的日誌。

您可以依日誌群組名稱、帳戶 ID 或帳戶標籤來篩選日誌群組。

4. 在查詢編輯器中，刪除目前的內容，然後輸入以下 `stats` 函式，並選擇 Run query (執行查詢)。

```
stats count(*) by @logStream
```

```
    | limit 100
```

結果會顯示每個記錄串流的日誌群組中的日誌事件數量。結果限制為 100 個資料列。

5. 選擇 Visualization (視覺化) 標籤。
6. 選取 Line (行) 旁邊的箭頭，然後選擇 Bar (列)。

此時將會顯示長條圖，顯示日誌群組中每個日誌串流的長條圖。

教學課程：執行查詢來產生時間序列視覺化

當您執行的查詢使用 `bin()` 函數，依時段來分組傳回的結果時，您可以透過折線圖、堆疊區域圖、圓餅圖或長條圖來檢視結果。這可協助您更有效率地視覺化日誌事件在一段時間內的趨勢。

執行查詢來產生視覺效果

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇 Logs (日誌)，然後選擇 Logs Insights (日誌洞察)。
3. 在 Select log group(s) (選取日誌群組) 下拉式選單中，選擇一個或多個要查詢的日誌群組。

如果這是 CloudWatch 跨帳戶觀察性的監視帳戶，您可以在來源帳戶和監視帳戶中選取記錄群組。單一查詢可以一次查詢來自不同帳戶的日誌。

您可以依日誌群組名稱、帳戶 ID 或帳戶標籤來篩選日誌群組。

4. 在查詢編輯器中，刪除目前的內容，然後輸入以下 `stats` 函式，並選擇 Run query (執行查詢)。

```
stats count(*) by bin(30s)
```

結果會顯示記錄群組中每 30 秒期間收到的 CloudWatch 記錄事件數目。

5. 選擇 Visualization (視覺化) 標籤。

結果會顯示為折線圖。若要切換至長條圖、圓餅圖或堆疊區域圖，請選擇圖表左上角 Line (線條) 旁邊的箭頭。

支援的日誌和探索的欄位

CloudWatch 日誌見解支援不同的記錄檔類型。對於傳送至標準類別日誌群組 Amazon CloudWatch 日誌的每個日誌，日 CloudWatch 誌洞見都會自動產生五個系統欄位：

- @message 包含原始未分析的日誌事件。這相當於中的message欄位InputLogevent。
- @timestamp 含有日誌事件 timestamp 欄位中的事件時間戳記。這相當於中的timestamp欄位InputLogevent。
- @ingestionTime包含記 CloudWatch 錄檔收到記錄事件的時間。
- @logStream 包含日誌事件新增到其中的日誌串流名稱。日誌串流透過產生日誌串流的相同程序對日誌進行分組。
- @log 是 *account-id*:*log-group-name* 形式的日誌群組識別碼。在查詢多個日誌群組時，這有助於識別特定事件所屬的日誌群組。

> **Note**
>
> 只有標準記錄類別中的記錄群組才支援欄位探索。如需記錄類別的詳細資訊，請參閱日誌類。

CloudWatch 日誌見解會在產生的欄位開頭插入 @ 符號。

對於許多記錄類型， CloudWatch 記錄檔也會自動探索記錄檔中包含的記錄檔欄位。下表顯示這些自動探索的欄位。

對於具有 Log Insights 無法自動探索之欄位的其他 CloudWatch 記錄類型，您可以使用parse命令擷取和建立擷取欄位，以便在該查詢中使用。如需詳細資訊，請參閱 CloudWatch 日誌見解查詢語法。

如果探索到的記錄檔欄位名稱以@字元開頭，則「 CloudWatch 日誌深入解析」會顯示該欄位，並在開頭@附加一個附加內容。例如，如果日誌欄位名稱是 @example.com，這個欄位名稱會顯示為 @@example.com。

日誌類型	探索的日誌欄位
Amazon VPC 流程日誌	@timestamp ，@logStream ，@message, accountId ，endTime, interfaceId ，logStatus ，startTime ，version, action, bytes, dstAddr, dstPort, packets, protocol, srcAddr, srcPort
Route 53 日誌	@timestamp ，@logStream ，@message, edgeLocation ，ednsClientSubnet ，hostZoneId ，protocol, queryName ，queryTimestamp ，queryType ，resolverIp ，responseCode ，version

日誌類型	探索的日誌欄位
Lambda 日誌	`@timestamp` ，`@logStream` ，`@message`, `@requestId` ，`@duration`, `@billedDuration` ，`@type`, `@maxMemoryUsed` ，`@memorySize` 如果 Lambda 日誌行包含 X-Ray 追蹤 ID，則也會包含以下欄位：`@xrayTraceId` 和 `@xraySegmentId` 。 CloudWatch 「日誌深入解析」會自動探索 Lambda 記錄中的記錄欄位，但僅適用於每個記錄事件中的第一個內嵌 JSON 片段。如果 Lambda 日誌事件包含多個 JSON 片段，您可以使用 **parse** 命令來剖析和擷取日誌欄位。如需詳細資訊，請參閱 JSON 日誌中的欄位。
CloudTrail 日誌 JSON 格式的日誌	如需詳細資訊，請參閱 JSON 日誌中的欄位。
其他日誌類型	`@timestamp` ，`@ingestionTime` ，`@logStream` ，`@message`, `@log`.

JSON 日誌中的欄位

使用 CloudWatch 日誌深入解析，您可以使用點標記法來代表 JSON 欄位。本節包含 JSON 事件範例和程式碼片段，示範如何使用點符號存取 JSON 欄位。

範例：JSON 事件

```
{
    "eventVersion": "1.0",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn: aws: iam: : 123456789012: user/Alice",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "accountId": "123456789012",
        "userName": "Alice"
    },
    "eventTime": "2014-03-06T21: 22: 54Z",
    "eventSource": "ec2.amazonaws.com",
    "eventName": "StartInstances",
```

```
    "awsRegion": "us-east-2",
    "sourceIPAddress": "192.0.2.255",
    "userAgent": "ec2-api-tools1.6.12.2",
    "requestParameters": {
        "instancesSet": {
            "items": [
                {
                    "instanceId": "i-abcde123"
                }
            ]
        }
    },
    "responseElements": {
        "instancesSet": {
            "items": [
                {
                    "instanceId": "i-abcde123",
                    "currentState": {
                        "code": 0,
                        "name": "pending"
                    },
                    "previousState": {
                        "code": 80,
                        "name": "stopped"
                    }
                }
            ]
        }
    }
}
```

範例 JSON 事件包含一個名為 userIdentity 的物件。userIdentity 包含名為 type 的欄位。若要使用點符號表示 type 的值，您可以使用 userIdentity.type。

範例 JSON 事件包含展平為巢狀欄位名稱和值清單的陣列。若要表示 requestParameters.instancesSet 中第一個項目 instanceId 的值，您可以使用 requestParameters.instancesSet.items.0.instanceId。放置在欄位 instanceID 前的數字 0 指的是欄位 items 的值的位置。下列範例包含一個程式碼片段，顯示如何存取 JSON 日誌事件中的巢狀 JSON 欄位。

範例：查詢

```
fields @timestamp, @message
```

```
| filter requestParameters.instancesSet.items.0.instanceId="i-abcde123"
| sort @timestamp desc
```

該程式碼片段顯示了一個查詢，該查詢使用帶有 `filter` 命令的點符號來存取巢狀 JSON 欄位 `instanceId` 的值。查詢會篩選出 `instanceId` 值等於 `"i-abcde123"` 的消息，並傳回包含指定值的所有日誌事件。

> **Note**
>
> CloudWatch 日誌深入解析最多可以從 JSON 記錄擷取 200 個記錄事件欄位。針對未擷取的額外欄位，可以使用 `parse` 命令來擷取訊息欄位中原始未剖析日誌事件的欄位。如需有關`parse`命令的詳細資訊，請參閱 Amazon CloudWatch 使用者指南中的查詢語法。

CloudWatch 日誌見解查詢語法

有了 CloudWatch 日誌深入解析，您可以使用查詢語言來查詢記錄群組。查詢語法支援不同的函式和運算，包含但不限於一般函式、算術和比較運算，以及正規表達式。

若要建立包含多個命令的查詢，請使用直立線符號字元 (|) 分隔命令。

若要建立包含註解的查詢，請使用雜湊字元 (#) 作為註解的開頭。

> **Note**
>
> CloudWatch Logs Insights 會自動探索不同記錄檔類型的欄位，並產生以 @ 字元開頭的欄位。如需有關這些欄位的詳細資訊，請參閱 Amazon CloudWatch 使用者指南中的支援日誌和探索到的欄位。

下表簡要描述每個命令。此資料表下面是每個命令的詳細說明，並附有範例。

> **Note**
>
> 標準 CloudWatch 記錄類別中的記錄群組支援所有記錄見解查詢命令。「不常存取」記錄檔類別中的記錄群組支援除`pattern`、`diff`和以外的所有查詢命令。`unmask`

display	在查詢結果中顯示一個或多個特定欄位。
fields	在查詢結果中顯示特定欄位，並支援可用於修改欄位值和建立要在查詢中使用之新欄位的函數和操作。
filter	篩選查詢以僅傳回符合一個或多個條件的日誌事件。
pattern	自動將您的日誌資料叢集化，以形成模式。模式是在記錄欄位之間重複出現的共用文字結構。 CloudWatch 日誌深入解析提供分析記錄事件中發現的模式的方法。如需詳細資訊，請參閱 模式分析。
diff	將要求期間中找到的記錄事件與先前時段相同長度的記錄事件進行比較，以便您可以尋找趨勢，並找出某些記錄事件是否為新的。
parse	從日誌欄位擷取資料，建立一個您可在查詢中處理的擷取欄位。**parse** 支援使用萬用字元的 glob 模式和規則運算式。
sort	以遞增 (asc) 或遞減 (desc) 方式顯示傳回的日誌事件。
stats	使用日誌欄位值計算彙總統計數字。
limit	指定您希望查詢傳回的日誌事件數目上限。使用 **sort** 可傳回「前 20 個」或「最近 20 個」結果。
dedup	根據您指定之欄位中的特定值移除重複的結果。
unmask	顯示因為資料保護政策而遮罩某些內容的某個日誌事件的全部內容。如需有關日誌群組中資料保護的詳細資訊，請參閱 使用遮罩功能協助保護敏感日誌資料。
其他操作和函數	CloudWatch Logs Insights 也支援許多比較、算術、日期時間、數值、字串、IP 位址，以及一般函式和作業。

以下各節提供有關 CloudWatch 日誌見解查詢命令的詳細資訊。

主題

- display
- fields

- 篩選條件
- pattern
- 差異
- parse
- sort
- 統計資料
- limit
- dedup
- unmask
- 布林值、比較、數值、日期時間和其他函數
- 包含特殊字元的欄位
- 在查詢中使用別名和註解

display

使用 display 在查詢結果中顯示一個或多個特定欄位。

display 命令僅顯示您指定的欄位。如果您的查詢包含多個 display 命令，查詢結果僅會顯示您在最終 display 命令中指定的欄位。

範例：顯示一個欄位

程式碼片段會顯示一個查詢範例，其使用剖析命令從 @message 中擷取資料，建立擷取欄位 loggingType 和 loggingMessage。查詢會傳回 loggingType 的值為 ERROR 的所有日誌事件。display 僅在查詢結果中顯示 loggingMessage 的值。

```
fields @message
| parse @message "[*] *" as loggingType, loggingMessage
| filter loggingType = "ERROR"
| display loggingMessage
```

> **Tip**
>
> 在查詢中僅使用一次 display。如果您在查詢中多次使用 display，則查詢結果只會顯示您最後一次使用 display 命令時指定的欄位。

fields

使用 fields 在查詢結果中顯示特定欄位。

如果您的查詢包含多個 fields 命令且未包含 display 命令，則結果會顯示在 fields 命令中指定的所有欄位。

範例：顯示特定欄位

下列範例顯示一個查詢，它傳回 20 個日誌事件並按降序顯示它們。會在查詢結果中顯示 @timestamp 和 @message 的值。

```
fields @timestamp, @message
| sort @timestamp desc
| limit 20
```

當您想要使用 fields 支援的不同函數和操作來修改欄位值並建立可在查詢中使用的新欄位時，使用 fields 而非 display。

您可以搭配使用 fields 命令和關鍵字 as，在日誌事件中建立使用欄位和函數的擷取欄位。例如：fields ispresent as isRes 會建立一個名為 isRes 的擷取欄位，而擷取欄位可在其餘查詢中使用。

篩選條件

使用 filter 來取得與一個或多個條件相符的日誌事件。

範例：使用一個條件篩選日誌事件

程式碼片段會顯示一個查詢範例，其會傳回 range 的值大於 3000 的所有日誌事件。該查詢將結果限制為 20 筆日誌事件，並按照 @timestamp 依遞減順序對日誌事件進行排序。

```
fields @timestamp, @message
| filter (range>3000)
| sort @timestamp desc
| limit 20
```

範例：使用多個條件篩選日誌事件

您可以使用關鍵字 and 和 or 以結合多個條件。

程式碼片段會顯示一個查詢範例，其會傳回 range 的值大於 3000 且 accountId 的值等於 123456789012 的日誌事件。該查詢將結果限制為 20 筆日誌事件，並按照 @timestamp 依遞減順序對日誌事件進行排序。

```
fields @timestamp, @message
| filter (range>3000 and accountId=123456789012)
| sort @timestamp desc
| limit 20
```

filter 命令中的比對和規則表達式

篩選命令支援使用規則表達式。您可以使用下列比較運算子 (=、!=、<、<=、>、>=) 和布林值運算子 (and、or 以及 not)。

您可以使用關鍵字 in 來測試設定的成員資格並檢查陣列中的元素。若要檢查陣列中的元素，將陣列放在 in 之後。您可以搭配 in 使用布林運算子 not。您可以建立查詢來使用 in 傳回欄位為字串相符的日誌事件。欄位必須是完整的字串。例如，下列程式碼片段會顯示查詢使用 in 來傳回欄位 logGroup 是完整的字串 example_group 的日誌事件。

```
fields @timestamp, @message
| filter logGroup in ["example_group"]
```

您可以使用關鍵字詞 like 和 not like 來比對子字串。您可以使用規則表達式運算子 =~ 來比對子字串。若要比對帶有 like 和 not like 的子字串，請將要比對的子字串放在單引號或雙引號中。您可以搭配 like 和 not like 使用規則表達式模式。若要使用規則表達式運算子來比對子字串，請以斜線括住想要比對的子字串。下列範例包含程式碼片段，示範如何使用 filter 命令來比對子字串。

範例：比對子字串

以下範例會傳回 f1 含有單字 Exception 的日誌事件。所有三個範例都會區分大小寫。

第一個範例比對帶有 like 的子字符。

```
fields f1, f2, f3
| filter f1 like "Exception"
```

第二個範例比對帶有 like 和規則表達式模式的子字串。

```
fields f1, f2, f3
| filter f1 like /Exception/
```

第三個範例會比對子字串與規則表達式。

```
fields f1, f2, f3
| filter f1 =~ /Exception/
```

範例：比對子字串與萬用字元

您可以使用句點符號 (.) 作為規則表達式中的萬用字元來比對子字串。在下列範例中，查詢會傳回與以字串 ServiceLog 開始的 f1 的值相符項目。

```
fields f1, f2, f3
| filter f1 like /ServiceLog./
```

您可以在句點符號 (.*) 後面放置一個星號符號，來建立窮盡數量詞，窮盡數量詞會傳回儘可能多的相符項目。例如，以下查詢會傳回與以字串 ServiceLog 開始而且還包含字串 ServiceLog 的 f1 的值相符項目。

```
fields f1, f2, f3
| filter f1 like /ServiceLog.*/
```

可能的相符項目格式如下所示：

- ServiceLogSampleApiLogGroup
- SampleApiLogGroupServiceLog

範例：從相符項目中排除子字串

以下範例會顯示會傳回日誌事件的查詢，傳回的日誌事件中 f1 不會含有單字 Exception。這個範例區分大小寫。

```
fields f1, f2, f3
| filter f1 not like "Exception"
```

範例：比對區分大小寫的子字串

您可以比對帶有 like 和規則表達式且區分大小寫的子字串。請將下列參數 (?i) 放置在想要比對的子字串之前。下列範例會顯示會傳回日誌事件的查詢，傳回的日誌事件中 f1 會含有單字 Exception 或 exception。

```
fields f1, f2, f3
```

```
| filter f1 like /(?i)Exception/
```

pattern

使用 pattern 自動將您的日誌資料叢集化，以形成模式。

模式是指日誌欄位之間反覆出現的共同文字結構。您可以使用pattern來顯示新興趨勢、監控已知錯誤，以及識別經常發生或高成本的記錄行。 CloudWatch Logs Insights 也提供主控台體驗，可讓您用來尋找和進一步分析記錄事件中的模式。如需詳細資訊，請參閱 模式分析。

由於該pattern命令會自動識別常見模式，因此您可以將其用作搜尋和分析記錄檔的起點。您也可以將 pattern 與 filter、 parse 或 sort 命令搭配使用，在更多經過微調的查詢中識別模式。

模式命令輸入

pattern 命令需有以下任何一項輸入：@message 欄位、以 parse 命令建立的擷取欄位，或使用一或多個字串函數操控的字串。

模式命令輸出

pattern 命令會產生以下輸出：

- @pattern：日誌事件欄位之間反覆出現的共同文字結構。模式中不同的欄位 (例如請求 ID 或時間戳記) 會以 <*> 表示。例如，[INFO] Request time: <*> ms 是日誌訊息 [INFO] Request time: 327 ms 可能的輸出。
- @ratio：所選期間和指定日誌群組中，符合已識別模式的日誌事件比例。例如，如果選取的日誌群組和期間中有一半的日誌事件符合模式，@ratio 就會傳回 0.50
- @sampleCount：所選期間和指定日誌群組中，符合已識別模式的日誌事件數量。
- @severityLabel：日誌嚴重性或層級，指明日誌中的資訊類型，例如 Error、Warning、Info 或 Debug。

範例

以下命令會識別所選時間範圍內指定日誌群組中具有類似結構的日誌，並依模式和數量將其分組

```
pattern @message
```

pattern 命令可以與 filter 命令搭配使用

```
filter @message like /ERROR/
```

```
| pattern @message
```

pattern 命令可與 parse 和 sort 命令搭配使用

```
filter @message like /ERROR/
| parse @message 'Failed to do: *' as cause
| pattern cause
| sort @sampleCount asc
```

差異

將要求期間內找到的記錄事件與先前時段相同長度的記錄事件進行比較。如此一來，您就可以尋找趨勢，並找出特定的記錄事件是否為新的。

將修飾詞新增至指diff令，以指定您要與之比較的期間：

- diff將目前選取時間範圍內的記錄事件與緊接之前時間範圍的記錄事件進行比較。
- diff previousDay將目前選取時間範圍內的記錄事件與前一天同一時間的記錄事件進行比較。
- diff previousWeek將目前選取時間範圍內的記錄事件與前一週同一時間的記錄事件進行比較。
- diff previousMonth將目前選取時間範圍內的記錄事件與前一個月相同時間的記錄事件進行比較。

如需詳細資訊，請參閱 與之前的時間範圍進行比較（差異）。

parse

使用 parse 從日誌欄位擷取資料，並建立一個您可在查詢中處理的擷取欄位。**parse** 支援使用萬用字元的 glob 模式和規則運算式。如需有關規則運算式語法的資訊，請參閱支援的規則運算式 (regex) 語法。

您可以使用規則表達式剖析巢狀 JSON 欄位。

範例：剖析巢狀 JSON 欄位

程式碼片段會示範如何剖析在擷取期間已扁平化的 JSON 日誌事件。

```
{'fieldsA': 'logs', 'fieldsB': [{'fA': 'a1'}, {'fA': 'a2'}]}
```

程式碼片段會顯示一個具有規則運算式的查詢，其會擷取 fieldsA 和 fieldsB 的值，以建立擷取欄位 fld 和 array。

```
parse @message "'fieldsA': '*', 'fieldsB': ['*']" as fld, array
```

具名擷取群組

當您將 **parse** 與正規表達式搭配使用時，您可以使用具名擷取群組將模式擷取到欄位中。語法是 `parse @message (?<Name>pattern).`。

以下範例在 VPC 流量日誌上使用擷取群組，將 ENI 擷取到名為 NetworkInterface 的欄位中。

```
parse @message /(?<NetworkInterface>eni-.*?) / display @timestamp, NetworkInterface
```

> **Note**
>
> JSON 日誌事件會在擷取期間扁平化。目前，不支援使用 glob 運算式剖析巢狀 JSON 欄位。您只能剖析包含不超過 200 個日誌事件欄位的 JSON 日誌事件。剖析巢狀 JSON 欄位時，您必須格式化查詢中的規則表達式，以符合 JSON 日誌事件的格式。

剖析命令的範例

使用 glob 運算式，從日誌欄位 **@message** 中擷取欄位 **@user**、**@method** 和 **@latency**，並傳回 **@method** 和 **@user** 各種不重複組合的平均延遲。

```
parse @message "user=*, method:*, latency := *" as @user,
    @method, @latency | stats avg(@latency) by @method,
    @user
```

使用規則運算式，從日誌欄位 **@message** 中擷取欄位 **@user2**、**@method2** 和 **@latency2**，並傳回 **@method2** 和 **@user2** 各種不重複組合的平均延遲。

```
parse @message /user=(?<user2>.*?), method:(?<method2>.*?),
    latency := (?<latency2>.*?)/ | stats avg(latency2) by @method2,
    @user2
```

擷取欄位 **loggingTime**、**loggingType** 和 **loggingMessage**，並篩選包含 **ERROR** 或 **INFO** 字串的日誌事件，然後針對包含 **ERROR** 字串的事件，僅顯示 **loggingMessage** 和 **loggingType** 欄位。

```
FIELDS @message
    | PARSE @message "* [*] *" as loggingTime, loggingType, loggingMessage
    | FILTER loggingType IN ["ERROR", "INFO"]
```

```
    | DISPLAY loggingMessage, loggingType = "ERROR" as isError
```

sort

使用 sort 按指定欄位以遞增 (asc) 或遞減 (desc) 順序顯示日誌事件。您可以搭配使用 limit 命令來建立「前 N 個」或「底部 N 個」查詢。

排序算法是自然排序的更新版本。如果您以遞增順序排序，則會使用下列邏輯。

- 所有非數字值都在所有數值之前。數值是僅包含數字的值，而不是數字和其他字元的混合。
- 對於非數字值，演算法會將連續的數字字元和連續的字母字元分組成不同的區塊以進行比較。它按 Unicode 值對非數字部分進行排序，並首先按其長度排序數字部分，然後按其數值排序。

如需有關 Unicode 順序的詳細資訊，請參閱 Unicode 字元清單。

例如，下列是以遞增順序排序的結果。

```
!: >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> sorted by unicode order
#
*%04
0# >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> Alphanumeric starting with numbers
5A
111A   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  Starts with more digits than 5A, so it sorted
 to be later than 5A
2345_
@ >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> 2345 is compared with @ in the unicode order,
@_
A >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> Values starting with letters
A9876fghj
a12345hfh
0 >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> Number values
01
1
2
3
```

如果您以遞減順序排序，則排序結果會相反。

例如，下列 Amazon VPC 流程日誌的查詢會找出跨主機的前 15 個封包傳輸。

```
stats sum(packets) as packetsTransferred by srcAddr, dstAddr
```

```
    | sort packetsTransferred  desc
    | limit 15
```

統計資料

使用 stats 建立日誌資料的視覺化效果，例如長條圖、折線圖和堆疊區域圖。這可協助您更有效率地識別記錄資料中的模式。 CloudWatch Logs Insights 會針對使用stats函數和一或多個彙總函數的查詢產生視覺效果。

例如，Route 53 日誌群組中的下列查詢會傳回視覺化效果，依查詢類型顯示每小時 Route 53 記錄的分佈情況。

```
stats count(*) by queryType, bin(1h)
```

所有這些查詢都可以產生長條圖。如果您的查詢使用 bin() 函式將資料以一個欄位與一段時間進行群組，那麼您也可以看到折線圖和堆疊區域圖。

bin 函數支援以下時間單位和縮寫。對於包含多個字元的所有單位和縮寫，支援加上 s 來表示複數。所以 hr 和 hrs 皆可用來指定時數。

- millisecond ms msec
- second s sec
- minute m min
- hour h hr
- day d
- week w
- month mo mon
- quarter q qtr
- year y yr

主題

視覺化呈現時間序列資料

時間序列視覺化適用於具有下列特性的查詢：

- 查詢包含一或多個彙總函數。如需詳細資訊，請參閱 Aggregation Functions in the Stats Command。
- 查詢使用 bin() 函數依一個欄位來分組資料。

這些查詢可以產生折線圖、堆疊區域圖、長條圖和圓餅圖。

範例

如需完整的教學，請參閱the section called “教學課程：執行查詢來產生時間序列視覺化”。

以下是更多適用於時間序列視覺化的查詢範例。

以下查詢為 myfield1 欄位的平均值產生視覺效果，其中每 5 分鐘建立一個資料點。每個資料點是日誌中前五分鐘的 myfield1 值的平均值彙總。

```
stats avg(myfield1) by bin(5m)
```

以下查詢根據不同欄位建立三個值的視覺效果，其中每 5 分鐘建立一個資料點。產生此覺化是因為查詢包含彙總函數，且使用 bin() 做為分組欄位。

```
stats avg(myfield1), min(myfield2), max(myfield3) by bin(5m)
```

折線圖和堆疊區域圖限制

彙總記錄項目資訊但不使用 bin() 函數的查詢可產生長條圖。不過，該查詢無法產生折線圖或堆疊區域圖。如需這些查詢類型的詳細資訊，請參閱 the section called “視覺化呈現依欄位分組的日誌資料”。

視覺化呈現依欄位分組的日誌資料

您可以為使用 stats 函數和一或多個彙總函數的查詢產生長條圖。如需詳細資訊，請參閱 Aggregation Functions in the Stats Command。

若要查看視覺化，請執行查詢。查詢 Visualization (視覺化) 標籤，選取 Line (線條) 旁邊的箭頭，然後選擇 Bar (長條)。長條圖中的視覺化限制為最多 100 個長條。

範例

如需完整的教學，請參閱the section called “教學課程：執行查詢以產生依日誌欄位分組的視覺效果”。以下段落包含更多可依據欄位進行視覺化的查詢範例。

下列 VPC 流程日誌查詢會尋找每個目的地位址、每個工作階段傳輸的平均位元組數。

```
stats avg(bytes) by dstAddr
```

您也可以產生一個圖表，其中包含每個結果值的多個長條。例如，下列 VPC 流程日誌查詢會尋找每個目的地位址、每個工作階段傳輸的平均和最大位元組數。

```
stats avg(bytes), max(bytes) by dstAddr
```

下列查詢會尋找每個查詢類型的 Amazon Route 53 查詢日誌數量。

```
stats count(*) by queryType
```

在單一查詢中使用多個統計資訊命令

您可以在單一查詢中使用多達兩個 `stats` 命令。這讓能您在第一個彙總的輸出上執行額外的彙總。

範例：使用兩個 **`stats`** 命令進行查詢

例如，以下查詢會先找出 5 分鐘區間的總流量，然後計算這些 5 分鐘區間的最高、最低和平均流量。

```
FIELDS strlen(@message) AS message_length
| STATS sum(message_length)/1024/1024 as logs_mb BY bin(5m)
| STATS max(logs_mb) AS peak_ingest_mb,
        min(logs_mb) AS min_ingest_mb,
        avg(logs_mb) AS avg_ingest_mb
```

範例：將多個統計資料命令與其他函數 (例如 **`filter`**、**`fields`**、**`bin`**) 相結合

您可以在單一命令中，將兩個 `stats` 命令與其他命令 (例如 `filter` 和 `fields`) 相結合。例如，以下查詢會尋找工作階段中不同 IP 地址的數目，並依用戶端平台尋找工作階段數目，篩選這些 IP 地址，最後再找出每個用戶端平台的工作階段請求的平均數。

```
STATS count_distinct(client_ip) AS session_ips,
      count(*) AS requests BY session_id, client_platform
| FILTER session_ips > 1
| STATS count(*) AS multiple_ip_sessions,
        sum(requests) / count(*) AS avg_session_requests BY client_platform
```

您可以在具有多個 stats 命令的查詢中使用 bin 和 dateceil 函數。例如，以下查詢會先將訊息合併成 5 分鐘的區塊，然後將這些 5 分鐘的區塊彙總為 10 分鐘的區塊，並計算每個 10 分鐘區塊內的最高、最低和平均流量。

```
FIELDS strlen(@message) AS message_length
| STATS sum(message_length) / 1024 / 1024 AS logs_mb BY BIN(5m) as @t
| STATS max(logs_mb) AS peak_ingest_mb,
        min(logs_mb) AS min_ingest_mb,
        avg(logs_mb) AS avg_ingest_mb BY dateceil(@t, 10m)
```

備註與限制

查詢最多可以有兩個 stats 命令。此配額無法變更。

如果您使用一個 sort 或 limit 命令，則其必須出現在第二個 stats 命令之後。如果在第二個 stats 命令之前，查詢無效。

當查詢有兩個 stats 命令時，在第一個 stats 彙總完成之前，不會開始顯示查詢的部分結果。

在單一查詢的第二個 stats 命令中，您只能參照第一個 stats 命令中定義的欄位。例如，以下查詢無效，因為 @message 欄位要在第一次 stats 彙總之後才可以使用。

```
FIELDS @message
| STATS SUM(Fault) by Operation
# You can only reference `SUM(Fault)` or Operation at this point
| STATS MAX(strlen(@message)) AS MaxMessageSize # Invalid reference to @message
```

您在第一個 stats 命令之後參照的任何欄位，都必須在該第一個 stats 命令中定義。

```
STATS sum(x) as sum_x by y, z
| STATS max(sum_x) as max_x by z
# You can only reference `max(sum_x)`, max_x or z at this point
```

Important

bin 函數始終以隱含的方式使用 @timestamp 欄位。這表示如果不使用第一個 stats 命令傳播 timestamp 欄位，就無法在第二個 stats 命令中使用 bin。例如，以下查詢無效。

```
FIELDS strlen(@message) AS message_length
 | STATS sum(message_length) AS ingested_bytes BY @logStream
 | STATS avg(ingested_bytes) BY bin(5m) # Invalid reference to @timestamp field
```

因此，應在第一個 stats 命令中定義 @timestamp 欄位，然後就可以在第二個 stats 命令中用來與 dateceil 搭配使用，如以下範例所示。

```
FIELDS strlen(@message) AS message_length
 | STATS sum(message_length) AS ingested_bytes, max(@timestamp) as @t BY
 @logStream
 | STATS avg(ingested_bytes) BY dateceil(@t, 5m)
```

與統計資料搭配使用的函數

CloudWatch 日誌洞見支援統計資料彙總函式和統計資料非彙總函式。

在 stats 命令中使用統計資料彙總函數，並用作其他函數的引數。

函式	結果類型	描述
avg(fieldName: NumericLogField)	數字	所指定欄位中的值的平均數。
count() count(fieldName: LogField)	number	計算日誌事件數。count() (或 count(*)) 計算查詢傳回的所有事件數，count(fieldName) 計算包含指定欄位名稱的所有記錄數。
count_distinct(fieldName: LogField)	number	傳回欄位的唯一值數目。如果欄位有極高的基數 (包含許多唯一值)，則 count_distinct 傳回的值只是近似值。
max(fieldName: LogField)	LogFieldValue	在所查詢的日誌中此日誌欄位的值上限。
min(fieldName: LogField)	LogFieldValue	在所查詢的日誌中此日誌欄位的值下限。
pct(fieldName: LogFieldValue, percent: number)	LogFieldValue	百分位數會指出資料集中相關準備好的值。例如，pct(@duration, 95) 傳回

函式	結果類型	描述
		@duration 值，其中 @duration 的值有 95% 低於這個值，有 5% 高於這個值。
stddev(fieldName: NumericLogField)	number	所指定欄位中的值的標準差。
sum(fieldName: NumericLogField)	number	所指定欄位中的值的總和。

Stats 非彙總函數

非彙總函數可用於 stats 命令，也可以作為其他函數的引數使用。

函式	結果類型	描述
earliest(fieldName: LogField)	LogField	從在查詢日誌中具有最早時間戳記的日誌事件中傳回 fieldName 的值。
latest(fieldName: LogField)	LogField	從在查詢日誌中具有最晚時間戳記的日誌事件中傳回 fieldName 的值。
sortsFirst(fieldName: LogField)	LogField	傳回在查詢日誌中最先排序的 fieldName 值。
sortsLast(fieldName: LogField)	LogField	傳回在查詢日誌中最後排序的 fieldName 值。

limit

使用 limit 指定您希望查詢傳回的日誌事件數目。

例如，下列範例僅傳回 25 個最新的日誌事件。

```
fields @timestamp, @message | sort @timestamp desc | limit 25
```

dedup

使用 dedup 根據指定欄位中的特定值移除重複的結果。可以將 dedup 與一個或多個欄位搭配使用。如果對 dedup 指定一個欄位，則只會針對該欄位的每個唯一值傳回一個日誌事件。如果指定多個欄位，則會針對這些欄位的每個唯一值組合傳回一個日誌事件。

系統會根據排序順序捨棄重複項目，只會保留排序順序中的第一個結果。建議您先對結果進行排序，然後再透過 dedup 命令進行排序。如果在透過 dedup 執行之前未對結果進行排序，則會採用使用 `@timestamp` 的預設遞減排序順序。

Null 值不會被視為評估的重複項目。系統會保留任何指定欄位之具有 Null 值的日誌事件。要消除具有 null 值的字段，請採用使用 `isPresent(field)` 函數的 **`filter`**。

可以在 dedup 命令之後的查詢中使用的唯一查詢命令為 `limit`。

範例：僅查看名為 **`server`** 之欄位的每個唯一值的最近日誌事件

下列範例顯示 `server` 的每個唯一值的最近事件的 `timestamp`、`server`、`severity` 和 `message` 欄位。

```
fields @timestamp, server, severity, message
| sort @timestamp desc
| dedup server
```

如需 CloudWatch 日誌見解查詢的更多範例，請參閱一般查詢。

unmask

使用 unmask 可以顯示因為資料保護政策而遮罩某些內容的某個日誌事件的全部內容。若要使用此命令，您必須擁有 `logs:Unmask` 許可。

如需有關日誌群組中資料保護的詳細資訊，請參閱 使用遮罩功能協助保護敏感日誌資料。

布林值、比較、數值、日期時間和其他函數

CloudWatch 日誌見解支援查詢中的許多其他操作和功能，如以下各節所述。

主題

- 算術運算子
- 布林值運算子
- 比較運算子

- 數值運算子
- 日期時間函數
- 一般函數
- IP 地址字串函數
- 字串函數

算術運算子

算術運算子可接受以數值資料類型作為引數，而且會傳回數值結果。算術運算子可用於 `filter` 和 `fields` 命令，也可以作為其他函數的引數使用。

作業	描述
`a + b`	加法
`a - b`	減法
`a * b`	乘法
`a / b`	除法
`a ^ b`	指數 (`2 ^ 3` 傳回 8)
`a % b`	餘數或模數 (`10 % 3` 傳回 1)

布林值運算子

使用布林值運算子 **`and`**、**`or`** 和 **`not`**。

> **Note**
>
> 布林值運算子僅限用於會傳回 TRUE 或 FALSE 的函數。

比較運算子

比較運算子可接受以所有資料類型作為引數，而且會傳回布林值結果。比較運算子可用於 `filter` 命令，也可以作為其他函數的引數使用。

運算子	描述
=	等於
!=	不等於
<	小於
>	大於
<=	小於或等於
>=	大於或等於

數值運算子

數值運算接受數值資料類型作為引數，並傳回數值結果。數值運算可用於 `filter` 和 `fields` 命令，也可以作為其他函數的引數使用。

作業	結果類型	描述
`abs(a: number)`	數字	絕對值
`ceil(a: number)`	number	無條件進位到上限 (大於 a 值的最小整數)
`floor(a: number)`	number	無條件捨去到下限 (小於 a 值的最大整數)
`greatest(a: number, ...numbers: number[])`	number	傳回最大值
`least(a: number, ...numbers: number[])`	number	傳回最小值
`log(a: number)`	number	自然對數

作業	結果類型	描述
`sqrt(a: number)`	number	平方根

日期時間函數

日期時間函數

日期時間函數可用於 `fields` 和 `filter` 命令，也可以作為其他函數的引數使用。如果查詢中使用了彙總函數，您可以使用這些函數來建立時段。使用由數字和下列其中一項組成的時間週期：

- `ms`毫秒
- `s`幾秒鐘
- `m`幾分鐘
- `h`數小時

例如，`10m` 是 10 分鐘，`1h` 是 1 小時。

> **Note**
>
> 為您的日期時間函數使用最合適的時間單位。 CloudWatch 記錄會根據您選擇的時間單位來限制您的要求。例如，它上限為 60 作為使用的任何請求的最大值`s`。因此，如果您指定`bin(300s)`， CloudWatch Logs 實際上將其實現為 60 秒，因為 60 是一分鐘內的秒數，因此 CloudWatch 日誌不會使用大於 60 的數字`s`。若要建立 5 分鐘的值區，請`bin(5m)`改用。的上限`ms`是 1000，`s`和的帽子`m`是 60，上限`h`是 24。

下表列出您可以在查詢命令中使用的不同日期時間函數。該表列出了每個函式的結果類型，並包含對每個函式的說明。

> **Tip**
>
> 建立查詢命令時，您可以使用時間間隔選擇器，來選取您要查詢的時間段。例如：您可以設定 5 分鐘到 30 分鐘的間隔；1 小時、3 小時和 12 小時的間隔；或是自訂的時間範圍。您也可以在特定日期之間設定時間段。

函式	結果類型	描述

bin(period: Period)	時間戳記	將 @timestamp 的值四捨五入到指定時間段，然後截斷。例如，bin(5m) 將 @timestamp 的值四捨五入至最接近的 5 分鐘。
		
		您可以使用此操作在查詢中將多筆日誌條目分組在一起。以下範例傳回每小時的例外情況計數。
		
		`filter @message like /Exception/`
		`  | stats count(*) as exceptionCount`
		`by bin(1h)`
		`  | sort exceptionCount desc`
		
		bin 函數支援以下時間單位和縮寫。對於包含多個字元的所有單位和縮寫，支援加上 s 來表示複數。所以 hr 和 hrs 皆可用來指定時數。
		
		• millisecond ms msec
		• second s sec
		• minute m min
		• hour h hr
		• day d
		• week w
		• month mo mon
		• quarter q qtr
		• year y yr

datefloor(timestamp: Timestamp, period: Period)	時間戳記	將時間戳記截斷為指定的期間。例如，datefloor(@timestamp, 1h) 將 @timestamp 的所有值截斷為半點小時。
dateceil(timestamp : Timestamp, period: Period)	時間戳記	將時間戳記無條件進位到指定期間，然後截斷。例如，dateceil(@timestamp, 1h) 將 @timestamp 的所有值截斷為整點小時。

函式	結果類型	描述
`fromMillis(fieldName: number)`	時間戳記	解譯輸入欄位為自 Unix epoch 以來的毫秒數，並將其轉換為時間戳記。
`toMillis(fieldName: Timestamp)`	number	將指定欄位中找到的時間戳記轉換為數字，代表自 Unix epoch 以來的毫秒數。例如：`toMillis(@timestamp)` 會將時間戳記 `2022-01-14T13:18:031.000-08:00` 轉換為 `1642195111000` 。

> **Note**
>
> 目前， CloudWatch 日誌深入解析不支援使用人類可讀的時間戳記篩選記錄。

一般函數

一般函數

一般函數可用於 `fields` 和 `filter` 命令，也可以作為其他函數的引數使用。

函式	結果類型	描述
`ispresent(fieldName: LogField)`	Boolean	如果欄位存在，傳回 `true`
`coalesce(fieldName: LogField, ...fieldNames: LogField[])`	LogField	傳回清單中的第一個非空值

IP 地址字串函數

IP 地址字串函數

IP 地址字串函數可用於 `filter` 和 `fields` 命令，也可以作為其他函數的引數使用。

函式	結果類型	描述
`isValidIp(fieldName: string)`	布林值	如果欄位是有效的 IPv4 或 IPv6 地址，則會傳回 `true`。
`isValidIpV4(fieldName: string)`	boolean	如果欄位是有效的 IPv4 地址，則傳回 `true`。
`isValidIpV6(fieldName: string)`	boolean	如果欄位是有效的 IPv6 地址，則傳回 `true`。
`isIpInSubnet(fieldName: string, subnet: string)`	boolean	如果欄位是指定 v4 或 v6 子網路內的有效 IPv4 或 IPv6 地址，則傳回 `true`。指定子網路時，請使用 CIDR 標記法，例如 `192.0.2.0/24` 或 `2001:db8::/32` ，其中 `192.0.2.0` 或 `2001:db8::` 是 CIDR 區塊的起始位置。
`isIpv4InSubnet(fieldName: string, subnet: string)`	boolean	如果欄位是指定 v4 子網路內的有效 IPv4 地址，則傳回 `true`。指定子網路時，請使用 CIDR 標記法，例如 `192.0.2.0/24` ，其中 `192.0.2.0` 是 CIDR 區塊的起始位置。
`isIpv6InSubnet(fieldName: string, subnet: string)`	boolean	如果欄位是指定 v6 子網路內的有效 IPv6 地址，則傳回 `true`。指定子網路時，請使用 CIDR 標記法，例如 `2001:db8::/32` ，其中 `2001:db8::` 是 CIDR 區塊的起始位置。

字串函數

字串函數

字串函數可用於 `fields` 和 `filter` 命令，也可以作為其他函數的引數使用。

函式	結果類型	描述
`isempty(fieldName: string)`	Number	如果欄位遺失或是空白字串，傳回 1。

函式	結果類型	描述
isblank(fieldName: string)	Number	如果欄位遺失、是空白字串或只包含空格，傳回 1。
concat(str: string, ...strings: string[])	string	串連字串。
ltrim(str: string) ltrim(str: string, trimChars: string)	string	如果函數沒有第二個引數，則會移除字串左側的空格。如果函數有第二個字串引數，則不會移除空格。而是從 str 左側移除 trimChars 中的字元。例如，ltrim("xyZxyfooxyZ","xyZ") 傳回 "fooxyZ"。
rtrim(str: string) rtrim(str: string, trimChars: string)	string	如果函數沒有第二個引數，則會移除字串右側的空格。如果函數有第二個字串引數，則不會移除空格。而是從 str 右側移除 trimChars 的字元。例如，rtrim("xyZfooxyxyZ","xyZ") 傳回 "xyZfoo"。
trim(str: string) trim(str: string, trimChars: string)	string	如果函數沒有第二個引數，則會移除字串兩側的空格。如果函數有第二個字串引數，則不會移除空格。而是從 str 兩側移除 trimChars 的字元。例如，trim("xyZxyfooxyxyZ","xyZ") 傳回 "foo"。
strlen(str: string)	number	以 Unicode 字碼指標傳回字串的長度。
toupper(str: string)	string	將字串轉換成大寫。

函式	結果類型	描述
tolower(str: string)	string	將字串轉換成小寫。
substr(str: string, startIndex: number) substr(str: string, startIndex: number, length: number)	string	傳回從數字引數指定的索引到字串結尾的子字串。如果函數有第二個數字引數，則是包含要擷取的子字串長度。例如，substr("xyZfooxyZ",3, 3) 傳回 "foo"。
replace(fieldName: string, searchValue: string, replaceValue: string)	string	以 replaceValue 取代 fieldName: string 中出現的所有 searchValue 。 例如：函式 replace(logGroup,"smoke_test","Smoke") 搜尋欄位 logGroup 中包含字串值 smoke_test 的日誌事件，並將值替換為字串 Smoke。
strcontains(str: string, searchValue: string)	number	如果 str 包含 searchValue ，則傳回 1，否則傳回 0。

包含特殊字元的欄位

如果欄位包含@符號或句號 (.) 以外的非英數字元，您必須在欄位中加上反引號字元 (`)。例如：日誌欄位 foo-bar 含有非英數字元，亦即連字號 (-)，因此必須置於反引號 (`foo-bar`) 之間。

在查詢中使用別名和註解

建立含有別名的查詢。將日誌欄位重新命名，或在擷取值並填入欄位時，都可使用別名。使用關鍵字 as 為日誌欄位或結果賦予別名。您可以在查詢中使用多個別名。您可以在下列任一命令中使用別名：

- fields
- parse
- sort

- stats

以下範例會示範如何建立含有別名的查詢。

範例

查詢的 fields 命令中含有別名。

```
fields @timestamp, @message, accountId as ID
| sort @timestamp desc
| limit 20
```

查詢會傳回欄位 @timestamp、@message 和 accountId 的值。結果以遞減方式排序，且限制為 20。accountId 的值會顯示於別名 ID 底下。

範例

查詢的 sort 和 stats 命令中含有別名。

```
stats count(*) by duration as time
| sort time desc
```

查詢會計算欄位 duration 出現於日誌群組中的次數，並以遞減方式將結果排序。duration 的值會顯示於別名 time 底下。

使用註解

CloudWatch 日誌見解支援查詢中的註解。使用雜湊字元 (#) 作為註解的開頭。您可以使用註解，忽略查詢或文件查詢中的行。

範例：查詢

以下查詢運行時，系統會忽略第二行。

```
fields @timestamp, @message, accountId
# | filter accountId not like "7983124201998"
| sort @timestamp desc
| limit 20
```

模式分析

CloudWatch 當您查詢記錄檔時，日誌深入解析會使用機器學習演算法來尋找模式。模式是在記錄欄位之間重複出現的共用文字結構。當您檢視查詢結果時，您可以選擇 [模式] 索引標籤，查看根據結果範例找到的 CloudWatch 記錄檔模式。或者，您可以將命`pattern`令附加到查詢中，以分析整個相符記錄事件集中的模式。

病毒碼對於分析大型記錄集很有用，因為通常可以將大量記錄事件壓縮成幾個病毒碼。

請考慮下列三個記錄事件的範例。

```
2023-01-01 19:00:01 [INFO] Calling DynamoDB to store for resource id 12342342k124-12345
2023-01-01 19:00:02 [INFO] Calling DynamoDB to store for resource id 324892398123-12345
2023-01-01 19:00:03 [INFO] Calling DynamoDB to store for resource id 3ff231242342-12345
```

在上一個範例中，所有三個記錄事件都遵循一個模式：

```
<*> <*> [INFO] Calling DynamoDB to store for resource id <*>
```

模式中的字段稱為令牌。模式中不同的欄位 (例如要求 ID 或時間戳記) 是動態權杖。每個動態權杖會`<*>`在 CloudWatch 記錄檔顯示時表示。

動態權杖的常見範例包括錯誤碼、時間戳記和要求 ID。令牌值表示動態令牌的特定值。例如，如果動態令牌代表 HTTP 錯誤代碼，則令牌值可能是`501`。

模式檢測也用於 CloudWatch 日誌異常檢測器和比較功能。如需詳細資訊，請參閱 記錄異常偵測 及 與之前的時間範圍進行比較（差異）。

開始使用模式分析

病毒碼偵測會自動在任何 CloudWatch 日誌深入解析查詢中執行。不包含`pattern`命令的查詢會在結果中同時取得記錄事件和模式。

如果您在查詢中包含該`pattern`命令，則會對整個相符的記錄事件集執行模式分析。這會提供更精確的模式結果，但是當您使用`pattern`指令時，不會傳回原始記錄事件。如果查詢未包含`pattern`，則模式結果會根據前 1000 個傳回的記錄事件或您在查詢中使用的限制值為基礎。如果您包含`pattern`在查詢中，則「模式」索引標籤中顯示的結果會從查詢符合的所有記錄事件衍生出來。

若要開始使用 CloudWatch 日誌深入解析中的模式分析

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/

2. 在導覽窗格中，選擇 [記錄]、[記錄深入解析]。

   在 Logs Insights (日誌洞察) 頁面上，查詢編輯器包含會傳回 20 筆最新日誌事件的預設查詢。

3. 移除查詢方塊中的這一`| limit 20`行，使查詢看起來如下所示：

   ```
   fields @timestamp, @message, @logStream, @log
   | sort @timestamp desc
   ```

4. 在 [選取記錄群組] 下拉式清單中，選擇一或多個要查詢的記錄群組。

5. (選用) 使用時間間隔選擇器，選取您要查詢的時間段。

   您可以選擇 5 分鐘和 30 分鐘的間隔；1 小時，3 小時和 12 小時的間隔；或自定義時間範圍。

6. 選擇「執行查詢」以啟動查詢。

   查詢完成執行後，[記錄] 索引標籤會顯示查詢傳回的記錄事件表格。表格上方是關於有多少條記錄匹配查詢的消息，類似於顯示 71,101 條記錄中的 1000 條匹配。

7. 選擇「樣式」標籤。

8. 資料表現在會顯示查詢中找到的模式。由於查詢不包含`pattern`命令，因此此索引標籤只會顯示在「記錄檔」索引標籤表格中顯示的 1000 個記錄事件中發現的病毒碼。

   針對每個樣式，會顯示下列資訊：

   - 模式，每個動態令牌顯示為`<*>`。
   - 事件計數，也就是模式出現在查詢的記錄事件中的次數。選擇「事件計數」欄標題，依頻率排序模式。
   - 事件比率，即包含此模式之查詢記錄事件的百分比。
   - 「嚴重性」類型，這將是下列其中一種：
     - 錯誤，如果模式包含「錯誤」一詞。
     - 如果模式包含「警告」一詞，但不包含「錯誤」，則會發出警告。
     - INFO，如果模式不包含警告或錯誤。

   選擇嚴重性資訊欄標題，依嚴重性排序模式。

9. 現在變更查詢。將查詢中的`| sort @timestamp desc`行取代為`| pattern @message`，以便完整的查詢如下所示：

   ```
   fields @timestamp, @message, @logStream, @log
   | pattern @message
   ```

10. 選擇 Run query (執行查詢)。

    查詢完成後，[記錄] 索引標籤中沒有任何結果。不過，[模式] 索引標籤可能會列出更多的模式，視查詢的記錄事件總數而定。

11. 無論您是否包含pattern在查詢中，都可以進一步檢查查詢傳回的模式。若要這麼做，請在「檢查」欄中選擇其中一個樣式的圖示。

    樣式檢查窗格隨即出現，並顯示下列內容：

    - 模式。在模式中選擇一個令牌以分析該令牌的值。
    - 顯示在查詢的時間範圍內模式出現次數的長條圖。這可以幫助您識別有趣的趨勢，例如模式發生突然增加。
    - [記錄範例] 索引標籤會顯示一些符合所選模式的記錄事件。
    - 如果您已選取動態權杖，則「記號值」標籤會顯示所選動態權杖的值。

      > **Note**
      >
      > 每個記號最多可擷取 10 個記號值。令牌計數可能不精確。 CloudWatch 日誌使用概率計數器來生成令牌計數，而不是絕對值。

    - 「相關陣列」標籤會顯示經常發生的其他陣列，與您正在檢查的陣列相同的時間。例如，如果ERROR郵件的模式通常伴隨另一個標記為其他詳細資料INFO的記錄事件，則會在此處顯示該模式。

有關模式命令的詳細信息

本節包含有關pattern命令及其用法的更多詳細資訊。

- 在上一個教程中，我們在添加時刪除了該sort命令，pattern因為如果查詢在pattern命令後包含命令，則該命sort令無效。有一個pattern之前是有效的sort。

  如需pattern語法的詳細資訊，請參閱pattern。
- 當您在查詢pattern中使用時，@message必須是在pattern指令中選取的其中一個欄位。
- 您可以在filter指令之前包括該pattern指令，以便僅將篩選的記錄事件集用作陣列分析的輸入。
- 若要查看特定欄位的模式結果，例如從parse指令衍生的欄位，請使用pattern @fieldname。
- 具有非記錄輸出的查詢 (例如使用stats命令的查詢) 不會傳回模式結果。

與之前的時間範圍進行比較（差異）

您可以使用 CloudWatch 日誌深入解析來比較一段時間內記錄事件中的變更。您可以將最近時間範圍內擷取的記錄事件與前一個時段的記錄檔進行比較。或者，您可以與類似的過去時間週期進行比較。這可協助您找出記錄檔中的錯誤是最近引入還是已經發生，並可協助您找出其他趨勢。

比較查詢只會傳回結果中的模式，而不會傳回原始記錄事件。傳回的模式可協助您快速查看一段時間內記錄事件的趨勢和變更。執行比較查詢並取得模式結果之後，您可以看到您感興趣之模式的範例原始記錄事件。如需記錄檔模式的詳細資訊，請參閱模式分析。

當您執行比較查詢時，系統會根據兩個不同的期間來分析查詢：您選取的原始查詢期間，以及比較期間。比較期間永遠與原始查詢期間的長度相同。比較的預設時間間隔如下。

- 前一個週期 — 與查詢期間之前的時段進行比較。
- 前一天 — 與查詢期間前一天的時間範圍進行比較。
- 前一週 — 與查詢期間之前一週的時間範圍進行比較。
- 上個月 — 與查詢期間之前一個月的時間範圍進行比較。

> **Note**
>
> 使用比較的查詢會產生類似於在合併時間範圍內執行單一 CloudWatch 記錄深入解析查詢的費用。如需詳細資訊，請參閱 Amazon CloudWatch 定價。

若要執行比較查詢

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇 [記錄]、[記錄深入解析]。

   預設查詢會出現在查詢方塊中。
3. 保留預設查詢或輸入不同的查詢。
4. 在 [選取記錄群組] 下拉式清單中，選擇一或多個要查詢的記錄群組。
5. (選用) 使用時間間隔選擇器，選取您要查詢的時間段。預設查詢是前一小時的記錄資料。
6. 在時間範圍選取器中，選擇比較。然後選擇您要與原始記錄檔比較的先前期間，然後選擇「套用」。
7. 選擇 Run query (執行查詢)。

若要使查詢從比較期間擷取資料，系統會將命`diff`令附加至您的查詢。

8. 選擇「模式」標籤以查看結果。

   此表格會顯示下列資訊：

   - 每個模式，與模式的可變部分由動態令牌符號替換`<*>`。如需詳細資訊，請參閱 模式分析。
   - 事件計數是在原始、較目前時段內具有該模式的記錄事件數目。
   - 差異事件計數是目前期間內相符記錄事件數目與比較期間的差異。積極的不同意味著在當前時間段內有更多此類事件。
   - 差異說明會簡要摘要彙總目前期間與比較期間之間該模式的變更。
   - 嚴重性類型是使用此模式記錄事件的可能嚴重性，根據記錄事件中的文字 (例如`FATALERROR`、和`WARN`)。

9. 若要進一步檢查清單中的其中一個樣式，請在「檢查」欄中選擇其中一個圖樣的圖示。

   樣式檢查窗格隨即出現，並顯示下列內容：

   - 模式。在模式中選擇一個令牌以分析該令牌的值。
   - 顯示在查詢的時間範圍內模式出現次數的長條圖。這可以幫助您識別有趣的趨勢，例如模式發生突然增加。
   - [記錄範例] 索引標籤會顯示一些符合所選模式的記錄事件。
   - 如果您已選取動態權杖，則「記號值」標籤會顯示所選動態權杖的值。

     > **Note**
     >
     > 每個記號最多可擷取 10 個記號值。令牌計數可能不精確。 CloudWatch 日誌使用概率計數器來生成令牌計數，而不是絕對值。

   - 「相關陣列」標籤會顯示經常發生的其他陣列，與您正在檢查的陣列相同的時間。例如，如果`ERROR`郵件的模式通常伴隨另一個標記為其他詳細資料`INFO`的記錄事件，則會在此處顯示該模式。

範例查詢

本節包含您可以在CloudWatch 控制台中運行的一般和有用的查詢命令列表。如需如何執行查詢命令的詳細資訊，請參閱 Amazon CloudWatch Logs 使用者指南中的教學課程：執行和修改範例查詢。

如需查詢語法的詳細資訊，請參閱CloudWatch 日誌見解查詢語法。

主題

一般查詢

尋找最近新增的 25 個日誌事件。

```
fields @timestamp, @message | sort @timestamp desc | limit 25
```

取得每小時的例外狀況數清單。

```
filter @message like /Exception/
    | stats count(*) as exceptionCount by bin(1h)
    | sort exceptionCount desc
```

取得非例外狀況的日誌事件清單。

```
fields @message | filter @message not like /Exception/
```

取得 **server** 欄位的每個唯一值的最近日誌事件。

```
fields @timestamp, server, severity, message
| sort @timestamp asc
| dedup server
```

取得每個 **severity** 類型的 **server** 欄位的每個唯一值的最近日誌事件。

```
fields @timestamp, server, severity, message
| sort @timestamp desc
| dedup server, severity
```

Lambda 日誌的查詢

查明過度佈建的記憶體數量。

```
filter @type = "REPORT"
    | stats max(@memorySize / 1000 / 1000) as provisonedMemoryMB,
        min(@maxMemoryUsed / 1000 / 1000) as smallestMemoryRequestMB,
        avg(@maxMemoryUsed / 1000 / 1000) as avgMemoryUsedMB,
        max(@maxMemoryUsed / 1000 / 1000) as maxMemoryUsedMB,
        provisonedMemoryMB - maxMemoryUsedMB as overProvisionedMB
```

建立延遲報告。

```
filter @type = "REPORT" |
    stats avg(@duration), max(@duration), min(@duration) by bin(5m)
```

搜尋緩慢的函數調用，並消除重試或用戶端程式碼可能產生的重複請求。在此查詢中，**@duration** 以毫秒為單位。

```
fields @timestamp, @requestId, @message, @logStream
| filter @type = "REPORT" and @duration > 1000
| sort @timestamp desc
| dedup @requestId
| limit 20
```

Amazon VPC 流程日誌的查詢

尋找主機之間的前 15 個封包傳輸：

```
stats sum(packets) as packetsTransferred by srcAddr, dstAddr
    | sort packetsTransferred  desc
    | limit 15
```

尋找特定子網路上主機的前 15 個位元組傳輸。

```
filter isIpv4InSubnet(srcAddr, "192.0.2.0/24")
    | stats sum(bytes) as bytesTransferred by dstAddr
    | sort bytesTransferred desc
    | limit 15
```

尋找使用 UDP 做為資料傳輸協定的 IP 地址。

```
filter protocol=17 | stats count(*) by srcAddr
```

尋找在擷取時段略過流程記錄的 IP 地址。

```
filter logStatus="SKIPDATA"
    | stats count(*) by bin(1h) as t
    | sort t
```

尋找每個連線的單一記錄，以協助疑難排解網路連線問題。

```
fields @timestamp, srcAddr, dstAddr, srcPort, dstPort, protocol, bytes
| filter logStream = 'vpc-flow-logs' and interfaceId = 'eni-0123456789abcdef0'
| sort @timestamp desc
| dedup srcAddr, dstAddr, srcPort, dstPort, protocol
| limit 20
```

Route 53 日誌的查詢

依查詢類型尋找每小時的記錄分佈。

```
stats count(*) by queryType, bin(1h)
```

尋找請求數最高的前 10 個 DNS 解析程式。

```
stats count(*) as numRequests by resolverIp
    | sort numRequests desc
    | limit 10
```

依網域和子網域尋找伺服器無法完成 DNS 請求的記錄數。

```
filter responseCode="SERVFAIL" | stats count(*) by queryName
```

CloudTrail 記錄檔查詢

尋找每個服務、事件類型和 AWS 區域的日誌項目數。

```
stats count(*) by eventSource, eventName, awsRegion
```

尋找在指定 AWS 區域中啟動或停止的 Amazon EC2 主機。

```
filter (eventName="StartInstances" or eventName="StopInstances") and awsRegion="us-
east-2"
```

尋找新建立 IAM 使用者的 AWS 區域、使用者名稱和 ARN。

```
filter eventName="CreateUser"
    | fields awsRegion, requestParameters.userName, responseElements.user.arn
```

尋找叫用 API **UpdateTrail** 時發生例外狀況的記錄數。

```
filter eventName="UpdateTrail" and ispresent(errorCode)
    | stats count(*) by errorCode, errorMessage
```

尋找使用 TLS 1.0 或 1.1 的日誌條目

```
filter tlsDetails.tlsVersion in [ "TLSv1", "TLSv1.1" ]
| stats count(*) as numOutdatedTlsCalls by userIdentity.accountId, recipientAccountId,
 eventSource, eventName, awsRegion, tlsDetails.tlsVersion, tlsDetails.cipherSuite,
 userAgent
| sort eventSource, eventName, awsRegion, tlsDetails.tlsVersion
```

尋找使用 TLS 1.0 或 1.1 版本之每項服務的呼叫次數

```
filter tlsDetails.tlsVersion in [ "TLSv1", "TLSv1.1" ]
| stats count(*) as numOutdatedTlsCalls by eventSource
| sort numOutdatedTlsCalls desc
```

查詢 Amazon API Gateway

找出最後 10 個 4XX 錯誤

```
fields @timestamp, status, ip, path, httpMethod
| filter status>=400 and status<=499
| sort @timestamp desc
| limit 10
```

識別 Amazon API Gateway 存取日誌群組中執行時間最長的 10 個 Amazon API Gateway 要求

```
fields @timestamp, status, ip, path, httpMethod, responseLatency
| sort responseLatency desc
| limit 10
```

傳回 Amazon API Gateway 存取記錄群組中最常用的 API 路徑清單

```
stats count(*) as requestCount by path
| sort requestCount desc
| limit 10
```

為您的 Amazon API Gateway 存取日誌群組建立整合延遲報告

```
filter status=200
| stats avg(integrationLatency), max(integrationLatency),
min(integrationLatency) by bin(1m)
```

NAT 閘道的查詢

如果您發現 AWS 帳單中的正常費用高於正常費用，則可以使用 CloudWatch 日誌深入解析來尋找最主要的貢獻者。如需有關下列查詢命令的詳細資訊，請參閱如何透過 VPC 中的 NAT 閘道尋找流量的主要貢獻者？ 在 AWS 高級支持頁面。

> **Note**
>
> 在以下查詢命令中，將 "x.x.x.x" 取代為 NAT 閘道的私有 IP，並將 "y.y" 替換為 VPC CIDR 範圍的前兩個八位元組。

查看透過 NAT 閘道傳送最多流量的執行個體。

```
filter (dstAddr like 'x.x.x.x' and srcAddr like 'y.y.')
| stats sum(bytes) as bytesTransferred by srcAddr, dstAddr
| sort bytesTransferred desc
| limit 10
```

確定進出 NAT 閘道中執行個體的流量。

```
filter (dstAddr like 'x.x.x.x' and srcAddr like 'y.y.') or (srcAddr like 'xxx.xx.xx.xx'
 and dstAddr like 'y.y.')
| stats sum(bytes) as bytesTransferred by srcAddr, dstAddr
| sort bytesTransferred desc
| limit 10
```

確定 VPC 中的執行個體在上傳和下載時，最經常與之通訊的網際網路目的地。

對於上傳

```
filter (srcAddr like 'x.x.x.x' and dstAddr not like 'y.y.')
| stats sum(bytes) as bytesTransferred by srcAddr, dstAddr
| sort bytesTransferred desc
| limit 10
```

對於下載

```
filter (dstAddr like 'x.x.x.x' and srcAddr not like 'y.y.')
| stats sum(bytes) as bytesTransferred by srcAddr, dstAddr
| sort bytesTransferred desc
| limit 10
```

Apache 伺服器日誌的查詢

您可以使用 CloudWatch 記錄深入分析來查詢 Apache 伺服器記錄檔。如需有關下列查詢的詳細資訊，請參閱 AWS 雲端作業與移轉部落格中的使用 CloudWatch 記錄深入解析簡化 Apache 伺服器記錄。

查看最相關的欄位，以在應用程式的 /admin 路徑中檢閱存取日誌並檢查流量。

```
fields @timestamp, remoteIP, request, status, filename| sort @timestamp desc
| filter filename="/var/www/html/admin"
```

```
| limit 20
```

查找以狀態碼 "200" (成功) 存取主頁面的不重複 GET 請求次數。

```
fields @timestamp, remoteIP, method, status
| filter status="200" and referrer= http://34.250.27.141/ and method= "GET"
| stats count_distinct(remoteIP) as UniqueVisits
| limit 10
```

查找 Apache 服務重新啟動的次數。

```
fields @timestamp, function, process, message
| filter message like "resuming normal operations"
| sort @timestamp desc
| limit 20
```

查詢 Amazon EventBridge

獲取按事件詳細信息類型分組的 EventBridge 事件數量

```
fields @timestamp, @message
| stats count(*) as numberOfEvents by `detail-type`
| sort numberOfEvents desc
```

剖析命令的範例

使用 glob 運算式，從日誌欄位 **@message** 中擷取欄位 **@user**、**@method** 和 **@latency**，並傳回 **@method** 和 **@user** 各種不重複組合的平均延遲。

```
parse @message "user=*, method:*, latency := *" as @user,
    @method, @latency | stats avg(@latency) by @method,
    @user
```

使用規則運算式，從日誌欄位 **@message** 中擷取欄位 **@user2**、**@method2** 和 **@latency2**，並傳回 **@method2** 和 **@user2** 各種不重複組合的平均延遲。

```
parse @message /user=(?<user2>.*?), method:(?<method2>.*?),
    latency := (?<latency2>.*?)/ | stats avg(latency2) by @method2,
    @user2
```

擷取欄位 **loggingTime**、**loggingType** 和 **loggingMessage**，並篩選包含 **ERROR** 或 **INFO** 字串的日誌事件，然後針對包含 **ERROR** 字串的事件，僅顯示 **loggingMessage** 和 **loggingType** 欄位。

```
FIELDS @message
    | PARSE @message "* [*] *" as loggingTime, loggingType, loggingMessage
    | FILTER loggingType IN ["ERROR", "INFO"]
    | DISPLAY loggingMessage, loggingType = "ERROR" as isError
```

在圖表中視覺化日誌資料

您可以使用長條圖、折線圖和堆疊區域圖等視覺效果，更有效地識別記錄資料中的模式。 CloudWatch Logs Insights 會針對使用`stats`函數和一或多個彙總函數的查詢產生視覺效果。如需詳細資訊，請參閱 stats。

儲存並重新執行 CloudWatch 日誌見解查詢

建立查詢之後，可以儲存該查詢，以便之後再次執行。查詢儲存在資料夾結構中，以便您可以組織它們。每個區域和每個帳戶最多可儲存 1000 個查詢。

若要儲存查詢，您必須登入具有許可 `logs:PutQueryDefinition` 的角色。若要查看已儲存查詢的清單，您必須登入具有許可 `logs:DescribeQueryDefinitions` 的角色。

儲存查詢

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇 Logs (日誌)，然後選擇 Logs Insights (日誌洞察)。
3. 在查詢編輯器中，建立查詢。
4. 選擇儲存。

   如果您沒有看到 [儲存] 按鈕，則需要變更為 CloudWatch 記錄主控台的新設計。若要這麼做：

   a. 在導覽窗格中，選擇 Log groups (日誌群組)。
   b. 選擇 Try the new design (嘗試新設計)。
   c. 在導覽窗格中，選擇 Insights (深入分析)，並返回此程序的步驟 3。
5. 輸入查詢的名稱。
6. (選用) 選擇您要儲存查詢的資料夾。選取 Create new (新建) 以建立資料夾。如果您建立新資料夾，您可以在資料夾名稱中使用斜線 (/) 字元，以定義資料夾結構。例如，命名新資料夾 **folder-**

level-1/folder-level-2 會建立名為 **folder-level-1** 的頂層資料夾，該資料夾中會有另一個資料夾名為 **folder-level-2**。查詢會儲存在 **folder-level-2** 中。

7. (選用) 變更查詢的日誌群組或查詢文字。
8. 選擇儲存。

> **Tip**
>
> 您可以使用 PutQueryDefinition 建立已儲存查詢的資料夾。若要為儲存的查詢建立資料夾，請使用正斜線 (/) ，在所需查詢名稱前加上想要的資料夾名稱：<*folder-name*>/<*query-name*>。若要取得有關此動作的更多資訊，請參閱PutQueryDefinition。

執行已儲存的查詢

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇 Logs (日誌)，然後選擇 Logs Insights (日誌洞察)。
3. 選擇右側的 Queries (查詢)。
4. 從 Saved queries (已儲存的查詢) 清單中選取查詢。它會出現在查詢編輯器中。
5. 選擇執行。

儲存新版本的已儲存查詢

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇 Logs (日誌)，然後選擇 Logs Insights (日誌洞察)。
3. 選擇右側的 Queries (查詢)。
4. 從 Saved queries (已儲存的查詢) 清單中選取查詢。它會出現在查詢編輯器中。
5. 修改查詢。如果您需要執行該功能以檢查您的工作，請選擇 Run query (執行查詢)。
6. 當您準備好儲存新版本，請選擇 Actions (動作)、Save as (另存新檔)。
7. 輸入查詢的名稱。
8. (選用) 選擇您要儲存查詢的資料夾。選取 Create new (新建) 以建立資料夾。如果您建立新資料夾，您可以在資料夾名稱中使用斜線 (/) 字元，以定義資料夾結構。例如，命名新資料夾 **folder-level-1/folder-level-2** 會建立名為 **folder-level-1** 的頂層資料夾，該資料夾中會有另一個資料夾名為 **folder-level-2**。查詢會儲存在 **folder-level-2** 中。

9. (選用) 變更查詢的日誌群組或查詢文字。
10. 選擇儲存。

若要刪除查詢，您必須登入具備 `logs:DeleteQueryDefinition` 許可的角色。

編輯或刪除已儲存的查詢

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇 Logs (日誌)，然後選擇 Logs Insights (日誌洞察)。
3. 選擇右側的 Queries (查詢)。
4. 從 Saved queries (已儲存的查詢) 清單中選取查詢。它會出現在查詢編輯器中。
5. 選擇 Actions (動作)、Edit (編輯) 或 Actions (動作)、Delete (刪除)。

將查詢新增到儀表板或匯出查詢結果

執行查詢後，您可以將查詢新增至 CloudWatch 儀表板，或將結果複製到剪貼簿。

每次載入儀表板和每次儀表板重新整理時，會執行新增到儀表板的查詢。這些查詢會計入 30 個並行 CloudWatch 記錄見解查詢的限制。

將查詢結果新增到儀表板

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇 Logs (日誌)，然後選擇 Logs Insights (日誌洞察)。
3. 選擇一或多個日誌群組並執行查詢。
4. 選擇 Add to dashboard (新增至儀表板)。
5. 選取儀表板，或選擇 Create new (新建)，為查詢結果建立儀表板。
6. 選取用於查詢結果的 Widget 類型。
7. 輸入 Widget 的名稱。
8. 選擇 Add to dashboard (新增至儀表板)。

將查詢結果複製到剪貼簿或下載查詢結果

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇 Logs (日誌)，然後選擇 Logs Insights (日誌洞察)。

3. 選擇一或多個日誌群組並執行查詢。
4. 選擇 Export results (匯出結果)，然後選擇您需要的選項。

檢視執行中的查詢或查詢歷史記錄

您可以檢視目前進行中的查詢，以及您的最新查詢歷史記錄。

目前執行中的查詢包括您已新增到儀表板的查詢。每個帳戶最多可同時進行 30 個 CloudWatch 日誌見解查詢，包括新增至儀表板的查詢。

檢視您的最新查詢歷史記錄

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇 Logs (日誌)，然後選擇 Logs Insights (日誌洞察)。
3. 如果您使用的是記錄主控台的新設計，請選擇「歷史 CloudWatch 記錄」。如果您是使用舊設計，請選擇 Actions (動作)、View query history for this account (檢視此帳戶的查詢歷程記錄)。

   隨即顯示您的最新查詢清單。您可以選取查詢，並選擇 Run (執行)，以再次執行任一查詢。

   在「狀態」下，「 CloudWatch 記錄」會針對目前正在執行的任何查詢顯示進行中。

使用加密查詢結果 AWS Key Management Service

根據預設， CloudWatch Logs 會使用預設的 Logs 伺服器端加密方法來加密 CloudWatch 記 CloudWatch 錄見解查詢的儲存結果。您可以選擇使用 AWS KMS 金鑰來加密這些結果。如果您將 AWS KMS 金鑰與加密結果產生關聯，則 CloudWatch Logs 會使用該金鑰來加密帳戶中所有查詢的儲存結果。

如果您稍後取消金鑰與查詢結果的關聯， CloudWatch Logs 會回復為預設加密方法供稍後查詢使用。但是在密鑰關聯時運行的查詢仍然使用該密鑰進行加密。 CloudWatch 取消關聯 KMS 金鑰之後，記錄檔仍然可以傳回這些結果，因為 CloudWatch 記錄檔仍可繼續參考金鑰。不過，如果金鑰稍後停用，則 CloudWatch 記錄檔將無法讀取使用該金鑰加密的查詢結果。

> **Important**
>
> CloudWatch 記錄僅支援對稱 KMS 金鑰。切勿使用非對稱金鑰加密查詢結果。如需詳細資訊，請參閱使用對稱和非對稱金鑰。

限制

- 若要執行下列步驟，您必須擁有下列許可：kms:CreateKey、kms:GetKeyPolicy 和 kms:PutKeyPolicy。
- 建立或取消金鑰與查詢結果的關聯後，操作會在 5 分鐘內生效。
- 如果您撤銷關聯金鑰的 CloudWatch 記錄存取權或刪除關聯的 KMS 金鑰，則無法再擷取 CloudWatch 記錄中的加密資料。
- 您無法使用 CloudWatch 控制台關聯金鑰，您必須使用 AWS CLI 或 CloudWatch 記錄 API。

步驟 1：建立 AWS KMS key

若要建立 KMS 金鑰，請使用以下 create-key 命令：

```
aws kms create-key
```

輸出包含金鑰 ID 和金鑰的 Amazon Resource Name (ARN)。下列為範例輸出：

```
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "Enabled": true,
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1478910250.94,
        "Arn": "arn:aws:kms:us-west-2:123456789012:key/6f815f63-e628-448c-8251-
e40cb0d29f59",
        "AWSAccountId": "123456789012",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

步驟 2：設定 KMS 金鑰許可

根據預設，所有 KMS 金鑰皆屬私有。只有資源擁有者可以使用它來加密和解密資料。然而，資源擁有者可以授予其他使用者和資源存取金鑰的許可。透過此步驟，您可以授與 CloudWatch 記錄服務主體使用金鑰的權限。此服務主體必須位於儲存金鑰的相同 AWS 區域。

最佳作法是，建議您將金鑰的使用限制為僅使用您指定的 AWS 帳戶。

首先，使`policy.json`用下列get-key-policy命令將 KMS 金鑰的預設原則儲存為：

```
aws kms get-key-policy --key-id key-id --policy-name default --output text > ./
policy.json
```

在文字編輯器中開啟 `policy.json` 檔案，並從下列其中一個陳述式中加入區段 (以粗體顯示)。使用逗號從新陳述式中分隔現有陳述式。這些陳述式使用`Condition`區段來增強 AWS KMS 金鑰的安全性。如需詳細資訊，請參閱 AWS KMS 金鑰與加密內容。

此範例中的`Condition`區段會限制指定帳戶中「 CloudWatch 記錄見解」查詢結果的 AWS KMS 金鑰使用。

```
{
 "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::account_ID:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt*",
                "kms:Decrypt*",
```

```
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                    "aws:SourceArn": "arn:aws:logs:region:account_ID:query-result:*"
                },
                "StringEquals": {
                    "aws:SourceAccount": "Your_account_ID"
                }
            }
        }
    ]
}
```

最後，使用下列put-key-policy命令新增更新的原則：

```
aws kms put-key-policy --key-id key-id --policy-name default --policy file://
policy.json
```

步驟 3：為 KMS 金鑰與您的查詢結果建立關聯

若要為 KMS 金鑰與帳戶中的查詢結果建立關聯

使用 disassociate-kms-key 命令，如下所示：

```
aws logs associate-kms-key --resource-identifier "arn:aws:logs:region:account-id:query-
result:*" --kms-key-id "key-arn"
```

步驟 4：將金鑰與帳戶中的查詢結果取消關聯

若要取消與查詢結果相關聯的 KMS 金鑰之關聯，請使用下列disassociate-kms-key命令：

```
aws logs disassociate-kms-key --resource-identifier "arn:aws:logs:region:account-
id:query-result:*"
```

使用自然語言產生和更新 CloudWatch 日誌見解查詢

> **Note**
>
> 此功能在美國東部 (維吉尼亞北部)、美國西部 (奧勒岡) 和亞太區域 (東京) 適用於 CloudWatch Logs。

CloudWatch 記錄支援自然語言查詢功能，可協助您產生和更新CloudWatch 日誌深入解析和CloudWatch 指標深入解析的查詢。

使用此功能，您可以用簡單的英文詢問或描述您正在尋找的 CloudWatch 日誌數據的問題。自然語言功能會根據您輸入的提示產生查詢，並提供查詢運作方式的 line-by-line 說明。也可以更新查詢以進一步調查您的資料。

視您的環境而定，您可以輸入類似的提示：「依位元組傳輸的前 100 個來源 IP 位址是多少？」 和「找到 10 個最慢的 Lambda 函數請求。」

若要使用此功能產生 CloudWatch Logs Insights 查詢，請開啟 CloudWatch Logs Insights 查詢編輯器，選取要查詢的記錄群組，然後選擇 [產生查詢]。

> **Important**
>
> 若要使用自然語言查詢功能，您必須使用CloudWatchLogsFullAccessCloudWatchLogsReadOnlyAccess、AdministratorAccess、或ReadOnlyAccess策略。
> 也可以在新的或現有的客戶管理政策或內嵌政策中包含 `cloudwatch:GenerateQuery` 動作。

查詢範例

本節中的範例說明如何使用自然語言功能產生及更新查詢。

> **Note**
>
> 如需有關日 CloudWatch 誌見解查詢編輯器和語法的詳細資訊，請參閱CloudWatch 記錄見解查詢語法。

範例：產生自然語言查詢

若要使用自然語言產生查詢，請輸入提示並選擇產生新查詢。此範例顯示執行基本搜尋的查詢。

提示

以下是一個提示範例，指示功能搜尋 10 個最慢的 Lambda 函數叫用。

```
Find the 10 slowest requests
```

Query

以下是自然語言功能根據提示產生的查詢範例。請注意提示在查詢前出現在註解中的方式。查詢之後，您可以閱讀描述查詢運作方式的說明。

```
# Find the 10 slowest requests
fields @timestamp, @message, @duration
| sort @duration desc
| limit 10
# This query retrieves the timestamp, message and duration fields from the logs and
 sorts them in descending order by duration to find the 10 slowest requests.
```

> **Note**
>
> 若要關閉提示的外觀以及查詢運作方式的說明，請使用編輯器中的齒輪圖示。

範例：更新自然語言查詢

可以透過編輯初始提示，然後選擇更新查詢來更新查詢。

更新提示

下列範例顯示先前提示的更新版本。此提示現在不會顯示搜尋 10 個最慢 Lambda 函數叫用的提示，而是指示搜尋 20 個最慢 Lambda 函數叫用的功能，並包含另一個資料欄以供其他記錄事件使用。

```
Show top 20 slowest requests instead and display requestId as a column
```

更新查詢

以下是更新查詢的範例。請注意更新後的提示在更新後的查詢前出現在註解中的方式。查詢之後，您可以閱讀描述原始查詢更新方式的說明。

```
# Show top 20 slowest requests instead and display requestId as a column
fields @timestamp, @message, @requestId, @duration
| sort @duration desc
| limit 20
# This query modifies the original query by replacing the @message field with the
 @requestId field and changing the limit from 10 to 20 to return the top 20 log events
 by duration instead of the top 10.
```

選擇不使用您的資料以改善服務

您提供用於訓練 AI 模型並產生相關查詢的自然語言提示資料僅用於提供和維護您的服務。此資料可能會用來改善 CloudWatch 日誌深入解析的品質。我們將您的信任和隱私以及內容安全性放在首位。如需詳細資訊，請參閱 AWS 服務條款和 AWS 負責任的 AI 政策。

透過建立 AI 服務退出政策，可選擇不將您的內容用於開發或改進自然語言查詢的品質。若要選擇退出所有 CloudWatch Logs AI 功能的資料收集 (包括查詢產生功能)，您必須建立 CloudWatch Logs 的退出政策。如需詳細資訊，請參閱《AWS Organizations 使用者指南》中的 AI 服務退出政策。

記錄異常偵測

您可以為每個記錄群組建立記錄異常偵測器。異常偵測器會掃描擷取到記錄群組中的記錄事件，並在記錄資料中尋找異常。異常偵測使用機器學習和模式辨識來建立典型日誌內容的基準。

在您為記錄群組建立異常偵測器之後，它會使用記錄群組中過去兩週的記錄事件進行訓練，以進行訓練。訓練期間最多可能需要 15 分鐘。訓練完成後，它會開始分析傳入的記錄檔以識別異常，而異常會顯示在 CloudWatch 記錄主控台中供您檢查。

CloudWatch 日誌模式識別通過識別日誌中的靜態和動態內容來提取日誌模式。病毒碼對於分析大型記錄集很有用，因為通常可以將大量記錄事件壓縮成幾個病毒碼。

例如，請參閱下列三個記錄事件的範例。

```
2023-01-01 19:00:01 [INFO] Calling DynamoDB to store for resource id 12342342k124-12345
2023-01-01 19:00:02 [INFO] Calling DynamoDB to store for resource id 324892398123-12345
2023-01-01 19:00:03 [INFO] Calling DynamoDB to store for resource id 3ff231242342-12345
```

在上一個範例中，所有三個記錄事件都遵循一個模式：

```
<*> <*> [INFO] Calling DynamoDB to store for resource id <*>
```

模式中的字段稱為令牌。模式中不同的欄位 (例如要求 ID 或時間戳記) 稱為動態權杖。<*>當 CloudWatch 記錄檔顯示病毒碼時，動態權杖會以表示。為動態令牌找到的每個不同值稱為令牌值。

動態權杖的常見範例包括錯誤碼、時間戳記和要求 ID。

記錄異常偵測會使用這些模式來尋找異常。異常偵測器模型訓練期結束後，系統會根據已知趨勢評估記錄。異常檢測器將重大波動標記為異常。

建立記錄異常偵測器不會產生費用。

異常和模式的嚴重性和優先順序

記錄異常偵測器找到的每個異常都會指派優先順序。找到的每個模式都會指派一個嚴重性。

- 系統會自動計算優先順序，並根據陣列的嚴重性等級和與預期值的偏差量而定。例如，如果某個令牌值突然增加 500%，即使其嚴重性為，該異常也可能被指定為HIGH優先順序。NONE
- 嚴重性僅基於模式中找到的關鍵字FATAL，例如ERROR、和WARN。如果找不到這些關鍵字，則模式的嚴重性會標記為NONE。

異常可見性時間

建立異常偵測器時，您可以指定異常偵測器的最大異常可見性期間。這是異常在主控台中顯示並由 ListAnomaliesAPI 作業傳回的天數。經過此時間段後發生異常，如果它繼續發生，它會自動被接受為常規行為，並且異常檢測器模型停止將其標記為異常。

如果您在建立異常偵測器時未調整可見性時間，預設值會使用 21 天。

抑制異常

發現異常後，您可以選擇暫時或永久抑制它。抑制異常會導致異常偵測器停止將此發生標記為您指定的時間量的異常。當您抑制異常時，您可以選擇僅隱藏該特定異常，或隱藏與發現異常的模式相關的所有異常。

您仍然可以在主控台中檢視隱藏的異常情況。您也可以選擇停止抑制它們。

常見問答集

是否會 AWS 使用我的資料來訓練機器學習演算法，以供其他客戶 AWS 使用？

沒有 訓練所建立的異常偵測模型是以記錄群組中的記錄事件為基礎，而且只能在該記錄群組和該 AWS 帳戶中使用。

哪些類型的記錄事件可與異常偵測搭配使用？

記錄異常偵測非常適合：應用程式記錄檔和其他類型的記錄，其中大多數記錄項目都符合典型模式。具有包含記錄層級或嚴重性關鍵字 (例如 INFO、ERROR 和 DEBUG) 的事件的記錄群組特別適合用於記錄異常偵測。

記錄異常偵測不適用於：具有極長 JSON 結構的記錄事件，例如 CloudTrail 記錄檔。模式分析最多只會分析記錄行的前 1500 個字元，因此會略過超出該限制的任何字元。

稽核或存取記錄 (例如 VPC 流程記錄) 在異常偵測方面的成功也會降低。異常偵測是為了找出應用程式問題，因此可能不適合網路或存取異常。

為了協助您判斷異常偵測器是否適用於特定記錄群組，請使用 CloudWatch 記錄檔模式分析來尋找群組中記錄事件中的模式數目。如果圖案的數量不超過 300，則異常偵測可能會運作良好。如需陣列分析的更多資訊，請參閱模式分析。

什麼被標記為異常？

發生下列情況可能會導致記錄事件標記為異常：

- 具有先前在記錄群組中看不到模式的記錄事件。
- 已知模式的顯著變化。
- 動態權杖的新值，其中包含一組離散的常用值。
- 動態令牌值出現次數的大幅變化。

雖然前面的所有項目都可能被標記為異常，但它們並不意味著應用程序的效能不佳。例如，一 higher-than-usual些成200功值可能會標記為異常。在這種情況下，您可能會考慮抑制這些不表示問題的異常情況。

被屏蔽的敏感數據會發生什麼？

不會掃描任何被遮罩為敏感資料的記錄事件部分是否有異常。如需有關遮罩機密資料的詳細資訊，請參閱使用遮罩來協助保護敏感記錄資料。

在日誌群組上啟用異常偵測

使用下列步驟來使用 CloudWatch 主控台建立日誌異常偵測器，以掃描記錄群組是否有異常。

您也可以透過程式設計方式建立異常偵測器。如需詳細資訊，請參閱CreateLogAnomalyDetector。

若要建立記錄異常偵測器

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 選擇「記錄檔」、「記錄異常」。
3. 選擇 [建立異常偵測器]。
4. 選取要為其建立此異常偵測器的記錄群組。
5. 在異常偵測器名稱中輸入偵測器的名稱。
6. (選擇性) 將評估頻率從預設值 5 分鐘變更。根據記錄群組接收新記錄檔的頻率來設定此值。例如，如果記錄群組每 10 分鐘批次收到新的記錄事件，則可能需要將評估頻率設定為 15 分鐘。
7. (選擇性) 若要將異常偵測器設定為僅在包含特定字詞或字串的記錄事件中尋找異常，請選擇篩選模式。

   然後，在異常偵測篩選器模式中輸入模式。如需有關模式語法的詳細資訊，請用於指標篩選條件、訂閱篩選條件、篩選條件日誌事件和 Live Tail 的篩選條件模式語法。

（可選）若要測試您的篩選器模式，請在「記錄事件訊息」中輸入一些記錄訊息，然後選擇「測試模式」。

8. (選擇性) 若要從預設值變更異常可見性期間，或將 AWS KMS 金鑰與此異常偵測器建立關聯，請選擇 [進階組態]。

   a. 若要從預設值變更異常可見性期間，請在異常可見性期間上限 (天數) 中輸入新值。

   b. 若要將金 AWS KMS 鑰與此異常偵測器產生關聯，請在 KMS 金鑰 ARN 中輸入 ARN。如果您分配了密鑰，則此檢測器發現的異常信息將在靜態時使用密鑰進行加密。使用者必須擁有此金鑰和異常偵測器的權限，才能擷取其找到之異常的相關資訊。

      您也必須確定 CloudWatch Logs 服務主體具有使用金鑰的權限。如需詳細資訊，請參閱 使用以下方式加密異常偵測器及其結果 AWS KMS。

9. 選擇「啟用異常偵測」。

   系統會建立異常偵測器，並根據記錄群組擷取的記錄事件開始訓練其模型。大約 15 分鐘後，異常偵測就會啟動，並開始尋找並顯示異常。

檢視已找到的異常

建立一或多個記錄異常偵測器之後，您可以使用 CloudWatch 主控台來檢視它們發現的異常狀況。

您可以以程式設計方式檢視異常。如需詳細資訊，請參閱ListAnomalies。

若要檢視所有記錄異常偵測器找到的異常

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 選擇「記錄檔」、「記錄異常」。

   「記錄異常」表格隨即出現。「記錄異常」旁邊頂端的數字會顯示表格中列出的記錄異常數目。表格中的每一列都會顯示下列資訊：

   - 「異常」欄會顯示異常的簡短摘要。這些摘要由 CloudWatch 記錄檔產生。
   - 異常的優先級。系統會根據記錄事件中的變更量、關鍵字 (例如在記錄事件中`Exception`發生) 等，自動計算優先順序。
   - 異常所依據的記錄模式。如需有關陣列的更多資訊，請參閱記錄異常偵測。
   - 異常記錄趨勢會顯示長條圖，說明符合該模式的記錄數量。
   - 上次偵測時間會顯示最近發現此異常的時間。

- 首次偵測時間會顯示第一次發現此異常的時間。
- 異常偵測器會顯示包含與此異常相關之記錄事件的記錄群組名稱。您可以選擇此名稱來查看日誌群組詳細資訊頁面。

3. 若要進一步檢查一個異常狀況，請選擇其列中的選項按鈕。

   樣式檢查窗格隨即出現，並顯示下列內容：

   - 該模式，這種異常是基於。在模式中選擇一個令牌以分析該令牌的值。
   - 顯示查詢時間範圍內異常發生次數的長條圖。
   - [記錄範例] 索引標籤會顯示屬於異常一部分的一些記錄事件。
   - 如果您已選取動態權杖，則「記號值」標籤會顯示所選動態權杖的值。

   > **Note**
   >
   > 每個記號最多可擷取 10 個記號值。令牌計數可能不精確。 CloudWatch 日誌使用概率計數器來生成令牌計數，而不是絕對值。

4. 若要隱藏異常，請選擇其列中的選項按鈕，然後執行下列動作：

   a. 選擇「動作」，「隱藏異常」。

   b. 然後指定要抑制異常的時間長度。

   c. 若要隱藏與此陣列相關的所有異常，請選取「抑制陣列」。

   d. 選擇「隱藏異常」。

若要檢視在單一記錄群組中發現的異常

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 選擇日誌、日誌群組。
3. 選擇記錄群組的名稱，然後選擇 [異常偵測] 索引標籤。

   「異常偵測」表隨即出現。「記錄異常」旁邊頂端的數字會顯示表格中列出的記錄異常數目。表格中的每一列都會顯示下列資訊：

   - 「異常」欄會顯示異常的簡短摘要。這些摘要由 CloudWatch 記錄檔產生。
   - 異常的優先級。系統會根據記錄事件中的變更量、關鍵字 (例如在記錄事件中`Exception`發生) 等，自動計算優先順序。

- 異常所依據的記錄模式。如需有關陣列的更多資訊，請參閱記錄異常偵測。
- 異常記錄趨勢會顯示長條圖，說明符合該模式的記錄數量。
- 上次偵測時間會顯示最近發現此異常的時間。
- 首次偵測時間會顯示第一次發現此異常的時間。

4. 若要進一步檢查一個異常狀況，請選擇其列中的選項按鈕。

   樣式檢查窗格隨即出現，並顯示下列內容：

   - 該模式，這種異常是基於。在模式中選擇一個令牌以分析該令牌的值。
   - 顯示查詢時間範圍內異常發生次數的長條圖。
   - [記錄範例] 索引標籤會顯示屬於異常一部分的一些記錄事件。
   - 如果您已選取動態權杖，則「記號值」標籤會顯示所選動態權杖的值。

   > **Note**
   >
   > 每個記號最多可擷取 10 個記號值。令牌計數可能不精確。 CloudWatch 日誌使用概率計數器來生成令牌計數，而不是絕對值。

5. 若要隱藏異常，請選擇其列中的選項按鈕，然後執行下列動作：

   a. 選擇「動作」，「隱藏異常」。

   b. 然後指定要抑制異常的時間長度。

   c. 若要隱藏與此陣列相關的所有異常，請選取「抑制陣列」。

   d. 選擇「隱藏異常」。

在日誌異常檢測器上創建警報

您可以為日誌群組中的日誌異常偵測器建立警示。您可以指定在指定期間內在記錄群組中發現指定數目的異常時，警示進入ALARM狀態。您也可以使用篩選器，以便警示僅計算指定優先順序的異常。

若要建立記錄異常偵測器的警示

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇 [記錄] > [記錄異常]。

   記錄異常偵測器的表格隨即出現。

3. 選擇您要設定鬧鐘的異常偵測器的圓形按鈕，然後選擇 [建立鬧鐘]。

   CloudWatch 警示建立精靈隨即出現。此LogAnomalyDetector欄位會顯示您選擇的異常偵測器的名稱。「度量名稱」欄位隨即顯示AnomalyCount。

4. (選擇性) 若要篩選此警示的異常優先順序，請執行下列其中一個動作：

   - 若要讓警示僅計數高優先順序異常，請輸入 **`HIGH`** for。LogAnomalyPriority
   - 若要讓警示僅計數高優先順序和中優先順序異常，請輸入 **`MEDIUM`** for。LogAnomalyPriority

   如需優先順序層級的詳細資訊，請參閱異常和模式的嚴重性和優先順序。

5. 選擇針對警示使用靜態或量度異常偵測閾值。此選項決定警示臨界值的設定方式。靜態臨界值表示警示臨界值是您選擇的靜態常數。異常偵測閾值表示 CloudWatch 決定常用值的範圍，如果實際計數超過此頻帶的閾值，警報就會觸發。您不必為記錄異常偵測警示選擇異常偵測。如需量度異常偵測的詳細資訊，請參閱使用 CloudWatch 異常偵測。

6. 對於無論何時如***`your-metric-name`***此。 ，選擇「大於」、「大於/等於」、「低/等於」或「小於」。對於相比...，為閾值指定一個數字。如果異常偵測器在「週期」指定的時間內發現超過此數目的警報，則警報會進入ALARM狀態。

7. 選擇 Additional configuration (其他組態)。針對 Datapoints to alarm (要警示的資料點)，請指定 (資料點) 必須處於 ALARM 狀態多少評估期間，才會觸發警示。如果此處的兩個值相符，您便可以建立警示，在許多連續期間違規時移至 ALARM 狀態。

   若要建立 N 個中有 M 個警示，請針對小於第二個值之數字的第一個值指定數字。如需詳細資訊，請參閱評估警示。

8. 對於 Missing data treatment (遺失資料處理方式)，選擇警示在遺失某些資料點時的行為。如需詳細資訊，請參閱設定 CloudWatch 警示如何處理遺失的資料。

9. 選擇下一步。

10. 對於通知，選擇新增通知，然後指定 Amazon SNS 主題，以在警示轉換到ALARM、OK或INSUFFICIENT_DATA狀態時通知。

    a. (選用) 若要針對相同警示狀態或不同警示狀態傳送多個通知，請選擇 Add notification (新增通知)。

    > **Note**
    >
    > 建議您設定警示，以便除了在進入警示狀態外，進入資料不足狀態時應採取動作。這是因為連線至資料來源的 Lambda 函數有許多問題可能會導致警示轉換為資料不足。

b. (選用) 若不傳送 Amazon SNS 通知，請選擇移除。

11. (選擇性) 如果您希望警示針對 Amazon EC2 Auto Scaling、Amazon EC2、票證執行動作，或者 AWS Systems Manager，請選擇適當的按鈕，然後指定警示狀態和動作。

> **Note**
>
> 警示僅在處於 ALARM 狀態時執行 Systems Manager 動作。如需有關 Systems Manager 動作的資訊，請參閱設 CloudWatch 定以建立 OpsItems和事件建立。

12. 選擇下一步。

13. 在 Name and description (名稱和描述) 下，輸入警示的名稱和描述，然後選擇 Next (下一步)。此名稱只能包含 UTF-8 字元，不能包含 ASCII 控制字元。說明可以包括降價格式，僅顯示在 CloudWatch 控制台的警報詳細資料索引標籤中。Markdown 對於將連結新增至執行手冊或其他內部資源很實用。

> **Tip**
>
> 警示名稱只能包含 UTF-8 字元。它不能包含 ASCII 控制字元。

14. 在 Preview and create (預覽及建立) 下，請確認警示資訊和條件都是正確的，然後選擇 Create alarm (建立警示)。

日誌異常檢測器發布的指標

CloudWatch 記錄會將指AnomalyCount標發佈至 CloudWatch 指標。此測量結果會發佈至命AWS/Logs名空間。

AnomalyCount量度會以下列維度發佈：

- LogAnomalyDetector— 異常檢測器的名稱
- LogAnomalyPriority— 異常的優先級

使用以下方式加密異常偵測器及其結果 AWS KMS

異常偵測器資料一律會在 CloudWatch 記錄中加密。根據預設， CloudWatch 記錄檔會對靜態資料使用伺服器端加密。您也可以使用 AWS Key Management Service 進行此加密。如果這樣做，則使用

密 AWS KMS 鑰完成加密。透過將 KMS 金鑰與異常偵測器建立關聯，可在異常偵測器層級啟用 AWS KMS 加密。

> **Important**
>
> CloudWatch 記錄僅支援對稱 KMS 金鑰。請勿使用非對稱金鑰來加密日誌群組中的資料。如需詳細資訊，請參閱使用對稱和非對稱金鑰。

限制

- 若要執行下列步驟，您必須擁有下列許可：`kms:CreateKey`、`kms:GetKeyPolicy` 和 `kms:PutKeyPolicy`。
- 將金鑰與異常偵測器建立關聯或取消關聯後，作業最多可能需要五分鐘才會生效。
- 如果您撤銷對關聯金鑰的 CloudWatch 記錄存取權或刪除關聯的 KMS 金鑰，則無法再擷取 CloudWatch 記錄中的加密資料。

步驟 1：建立 AWS KMS 金鑰

若要建立 KMS 金鑰，請使用以下 create-key 命令：

```
aws kms create-key
```

輸出包含金鑰 ID 和金鑰的 Amazon Resource Name (ARN)。下列為範例輸出：

```
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "key-default-1",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "Enabled": true,
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1478910250.94,
        "Arn": "arn:aws:kms:us-west-2:123456789012:key/key-default-1",
        "AWSAccountId": "123456789012",
```

```
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

步驟 2：設定 KMS 金鑰許可

依預設，所有 AWS KMS 金鑰都是私密金鑰。只有資源擁有者可以使用它來加密和解密資料。然而，資源擁有者可以授與其他使用者和資源存取 KMS 金鑰的許可。透過此步驟，您可以授與 CloudWatch 記錄服務主體使用金鑰的權限。此服務主體必須位於儲存 KMS 金鑰的相同 AWS 區域。

最佳做法是，建議您將 KMS 金鑰的使用限制為僅使用您指定的 AWS 帳戶或異常偵測器。

首先，使`policy.json`用下列get-key-policy命令將 KMS 金鑰的預設原則儲存為：

```
aws kms get-key-policy --key-id key-id --policy-name default --output text > ./
policy.json
```

在文字編輯器中開啟 `policy.json` 檔案，並從下列其中一個陳述式中加入區段 (以粗體顯示)。使用逗號從新陳述式中分隔現有陳述式。這些陳述式使用`Condition`區段來增強 AWS KMS 金鑰的安全性。如需詳細資訊，請參閱 AWS KMS 金鑰與加密內容。

此範例中的`Condition`部分將 AWS KMS 金鑰的使用限制在指定帳戶，但可用於任何異常偵測器。

```
{
  "Version": "2012-10-17",
  "Id": "key-default-1",
  "Statement": [
    {
      "Sid": "Enable IAM User Permissions",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::Your_account_ID:root"
      },
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Principal": {
```

```
        "Service": "logs.REGION.amazonaws.com"
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "*",
      "Condition": {
        "ArnLike": {
          "kms:EncryptionContext:aws:logs:arn":
 "arn:aws:logs:REGION:Your_account_ID:anomaly-detector:*"
        }
      }
    },
    {
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.REGION.amazonaws.com"
        },
        "Action": [
            "kms:Encrypt",
            "kms:Decrypt",
            "kms:ReEncrypt*",
            "kms:GenerateDataKey*",
            "kms:DescribeKey"
        ],
        "Resource": "*",
        "Condition": {
            "ArnLike": {
                "kms:EncryptionContext:aws-crypto-ec:aws:logs:arn":
 "arn:aws:logs:REGION:Your_account_ID:anomaly-detector:*"
            }
        }
    }
  ]
}
```

最後，使用下列put-key-policy命令新增更新的原則：

```
aws kms put-key-policy --key-id key-id --policy-name default --policy file://
policy.json
```

步驟 3：將 KMS 金鑰與異常偵測器建立關聯

當您在主控台中建立或使用或 API 時，您可以將 KMS 金鑰與異常偵測器建立關 AWS CLI 聯。

步驟 4：將金鑰與異常偵測器取消關聯

當金鑰與異常偵測器相關聯之後，您就無法更新金鑰。刪除密鑰的唯一方法是刪除異常檢測器，然後重新創建它。

使用日誌群組和日誌串流

日誌串流是共享相同來源的一系列日誌事件。記錄檔中的每個個別記錄來源都會組成個別的 CloudWatch 記錄資料流。

日誌群組是共享相同保留、監控和存取控制設定的日誌串流群組。您可以定義日誌群組，並指定放入每個群組的串流。可以屬於一個日誌群組的日誌串流數量並沒有限制。

使用本節中的程序來運用日誌群組和日誌串流。

在記錄檔中建立 CloudWatch 記錄群組

當您使用 Amazon CloudWatch 日誌使用者指南前幾節中的步驟在 Amazon EC2 執行個體上安裝 CloudWatch 日誌代理程式時，會在該程序中建立日誌群組。您也可以直接在 CloudWatch 主控台中建立記錄群組。

欲建立日誌群組

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇 Log groups (日誌群組)。
3. 選擇 Actions (動作)，然後選擇 Create log group (建立日誌群組)。
4. 輸入日誌群組名稱，然後選擇 Create log group (建立日誌群組)。

> **Tip**
>
> 您可以在 CloudWatch 主控台導覽窗格中，在 Favorites and recents (我的最愛和最近的項目) 選單中將日誌群組、儀表板以及警示加入最愛。在 Recently visited (最近造訪) 欄下方，將滑鼠游標停留在要加入最愛的日誌群組上，然後選擇其旁邊的星號。

將日誌傳送到日誌群組

CloudWatch 記錄檔會自動從數個 AWS 服務接收記錄事件。您也可以使用下列其中一種方法，將其他 CloudWatch 記錄事件傳送至記錄檔：

- CloudWatch 代理程式 — 統一的 CloudWatch 代理程式可以將指標和記錄檔傳送至 CloudWatch 防護記錄。如需安裝和使用 CloudWatch 代理程式的相關資訊，請參閱 Amazon 使用者指南中的使用 CloudWatch 代理程式從 Amazon EC2 執行個體和現場部署伺服器收集 CloudWatch 指標和日誌。
- AWS CLI將記錄事件的批次put-log-events上傳至 CloudWatch 記錄。
- 以程式設計方式 — PutLogEventsAPI 可讓您以程式設計方式將記錄事件批次上傳至 CloudWatch 記錄。

檢視傳送至 CloudWatch 記錄的記錄檔資料

您可以根據 CloudWatch Logs 代理程式傳送至「記錄檔」的 stream-by-stream 基礎來檢視和捲動記 CloudWatch 錄檔資料。您可以檢視指定時間範圍的日誌資料。

欲查看日誌資料

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇 Log groups (日誌群組)。
3. 針對 Log Groups (日誌群組)，選擇日誌群組以檢視串流。
4. 在日誌群組清單中，選擇您要檢視的日誌群組清單。
5. 在日誌串流清單中，選擇您要檢視的日誌串流名稱。
6. 若要變更日誌資料的顯示方式，請執行以下其中一項：
   - 若要展開單一日誌事件，選擇日誌事件旁的箭頭。
   - 若要展開所有日誌事件並以純文字檢視，請在日誌事件清單上方選擇 Text (文字)。
   - 若要篩選日誌事件，在搜尋欄位中輸入所需的搜尋篩選條件。如需詳細資訊，請參閱使用篩選條件從日誌事件建立指標。
   - 若要檢視指定日期和時間範圍內的日誌資料，請在搜尋篩選條件旁，選擇日期和時間旁的箭頭。若要指定日期和時間範圍，請選擇 Absolute (絕對)。若要選擇預先定義的分鐘數、小時數、天數或週數，請選擇 Relative (相對)。您也可以在 UTC 和 Local timezone (本機時區) 間進行切換。

使用 Live Tail 以近乎即時的方式檢視日誌

CloudWatch Logs Live Tail 可透過檢視擷取新記錄事件的串流清單，協助您快速進行事件疑難排解。可以近乎即時地檢視、篩選和反白顯示擷取的日誌，協助您快速偵測並解決問題。可以根據指定的詞彙篩選日誌，並反白顯示包含指定詞彙的日誌，以協助您快速找到查詢內容。

Live Tail 工作階段會按工作階段使用時間 (每分鐘) 產生費用。如需有關定價的詳細資訊，請參閱 Amazon 定 CloudWatch 價中的日誌索引標籤。

> **Note**
>
> 只有標準記錄檔類別中的記錄群組才支援 Live Tail。如需記錄類別的詳細資訊，請參閱日誌類。

以下各節說明如何在主控台中使用 Live Tail。您也可以以程式設計方式啟動 Live Tail 工作階段。如需詳細資訊，請參閱StartLiveTail。如需 SDK 範例，請參閱使用 AWS SDK 啟動即時尾端工作階段。

開始 Live Tail 工作階段

您可以使用 CloudWatch 主控台來啟動即時尾端工作階段。下列程序說明如何使用左側導覽窗格中的 Live tail 選項來啟動 Live Tail 工作階段。您也可以從 [記錄群組] 頁面或 [CloudWatch 記錄檔見解] 頁面啟動即時尾端工作階段。

> **Note**
>
> 如果您使用資料保護政策來遮罩透過 Live Tail 檢視之日誌群組中的敏感資料，則敏感資料在 Live Tail 工作階段中顯示為遮罩狀態。如需有關遮罩日誌群組敏感資料的詳細資訊，請參閱 使用遮罩功能協助保護敏感日誌資料。

開始 Live Tail 工作階段

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇日誌，然後選擇 Live tail。
3. 針對選取日誌群組，在 Live Tail 工作階段中，選取您要從中檢視事件的日誌群組。您最多可以選取 10 個日誌群組。
4. (選用) 如果您只選取一個日誌群組，則可以選取一個或多個日誌串流來檢視日誌事件，從而進一步篩選 Live Tail 工作階段。若要這樣做，請在選取日誌串流下，從下拉式清單中選取日誌串流的名稱。或者，您也可以使用選取日誌串流下的第二個方塊輸入日誌串流名稱字首，然後選取名稱與該字首相符的所有日誌串流。
5. (選用) 若要僅顯示包含特定字詞或其他字串的日誌事件，請在 `Add filter patterns` 中輸入字詞或字串。

例如，若僅顯示包含 Warning 字詞的日誌事件，請輸入 **Warning**。篩選條件欄位區分大小寫。可以在此欄位中包含多個詞彙和模式運算子。

- **error 404** 僅顯示包含 error 和 404 的日誌事件
- **?Error ?error** 顯示包含 Error 或 error 的日誌事件
- **-INFO** 顯示不包含 INFO 的所有日誌事件
- **{ $.eventType = "UpdateTrail" }** 顯示事件類型欄位值為 UpdateTrail 的所有 JSON 日誌事件

您也可以使用規則表達式 (regex) 來篩選：

- **%ERROR%** 使用 regex 顯示包含 ERROR 關鍵字的所有日誌事件
- **{ $.names = %Steve% }** 使用 regex 顯示 Steve 為 "name" 屬性中的 JSON 日誌事件
- **[w1 = %abc%, w2]** 使用 regex 顯示以空格分隔且第一個單詞為 abc 的日誌事件

如需有關模式語法的詳細資訊，請參閱篩選條件模式語法。

6. (選用) 若要反白顯示某些顯示的日誌事件，請輸入要搜尋的詞彙，並在 Live Tail 下反白顯示。一次輸入一個反白顯示詞彙。如果新增多個詞彙進行反白顯示，則會指派不同的顏色來代表每個詞彙。反白顯示指示器會顯示在任何包含指定詞彙的日誌事件的左側，當您展開主視窗中的日誌事件以檢視完整日誌事件時，也會出現在詞彙本身下方。

   您可以使用篩選功能以及反白顯示來快速疑難排解問題。例如，您可以篩選事件以僅顯示包含 Error 的事件，然後反白顯示包含 404 的事件。

7. 若要啟動工作階段，請選擇套用篩選條件

   相符的日誌事件開始出現在視窗中。也會顯示下列資訊：

   - 計時器會顯示 Live Tail 工作階段已啟用的時間長度。
   - 事件數/秒會顯示每秒有多少個擷取的日誌事件與您設定的篩選條件相符。
   - 為了避免工作階段捲動速度過快，因為許多事件符合篩選器， CloudWatch 記錄檔可能只會顯示一些相符的事件。如果發生這種情況，螢幕上顯示的相符事件百分比會以 % 的形式顯示。

8. 若要暫停事件流程以調查目前顯示的內容，請按一下事件視窗中的任意位置。
9. 在工作階段期間，您可以使用下列項目來查看有關每個日誌事件的詳細資訊。

   - 若要在主視窗中顯示日誌事件的完整文字，請選擇該日誌事件旁邊的箭頭。

- 若要在側視窗中顯示日誌事件的完整文字，請選擇該日誌事件旁邊的 + 放大鏡。事件流程會暫停，並顯示側視窗。

  在側視窗中顯示日誌事件文字非常有用，可比較其文字與主視窗中的其他事件。

10. 若要停止 Live Tail 工作階段，請選擇停止。
11. 若要重新啟動工作階段，可選擇使用篩選面板修改篩選條件，然後選擇套用篩選條件。然後選擇 Start (啟動)。

使用篩選條件模式搜尋日誌資料

您可以使用用於指標篩選條件、訂閱篩選條件、篩選條件日誌事件和 Live Tail 的篩選條件模式語法來搜尋日誌資料。您可以搜尋日誌群組內的所有日誌資料流，或使用 AWS CLI 您也可以搜尋特定的日誌串流。每個搜尋執行時，它會傳回最多找到的第一個資料頁面，以及可擷取下一個頁面資料或繼續搜尋的字符。如果沒有傳回結果，您可以繼續搜尋。

您可以設定您想要查詢的時間範圍，以限制搜尋的範圍。您可以先從較大範圍開始，以查看您有興趣記錄分類的日誌行，然後縮短時間範圍以將視圖範圍限制在您感興趣的時間範圍之日誌。

您也可以直接從您的日誌擷取的指標轉換到對應的日誌。

如果您已登入設定為 CloudWatch 跨帳戶可觀察性監控帳戶的帳戶，則可以從連結至此監視帳戶的來源帳戶中搜尋和篩選記錄事件。如需詳細資訊，請參閱CloudWatch 跨帳戶可觀察性。

使用主控台搜尋日誌項目

您可以使用主控台搜尋與指定條件相符的日誌項目。

若要使用主控台搜尋日誌

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇 Log groups (日誌群組)。
3. 針對 Log Groups (日誌群組)，輸入包含要搜尋之日誌串流的日誌群組名稱。
4. 對於 Log Streams (日誌串流)，選擇要搜尋的日誌串流名稱。
5. 在 Log events (日誌事件) 下方，輸入要使用的篩選條件語法。

若要使用主控台搜尋某時間範圍內的所有日誌項目

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/

2. 在導覽窗格中，選擇 Log groups (日誌群組)。
3. 針對 Log Groups (日誌群組)，輸入包含要搜尋之日誌串流的日誌群組名稱。
4. 選擇 Search log group (搜尋日誌群組)。
5. 針對 Log events (日誌事件)，選取日期和時間範圍，然後輸入篩選條件語法。

使用搜尋記錄項目 AWS CLI

您可以使用搜尋符合指定條件的記錄項目 AWS CLI。

使用搜尋記錄項目 AWS CLI

在命令提示字元中，執行下列filter-log-events命令。使用 `--filter-pattern`，將結果限制在指定的篩選條件模式，並使用 `--log-stream-names`，將結果限制在指定的日誌串流。

```
aws logs filter-log-events --log-group-name my-group [--log-stream-
names LIST_OF_STREAMS_TO_SEARCH] [--filter-pattern VALID_METRIC_FILTER_PATTERN]
```

若要使用搜尋指定時間範圍內的記錄項目 AWS CLI

在命令提示字元中，執行下列filter-log-events命令：

```
aws logs filter-log-events --log-group-name my-group [--log-stream-
names LIST_OF_STREAMS_TO_SEARCH] [--start-time 1482197400000] [--end-
time 1482217558365] [--filter-pattern VALID_METRIC_FILTER_PATTERN]
```

從指標轉換到日誌

您可以從主控台的其他部分取得特定的日誌項目。

若要從儀表板 widget 取得日誌

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇 Dashboards (儀表板)。
3. 選擇儀表板。
4. 在 widget 中，選擇 View logs (檢視日誌) 圖示，然後選擇 View logs in this time range (查看在這個時間範圍內的日誌)。如果有多個指標篩選條件，從清單中選取一個。如果指標篩選條件的數量

多於我們可在清單中顯示的數量，選擇 More metric filters (更多指標篩選條件)，然後選取或搜尋指標篩選條件。

從指標取得到日誌

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇 指標。
3. 在搜尋欄位中 All metrics (所有指標) 索引標籤上，輸入指標的名稱，然後按 Enter 鍵。
4. 從搜尋結果選取一或多個指標。
5. 選擇 Actions (動作)、View logs (查看日誌)。如果有多個指標篩選條件，從清單中選取一個。如果指標篩選條件的數量多於我們可在清單中顯示的數量，選擇 More metric filters (更多指標篩選條件)，然後選取或搜尋指標篩選條件。

故障診斷

Search takes too long to complete (需要很長的時間才能完成搜尋)

如果您有許多日誌資料，搜尋可能需要很長的時間來完成。若要加速搜尋，您可以執行以下操作：

- 如果您使用的是 AWS CLI，您可以將搜尋限制為只有您感興趣的記錄資料流。例如，如果您的記錄群組有 1000 個記錄串流，但您只想看到三個您知道相關的記錄資料流，您可以使用 AWS CLI 將搜尋限制在日誌群組中只有這三個記錄資料流。
- 使用較短、更精細的時間範圍，這可減少搜尋的資料量，和加速查詢。

變更 CloudWatch 記錄檔中的記錄資料保留

根據預設，記錄資料會無限期地儲存在 CloudWatch 防護記錄中。不過，您可以設定要將日誌群組中的日誌資料存放多久時間。超過目前保留期間設定的任何資料都會被刪除。您可隨時變更每一群組的日誌保留期間。

> **Note**
>
> CloudWatch Logs 達到保留設定時，不會立即刪除記錄事件。通常需要經過長達 72 小時才會刪除日誌事件，但在極少數情況下可能需要更長的時間。
> 這表示，如果您將日誌群組變更為在包含超過到期日但尚未實際刪除的日誌事件時具有較長的保留設定，則在到達新的保留日期之後，這些日誌事件最多需要 72 小時才會刪除。若要確保

永久刪除日誌資料，請將日誌群組保持為較低的保留設定，直到先前的保留期間結束後 72 小時為止，或者您確認已刪除舊版的日誌事件為止。
當日誌事件達到其保留設定時，系統會將其標示以供刪除。經標示以供刪除後，即使之後並未真正刪除，它們也不會再加入您的封存儲存成本。當您使用 API 來擷取 `storedBytes` 值來查看日誌群組正在儲存多少位元組時，這些經標示以供刪除的日誌事件不會納入其中。

若要變更日誌保留期間設定

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中依序選擇 Logs (日誌)、Log groups (日誌群組)。
3. 尋找日誌群組以更新。
4. 在該記錄群組的 [保留] 資料行中，選擇目前的保留設定，例如 [永不過期]。
5. 在 [保留] 設定中，選擇記錄保留值，然後選擇 [儲存]。

在 Amazon CloudWatch 日誌中標記日誌群組

您可以以標籤形式將自己的中繼資料指派給在 Amazon CloudWatch Logs 中建立的日誌群組。標籤是您為日誌群組定義的鍵值對。使用標籤是管理 AWS 資源和組織資料 (包括帳單資料) 的簡單而強大的方式。

> **Note**
>
> 您可以使用標籤來控制對 CloudWatch 記錄檔資源的存取，包括記錄群組和目的地。由於日誌群組與日誌串流之間的階層關係，系統會在日誌群組層級控制對日誌串流的存取。如需有關使用標籤來控制存取的詳細資訊，請參閱使用標籤控制對 Amazon Web Services 資源的存取。

目錄

標籤基本概念

您可以 AWS CloudFormation 使 AWS CLI用或 CloudWatch 記錄 API 來完成下列工作：

- 當您建立日誌群組時為其新增標籤。
- 新增標籤到現有的日誌群組。
- 列出日誌群組的標籤。
- 從日誌群組移除標籤。

您可以使用標籤來分類您的日誌群組。例如，您可以依用途、擁有者或環境來為它們分類。由於您定義了每個標籤的金鑰和值，您可以建立一組自訂的類別，以符合您的特定需求。例如，您可以定義一組標籤，可協助您依照日誌群組的擁有者和關聯的應用程式來追蹤日誌群組。以下是數個標籤的範例：

- 專案：專案名稱
- 擁有者：名稱
- 用途：負載測試
- 應用程式：應用程式名稱
- 環境：生產

使用標記追蹤成本

您可以使用標籤來分類和追蹤 AWS 成本。當您將標記套用至 AWS 資源 (包括記錄群組) 時，您的 AWS 成本分配報告會包含依標籤彙總的使用量和成本。您可以套用代表業務類別 (例如成本中心、應用程式名稱或擁有者) 的標籤，來整理多個服務中的成本。如需詳細資訊，請參閱《AWS Billing 使用者指南》中的將成本分配標籤用於自訂帳單報告。

標籤限制

下列限制適用於標籤。

基本限制

- 每個日誌群組的標籤數上限為 50。
- 標籤鍵與值皆區分大小寫。
- 您無法變更或編輯已刪除日誌群組的標籤。

標籤鍵限制

- 每個標籤鍵都必須是唯一的。如果您新增具有已使用金鑰的標籤，則新的標籤會覆寫現有金鑰值對。
- 您不能以標籤鍵開頭，aws: 因為此前綴保留供使用 AWS。 AWS 代表您建立以此首碼開頭的標籤，但您無法編輯或刪除它們。
- 標籤鍵的長度必須介於 1 到 128 個 Unicode 字元之間。
- 標籤鍵必須包含下列字元：Unicode 字母、數字、空格以及下列特殊字元：_ . / = + - @。

標籤值限制

- 標籤值的長度必須介於 0 到 255 個 Unicode 字元之間。
- 標籤值可以空白。否則，它們必須包含下列字元：Unicode 字母、數字、空格以及下列任何特殊字元：_ . / = + - @。

使用標記記錄群組 AWS CLI

您可以使用 AWS CLI來新增、列出和移除標籤。如需範例，請參閱下列文件：

create-log-group

建立一個日誌群組。當您建立日誌群組時，您可以選擇性新增標籤。

tag-resource

為指定的 CloudWatch 記錄資源指派一或多個標籤 (鍵值配對)。

list-tags-for-resource

顯示與 CloudWatch 記錄資源關聯的標籤。

untag-resource

從指定的 CloudWatch 記錄資源中移除一或多個標籤。

使用記錄 API 標記記 CloudWatch 錄群組

您可以使用 CloudWatch 記錄 API 新增、列出和移除標籤。如需範例，請參閱下列文件：

CreateLogGroup

建立一個日誌群組。當您建立日誌群組時，您可以選擇性新增標籤。

TagResource

為指定的 CloudWatch 記錄資源指派一或多個標籤 (鍵值配對)。

ListTagsForResource

顯示與 CloudWatch 記錄資源關聯的標籤。

UntagResource

從指定的 CloudWatch 記錄資源中移除一或多個標籤。

使用加密記 CloudWatch 錄檔中的記錄資料 AWS Key Management Service

記錄群組資料一律會在 CloudWatch 記錄檔中加密。根據預設， CloudWatch Logs 會針對靜態記錄資料使用伺服器端加密。您也可以使用 AWS Key Management Service 進行此加密。如果這樣做，則使用密 AWS KMS 鑰完成加密。使用加密 AWS KMS 是在記錄群組層級啟用的，方法是在建立記錄群組時或在記錄群組存在之後將 KMS 金鑰與記錄群組產生關聯。

> **Important**
>
> CloudWatch 記錄現在支援加密內容，使用`kms:EncryptionContext:aws:logs:arn`作為日誌群組的金鑰和 ARN 作為該金鑰的值。如果您有日誌群組已使用 KMS 加密，並希望能限制金鑰與單一帳戶和日誌群組搭配使用，您應該在 IAM 政策中指派包含條件的新 KMS 金鑰。如需詳細資訊，請參閱 AWS KMS 金鑰與加密內容。

建立 KMS 金鑰與日誌群組的關聯後，針對該日誌群組新擷取的所有資料，就會使用此金鑰加密。此資料會在整個保留期間以加密格式儲存。 CloudWatch 記錄檔會在要求時解密此資料。 CloudWatch 每當要求加密資料時，記錄都必須具有 KMS 金鑰的權限。

如果您稍後取消 KMS 金鑰與記錄群組的關聯， CloudWatch Logs 會使用記 CloudWatch 錄預設加密方法加密新擷取的資料。先前使用 KMS 金鑰加密的所有擷取資料都會使用 KMS 金鑰加密。CloudWatch 取消關聯 KMS 金鑰之後，記錄檔仍然可以傳回該資料，因為 CloudWatch 記錄檔仍可繼續參考金鑰。不過，如果金鑰稍後停用，則 CloudWatch 記錄檔將無法讀取使用該金鑰加密的記錄檔。

> **Important**
>
> CloudWatch 記錄僅支援對稱 KMS 金鑰。請勿使用非對稱金鑰來加密日誌群組中的資料。如需詳細資訊，請參閱使用對稱和非對稱金鑰。

限制

- 若要執行下列步驟，您必須擁有下列許可：kms:CreateKey、kms:GetKeyPolicy 和 kms:PutKeyPolicy。
- 在您建立或取消金鑰與日誌群組的關聯後，操作將在 5 分鐘內生效。
- 如果您撤銷對關聯金鑰的 CloudWatch 記錄存取權或刪除關聯的 KMS 金鑰，則無法再擷取 CloudWatch 記錄中的加密資料。
- 您無法使用 CloudWatch 主控台將 KMS 金鑰與記錄群組產生關聯。

步驟 1：建立 AWS KMS 金鑰

若要建立 KMS 金鑰，請使用以下 create-key 命令：

```
aws kms create-key
```

輸出包含金鑰 ID 和金鑰的 Amazon Resource Name (ARN)。下列為範例輸出：

```
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "Enabled": true,
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1478910250.94,
        "Arn": "arn:aws:kms:us-west-2:123456789012:key/6f815f63-e628-448c-8251-
e40cb0d29f59",
        "AWSAccountId": "123456789012",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
```

```
        ]
    }
}
```

步驟 2：設定 KMS 金鑰許可

依預設，所有 AWS KMS 金鑰都是私密金鑰。只有資源擁有者可以使用它來加密和解密資料。然而，資源擁有者可以授與其他使用者和資源存取 KMS 金鑰的許可。透過此步驟，您可以授與 CloudWatch 記錄服務主體使用金鑰的權限。此服務主體必須位於儲存 KMS 金鑰的相同 AWS 區域。

最佳做法是，建議您將 KMS 金鑰的使用限制為僅使用您指定的 AWS 帳戶或記錄群組。

首先，使`policy.json`用下列get-key-policy命令將 KMS 金鑰的預設原則儲存為：

```
aws kms get-key-policy --key-id key-id --policy-name default --output text > ./
policy.json
```

在文字編輯器中開啟 `policy.json` 檔案，並從下列其中一個陳述式中加入區段 (以粗體顯示)。使用逗號從新陳述式中分隔現有陳述式。這些陳述式使用`Condition`區段來增強 AWS KMS 金鑰的安全性。如需詳細資訊，請參閱 AWS KMS 金鑰與加密內容。

此範例中的 `Condition` 區段會將金鑰限制為單一日誌群組 ARN。

```
{
 "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::Your_account_ID:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
```

```
                "kms:Encrypt*",
                "kms:Decrypt*",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                    "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:region:account-
id:log-group:log-group-name"
                }
            }
        }
    ]
}
```

本範例中的 Condition 區段將 AWS KMS 金鑰限用於指定的帳戶，但可用於任何日誌群組。

```
{
    "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::Your_account_ID:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt*",
                "kms:Decrypt*",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
```

```
            ],
            "Resource": "*",
            "Condition": {
                "ArnLike": {
                    "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:region:account-
id:*"
                }
            }
        }
    ]
}
```

最後，使用下列put-key-policy命令新增更新的原則：

```
aws kms put-key-policy --key-id key-id --policy-name default --policy file://
policy.json
```

步驟 3：為 KMS 金鑰與日誌群組建立關聯

您可以在建立日誌群組時或建立完成後，為 KMS 金鑰與日誌群組建立關聯。

若要尋找記錄群組是否已有關聯的 KMS 金鑰，請使用下列describe-log-groups命令：

```
aws logs describe-log-groups --log-group-name-prefix "log-group-name-prefix"
```

如果輸出包含 kmsKeyId 欄位，則日誌群組會與該欄位值所顯示的索引鍵相關聯。

若要在建立日誌群組時與 KMS 金鑰建立關聯

使用 create-log-group 命令，如下所示：

```
aws logs create-log-group --log-group-name my-log-group --kms-key-id "key-arn"
```

若要為 KMS 金鑰與現有的日誌群組建立關聯

使用 associate-kms-key 命令，如下所示：

```
aws logs associate-kms-key --log-group-name my-log-group --kms-key-id "key-arn"
```

步驟 4：取消金鑰與日誌群組的關聯

若要取消與記錄群組相關聯的 KMS 金鑰關聯，請使用下列disassociate-kms-key命令：

```
aws logs disassociate-kms-key --log-group-name my-log-group
```

AWS KMS 金鑰與加密內容

為了增強 AWS Key Management Service 金鑰和加密記錄群組的安全性， CloudWatch 記錄檔現在會將記錄群組 ARN 放置為用來加密記錄資料的加密內容的一部分。加密內容是一組做為額外驗證資料的索引鍵/值組。加密內容可讓您使用 IAM 政策條件，依 AWS 帳戶和日誌群組限制對 AWS KMS 金鑰的存取。如需詳細資訊，請參閱加密內容和 IAM JSON 政策元素：Condition。

建議您針對每個加密的日誌群組使用不同的 KMS 金鑰。

如果您有先前加密的日誌群組，但現在希望將日誌群組變更為使用只適用於該日誌群組的新 KMS 金鑰，請依照下列步驟執行。

透過政策將金鑰設為特定日誌群組專用，將加密日誌群組轉換為使用 KMS 金鑰

1. 請輸入下列指令，以尋找日誌群組目前金鑰的 ARN：

   ```
   aws logs describe-log-groups
   ```

   輸出包括這一行。記下 ARN。步驟 7 中會用到。

   ```
   ...
   "kmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/01234567-89ab-
   cdef-0123-456789abcdef"
   ...
   ```

2. 輸入以下命令來建立新的 KMS 金鑰：

   ```
   aws kms create-key
   ```

3. 輸入下列命令，將新金鑰的政策儲存至 policy.json 檔案：

   ```
   aws kms get-key-policy --key-id new-key-id --policy-name default --output text > ./
   policy.json
   ```

4. 使用文字編輯器來開啟 policy.json，並將 Condition 運算式新增至政策：

   ```
   {
       "Version": "2012-10-17",
   ```

```
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::ACCOUNT-ID:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt*",
                "kms:Decrypt*",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
            "Condition": {
                "ArnLike": {
                    "kms:EncryptionContext:aws:logs:arn":
 "arn:aws:logs:REGION:ACCOUNT-ID:log-
group:LOG-GROUP-NAME"
                }
            }
        }
    ]
}
```

5. 輸入下列命令，將更新的政策新增至新的 KMS 金鑰：

```
aws kms put-key-policy --key-id new-key-ARN --policy-name default --policy file://
policy.json
```

6. 輸入下列命令，將政策與日誌群組建立關聯：

```
aws logs associate-kms-key --log-group-name my-log-group --kms-key-id new-key-ARN
```

CloudWatch 記錄現在會使用新金鑰加密所有新資料。

7. 接下來，撤銷舊金鑰的所有權限，除了 Decrypt 以外。首先，輸入下列命令以擷取舊政策：

```
aws kms get-key-policy --key-id old-key-ARN --policy-name default --output text
 > ./policy.json
```

8. 使用文字編輯器來開啟 policy.json，並移除 Action 清單中的所有值，除了 kms:Decrypt* 以外

```
{
    "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::Your_account_ID:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
                "kms:Decrypt*"
            ],
            "Resource": "*"
        }
    ]
}
```

9. 輸入下列命令，將更新的政策新增至舊金鑰：

```
aws kms put-key-policy --key-id old-key-ARN --policy-name default --policy file://
policy.json
```

使用遮罩功能協助保護敏感日誌資料

您可以使用記錄群組資料保護原則，協助保護 CloudWatch 記錄所擷取的敏感資料。這些政策可讓您稽核和遮罩出現在帳戶中日誌群組擷取之日誌事件中的敏感資料。

當您建立資料保護原則時，依預設，會在所有輸出點 (包括 CloudWatch Logs Insights、指標篩選器和訂閱篩選器) 上遮罩符合您選取之資料識別碼的敏感資料。只有具有 `logs:Unmask` IAM 許可的使用者才能檢視未遮罩的資料。

您可以為帳戶中的所有日誌群組建立資料保護政策，也可以為個別日誌群組建立資料保護政策。當您為整個帳戶建立政策時，它會套用至現有的日誌群組和未來建立的日誌群組。

如果您為整個帳戶建立資料保護政策，並且也為單一日誌群組建立政策，則這兩個政策都會套用至該日誌群組。在任一政策中指定的所有受管資料識別符都會在該日誌群組中進行稽核和遮罩。

> **Note**
>
> 只有標準記錄類別中的記錄群組才支援遮罩敏感資料。如果您為帳戶中的所有記錄群組建立資料保護政策，它只會套用至標準記錄類別中的記錄群組。如需記錄類別的詳細資訊，請參閱日誌類。

每個日誌群組只能有一個日誌群組層級資料保護政策，但該政策可以指定許多受管資料識別符來稽核和遮罩。資料保護政策的限制為 30,720 個字元。

> **Important**
>
> 將敏感資料擷取至日誌群組時，系統會偵測這些資料並加以遮罩。系統不會遮罩在您設定資料保護政策之前擷取至日誌群組的日誌事件。

CloudWatch Logs 支援許多受管資料識別碼，這些識別碼提供預先設定的資料類型，您可以選擇保護財務資料、個人健康資訊 (PHI) 和個人識別資訊 (PII)。 CloudWatch 日誌資料保護可讓您利用模式比對和機器學習模型來偵測機密資料。對於某些類型的託管數據標識符，檢測還取決於找到與敏感數據相鄰的某些關鍵字。您還可以使用自定義數據標識符來創建根據您的特定用例量身定制的數據標識符。

CloudWatch 當偵測到符合您選取的資料識別碼的敏感資料時，就會發出指標。這是LogEventsWithFindings量度，會在 AWS/ 記錄檔命名空間中發出。您可以使用此量度建立

CloudWatch 警示，也可以在圖形和儀表板中將其視覺化。資料保護發出的指標是付費指標，但在此免費提供。如需有關 CloudWatch 記錄檔傳送至的指標的詳細資訊 CloudWatch，請參閱使用 CloudWatch 指標監控。

每個受管理資料識別碼都是為了偵測特定類型的敏感資料而設計，例如特定國家或地區的信用卡號碼、AWS 秘密存取金鑰或護照號碼。建立資料保護政策時，您可以將 CloudWatch Logs 設定為使用這些識別符，來分析由日誌群組擷取的日誌，並在偵測到日誌時採取動作。

CloudWatch 記錄檔資料安全防護可以使用受管資料識別碼來偵測下列類別的敏感資料：

- 認證，例如私密金鑰或 AWS 秘密存取金鑰
- 財務資訊，例如信用卡號碼。
- 個人身分識別資訊 (PII)，例如駕照或社會安全號碼
- 受保護醫療資訊 (PHI)，例如健康保險或醫療識別號碼
- 裝置識別符，例如 IP 地址或 MAC 地址

如需有關可保護之資料類型的詳細資訊，請參閱您可以保護的資料類型。

內容

了解資料保護政策

主題

什麼是資料保護政策？

CloudWatch 記錄檔會使用資料保護政策來選取您要掃描的敏感資料，以及您要採取的動作來保護該資料。若要選取感興趣的敏感資料，請使用資料識別碼。 CloudWatch 記錄資料安全防護，然後使用機器學習和病毒碼比對來偵測機密資料。若要根據找到的資料識別符採取行動，您可以定義稽核和去識別化操作。這些操作可讓您記錄找到 (或未找到) 的敏感資料，並在檢視日誌事件時遮罩敏感資料。

資料保護政策的結構如何？

如下圖所示，資料保護政策文件包含以下元素：

- 在文件最上方選用的整體政策資訊
- 定義稽核和去識別動作的一條陳述式

每個 CloudWatch 記錄檔記錄群組只能定義一個資料保護原則。資料保護政策可以有一或多個拒絕或去識別化陳述式，但只能有一個稽核陳述式。

資料保護政策的 JSON 屬性

資料保護政策需要下列基本政策資訊才能識別：

- Name - 政策名稱。
- Description (選用) - 政策描述。
- Version - 政策語言版本。目前版本是 2021-06-01。
- Statement - 指定資料保護政策動作的陳述式清單。

```
{
  "Name": "CloudWatchLogs-PersonalInformation-Protection",
  "Description": "Protect basic types of sensitive data",
  "Version": "2021-06-01",
  "Statement": [
        ...
  ]
}
```

政策陳述式的 JSON 屬性

政策陳述式會設定資料保護操作的偵測內容。

- Sid (選用) - 陳述式識別符。
- DataIdentifier— CloudWatch 防護記錄應掃描的敏感資料。例如，姓名、地址或電話號碼。
- 「操作」— 后續操作 (「稽核」或「取消標識」)。 CloudWatch 記錄檔會在找到敏感資料時執行這些動作。

```
{
  ...
  "Statement": [
    {
      "Sid": "audit-policy",
      "DataIdentifier": [
        "arn:aws:dataprotection::aws:data-identifier/Address"
      ],
      "Operation": {
        "Audit": {
          "FindingsDestination": {}
        }
      }
    },
```

政策陳述式操作的 JSON 屬性

政策陳述式會設定下列其中一項資料保護操作。

- 稽核 – 發出指標和問題清單報告，而不會中斷日誌記錄。符合的字串會增加 CloudWatch 日誌發佈至 AWS/Logs 命名空間的量LogEventsWithFindings度。 CloudWatch您可以使用這些指標建立警示。

  如需問題清單報告的範例，請參閱 稽核問題清單報告。

  如需有關 CloudWatch 記錄檔傳送至的指標的詳細資訊 CloudWatch，請參閱使用 CloudWatch 指標監控。
- 去識別 – 遮罩敏感資料，而不會中斷日誌記錄。

必須具備 IAM 許可才能建立或使用資料保護政策

若要能夠使用日誌群組的資料保護政策，您必須具有下表所示的特定許可。帳戶層級的資料保護政策和套用至單一日誌群組的資料保護政策的許可有所不同。

帳戶層級資料保護政策所需的許可

> **Note**
>
> 如果您要在 Lambda 函數內執行這些作業，Lambda 執行角色和許可界限也必須包含下列許可。

作業	需要 IAM 許可	資源
建立不具有稽核目的地的資料保護政策	`logs:PutAccountPolicy`	*
	`logs:PutDataProtectionPolicy`	*
使用 CloudWatch 記錄檔做為稽核目標建立資料保護策略	`logs:PutAccountPolicy`	*
	`logs:PutDataProtectionPolicy`	*
	`logs:CreateLogDelivery`	*
	`logs:PutResourcePolicy`	*
	`logs:DescribeResourcePolicies`	*
	`logs:DescribeLogGroups`	*

作業	需要 IAM 許可	資源
使用 Firehose 作為稽核目標建立資料保護政策	logs:PutAccountPolicy	*
	logs:PutDataProtectionPolicy	*
	logs:CreateLogDelivery	*
	firehose:TagDeliveryStream	arn:aws:logs:::deliverystream/ *YOUR_DELIVERY_STREAM*
建立以 Amazon S3 為稽核目的地的資料保護政策	logs:PutAccountPolicy	*
	logs:PutDataProtectionPolicy	*
	logs:CreateLogDelivery	*
	s3:GetBucketPolicy	arn:aws:s3::: *YOUR_BUCKET*
	s3:PutBucketPolicy	arn:aws:s3::: *YOUR_BUCKET*
取消遮罩指定日誌群組中的遮罩日誌事件	logs:Unmask	arn:aws:logs:::log-group:*
檢視現有的資料保護政策	logs:GetDataProtectionPolicy	*
刪除資料保護政策	logs:DeleteAccountPolicy	*

作業	需要 IAM 許可	資源
	logs:DeleteDataProtectionPolicy	*

如果有任何資料保護稽核日誌已傳送至目的地，則也將日誌傳送至相同目的地的其他策略僅需要 logs:PutDataProtectionPolicy 和 logs:CreateLogDelivery 許可。

單一日誌群組之資料保護政策所需的許可

> **Note**
>
> 如果您要在 Lambda 函數內執行這些作業，Lambda 執行角色和許可界限也必須包含下列許可。

作業	需要 IAM 許可	資源
建立不具有稽核目的地的資料保護政策	logs:PutDataProtectionPolicy	arn:aws:logs:::log-group: *YOUR_LOG_GROUP* :*
使用 CloudWatch 記錄檔做為稽核目標建立資料保護策略	logs:PutDataProtectionPolicy logs:CreateLogDelivery logs:PutResourcePolicy logs:DescribeResourcePolicies logs:DescribeLogGroups	arn:aws:logs:::log-group: *YOUR_LOG_GROUP* :* * * * *

作業	需要 IAM 許可	資源
使用 Firehose 作為稽核目標建立資料保護政策	logs:PutDataProtectionPolicy logs:CreateLogDelivery firehose:TagDeliveryStream	arn:aws:logs:::log-group: *YOUR_LOG_GROUP* :* * arn:aws:logs:::deliverystream/ *YOUR_DELIVERY_STREAM*
建立以 Amazon S3 為稽核目的地的資料保護政策	logs:PutDataProtectionPolicy logs:CreateLogDelivery s3:GetBucketPolicy s3:PutBucketPolicy	arn:aws:logs:::log-group: *YOUR_LOG_GROUP* :* * arn:aws:s3::: *YOUR_BUCKET* arn:aws:s3::: *YOUR_BUCKET*
取消遮罩已遮罩的日誌事件	logs:Unmask	arn:aws:logs:::log-group: *YOUR_LOG_GROUP* :*
檢視現有的資料保護政策	logs:GetDataProtectionPolicy	arn:aws:logs:::log-group: *YOUR_LOG_GROUP* :*
刪除資料保護政策	logs:DeleteDataProtectionPolicy	arn:aws:logs:::log-group: *YOUR_LOG_GROUP* :*

如果有任何資料保護稽核日誌已傳送至目的地，則也將日誌傳送至相同目的地的其他策略僅需要 `logs:PutDataProtectionPolicy` 和 `logs:CreateLogDelivery` 許可。

資料保護政策範例

下列政策範例可讓使用者建立、檢視及刪除資料保護政策，這些政策可將稽核調查結果傳送至全部三種稽核目的地類型。它不允許使用者檢視未遮罩的資料。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "YOUR_SID_1",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:PutResourcePolicy",
                "logs:DescribeLogGroups",
                "logs:DescribeResourcePolicies"
            ],
            "Resource": "*"
        },
        {
            "Sid": "YOUR_SID_2",
            "Effect": "Allow",
            "Action": [
                "logs:GetDataProtectionPolicy",
                "logs:DeleteDataProtectionPolicy",
                "logs:PutDataProtectionPolicy",
                "s3:PutBucketPolicy",
                "firehose:TagDeliveryStream",
                "s3:GetBucketPolicy"
            ],
            "Resource": [
                "arn:aws:firehose:::deliverystream/YOUR_DELIVERY_STREAM",
                "arn:aws:s3:::YOUR_BUCKET",
                "arn:aws:logs:::log-group:YOUR_LOG_GROUP:*"
            ]
        }
    ]
}
```

建立帳戶層級資料保護政策

您可以使用 CloudWatch 記錄主控台或 AWS CLI 命令建立資料保護政策，以遮罩帳戶中所有記錄群組的機密資料。這樣做會影響目前的日誌群組和您未來建立的日誌群組。

> **Important**
>
> 將敏感資料擷取至日誌群組時，系統會偵測這些資料並加以遮罩。系統不會遮罩在您設定資料保護政策之前擷取至日誌群組的日誌事件。

主題

主控台

使用主控台建立帳戶層級資料保護政策

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇設定。它位於清單底部附近。
3. 選擇 Logs (日誌) 索引標籤。
4. 選擇設定。
5. 針對受管理的資料識別碼，選取您要稽核和遮罩所有記錄群組的資料類型。您可以在選取方塊中輸入內容以尋找所需的識別符。

   我們建議您只選取與日誌資料和業務相關的資料識別符。選擇的資料類型過多可能會導致誤報。

   如需有關可保護的資料類型的詳細資訊，請參閱您可以保護的資料類型。
6. (選擇性) 如果您想要使用自訂資料識別碼來稽核和遮罩其他類型的資料，請選擇 [新增自訂資料識別碼]。然後輸入資料類型的名稱和規則運算式，以便在記錄事件中搜尋該類型的資料。如需詳細資訊，請參閱 自訂資料識別符。

   單一資料保護政策最多可包含 10 個自訂資料識別碼。定義自訂資料識別碼的每個規則運算式必須少於 200 個字元。
7. (選用) 選擇向其傳送稽核問題清單的一或多個服務。即使您選擇不將稽核問題清單傳送至任何這些服務，系統仍然會遮罩您選取的敏感資料類型。

8. 選擇 Activate data protection (啟動資料保護)。

AWS CLI

若要使用建 AWS CLI 立資料安全防護原則

1. 使用文字編輯器來建立名為 DataProtectionPolicy.json 的政策檔案。如需有關政策語法的資訊，請參閱下一節。
2. 輸入以下命令：

```
aws logs put-account-policy \
--policy-name TEST_POLICY --policy-type "DATA_PROTECTION_POLICY" \
--policy-document file://policy.json \
--scope "ALL" \
--region us-west-2
```

AWS CLI 或 API 作業的資料保護政策語法

當您建立要在 AWS CLI 命令或 API 作業中使用的 JSON 資料保護政策時，原則必須包含兩個 JSON 區塊：

- 第一個區塊必須同時包含 DataIdentifer 陣列和具有 Audit 動作的 Operation 屬性。DataIdentifer 陣列會列出您要遮罩的敏感資料類型。如需所有可用選項的詳細資訊，請參閱您可以保護的資料類型。

  具有 Audit 動作的 Operation 屬性為必要項目，如此才能找到敏感資料術語。此 Audit 動作必須包含 FindingsDestination 物件。您可以選擇使用此 FindingsDestination 物件，來列出要向其傳送稽核問題清單報告的一或多個目的地。如果您指定日誌群組、Amazon 資料 Firehose 串流和 S3 儲存貯體等目的地，它們必須已經存在。如需稽核問題清單報告的範例，請參閱 稽核問題清單報告。
- 第二個區塊必須同時包含 DataIdentifer 陣列和具有 Deidentify 動作的 Operation 屬性。DataIdentifer 陣列必須與政策第一個區塊中的 DataIdentifer 陣列完全一致。

  具有 Deidentify 動作的 Operation 屬性用於實際遮罩資料，其必須包含 "MaskConfig": {} 物件。 "MaskConfig": {} 物件必須是空的。

以下是僅使用受管理資料識別碼的資料保護政策範例。此政策會遮罩電子郵件地址和美國駕照。

如需有關指定自訂資料識別碼之策略的資訊，請參閱在您的資料保護政策中使用自訂資料識別符。

```
{
    "Name": "data-protection-policy",
    "Description": "test description",
    "Version": "2021-06-01",
    "Statement": [{
            "Sid": "audit-policy",
            "DataIdentifier": [
                "arn:aws:dataprotection::aws:data-identifier/EmailAddress",
                "arn:aws:dataprotection::aws:data-identifier/DriversLicense-US"
            ],
            "Operation": {
                "Audit": {
                    "FindingsDestination": {
                        "CloudWatchLogs": {
                            "LogGroup": "EXISTING_LOG_GROUP_IN_YOUR_ACCOUNT,"
                        },
                        "Firehose": {
                            "DeliveryStream": "EXISTING_STREAM_IN_YOUR_ACCOUNT"
                        },
                        "S3": {
                            "Bucket": "EXISTING_BUCKET"
                        }
                    }
                }
            }
        },
        {
            "Sid": "redact-policy",
            "DataIdentifier": [
                "arn:aws:dataprotection::aws:data-identifier/EmailAddress",
                "arn:aws:dataprotection::aws:data-identifier/DriversLicense-US"
            ],
            "Operation": {
                "Deidentify": {
                    "MaskConfig": {}
                }
            }
        }
    ]
}
```

建立單一日誌群組的資料保護政策

您可以使用 CloudWatch 記錄主控台或 AWS CLI 命令來建立資料安全防護原則，以遮罩機密資料。

您可以為每個日誌群組指派一個資料保護政策。每個資料保護政策都可以稽核多種類型的資訊。每個資料保護政策都可以包含一份稽核聲明。

主題

主控台

若要使用主控台建立資料保護政策

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中依序選擇 Logs (日誌)、Log groups (日誌群組)。
3. 選擇日誌群組的名稱。
4. 選擇 Actions (動作)、Create data protection policy (建立資料保護政策)。
5. 針對受管理的資料識別碼，選取您要在此記錄群組中稽核和遮罩的資料類型。您可以在選取方塊中輸入內容以尋找所需的識別符。

   我們建議您只選取與日誌資料和業務相關的資料識別符。選擇的資料類型過多可能會導致誤報。

   如需有關您可以使用受管資料識別碼來保護哪些資料類型的詳細資訊，請參閱您可以保護的資料類型。
6. (選擇性) 如果您想要使用自訂資料識別碼來稽核和遮罩其他類型的資料，請選擇 [新增自訂資料識別碼]。然後輸入資料類型的名稱和規則運算式，以便在記錄事件中搜尋該類型的資料。如需詳細資訊，請參閱 自訂資料識別符。

   單一資料保護政策最多可包含 10 個自訂資料識別碼。定義自訂資料識別碼的每個規則運算式必須少於 200 個字元。
7. (選用) 選擇向其傳送稽核問題清單的一或多個服務。即使您選擇不將稽核問題清單傳送至任何這些服務，系統仍然會遮罩您選取的敏感資料類型。
8. 選擇 Activate data protection (啟動資料保護)。

AWS CLI

若要使用建 AWS CLI 立資料安全防護原則

1. 使用文字編輯器來建立名為 DataProtectionPolicy.json 的政策檔案。如需有關政策語法的資訊，請參閱下一節。
2. 輸入以下命令：

```
aws logs put-data-protection-policy --log-group-identifier "my-log-group" --policy-
document file:///Path/DataProtectionPolicy.json --region us-west-2
```

AWS CLI 或 API 作業的資料保護政策語法

當您建立要在 AWS CLI 命令或 API 作業中使用的 JSON 資料保護政策時，原則必須包含兩個 JSON 區塊：

- 第一個區塊必須同時包含 DataIdentifer 陣列和具有 Audit 動作的 Operation 屬性。DataIdentifer 陣列會列出您要遮罩的敏感資料類型。如需所有可用選項的詳細資訊，請參閱您可以保護的資料類型。

  具有 Audit 動作的 Operation 屬性為必要項目，如此才能找到敏感資料術語。此 Audit 動作必須包含 FindingsDestination 物件。您可以選擇使用此 FindingsDestination 物件，來列出要向其傳送稽核問題清單報告的一或多個目的地。如果您指定日誌群組、Amazon 資料 Firehose 串流和 S3 儲存貯體等目的地，它們必須已經存在。如需稽核問題清單報告的範例，請參閱 稽核問題清單報告。
- 第二個區塊必須同時包含 DataIdentifer 陣列和具有 Deidentify 動作的 Operation 屬性。DataIdentifer 陣列必須與政策第一個區塊中的 DataIdentifer 陣列完全一致。

  具有 Deidentify 動作的 Operation 屬性用於實際遮罩資料，其必須包含 "MaskConfig": {} 物件。 "MaskConfig": {} 物件必須是空的。

以下是遮罩電子郵件地址和美國駕照的資料保護政策範例。

```
{
    "Name": "data-protection-policy",
    "Description": "test description",
    "Version": "2021-06-01",
    "Statement": [{
```

```
            "Sid": "audit-policy",
            "DataIdentifier": [
                "arn:aws:dataprotection::aws:data-identifier/EmailAddress",
                "arn:aws:dataprotection::aws:data-identifier/DriversLicense-US"
            ],
            "Operation": {
                "Audit": {
                    "FindingsDestination": {
                        "CloudWatchLogs": {
                            "LogGroup": "EXISTING_LOG_GROUP_IN_YOUR_ACCOUNT,"
                        },
                        "Firehose": {
                            "DeliveryStream": "EXISTING_STREAM_IN_YOUR_ACCOUNT"
                        },
                        "S3": {
                            "Bucket": "EXISTING_BUCKET"
                        }
                    }
                }
            }
        },
        {
            "Sid": "redact-policy",
            "DataIdentifier": [
                "arn:aws:dataprotection::aws:data-identifier/EmailAddress",
                "arn:aws:dataprotection::aws:data-identifier/DriversLicense-US"
            ],
            "Operation": {
                "Deidentify": {
                    "MaskConfig": {}
                }
            }
        }
    ]
}
```

檢視未遮罩的資料

若要檢視未遮罩的資料，使用者必須具有 `logs:Unmask` 許可。具有此許可的使用者可透過以下方式查看未遮罩的資料：

- 檢視日誌串流中的事件時，選擇 Display (顯示)、Unmask (解除遮罩)。

- 使用包含取消遮罩 (@message) 命令的 CloudWatch 日誌見解查詢。下列範例查詢會以未遮罩的方式顯示串流中最近 20 個日誌事件：

```
fields @timestamp, @message, unmask(@message)
| sort @timestamp desc
| limit 20
```

  如需有關 CloudWatch 記錄檔見解命令的詳細資訊，請參閱CloudWatch 日誌見解查詢語法。

- 搭配unmask參數使用 GetLogEvents或 FilterLogEvents作業。

該CloudWatchLogsFullAccess策略包括logs:Unmask權限。若logs:Unmask要授予沒有使用者 CloudWatchLogsFullAccess，您可以將自訂 IAM 政策附加到該使用者。如需詳細資訊，請參閱新增許可到使用者 (主控台)。

稽核問題清單報告

如果您將 CloudWatch 日誌資料保護稽核政策設定為將稽核報告寫入 CloudWatch 日誌、Amazon S3 或 Firehose，則這些發現項目報告與下列範例類似。 CloudWatch 記錄檔會為每個包含敏感資料的記錄事件寫入一份發現項目報告。

```
{
    "auditTimestamp": "2023-01-23T21:11:20Z",
    "resourceArn": "arn:aws:logs:us-west-2:111122223333:log-group:/aws/lambda/
MyLogGroup:*",
    "dataIdentifiers": [
        {
            "name": "EmailAddress",
            "count": 2,
            "detections": [
                {
                    "start": 13,
                    "end": 26
                },
{
                    "start": 30,
                    "end": 43
                }
            ]
        }
    ]
```

```
}
```

報告中的欄位如下所示：

- `resourceArn` 欄位會顯示在其中找到敏感資料的日誌群組。
- `dataIdentifiers` 物件會顯示您正在稽核的某種敏感資料的問題清單相關資訊。
- `name` 欄位可識別此區段所報告的敏感資料類型。
- `count` 欄位會顯示此種敏感資料在日誌事件中出現的次數。
- `start` 和 `end` 欄位會依字元計數顯示日誌事件中每次出現敏感資料的位置。

上一個範例顯示在一個日誌事件中尋找兩個電子郵件地址的報告。第一個電子郵件地址從日誌事件的第 13 個字元開始，並在第 26 個字元處結束。第二個電子郵件地址從第 30 個字元一直到第 43 個字元。即使此日誌事件有兩個電子郵件地址，`LogEventsWithFindings` 指標的值也只會遞增 1，因為該指標會對包含敏感資料的日誌事件計數，而非計算敏感資料的出現次數。

將稽核發現項目傳送至受保護的值區的必要金鑰政策 AWS KMS

透過啟用採用 Amazon S3 受管金鑰 (SSE-S3) 的伺服器端加密或採用 KMS 金鑰 (SSE-KMS) 的伺服器端加密，您可以保護 Amazon S3 儲存貯體中的資料。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的使用伺服器端加密保護資料。

如果您將稽核調查結果發送至以 SSE-S3 保護的儲存貯體，則不需要其他組態。Amazon S3 會處理加密金鑰。

如果您將稽核調查結果發送至以 SSE-KMS 保護的儲存貯體，您必須更新 KMS 金鑰的金鑰政策，讓日誌傳遞帳戶能夠寫入您的 S3 儲存貯體。如需與 SSE-KMS 搭配使用所需金鑰政策的詳細資訊，請參閱 Amazon CloudWatch 日誌使用者指南Amazon S3中的。

您可以保護的資料類型

本節包含您可以在 CloudWatch 記錄檔資料保護政策中保護之資料類型的相關資訊。 CloudWatch 記錄受管資料識別碼提供預先設定的資料類型，以保護財務資料、個人健康資訊 (PHI) 和個人識別資訊 (PII)。您還可以使用自定義數據標識符來創建根據您的特定用例量身定制的數據標識符。

內容

- 裝置識別符
  - 裝置資料類型的資料識別符 ARN
- 財務資訊
  - 財務資料類型的資料識別符 ARN
- 受保護醫療資訊 (PHI)
  - 受保護醫療資訊 (PHI) 資料類型的資料識別符 ARN
- 個人身分識別資訊 (PII)
  - 駕照識別號碼的關鍵字
  - 國民身分證號碼的關鍵字
  - 護照號碼的關鍵字
  - 納稅識別號碼及參考號碼的關鍵字
  - 個人身分識別資訊 (PII) 的資料識別符 ARN
- 自訂資料識別符
  - 什麼是 SNS 自訂資料識別符？
  - 自訂資料識別符的限制
  - 在主控台中使用自訂資料識別碼
  - 在您的資料保護政策中使用自訂資料識別符

CloudWatch 記錄敏感資料類型的受管理資料識別碼

本節包含您可以使用受管資料識別碼保護的資料類型，以及哪些國家和地區與每種資料類型相關的資訊。

對於某些類型的機密資料，「CloudWatch 記錄檔資料保護」會掃描資料鄰近的關鍵字，並且只有在找到該關鍵字時才尋找相符項目。如果關鍵字必須與特定類型的資料相鄰，則關鍵字通常必須在 30 個字元以內 (包含在內) 資料。

如果關鍵字包含空格，則 CloudWatch 記錄資料保護會自動比對缺少空格或包含底線 (_) 或連字號 (-) 而非空格的關鍵字變體。在某些情況下， CloudWatch Logs 還會展開或縮寫關鍵字，以解決關鍵字的常見變化。

下表列出了 CloudWatch Logs 可以使用受管理資料識別碼偵測的認證類型、裝置、財務、醫療和受保護的健康資訊 (PHI)。這些是某些資料類型的個人身分識別資訊 (PII) 等資料。

支援的識別符 (與語言和區域無關)

識別符	類別
Address	個人
AwsSecretKey	登入資料
CreditCardExpiration	金融
CreditCardNumber	金融
CreditCardSecurityCode	金融
EmailAddress	個人
IpAddress	個人
LatLong	個人
Name	個人
OpenSshPrivateKey	登入資料
PgpPrivateKey	登入資料
PkcsPrivateKey	登入資料
PuttyPrivateKey	登入資料
VehicleIdentificationNumber	個人

與區域相關的資料識別符需要包含識別符名稱、一個連字號，以及兩個字母 (ISO 3166-1 alpha-2) 代碼。例如 DriversLicense-US。

支援的識別符 (必須包含兩個字母的國家或地區碼)

識別符	類別	國家/地區與語言
BankAccountNumber	金融	DE、ES、FR、GB、IT
CepCode	個人	BR

識別符	類別	國家/地區與語言
Cnpj	個人	BR
CpfCode	個人	BR
DriversLicense	個人	AT、AU、BE、 BG、CA、CY、 CZ、DE、DK、EE、ES、FI、 FR、GB、GR、 HR、HU、IE、IT、LT、LU、 LV、MT、NL、 PL、PT、RO、SE、SI、SK、 US
DrugEnforcementAgencyNumber	醫療保健	US
ElectoralRollNumber	個人	GB
HealthInsuranceCardNumber	醫療保健	歐盟
HealthInsuranceClaimNumber	醫療保健	US
HealthInsuranceNumber	醫療保健	法國
HealthcareProcedureCode	醫療保健	US
IndividualTaxIdentificationNumber	個人	美國
InseeCode	個人	法國
MedicareBeneficiaryNumber	醫療保健	US
NationalDrugCode	醫療保健	US
NationalIdentificationNumber	個人	DE、ES、IT
NationalInsuranceNumber	個人	GB

識別符	類別	國家/地區與語言
NationalProviderId	醫療保健	US
NhsNumber	醫療保健	GB
NieNumber	個人	ES
NifNumber	個人	ES
PassportNumber	個人	CA、DE、ES、FR、GB、IT、US
PermanentResidenceNumber	個人	CA
PersonalHealthNumber	醫療保健	CA
PhoneNumber	個人	BR、DE、ES、FR、GB、IT、US
PostalCode	個人	CA
RgNumber	個人	BR
SocialInsuranceNumber	個人	CA
Ssn	個人	ES、US
TaxId	個人	DE、ES、FR、GB
ZipCode	個人	美國

登入資料

CloudWatch 記錄檔資料保護可以找到下列類型的認證。

資料類型	資料識別符 ID	必要的關鍵字	國家和地區
AWS 秘密訪問密鑰	AwsSecretKey	aws_secret_access_key , credentials , secret access key, secret key, set-awscredential	全部
OpenSSH 私密金鑰	OpenSSHPrivateKey	無	全部
PGP 私密金鑰	PgpPrivateKey	無	全部
Pkcs 私有金鑰	PkcsPrivateKey	無	全部
PuTTY 私密金鑰	PuttyPrivateKey	無	全部

憑證資料類型的資料識別符 ARN

以下列出您可新增至資料保護政策的資料識別符 Amazon Resource Name (ARN)。

憑證資料識別符 ARN
arn:aws:dataprotection::aws:data-identifier/AwsSecretKey
arn:aws:dataprotection::aws:data-identifier/OpenSshPrivateKey
arn:aws:dataprotection::aws:data-identifier/PgpPrivateKey
arn:aws:dataprotection::aws:data-identifier/PkcsPrivateKey
arn:aws:dataprotection::aws:data-identifier/PuttyPrivateKey

裝置識別符

CloudWatch 記錄檔資料保護可以找到下列類型的裝置識別碼。

資料類型	資料識別符 ID	必要的關鍵字	國家和地區
IP 地址	`IpAddress`	無	全部

裝置資料類型的資料識別符 ARN

以下列出您可新增至資料保護政策的資料識別符 Amazon Resource Name (ARN)。

裝置資料識別符 ARN
`arn:aws:dataprotection::aws:data-identifier/IpAddress`

財務資訊

CloudWatch 記錄檔資料保護可以找到下列類型的財務資訊。

如果您設定資料保護政策，則無論 CloudWatch 記錄群組位於何種地理位置，記錄檔都會掃描您指定的資料識別碼。此資料表中國家與地區資料欄中的資訊指出，是否必須在資料識別符後附加兩個字母的國家/地區碼，以偵測這些國家和地區的相應關鍵字。

資料類型	資料識別符 ID	必要的關鍵字	國家和地區	備註
銀行帳戶號碼	`BankAccountNumber`	是。不同的關鍵字適用於不同的國家/地區。如需詳細資訊，請參閱本節後文的銀行帳戶號碼的關鍵字資料表。	法國、德國、義大利、西班牙、英國	包括國際銀行帳戶號碼 (IBAN)，此號碼最多由 34 個英數

資料類型	資料識別符 ID	必要的關鍵字	國家和地區	備註
				字元組成，包括國家/地區碼等元素。
信用卡到期日	CreditCardExpiration	exp d, exp m, exp y, expiration , expiry	全部	

資料類型	資料識別符 ID	必要的關鍵字	國家和地區	備註
信用卡號碼	CreditCardNumber	account number, american express, amex, bank card, card, card number, card num, cc #, ccn, check card, credit, credit card#, dankort, debit, debit card, diners club, discover, electron, japanese card bureau, jcb, mastercard , mc, pan, payment account number, payment card number, pcn, union pay, visa	全部	檢測要求數據是符合 Luhn 檢查公式的 13—19 位數序列，並為以下任何類型的信用卡使用標準卡號前綴：美國運通卡，Dankort，晚餐俱樂部，發現，電子，日本卡局（JCB），萬事達卡和 Visa。UnionPay

資料類型	資料識別符 ID	必要的關鍵字	國家和地區	備註
信用卡驗證碼	CreditCardSecurityCode	card id, card identification code, card identification number , card security code, card validation code , card validation number , card verification data , card verification value, cvc, cvc2, cvv, cvv2, elo verification code	全部	

銀行帳戶號碼的關鍵字

使用下列關鍵字來偵測最多由 34 個英數字元 (包括國家/地區碼等元素) 組成的國際銀行帳戶號碼 (IBAN)。

Country	關鍵字
法國	account code, account number, accountno# , accountnumber# , bban, code bancaire, compte bancaire, customer account id, customer account number, customer bank account id, iban, numéro de compte
德國	account code, account number, accountno# , accountnumber# , bankleitzahl , bban, customer account id, customer account number, customer bank account id, geheimzahl , iban, kartennummer , kontonummer , kreditkartennummer , sepa
義大利	account code, account number, accountno# , accountnumber# , bban, codice bancario, conto bancario, customer account id,

Country	關鍵字
	customer account number, customer bank account id, iban, numero di conto
西班牙	account code, account number, accountno# , accountnumber# , bban, código cuenta, código cuenta bancaria, cuenta cliente id, customer account ID, customer account number, customer bank account id, iban, número cuenta bancaria cliente, número cuenta cliente
英國	account code, account number, accountno# , accountnumber# , bban, customer account ID, customer account number, customer bank account id, iban, sepa
美國	bank account, bank acct, checking account, checking acct, deposit account, deposit acct, savings account, savings acct, chequing account, chequing acct

CloudWatch 日誌不會報告以下順序的發生，信用卡發卡機構已保留用於公開測試。

```
122000000000003, 2222405343248877, 2222990905257051, 2223007648726984,
 2223577120017656,
30569309025904, 34343434343434, 3528000700000000, 3530111333300000, 3566002020360505,
 36148900647913,
36700102000000, 371449635398431, 378282246310005, 378734493671000, 38520000023237,
 4012888888881881,
4111111111111111, 4222222222222, 4444333322221111, 4462030000000000, 4484070000000000,
 4911830000000,
4917300800000000, 4917610000000000, 4917610000000000003, 5019717010103742,
 5105105105105100,
5111010030175156, 5185540810000019, 5200828282828210, 5204230080000017,
 5204740009900014, 5420923878724339,
5454545454545454, 5455330760000018, 5506900490000436, 5506900490000444,
 5506900510000234, 5506920809243667,
5506922400634930, 5506927427317625, 5553042241984105, 5555553753048194,
 5555555555554444, 5610591081018250,
6011000990139424, 6011000400000000, 6011111111111117, 630490017740292441,
 630495060000000000,
6331101999990016, 6759649826438453, 6799990100000000019, and 76009244561.
```

財務資料類型的資料識別符 ARN

以下列出您可新增至資料保護政策的資料識別符 Amazon Resource Name (ARN)。

財務資料識別符 ARN
arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-DE
arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-ES
arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-FR
arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-GB
arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-IT
arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-US
arn:aws:dataprotection::aws:data-identifier/CreditCardExpiration
arn:aws:dataprotection::aws:data-identifier/CreditCardNumber
arn:aws:dataprotection::aws:data-identifier/CreditCardSecurityCode

受保護醫療資訊 (PHI)

CloudWatch 記錄檔資料保護可以找到下列類型的受保護健康資訊 (PHI)。

如果您設定資料保護政策，則無論 CloudWatch 記錄群組位於何種地理位置，記錄檔都會掃描您指定的資料識別碼。此資料表中國家與地區資料欄中的資訊指出，是否必須在資料識別符後附加兩個字母的國家/地區碼，以偵測這些國家和地區的相應關鍵字。

資料類型	資料識別符 ID	必要的關鍵字	國家和地區
緝毒署 (DEA) 註冊號碼	DrugEnforcementAgencyNumber	dea number, dea registration	美國

資料類型	資料識別符 ID	必要的關鍵字	國家和地區
健康保險卡號碼 (EHIC)	HealthInsuranceCardNumber	assicurazione sanitaria numero, carta assicurazione numero, carte d’assurance maladie , carte européenne d'assurance maladie , ceam, ehic, ehic#, finlandehicnumber# , gesundheitskarte , hälsokort , health card, health card number, health insurance card, health insurance number, insurance card number, krankenversicherungskarte , krankenversicherungsnummer , medical account number, numero conto medico, numéro d’assurance maladie , numéro de carte d’assurance , numéro de compte medical, número de cuenta médica, número de seguro de salud, número de tarjeta de seguro, sairaanhoitokortin , sairausvakuutuskortti , sairausvakuutusnumero , sjukförsäkring	歐盟

資料類型	資料識別符 ID	必要的關鍵字	國家和地區
		nummer, sjukförsäkringskort , suomi ehic-numero , tarjeta de salud, terveyskortti , tessera sanitaria assicurazione numero , versicherungsnummer	
健康保險索償編碼 (HICN)	HealthInsuranceClaimNumber	health insurance claim number, hic no, hic no., hic number, hic#, hicn, hicn#, hicno#	美國
健康保險或醫療識別號碼	HealthInsuranceNumber	carte d'assuré social, carte vitale, insurance card	法國
醫療保健通用程序編碼系統 (HCPCS) 代碼	HealthcareProcedureCode	current procedural terminology , hcpcs, healthcare common procedure coding system	美國
聯邦醫療保險受益人號碼 (MBN)	MedicareBeneficiaryNumber	mbi, medicare beneficiary	美國
國家藥物法規 (NDC)	NationalDrugCode	national drug code, ndc	美國
國家提供者識別符 (NPI)	NationalProviderId	hipaa, n.p.i., national provider, npi	美國

資料類型	資料識別符 ID	必要的關鍵字	國家和地區
國民保健署 (NHS) 號碼	NhsNumber	national health service, NHS	英國
個人健康號碼	PersonalHealthNumber	canada healthcare number, msp number, care number, phn, soins de santé	加拿大

受保護醫療資訊 (PHI) 資料類型的資料識別符 ARN

以下列出可用於受保護醫療資訊 (PHI) 資料保護政策的資料識別符 Amazon Resource Name (ARN)。

PHI 資料識別符 ARN
arn:aws:dataprotection::aws:data-identifier/DrugEnforcementAgencyNumber-US
arn:aws:dataprotection::aws:data-identifier/HealthcareProcedureCode-US
arn:aws:dataprotection::aws:data-identifier/HealthInsuranceCardNumber-EU
arn:aws:dataprotection::aws:data-identifier/HealthInsuranceClaimNumber-US
arn:aws:dataprotection::aws:data-identifier/HealthInsuranceNumber-FR
arn:aws:dataprotection::aws:data-identifier/MedicareBeneficiaryNumber-US
arn:aws:dataprotection::aws:data-identifier/NationalDrugCode-US

PHI 資料識別符 ARN
arn:aws:dataprotection::aws:data-identifier/NationalInsuranceNumber-GB
arn:aws:dataprotection::aws:data-identifier/NationalProviderId-US
arn:aws:dataprotection::aws:data-identifier/NhsNumber-GB
arn:aws:dataprotection::aws:data-identifier/PersonalHealthNumber-CA

個人身分識別資訊 (PII)

CloudWatch 日誌數據保護可以找到以下類型的個人身份信息（PII）。

如果您設定資料保護政策，則無論 CloudWatch 記錄群組位於何種地理位置，記錄檔都會掃描您指定的資料識別碼。此資料表中國家與地區資料欄中的資訊指出，是否必須在資料識別符後附加兩個字母的國家/地區碼，以偵測這些國家和地區的相應關鍵字。

資料類型	資料識別符 ID	必要的關鍵字	國家和地區	備註
出生日期	DateOfBirth	dob, date of birth, birthdate , birth date, birthday, b-day, bday	任何	支援大多數日期格式，例如所有數字以及數字和月份名稱的組合。您可以用空格、斜線 (/) 或連字號

資料類型	資料識別符 ID	必要的關鍵字	國家和地區	備註
				(-) 分隔日期組成部分。
Código de Endereçamento Postal (CEP)	CepCode	cep, código de endereçamento postal, codigo de endereçamento postal	巴西	
Cadastro Nacional da Pessoa Jurídica (CNPJ)	Cnpj	cadastro nacional da pessoa jurídica, cadastro nacional da pessoa juridica, cnpj	巴西	
Cadastro de Pessoas Físicas (CPF)	CpfCode	Cadastro de pessoas fisicas, cadastro de pessoas físicas, cadastro de pessoa física, cadastro de pessoa fisica, cpf	巴西	
駕照識別號碼	DriversLicense	是。不同的關鍵字適用於不同的國家/地區。如需詳細資訊，請參閱本節後文的駕照識別號碼資料表。	許多國家/地區。如需詳細資訊，請參閱駕照識別號碼資料表。	

資料類型	資料識別符 ID	必要的關鍵字	國家和地區	備註
選民名冊號碼	ElectoralRollNumber	electoral #, electoral number, electoral roll #, electoral roll no., electoral roll number, electoralrollno	英國	
個人納稅識別號碼	IndividualTaxIdentificationNumber	是。不同的關鍵字適用於不同的國家/地區。如需詳細資訊，請參閱本節後文的個人納稅人識別號碼資料表。	巴西、法國、德國、西班牙、英國	
國家統計和經濟研究所 (INSEE)	InseeCode	是。不同的關鍵字適用於不同的國家/地區。如需詳細資訊，請參閱本節後文的國民身分證號碼的關鍵字資料表。	法國	

資料類型	資料識別符 ID	必要的關鍵字	國家和地區	備註
國民身分證號碼	NationalIdentificationNumber	是。如需詳細資訊，請參閱本節後文的國民身分證號碼的關鍵字資料表。	德國、義大利、西班牙	這包括 Documento Nacional de Identidad (DNI) 識別符 (西班牙)、Codice fiscale codes (義大利) 和國民身分證號碼 (德國)。
國民保險號碼 (NINO)	NationalInsuranceNumber	insurance no., insurance number, insurance# , national insurance number, nationalinsurance# , nationalinsurancenumber , nin, nino	英國	–
Número de identidad de extranjero (NIE)	NieNumber	是。不同的關鍵字適用於不同的國家/地區。如需詳細資訊，請參閱本節後文的個人納稅人識別號碼資料表。	西班牙	

資料類型	資料識別符 ID	必要的關鍵字	國家和地區	備註
Número de Identificación Fiscal (NIF)	NifNumber	是。不同的關鍵字適用於不同的國家/地區。如需詳細資訊，請參閱本節後文的個人納稅人識別號碼資料表。	西班牙	
護照號碼	PassportNumber	是。不同的關鍵字適用於不同的國家/地區。如需詳細資訊，請參閱本節後文的護照號碼的關鍵字資料表。	加拿大、法國、德國、義大利、西班牙、英國、美國	
永久居留號碼	Permanent Residence Number	carte résident permanent , numéro carte résident permanent , numéro résident permanent , permanent resident card, permanent resident card number, permanent resident no, permanent resident no., permanent resident number, pr no, pr no., pr non, pr number, résident permanent no., résident permanent non	加拿大	

資料類型	資料識別符 ID	必要的關鍵字	國家和地區	備註
電話號碼	PhoneNumber	巴西：關鍵字還包括：cel、celular、fone、m residencial 、numero residencial 、telefone 其他：cell、contact、fax、number、mobile、phone、number、tel、telephone 、telephone number	巴西、加拿大、法國、德國、義大利、西班牙、英國、美國	這包括美國免付費電話號碼和傳真號碼。如果關鍵字與資料相鄰，則該號碼不必包含國家/地區代碼。如果關鍵字不在資料附近，則該數字必須包含國家/地區代碼。
郵遞區號	PostalCode	無	加拿大	
Registro Geral (RG)	RgNumber	是。不同的關鍵字適用於不同的國家/地區。如需詳細資訊，請參閱本節後文的個人納稅人識別號碼資料表。	巴西	

資料類型	資料識別符 ID	必要的關鍵字	國家和地區	備註
社會保險號碼 (SIN)	SocialInsuranceNumber	canadian id, numéro d'assurance sociale, social insurance number, sin	加拿大	
社會安全號碼 (SSN)	Ssn	西班牙 – número de la seguridad social、social security no.、social security no、número de la seguridad social、social security number、socialsecurityno# 、ssn、ssn# 美國 – social security、ss#、ssn	西班牙、美國	

資料類型	資料識別符 ID	必要的關鍵字	國家和地區	備註
納稅識別號碼或參考號碼	TaxId	是。不同的關鍵字適用於不同的國家/地區。如需詳細資訊，請參閱本節後文的個人納稅人識別號碼資料表。.	法國、德國、西班牙、英國	這包括 TIN (法國)；Steueridentifikationsnummer (德國)；CIF (西班牙)；以及 TRN、UTR (英國)。
郵遞區號	ZipCode	zip code, zip+4	美國	美國郵遞區號。
郵寄地址	Address	無	澳洲、加拿大、法國、德國、義大利、西班牙、英國、美國	雖然不需要使用關鍵字，但偵測需要地址中包含城市或地點的名稱以及郵遞區號。

資料類型	資料識別符 ID	必要的關鍵字	國家和地區	備註
電子郵件地址	EmailAddress	無	任何	

資料類型	資料識別符 ID	必要的關鍵字	國家和地區	備註
全球定位系統 (GPS) 座標	LatLong	coordinate ， coordinates ，lat long， latitude longitude ，location, position	任何	CloudWatch 如果緯度和經度座標以一對形式儲存，且使用十進位度 (DD) 格式 (例如 41.948614 , -87.655311)，記錄檔就可以偵測 GPS 座標。支援不包括度數十進位分鐘 (DDM) 格式的座標 (例如 41°56.9168'N 87°39.318

資料類型	資料識別符 ID	必要的關鍵字	國家和地區	備註
				7'W) 或度、分、秒 (DMS) 格式 (例如 41°56'55.0104"N 87°39'19.1196"W)。
全名	Name	無	任何	CloudWatch 記錄檔只能偵測完整名稱。支援僅限於拉丁字元集。

資料類型	資料識別符 ID	必要的關鍵字	國家和地區	備註
車輛識別符 (VIN)	VehicleIdentificationNumber	Fahrgestellnummer , niv, numarul de identificare , numarul seriei de sasiu, serie sasiu, numer VIN, Número de Identificação do Veículo, Número de Identificación de Automóviles , numéro d'identification du véhicule, vehicle identification number, vin, VIN numeris	任何	CloudWatch 記錄檔可偵測包含 17 個字元序列且符合 ISO 3779 和 3780 標準的 VIN。這些標準是專為全球使用而設計的。

駕照識別號碼的關鍵字

為了檢測各種類型的駕駛執照識別號碼， CloudWatch Logs 要求關鍵字與數字相鄰。下表列出 CloudWatch Logs 可在特定國家和地區辨識的關鍵字。

國家/地區或區域	關鍵字
澳洲	dl# dl:, dl :, dlno# driver licence, driver license, driver permit, drivers lic., drivers licence, driver's licence, drivers license, driver's license, drivers permit, driver's permit, drivers permit number, driving licence, driving license, driving permit

國家/地區或區域	關鍵字
奧地利	führerschein, fuhrerschein, führerschein republik österreich, fuhrerschein republik osterreich
比利時	fuehrerschein, fuehrerschein- nr, fuehrersc heinnummer, fuhrerschein, führerschein, fuhrerschein- nr, führerschein- nr, fuhrersch einnummer, führerscheinnummer, numéro permis conduire, permis de conduire, rijbewijs, rijbewijsnummer
保加利亞	превозно средство, свидетелство за управление на моторно, свидетелство за управление на мпс, сумпс, шофьорска книжка
加拿大	dl#, dl:, dlno#, driver licence, driver licences, driver license, driver licenses, driver permit, drivers lic., drivers licence, driver's licence, drivers licences, driver's licences, drivers license, driver's license, drivers licenses, driver's licenses, drivers permit, driver's permit, drivers permit number, driving licence, driving license, driving permit, permis de conduire
克羅埃西亞	vozačka dozvola
賽普勒斯	άδεια οδήγησης
捷克	číslo licence, číslo licence řidiče, číslo řidičského průkazu, ovladače lic., povolení k jízdě, povolení řidiče, řidiči povolení, řidičský prúkaz, řidičský průkaz
丹麥	kørekort, kørekortnummer

國家/地區或區域	關鍵字
愛沙尼亞	juhi litsentsi number, juhiloa number, juhiluba, juhiluba number
芬蘭	ajokortin numero, ajokortti, förare lic., körkort, körkort nummer, kuljettaja lic., permis de conduire
法國	permis de conduire
德國	fuehrerschein, fuehrerschein- nr, fuehrersc heinnummer, fuhrerschein, führerschein, fuhrerschein- nr, führerschein- nr, fuhrersch einnummer, führerscheinnummer
希臘	δεια οδήγησης, adeia odigisis
匈牙利	illesztőprogramok lic, jogosítvány, jogsi, licencszám, vezető engedély, vezetői engedély
愛爾蘭	ceadúnas tiomána
義大利	patente di guida, patente di guida numero, patente guida, patente guida numero
拉脫維亞	autovadītāja apliecība, licences numurs, vadītāja apliecība, vadītāja apliecības numurs, vadītāja atļauja, vadītāja licences numurs, vadītāji lic.
立陶宛	vairuotojo pažymėjimas
盧森堡	fahrerlaubnis, führerschäin
馬爾他	liċenzja tas-sewqan
荷蘭	permis de conduire, rijbewijs, rijbewijsnummer

國家/地區或區域	關鍵字
波蘭	numer licencyjny, prawo jazdy, zezwolenie na prowadzenie
葡萄牙	carta de condução, carteira de habilitação, carteira de motorist, carteira habilitação, carteira motorist, licença condução, licença de condução, número de licença, número licença, permissão condução, permissão de condução
羅馬尼亞	numărul permisului de conducere, permis de conducere
斯洛伐克	číslo licencie, číslo vodičského preukazu, ovládače lic., povolenia vodičov, povolenie jazdu, povolenie na jazdu, povolenie vodiča, vodičský preukaz
斯洛維尼亞	vozniško dovoljenje
西班牙	carnet conducer, el carnet de conducer, licencia conducer, licencia de manejo, número carnet conducer, número de carnet de conducer, número de permiso conducer, número de permiso de conducer, número licencia conducer, número permiso conducer, permiso conducción, permiso conducer, permiso de conducción
瑞典	ajokortin numero, dlno# ajokortti, drivere lic., förare lic., körkort, körkort nummer, körkortsn ummer, kuljettajat lic.

國家/地區或區域	關鍵字
英國	dl#, dl:, dlno#, driver licence, driver licences, driver license, driver licenses, driver permit, drivers lic., drivers licence, driver's licence, drivers licences, driver's licences, drivers license, driver's license, drivers licenses, driver's licenses, drivers permit, driver's permit, drivers permit number, driving licence, driving license, driving permit
美國	dl#, dl:, dlno#, driver licence, driver licences, driver license, driver licenses, driver permit, drivers lic., drivers licence, driver's licence, drivers licences, driver's licences, drivers license, driver's license, drivers licenses, driver's licenses, drivers permit, driver's permit, drivers permit number, driving licence, driving license, driving permit

國民身分證號碼的關鍵字

為了檢測各種類型的國家身份識別號碼， CloudWatch Logs 要求關鍵字與數字相近。這包括 Documento Nacional de Identidad (DNI) 識別符 (西班牙)、法國國家統計和經濟研究所 (INSEE) 代碼、德國國民身分證號碼和 Registro Geral (RG) 號碼 (巴西)。

下表列出 CloudWatch Logs 可在特定國家和地區辨識的關鍵字。

國家/地區或區域	關鍵字
巴西	registro geral, rg
法國	assurance sociale, carte nationale d’identité, cni, code sécurité sociale, French social security number, fssn#, insee, insurance number, national id number, nationalid#, numéro d'assurance, sécurité sociale, sécurité

國家/地區或區域	關鍵字
	sociale non., sécurité sociale numéro, social, social security, social security number, socialsecuritynumber, ss#, ssn, ssn#
德國	ausweisnummer, id number, identification number, identity number, insurance number, personal id, personalausweis
義大利	codice fiscal, dati anagrafici, ehic, health card, health insurance card, p. iva, partita i.v.a., personal data, tax code, tessera sanitaria
西班牙	dni, dni#, dninúmero#, documento nacional de identidad, identidad único, identidadúnico#, insurance number, national identification number, national identity, nationalid#, nationali dno#, número nacional identidad, personal identification number, personal identity no, unique identity number, uniqueid#

護照號碼的關鍵字

為了檢測各種類型的護照號碼， CloudWatch Logs 要求關鍵字與數字相鄰。下表列出 CloudWatch Logs 可在特定國家和地區辨識的關鍵字。

國家/地區或區域	關鍵字
加拿大	passeport, passeport#, passport, passport#, passportno, passportno#
法國	numéro de passeport, passeport, passeport #, passeport #, passeportn °, passeport n °, passeportNon, passeport non

國家/地區或區域	關鍵字
德國	ausstellungsdatum, ausstellungsort, geburtsdatum, passport, passports, reisepass, reisepass–nr, reisepassnummer
義大利	italian passport number, numéro passeport , numéro passeport italien, passaporto, passaporto italiana, passaporto numero, passport number, repubblica italiana passaporto
西班牙	españa pasaporte, libreta pasaporte, número pasaporte, pasaporte, passport, passport book, passport no, passport number, spain passport
英國	passeport #, passeport n °, passeportNon, passeport non, passeportn °, passport #, passport no, passport number, passport#, passportid
美國	passport, travel document

納稅識別號碼及參考號碼的關鍵字

為了檢測各種類型的納稅人身份和參考號碼， CloudWatch 日誌要求關鍵字與數字相鄰。下表列出 CloudWatch Logs 可在特定國家和地區辨識的關鍵字。

國家/地區或區域	關鍵字
巴西	cadastro de pessoa física, cadastro de pessoa fisica, cadastro de pessoas físicas, cadastro de pessoas fisicas, cadastro nacional da pessoa jurídica, cadastro nacional da pessoa juridica, cnpj, cpf
法國	numéro d'identification fiscale, tax id, tax identification number, tax number, tin, tin#

國家/地區或區域	關鍵字
德國	identifikationsnummer, steuer id, steueride ntifikationsnummer, steuernummer, tax id, tax identification number, tax number
西班牙	cif, cif número, cifnúmero#, nie, nif, número de contribuyente, número de identidad de extranjero, número de identificación fiscal, número de impuesto corporativo, personal tax number, tax id, tax identification number, tax number, tin, tin#
英國	paye, tax id, tax id no., tax id number, tax identification, tax identification#, tax no., tax number, tax reference, tax#, taxid#, temporary reference number, tin, trn, unique tax reference, unique taxpayer reference, utr
美國	個人納稅人識別號碼，itin，i.t.i.n。

個人身分識別資訊 (PII) 的資料識別符 ARN

下表列出您可新增至資料保護政策的個人身分識別資訊 (PII) 資料識別符的 Amazon Resource Name (ARN)。

PII 資料識別符 ARN
`arn:aws:dataprotection::aws:data-identifier/Address`
`arn:aws:dataprotection::aws:data-identifier/CepCode-BR`
`arn:aws:dataprotection::aws:data-identifier/Cnpj-BR`
`arn:aws:dataprotection::aws:data-identifier/CpfCode-BR`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-AT`

PII 資料識別符 ARN
arn:aws:dataprotection::aws:data-identifier/DriversLicense-AU
arn:aws:dataprotection::aws:data-identifier/DriversLicense-BE
arn:aws:dataprotection::aws:data-identifier/DriversLicense-BG
arn:aws:dataprotection::aws:data-identifier/DriversLicense-CA
arn:aws:dataprotection::aws:data-identifier/DriversLicense-CY
arn:aws:dataprotection::aws:data-identifier/DriversLicense-CZ
arn:aws:dataprotection::aws:data-identifier/DriversLicense-DE
arn:aws:dataprotection::aws:data-identifier/DriversLicense-DK
arn:aws:dataprotection::aws:data-identifier/DriversLicense-EE
arn:aws:dataprotection::aws:data-identifier/DriversLicense-ES
arn:aws:dataprotection::aws:data-identifier/DriversLicense-FI
arn:aws:dataprotection::aws:data-identifier/DriversLicense-FR
arn:aws:dataprotection::aws:data-identifier/DriversLicense-GB
arn:aws:dataprotection::aws:data-identifier/DriversLicense-GR
arn:aws:dataprotection::aws:data-identifier/DriversLicense-HR
arn:aws:dataprotection::aws:data-identifier/DriversLicense-HU
arn:aws:dataprotection::aws:data-identifier/DriversLicense-IE
arn:aws:dataprotection::aws:data-identifier/DriversLicense-IT
arn:aws:dataprotection::aws:data-identifier/DriversLicense-LT
arn:aws:dataprotection::aws:data-identifier/DriversLicense-LU

PII 資料識別符 ARN
arn:aws:dataprotection::aws:data-identifier/DriversLicense-LV
arn:aws:dataprotection::aws:data-identifier/DriversLicense-MT
arn:aws:dataprotection::aws:data-identifier/DriversLicense-NL
arn:aws:dataprotection::aws:data-identifier/DriversLicense-PL
arn:aws:dataprotection::aws:data-identifier/DriversLicense-PT
arn:aws:dataprotection::aws:data-identifier/DriversLicense-RO
arn:aws:dataprotection::aws:data-identifier/DriversLicense-SE
arn:aws:dataprotection::aws:data-identifier/DriversLicense-SI
arn:aws:dataprotection::aws:data-identifier/DriversLicense-SK
arn:aws:dataprotection::aws:data-identifier/DriversLicense-US
arn:aws:dataprotection::aws:data-identifier/ElectoralRollNumber-GB
arn:aws:dataprotection::aws:data-identifier/EmailAddress
arn:aws:dataprotection::aws:data-identifier/IndividualTaxIdentificationNumber-US
arn:aws:dataprotection::aws:data-identifier/InseeCode-FR
arn:aws:dataprotection::aws:data-identifier/LatLong
arn:aws:dataprotection::aws:data-identifier/Name
arn:aws:dataprotection::aws:data-identifier/NationalIdentificationNumber-DE
arn:aws:dataprotection::aws:data-identifier/NationalIdentificationNumber-ES

PII 資料識別符 ARN
arn:aws:dataprotection::aws:data-identifier/NationalIdentificationNumber-IT
arn:aws:dataprotection::aws:data-identifier/NieNumber-ES
arn:aws:dataprotection::aws:data-identifier/NifNumber-ES
arn:aws:dataprotection::aws:data-identifier/PassportNumber-CA
arn:aws:dataprotection::aws:data-identifier/PassportNumber-DE
arn:aws:dataprotection::aws:data-identifier/PassportNumber-ES
arn:aws:dataprotection::aws:data-identifier/PassportNumber-FR
arn:aws:dataprotection::aws:data-identifier/PassportNumber-GB
arn:aws:dataprotection::aws:data-identifier/PassportNumber-IT
arn:aws:dataprotection::aws:data-identifier/PassportNumber-US
arn:aws:dataprotection::aws:data-identifier/PermanentResidenceNumber-CA
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-BR
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-DE
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-ES
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-FR
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-GB
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-IT
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-US
arn:aws:dataprotection::aws:data-identifier/PostalCode-CA

PII 資料識別符 ARN
arn:aws:dataprotection::aws:data-identifier/RgNumber-BR
arn:aws:dataprotection::aws:data-identifier/SocialInsuranceNumber-CA
arn:aws:dataprotection::aws:data-identifier/Ssn-ES
arn:aws:dataprotection::aws:data-identifier/Ssn-US
arn:aws:dataprotection::aws:data-identifier/TaxId-DE
arn:aws:dataprotection::aws:data-identifier/TaxId-ES
arn:aws:dataprotection::aws:data-identifier/TaxId-FR
arn:aws:dataprotection::aws:data-identifier/TaxId-GB
arn:aws:dataprotection::aws:data-identifier/VehicleIdentificationNumber
arn:aws:dataprotection::aws:data-identifier/ZipCode-US

自訂資料識別符

主題

什麼是 SNS 自訂資料識別符？

自訂資料識別符 (CDI) 可讓您定義自訂的規則運算式，以用於資料保護政策。使用自訂資料識別符就可以鎖定受管資料識別符無法提供的企業特定個人身分識別資訊 (PII) 使用案例。例如，您可以使用自訂資料識別符來尋找公司專屬員工 ID。自訂資料識別符可與受管資料識別符搭配使用。

自訂資料識別符的限制

CloudWatch 記錄檔自訂資料識別碼有下列限制：

- 每個資料保護政策最多可支援 10 個自訂資料識別符。
- 自訂資料識別符名稱的長度上限為 128 個字元。支援的字元如下：
  - 英數字：(a-zA-Z0-9)
  - 符號：('_' | '-')
- RegEx 的長度上限為 200 個字元。支援的字元如下：
  - 英數字：(a-zA-Z0-9)
  - 符號：('_' | '#' | '=' | '@' |'/' | ';' | ',' | '-' | ' ')
  - RegEx 保留字元：('^' | '$' | '?' | '[' | ']' | '{' | '}' | '|' | '\\' | '*' | '+' | '.')
- 自訂資料識別符的名稱不可與受管資料識別符的名稱相同。
- 您可以在帳戶層級資料保護原則或記錄群組層級資料保護原則中指定自訂資料識別碼。與受管資料識別碼類似，帳戶層級政策中定義的自訂資料識別碼會與記錄群組層級原則中定義的自訂資料識別碼搭配使用。

在主控台中使用自訂資料識別碼

當您使用 CloudWatch 主控台建立或編輯資料保護原則時，若要指定自訂資料識別碼，只要輸入資料識別碼的名稱和一般運算式即可。例如，您可以輸**Employee_ID**入名稱和**EmployeeID-\d{9}**規則運算式。這個規則運算式會偵測並遮罩 9 個數字之後的記錄事件EmployeeID-。例如：EmployeeID-123456789

在您的資料保護政策中使用自訂資料識別符

如果您使用 AWS CLI 或 AWS API 來指定自訂資料識別碼，則需要在用於定義資料保護政策的 JSON 政策中包含資料識別碼名稱和規則運算式。下列資料保護原則會偵測並遮罩包含公司特定員工 ID 的記錄事件。

1. 在您的資料保護政策內建立 Configuration 區塊。
2. 輸入自訂資料識別符的 Name。例如 **EmployeeId**。
3. 輸入自訂資料識別符的 Regex。例如 **EmployeeID-\d{9}**。此規則運算式會比對包含 EmployeeID- 9 位數之後的記錄事件EmployeeID-。例如：EmployeeID-123456789
4. 請參閱政策聲明中的下列自訂資料識別符。

```
{
    "Name": "example_data_protection_policy",
    "Description": "Example data protection policy with custom data identifiers",
    "Version": "2021-06-01",
    "Configuration": {
      "CustomDataIdentifier": [
        {"Name": "EmployeeId", "Regex": "EmployeeId-\\d{9}"}
      ]
    },
    "Statement": [
        {
            "Sid": "audit-policy",
            "DataIdentifier": [
                "EmployeeId"
            ],
            "Operation": {
                "Audit": {
                    "FindingsDestination": {
                        "S3": {
                            "Bucket": "EXISTING_BUCKET"
                        }
                    }
                }
            }
        },
        {
            "Sid": "redact-policy",
            "DataIdentifier": [
            "EmployeeId"
            ],
            "Operation": {
                "Deidentify": {
                    "MaskConfig": {
                    }
                }
            }
        }
    ]
}
```

5. (選用) 視需要繼續將其他自訂資料識別符新增至 Configuration 區塊。資料保護政策目前最多可支援 10 個自訂資料識別符。

使用篩選條件從日誌事件建立指標

您可以建立一或多個量度篩選器，搜尋和篩選進入 CloudWatch 記錄檔的記錄資料。量度篩選器會定義傳送至記錄檔時要在 CloudWatch 記錄檔資料中尋找的術語和模式。 CloudWatch 日誌使用這些 CloudWatch 指標過濾器將日誌數據轉換為可以繪製圖形或設置警報的數字指標。

當您從記錄篩選條件建立指標時，您也可以選擇指派指標的維度和單位。如果您指定單位，請務必在建立篩選條件時指定正確的單位。後來再變更篩選條件的單位沒有作用。

> **Note**
>
> 只有「標準」記錄檔類別中的記錄群組才支援度量篩選器。如需記錄類別的詳細資訊，請參閱日誌類。

檢視這些量度或設定警示時，您可以使用任何類型的 CloudWatch 統計資料，包括百分位數統計資料。

> **Note**
>
> 只有在指標值全都不是負數時，才會支援百分位數統計資料。如果您設定指標篩選條件，使其可以回報負數，百分位數統計資料在擁有負數的值時將無法用於該指標。如需詳細資訊，請參閱百分位數。

篩選條件不追溯篩選條件資料。篩選條件只針對篩選條件建立後發生的事件發佈指標資料點。篩選條件結果會傳回前 50 行，如果篩選結果的時間戳記早於指標建立時間，這些行則不會顯示。

目錄

概念

每個指標篩選條件是由下列關鍵元素組成：

預設值

值會在日誌被擷取但沒有發現相符日誌的時間段內，回報至指標篩選條件。將此設定為 0，可確保每個時間段都會回報資料，以免某些時間段沒有相符的資料，而產生起伏不定的指標。不過，如果在 1 分鐘的時間段內沒有擷取任何日誌事件，則不會回報任何值。

如果您將維度指派給由指標篩選條件建立的指標，則無法為該指標指派預設值。

維度

維度是進一步定義指標的鍵值組。您可以將維度指派給從指標篩選條件建立的指標。由於維度是指標唯一識別符的一部分，每當您從日誌擷取唯一名稱/值組時，就是在建立該指標的新變化。

篩選條件模式

CloudWatch 記錄應如何解譯每個記錄事件中的資料的符號描述。例如，日誌項目可能包含時間戳記、IP 地址、字串，以此類推。您可以使用模式以指定要在日誌檔中尋找的項目。

指標名稱

要發佈監督日誌資訊的 CloudWatch 測量結果名稱。例如，您可以發佈到名為的量度 ErrorCount。

指標命名空間

新 CloudWatch 測量結果的目的地命名空間。

指標值

每次找到相符日誌時要發佈到指標的數值。例如，如果您是計數特定詞彙 (像是 "Error") 的出現次數，每個出現次數的值將為 "1"。如果您是計數傳出的位元組，您可以透過日誌事件中找到的實際位元組數來遞增計數。

指標篩選條件的篩選條件模式語法

> **Note**
>
> 量度篩選器如何不同 CloudWatch 日誌見解查詢
> 量度篩選器與 CloudWatch Logs Insights 查詢不同之處在於，每次找到相符的記錄檔時，都會將指定的數值新增至量度篩選器。如需詳細資訊，請參閱 配置指標篩選條件的指標值。

如需如何使用 Amazon CloudWatch 日誌洞見查詢語言查詢日誌群組的相關資訊，請參閱CloudWatch 日誌見解查詢語法。

一般篩選條件模式範例

如需適用於指標篩選條件以及訂閱篩選條件和篩選條件日誌事件的一般篩選條件模式語法，請參閱適用於指標篩選條件、訂閱篩選條件和篩選條件日誌事件的篩選條件模式語法，其中包括下列範例：

- 支援的規則運算式 (regex) 語法
- 在非結構化日誌事件中比對詞彙
- 在 JSON 日誌事件中比對詞彙
- 比對以空格分隔的日誌事件中的詞彙

指標過濾器允許您搜索和過濾進入 CloudWatch 日誌的日誌數據，從過濾的日誌數據中提取指標觀測，並將數據點轉換為 CloudWatch 日誌指標。您可以定義要在記錄檔資料傳送至記錄檔時在 CloudWatch 記錄檔中尋找的術語和模式。指標篩選條件指派給日誌群組，並且指派給日誌群組的所有篩選條件都會套用於其日誌串流。

當指標篩選條件與某個詞彙相符時，便會以指定的數值增加指標的計數。例如：您可以建立指標篩選條件來搜尋與計算日誌事件中單字 ERROR (錯誤) 的出現次數。

您可以對指標指派度量單位和維度。例如：如果您建立了一個指標篩選條件，以計算單字 ERROR (錯誤) 在日誌事件中出現的次數，則可以指定一個名為 `ErrorCode` 的維度，以顯示包含 ERROR (錯誤) 這個單字的日誌事件總數，並根據回報的錯誤代碼篩選資料。

Tip

指派度量單位給指標時，務必指定正確的單位。如果隨後再更改單位，則更改可能無法生效。如需 CloudWatch 支援的單位的完整清單，請參閱 Amazon CloudWatch API 參考MetricDatum中的。

主題

- 配置指標篩選條件的指標值
- 從 JSON 或空格分隔日誌事件的值中將維度連同指標一起發佈
- 使用日誌事件中的值來增加指標的值

配置指標篩選條件的指標值

建立指標篩選條件時，您可以定義篩選條件模式，並指定指標的值和預設值。您可以將指標值設置為數字、名稱標識符或數字識別碼。如果您未指定預設值，則在量度篩選器找不到相符項目時，CloudWatch 將不會回報資料。我們建議您指定預設值，即使該值為 0。設定預設值有助於更準確地 CloudWatch 報告資料，並防 CloudWatch 止彙總不定量度。 CloudWatch 每分鐘彙總和報告量度值。

指標篩選條件在日誌事件中找到相符項目時，會根據指標值增加指標的計數。如果您的量度篩選器找不到相符項目，則會 CloudWatch 報告量度的預設值。例如：假設有一個日誌群組，每分鐘發佈兩個記錄且指標值為 1，預設值為 0。如果指標篩選條件第一分鐘內，在兩個日誌記錄中找到相符項目，則該分鐘的指標值為 2。如果指標篩選條件第二分鐘內，未在任何一筆記錄中找到相符項目，則該分鐘的預設值為 0。如果將維度分配給指標篩選條件產生的指標，則無法為這些指標指定預設值。

您也可以設置指標篩選條件，使用從日誌事件中擷取的值 (而非靜態值) 來增加指標。如需詳細資訊，請參閱 使用日誌事件中的值來增加指標的值。

從 JSON 或空格分隔日誌事件的值中將維度連同指標一起發佈

您可以使用 CloudWatch 主控台或 AWS CLI 建立指標篩選器，以發佈具有 JSON 和空格分隔記錄事件產生的度量的維度。維度是名稱/數值配對，僅適用於 JSON 和空格分隔的篩選條件模式。您可以建立最多包含三個維度的 JSON 和空格分隔的指標篩選條件。如需維度以及如何將維度指派給指標的詳細資訊，請參閱下列各節：

- Amazon CloudWatch 用戶指南中的尺寸
- 範例：從 Apache 日誌擷取欄位，並在 Amazon CloudWatch 日誌使用者指南中指派維度

> **Important**
>
> 維度包含與自訂指標相同收費的值。為了避免費用出乎意料，請勿將高基數欄位指定為維度，例如 `IPAddress` 或 `requestID`。
> 從日誌事件擷取的指標會以自訂指標收費。為了避免費用意外過高，如果指標篩選條件針對您已指定的維度，在一定時間內產生 1000 個不同的名稱/值組，Amazon 可能會停用該指標篩選條件。
> 您可以建立帳單警示，通知您預估的費用。如需詳細資訊，請參閱建立帳單警示以監控您的預估 AWS 費用。

從 JSON 日誌事件將維度連同指標一起發佈

下列範例程式碼片段描述如何在 JSON 指標篩選條件中指定維度。

Example: JSON log event

```
{
  "eventType": "UpdateTrail",
  "sourceIPAddress": "111.111.111.111",
  "arrayKey": [
        "value",
        "another value"
  ],
  "objectList": [
       {"name": "a",
         "id": 1
       },
       {"name": "b",
         "id": 2
       }
  ]

}
```

> **Note**
>
> 如果要使用 JSON 日誌事件範例測試指標篩選條件範例，則必須在單行中輸入JSON 日誌範例。

Example: Metric filter

每當 JSON 日誌事件包含屬性 `eventType` 和 `"sourceIPAddress"` 時，指標篩選條件會增加指標。

```
{ $.eventType = "*" && $.sourceIPAddress != 123.123.* }
```

當您建立 JSON 指標篩選條件時，您可以將指標篩選條件中的任何屬性指定為維度。例如：若要設定 `eventType` 作為維度，請使用下列內容：

```
"eventType" : $.eventType
```

指標範例包含一個名為 "eventType" 的維度，其維度值在日誌事件範例中為 "UpdateTrail"。

從空格分隔日誌事件將維度連同指標一起發佈

下列範例程式碼片段，描述如何在空格分隔的指標篩選條件中指定維度。

Example: Space-delimited log event

```
127.0.0.1 Prod frank [10/Oct/2000:13:25:15 -0700] "GET /index.html HTTP/1.0" 404
 1534
```

Example: Metric filter

```
[ip, server, username, timestamp, request, status_code, bytes > 1000]
```

當空格分隔的日誌事件包含篩選條件中指定的任何欄位時，該指標篩選條件會增加指標。例如：指標篩選條件在空格分隔的日誌事件範例中查找下列欄位和值。

```
{
   "$bytes": "1534",
   "$status_code": "404",

   "$request": "GET /index.html HTTP/1.0",
   "$timestamp": "10/Oct/2000:13:25:15 -0700",
   "$username": "frank",
   "$server": "Prod",
   "$ip": "127.0.0.1"
}
```

當您建立空格分隔的指標篩選條件時，您可以將指標篩選條件中的任何欄位指定為維度。例如：若要設定 server 作為維度，請使用下列內容：

```
"server" : $server
```

指標篩選條件範例中有一個名為 server 的維度，其維度值在日誌事件範例中為 "Prod"。

Example: Match terms with AND (&&) and OR (||)

您可以使用邏輯運算子邏輯與 ("&&") 和邏輯或 ("||") 建立包含條件的空格分隔的指標篩條件。下列指標篩選條件會傳回第一個單字為 ERROR 或 WARN 超級字串的日誌事件。

```
[w1=ERROR || w1=%WARN%, w2]
```

使用日誌事件中的值來增加指標的值

您可以建立指標篩選條件來發佈日誌事件中找到的數值。本節中的過程中使用下列範例指標篩選條件，來示範如何將 JSON 日誌事件中的數值發佈到指標。

```
{ $.latency = * } metricValue: $.latency
```

建立在日誌事件中發佈的值的指標篩選條件

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇 Logs (日誌)，然後選擇 Log groups (日誌群組)。
3. 選擇或建立日誌群組。

   如需如何建立日誌群組的詳細資訊，請參閱 Amazon 日誌使用指南中的在 CloudWatch 日誌中建立 CloudWatch 日誌群組。
4. 選擇 Actions (動作)，然後選擇 Create metric filter (建立指標篩選條件)。
5. 針對 Filter Pattern (篩選條件模式)，輸入 **`{ $.latency = * }`**，然後選擇 Next (下一步)。
6. 針對 Metric Name (指標名稱)，輸入 myMetric。
7. 針對 Metric Value (指標值)，輸入 **`$.latency`**。

8. (選用) 針對 Default Value (預設值)，輸入 0，然後選擇 Next (下一步)。

   我們建議您指定預設值，即使該值為 0。設定預設值有助於更準確地 CloudWatch 報告資料，並防 CloudWatch 止彙總不定量度。 CloudWatch 每分鐘彙總和報告量度值。

9. 選擇 Create metric filter (建立指標篩選條件)。

指標篩選條件的範例與 JSON 日誌事件範例中的詞彙 "latency" 相符，並將數值 50 發佈到指標 myMetric。

```
{
"latency": 50,
"requestType": "GET"
}
```

建立指標篩選條件

下列程序和範例示範如何建立指標篩選條件。

範例

- 建立日誌群組的指標篩選條件
- 範例：計算日誌事件數量
- 範例：計算詞彙的出現次數
- 範例：Count HTTP 404 代碼
- 範例：Count HTTP 4xx 代碼
- 範例：從 Apache 日誌擷取欄位並指派維度

建立日誌群組的指標篩選條件

若要建立日誌群組的指標篩選條件，請遵循下列步驟。在有一些資料點之前，是看不到該指標的。

使用 CloudWatch 主控台建立量度篩選

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇 Logs (日誌)，然後選擇 Log groups (日誌群組)。
3. 選擇日誌群組的名稱。

4. 選擇 `Actions`，然後選擇 Create metric filter (建立指標篩選條件)。
5. 針對 Filter pattern (篩選條件模式)，輸入篩選條件模式。如需詳細資訊，請參閱 用於指標篩選條件、訂閱篩選條件、篩選條件日誌事件和 Live Tail 的篩選條件模式語法。
6. (選用) 若要測試篩選條件模式，請在 Test Pattern (測試模式) 下方，輸入一個或多個日誌事件來測試模式。每個日誌事件必須在一行中格式化。分行符號用於分隔 Log event messages (日誌事件訊息) 方塊中的日誌事件。
7. 選擇 Next (下一步)，然後輸入指標篩選條件的名稱。
8. 在「測量結果詳細資訊」下，針對測量結果命名 CloudWatch 空間，輸入要在其中發行測量結果的命名空間名稱 如果命名空間不存在，請務必選取 Create new (新建)。
9. 針對 Metric name (指標名稱)，輸入新指標的名稱。
10. 針對 Metric value (指標值)，如果指標篩選條件計算篩選條件中的關鍵字出現次數，請輸入 1。如此會針對包含其中一個關鍵字的每個日誌事件，將指標遞增 1。

    或者輸入字符，例如 **`$size`**。如此會針對包含 `size` 欄位的每個日誌事件，以 `size` 欄位中的數值遞增指標。
11. (選用) 針對 Unit (單位)，選取要指派給指標的單位。如果您未指定單位，單位會設為 None。
12. (選用) 最多為指標的三個維度輸入名稱和字符。如果將維度分配給指標篩選條件建立的指標，則無法為這些指標指派預設值。

    > **Note**
    >
    > 僅在 JSON 或空格分隔指標篩選條件中支援維度。

13. 選擇 Create metric filter (建立指標篩選條件)。可以從導覽窗格中找到您建立的指標篩選條件。選擇 Logs (日誌)，然後選擇 Log groups (日誌群組)。選擇您為其建立指標篩選條件的日誌群組名稱，然後選取 Metric filters (指標篩選條件) 標籤。

範例：計算日誌事件數量

日誌事件監控的最簡單類型就是計數發生的日誌事件數。您可以這樣做以保留所有事件的計數，以建立「活動訊號」樣式監控或僅練習建立指標篩選條件。

在下列 CLI 範例中，會將名 MyAppAccessCount 為的度量篩選器套用至記錄群組 MyApp / access.log，以便在 CloudWatch命名空間 EventCount 中建立度量 MyNamespace。系統會將篩選條件設定為符合任何日誌事件的內容，並以「1」遞增指標。

使用 CloudWatch 主控台建立量度篩選

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇 Log groups (日誌群組)。
3. 選擇日誌群組的名稱。
4. 選擇 `Actions` > Create metric filter (建立指標篩選條件)。
5. 將 Filter Pattern (篩選條件模式) 和 Select Log Data to Test (選取要測試的日誌資料) 保留空白。
6. 選擇 Next (下一步)，然後針對 Filter Name (篩選條件名稱)，輸入 **`EventCount`**。
7. 在 Metric Details (指標詳細資訊) 下的 Metric Namespace (指標命名空間) 中，輸入 **`MyNameSpace`**。
8. 針對 Metric Name (指標名稱)，輸入 **`MyAppEventCount`**。
9. 確認 Metric Value (指標值) 為 1。這會指定針對每個日誌事件的計數以 1 遞增。
10. 針對 Default Value (預設值)，輸入 0，然後選擇 Next (下一步)。指定預設值可確保即使沒有任何日誌事件發生時仍會報告資料，以避免發生 spotty 指標 (資料有時不存在)。
11. 選擇 Create metric filter (建立指標篩選條件)。

使用建立度量篩選 AWS CLI

在命令提示中，執行下列命令：

```
aws logs put-metric-filter \
  --log-group-name MyApp/access.log \
  --filter-name EventCount \
  --filter-pattern " " \
  --metric-transformations \
  metricName=MyAppEventCount,metricNamespace=MyNamespace,metricValue=1,defaultValue=0
```

您可以透過張貼任何事件資料來測試這個新政策。您應該會看到已發佈至指標的資料點 MyAppAccessEventCount。

若要使用 AWS CLI

在命令提示中，執行下列命令：

```
aws logs put-log-events \
  --log-group-name MyApp/access.log --log-stream-name TestStream1 \
```

```
  --log-events \
    timestamp=1394793518000,message="Test event 1" \
    timestamp=1394793518000,message="Test event 2" \
    timestamp=1394793528000,message="This message also contains an Error"
```

範例：計算詞彙的出現次數

日誌事件經常包含您想要計數，或是有關操作成功或失敗操作的重要訊息。例如，若指定的操作失敗，錯誤可能會發生且系統會將該錯誤記錄到日誌檔。您可能想要監控這些項目，以了解錯誤的趨勢。

在下例中，建立指標篩選條件來監控「Error」詞彙。此原則已建立並新增至記錄群組 MyApp/message.log。 CloudWatch 記錄會將資料點發佈至 MyApp/message.log 命名空間 ErrorCount 中的 CloudWatch 自訂量度，對於每個包含 Error 的事件，其值為「1」。如果事件不包含單字「Error」，則會發佈值 0。在 CloudWatch主控台中繪製此資料圖形時，請務必使用總和統計資料。

建立量度篩選後，您可以在 CloudWatch 主控台中檢視指標。選取要檢視的指標時，請選取符合日誌群組名稱的指標命名空間。如需詳細資訊，請參閱檢視可用的指標。

使用 CloudWatch 主控台建立量度篩選

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇 Log groups (日誌群組)。
3. 選擇日誌群組的名稱。
4. 選擇 Actions (動作) > Create metric filter (建立指標篩選條件)。
5. 針對 Filter Pattern (篩選條件模式)，輸入 **`Error`**。

   > **Note**
   >
   > 在 Filter Pattern (篩選條件模式) 中的所有項目都會區分大小寫。

6. (選用) 若要測試篩選條件模式，請在 Test Pattern (測試模式) 下方，輸入一個或多個日誌事件，用以測試模式。每個日誌事件都必須在一行內，因為 Log event messages (日誌事件訊息) 方塊中使用換行來分隔日誌事件。
7. 選擇 Next (下一步)，然後在 Assign metric (指派指標) 頁面上，針對 Filter Name (篩選條件名稱) 輸入 **`MyAppErrorCount`**。
8. 在「測量結果詳細資訊」下，針對測量結果命名 MyNameSpace
9. 針對 Metric Name (指標名稱)，輸入 ErrorCount。

10. 確認 Metric Value (指標值) 為 1。這會指定針對每個包含「Error」的日誌事件計數以 1 的方式遞增。
11. 針對 Default Value (預設值)，輸入 0，然後選擇 Next (下一步)。
12. 選擇 Create metric filter (建立指標篩選條件)。

使用建立度量篩選 AWS CLI

在命令提示中，執行下列命令：

```
aws logs put-metric-filter \
  --log-group-name MyApp/message.log \
  --filter-name MyAppErrorCount \
  --filter-pattern 'Error' \
  --metric-transformations \
      metricName=ErrorCount,metricNamespace=MyNamespace,metricValue=1,defaultValue=0
```

您可以透過張貼在訊息中包含「錯誤」單字的事件來測試這個新政策。

若要使用 AWS CLI

畫面出現命令提示時，執行下列 命令。請注意，模式會區分大小寫。

```
aws logs put-log-events \
  --log-group-name MyApp/access.log --log-stream-name TestStream1 \
  --log-events \
    timestamp=1394793518000,message="This message contains an Error" \
    timestamp=1394793528000,message="This message also contains an Error"
```

範例：Count HTTP 404 代碼

使用 CloudWatch 日誌，您可以監視 Apache 服務器返回 HTTP 404 響應的次數，也就是找不到頁面的響應代碼。您可能想要監控此次數以了解您的網站訪客找不到所需資源的頻率。假設您的日誌記錄的建置是包含每個日誌事件 (網站瀏覽) 的以下資訊：

- 請求者 IP 地址
- RFC 1413 身分
- 使用者名稱
- 時間戳記

- 含請求資源和通訊協定的請求方法
- 要請求的 HTTP 回應碼
- 請求中傳入的位元組數

此範例看起來與以下類似：

```
127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 404 2326
```

您可以指定規則，該規則會嘗試比對該結構的事件是否含 HTTP 404 錯誤，如下範例所示：

使用 CloudWatch 主控台建立量度篩選

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇 Log groups (日誌群組)。
3. 選擇 `Actions` > Create metric filter (建立指標篩選條件)。
4. 針對 Filter Pattern (篩選條件模式)，輸入 **`[IP, UserInfo, User, Timestamp, RequestInfo, StatusCode=404, Bytes]`**。
5. (選用) 若要測試篩選條件模式，請在 Test Pattern (測試模式) 下方，輸入一個或多個日誌事件，用以測試模式。每個日誌事件都必須在一行內，因為 Log event messages (日誌事件訊息) 方塊中使用換行來分隔日誌事件。
6. 選擇 Next (下一步)，然後針對 Filter Name (篩選條件名稱)，輸入 HTTP404Errors。
7. 在 Metric Details (指標詳細資料) 下的 Metric Namespace (指標命名空間) 中，輸入 **`MyNameSpace`**。
8. 針對 Metric Name (指標名稱)，輸入 **`ApacheNotFoundErrorCount`**。
9. 確認 Metric Value (指標值) 為 1。這會指定針對每個 404 錯誤事件的計數增加 1。
10. 針對 Default Value (預設值)，輸入 0，然後選擇 Next (下一步)。
11. 選擇 Create metric filter (建立指標篩選條件)。

使用建立度量篩選 AWS CLI

在命令提示中，執行下列命令：

```
aws logs put-metric-filter \
  --log-group-name MyApp/access.log \
```

```
  --filter-name HTTP404Errors \
  --filter-pattern '[ip, id, user, timestamp, request, status_code=404, size]' \
  --metric-transformations \
      metricName=ApacheNotFoundErrorCount,metricNamespace=MyNamespace,metricValue=1
```

在這個範例中，會使用到左右方括號、雙引號和字元字串 404 等常值字元。此模式需與被視為監控之日誌事件的整個日誌事件訊息相比對。

您可以使用 describe-metric-filters 命令來驗證指標篩選條件的建立。您應該會看到輸出，如下所示：

```
aws logs describe-metric-filters --log-group-name MyApp/access.log

{
    "metricFilters": [
        {
            "filterName": "HTTP404Errors",
            "metricTransformations": [
                {
                    "metricValue": "1",
                    "metricNamespace": "MyNamespace",
                    "metricName": "ApacheNotFoundErrorCount"
                }
            ],
            "creationTime": 1399277571078,
            "filterPattern": "[ip, id, user, timestamp, request, status_code=404,
 size]"
        }
    ]
}
```

現在您可以手動張貼幾個事件：

```
aws logs put-log-events \
--log-group-name MyApp/access.log --log-stream-name hostname \
--log-events \
timestamp=1394793518000,message="127.0.0.1 - bob [10/Oct/2000:13:55:36 -0700] \"GET /
apache_pb.gif HTTP/1.0\" 404 2326" \
timestamp=1394793528000,message="127.0.0.1 - bob [10/Oct/2000:13:55:36 -0700] \"GET /
apache_pb2.gif HTTP/1.0\" 200 2326"
```

放置這些示例日誌事件後不久，您可以檢索在控制 CloudWatch 台中命名為的指標 ApacheNotFoundErrorCount。

範例：Count HTTP 4xx 代碼

在上述範例中，您可能想要監控 web 服務存取日誌和監控 HTTP 回應碼層級。例如，您可能想要監控所有 HTTP 400 層級錯誤。不過，您可能不會想要為每個傳回程式碼指定新指標篩選條件。

以下範例示範如何建立指標，其中包含使用從 範例：Count HTTP 404 代碼 範例之 Apache 存取日誌格式來自存取日誌的所有 400 層級 HTTP 程式碼回應。

使用 CloudWatch 主控台建立量度篩選

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇 Log groups (日誌群組)。
3. 選擇 Apache 伺服器的日誌群組名稱。
4. 選擇 `Actions` > Create metric filter (建立指標篩選條件)。
5. 針對 Filter Pattern (篩選條件模式)，輸入 **`[ip, id, user, timestamp, request, status_code=4*, size]`**。
6. (選用) 若要測試篩選條件模式，請在 Test Pattern (測試模式) 下方，輸入一個或多個日誌事件，用以測試模式。每個日誌事件都必須在一行內，因為 Log event messages (日誌事件訊息) 方塊中使用換行來分隔日誌事件。
7. 選擇 Next (下一步)，然後針對 Filter Name (篩選條件名稱)，輸入 **`HTTP4xxErrors`**。
8. 在 Metric (指標詳細資訊) 下的 Metric Namespace (指標命名空間) 中，輸入 **`MyNameSpace`**。
9. 針對 Metric name (指標名稱)，輸入 HTTP4xxErrors。
10. 針對 Metric value (指標值)，輸入 1。這會指定針對每個包含「4xx 錯誤」的日誌事件以 1 的方式遞增計數。
11. 針對 Default value (預設值)，輸入 0，然後選擇 Next (下一步)。
12. 選擇 Create metric filter (建立指標篩選條件)。

使用建立度量篩選 AWS CLI

在命令提示中，執行下列命令：

```
aws logs put-metric-filter \
  --log-group-name MyApp/access.log \
  --filter-name HTTP4xxErrors \
  --filter-pattern '[ip, id, user, timestamp, request, status_code=4*, size]' \
  --metric-transformations \
```

```
metricName=HTTP4xxErrors,metricNamespace=MyNamespace,metricValue=1,defaultValue=0
```

您可以使用 put-event 呼叫中的以下資料來測試這個規則。如果您沒有在之前的範例中移除監控規則，您將會產生兩種不同的指標。

```
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /~test/ HTTP/1.1" 200 3
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:51:34 -0700] "GET /~test/index.html HTTP/1.1" 200 3
```

範例：從 Apache 日誌擷取欄位並指派維度

有時，不使用計數，而是在指標值的個別日誌事件中使用值很有用。此範例顯示如何建立擷取規則來建立指標，該指標會量測 Apache Web 伺服器傳出的位元組數。

此範例也會說明如何將維度指派給您要建立的指標。

使用 CloudWatch 主控台建立量度篩選

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇 Log groups (日誌群組)。
3. 選擇 Apache 伺服器的日誌群組名稱。
4. 選擇 `Actions` > Create metric filter (建立指標篩選條件)。
5. 針對 Filter Pattern (篩選條件模式)，輸入 **`[ip, id, user, timestamp, request, status_code, size]`**。
6. (選用) 若要測試篩選條件模式，請在 Test Pattern (測試模式) 下方，輸入一個或多個日誌事件，用以測試模式。每個日誌事件都必須在一行內，因為 Log event messages (日誌事件訊息) 方塊中使用換行來分隔日誌事件。
7. 選擇 Next (下一步)，然後針對 Filter Name (篩選條件名稱)，輸入 **`size`**。
8. 在 Metric (指標詳細資訊) 下的 Metric Namespace (指標命名空間) 中，輸入 **`MyNameSpace`**。因為這是新的命名空間，請務必選取 Create new (新建)。
9. 針對 Metric name (指標名稱)，輸入 **`BytesTransferred`**
10. 針對 Metric value (指標值)，輸入 **`$size`**。
11. 針對 Unit (單位)，選取Bytes (位元組)。
12. 針對 Dimension Name (維度名稱)，輸入 **`IP`**。

13. 針對 Dimension Value (維度值)，輸入 **$ip**，然後選擇 Next (下一步)。
14. 選擇 Create metric filter (建立指標篩選條件)。

使用建立此測量結果篩選 AWS CLI

在命令提示中，執行下列命令

```
aws logs put-metric-filter \
--log-group-name MyApp/access.log \
 --filter-name BytesTransferred \
 --filter-pattern '[ip, id, user, timestamp, request, status_code, size]' \
 --metric-transformations  \
 metricName=BytesTransferred,metricNamespace=MyNamespace,metricValue='$size'
```

```
aws logs put-metric-filter \
--log-group-name MyApp/access.log \
--filter-name BytesTransferred \
--filter-pattern '[ip, id, user, timestamp, request, status_code, size]' \
--metric-transformations  \
metricName=BytesTransferred,metricNamespace=MyNamespace,metricValue='$size',unit=Bytes,dimensio
$ip}}'
```

> **Note**
>
> 在此命令中，使用此格式指定多個維度。
>
> ```
> aws logs put-metric-filter \
> --log-group-name my-log-group-name \
> --filter-name my-filter-name \
> --filter-pattern 'my-filter-pattern' \
> --metric-transformations  \
> metricName=my-metric-name,metricNamespace=my-metric-namespace,metricValue=my-
> token,unit=unit,dimensions='{dimension1=$dim,dimension2=$dim2,dim3=$dim3}'
> ```

您可以在 put-log-event 呼叫中使用下列資料來測試此規則。如果您沒有在之前的範例中移除監控規則，此將會產生兩種不同的指標。

```
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
```

```
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /~test/ HTTP/1.1" 200 3
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:51:34 -0700] "GET /~test/index.html HTTP/1.1" 200 3
```

列出指標篩選條件

您可以列出日誌群組中所有指標篩選條件。

使用 CloudWatch 主控台列出量度篩選

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇 Log groups (日誌群組)。
3. 在日誌群組清單的內容窗格中，選擇在 Metric Filters (指標篩選條件) 欄中的篩選條件數。

   Log Groups > Filters for (日誌群組 > 篩選條件) 畫面會列出與日誌群組相關聯的所有指標篩選條件。

使用列示量度篩選 AWS CLI

在命令提示中，執行下列命令：

```
aws logs describe-metric-filters --log-group-name MyApp/access.log
```

下列為範例輸出：

```
{
    "metricFilters": [
        {
            "filterName": "HTTP404Errors",
            "metricTransformations": [
                {
                    "metricValue": "1",
                    "metricNamespace": "MyNamespace",
                    "metricName": "ApacheNotFoundErrorCount"
                }
            ],
            "creationTime": 1399277571078,
```

```
            "filterPattern": "[ip, id, user, timestamp, request, status_code=404,
 size]"
        }
    ]
}
```

刪除指標篩選條件

您可透過政策名稱和其所屬的日誌群組來辨識該政策。

使用 CloudWatch 主控台刪除量度篩選

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇 Log groups (日誌群組)。
3. 在內容窗格的 Metric Filter (指標篩選條件) 欄中，選擇日誌群組的指標篩選條件數。
4. 在 Metric Filters (指標篩選條件) 畫面下，針對您要刪除的篩選條件，選取其名稱右側的核取方塊。然後選擇 Delete (刪除)。
5. 出現確認提示時，請選擇刪除。

使用刪除測量結果篩選 AWS CLI

在命令提示中，執行下列命令：

```
aws logs delete-metric-filter --log-group-name MyApp/access.log \
 --filter-name MyFilterName
```

使用訂閱即時處理日誌資料

您可以使用訂閱從 CloudWatch 日誌存取日誌事件的即時摘要，並將其交付到其他服務，例如 Amazon Kinesis 串流、Amazon Data Firehose 串流，或用 AWS Lambda 於自訂處理、分析或載入到其他系統。日誌事件傳送至接收端服務時會以 Base64 編碼並以 gzip 格式壓縮。

若要開始訂閱日誌事件，請建立接收端資源 (例如 Kinesis Data Streams 串流)，事件將傳送到此。訂閱篩選器會定義篩選器模式，用於篩選哪些記錄事件會傳送至您的 AWS 資源，以及有關將相符記錄事件傳送至何處的資訊。

您可以在帳戶層級和記錄群組層級建立訂閱。每個帳戶可以有一個帳戶級訂閱過濾器。每個日誌群組最多有兩個相關聯的訂閱篩選條件。

> **Note**
>
> 如果目的地服務傳回可重試的錯誤，例如節流例外狀況或可重試的服務例外狀況 (例如 HTTP 5xx)，則 CloudWatch 記錄檔會持續重試傳遞最多 24 小時。 CloudWatch 如果錯誤是不可重試的錯誤，例如或，記錄檔不會嘗試重新傳送。 AccessDeniedException ResourceNotFoundException在這些情況下，訂閱篩選器會停用最多 10 分鐘，然後 CloudWatch Logs 會重試將記錄檔傳送至目的地。在此停用期間，會略過記錄檔。

CloudWatch 記錄檔也會產生有關將記錄事件轉送至訂閱的 CloudWatch 指標。如需詳細資訊，請參閱使用 CloudWatch 指標監控。

您也可以使用 CloudWatch 日誌訂閱將日誌資料以近乎即時的方式串流到 Amazon OpenSearch 服務叢集。如需詳細資訊，請參閱將 CloudWatch 日誌資料串流至 Amazon OpenSearch 服務。

只有標準記錄檔類別中的記錄群組才支援訂閱。如需記錄類別的詳細資訊，請參閱日誌類。

> **Note**
>
> 訂閱篩選器可能會批次處理記錄事件，以最佳化傳輸並減少對目的地進行呼叫的數量。不保證批次處理，但會盡可能使用。

目錄

- 記錄群組層級訂閱篩選器
- 帳戶層級訂閱過濾器
- 跨帳戶跨區域訂閱
- 預防混淆代理人
- 防止記錄遞迴

概念

每個訂閱篩選條件是由下列關鍵元素組成：

篩選條件模式

CloudWatch 記錄應如何解譯每個記錄事件中的資料的符號描述，以及限制傳遞至目的地 AWS 資源的篩選運算式。如需篩選條件模式語法的詳細資訊，請參閱 用於指標篩選條件、訂閱篩選條件、篩選條件日誌事件和 Live Tail 的篩選條件模式語法。

目的地 ARN

您要用作訂閱摘要目的地的 Kinesis 資料串流、Firehose 串流或 Lambda 函數的 Amazon 資源名稱 (ARN)。

角色 ARN

授予 CloudWatch 記錄檔必要許可的 IAM 角色，以將資料放入所選目的地。Lambda 目的地不需要此角色，因為 CloudWatch 日誌可以從 Lambda 函數本身的存取控制設定中取得必要的許可。

分佈

當目的地是 Amazon Kinesis Data Streams 中的串流時,用來將日誌資料分送到目的地的方法。在預設情況下，日誌資料是依日誌串流來分組的。如需進行更多分發，您可以將日誌資料隨機分組。

對於記錄群組層級訂閱，也會包含下列金鑰元素：

日誌群組名稱

要與訂閱篩選條件關聯的日誌群組。所有上傳到此日誌群組的日誌事件取決於訂閱篩選條件，符合篩選條件的日誌事件會傳送至負責接收相符日誌事件的目的地服務。

對於帳戶層級訂閱，也會包含下列金鑰元素：

選擇條件

用於選取已套用帳戶層級訂閱篩選器的記錄群組的條件。如果未指定此項，帳戶層級訂閱篩選器會套用至帳戶中的所有記錄群組。此欄位用於防止無限的記錄迴圈。如需有關無限記錄迴圈問題的詳細資訊，請參閱防止記錄遞迴。

選取準則的大小限制為 25 KB。

記錄群組層級訂閱篩選器

您可以將訂閱篩選器與 Kinesis Data Streams、Lambda 或 Firehose 搭配使用。透過訂閱篩選條件傳送至接收端服務的日誌是以 Base64 編碼，並以 gzip 格式壓縮。

您可以使用篩選條件和模式語法來搜尋日誌資料。

範例

- 範例 1：訂閱篩選條件與 Kinesis Data Streams 搭配使用
- 範例 2：訂閱篩選器 AWS Lambda
- 範例 3：使用 Amazon 資料 Firehose 的訂閱篩選器

範例 1：訂閱篩選條件與 Kinesis Data Streams 搭配使用

下列範例會將訂閱篩選器與包含 AWS CloudTrail 事件的記錄群組產生關聯。訂閱篩選器會將「根」AWS 登入資料所做的所有記錄活動提供給 Kinesis Data Streams 中稱為「」的串流RootAccess。如需如何將 AWS CloudTrail 事件傳送至 CloudWatch 記錄檔的相關資訊，請參閱《AWS CloudTrail 使用指南》中的〈將 CloudTrail 事件傳送至 CloudWatch 記錄檔〉。

> **Note**
>
> 在您建立 串流前，計算將產生的日誌資料磁碟區。請務必使用足夠碎片建立 串流，以處理此磁碟區。如果串流沒有足夠的碎片，日誌串流將受到限制。如需更多有關串流磁碟區限制的資訊，請參閱配額與限制。
> 限流的交付項目會持續重試，時間長達 24 小時。24 小時後，失敗的交付項目就會捨棄。
> 若要降低限流風險，您可以採取下步驟：
>
> - 指定使random用distribution或建立訂閱篩選器的 PutSubscriptionFilter時間 put-subscription-filter。根據預設，串流篩選器分佈是依照記錄資料流，這可能會造成節流。

- 使用 CloudWatch 指標監控串流。如此可協助您找出任何限流，並根據實際情況調整您的組態。例如，DeliveryThrottling量度可用來追蹤將資料轉送至訂閱目的地時，已限制 CloudWatch 記錄的記錄事件數目。如需監控的詳細資訊，請參閱使用 CloudWatch 指標監控。
- 在 Kinesis Data Streams 中為您的串流使用隨需容量模式。隨需模式會在您的工作負載上升或下降時，立即為您的工作負載調整所需的容量。有關隨需容量模式的詳細資訊，請參閱隨需模式。
- 限制訂 CloudWatch 閱篩選器模式，使其符合 Kinesis 資料串流中串流的容量。如果您傳送太多資料至串流，您可能需要減少篩選條件大小或調整篩選條件標準。

建立 Kinesis Data Streams 的訂閱篩選條件

1. 使用下列命令建立目的地 串流：

```
$ C:\>  aws kinesis create-stream --stream-name "RootAccess" --shard-count 1
```

2. 等到 串流成為作用中 (這可能需要花費幾分鐘)。您可以使用下列 Kinesis 資料串流描述串流命令來檢查。StreamDescription StreamStatus財產。此外，請注意 StreamDescription.Streamarn 值，因為您在稍後的步驟中將需要它：

```
aws kinesis describe-stream --stream-name "RootAccess"
```

下列為範例輸出：

```
{
    "StreamDescription": {
        "StreamStatus": "ACTIVE",
        "StreamName": "RootAccess",
        "StreamARN": "arn:aws:kinesis:us-east-1:123456789012:stream/RootAccess",
        "Shards": [
            {
                "ShardId": "shardId-000000000000",
                "HashKeyRange": {
                    "EndingHashKey": "340282366920938463463374607431768211455",
                    "StartingHashKey": "0"
                },
                "SequenceNumberRange": {
                    "StartingSequenceNumber":
```

```
                    "495511352186888184566795038319814587845913527021815726l0"
                }
            }
        ]
    }
}
```

3. 建立 IAM 角色，以授與 CloudWatch Log 權限，以將資料放入串流。首先，您將需要在檔案中建立信任政策 (例如，~/TrustPolicyForCWL-Kinesis.json)。請使用文字編輯器來建立此政策。請勿使用 IAM 主控台建立這一項。

   此政策包含 aws:SourceArn 全域條件內容金鑰，以協助預防混淆代理人安全問題。如需詳細資訊，請參閱 預防混淆代理人。

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.amazonaws.com" },
    "Action": "sts:AssumeRole",
    "Condition": {
        "StringLike": { "aws:SourceArn": "arn:aws:logs:region:123456789012:*" }
     }
   }
}
```

4. 使用 create-role 命令來建立 IAM 角色，並指定信任政策檔案。請注意傳回的 Role.Arn 值，因您將在後續步驟需要此值：

```
aws iam create-role --role-name CWLtoKinesisRole --assume-role-policy-document
 file://~/TrustPolicyForCWL-Kinesis.json
```

   以下為輸出範例。

```
{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.amazonaws.com"
                },
```

```
                "Condition": {
                    "StringLike": {
                        "aws:SourceArn": { "arn:aws:logs:region:123456789012:*" }
                    }
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisRole",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/CWLtoKinesisRole"
    }
}
```

5. 建立權限原則，以定義 CloudWatch Logs 可以對您的帳戶執行的動作。首先，您將需要在檔案中建立許可政策 (例如，~/PermissionsForCWL-Kinesis.json)。請使用文字編輯器來建立此政策。請勿使用 IAM 主控台建立這一項。

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:region:123456789012:stream/RootAccess"
    }
  ]
}
```

6. 使用下列put-role-policy命令將權限原則與角色產生關聯：

```
aws iam put-role-policy  --role-name CWLtoKinesisRole  --policy-name Permissions-
Policy-For-CWL  --policy-document file://~/PermissionsForCWL-Kinesis.json
```

7. 串流處於作用中狀態且您已建立 IAM 角色之後，您可以建立 CloudWatch 記錄訂閱篩選器。訂閱篩選條件會立即開始將即時日誌資料從所選的日誌群組傳送到 串流：

```
aws logs put-subscription-filter \
    --log-group-name "CloudTrail/logs" \
    --filter-name "RootAccess" \
    --filter-pattern "{$.userIdentity.type = Root}" \
    --destination-arn "arn:aws:kinesis:region:123456789012:stream/RootAccess" \
```

```
    --role-arn "arn:aws:iam::123456789012:role/CWLtoKinesisRole"
```

8. 設定訂閱篩選器之後， CloudWatch Logs 會將符合篩選器模式的所有傳入記錄事件轉寄至您的串流。您可以抓取 Kinesis Data Streams 碎片疊代器，並使用 Kinesis Data Streams get-records 命令來擷取一些 Kinesis Data Streams 記錄，以確認確有其事：

```
aws kinesis get-shard-iterator --stream-name RootAccess --shard-id
 shardId-000000000000 --shard-iterator-type TRIM_HORIZON
```

```
{
    "ShardIterator":
    "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL
+wev+e2P4djJg4L9wmXKvQYoE+rMUiFq
+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f+OIK8zM5My8ID
+g6rMo7UKWeI4+IWiK2OSh0uP"
}
```

```
aws kinesis get-records --limit 10 --shard-iterator "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL
+wev+e2P4djJg4L9wmXKvQYoE+rMUiFq
+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f+OIK8zM5My8ID
+g6rMo7UKWeI4+IWiK2OSh0uP"
```

請注意，您可能需要進行幾次此呼叫，Kinesis Data Streams 才會開始傳回資料。

您應該預期會看到含一系列的記錄的回應。Kinesis Data Streams 記錄中的 資料屬性採用 Base64 編碼並以 gzip 格式壓縮。您可以使用以下 Unix 命令來透過命令列檢查原始資料：

```
echo -n "<Content of Data>" | base64 -d | zcat
```

Base64 解碼和解壓縮資料是以 JSON 形式並以下列結構進行格式化：

```
{
    "owner": "111111111111",
    "logGroup": "CloudTrail/logs",
    "logStream": "111111111111_CloudTrail/logs_us-east-1",
    "subscriptionFilters": [
        "Destination"
    ],
```

```
    "messageType": "DATA_MESSAGE",
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424298221568",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221569",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221570",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        }
    ]
}
```

在上述資料結構的關鍵元素如下：

owner

原始記錄檔資料的 AWS 帳戶 ID。

logGroup

原始日誌資料的日誌群組名稱。

logStream

原始日誌資料的日誌串流名稱。

subscriptionFilters

與原始日誌資料相符的訂閱篩選條件名稱清單。

messageType

資料訊息將使用「DATA_MESSAGE」類型。有時，記 CloudWatch 錄檔可能會發出具有「CONTROL_MESSAGE」類型的 Kinesis Data Streams 記錄，主要用於檢查目的地是否可連線。

logEvents

實際的日誌資料，以一系列的日誌事件記錄呈現。「id」屬性是每個記錄事件的唯一識別符。

範例 2：訂閱篩選器 AWS Lambda

在此示例中，您將創建一個 CloudWatch 日誌訂閱過濾器，該過濾器將日誌數據發送到您的 AWS Lambda 函數。

> **Note**
>
> 建立 Lambda 函數前，請計算將產生的日誌資料量。請務必建立可以處理此磁碟區的函數。如果函數沒有足夠的磁碟區，日誌串流將受到限制。如需 Lambda 限制的詳細資訊，請參閱 AWS Lambda 限制。

建立 Lambda 的訂閱篩選條件

1. 建立 AWS Lambda 函數。

   確保您已設定 Lambda 執行角色。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的步驟 2.2：建立 IAM 角色 (執行角色)。

2. 開啟文字編輯器，並建立名為 `helloWorld.js` 的檔案，內含下列內容：

```
var zlib = require('zlib');
exports.handler = function(input, context) {
    var payload = Buffer.from(input.awslogs.data, 'base64');
    zlib.gunzip(payload, function(e, result) {
        if (e) {
            context.fail(e);
        } else {
            result = JSON.parse(result.toString());
            console.log("Event Data:", JSON.stringify(result, null, 2));
            context.succeed();
```

```
        }
    });
};
```

3. 壓縮檔案 helloWorld.js，並以名稱 `helloWorld.zip` 將其儲存。
4. 使用下列命令，其中角色是您在第一個步驟中設定的 Lambda 執行角色：

```
aws lambda create-function \
    --function-name helloworld \
    --zip-file fileb://file-path/helloWorld.zip \
    --role lambda-execution-role-arn \
    --handler helloWorld.handler \
    --runtime nodejs12.x
```

5. 授予 CloudWatch 記錄執行函數的權限。使用下列命令，將預留位置帳戶取代為您自己的帳戶且將預留位置日誌群組取代為要處理的日誌群組：

```
aws lambda add-permission \
    --function-name "helloworld" \
    --statement-id "helloworld" \
    --principal "logs.amazonaws.com" \
    --action "lambda:InvokeFunction" \
    --source-arn "arn:aws:logs:region:123456789123:log-group:TestLambda:*" \
    --source-account "123456789012"
```

6. 使用下列命令建立訂閱篩選條件，將預留位置帳戶取代為您自己的帳戶且將預留位置日誌群組取代為要處理的日誌群組：

```
aws logs put-subscription-filter \
    --log-group-name myLogGroup \
    --filter-name demo \
    --filter-pattern "" \
    --destination-arn arn:aws:lambda:region:123456789123:function:helloworld
```

7. (選用) 使用範例日誌事件進行測試。在命令提示字元中執行下列命令，這會將簡單日誌訊息放置到訂閱的串流。

   若要查看 Lambda 函數的輸出，請導覽至 Lambda 函數，其中您將會在 /aws/lambda/helloworld 中看到輸出：

```
aws logs put-log-events --log-group-name myLogGroup --log-stream-name stream1 --
log-events "[{\"timestamp\":<CURRENT TIMESTAMP MILLIS> , \"message\": \"Simple
 Lambda Test\"}]"
```

預期會看到含一系列 Lambda 的回應。Lambda 記錄中的 Data (資料) 屬性是以 Base64 編碼並以 gzip 格式壓縮。Lambda 收到的實際酬載為以下格式：`{ "awslogs": {"data": "BASE64ENCODED_GZIP_COMPRESSED_DATA"} }`。您可以從命令列使用以下 Unix 命令來檢視原始資料：

```
echo -n "<BASE64ENCODED_GZIP_COMPRESSED_DATA>" | base64 -d | zcat
```

Base64 解碼和解壓縮資料是以 JSON 形式並以下列結構進行格式化：

```
{
    "owner": "123456789012",
    "logGroup": "CloudTrail",
    "logStream": "123456789012_CloudTrail_us-east-1",
    "subscriptionFilters": [
        "Destination"
    ],
    "messageType": "DATA_MESSAGE",
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424298221568",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221569",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221570",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        }
```

```
    ]
}
```

在上述資料結構的關鍵元素如下：

owner

原始記錄檔資料的 AWS 帳戶 ID。

logGroup

原始日誌資料的日誌群組名稱。

logStream

原始日誌資料的日誌串流名稱。

subscriptionFilters

與原始日誌資料相符的訂閱篩選條件名稱清單。

messageType

資料訊息將使用「DATA_MESSAGE」類型。有時 CloudWatch 日誌可能會發出具有「CONTROL_MESSAGE」類型的 Lambda 記錄，主要用於檢查目標是否可訪問。

logEvents

實際的日誌資料，以一系列的日誌事件記錄呈現。「id」屬性是每個記錄事件的唯一識別符。

範例 3：使用 Amazon 資料 Firehose 的訂閱篩選器

在此範例中，您將建立 CloudWatch 日誌訂閱，將符合定義篩選器的任何傳入日誌事件傳送到 Amazon Data Firehose 交付串流。從 CloudWatch 日誌傳送到 Amazon 資料 Firehose 的資料已使用 gzip 等級 6 壓縮進行壓縮，因此您不需要在 Firehose 交付串流中使用壓縮。然後，您可以使用 Firehose 中的解壓縮功能來自動解壓縮記錄檔。如需詳細資訊，請參閱使 CloudWatch 用記錄寫入 Kinesis Data Firehose。

> **Note**
>
> 建立 Firehose 串流之前，請先計算將產生的記錄資料量。請務必建立可處理此磁碟區的 Firehose 串流。如果串流無法處理磁碟區、日誌串流將受到限制。如需有關 Firehose 串流音量限制的詳細資訊，請參閱 Amazon 資料 Firehose 資料限制。

若要建立 Firehose 的訂閱篩選器

1. 建立 Amazon Simple Storage Service (Amazon S3) 儲存貯體。我們建議您使用專門為 CloudWatch Logs 建立的值區。不過，如果您想要使用現有的儲存貯體，請跳到步驟 2。

   執行以下命令，將預留位置 Region 換成您想要使用的區域：

   ```
   aws s3api create-bucket --bucket my-bucket --create-bucket-configuration
    LocationConstraint=region
   ```

   下列為範例輸出：

   ```
   {
       "Location": "/my-bucket"
   }
   ```

2. 建立 IAM 角色，以授與 Amazon 資料 Firehose 將資料放入您的 Amazon S3 儲存貯體的權限。

   如需詳細資訊，請參閱 Amazon 資料 Firehose 開發人員指南中的使用 Amazon 資料 Firehose 控制存取。

   首先，請按如下所示，使用文字編輯器來建立檔案 ~/TrustPolicyForFirehose.json 中的信任政策：

   ```
   {
     "Statement": {
       "Effect": "Allow",
       "Principal": { "Service": "firehose.amazonaws.com" },
       "Action": "sts:AssumeRole"
       }
   }
   ```

3. 使用 create-role 命令來建立 IAM 角色，並指定信任政策檔案。請注意傳回的 Role.Arn 值，因您將在後續步驟需要此值：

   ```
   aws iam create-role \
    --role-name FirehosetoS3Role \
    --assume-role-policy-document file://~/TrustPolicyForFirehose.json

   {
       "Role": {
   ```

```
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "firehose.amazonaws.com"
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "FirehosetoS3Role",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/FirehosetoS3Role"
    }
}
```

4. 建立權限原則，以定義 Firehose 可以對您的帳戶執行的動作。首先，使用文字編輯器來建立檔案 ~/PermissionsForFirehose.json 中的許可政策：

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "s3:AbortMultipartUpload",
          "s3:GetBucketLocation",
          "s3:GetObject",
          "s3:ListBucket",
          "s3:ListBucketMultipartUploads",
          "s3:PutObject" ],
      "Resource": [
          "arn:aws:s3:::my-bucket",
          "arn:aws:s3:::my-bucket/*" ]
    }
  ]
}
```

5. 使用下列 put-role-policy命令將權限原則與角色產生關聯：

```
aws iam put-role-policy --role-name FirehosetoS3Role --policy-name Permissions-Policy-For-Firehose --policy-document file://~/PermissionsForFirehose.json
```

6. 如下所示建立目的地 Firehose 傳遞串流，並以您建立的角色和值區 ARN 取代 Ro leARN 和 BuckeTARN 的預留位置值：

```
aws firehose create-delivery-stream \
   --delivery-stream-name 'my-delivery-stream' \
   --s3-destination-configuration \
  '{"RoleARN": "arn:aws:iam::123456789012:role/FirehosetoS3Role", "BucketARN":
 "arn:aws:s3:::my-bucket"}'
```

請注意，Firehose 會針對交付的 Amazon S3 物件，自動使用 YYYY/MM/DD/HH UTC 時間格式的前置詞。您可以指定在時間格式前綴前要新增的額外前綴。如果字首結尾是斜線 (/)，則會在 Amazon S3 儲存貯體中顯示為資料夾。

7. 等到串流成為作用中 (這可能需要花費幾分鐘)。您可以使用「Firehose」describe-delivery-stream指令來檢查。DeliveryStreamDescription DeliveryStreamStatus財產。此外，請注意 DeliveryStreamDescription. DeliveryStreamARN 值，因為您將在後面的步驟中需要它：

```
aws firehose describe-delivery-stream --delivery-stream-name "my-delivery-stream"
{
    "DeliveryStreamDescription": {
        "HasMoreDestinations": false,
        "VersionId": "1",
        "CreateTimestamp": 1446075815.822,
        "DeliveryStreamARN": "arn:aws:firehose:us-
east-1:123456789012:deliverystream/my-delivery-stream",
        "DeliveryStreamStatus": "ACTIVE",
        "DeliveryStreamName": "my-delivery-stream",
        "Destinations": [
            {
                "DestinationId": "destinationId-000000000001",
                "S3DestinationDescription": {
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "RoleARN": "delivery-stream-role",
                    "BucketARN": "arn:aws:s3:::my-bucket",
                    "BufferingHints": {
                        "IntervalInSeconds": 300,
                        "SizeInMBs": 5
                    }
```

```
                }
            }
        ]
    }
}
```

8. 建立 IAM 角色，以授與 CloudWatch 記錄檔權限，以將資料放入 Firehose 交付串流。首先，使用文字編輯器來建立檔案 ~/TrustPolicyForCWL.json 中的信任政策：

   此政策包含 aws:SourceArn 全域條件內容金鑰，以協助預防混淆代理人安全問題。如需詳細資訊，請參閱 預防混淆代理人。

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.amazonaws.com" },
    "Action": "sts:AssumeRole",
    "Condition": {
         "StringLike": {
             "aws:SourceArn": "arn:aws:logs:region:123456789012:*"
         }
     }
  }
}
```

9. 使用 create-role 命令來建立 IAM 角色，並指定信任政策檔案。請注意傳回的 Role.Arn 值，因您將在後續步驟需要此值：

```
aws iam create-role \
--role-name CWLtoKinesisFirehoseRole \
--assume-role-policy-document file://~/TrustPolicyForCWL.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.amazonaws.com"
                },
                "Condition": {
                     "StringLike": {
```

```
                        "aws:SourceArn": "arn:aws:logs:region:123456789012:*"
                    }
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisFirehoseRole",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/CWLtoKinesisFirehoseRole"
    }
}
```

10. 建立權限原則，以定義 CloudWatch Logs 可以對您的帳戶執行的動作。首先，使用文字編輯器來建立許可政策檔案 (例如，~/PermissionsForCWL.json)：

```
{
    "Statement":[
      {
        "Effect":"Allow",
        "Action":["firehose:PutRecord"],
        "Resource":[
            "arn:aws:firehose:region:account-id:deliverystream/delivery-stream-
name"]
      }
    ]
}
```

11. 使用以下 put-role-policy命令將權限原則與角色產生關聯：

```
aws iam put-role-policy --role-name CWLtoKinesisFirehoseRole --policy-
name Permissions-Policy-For-CWL --policy-document file://~/PermissionsForCWL.json
```

12. Amazon Data Firehose 交付串流處於使用中狀態，且您已建立 IAM 角色之後，您可以建立 CloudWatch 日誌訂閱篩選器。訂閱篩選器會立即啟動從所選日誌群組到 Amazon Data Firehose 交付串流的即時日誌資料流程：

```
aws logs put-subscription-filter \
    --log-group-name "CloudTrail" \
    --filter-name "Destination" \
    --filter-pattern "{$.userIdentity.type = Root}" \
    --destination-arn "arn:aws:firehose:region:123456789012:deliverystream/my-
delivery-stream" \
```

```
    --role-arn "arn:aws:iam::123456789012:role/CWLtoKinesisFirehoseRole"
```

13. 設定訂閱篩選器後，CloudWatch 日誌會將符合篩選器模式的所有傳入日誌事件轉寄至 Amazon Data Firehose 交付串流。您的資料將會根據 Amazon 資料 Firehose 交付串流上設定的時間緩衝區間隔開始顯示在 Amazon S3 中。一旦經過足夠的時間，您就可以檢查 Amazon S3 儲存貯體來驗證資料。

```
aws s3api list-objects --bucket 'my-bucket' --prefix 'firehose/'
{
    "Contents": [
        {
            "LastModified": "2015-10-29T00:01:25.000Z",
            "ETag": "\"a14589f8897f4089d3264d9e2d1f1610\"",
            "StorageClass": "STANDARD",
            "Key": "firehose/2015/10/29/00/my-delivery-stream-2015-10-29-00-01-21-
a188030a-62d2-49e6-b7c2-b11f1a7ba250",
            "Owner": {
                "DisplayName": "cloudwatch-logs",
                "ID": "1ec9cf700ef6be062b19584e0b7d84ecc19237f87b5"
            },
            "Size": 593
        },
        {
            "LastModified": "2015-10-29T00:35:41.000Z",
            "ETag": "\"a7035b65872bb2161388ffb63dd1aec5\"",
            "StorageClass": "STANDARD",
            "Key": "firehose/2015/10/29/00/my-delivery-
stream-2015-10-29-00-35-40-7cc92023-7e66-49bc-9fd4-fc9819cc8ed3",
            "Owner": {
                "DisplayName": "cloudwatch-logs",
                "ID": "1ec9cf700ef6be062b19584e0b7d84ecc19237f87b6"
            },
            "Size": 5752
        }
    ]
}
```

```
aws s3api get-object --bucket 'my-bucket' --key 'firehose/2015/10/29/00/my-
delivery-stream-2015-10-29-00-01-21-a188030a-62d2-49e6-b7c2-b11f1a7ba250'
 testfile.gz

{
```

```
    "AcceptRanges": "bytes",
    "ContentType": "application/octet-stream",
    "LastModified": "Thu, 29 Oct 2015 00:07:06 GMT",
    "ContentLength": 593,
    "Metadata": {}
}
```

在 Amazon S3 物件中的資料會以 gzip 格式壓縮。您可以使用以下 Unix 命令來透過命令列檢查原始資料：

```
zcat testfile.gz
```

帳戶層級訂閱過濾器

Important

使用訂閱篩選器可能會造成無限遞迴迴圈的風險，如果未解決，擷取計費可能會大幅增加。為了降低此風險，建議您在帳戶層級訂閱篩選器中使用選取準則，排除從屬於訂閱傳遞工作流程一部分之資源擷取記錄資料的記錄群組。如需有關此問題以及判斷要排除哪些記錄群組的詳細資訊，請參閱防止記錄遞迴。

您可以設定帳戶層級訂閱政策，其中包含帳戶中的記錄群組子集。帳戶訂閱政策可與 Kinesis Data Streams、Lambda 或 Firehose 搭配使用。透過帳戶層級訂閱原則傳送至接收服務的記錄會以 base64 編碼，並以 gzip 格式壓縮。

Note

若要檢視您帳戶中所有訂閱篩選原則的清單，請使用具有`--policy-type`參數值`SUBSCRIPTION_FILTER_POLICY`的`describe-account-policies`命令。如需詳細資訊，請參閱 describe-account-policies¶。

範例

- 範例 1：訂閱篩選條件與 Kinesis Data Streams 搭配使用
- 範例 2：訂閱篩選器 AWS Lambda
- 範例 3：使用 Amazon 資料 Firehose 的訂閱篩選器

範例 1：訂閱篩選條件與 Kinesis Data Streams 搭配使用

建立 Kinesis Data Streams 資料串流以搭配帳戶層級訂閱政策使用之前，請先計算將產生的記錄資料量。請務必使用足夠碎片建立 串流，以處理此磁碟區。如果串流沒有足夠的碎片，則會進行節流。如需串流磁碟區限制的詳細資訊，請參閱 Kinesis Data Streams 說明文件中的配額和限制。

> **Warning**
>
> 由於多個記錄群組的記錄事件會轉寄至目的地，因此存在限制的風險。限流的交付項目會持續重試，時間長達 24 小時。24 小時後，失敗的交付項目就會捨棄。
> 若要降低限流風險，您可以採取下步驟：
>
> - 使用 CloudWatch 指標監控您的 Kinesis Data Streams。這有助於您識別節流並相應地調整配置。例如，`DeliveryThrottling`量度會追蹤將資料轉送至訂閱目的地時，已限制 CloudWatch 記錄檔的記錄事件數目。如需詳細資訊，請參閱 使用 CloudWatch 指標監控。
> - 在 Kinesis Data Streams 中為您的串流使用隨需容量模式。隨需模式會在您的工作負載上升或下降時，立即為您的工作負載調整所需的容量。如需詳細資訊，請參閱隨選模式。
> - 限制您的 CloudWatch Log 訂閱篩選器模式，使其符合 Kinesis 資料串流中串流的容量。如果您傳送太多資料至串流，您可能需要減少篩選條件大小或調整篩選條件標準。

下列範例使用帳戶層級訂閱政策，將所有記錄事件轉寄至 Kinesis Data Streams 中的串流。篩選器模式會將任何記錄事件與文字相符，`Test`並將它們轉送至 Kinesis 資料串流中的串流。

建立 Kinesis Data Streams 的帳戶層級訂閱政策

1. 使用下列命令建立目的地 串流：

```
$ C:\>  aws kinesis create-stream –stream-name "TestStream" –shard-count 1
```

2. 等待幾分鐘，讓串流變為作用中狀態。您可以使用描述串流命令來檢查資料流是否處於作用中狀態。StreamDescription StreamStatus財產。

```
aws kinesis describe-stream --stream-name "TestStream"
```

下列為範例輸出：

```
{
    "StreamDescription": {
```

```
        "StreamStatus": "ACTIVE",
        "StreamName": "TestStream",
        "StreamARN": "arn:aws:kinesis:region:123456789012:stream/TestStream",
        "Shards": [
            {
                "ShardId": "shardId-000000000000",
                "HashKeyRange": {
                    "EndingHashKey": "EXAMPLE84634633746074317682114555",
                    "StartingHashKey": "0"
                },
                "SequenceNumberRange": {
                    "StartingSequenceNumber":
                    "EXAMPLE6888184566795038319814587845913527021815726l0"
                }
            }
        ]
    }
}
```

3. 建立 IAM 角色，以授與 CloudWatch Log 權限，以將資料放入串流。首先，您將需要在檔案中建立信任政策 (例如，~/TrustPolicyForCWL-Kinesis.json)。請使用文字編輯器來建立此政策。

   此政策包含 aws:SourceArn 全域條件內容金鑰，以協助預防混淆代理人安全問題。如需詳細資訊，請參閱 預防混淆代理人。

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.amazonaws.com" },
    "Action": "sts:AssumeRole",
    "Condition": {
        "StringLike": { "aws:SourceArn": "arn:aws:logs:region:123456789012:*" }
     }
   }
}
```

4. 使用 create-role 命令來建立 IAM 角色，並指定信任政策檔案。請注意傳回的 Role.Arn 值，因您將在後續步驟需要此值：

```
aws iam create-role --role-name CWLtoKinesisRole --assume-role-policy-document
 file://~/TrustPolicyForCWL-Kinesis.json
```

以下為輸出範例。

```
{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.amazonaws.com"
                },
                "Condition": {
                    "StringLike": {
                        "aws:SourceArn": { "arn:aws:logs:region:123456789012:*" }
                    }
                }
            }
        },
        "RoleId": "EXAMPLE450GAB4HC5F431",
        "CreateDate": "2023-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisRole",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/CWLtoKinesisRole"
    }
}
```

5. 建立權限原則，以定義 CloudWatch Logs 可以對您的帳戶執行的動作。首先，您將需要在檔案中建立許可政策 (例如，~/PermissionsForCWL-Kinesis.json)。請使用文字編輯器來建立此政策。請勿使用 IAM 主控台來建立它。

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:region:123456789012:stream/TestStream"
    }
  ]
}
```

6. 使用下列put-role-policy命令將權限原則與角色產生關聯：

```
aws iam put-role-policy  --role-name CWLtoKinesisRole  --policy-name Permissions-
Policy-For-CWL  --policy-document file://~/PermissionsForCWL-Kinesis.json
```

7. 串流處於作用中狀態且您已建立 IAM 角色之後，您可以建立 CloudWatch 記錄訂閱篩選政策。此原則會立即啟動串流的即時記錄資料流。在此範例中，所有包含該字串的記錄事件ERROR都會串流處理，但名為LogGroupToExclude1和LogGroupToExclude2的記錄群組中的事件除外。

```
aws logs put-account-policy \
    --policy-name "ExamplePolicy" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document '{"RoleArn":"arn:aws:iam::123456789012:role/
CWLtoKinesisRole", "DestinationArn":"arn:aws:kinesis:region:123456789012:stream/
TestStream", "FilterPattern": "Test", "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

8. 設定訂閱篩選器之後， CloudWatch Logs 會將符合篩選器模式和選取條件的所有傳入記錄事件轉寄至串流。

   此selection-criteria欄位是選擇性欄位，但對於排除可能導致訂閱篩選器無限記錄遞迴的記錄群組而言非常重要。如需有關此問題以及判斷要排除哪些記錄群組的詳細資訊，請參閱防止記錄遞迴。目前，NOT IN 是唯一支援的運算子selection-criteria。

   您可以使用 Kinesis 資料串流碎片迭代器，並使用 Kinesis Data Streams get-records 命令擷取一些 Kinesis Data Streams 記錄，以驗證記錄事件的流程：

```
aws kinesis get-shard-iterator --stream-name TestStream --shard-id
 shardId-000000000000 --shard-iterator-type TRIM_HORIZON
```

```
{
    "ShardIterator":
    "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL
+wev+e2P4djJg4L9wmXKvQYoE+rMUiFq
+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f+OIK8zM5My8ID
+g6rMo7UKWeI4+IWiK2OSh0uP"
}
```

```
aws kinesis get-records --limit 10 --shard-iterator "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL
+wev+e2P4djJg4L9wmXKvQYoE+rMUiFq
+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f+OIK8zM5My8ID
+g6rMo7UKWeI4+IWiK2OSh0uP"
```

Kinesis 資料串流開始傳回資料之前，您可能需要使用這個命令幾次。

您應該預期會看到含一系列的記錄的回應。Kinesis Data Streams 記錄中的 資料屬性採用 Base64 編碼並以 gzip 格式壓縮。您可以使用以下 Unix 命令來透過命令列檢查原始資料：

```
echo -n "<Content of Data>" | base64 -d | zcat
```

Base64 解碼和解壓縮資料是以 JSON 形式並以下列結構進行格式化：

```
{
    "messageType": "DATA_MESSAGE",
    "owner": "123456789012",
    "logGroup": "Example1",
    "logStream": "logStream1",
    "subscriptionFilters": [
        "ExamplePolicy"
    ],
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424298221568",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221569",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221570",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
```

```
        }
    ],
    "policyLevel": "ACCOUNT_LEVEL_POLICY"
}
```

數據結構中的關鍵要素如下：

messageType

資料訊息將使用「DATA_MESSAGE」類型。有時，記 CloudWatch 錄檔可能會發出具有「CONTROL_MESSAGE」類型的 Kinesis Data Streams 記錄，主要用於檢查目的地是否可連線。

owner

原始記錄檔資料的 AWS 帳戶 ID。

logGroup

原始日誌資料的日誌群組名稱。

logStream

原始日誌資料的日誌串流名稱。

subscriptionFilters

與原始日誌資料相符的訂閱篩選條件名稱清單。

logEvents

實際的日誌資料，以一系列的日誌事件記錄呈現。「id」屬性是每個記錄事件的唯一識別符。

政策層級

強制執行策略的層級。「帳戶層級 _ 政策」是用於帳戶層級的`policyLevel`訂閱過濾策略。

範例 2：訂閱篩選器 AWS Lambda

在此示例中，您將創建 CloudWatch Logs 帳戶級訂閱過濾策略，該策略將日誌數據發送到您 AWS Lambda 的函數。

> **Warning**
>
> 建立 Lambda 函數前，請計算將產生的日誌資料量。請務必建立可以處理此磁碟區的函數。如果函數無法處理磁碟區，則會限制記錄串流。因為所有記錄群組或帳戶記錄群組子集的記錄事件都會轉送至目的地，因此可能會有限制的風險。如需 Lambda 限制的詳細資訊，請參閱 AWS Lambda 限制。

若要建立 Lambda 的帳戶層級訂閱篩選政策

1. 建立 AWS Lambda 函數。

   確保您已設定 Lambda 執行角色。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的步驟 2.2：建立 IAM 角色 (執行角色)。

2. 開啟文字編輯器，並建立名為 `helloWorld.js` 的檔案，內含下列內容：

```
var zlib = require('zlib');
exports.handler = function(input, context) {
    var payload = Buffer.from(input.awslogs.data, 'base64');
    zlib.gunzip(payload, function(e, result) {
        if (e) {
            context.fail(e);
        } else {
            result = JSON.parse(result.toString());
            console.log("Event Data:", JSON.stringify(result, null, 2));
            context.succeed();
        }
    });
};
```

3. 壓縮檔案 helloWorld.js，並以名稱 `helloWorld.zip` 將其儲存。
4. 使用下列命令，其中角色是您在第一個步驟中設定的 Lambda 執行角色：

```
aws lambda create-function \
    --function-name helloworld \
    --zip-file fileb://file-path/helloWorld.zip \
    --role lambda-execution-role-arn \
    --handler helloWorld.handler \
    --runtime nodejs18.x
```

5. 授予 CloudWatch 記錄執行函數的權限。使用以下命令，用您自己的帳戶替換佔位符帳戶。

```
aws lambda add-permission \
    --function-name "helloworld" \
    --statement-id "helloworld" \
    --principal "logs.amazonaws.com" \
    --action "lambda:InvokeFunction" \
    --source-arn "arn:aws:logs:region:123456789012:log-group:*" \
    --source-account "123456789012"
```

6. 使用下列命令建立帳戶層級訂閱篩選原則，並以您自己的帳戶取代預留位置帳戶。在此範例中，所有包含該字串的記錄事件ERROR都會串流處理，但名為LogGroupToExclude1和LogGroupToExclude2的記錄群組中的事件除外。

```
aws logs put-account-policy \
    --policy-name "ExamplePolicyLambda" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document
 '{"DestinationArn":"arn:aws:lambda:region:123456789012:function:helloWorld",
 "FilterPattern": "Test", "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

設定訂閱篩選器之後， CloudWatch Logs 會將符合篩選器模式和選取條件的所有傳入記錄事件轉寄至串流。

此selection-criteria欄位是選擇性欄位，但對於排除可能導致訂閱篩選器無限記錄遞迴的記錄群組而言非常重要。如需有關此問題以及判斷要排除哪些記錄群組的詳細資訊，請參閱防止記錄遞迴。目前，NOT IN 是唯一支援的運算子selection-criteria。

7. (選用) 使用範例日誌事件進行測試。在命令提示字元中執行下列命令，這會將簡單日誌訊息放置到訂閱的串流。

若要查看 Lambda 函數的輸出，請導覽至 Lambda 函數，其中您將會在 /aws/lambda/helloworld 中看到輸出：

```
aws logs put-log-events --log-group-name Example1 --log-stream-name logStream1 --
log-events "[{\"timestamp\":CURRENT TIMESTAMP MILLIS , \"message\": \"Simple Lambda
 Test\"}]"
```

預期會看到含一系列 Lambda 的回應。Lambda 記錄中的 Data (資料) 屬性是以 Base64 編碼並以 gzip 格式壓縮。Lambda 收到的實際酬載為以下格式：`{ "awslogs": {"data": "BASE64ENCODED_GZIP_COMPRESSED_DATA"} }`。您可以從命令列使用以下 Unix 命令來檢視原始資料：

```
echo -n "<BASE64ENCODED_GZIP_COMPRESSED_DATA>" | base64 -d | zcat
```

Base64 解碼和解壓縮資料是以 JSON 形式並以下列結構進行格式化：

```
{
    "messageType": "DATA_MESSAGE",
    "owner": "123456789012",
    "logGroup": "Example1",
    "logStream": "logStream1",
    "subscriptionFilters": [
        "ExamplePolicyLambda"
    ],
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424298221568",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221569",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221570",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        }
    ],
    "policyLevel": "ACCOUNT_LEVEL_POLICY"
}
```

> **Note**
>
> 帳戶層級訂閱篩選器不會套用至目的地 Lambda 函數的日誌群組。這是為了防止可能導致擷取計費增加的無限記錄遞迴。如需此問題的詳細資訊，請參閱防止記錄遞迴。

數據結構中的關鍵要素如下：

messageType

資料訊息將使用「DATA_MESSAGE」類型。有時，記 CloudWatch 錄檔可能會發出具有「CONTROL_MESSAGE」類型的 Kinesis Data Streams 記錄，主要用於檢查目的地是否可連線。

owner

原始記錄檔資料的 AWS 帳戶 ID。

logGroup

原始日誌資料的日誌群組名稱。

logStream

原始日誌資料的日誌串流名稱。

subscriptionFilters

與原始日誌資料相符的訂閱篩選條件名稱清單。

logEvents

實際的日誌資料，以一系列的日誌事件記錄呈現。「id」屬性是每個記錄事件的唯一識別符。

政策層級

強制執行策略的層級。「帳戶層級 _ 政策」是用於帳戶層級的`policyLevel`訂閱過濾策略。

範例 3：使用 Amazon 資料 Firehose 的訂閱篩選器

在此範例中，您將建立 CloudWatch Logs 帳戶層級訂閱篩選政策，該政策會將符合定義篩選器的傳入日誌事件傳送到 Amazon Data Firehose 交付串流。從 CloudWatch 日誌傳送到 Amazon 資料 Firehose 的資料已使用 gzip 等級 6 壓縮進行壓縮，因此您不需要在 Firehose 交付串流中使用壓

縮。然後，您可以使用 Firehose 中的解壓縮功能來自動解壓縮記錄檔。如需詳細資訊，請參閱使 CloudWatch 用記錄寫入 Kinesis Data Firehose。

> **Warning**
>
> 建立 Firehose 串流之前，請先計算將產生的記錄資料量。請務必建立可處理此磁碟區的 Firehose 串流。如果串流無法處理磁碟區、日誌串流將受到限制。如需有關 Firehose 串流音量限制的詳細資訊，請參閱 Amazon 資料 Firehose 資料限制。

若要建立 Firehose 的訂閱篩選器

1. 建立 Amazon Simple Storage Service (Amazon S3) 儲存貯體。我們建議您使用專門為 CloudWatch Logs 建立的值區。不過，如果您想要使用現有的儲存貯體，請跳到步驟 2。

   執行以下命令，將預留位置 Region 換成您想要使用的區域：

   ```
   aws s3api create-bucket --bucket my-bucket --create-bucket-configuration
    LocationConstraint=region
   ```

   下列為範例輸出：

   ```
   {
       "Location": "/my-bucket"
   }
   ```

2. 建立 IAM 角色，以授與 Amazon 資料 Firehose 將資料放入您的 Amazon S3 儲存貯體的權限。

   如需詳細資訊，請參閱 Amazon 資料 Firehose 開發人員指南中的使用 Amazon 資料 Firehose 控制存取。

   首先，請按如下所示，使用文字編輯器來建立檔案 ~/TrustPolicyForFirehose.json 中的信任政策：

   ```
   {
     "Statement": {
       "Effect": "Allow",
       "Principal": { "Service": "firehose.amazonaws.com" },
       "Action": "sts:AssumeRole"
       }
   ```

```
}
```

3. 使用 create-role 命令來建立 IAM 角色，並指定信任政策檔案。記下返回的 Role.Arn 值，因為您將在後面的步驟中需要它：

```
aws iam create-role \
 --role-name FirehosetoS3Role \
 --assume-role-policy-document file://~/TrustPolicyForFirehose.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "firehose.amazonaws.com"
                }
            }
        },
        "RoleId": "EXAMPLE50GAB4HC5F431",
        "CreateDate": "2023-05-29T13:46:29.431Z",
        "RoleName": "FirehosetoS3Role",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/FirehosetoS3Role"
    }
}
```

4. 建立權限原則，以定義 Firehose 可以對您的帳戶執行的動作。首先，使用文字編輯器來建立檔案 ~/PermissionsForFirehose.json 中的許可政策：

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "s3:AbortMultipartUpload",
          "s3:GetBucketLocation",
          "s3:GetObject",
          "s3:ListBucket",
          "s3:ListBucketMultipartUploads",
          "s3:PutObject" ],
      "Resource": [
```

```
            "arn:aws:s3:::my-bucket",
            "arn:aws:s3:::my-bucket/*" ]
    }
  ]
}
```

5. 使用下列 put-role-policy命令將權限原則與角色產生關聯：

```
aws iam put-role-policy --role-name FirehosetoS3Role --policy-name Permissions-
Policy-For-Firehose --policy-document file://~/PermissionsForFirehose.json
```

6. 如下所示建立目的地 Firehose 傳遞串流，並以您建立的角色和值區 ARN 取代 Ro leARN 和 BuckeTARN 的預留位置值：

```
aws firehose create-delivery-stream \
   --delivery-stream-name 'my-delivery-stream' \
   --s3-destination-configuration \
  '{"RoleARN": "arn:aws:iam::123456789012:role/FirehosetoS3Role", "BucketARN":
 "arn:aws:s3:::my-bucket"}'
```

對於交付的 Amazon S3 物件，自動使用以 YYYY/MM/DD/HH UTC 時間格式為單位的前綴。您可以指定在時間格式前綴前要新增的額外前綴。如果字首結尾是斜線 (/)，則會在 Amazon S3 儲存貯體中顯示為資料夾。

7. 等待幾分鐘，讓串流變為作用中狀態。您可以使用「Firehose」describe-delivery-stream指令來檢查。DeliveryStreamDescription DeliveryStreamStatus財產。此外，請注意 DeliveryStreamDescription. DeliveryStreamARN 值，因為您將在後面的步驟中需要它：

```
aws firehose describe-delivery-stream --delivery-stream-name "my-delivery-stream"
{
    "DeliveryStreamDescription": {
        "HasMoreDestinations": false,
        "VersionId": "1",
        "CreateTimestamp": 1446075815.822,
        "DeliveryStreamARN": "arn:aws:firehose:us-
east-1:123456789012:deliverystream/my-delivery-stream",
        "DeliveryStreamStatus": "ACTIVE",
        "DeliveryStreamName": "my-delivery-stream",
        "Destinations": [
            {
                "DestinationId": "destinationId-000000000001",
```

```
                "S3DestinationDescription": {
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "RoleARN": "delivery-stream-role",
                    "BucketARN": "arn:aws:s3:::my-bucket",
                    "BufferingHints": {
                        "IntervalInSeconds": 300,
                        "SizeInMBs": 5
                    }
                }
            }
        ]
    }
}
```

8. 建立 IAM 角色，以授與 CloudWatch 記錄檔權限，以將資料放入 Firehose 交付串流。首先，使用文字編輯器來建立檔案 ~/TrustPolicyForCWL.json 中的信任政策：

   此政策包含 aws:SourceArn 全域條件內容金鑰，以協助預防混淆代理人安全問題。如需詳細資訊，請參閱 預防混淆代理人。

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.amazonaws.com" },
    "Action": "sts:AssumeRole",
    "Condition": {
         "StringLike": {
             "aws:SourceArn": "arn:aws:logs:region:123456789012:*"
         }
     }
  }
}
```

9. 使用 create-role 命令來建立 IAM 角色，並指定信任政策檔案。記下返回的 Role.Arn 值，因為您將在後面的步驟中需要它：

```
aws iam create-role \
--role-name CWLtoKinesisFirehoseRole \
--assume-role-policy-document file://~/TrustPolicyForCWL.json
```

```
{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.amazonaws.com"
                },
                "Condition": {
                     "StringLike": {
                         "aws:SourceArn": "arn:aws:logs:region:123456789012:*"
                     }
                 }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisFirehoseRole",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/CWLtoKinesisFirehoseRole"
    }
}
```

10. 建立權限原則，以定義 CloudWatch Logs 可以對您的帳戶執行的動作。首先，使用文字編輯器來建立許可政策檔案 (例如，~/PermissionsForCWL.json)：

```
{
    "Statement":[
      {
        "Effect":"Allow",
        "Action":["firehose:PutRecord"],
        "Resource":[
            "arn:aws:firehose:region:account-id:deliverystream/delivery-stream-
name"]
      }
    ]
}
```

11. 使用以下 put-role-policy命令將權限原則與角色產生關聯：

```
aws iam put-role-policy --role-name CWLtoKinesisFirehoseRole --policy-
name Permissions-Policy-For-CWL --policy-document file://~/PermissionsForCWL.json
```

12. Amazon Data Firehose 交付串流處於使用中狀態，且您已建立 IAM 角色之後，您可以建立 CloudWatch 日誌帳戶層級訂閱篩選政策。政策會立即啟動從所選日誌群組到 Amazon Data Firehose 交付串流的即時日誌資料流程：

```
aws logs put-account-policy \
    --policy-name "ExamplePolicyFirehose" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document '{"RoleArn":"arn:aws:iam::123456789012:role/
CWLtoKinesisFirehoseRole", "DestinationArn":"arn:aws:firehose:us-
east-1:123456789012:deliverystream/delivery-stream-name", "FilterPattern": "Test",
 "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

13. 設定訂閱篩選器後， CloudWatch 日誌會將符合篩選器模式的傳入日誌事件轉寄到 Amazon Data Firehose 交付串流。

    此`selection-criteria`欄位是選擇性欄位，但對於排除可能導致訂閱篩選器無限記錄遞迴的記錄群組而言非常重要。如需有關此問題以及判斷要排除哪些記錄群組的詳細資訊，請參閱防止記錄遞迴。目前，NOT IN 是唯一支援的運算子`selection-criteria`。

    您的資料將會根據 Amazon 資料 Firehose 交付串流上設定的時間緩衝區間隔開始顯示在 Amazon S3 中。一旦經過足夠的時間，您就可以檢查 Amazon S3 儲存貯體來驗證資料。

```
aws s3api list-objects --bucket 'my-bucket' --prefix 'firehose/'
{
    "Contents": [
        {
            "LastModified": "2023-10-29T00:01:25.000Z",
            "ETag": "\"a14589f8897f4089d3264d9e2d1f1610\"",
            "StorageClass": "STANDARD",
            "Key": "firehose/2015/10/29/00/my-delivery-stream-2015-10-29-00-01-21-
a188030a-62d2-49e6-b7c2-b11f1a7ba250",
            "Owner": {
                "DisplayName": "cloudwatch-logs",
                "ID": "1ec9cf700ef6be062b19584e0b7d84ecc19237f87b5"
            },
            "Size": 593
        },
        {
            "LastModified": "2015-10-29T00:35:41.000Z",
```

```
            "ETag": "\"a7035b65872bb2161388ffb63dd1aec5\"",
            "StorageClass": "STANDARD",
            "Key": "firehose/2023/10/29/00/my-delivery-stream-2023-10-29-00-35-40-
EXAMPLE-7e66-49bc-9fd4-fc9819cc8ed3",
            "Owner": {
                "DisplayName": "cloudwatch-logs",
                "ID": "EXAMPLE6be062b19584e0b7d84ecc19237f87b6"
            },
            "Size": 5752
        }
    ]
}
```

```
aws s3api get-object --bucket 'my-bucket' --key 'firehose/2023/10/29/00/my-
delivery-stream-2023-10-29-00-01-21-a188030a-62d2-49e6-b7c2-b11f1a7ba250'
 testfile.gz

{
    "AcceptRanges": "bytes",
    "ContentType": "application/octet-stream",
    "LastModified": "Thu, 29 Oct 2023 00:07:06 GMT",
    "ContentLength": 593,
    "Metadata": {}
}
```

在 Amazon S3 物件中的資料會以 gzip 格式壓縮。您可以使用以下 Unix 命令來透過命令列檢查原始資料：

```
zcat testfile.gz
```

跨帳戶跨區域訂閱

您可以與不同 AWS 帳戶的擁有者共同作業，並在 AWS 資源上接收他們的日誌事件，例如 Amazon Kinesis 或 Amazon Data Firehose 串流 (這稱為跨帳戶資料共用)。例如，您可以從集中式 Kinesis 資料串流或 Firehose 串流讀取此記錄事件資料，以執行自訂處理和分析。自訂處理在您進行跨多個帳戶的協作和分析資料時特別有用。

例如，公司的資訊安全群組可能要分析即時入侵偵測或異常行為的資料，讓它可以對公司所有部門的帳戶進行稽核，方法是收集他們的聯合身分生產日誌以集中處理。這些帳戶之間的事件資料即時串流可以

組合，然後傳送給資訊安全群組，然後再使用 Kinesis Data Streams 將資料連接到其現有的安全分析系統。

> **Note**
>
> 記錄群組和目的地必須位於相同的 AWS 區域。但是，目標指向的 AWS 資源可以位於不同的區域中。在以下各節的範例中，所有區域特定的資源都是在美國東部 (維吉尼亞北部) 建立的。

主題

使用 Kinesis 資料串流進行跨帳戶跨區域記錄資料共用

建立跨帳戶訂閱時，您可以指定單一帳戶或一個組織作為寄件者。如果您指定組織，則此程序會讓組織中的所有帳戶都能將日誌傳送至接收者帳戶。

若要跨帳戶共用日誌資料，您需要建立日誌資料寄件者和接收者：

- 記錄資料寄件者 — 從收件者取得目的地資訊，並讓 CloudWatch 記錄檔知道已準備好將記錄事件傳送至指定的目的地。在本節其餘部分的程序中，記錄資料傳送者會顯示虛構 AWS 帳號 111111111。

  如果您要讓一個組織中的多個帳戶將日誌傳送至一個收件人帳戶，則可以建立一個政策，授予組織中所有帳戶將日誌傳送至收件人帳戶的許可。您仍然必須為每個寄件者帳戶設定個別的訂閱篩選條件。
- 記錄資料收件者 — 設定封裝 Kinesis Data Streams 串流的目的地，並讓 CloudWatch 記錄知道收件者想要接收記錄資料。然後，收件人接著會與寄件者共用與其目的地有關的資訊。在本節其餘部分的程序中，記錄資料收件者會顯示虛構的 AWS 帳戶號碼 999999999999。

若要開始接收跨帳戶使用者的記錄事件，記錄資料收件者會先建立 CloudWatch 記錄目的地。每個目的地包含以下關鍵元素：

目的地名稱

您要建立的目的地名稱。

目標 ARN

您要用作訂閱摘要目的地之 AWS 資源的 Amazon 資源名稱 (ARN)。

角色 ARN

授與 CloudWatch Log 必要權限的 AWS Identity and Access Management (IAM) 角色，可將資料放入所選串流。

存取政策

IAM 政策文件 (JSON 格式，使用 IAM 政策文法撰寫)，決定允許一組使用者寫入您的目的地。

> **Note**
>
> 記錄群組和目的地必須位於相同的 AWS 區域。不過，目的地指向的 AWS 資源可以位於不同的區域。在以下各節的範例中，區域特定的所有資源都在美國東部 (維吉尼亞北部) 建立。

主題

設定新的跨帳戶訂閱

按照這些章節中的步驟設定新的跨帳戶日誌訂閱。

主題

步驟 1：建立目的地

> **Important**
>
> 此程序中的所有步驟必須在日誌資料收件人帳戶中完成。

在此範例中，記錄資料收件者帳戶的帳戶識別碼為 999999999999，而記錄資料寄件者 AWS 帳戶識別碼是 AWS 111111111111。

此範例使用名為的 Kinesis Data Streams 建立目的地 RecipientStream，以及可讓記 CloudWatch 錄寫入資料的角色。

建立目的地時， CloudWatch Logs 會代表收件者帳戶將測試訊息傳送至目的地。稍後訂閱篩選器處於作用中狀態時， CloudWatch Logs 會代表來源帳戶將記錄事件傳送至目的地。

若要建立目的地

1. 在收件人帳戶中，在 Kinesis Data Streams 中建立目的地串流。在命令提示字元中輸入：

   ```
   aws kinesis create-stream --stream-name "RecipientStream" --shard-count 1
   ```

2. 等到 串流變成作用中。您可以使用 aws 室運動描述流命令來檢查. StreamDescription StreamStatus財產。此外，請記下 StreamDescription.Streamarn 值，因為您稍後會將其傳送至 CloudWatch 記錄檔：

   ```
   aws kinesis describe-stream --stream-name "RecipientStream"
   {
     "StreamDescription": {
       "StreamStatus": "ACTIVE",
       "StreamName": "RecipientStream",
       "StreamARN": "arn:aws:kinesis:us-east-1:999999999999:stream/RecipientStream",
       "Shards": [
         {
           "ShardId": "shardId-000000000000",
           "HashKeyRange": {
             "EndingHashKey": "340282366920938463463374607431768EXAMPLE",
             "StartingHashKey": "0"
           },
           "SequenceNumberRange": {
             "StartingSequenceNumber":
    "49551135218688818456679503831981458784591352702 18EXAMPLE"
   ```

```
        }
      }
    ]
  }
}
```

這可能需要花費幾分鐘，讓串流以作用中狀態出現。

3. 建立 IAM 角色，以授與 CloudWatch 記錄將資料放入串流的權限。首先，您需要在~/TrustPolicyForCW L.json 文件中創建一個信任策略。使用文字編輯器來建立此政策檔案，請勿使用 IAM 主控台。

   此政策包含 aws:SourceArn 全域條件內容金鑰，可指定 sourceAccountId 以協助預防混淆代理人安全問題。如果您在第一次呼叫中還不知道來源帳戶 ID，我們建議您將目的地 ARN 放在來源 ARN 欄位中。在後續呼叫中，應將來源 ARN 設定為從第一次呼叫中收集的實際來源 ARN。如需詳細資訊，請參閱 預防混淆代理人。

```
{
    "Statement": {
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.amazonaws.com"
        },
        "Condition": {
            "StringLike": {
                "aws:SourceArn": [
                    "arn:aws:logs:region:sourceAccountId:*",
                    "arn:aws:logs:region:recipientAccountId:*"
                ]
            }
        },
        "Action": "sts:AssumeRole"
    }
}
```

4. 使用 aws iam create-role 命令來建立 IAM 角色，並指定信任政策檔案。請記下傳回的 Role.Arn 值，因為它也會在稍後傳送至 CloudWatch 記錄檔：

```
aws iam create-role \
--role-name CWLtoKinesisRole \
--assume-role-policy-document file://~/TrustPolicyForCWL.json
```

```
{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Condition": {
                    "StringLike": {
                        "aws:SourceArn": [
                            "arn:aws:logs:region:sourceAccountId:*",
                            "arn:aws:logs:region:recipientAccountId:*"
                        ]
                    }
                },
                "Principal": {
                    "Service": "logs.amazonaws.com"
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisRole",
        "Path": "/",
        "Arn": "arn:aws:iam::999999999999:role/CWLtoKinesisRole"
    }
}
```

5. 建立權限原則，以定義 CloudWatch 記錄檔可對您的帳戶執行哪些動作。首先，使用文本編輯器在文件〜/PermissionsForCW L.json 中創建權限策略：

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:region:999999999999:stream/RecipientStream"
    }
  ]
}
```

6. 使用 aws iam put-role-policy 命令將許可政策與角色相關聯：

```
aws iam put-role-policy \
```

```
    --role-name CWLtoKinesisRole \
    --policy-name Permissions-Policy-For-CWL \
    --policy-document file://~/PermissionsForCWL.json
```

7. 串流處於使用中狀態且您已建立 IAM 角色之後，您可以建立 CloudWatch 記錄目的地。

   a. 此步驟不會將存取政策與您的目的地相關聯，且只是完成目的地建立兩步驟中的第一步。記下DestinationArn有效負載中返回的內容：

```
aws logs put-destination \
    --destination-name "testDestination" \
    --target-arn "arn:aws:kinesis:region:999999999999:stream/RecipientStream" \
    --role-arn "arn:aws:iam::999999999999:role/CWLtoKinesisRole"

{
  "DestinationName" : "testDestination",
  "RoleArn" : "arn:aws:iam::999999999999:role/CWLtoKinesisRole",
  "DestinationArn" : "arn:aws:logs:us-
east-1:999999999999:destination:testDestination",
  "TargetArn" : "arn:aws:kinesis:us-east-1:999999999999:stream/RecipientStream"
}
```

   b. 步驟 7a 完成後，即可在日誌資料收件人帳戶中，將存取政策與目的地建立關聯。此原則必須指定 log: PutSubscriptionFilter 動作，並授與寄件者帳戶存取目的地的權限。

      此原則會授與傳送記錄檔之 AWS 帳戶的權限。您可以在政策中僅指定這一個帳戶，或者如果寄件者帳戶是組織的成員，則政策可以指定組織的組織 ID。如此一來，您可以僅建立一個政策，就能允許一個組織中的多個帳戶將日誌傳送至此目的地帳戶。

      使用文字編輯器建立名為 ~/AccessPolicy.json 的檔案，並隨附下列其中一個政策陳述。

      此第一個範例政策允許組織中具有 ID 為 o-1234567890 的所有帳戶將日誌傳送至收件人帳戶。

```
{
    "Version" : "2012-10-17",
    "Statement" : [
        {
            "Sid" : "",
            "Effect" : "Allow",
            "Principal" : "*",
            "Action" : "logs:PutSubscriptionFilter",
```

```
            "Resource" :
 "arn:aws:logs:region:999999999999:destination:testDestination",
            "Condition": {
               "StringEquals" : {
                   "aws:PrincipalOrgID" : ["o-1234567890"]
                }
            }
        }
    ]
}
```

此下一個範例只允許日誌資料寄件者帳戶 (111111111111) 將日誌傳送至日誌資料收件人帳戶。

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111111111111"
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" :
 "arn:aws:logs:region:999999999999:destination:testDestination"
    }
  ]
}
```

c. 將您在上一步驟建立的政策連接到目的地。

```
aws logs put-destination-policy \
    --destination-name "testDestination" \
    --access-policy file://~/AccessPolicy.json
```

*####### AWS ####### 1111111111 ###### ARN arn: aw: ##:# #:99999999999999: ###:**PutSubscriptionFilter**########*任何其他用戶嘗試撥打此目 PutSubscriptionFilter 的地的嘗試將被拒絕。

若要驗證使用者的權限符合存取政策，請參閱《IAM 使用者指南》中的使用政策驗證程式。

完成後，如果您正在使用 AWS Organizations 跨帳戶權限，請按照中步驟 2：(僅限於使用組織時) 建立 IAM 角色的步驟操作。如果您要將許可直接授予給其他帳戶，而不是使用 Organizations，則可以略過該步驟並繼續進行 步驟 4：建立訂閱篩選條件。

步驟 2：(僅限於使用組織時) 建立 IAM 角色

在上一節中，如果您藉由使用授予許可給帳戶 111111111111 所屬組織的存取政策來建立目的地，而不是將許可直接授予給帳戶 111111111111，則按照本節中的步驟進行。若否，則可跳至步驟 步驟 4：建立訂閱篩選條件。

本節中的步驟會建立 IAM 角色，該角色 CloudWatch 可假設並驗證寄件者帳戶是否具有針對收件者目的地建立訂閱篩選器的權限。

在寄件者帳戶中執行此區段中的步驟。角色必須存在於寄件者帳戶中，而且您要在訂閱篩選條件中指定此角色的 ARN。在此範例中，使用者帳戶為 111111111111。

使用 AWS Organizations建立跨帳戶日誌訂閱所需的 IAM 角色

1. 在檔案 /TrustPolicyForCWLSubscriptionFilter.json 中建立下列信任政策。使用文字編輯器來建立此政策檔案，請勿使用 IAM 主控台。

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.amazonaws.com" },
    "Action": "sts:AssumeRole"
  }
}
```

2. 建立使用此政策的 IAM 角色。記下命令傳回的 Arn 值，之後在此程序中會用到。在此範例中，我們使用 CWLtoSubscriptionFilterRole 作為要建立的角色的名稱。

```
aws iam create-role \
     --role-name CWLtoSubscriptionFilterRole \
     --assume-role-policy-document file://~/
TrustPolicyForCWLSubscriptionFilter.json
```

3. 建立權限原則，以定義記 CloudWatch 錄檔可在您的帳戶上執行的動作。

   a. 首先，使用文字編輯器在名為 ~/PermissionsForCWLSubscriptionFilter.json 的檔案中建立下列許可政策。

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "logs:PutLogEvents",
            "Resource": "arn:aws:logs:region:111111111111:log-
group:LogGroupOnWhichSubscriptionFilterIsCreated:*"
        }
    ]
}
```

b. 輸入下列命令，將您剛建立的許可政策與您在步驟 2 中建立的角色相關聯。

```
aws iam put-role-policy
    --role-name CWLtoSubscriptionFilterRole
    --policy-name Permissions-Policy-For-CWL-Subscription-filter
    --policy-document file://~/PermissionsForCWLSubscriptionFilter.json
```

完成後，可以繼續進行 步驟 4：建立訂閱篩選條件。

步驟 3：新增/驗證跨帳戶目的地的 IAM 許可

根據 AWS 跨帳戶原則評估邏輯，若要存取任何跨帳號資源 (例如作為訂閱篩選器目的地使用的 Kinesis 或 Firehose 串流)，您必須在傳送帳戶中具有以身分識別為基礎的政策，以提供跨帳戶目標資源的明確存取權。如需有關政策評估邏輯的詳細資訊，請參閱《跨帳戶政策評估邏輯》。

您可以將身分型政策連接至用來建立訂閱篩選條件的 IAM 角色或 IAM 使用者。傳送帳戶中必須有此政策存在。如果您使用管理員角色建立訂閱篩選條件，則可以略過此步驟並繼續進行 步驟 4：建立訂閱篩選條件。

新增或驗證跨帳戶所需的 IAM 許可

1. 輸入以下命令以檢查正在使用哪個 IAM 角色或 IAM 使用者來執行 AWS 日誌命令。

```
aws sts get-caller-identity
```

此命令會傳回類似以下的輸出：

```
{
"UserId": "User ID",
```

```
"Account": "sending account id",
"Arn": "arn:aws:sending account id:role/user:RoleName/UserName"
}
```

記下*RoleName*或所代表的值*UserName*。

2. 登入傳送帳戶，然後使用您 AWS Management Console 在步驟 1 輸入的命令輸出中傳回的 IAM 角色或 IAM 使用者搜尋附加的政策。
3. 確認連接至此角色或使用者的政策提供明確的許可，可對跨帳戶目的地資源呼叫 `logs:PutSubscriptionFilter`。以下範例政策顯示建議的許可。

   下列原則提供僅在單一帳號 (AWS 帳號) 中針對任何目標資源建立訂閱篩選器的權限123456789012：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow subscription filters on any resource in one specific
 account",
            "Effect": "Allow",
            "Action": "logs:PutSubscriptionFilter",
            "Resource": [
                "arn:aws:logs:*:*:log-group:*",
                "arn:aws:logs:*:123456789012:destination:*"
            ]
        }
    ]
}
```

   下列原則提供權限，可讓您僅sampleDestination在單一帳號 (AWS 帳號) 中名為的特定目標資源上建立訂閱篩選器123456789012：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow subscription filters on one specific resource in one
 specific account",
            "Effect": "Allow",
            "Action": "logs:PutSubscriptionFilter",
            "Resource": [
```

```
                "arn:aws:logs:*:*:log-group:*",
                "arn:aws:logs:*:123456789012:destination:sampleDestination"
            ]
        }
    ]
}
```

步驟 4：建立訂閱篩選條件

在建立目的地後，日誌資料收件人帳戶可以與其他 AWS 帳戶共用目的地 ARN (arn:aws:logs:us-east-1:999999999999:destination:testDestination)，讓他們能將日誌事件傳送到相同目的地。然後，這些其他傳送帳戶使用者接著會在個別的日誌群組針對此目的地建立訂閱篩選條件。訂閱篩選條件會立即開始將即時日誌資料從所選的日誌群組傳送到指定的目標。

> **Note**
>
> 如果您要將訂閱篩選條件的許可授予給整個組織，您需要使用您在 步驟 2：(僅限於使用組織時) 建立 IAM 角色 中建立的 IAM 角色 ARN。

在下列範例中，會在傳送帳戶中建立訂閱篩選器。篩選器與包含 AWS CloudTrail 事件的記錄群組相關聯，因此，「根」 AWS 認證所做的每個記錄活動都會傳遞至您先前建立的目的地。該目的地封裝了一個名為「RecipientStream」的流。

下列各節的其餘步驟假設您已遵循使用指南中〈將 CloudTrail事件傳送至 CloudWatch 記錄檔〉中的AWS CloudTrail 指示，並建立了包含您的 CloudTrail 事件的記錄群組。這些步驟假定此日誌群組的名稱為 `CloudTrail/logs`。

當您輸入以下命令時，確認您以 IAM 使用者身分登入，或是使用您在 步驟 3：新增/驗證跨帳戶目的地的 IAM 許可 中為其新增政策的 IAM 角色登入。

```
aws logs put-subscription-filter \
    --log-group-name "CloudTrail/logs" \
    --filter-name "RecipientStream" \
    --filter-pattern "{$.userIdentity.type = Root}" \
    --destination-arn "arn:aws:logs:region:999999999999:destination:testDestination"
```

記錄群組和目的地必須位於相同的 AWS 區域。但是，目標可以指向位於不同區域的 AWS 資源，例如 Kinesis Data Streams 流。

驗證日誌事件的流動

建立訂閱篩選器之後，CloudWatch 記錄會將符合篩選器模式的所有傳入記錄事件轉寄至封裝在目的地資料流中的資料流 (稱為 "」RecipientStream)。目的地擁有者可以使用 aws kinesis get-shard-iterator 命令擷取 Kinesis 資料串流碎片，並使用 aws kinesis 取得記錄命令來擷取一些 Kinesis Data Streams 記錄來確認是否發生這種情況：

```
aws kinesis get-shard-iterator \
      --stream-name RecipientStream \
      --shard-id shardId-000000000000 \
      --shard-iterator-type TRIM_HORIZON

{
    "ShardIterator":
    "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
}

aws kinesis get-records \
      --limit 10 \
      --shard-iterator
      "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
```

> **Note**
>
> 您可能需要重新執行幾次 get-records 命令，Kinesis Data Streams 才會開始傳回資料。

您應該會看到一系列 Kinesis Data Streams 記錄的回應。Kinesis Data Streams 記錄中的資料屬性，是以 gzip 格式壓縮並採用 Base64 編碼。您可以使用以下 Unix 命令來透過命令列檢查原始資料：

```
echo -n "<Content of Data>" | base64 -d | zcat
```

Base64 解碼和解壓縮資料是以 JSON 形式並以下列結構進行格式化：

```
{
```

```
    "owner": "111111111111",
    "logGroup": "CloudTrail/logs",
    "logStream": "111111111111_CloudTrail/logs_us-east-1",
    "subscriptionFilters": [
        "RecipientStream"
    ],
    "messageType": "DATA_MESSAGE",
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        }
    ]
}
```

在資料結構的關鍵元素如下：

owner

原始記錄檔資料的 AWS 帳戶 ID。

logGroup

原始日誌資料的日誌群組名稱。

logStream

原始日誌資料的日誌串流名稱。

subscriptionFilters

與原始日誌資料相符的訂閱篩選條件名稱清單。

messageType

資料訊息使用 "DATA_MESSAGE" 類型。有時，記 CloudWatch 錄檔可能會發出具有「CONTROL_MESSAGE」類型的 Kinesis Data Streams 記錄，主要用於檢查目的地是否可連線。

logEvents

實際的日誌資料，以一系列的日誌事件記錄呈現。ID 屬性是每個記錄事件的唯一識別符。

在執行期修改目的地成員資格

您可能遇到以下情況：您需要從擁有的目的地中新增或移除某些使用者的成員資格。您可以在目的地使用 `put-destination-policy` 命令並指定新的存取政策。在下列範例中，之前新增帳戶 111111111111 會停止傳送任何更多資料，且帳戶 222222222222 會啟用。

1. 擷取目前與目的地 testD estination 相關聯的原則，並記下 AccessPolicy：

```
aws logs describe-destinations \
    --destination-name-prefix "testDestination"

{
 "Destinations": [
   {
     "DestinationName": "testDestination",
     "RoleArn": "arn:aws:iam::999999999999:role/CWLtoKinesisRole",
     "DestinationArn":
 "arn:aws:logs:region:999999999999:destination:testDestination",
     "TargetArn": "arn:aws:kinesis:region:999999999999:stream/RecipientStream",
     "AccessPolicy": "{\"Version\": \"2012-10-17\", \"Statement\":
 [{\"Sid\": \"\", \"Effect\": \"Allow\", \"Principal\": {\"AWS\":
 \"111111111111\"}, \"Action\": \"logs:PutSubscriptionFilter\", \"Resource\":
 \"arn:aws:logs:region:999999999999:destination:testDestination\"}] }"
   }
 ]
}
```

2. 更新政策，以反映帳戶 111111111111 已停用，且帳戶 222222222222 已啟用。將此策略放在~/NewAccessPolicy.json 文件中：

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
```

```
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "222222222222"
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" : "arn:aws:logs:region:999999999999:destination:testDestination"
    }
  ]
}
```

3. 呼叫PutDestinationPolicy將 NewAccessPolicy.json 檔案中定義的原則與目的地產生關聯：

```
aws logs put-destination-policy \
--destination-name "testDestination" \
--access-policy file://~/NewAccessPolicy.json
```

這最終會從帳戶 ID 111111111111 停用日誌事件。在帳戶 222222222222 的擁有者建立訂閱篩選條件後，來自帳戶 ID 222222222222 的日誌事件會立刻開始流向目的地。

更新現有的跨帳戶訂閱

如果您目前有跨帳戶日誌訂閱，其中目的地帳戶僅授予特定寄件者帳戶許可，而您想要更新此訂閱，讓目的地帳戶授予組織中所有帳戶的存取權，請按照本節中的步驟進行。

主題

- 步驟 1：更新訂閱篩選條件
- 步驟 2：更新現有的目的地存取政策

步驟 1：更新訂閱篩選條件

> **Note**
>
> 只有跨帳戶訂閱由 啟用從 AWS 服務記錄 所列服務建立的日誌才需要此步驟。如果您不使用這些日誌群組之一建立的日誌，則可以跳至 步驟 2：更新現有的目的地存取政策。

在某些情況下，您必須更新所有傳送日誌至目的地帳戶的寄件者帳戶中的訂閱篩選條件。此更新新增 IAM 角色，該角色 CloudWatch 可假設並驗證寄件者帳戶是否具有將記錄傳送至收件者帳戶的權限。

針對您想要更新的每個寄件者帳戶，按照本節中的步驟進行，以將組織 ID 用於跨帳戶訂閱許可。

在本節的範例中，111111111111 和 222222222222 兩個帳戶已經建立訂閱篩選條件，以將日誌傳送至帳戶 999999999999。現有的訂閱篩選條件值如下：

```
## Existing Subscription Filter parameter values
    \ --log-group-name "my-log-group-name"
    \ --filter-name "RecipientStream"
    \ --filter-pattern "{$.userIdentity.type = Root}"
    \ --destination-arn "arn:aws:logs:region:999999999999:destination:testDestination"
```

如果需要尋找目前的訂閱篩選條件參數值，請輸入下列命令。

```
aws logs describe-subscription-filters
    \ --log-group-name "my-log-group-name"
```

更新訂閱篩選條件以開始將組織 ID 用於跨帳戶日誌許可

1. 在檔案 ~/TrustPolicyForCWL.json 中建立下列信任政策。使用文字編輯器來建立此政策檔案，請勿使用 IAM 主控台。

   ```
   {
     "Statement": {
       "Effect": "Allow",
       "Principal": { "Service": "logs.amazonaws.com" },
       "Action": "sts:AssumeRole"
     }
   }
   ```

2. 建立使用此政策的 IAM 角色。記下命令傳回的 Arn 值的 Arn 值，之後在此程序中會用到。在此範例中，我們使用 CWLtoSubscriptionFilterRole 作為要建立的角色的名稱。

   ```
   aws iam create-role
       \ --role-name CWLtoSubscriptionFilterRole
       \ --assume-role-policy-document file://~/TrustPolicyForCWL.json
   ```

3. 建立權限原則，以定義記 CloudWatch 錄檔可在您的帳戶上執行的動作。

   a. 首先，使用文字編輯器在名為 /PermissionsForCWLSubscriptionFilter.json 的檔案中建立下列許可政策。

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "logs:PutLogEvents",
            "Resource": "arn:aws:logs:region:111111111111:log-
group:LogGroupOnWhichSubscriptionFilterIsCreated:*"
        }
    ]
}
```

b. 輸入下列命令，將您剛建立的許可政策與您在步驟 2 中建立的角色相關聯。

```
aws iam put-role-policy
    --role-name CWLtoSubscriptionFilterRole
    --policy-name Permissions-Policy-For-CWL-Subscription-filter
    --policy-document file://~/PermissionsForCWLSubscriptionFilter.json
```

4. 輸入下列命令以更新訂閱篩選條件。

```
aws logs put-subscription-filter
    \ --log-group-name "my-log-group-name"
    \ --filter-name "RecipientStream"
    \ --filter-pattern "{$.userIdentity.type = Root}"
    \ --destination-arn
 "arn:aws:logs:region:999999999999:destination:testDestination"
    \ --role-arn "arn:aws:iam::111111111111:role/CWLtoSubscriptionFilterRole"
```

步驟 2：更新現有的目的地存取政策

更新所有寄件者帳戶中的訂閱篩選條件之後，您可以更新收件人帳戶中的目的地存取政策。

下列範例中，收件人帳戶為 999999999999，且目的地名稱為 testDestination。

此更新可讓屬於組織且 ID 為 o-1234567890 的所有帳戶傳送日誌至收件人帳戶。只有已建立訂閱篩選條件的帳戶才會真的傳送日誌至收件人帳戶。

更新收件人帳戶中的目的地存取政策，以開始將組織 ID 用於許可

1. 在收件人帳戶中，使用文字編輯器建立 ~/AccessPolicy.json 檔案，其中包含以下內容。

```
{
    "Version" : "2012-10-17",
    "Statement" : [
        {
            "Sid" : "",
            "Effect" : "Allow",
            "Principal" : "*",
            "Action" : "logs:PutSubscriptionFilter",
            "Resource" :
 "arn:aws:logs:region:999999999999:destination:testDestination",
            "Condition": {
               "StringEquals" : {
                   "aws:PrincipalOrgID" : ["o-1234567890"]
                }
            }
        }
    ]
}
```

2. 輸入下列命令，將您剛建立的政策連接到現有的目的地。若要更新目的地，以使用具有組織 ID 的存取政策，而不是列出特定 AWS 帳戶 ID 的存取政策，則應包含 `force` 參數。

> **Warning**
>
> 如果您正在使用中列出的 AWS 服務傳送的記錄檔啟用從 AWS 服務記錄，則在執行此步驟之前，您必須先更新所有寄件者帳戶中的訂閱篩選器，如中所述步驟 1：更新訂閱篩選條件。

```
aws logs put-destination-policy
    \ --destination-name "testDestination"
    \ --access-policy file://~/AccessPolicy.json
    \ --force
```

使用 Firehose 進行跨帳戶跨區域記錄資料分享

若要跨帳戶共用日誌資料，您需要建立日誌資料寄件者和接收者：

- 記錄資料寄件者 — 從收件者取得目的地資訊，並讓 CloudWatch 記錄檔知道已準備好將其記錄事件傳送至指定的目的地。在本節其餘部分的程序中，記錄資料傳送者會顯示虛構 AWS 帳號 1111111111。
- 記錄資料收件者 — 設定封裝 Kinesis Data Streams 串流的目的地，並讓 CloudWatch 記錄知道收件者想要接收記錄資料。然後，收件人接著會與寄件者共用與其目的地有關的資訊。在本節其餘部分的程序中，記錄資料收件者會以虛構的 AWS 帳號 2222222222 顯示。

本節中的範例使用 Firehose 交付串流搭配 Amazon S3 儲存。您也可以使用不同的設定來設定 Firehose 交付串流。如需詳細資訊，請參閱建立 Firehose 傳遞串流。

> **Note**
>
> 記錄群組和目的地必須位於相同的 AWS 區域。不過，目的地指向的 AWS 資源可以位於不同的區域。

> **Note**
>
> 支援相同帳戶和跨區域交付串流的 Firehose 訂閱篩選器。

主題

步驟 1：建立 Firehose 傳送串流

> **Important**
>
> 在完成下列步驟之前，您必須使用存取政策，以便 Firehose 可以存取您的 Amazon S3 儲存貯體。如需詳細資訊，請參閱 Amazon 資料 Firehose 開發人員指南中的控制存取。

必須在日誌資料收件人帳戶中完成本區段 (步驟 1) 中的所有步驟。
在以下範例命令中使用美國東部 (維吉尼亞北部)。請將此區域替換成您部署的正確區域。

若要建立要用作目的地的 Firehose 傳送串流

1. 建立 Amazon S3 儲存貯體：

```
aws s3api create-bucket --bucket firehose-test-bucket1 --create-bucket-
configuration LocationConstraint=us-east-1
```

2. 建立 IAM 角色，以授予 Firehose 將資料放入值區的權限。

   a. 首先，使用文字編輯器在檔案 ~/TrustPolicyForFirehose.json 中建立信任政策。

```
{ "Statement": { "Effect": "Allow", "Principal": { "Service":
 "firehose.amazonaws.com" }, "Action": "sts:AssumeRole", "Condition":
 { "StringEquals": { "sts:ExternalId":"222222222222" } } } }
```

   b. 建立 IAM 角色，並指定您剛建立的信任政策檔案。

```
aws iam create-role \
    --role-name FirehosetoS3Role \
    --assume-role-policy-document file://~/TrustPolicyForFirehose.json
```

   c. 此命令的輸出看起來如下：記下角色名稱和角色 ARN。

```
{
    "Role": {
        "Path": "/",
        "RoleName": "FirehosetoS3Role",
        "RoleId": "AROAR3BXASEKW7K635M53",
        "Arn": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
        "CreateDate": "2021-02-02T07:53:10+00:00",
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Effect": "Allow",
                "Principal": {
                    "Service": "firehose.amazonaws.com"
                },
                "Action": "sts:AssumeRole",
                "Condition": {
```

```
                    "StringEquals": {
                        "sts:ExternalId": "222222222222"
                    }
                }
            }
        }
    }
}
```

3. 建立權限原則，以定義 Firehose 可在您帳戶中執行的動作。

   a. 首先，使用文字編輯器在名為 ~/PermissionsForFirehose.json 的檔案中建立下列許可政策。根據您的使用案例，可能需要為此檔案新增更多許可。

```
{
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "s3:PutObject",
            "s3:PutObjectAcl",
            "s3:ListBucket"
        ],
        "Resource": [
            "arn:aws:s3:::firehose-test-bucket1",
            "arn:aws:s3:::firehose-test-bucket1/*"
        ]
    }]
}
```

   b. 輸入下列命令，將您剛建立的許可政策與 IAM 角色相關聯。

```
aws iam put-role-policy --role-name FirehosetoS3Role --policy-name
 Permissions-Policy-For-Firehose-To-S3 --policy-document file://~/
PermissionsForFirehose.json
```

4. 輸入下列命令以建立 Firehose 傳送串流。以正確*my-bucket-arn*的部署值取代*my-role-arn*和。

```
aws firehose create-delivery-stream \
   --delivery-stream-name 'my-delivery-stream' \
   --s3-destination-configuration \
```

```
  '{"RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role", "BucketARN":
 "arn:aws:s3:::firehose-test-bucket1"}'
```

輸出格式應類似以下內容：

```
{
    "DeliveryStreamARN": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream"
}
```

步驟 2：建立目的地

> **Important**
>
> 此程序中的所有步驟必須在日誌資料收件人帳戶中完成。

建立目的地時，CloudWatch Logs 會代表收件者帳戶將測試訊息傳送至目的地。稍後訂閱篩選器處於作用中狀態時，CloudWatch Logs 會代表來源帳戶將記錄事件傳送至目的地。

若要建立目的地

1. 等到您在其中建立的 Firehose 串流步驟 1：建立 Firehose 傳送串流變為作用中。您可以使用以下命令來檢查 StreamDescription. StreamStatus財產。

   ```
   aws firehose describe-delivery-stream --delivery-stream-name "my-delivery-stream"
   ```

   此外，請注意的DeliveryStreamDescription。 DeliveryStreamARN 值，因為您將需要在後面的步驟中使用它。此命令的範例輸出：

   ```
   {
       "DeliveryStreamDescription": {
           "DeliveryStreamName": "my-delivery-stream",
           "DeliveryStreamARN": "arn:aws:firehose:us-
   east-1:222222222222:deliverystream/my-delivery-stream",
           "DeliveryStreamStatus": "ACTIVE",
           "DeliveryStreamEncryptionConfiguration": {
               "Status": "DISABLED"
           },
   ```

```
        "DeliveryStreamType": "DirectPut",
        "VersionId": "1",
        "CreateTimestamp": "2021-02-01T23:59:15.567000-08:00",
        "Destinations": [
            {
                "DestinationId": "destinationId-000000000001",
                "S3DestinationDescription": {
                    "RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
                    "BucketARN": "arn:aws:s3:::firehose-test-bucket1",
                    "BufferingHints": {
                        "SizeInMBs": 5,
                        "IntervalInSeconds": 300
                    },
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "CloudWatchLoggingOptions": {
                        "Enabled": false
                    }
                },
                "ExtendedS3DestinationDescription": {
                    "RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
                    "BucketARN": "arn:aws:s3:::firehose-test-bucket1",
                    "BufferingHints": {
                        "SizeInMBs": 5,
                        "IntervalInSeconds": 300
                    },
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "CloudWatchLoggingOptions": {
                        "Enabled": false
                    },
                    "S3BackupMode": "Disabled"
                }
            }
        ],
        "HasMoreDestinations": false
    }
}
```

可能需要花費幾分鐘，交付串流才會變成作用中狀態。

2. 當交付串流處於作用中狀態時，請建立 IAM 角色，以授與 CloudWatch 記錄檔將資料放入 Firehose 串流的權限。首先，您需要在~/TrustPolicyForCW L.json 文件中創建一個信任策略。請使用文字編輯器來建立此政策。如需有關 CloudWatch 日誌端點的詳細資訊，請參閱 Amazon CloudWatch 日誌端點和配額。

   此政策包含 aws:SourceArn 全域條件內容金鑰，可指定 sourceAccountId 以協助預防混淆代理人安全問題。如果您在第一次呼叫中還不知道來源帳戶 ID，我們建議您將目的地 ARN 放在來源 ARN 欄位中。在後續呼叫中，應將來源 ARN 設定為從第一次呼叫中收集的實際來源 ARN。如需詳細資訊，請參閱 預防混淆代理人。

```
{
    "Statement": {
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.region.amazonaws.com"
        },
        "Action": "sts:AssumeRole",
        "Condition": {
            "StringLike": {
                "aws:SourceArn": [
                    "arn:aws:logs:region:sourceAccountId:*",
                    "arn:aws:logs:region:recipientAccountId:*"
                ]
            }
        }
     }
}
```

3. 使用 aws iam create-role 命令來建立 IAM 角色，並指定您剛建立的信任政策檔案。

```
aws iam create-role \
      --role-name CWLtoKinesisFirehoseRole \
      --assume-role-policy-document file://~/TrustPolicyForCWL.json
```

   下列為範例輸出。請留意傳回的 Role.Arn 值，因為後續步驟中需要用到。

```
{
    "Role": {
        "Path": "/",
```

```
        "RoleName": "CWLtoKinesisFirehoseRole",
        "RoleId": "AROAR3BXASEKYJYWF243H",
        "Arn": "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole",
        "CreateDate": "2021-02-02T08:10:43+00:00",
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.region.amazonaws.com"
                },
                "Action": "sts:AssumeRole",
                "Condition": {
                    "StringLike": {
                        "aws:SourceArn": [
                            "arn:aws:logs:region:sourceAccountId:*",
                            "arn:aws:logs:region:recipientAccountId:*"
                        ]
                    }
                }
            }
        }
    }
}
```

4. 建立權限原則，以定義 CloudWatch 記錄檔可對您的帳戶執行哪些動作。首先，使用文本編輯器在文件〜/PermissionsForCW L.json 中創建權限策略：

```
{
    "Statement":[
      {
        "Effect":"Allow",
        "Action":["firehose:*"],
        "Resource":["arn:aws:firehose:region:222222222222:*"]
      }
    ]
}
```

5. 輸入以下命令，將許可政策與角色建立關聯：

```
aws iam put-role-policy --role-name CWLtoKinesisFirehoseRole --policy-name
 Permissions-Policy-For-CWL --policy-document file://~/PermissionsForCWL.json
```

6. Firehose 交付串流處於使用中狀態且您已建立 IAM 角色之後，您可以建立 CloudWatch 記錄目標。

   a. 此步驟不會將存取政策與您的目的地建立關聯，且為完成建立目的地之兩個步驟的僅第一個步驟。記下承載中傳回的新目的地的 ARN，因為您會在後續步驟中使用它作為 `destination.arn`。

```
aws logs put-destination \

    --destination-name "testFirehoseDestination" \
    --target-arn "arn:aws:firehose:us-east-1:222222222222:deliverystream/my-
delivery-stream" \
    --role-arn "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole"

{
    "destination": {
        "destinationName": "testFirehoseDestination",
        "targetArn": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream",
        "roleArn": "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole",
        "arn": "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"}
}
```

   b. 上一個步驟完成後，請在日誌資料收件人帳戶中 (222222222222)，將存取政策與目的地建立關聯。

      此政策可讓日誌資料寄件者帳戶 (111111111111) 只能在日誌資料收件人帳戶 (222222222222) 中存取目的地。您可以使用文本編輯器將此策略放在 ~/ AccessPolicy .json 文件中：

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111111111111"
      },
      "Action" : "logs:PutSubscriptionFilter",
```

```
      "Resource" : "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"
    }
  ]
}
```

c. 這會建立可定義誰擁有對目的地的寫入存取權之政策。此原則必須指定 log:PutSubscriptionFilter 動作才能存取目的地。跨帳戶使用者將使用此PutSubscriptionFilter動作將記錄事件傳送至目的地：

```
aws logs put-destination-policy \
    --destination-name "testFirehoseDestination" \
    --access-policy file://~/AccessPolicy.json
```

步驟 3：新增/驗證跨帳戶目的地的 IAM 許可

根據 AWS 跨帳戶原則評估邏輯，若要存取任何跨帳號資源 (例如作為訂閱篩選器目的地使用的 Kinesis 或 Firehose 串流)，您必須在傳送帳戶中具有以身分識別為基礎的政策，以提供跨帳戶目標資源的明確存取權。如需有關政策評估邏輯的詳細資訊，請參閱《跨帳戶政策評估邏輯》。

您可以將身分型政策連接至用來建立訂閱篩選條件的 IAM 角色或 IAM 使用者。傳送帳戶中必須有此政策存在。如果您使用管理員角色建立訂閱篩選條件，則可以略過此步驟並繼續進行 步驟 4：建立訂閱篩選條件。

新增或驗證跨帳戶所需的 IAM 許可

1. 輸入以下命令以檢查正在使用哪個 IAM 角色或 IAM 使用者來執行 AWS 日誌命令。

```
aws sts get-caller-identity
```

此命令會傳回類似以下的輸出：

```
{
"UserId": "User ID",
"Account": "sending account id",
"Arn": "arn:aws:sending account id:role/user:RoleName/UserName"
}
```

記下*RoleName*或所代表的值*UserName*。

2. 登入傳送帳戶，然後使用您 AWS Management Console 在步驟 1 輸入的命令輸出中傳回的 IAM 角色或 IAM 使用者搜尋附加的政策。
3. 確認連接至此角色或使用者的政策提供明確的許可，可對跨帳戶目的地資源呼叫 `logs:PutSubscriptionFilter`。以下範例政策顯示建議的許可。

   下列原則提供僅在單一帳號 (AWS 帳號) 中針對任何目標資源建立訂閱篩選器的權限`123456789012`：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow subscription filters on any resource in one specific
 account",
            "Effect": "Allow",
            "Action": "logs:PutSubscriptionFilter",
            "Resource": [
                "arn:aws:logs:*:*:log-group:*",
                "arn:aws:logs:*:123456789012:destination:*"
            ]
        }
    ]
}
```

   下列原則提供權限，可讓您僅`sampleDestination`在單一帳號 (AWS 帳號) 中名為的特定目標資源上建立訂閱篩選器`123456789012`：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow subscription filters on one specific resource in one
 specific account",
            "Effect": "Allow",
            "Action": "logs:PutSubscriptionFilter",
            "Resource": [
                "arn:aws:logs:*:*:log-group:*",
                "arn:aws:logs:*:123456789012:destination:sampleDestination"
            ]
        }
    ]
```

```
}
```

步驟 4：建立訂閱篩選條件

切換到傳送端帳戶，在此範例中是 111111111111。您現在將在傳送端帳戶中建立訂閱篩選條件。在此範例中，篩選器與包含 AWS CloudTrail 事件的記錄群組相關聯，因此，「根」 AWS 認證所做的每個記錄活動都會傳送至您先前建立的目的地。如需如何將 AWS CloudTrail 事件傳送至 CloudWatch 記錄檔的相關資訊，請參閱《AWS CloudTrail 使用指南》中的〈將 CloudTrail事件傳送至 CloudWatch 記錄檔〉。

當您輸入以下命令時，確認您以 IAM 使用者身分登入，或是使用您在 步驟 3：新增/驗證跨帳戶目的地的 IAM 許可 中為其新增政策的 IAM 角色登入。

```
aws logs put-subscription-filter \
    --log-group-name "aws-cloudtrail-logs-111111111111-300a971e" \
    --filter-name "firehose_test" \
    --filter-pattern "{$.userIdentity.type = AssumedRole}" \
    --destination-arn "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"
```

記錄群組和目的地必須位於相同的 AWS 區域。不過，目的地可以指向位於不同區域的 AWS 資源，例如 Firehose 串流。

驗證日誌事件的流程

建立訂閱篩選器之後， CloudWatch Logs 會將符合篩選器模式的所有傳入記錄事件轉寄至 Firehose 傳送串流。資料會根據 Firehose 交付串流上設定的時間緩衝區間隔開始顯示在 Amazon S3 儲存貯體中。一旦經過足夠的時間，您就可以檢查 Amazon S3 儲存貯體來驗證資料。若要檢查儲存貯體，請輸入以下命令：

```
aws s3api list-objects --bucket 'firehose-test-bucket1'
```

該命令的輸出類似如下：

```
{
    "Contents": [
        {
            "Key": "2021/02/02/08/my-delivery-
stream-1-2021-02-02-08-55-24-5e6dc317-071b-45ba-a9d3-4805ba39c2ba",
```

```
            "LastModified": "2021-02-02T09:00:26+00:00",
            "ETag": "\"EXAMPLEa817fb88fc770b81c8f990d\"",
            "Size": 198,
            "StorageClass": "STANDARD",
            "Owner": {
                "DisplayName": "firehose+2test",
                "ID": "EXAMPLE27fd05889c665d2636218451970ef79400e3d2aecca3adb1930042e0"
            }
        }
    ]
}
```

然後，您可以輸入下列命令，從儲存貯體擷取特定物件。將 key 的值換成您在前一個命令中找到的值。

```
aws s3api get-object --bucket 'firehose-test-bucket1' --key '2021/02/02/08/my-delivery-stream-1-2021-02-02-08-55-24-5e6dc317-071b-45ba-a9d3-4805ba39c2ba' testfile.gz
```

在 Amazon S3 物件中的資料會以 gzip 格式壓縮。您可以從命令列使用下列其中一個命令來檢查原始資料：

Linux：

```
zcat testfile.gz
```

macOS：

```
zcat <testfile.gz
```

在執行時間修改目的地成員資格

在某些情況下，您可能需要在您擁有的目的地中新增或移除日誌寄件者。您可以使用新存取原則對目的地使用此PutDestinationPolicy動作。在下列範例中，之前新增的帳戶 111111111111 會停止傳送更多日誌資料，且帳戶 333333333333 會啟用。

1. 擷取目前與目的地 testD estination 相關聯的原則，並記下 AccessPolicy：

```
aws logs describe-destinations \
    --destination-name-prefix "testFirehoseDestination"

{
```

```
    "destinations": [
        {
            "destinationName": "testFirehoseDestination",
            "targetArn": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream",
            "roleArn": "arn:aws:iam:: 222222222222:role/CWLtoKinesisFirehoseRole",
            "accessPolicy": "{\n  \"Version\" : \"2012-10-17\",\n  \"Statement
\" : [\n    {\n      \"Sid\" : \"\",\n      \"Effect\" : \"Allow\",\n
 \"Principal\" : {\n        \"AWS\" : \"111111111111 \"\n      },\n      \"Action
\" : \"logs:PutSubscriptionFilter\",\n      \"Resource\" : \"arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination\"\n    }\n  ]\n}\n\n",
            "arn": "arn:aws:logs:us-east-1:
 222222222222:destination:testFirehoseDestination",
            "creationTime": 1612256124430
        }
    ]
}
```

2. 更新政策，以反映帳戶 111111111111 已停用，且帳戶 333333333333 已啟用。將此策略放在~/NewAccessPolicy.json 文件中：

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "333333333333 "
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" : "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"
    }
  ]
}
```

3. 使用下列命令，將 NewAccessPolicy.json 檔案中定義的原則與目的地產生關聯：

```
aws logs put-destination-policy \
    --destination-name "testFirehoseDestination" \

    --access-policy file://~/NewAccessPolicy.json
```

這最終會停止來自帳戶 ID 111111111111 的日誌事件。在帳戶 333333333333 的擁有者建立訂閱篩選條件後，來自帳戶 ID 333333333333 的日誌事件會立刻開始流向目的地。

使用 Kinesis Data Streams 的跨帳戶跨區域帳戶層級訂閱

建立跨帳戶訂閱時，您可以指定單一帳戶或一個組織作為寄件者。如果您指定組織，則此程序會讓組織中的所有帳戶都能將日誌傳送至接收者帳戶。

若要跨帳戶共用日誌資料，您需要建立日誌資料寄件者和接收者：

- 記錄資料寄件者 — 從收件者取得目的地資訊，並讓 CloudWatch 記錄檔知道已準備好將記錄事件傳送至指定的目的地。在本節其餘部分的程序中，記錄資料傳送者會顯示虛構 AWS 帳號 1111111111。

  如果您要讓一個組織中的多個帳戶將日誌傳送至一個收件人帳戶，則可以建立一個政策，授予組織中所有帳戶將日誌傳送至收件人帳戶的許可。您仍然必須為每個寄件者帳戶設定個別的訂閱篩選條件。
- 記錄資料收件者 — 設定封裝 Kinesis Data Streams 串流的目的地，並讓 CloudWatch 記錄知道收件者想要接收記錄資料。然後，收件人接著會與寄件者共用與其目的地有關的資訊。在本節其餘部分的程序中，記錄資料收件者會顯示虛構的 AWS 帳戶號碼 999999999999。

若要開始接收跨帳戶使用者的記錄事件，記錄資料收件者會先建立 CloudWatch 記錄目的地。每個目的地包含以下關鍵元素：

目的地名稱

您要建立的目的地名稱。

目標 ARN

您要用作訂閱摘要目的地之 AWS 資源的 Amazon 資源名稱 (ARN)。

角色 ARN

授與 CloudWatch Log 必要權限的 AWS Identity and Access Management (IAM) 角色，可將資料放入所選串流。

存取政策

IAM 政策文件 (JSON 格式，使用 IAM 政策文法撰寫)，決定允許一組使用者寫入您的目的地。

> **Note**
>
> 記錄群組和目的地必須位於相同的 AWS 區域。不過，目的地指向的 AWS 資源可以位於不同的區域。在以下各節的範例中，區域特定的所有資源都在美國東部 (維吉尼亞北部) 建立。

主題

設定新的跨帳戶訂閱

按照這些章節中的步驟設定新的跨帳戶日誌訂閱。

主題

步驟 1：建立目的地

> **Important**
>
> 此程序中的所有步驟必須在日誌資料收件人帳戶中完成。

在此範例中，記錄資料收件者帳戶的帳戶識別碼為 999999999999，而記錄資料寄件者 AWS 帳戶識別碼是 AWS 111111111111。

此範例使用名為的 Kinesis Data Streams 建立目的地 RecipientStream，以及可讓記 CloudWatch 錄寫入資料的角色。

建立目的地時， CloudWatch Logs 會代表收件者帳戶將測試訊息傳送至目的地。稍後訂閱篩選器處於作用中狀態時， CloudWatch Logs 會代表來源帳戶將記錄事件傳送至目的地。

若要建立目的地

1. 在收件人帳戶中，在 Kinesis Data Streams 中建立目的地串流。在命令提示字元中輸入：

```
aws kinesis create-stream --stream-name "RecipientStream" --shard-count 1
```

2. 等到 串流變成作用中。您可以使用 aws 室運動描述流命令來檢查. StreamDescription StreamStatus財產。此外，請記下 StreamDescription.Streamarn 值，因為您稍後會將其傳送至 CloudWatch 記錄檔：

```
aws kinesis describe-stream --stream-name "RecipientStream"
{
  "StreamDescription": {
    "StreamStatus": "ACTIVE",
    "StreamName": "RecipientStream",
    "StreamARN": "arn:aws:kinesis:us-east-1:999999999999:stream/RecipientStream",
    "Shards": [
      {
        "ShardId": "shardId-000000000000",
        "HashKeyRange": {
          "EndingHashKey": "340282366920938463463374607431768EXAMPLE",
          "StartingHashKey": "0"
        },
        "SequenceNumberRange": {
          "StartingSequenceNumber":
 "49551135218688818456679503831981458784591352702 18EXAMPLE"
        }
      }
    ]
  }
}
```

這可能需要花費幾分鐘，讓串流以作用中狀態出現。

3. 建立 IAM 角色，以授與 CloudWatch 記錄將資料放入串流的權限。首先，您需要在〜/TrustPolicyForCW L.json 文件中創建一個信任策略。使用文字編輯器來建立此政策檔案，請勿使用 IAM 主控台。

   此政策包含 `aws:SourceArn` 全域條件內容金鑰，可指定 `sourceAccountId` 以協助預防混淆代理人安全問題。如果您在第一次呼叫中還不知道來源帳戶 ID，我們建議您將目的地 ARN 放在來源 ARN 欄位中。在後續呼叫中，應將來源 ARN 設定為從第一次呼叫中收集的實際來源 ARN。如需詳細資訊，請參閱 預防混淆代理人。

```
{
    "Statement": {
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.amazonaws.com"
        },
        "Condition": {
            "StringLike": {
                "aws:SourceArn": [
                    "arn:aws:logs:region:sourceAccountId:*",
                    "arn:aws:logs:region:recipientAccountId:*"
                ]
            }
        },
        "Action": "sts:AssumeRole"
    }
}
```

4. 使用 aws iam create-role 命令來建立 IAM 角色，並指定信任政策檔案。請記下傳回的 Role.Arn 值，因為它也會在稍後傳送至 CloudWatch 記錄檔：

```
aws iam create-role \
--role-name CWLtoKinesisRole \
--assume-role-policy-document file://~/TrustPolicyForCWL.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Condition": {
                    "StringLike": {
                        "aws:SourceArn": [
                            "arn:aws:logs:region:sourceAccountId:*",
                            "arn:aws:logs:region:recipientAccountId:*"
                        ]
                    }
                },
                "Principal": {
                    "Service": "logs.amazonaws.com"
                }
```

```
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2023-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisRole",
        "Path": "/",
        "Arn": "arn:aws:iam::999999999999:role/CWLtoKinesisRole"
    }
}
```

5. 建立權限原則，以定義 CloudWatch 記錄檔可對您的帳戶執行哪些動作。首先，使用文本編輯器在文件~/PermissionsForCW L.json 中創建權限策略：

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:region:999999999999:stream/RecipientStream"
    }
  ]
}
```

6. 使用 aws iam put-role-policy 命令將許可政策與角色相關聯：

```
aws iam put-role-policy \
    --role-name CWLtoKinesisRole \
    --policy-name Permissions-Policy-For-CWL \
    --policy-document file://~/PermissionsForCWL.json
```

7. 串流處於使用中狀態且您已建立 IAM 角色之後，您可以建立 CloudWatch 記錄目的地。

   a. 此步驟不會將存取政策與您的目的地相關聯，且只是完成目的地建立兩步驟中的第一步。記下DestinationArn有效負載中返回的內容：

```
aws logs put-destination \
    --destination-name "testDestination" \
    --target-arn "arn:aws:kinesis:region:999999999999:stream/RecipientStream" \
    --role-arn "arn:aws:iam::999999999999:role/CWLtoKinesisRole"

{
  "DestinationName" : "testDestination",
  "RoleArn" : "arn:aws:iam::999999999999:role/CWLtoKinesisRole",
```

```
  "DestinationArn" : "arn:aws:logs:us-
east-1:999999999999:destination:testDestination",
  "TargetArn" : "arn:aws:kinesis:us-east-1:999999999999:stream/RecipientStream"
}
```

b. 步驟 7a 完成後，即可在日誌資料收件人帳戶中，將存取政策與目的地建立關聯。此原則必須指定 log: PutSubscriptionFilter 動作，並授與寄件者帳戶存取目的地的權限。

此原則會授與傳送記錄檔之 AWS 帳戶的權限。您可以在政策中僅指定這一個帳戶，或者如果寄件者帳戶是組織的成員，則政策可以指定組織的組織 ID。如此一來，您可以僅建立一個政策，就能允許一個組織中的多個帳戶將日誌傳送至此目的地帳戶。

使用文字編輯器建立名為 ~/AccessPolicy.json 的檔案，並隨附下列其中一個政策陳述。

此第一個範例政策允許組織中具有 ID 為 o-1234567890 的所有帳戶將日誌傳送至收件人帳戶。

```
{
    "Version" : "2012-10-17",
    "Statement" : [
        {
            "Sid" : "",
            "Effect" : "Allow",
            "Principal" : "*",
            "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
            "Resource" :
 "arn:aws:logs:region:999999999999:destination:testDestination",
            "Condition": {
               "StringEquals" : {
                   "aws:PrincipalOrgID" : ["o-1234567890"]
                }
            }
        }
    ]
}
```

此下一個範例只允許日誌資料寄件者帳戶 (111111111111) 將日誌傳送至日誌資料收件人帳戶。

```
{
  "Version" : "2012-10-17",
  "Statement" : [
```

```
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111111111111"
      },
      "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
      "Resource" :
 "arn:aws:logs:region:999999999999:destination:testDestination"
    }
  ]
}
```

c. 將您在上一步驟建立的政策連接到目的地。

```
aws logs put-destination-policy \
    --destination-name "testDestination" \
    --access-policy file://~/AccessPolicy.json
```

*####### AWS ####### 111111111 ###### ARN arn: aw: ##:# #:99999999999999: ###:**PutSubscriptionFilter**########*任何其他用戶嘗試撥打此目 PutSubscriptionFilter 的地的嘗試將被拒絕。

若要驗證使用者的權限符合存取政策，請參閱《IAM 使用者指南》中的使用政策驗證程式。

完成後，如果您正在使用 AWS Organizations 跨帳戶權限，請按照中步驟 2：(僅限於使用組織時) 建立 IAM 角色的步驟操作。如果您要將許可直接授予給其他帳戶，而不是使用 Organizations，則可以略過該步驟並繼續進行 步驟 3：建立帳戶層級訂閱篩選政策。

步驟 2：(僅限於使用組織時) 建立 IAM 角色

在上一節中，如果您藉由使用授予許可給帳戶 111111111111 所屬組織的存取政策來建立目的地，而不是將許可直接授予給帳戶 111111111111，則按照本節中的步驟進行。若否，則可跳至步驟 步驟 3：建立帳戶層級訂閱篩選政策。

本節中的步驟會建立 IAM 角色，該角色 CloudWatch 可假設並驗證寄件者帳戶是否具有針對收件者目的地建立訂閱篩選器的權限。

在寄件者帳戶中執行此區段中的步驟。角色必須存在於寄件者帳戶中，而且您要在訂閱篩選條件中指定此角色的 ARN。在此範例中，使用者帳戶為 111111111111。

使用 AWS Organizations建立跨帳戶日誌訂閱所需的 IAM 角色

1. 在檔案 /TrustPolicyForCWLSubscriptionFilter.json 中建立下列信任政策。使用文字編輯器來建立此政策檔案，請勿使用 IAM 主控台。

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.amazonaws.com" },
    "Action": "sts:AssumeRole"
  }
}
```

2. 建立使用此政策的 IAM 角色。記下命令傳回的 Arn 值，之後在此程序中會用到。在此範例中，我們使用 CWLtoSubscriptionFilterRole 作為要建立的角色的名稱。

```
aws iam create-role \
     --role-name CWLtoSubscriptionFilterRole \
     --assume-role-policy-document file://~/
TrustPolicyForCWLSubscriptionFilter.json
```

3. 建立權限原則，以定義記 CloudWatch 錄檔可在您的帳戶上執行的動作。

   a. 首先，使用文字編輯器在名為 ~/PermissionsForCWLSubscriptionFilter.json 的檔案中建立下列許可政策。

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "logs:PutLogEvents",
            "Resource": "arn:aws:logs:region:111111111111:log-
group:LogGroupOnWhichSubscriptionFilterIsCreated:*"
        }
    ]
}
```

   b. 輸入下列命令，將您剛建立的許可政策與您在步驟 2 中建立的角色相關聯。

```
aws iam put-role-policy
    --role-name CWLtoSubscriptionFilterRole
    --policy-name Permissions-Policy-For-CWL-Subscription-filter
```

```
    --policy-document file://~/PermissionsForCWLSubscriptionFilter.json
```

完成後，可以繼續進行 步驟 3：建立帳戶層級訂閱篩選政策。

步驟 3：建立帳戶層級訂閱篩選政策

在建立目的地後，日誌資料收件人帳戶可以與其他 AWS 帳戶共用目的地 ARN (arn:aws:logs:us-east-1:999999999999:destination:testDestination)，讓他們能將日誌事件傳送到相同目的地。然後，這些其他傳送帳戶使用者接著會在個別的日誌群組針對此目的地建立訂閱篩選條件。訂閱篩選條件會立即開始將即時日誌資料從所選的日誌群組傳送到指定的目標。

> **Note**
>
> 如果您要將訂閱篩選條件的許可授予給整個組織，您需要使用您在 步驟 2：(僅限於使用組織時) 建立 IAM 角色 中建立的 IAM 角色 ARN。

在下列範例中，會在傳送帳戶中建立帳戶層級的訂閱篩選原則。篩選器會與寄件者帳戶相關聯，111111111111因此每個符合篩選條件和選取準則的記錄事件都會傳送至您先前建立的目的地。該目的地封裝了一個名為「RecipientStream」的流。

此selection-criteria欄位是選擇性欄位，但對於排除可能導致訂閱篩選器無限記錄遞迴的記錄群組而言非常重要。如需有關此問題以及判斷要排除哪些記錄群組的詳細資訊，請參閱防止記錄遞迴。目前，NOT IN 是唯一支援的運算子selection-criteria。

```
aws logs put-account-policy \
    --policy-name "CrossAccountStreamsExamplePolicy" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document
 '{"DestinationArn":"arn:aws:logs:region:999999999999:destination:testDestination",
 "FilterPattern": "", "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

寄件者帳戶的記錄群組和目的地必須位於相同的 AWS 區域。但是，目標可以指向位於不同區域的 AWS 資源，例如 Kinesis Data Streams 流。

驗證日誌事件的流動

建立帳戶層級訂閱篩選原則之後， CloudWatch 記錄會將符合篩選器模式和選取準則的所有內送記錄事件轉寄至封裝在目的地資料流中的資料流 (稱為 "」)。RecipientStream目的地擁有者可以使用 aws kinesis get-shard-iterator 命令擷取 Kinesis 資料串流碎片，並使用 aws kinesis 取得記錄命令來擷取一些 Kinesis Data Streams 記錄來確認是否發生這種情況：

```
aws kinesis get-shard-iterator \
      --stream-name RecipientStream \
      --shard-id shardId-000000000000 \
      --shard-iterator-type TRIM_HORIZON

{
    "ShardIterator":
    "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
}

aws kinesis get-records \
      --limit 10 \
      --shard-iterator
      "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
```

> **Note**
>
> Kinesis 資料串流開始傳回資料之前，您可能需要重新執行幾次`get-records`命令。

您應該會看到一系列 Kinesis Data Streams 記錄的回應。Kinesis Data Streams 記錄中的資料屬性，是以 gzip 格式壓縮並採用 Base64 編碼。您可以使用以下 Unix 命令來透過命令列檢查原始資料：

```
echo -n "<Content of Data>" | base64 -d | zcat
```

Base64 解碼和解壓縮資料是以 JSON 形式並以下列結構進行格式化：

```
{
```

```
    "owner": "111111111111",
    "logGroup": "CloudTrail/logs",
    "logStream": "111111111111_CloudTrail/logs_us-east-1",
    "subscriptionFilters": [
        "RecipientStream"
    ],
    "messageType": "DATA_MESSAGE",
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        }
    ]
}
```

數據結構中的關鍵要素如下：

messageType

資料訊息將使用「DATA_MESSAGE」類型。有時，記 CloudWatch 錄檔可能會發出具有「CONTROL_MESSAGE」類型的 Kinesis Data Streams 記錄，主要用於檢查目的地是否可連線。

owner

原始記錄檔資料的 AWS 帳戶 ID。

logGroup

原始日誌資料的日誌群組名稱。

logStream

原始日誌資料的日誌串流名稱。

subscriptionFilters

與原始日誌資料相符的訂閱篩選條件名稱清單。

logEvents

實際的日誌資料，以一系列的日誌事件記錄呈現。「id」屬性是每個記錄事件的唯一識別符。

政策層級

強制執行策略的層級。「帳戶層級 _ 政策」是用於帳戶層級的policyLevel訂閱過濾策略。

在執行期修改目的地成員資格

您可能遇到以下情況：您需要從擁有的目的地中新增或移除某些使用者的成員資格。您可以在目的地使用 put-destination-policy 命令並指定新的存取政策。在下列範例中，之前新增帳戶 111111111111 會停止傳送任何更多資料，且帳戶 222222222222 會啟用。

1. 擷取目前與目的地 testD estination 相關聯的原則，並記下 AccessPolicy：

```
aws logs describe-destinations \
    --destination-name-prefix "testDestination"

{
 "Destinations": [
   {
     "DestinationName": "testDestination",
     "RoleArn": "arn:aws:iam::999999999999:role/CWLtoKinesisRole",
     "DestinationArn":
 "arn:aws:logs:region:999999999999:destination:testDestination",
     "TargetArn": "arn:aws:kinesis:region:999999999999:stream/RecipientStream",
     "AccessPolicy": "{\"Version\": \"2012-10-17\", \"Statement\":
 [{\"Sid\": \"\", \"Effect\": \"Allow\", \"Principal\": {\"AWS\":
 \"111111111111\"}, \"Action\": \"logs:PutSubscriptionFilter\", \"Resource\":
 \"arn:aws:logs:region:999999999999:destination:testDestination\"}] }"
   }
 ]
}
```

2. 更新政策，以反映帳戶 111111111111 已停用，且帳戶 222222222222 已啟用。將此策略放在~/NewAccessPolicy.json 文件中：

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "222222222222"
      },
      "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
      "Resource" : "arn:aws:logs:region:999999999999:destination:testDestination"
    }
  ]
}
```

3. 呼叫PutDestinationPolicy將 NewAccessPolicy.json 檔案中定義的原則與目的地產生關聯：

```
aws logs put-destination-policy \
--destination-name "testDestination" \
--access-policy file://~/NewAccessPolicy.json
```

這最終會從帳戶 ID 111111111111 停用日誌事件。在帳戶 222222222222 的擁有者建立訂閱篩選條件後，來自帳戶 ID 222222222222 的日誌事件會立刻開始流向目的地。

更新現有的跨帳戶訂閱

如果您目前有跨帳戶日誌訂閱，其中目的地帳戶僅授予特定寄件者帳戶許可，而您想要更新此訂閱，讓目的地帳戶授予組織中所有帳戶的存取權，請按照本節中的步驟進行。

主題

- 步驟 1：更新訂閱篩選條件
- 步驟 2：更新現有的目的地存取政策

步驟 1：更新訂閱篩選條件

> **Note**
>
> 只有跨帳戶訂閱由 啟用從 AWS 服務記錄 所列服務建立的日誌才需要此步驟。如果您不使用這些日誌群組之一建立的日誌，則可以跳至 步驟 2：更新現有的目的地存取政策。

在某些情況下，您必須更新所有傳送日誌至目的地帳戶的寄件者帳戶中的訂閱篩選條件。此更新新增 IAM 角色，該角色 CloudWatch 可假設並驗證寄件者帳戶是否具有將記錄傳送至收件者帳戶的權限。

針對您想要更新的每個寄件者帳戶，按照本節中的步驟進行，以將組織 ID 用於跨帳戶訂閱許可。

在本節的範例中，111111111111 和 222222222222 兩個帳戶已經建立訂閱篩選條件，以將日誌傳送至帳戶 999999999999。現有的訂閱篩選條件值如下：

```
## Existing Subscription Filter parameter values
{
    "DestinationArn": "arn:aws:logs:region:999999999999:destination:testDestination",
    "FilterPattern": "{$.userIdentity.type = Root}",
    "Distribution": "Random"
}
```

如果需要尋找目前的訂閱篩選條件參數值，請輸入下列命令。

```
aws logs describe-account-policies \
--policy-type "SUBSCRIPTION_FILTER_POLICY" \
--policy-name "CrossAccountStreamsExamplePolicy"
```

更新訂閱篩選條件以開始將組織 ID 用於跨帳戶日誌許可

1. 在檔案 ~/TrustPolicyForCWL.json 中建立下列信任政策。使用文字編輯器來建立此政策檔案，請勿使用 IAM 主控台。

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.amazonaws.com" },
    "Action": "sts:AssumeRole"
  }
```

```
}
```

2. 建立使用此政策的 IAM 角色。記下命令傳回的 Arn 值的 Arn 值，之後在此程序中會用到。在此範例中，我們使用 CWLtoSubscriptionFilterRole 作為要建立的角色的名稱。

```
aws iam create-role
    \ --role-name CWLtoSubscriptionFilterRole
    \ --assume-role-policy-document file://~/TrustPolicyForCWL.json
```

3. 建立權限原則，以定義記 CloudWatch 錄檔可在您的帳戶上執行的動作。

   a. 首先，使用文字編輯器在名為 /PermissionsForCWLSubscriptionFilter.json 的檔案中建立下列許可政策。

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "logs:PutLogEvents",
            "Resource": "arn:aws:logs:region:111111111111:log-
group:LogGroupOnWhichSubscriptionFilterIsCreated:*"
        }
    ]
}
```

   b. 輸入下列命令，將您剛建立的許可政策與您在步驟 2 中建立的角色相關聯。

```
aws iam put-role-policy
    --role-name CWLtoSubscriptionFilterRole
    --policy-name Permissions-Policy-For-CWL-Subscription-filter
    --policy-document file://~/PermissionsForCWLSubscriptionFilter.json
```

4. 輸入下列命令以更新訂閱篩選原則。

```
aws logs put-account-policy \
    --policy-name "CrossAccountStreamsExamplePolicy" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document
 '{"DestinationArn":"arn:aws:logs:region:999999999999:destination:testDestination",
 "FilterPattern": "{$.userIdentity.type = Root}", "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
```

```
    --scope "ALL"
```

步驟 2：更新現有的目的地存取政策

更新所有寄件者帳戶中的訂閱篩選條件之後，您可以更新收件人帳戶中的目的地存取政策。

下列範例中，收件人帳戶為 999999999999，且目的地名稱為 testDestination。

此更新可讓屬於組織且 ID 為 o-1234567890 的所有帳戶傳送日誌至收件人帳戶。只有已建立訂閱篩選條件的帳戶才會真的傳送日誌至收件人帳戶。

更新收件人帳戶中的目的地存取政策，以開始將組織 ID 用於許可

1. 在收件人帳戶中，使用文字編輯器建立 ~/AccessPolicy.json 檔案，其中包含以下內容。

```
{
"Version" : "2012-10-17",
    "Statement" : [
        {
"Sid" : "",
            "Effect" : "Allow",
            "Principal" : "*",
            "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
            "Resource" :
 "arn:aws:logs:region:999999999999:destination:testDestination",
            "Condition": {
"StringEquals" : {
"aws:PrincipalOrgID" : ["o-1234567890"]
                }
            }
        }
    ]
}
```

2. 輸入下列命令，將您剛建立的政策連接到現有的目的地。若要更新目的地，以使用具有組織 ID 的存取政策，而不是列出特定 AWS 帳戶 ID 的存取政策，則應包含 force 參數。

> **Warning**
>
> 如果您正在使用中列出的 AWS 服務傳送的記錄檔啟用從 AWS 服務記錄，則在執行此步驟之前，您必須先更新所有寄件者帳戶中的訂閱篩選器，如中所述步驟 1：更新訂閱篩選條件。

```
aws logs put-destination-policy
    \ --destination-name "testDestination"
    \ --access-policy file://~/AccessPolicy.json
    \ --force
```

使用 Firehose 進行跨帳戶跨區域帳戶層級訂閱

若要跨帳戶共用日誌資料，您需要建立日誌資料寄件者和接收者：

- 記錄資料寄件者 — 從收件者取得目的地資訊，並讓 CloudWatch 記錄檔知道已準備好將其記錄事件傳送至指定的目的地。在本節其餘部分的程序中，記錄資料傳送者會顯示虛構 AWS 帳號 1111111111。
- 記錄資料收件者 — 設定封裝 Kinesis Data Streams 串流的目的地，並讓 CloudWatch 記錄知道收件者想要接收記錄資料。然後，收件人接著會與寄件者共用與其目的地有關的資訊。在本節其餘部分的程序中，記錄資料收件者會以虛構的 AWS 帳號 2222222222 顯示。

本節中的範例使用 Firehose 交付串流搭配 Amazon S3 儲存。您也可以使用不同的設定來設定 Firehose 交付串流。如需詳細資訊，請參閱建立 Firehose 傳遞串流。

> **Note**
>
> 記錄群組和目的地必須位於相同的 AWS 區域。不過，目的地指向的 AWS 資源可以位於不同的區域。

> **Note**
>
> 支援相同帳戶和跨區域交付串流的 Firehose 訂閱篩選器。

主題

步驟 1：建立 Firehose 傳送串流

> **Important**
>
> 在完成下列步驟之前，您必須使用存取政策，以便 Firehose 可以存取您的 Amazon S3 儲存貯體。如需詳細資訊，請參閱 Amazon 資料 Firehose 開發人員指南中的控制存取。
> 必須在日誌資料收件人帳戶中完成本區段 (步驟 1) 中的所有步驟。
> 在以下範例命令中使用美國東部 (維吉尼亞北部)。請將此區域替換成您部署的正確區域。

若要建立要用作目的地的 Firehose 傳送串流

1. 建立 Amazon S3 儲存貯體：

```
aws s3api create-bucket --bucket firehose-test-bucket1 --create-bucket-
configuration LocationConstraint=us-east-1
```

2. 建立 IAM 角色，以授予 Firehose 將資料放入值區的權限。

   a. 首先，使用文字編輯器在檔案 ~/TrustPolicyForFirehose.json 中建立信任政策。

```
{ "Statement": { "Effect": "Allow", "Principal": { "Service":
 "firehose.amazonaws.com" }, "Action": "sts:AssumeRole", "Condition":
 { "StringEquals": { "sts:ExternalId":"222222222222" } } } }
```

   b. 建立 IAM 角色，並指定您剛建立的信任政策檔案。

```
aws iam create-role \
    --role-name FirehosetoS3Role \
    --assume-role-policy-document file://~/TrustPolicyForFirehose.json
```

c. 此命令的輸出看起來如下：記下角色名稱和角色 ARN。

```
{
    "Role": {
        "Path": "/",
        "RoleName": "FirehosetoS3Role",
        "RoleId": "AROAR3BXASEKW7K635M53",
        "Arn": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
        "CreateDate": "2021-02-02T07:53:10+00:00",
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Effect": "Allow",
                "Principal": {
                    "Service": "firehose.amazonaws.com"
                },
                "Action": "sts:AssumeRole",
                "Condition": {
                    "StringEquals": {
                        "sts:ExternalId": "222222222222"
                    }
                }
            }
        }
    }
}
```

3. 建立權限原則，以定義 Firehose 可在您帳戶中執行的動作。

a. 首先，使用文字編輯器在名為 ~/PermissionsForFirehose.json 的檔案中建立下列許可政策。根據您的使用案例，可能需要為此檔案新增更多許可。

```
{
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "s3:PutObject",
            "s3:PutObjectAcl",
            "s3:ListBucket"
        ],
        "Resource": [
            "arn:aws:s3:::firehose-test-bucket1",
            "arn:aws:s3:::firehose-test-bucket1/*"
        ]
```

```
    }]
}
```

b. 輸入下列命令，將您剛建立的許可政策與 IAM 角色相關聯。

```
aws iam put-role-policy --role-name FirehosetoS3Role --policy-name
 Permissions-Policy-For-Firehose-To-S3 --policy-document file://~/
PermissionsForFirehose.json
```

4. 輸入下列命令以建立 Firehose 傳送串流。以正確*my-bucket-arn*的部署值取代*my-role-arn*和。

```
aws firehose create-delivery-stream \
   --delivery-stream-name 'my-delivery-stream' \
   --s3-destination-configuration \
  '{"RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role", "BucketARN":
 "arn:aws:s3:::firehose-test-bucket1"}'
```

輸出格式應類似以下內容：

```
{
    "DeliveryStreamARN": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream"
}
```

步驟 2：建立目的地

> **Important**
>
> 此程序中的所有步驟必須在日誌資料收件人帳戶中完成。

建立目的地時，CloudWatch Logs 會代表收件者帳戶將測試訊息傳送至目的地。稍後訂閱篩選器處於作用中狀態時，CloudWatch Logs 會代表來源帳戶將記錄事件傳送至目的地。

若要建立目的地

1. 等到您在其中建立的 Firehose 串流步驟 1：建立 Firehose 傳送串流變為作用中。您可以使用以下命令來檢查 StreamDescription. StreamStatus財產。

```
aws firehose describe-delivery-stream --delivery-stream-name "my-delivery-stream"
```

此外，請注意的DeliveryStreamDescription。 DeliveryStreamARN 值，因為您將需要在後面的步驟中使用它。此命令的範例輸出：

```
{
    "DeliveryStreamDescription": {
        "DeliveryStreamName": "my-delivery-stream",
        "DeliveryStreamARN": "arn:aws:firehose:us-
east-1:222222222222:deliverystream/my-delivery-stream",
        "DeliveryStreamStatus": "ACTIVE",
        "DeliveryStreamEncryptionConfiguration": {
            "Status": "DISABLED"
        },
        "DeliveryStreamType": "DirectPut",
        "VersionId": "1",
        "CreateTimestamp": "2021-02-01T23:59:15.567000-08:00",
        "Destinations": [
            {
                "DestinationId": "destinationId-000000000001",
                "S3DestinationDescription": {
                    "RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
                    "BucketARN": "arn:aws:s3:::firehose-test-bucket1",
                    "BufferingHints": {
                        "SizeInMBs": 5,
                        "IntervalInSeconds": 300
                    },
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "CloudWatchLoggingOptions": {
                        "Enabled": false
                    }
                },
                "ExtendedS3DestinationDescription": {
                    "RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
                    "BucketARN": "arn:aws:s3:::firehose-test-bucket1",
                    "BufferingHints": {
                        "SizeInMBs": 5,
                        "IntervalInSeconds": 300
                    },
```

```
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "CloudWatchLoggingOptions": {
                        "Enabled": false
                    },
                    "S3BackupMode": "Disabled"
                }
            }
        ],
        "HasMoreDestinations": false
    }
}
```

可能需要花費幾分鐘，交付串流才會變成作用中狀態。

2. 當交付串流處於作用中狀態時，請建立 IAM 角色，以授與 CloudWatch 記錄檔將資料放入 Firehose 串流的權限。首先，您需要在~/TrustPolicyForCW L.json 文件中創建一個信任策略。請使用文字編輯器來建立此政策。如需有關 CloudWatch 日誌端點的詳細資訊，請參閱 Amazon CloudWatch 日誌端點和配額。

   此政策包含 aws:SourceArn 全域條件內容金鑰，可指定 sourceAccountId 以協助預防混淆代理人安全問題。如果您在第一次呼叫中還不知道來源帳戶 ID，我們建議您將目的地 ARN 放在來源 ARN 欄位中。在後續呼叫中，應將來源 ARN 設定為從第一次呼叫中收集的實際來源 ARN。如需詳細資訊，請參閱 預防混淆代理人。

```
{
    "Statement": {
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.amazonaws.com"
        },
        "Action": "sts:AssumeRole",
        "Condition": {
            "StringLike": {
                "aws:SourceArn": [
                    "arn:aws:logs:region:sourceAccountId:*",
                    "arn:aws:logs:region:recipientAccountId:*"
                ]
            }
        }
```

```
     }
}
```

3. 使用 aws iam create-role 命令來建立 IAM 角色，並指定您剛建立的信任政策檔案。

```
aws iam create-role \
      --role-name CWLtoKinesisFirehoseRole \
      --assume-role-policy-document file://~/TrustPolicyForCWL.json
```

下列為範例輸出。請留意傳回的 Role.Arn 值，因為後續步驟中需要用到。

```
{
    "Role": {
        "Path": "/",
        "RoleName": "CWLtoKinesisFirehoseRole",
        "RoleId": "AROAR3BXASEKYJYWF243H",
        "Arn": "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole",
        "CreateDate": "2023-02-02T08:10:43+00:00",
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.amazonaws.com"
                },
                "Action": "sts:AssumeRole",
                "Condition": {
                    "StringLike": {
                        "aws:SourceArn": [
                            "arn:aws:logs:region:sourceAccountId:*",
                            "arn:aws:logs:region:recipientAccountId:*"
                        ]
                    }
                }
            }
        }
    }
}
```

4. 建立權限原則，以定義 CloudWatch 記錄檔可對您的帳戶執行哪些動作。首先，使用文本編輯器在文件ㄧ/PermissionsForCW L.json 中創建權限策略：

```
{
```

```
    "Statement":[
      {
        "Effect":"Allow",
        "Action":["firehose:*"],
        "Resource":["arn:aws:firehose:region:222222222222:*"]
      }
    ]
}
```

5. 輸入以下命令，將許可政策與角色建立關聯：

```
aws iam put-role-policy --role-name CWLtoKinesisFirehoseRole --policy-name
 Permissions-Policy-For-CWL --policy-document file://~/PermissionsForCWL.json
```

6. Firehose 交付串流處於使用中狀態且您已建立 IAM 角色之後，您可以建立 CloudWatch 記錄目標。

   a. 此步驟不會將存取政策與您的目的地建立關聯，且為完成建立目的地之兩個步驟的僅第一個步驟。記下承載中傳回的新目的地的 ARN，因為您會在後續步驟中使用它作為 `destination.arn`。

```
aws logs put-destination \

    --destination-name "testFirehoseDestination" \
    --target-arn "arn:aws:firehose:us-east-1:222222222222:deliverystream/my-
delivery-stream" \
    --role-arn "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole"

{
    "destination": {
        "destinationName": "testFirehoseDestination",
        "targetArn": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream",
        "roleArn": "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole",
        "arn": "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"}
}
```

   b. 上一個步驟完成後，請在日誌資料收件人帳戶中 (222222222222)，將存取政策與目的地建立關聯。此政策可讓日誌資料寄件者帳戶 (111111111111) 只能在日誌資料收件人帳戶 (222222222222) 中存取目的地。您可以使用文字編輯器將此原則放入`~/AccessPolicy.json`檔案中：

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111111111111"
      },
      "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
      "Resource" : "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"
    }
  ]
}
```

c. 這會建立可定義誰擁有對目的地的寫入存取權之政策。此原則必須指定logs:PutSubscriptionFilter和logs:PutAccountPolicy動作才能存取目的地。跨帳戶使用者將使用PutSubscriptionFilter和PutAccountPolicy動作將記錄事件傳送至目的地。

```
aws logs put-destination-policy \
    --destination-name "testFirehoseDestination" \
    --access-policy file://~/AccessPolicy.json
```

步驟 3：建立帳戶層級訂閱篩選政策

切換到傳送端帳戶，在此範例中是 111111111111。現在，您將在發送帳戶中創建帳戶級訂閱過濾策略。在此範例中，篩選器會導致除了兩個記錄群組以外的所有記錄檔群組ERROR中包含字串的每個記錄事件傳送至您先前建立的目的地。

```
aws logs put-account-policy \
    --policy-name "CrossAccountFirehoseExamplePolicy" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document '{"DestinationArn":"arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination", "FilterPattern":
 "{$.userIdentity.type = AssumedRole}", "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
```

```
    --scope "ALL"
```

傳送帳戶的記錄群組和目的地必須位於相同的 AWS 區域。不過，目的地可以指向位於不同區域的 AWS 資源，例如 Firehose 串流。

驗證日誌事件的流程

建立訂閱篩選器之後， CloudWatch Logs 會將符合篩選器模式和選取條件的所有傳入記錄事件轉寄至 Firehose 傳送串流。資料會根據 Firehose 交付串流上設定的時間緩衝區間隔開始顯示在 Amazon S3 儲存貯體中。一旦經過足夠的時間，您就可以檢查 Amazon S3 儲存貯體來驗證資料。若要檢查儲存貯體，請輸入以下命令：

```
aws s3api list-objects --bucket 'firehose-test-bucket1'
```

該命令的輸出類似如下：

```
{
    "Contents": [
        {
            "Key": "2021/02/02/08/my-delivery-
stream-1-2021-02-02-08-55-24-5e6dc317-071b-45ba-a9d3-4805ba39c2ba",
            "LastModified": "2023-02-02T09:00:26+00:00",
            "ETag": "\"EXAMPLEa817fb88fc770b81c8f990d\"",
            "Size": 198,
            "StorageClass": "STANDARD",
            "Owner": {
                "DisplayName": "firehose+2test",
                "ID": "EXAMPLE27fd05889c665d2636218451970ef79400e3d2aecca3adb1930042e0"
            }
        }
    ]
}
```

然後，您可以輸入下列命令，從儲存貯體擷取特定物件。將 key 的值換成您在前一個命令中找到的值。

```
aws s3api get-object --bucket 'firehose-test-bucket1' --key '2021/02/02/08/my-delivery-
stream-1-2021-02-02-08-55-24-5e6dc317-071b-45ba-a9d3-4805ba39c2ba' testfile.gz
```

在 Amazon S3 物件中的資料會以 gzip 格式壓縮。您可以從命令列使用下列其中一個命令來檢查原始資料：

Linux：

```
zcat testfile.gz
```

macOS：

```
zcat <testfile.gz
```

在執行時間修改目的地成員資格

在某些情況下，您可能需要在您擁有的目的地中新增或移除日誌寄件者。您可以在目的地上使用PutDestinationPolicy和PutAccountPolicy動作搭配新的存取原則。在下列範例中，之前新增的帳戶 111111111111 會停止傳送更多日誌資料，且帳戶 333333333333 會啟用。

1. 擷取目前與目的地 testD estination 相關聯的原則，並記下 AccessPolicy：

   ```
   aws logs describe-destinations \
       --destination-name-prefix "testFirehoseDestination"
   ```

   返回的數據可能如下所示。

   ```
   {
       "destinations": [
           {
               "destinationName": "testFirehoseDestination",
               "targetArn": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
   my-delivery-stream",
               "roleArn": "arn:aws:iam:: 222222222222:role/CWLtoKinesisFirehoseRole",
               "accessPolicy": "{\n  \"Version\" : \"2012-10-17\",\n  \"Statement
   \" : [\n    {\n      \"Sid\" : \"\",\n      \"Effect\" : \"Allow\",\n
    \"Principal\" : {\n        \"AWS\" : \"111111111111 \"\n      },\n      \"Action
   \" : \"logs:PutSubscriptionFilter\",\n      \"Resource\" : \"arn:aws:logs:us-
   east-1:222222222222:destination:testFirehoseDestination\"\n    }\n  ]\n}\n\n",
               "arn": "arn:aws:logs:us-east-1:
    222222222222:destination:testFirehoseDestination",
               "creationTime": 1612256124430
           }
       ]
   }
   ```

2. 更新政策，以反映帳戶 111111111111 已停用，且帳戶 333333333333 已啟用。將此策略放在~/NewAccessPolicy.json 文件中：

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "333333333333 "
      },
      "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
      "Resource" : "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"
    }
  ]
}
```

3. 使用下列命令，將 NewAccessPolicy.json 檔案中定義的原則與目的地產生關聯：

```
aws logs put-destination-policy \
    --destination-name "testFirehoseDestination" \

    --access-policy file://~/NewAccessPolicy.json
```

這最終會停止來自帳戶 ID 111111111111 的日誌事件。在帳戶 333333333333 的擁有者建立訂閱篩選條件後，來自帳戶 ID 333333333333 的日誌事件會立刻開始流向目的地。

預防混淆代理人

混淆代理人問題屬於安全性議題，其中沒有執行動作許可的實體可以強制具有更多許可的實體執行該動作。在中 AWS，跨服務模擬可能會導致混淆的副問題。在某個服務 (呼叫服務) 呼叫另一個服務 (被呼叫服務) 時，可能會發生跨服務模擬。可以操縱呼叫服務來使用其許可，以其不應有存取許可的方式對其他客戶的資源採取動作。為了防止這種情況發生， AWS 提供的工具可協助您透過已授予您帳戶中資源存取權的服務主體來保護所有服務的資料。

我們建議在資源策略中使用aws:SourceArnaws:SourceAccountaws:SourceOrgID、、和aws:SourceOrgPaths全域條件前後關聯鍵字，以限制將其他服務提供給資源的權限。用於僅aws:SourceArn將一個資源與跨服務存取相關聯。用於aws:SourceAccount讓該帳號中的任何

資源與跨服務使用相關聯。用於aws:SourceOrgID允許組織內任何帳號的任何資源與跨服務使用相關聯。用於aws:SourceOrgPaths將 AWS Organizations 路徑中帳號的任何資源與跨服務使用相關聯。如需有關使用和瞭解路徑的詳細資訊，請參閱瞭解 AWS Organizations 實體路徑。

防範混淆代理人問題的最有效方法是使用 aws:SourceArn 全域條件內容索引鍵，以及資源的完整 ARN。如果不知道資源的完整 ARN，或者如果您指定了多個資源，請使用 aws:SourceArn 全域內容條件索引鍵搭配萬用字元 (*) 來表示 ARN 的未知部分。例如 arn:aws:*servicename*:*:*123456789012*:*。

如果 aws:SourceArn 值不包含帳戶 ID (例如 Amazon S3 儲存貯體 ARN)，您必須同時使用 aws:SourceAccount 和 aws:SourceArn 來限制許可。

若要大規模防範混淆代理人問題，請在資源型政策中使用 aws:SourceOrgID 或 aws:SourceOrgPaths 全域條件內容鍵和資源的組織 ID 或組織路徑。當您新增、移除或移動組織中的帳戶時，包含 aws:SourceOrgID 或 aws:SourceOrgPaths 鍵的政策將會自動包含正確的帳戶，您無需手動更新政策。

在中授予對記錄的存取權以將資料寫入 Kinesis Data Streams 和 Firehose 的政策所記 CloudWatch 錄，步驟 1：建立目的地並說步驟 2：建立目的地明如何使用 aws: SourceArn 全域條件內容金鑰來協助防止混淆的副問題。

防止記錄遞迴

使用訂閱篩選器可能會造成無限記錄遞迴的風險，這可能會導致 CloudWatch 記錄檔和目的地的擷取計費大幅增加 (如果未遭到阻止)。當訂閱篩選器與因訂閱傳遞工作流程而接收記錄事件的記錄群組相關聯時，就會發生這種情況。擷取至記錄群組的記錄會傳遞至目的地，造成記錄群組擷取更多記錄檔，然後再次轉送至目的地，建立遞迴迴圈。

例如，假設使用目的地為 Firehose 的訂閱篩選器，該篩選器會將日誌事件傳送到 Amazon S3。此外，還有一個 Lambda 函數可以處理交付到 Amazon S3 的新事件，並自行產生一些日誌。如果將訂閱篩選器套用至 Lambda 函數的日誌群組，則函數產生的日誌事件會轉送到目的地的 Firehose 和 Amazon S3，然後再次叫用函數，進而產生更多日誌並將其轉送到 Firehose 和 Amazon S3，從而導致另一次呼叫函數等。這會在無限迴圈中發生，導致日誌擷取、Firehose 和 Amazon S3 的計費意外增加。

如果 Lambda 函數連接至已為記錄啟用流程記錄的 VPC，則 VPC 的 CloudWatch 記錄群組也可能導致記錄遞迴。

建議您不要將訂閱篩選器套用至屬於訂閱傳送工作流程一部分的記錄群組。對於帳戶層級訂閱篩選器，請使用 PutAccountPolicy API 中的selectionCriteria參數將這些記錄群組從政策中排除。

排除記錄群組時，請考慮下列產生記錄檔的 AWS 服務，且可能是訂閱傳遞工作流程的一部分：

- Amazon EC2 與 Fargate
- Lambda
- AWS Step Functions
- 已針對日誌啟用的 CloudWatch Amazon VPC 流程日誌

> **Note**
>
> 針對帳戶層級訂閱篩選原則，Lambda 目的地的日誌群組產生的記錄事件不會轉送回 Lambda 函數。在這種情況下，帳戶訂閱政策不需要使用`selectionCriteria`目的地 Lambda 函數的日誌群組。

用於指標篩選條件、訂閱篩選條件、篩選條件日誌事件和 Live Tail 的篩選條件模式語法

> **Note**
>
> 如需如何使用 Amazon CloudWatch 日誌洞見查詢語言查詢日誌群組的相關資訊，請參閱CloudWatch 日誌見解查詢語法。

透過 CloudWatch 記錄，您可以使用指標篩選器將記錄資料轉換為可採取動作的指標、訂閱篩選器將記錄事件路由傳送至其他 AWS 服務、篩選記錄事件以搜尋記錄事件，以及 Live Tail 在擷取記錄時以互動方式即時檢視記錄檔。

篩選條件模式構成了指標篩選條件、訂閱篩選條件、篩選條件日誌事件和 Live Tail 用來比對日誌事件中詞彙的語法。詞彙可以是單字、完全相符片語或數值。規則運算式 (regex) 可用於建立獨立的篩選條件模式，或與 JSON 和以空格分隔的篩選條件模式整合。

使用要比對的詞彙建立篩選條件模式。篩選條件模式僅傳回包含您所定義的詞彙的日誌事件。您可以在 CloudWatch 主控台中測試篩選器模式。

主題

支援的規則運算式 (regex) 語法

支援的 regex 語法

使用 regex 搜尋和篩選日誌資料時，必須用 % 括住運算式。

包含 regex 的篩選條件模式只能包括下列字元：

- 英數字元 – 英數字元包含字母 (A 到 Z 或 a 到 z) 或數字 (0 到 9) 字元。

- 支援的符號字元 – 包括：'_'、'#'、'='、'@'、'/'、';'、',' 和 '-'。例如，由於不支援 '!'，因此 %something!% 會遭拒。
- 支援的運算子 - 包括：'^'、'$'、'?'、'['、']'、'{'、'}'、'|'、'\'、'*'、'+' 和 '.'。

不支援運算子（ 和 ）。您不可以使用括號來定義子模式。

不支援多位元組字元。

> **Note**
>
> 配額
> 建立指標篩選條件或訂閱篩選條件時，每個日誌群組最多有 5 個包含 regex 的篩選條件模式。在為指標篩選條件和訂閱篩選條件建立分隔或 JSON 篩選條件模式，或在篩選條件日誌事件或 Live Tail 時，每個篩選模式限制為 2 個 regex。

受支援運算子的使用

- ^：將比對錨定到字串的開頭。例如，%^[hc]at% 比對 "hat" 和 "cat"，但只比對字串的開頭。
- $：將比對錨定到字串的結尾。例如，%[hc]at$% 比對 "hat" 和 "cat"，但只比對字串的結尾。
- ?：符合前述詞彙的零個或多個執行個體。例如，%colou?r% 可同時比對 "color" 和 "colour"。
- []：定義字元類別。比對包含在括號內的字元清單或字元範圍。例如，%[abc]% 比對 "a"、"b" 或 "c"；%[a-z]% 比對從 "a" 到 "z" 的任何小寫字元；以及 %[abcx-z]% 比對 "a"、"b"、"c"、"x"、"y" 或 "z"。
- {m, n}：比對前一個詞彙至少 m 次且不超過 n 次。例如，%a{3,5}% 僅比對 "aaa"、"aaaa" 和 "aaaaa"。

  > **Note**
  >
  > 如果您選擇不定義下限或上限，則可省略 m 或 n。

- |：布林值 "Or"，比對垂直列任一側的詞彙。例如，%gra|ey% 可比對 "gray" 或 "grey"。

  > **Note**
  >
  > 術語是使用下列其中一個運算子的單一字元或重複字元類別：?、*、+ 或 {n,m}。

- \：逸出字元，可讓您使用運算子的文字含義，而非其特殊含義。例如，因為括號被逸出，%\[.\]% 比對任何用 "[" 和 "]" 括住的單一字元，例如 "[a]"、"[b]"、"[7]"、"[@]"、"[]]" 和 "[ ]"。

  > **Note**
  >
  > %10\.10\.0\.1% 是建立可比對 IP 地址 10.10.0.1 的 regex 的正確方法。

- *：符合前述詞彙的零個或多個執行個體。例如， %ab*c% 可比對 "ac"、"abc" 和 "abbbc"；%ab[0-9]*% 可比對 "ab"、"ab0" 和 "ab129"。
- +：符合前述詞彙的一或多個實例。例如，%ab+c% 可比對 "abc"、"abbc" 和 "abbbc"，而非 "ac"。
- .：比對任一個單一字元。例如，%.at% 比對以 "at" 結尾的任意三個字串，包括 "hat"、"cat"、"bat"、"4at"、"#at" 和 "at" (以空格開始)。

  > **Note**
  >
  > 建立比對 IP 地址的 regex 時，逸出 . 運算子很重要。例如，%10.10.0.1% 可比對 "10010,051"，這可能不是運算式的實際預期用途。

- \d、\D：比對數字/非數字字元。例如，%\d% 等於 %[0-9]%，%\D% 等於 %[^0-9]%。

  > **Note**
  >
  > 大寫運算子表示其對應小寫運算子的反轉。

- \s、\S：比對空白字元/非空白字元。

  > **Note**
  >
  > 大寫運算子表示其對應小寫運算子的反轉。空白字元包括 tab (\t)、空格 ( ) 和換行符 (\n) 字元。

- \w、\W：比對英數字元/非英數字元。例如，%\w% 等於 %[a-zA-Z_0-9]%，%\W% 等於 %[^a-zA-Z_0-9]%。

  > **Note**
  >
  > 大寫運算子表示其對應小寫運算子的反轉。

- \xhh：比對兩位數十六進位字元的 ASCII 映射。\x 為逸出序列，表示下列字元代表 ASCII 的十六進位值。hh 指定指向 ASCII 資料表中某個字元的兩個十六進位數字 (0-9 和 A-F)。

> **Note**
>
> 您可以使用 \xhh 來比對篩選條件模式不支援的符號字元。例如，%\x3A% 比對：；%\x28% 比對 (。

使用篩選條件模式來比對詞彙與規則運算式 (regex)

使用 regex 比對詞彙

您可以使用以 % (regex 模式之前和之後的百分比符號) 括住的 regex 模式來比對日誌事件中的術語。以下程式碼片段為篩選條件模式的範例，其會傳回包含 AUTHORIZED 關鍵字的所有日誌事件。

如需支援的規則運算式清單，請參閱支援的規則運算式。

```
%AUTHORIZED%
```

此篩選條件模式會傳回如下日誌事件訊息：

- `[ERROR 401] UNAUTHORIZED REQUEST`
- `[SUCCESS 200] AUTHORIZED REQUEST`

使用篩選條件模式來比對日誌事件中的詞彙

在非結構化日誌事件中比對詞彙

下列程式碼片段範例示範了如何使用篩選條件模式，來比對非結構化日誌事件中的詞彙。

> **Note**
>
> 篩選條件模式區分大小寫。將完全相符字詞和包含非英數字元的詞彙括在雙引號 ("") 中。

Example: Match a single term

下方程式碼片段為單一詞彙篩選條件模式的範例，其會傳回所有訊息中包含單字 ERROR (錯誤) 的日誌事件。

```
ERROR
```

此篩選條件模式會比對以下日誌事件訊息：

- `[ERROR 400] BAD REQUEST`
- `[ERROR 401] UNAUTHORIZED REQUEST`
- `[ERROR 419] MISSING ARGUMENTS`
- `[ERROR 420] INVALID ARGUMENTS`

Example: Match multiple terms

以下程式碼片段為多項詞彙篩選條件範例，其會傳回所有訊息中包含單字 ERROR (錯誤) 和 ARGUMENTS (引數) 的日誌事件。

```
ERROR ARGUMENTS
```

篩選條件會傳回如下的日誌事件訊息：

- `[ERROR 419] MISSING ARGUMENTS`
- `[ERROR 420] INVALID ARGUMENTS`

此篩選條件模式不會傳回以下日誌事件訊息，因為其不包含在篩選條件模式中指定的兩個詞彙。

- `[ERROR 400] BAD REQUEST`
- `[ERROR 401] UNAUTHORIZED REQUEST`

Example: Match optional terms

您可以使用模式比對，來建立會傳回包含選用詞彙之日誌事件的篩選條件模式。將問號 ("?") 置於想要比對的詞彙之前。下列程式碼片段為篩選條件模式的範例，篩選條件模式會傳回所有訊息中包含單字 ERROR 或單字 ARGUMENTS 的日誌事件。

```
?ERROR ?ARGUMENTS
```

此篩選條件模式會比對以下日誌事件訊息：

- `[ERROR 400] BAD REQUEST`
- `[ERROR 401] UNAUTHORIZED REQUEST`
- `[ERROR 419] MISSING ARGUMENTS`
- `[ERROR 420] INVALID ARGUMENTS`

> **Note**
>
> 無法結合使用問號 ("?") 和其他篩選模式，例如包含和排除詞彙。如果結合使用 "?" 和其他篩選模式，則將會忽略問號 ("?")。
> 例如，下列篩選模式會比對包含 REQUEST 字詞的所有事件，但問號 ("?") 篩選條件會被忽略且無效。
>
> ```
> ?ERROR ?ARGUMENTS REQUEST
> ```
>
> 日誌事件相符
>
> - `[INFO] REQUEST FAILED`
> - `[WARN] UNAUTHORIZED REQUEST`
> - `[ERROR] 400 BAD REQUEST`

Example: Match exact phrases

下列程式碼片段為篩選條件的範例，其會傳回訊息中包含完全相符字詞 INTERNAL SERVER ERROR (內部伺服器錯誤) 的日誌事件。

```
"INTERNAL SERVER ERROR"
```

此篩選條件模式會傳回以下日誌事件訊息：

- `[ERROR 500] INTERNAL SERVER ERROR`

Example: Include and exclude terms

您可以建立篩選條件模式，令其傳回訊息中包含某些詞彙，並排除其他詞彙的日誌事件。將減號 ("-") 置於想要排除的詞彙之前。下列程式碼片段為篩選條件模式的範例，其會傳回訊息中包含詞彙 ERROR (錯誤) 並排除詞彙 ARGUMENTS (引數) 的日誌事件。

```
ERROR -ARGUMENTS
```

此篩選條件模式會傳回如下日誌事件訊息：

- `[ERROR 400] BAD REQUEST`
- `[ERROR 401] UNAUTHORIZED REQUEST`

此篩選條件模式不會傳回以下日誌事件訊息，因為其包含單字 ARGUMENTS。

- `[ERROR 419] MISSING ARGUMENTS`
- `[ERROR 420] INVALID ARGUMENTS`

Example: Match everything

您可以用雙引號比對日誌事件中的所有內容。下列程式碼片段為篩選條件模式的範例，其會傳回所有日誌事件。

```
" "
```

使用篩選條件模式來比對 JSON 日誌事件中的詞彙

編寫 JSON 日誌事件的篩選條件模式

下列範例示範了如何寫入篩選條件模式的語法，來比對包含字串和數值和 JSON 詞彙。

Writing filter patterns that match strings

您可以建立篩選條件模式以比對 JSON 日誌事件中的字串。下列程式碼片段顯示的範例為以字串為基礎的篩選條件模式。

```
{ PropertySelector EqualityOperator String }
```

用大括號 ("{}") 括住篩選條件模式。以字串為基礎的篩選條件模式必須包含以下部分：

- Property selector (屬性選擇器)

  屬性選擇器以後面帶一個句點的貨幣符號 ("$.") 開始。屬性選擇器都是英數字元字串，並支援連字號 ("-") 和底線 ("_") 字元。字串不支援科學符號。屬性選擇器指向 JSON 日誌事件中的值節點。值節點可以是字串或數字。將陣列放在屬性選擇器之後。陣列中的元素依循從零開始的編號系統，意即陣列中的第一個元素是元素 0，第二個元素是元素 1，依此類推。將元素括在中括號內 ("[]")。如果屬性選擇器指向陣列或物件，則篩選條件模式會與日誌格式不相符。如果 JSON 屬性包含句點 ("."")，則可以使用括號標記法來選取該屬性。

  > **Note**
  >
  > 萬用字元選取器
  > 您可以使用 JSON 萬用字元來選取任何陣列元素或任何 JSON 物件欄位。
  > 配額
  > 在屬性選取器中，您只能使用最多一個萬用字元選取器。

- Equality operator (等式運算子)

等式運算子以下列符號之一開始：等於 ("=") 或不等於 ("!=")。等式運算子傳回布林值 (true 或 false)。

- 字串

  您可以用雙引號 ("") 括住字串。包含除了英數字元和底線符號以外類型的字串必須置於雙引號中。使用星號 ("*") 作為萬用字元來比對文本。

  > **Note**
  >
  > 建立篩選條件模式來比對 JSON 日誌事件中的詞彙時，您可以使用任何條件式規則運算式。如需支援的規則運算式清單，請參閱支援的規則運算式。

下列程式碼片段包含一個篩選條件模式範例，示範如何設定篩選條件模式的格式，使用字串來比對 JSON 詞彙。

```
{ $.eventType = "UpdateTrail" }
```

Writing filter patterns that match numeric values

您可以建立篩選條件模式以比對 JSON 日誌事件中的數值。下列程式碼片段為與數值相符之篩選條件模式的語法範例。

```
{ PropertySelector NumericOperator Number }
```

用大括號 ("{}") 括住篩選條件模式。比對數值的篩選條件模式必須包含以下部分：

- Property selector (屬性選擇器)

  屬性選擇器以後面帶一個句點的貨幣符號 ("$.") 開始。屬性選擇器都是英數字元字串，並支援連字號 ("-") 和底線 ("_") 字元。字串不支援科學符號。屬性選擇器指向 JSON 日誌事件中的值節點。值節點可以是字串或數字。將陣列放在屬性選擇器之後。陣列中的元素依循從零開始的編號系統，意即陣列中的第一個元素是元素 0，第二個元素是元素 1，依此類推。將元素括在中括號內 ("[]")。如果屬性選擇器指向陣列或物件，則篩選條件模式會與日誌格式不相符。如果 JSON 屬性包含句點 ("."")，則可以使用括號標記法來選取該屬性。

> **Note**
>
> 萬用字元選取器
> 您可以使用 JSON 萬用字元來選取任何陣列元素或任何 JSON 物件欄位。
> 配額
> 在屬性選取器中，您只能使用最多一個萬用字元選取器。

- Numeric operator (數值運算子)

  數值運算子以下列符號之一開始：大於 (">")、小於 ("<")、等於 ("=")、不等於 ("!=")、大於等於 (">=") 或是小於等於 ("<=")。

- 數字

  您可以使用包含加號 ("+") 或減號 ("-") 符號的整數，並遵守科學表示法。使用星號 ("*") 作為萬用字元來比對數字。

下列程式碼片段示範如何設定篩選條件模式的格式，使用數值來比對 JSON 詞彙。

```
// Filter pattern with greater than symbol
{ $.bandwidth > 75 }
// Filter pattern with less than symbol
{ $.latency < 50 }
// Filter pattern with greater than or equal to symbol
{ $.refreshRate >= 60 }
// Filter pattern with less than or equal to symbol
{ $.responseTime <= 5 }
// Filter pattern with equal sign
{ $.errorCode = 400}
// Filter pattern with not equal sign
{ $.errorCode != 500 }
// Filter pattern with scientific notation and plus symbol
{ $.number[0] = 1e-3 }
// Filter pattern with scientific notation and minus symbol
{ $.number[0] != 1e+3 }
```

使用簡單運算式來比對 JSON 日誌事件中的詞彙

下列範例中的程式碼片段，示範篩選條件模式如何比對 JSON 日誌事件中的詞彙。

> **Note**
>
> 如果要使用 JSON 日誌事件範例測試篩選條件模式範例，則必須在單行中輸入 JSON 日誌範例。

JSON 日誌事件

```
{
      "eventType": "UpdateTrail",
      "sourceIPAddress": "111.111.111.111",
      "arrayKey": [
            "value",
            "another value"
      ],
      "objectList": [
           {
             "name": "a",
             "id": 1
           },
           {
             "name": "b",
             "id": 2
           }
      ],
      "SomeObject": null,
      "cluster.name": "c"
}
```

Example: Filter pattern that matches string values

此篩選條件模式會比對屬性 "UpdateTrail" 中的字串 "eventType"。

```
{ $.eventType = "UpdateTrail" }
```

Example: Filter pattern that matches string values (IP address)

篩選條件模式包含一個萬用字元，並與屬性 "sourceIPAddress" 相符，因為其不包含帶有前綴的數字 "123.123."。

```
{ $.sourceIPAddress != 123.123.* }
```

Example: Filter pattern that matches a specific array element with a string value

此篩選條件模式會比對陣列 "value" 中的元素 "arrayKey"。

```
{ $.arrayKey[0] = "value" }
```

Example: Filter pattern that matches a string using regex

此篩選條件模式會比對屬性 "Trail" 中的字串 "eventType"。

```
{ $.eventType = %Trail% }
```

Example: Filter pattern that uses a wildcard to match values of any element in the array using regex

篩選條件模式包含可比對陣列 "arrayKey" 中元素 "value" 的 regex。

```
{ $.arrayKey[*] = %val.{2}% }
```

Example: Filter pattern that uses a wildcard to match values of any element with a specific prefix and subnet using regex (IP address)

此篩選條件模式包含可比對屬性 "sourceIPAddress" 中元素 "111.111.111.111" 的 regex。

```
{ $.* = %111\.111\.111\.1[0-9]{1,2}% }
```

> **Note**
>
> 配額

在屬性選取器中，您只能使用最多一個萬用字元選取器。

Example: Filter pattern that matches a JSON property with a period (.) in the key

```
{ $.['cluster.name'] = "c" }
```

Example: Filter pattern that matches JSON logs using IS

您可以使用 IS 變數建立比對 JSON 日誌中欄位的篩選條件模式。IS 變數可以比對包含 NULL、TRUE 或 FALSE 值的欄位。下列篩選條件模式會傳回 SomeObject 值為 NULL 的 JSON 日誌。

```
{ $.SomeObject IS NULL }
```

Example: Filter pattern that matches JSON logs using NOT EXISTS

您可以使用NOT EXISTS變數建立篩選器模式，以傳回記錄資料中不包含特定欄位的 JSON 記錄。下列篩選條件模式會使用 NOT EXISTS 傳回不包含欄位 SomeOtherObject 的 JSON 日誌。

```
{ $.SomeOtherObject NOT EXISTS }
```

> **Note**
>
> 目前不支援變數 IS NOT 和 EXISTS。

使用複合運算式來比對 JSON 物件中的詞彙

您可以在篩選條件模式中使用邏輯運算子邏輯與 ("&&") 和邏輯或 ("||") 來建立兩個或多個條件為真的，與日誌事件相符的複合表達式。複合表達式支援使用小括號 ("()") 和以下標準運算次序：() > && > ||。下列範例中的程式碼片段，示範如何將篩選條件模式與複合表達式相結合，來比對 JSON 物件中的詞彙。

JSON 物件

```
{
    "user": {
        "id": 1,
        "email": "John.Stiles@example.com"
    },
    "users": [
        {
         "id": 2,
         "email": "John.Doe@example.com"
        },
        {
         "id": 3,
         "email": "Jane.Doe@example.com"
        }
    ],
    "actions": [
        "GET",
        "PUT",
        "DELETE"
    ],
    "coordinates": [
        [0, 1, 2],
        [4, 5, 6],
        [7, 8, 9]
    ]
}
```

Example: Expression that matches using AND (&&)

此篩選條件模式包含的複合表達式，將 "user" 中的 "id" 與數值 1，和 "users" 陣列中第一個元素的 "email" 與字串 "John.Doe@example.com" 做比對。

```
{ ($.user.id = 1) && ($.users[0].email = "John.Doe@example.com") }
```

Example: Expression that matches using OR (||)

此篩選條件模式包含的複合表達式，將 "user" 中的 "email" 與字串 "John.Stiles@example.com" 相比對。

```
{ $.user.email = "John.Stiles@example.com" || $.coordinates[0][1] = "nonmatch" &&
 $.actions[2] = "nonmatch" }
```

Example: Expression that doesn't match using AND (&&)

此篩選條件模式包含一個找不到相符項目的複合表達式，因為該表達式與 "actions" 中的第三個動作不相符。

```
{ ($.user.email = "John.Stiles@example.com" || $.coordinates[0][1] = "nonmatch") &&
 $.actions[2] = "nonmatch" }
```

> **Note**
>
> 配額
> 在屬性選取器中，您只能使用最多一個萬用字元選取器，在具有複合運算式的篩選條件模式中，只能使用最多三個萬用字元選取器。

Example: Expression that doesn't match using OR (||)

此篩選條件模式包含一個找不到相符項目的複合表達式，因為該表達式與 "users" 中的第一個屬性，或 "actions" 中的第三個動作不符。

```
{ ($.user.id = 2 && $.users[0].email = "nonmatch") || $.actions[2] = "GET" }
```

使用篩選條件模式來比對以空格分隔的日誌事件中的詞彙

編寫以空格分隔的日誌事件的篩選條件模式

您可以建立篩選條件模式來比對以空格分隔的日誌事件中的詞彙。下面提供了以空格分隔的日誌事件範例，示範了如何編寫用於比對空格分隔日誌事件中詞彙的篩選條件模式語法。

> **Note**
>
> 建立篩選條件模式來比對以空格分隔的日誌事件中的詞彙時，您可以使用任何條件式規則運算式。如需支援的規則運算式清單，請參閱支援的規則運算式。

Example: Space-delimited log event

以下程式碼片段為以空格分隔的日誌事件，其中包含七個欄位：ip、user、username、timestamp、request、status_code，和 bytes。

```
127.0.0.1 Prod frank [10/Oct/2000:13:25:15 -0700] "GET /index.html HTTP/1.0" 404
 1534
```

> **Note**
>
> 中括號 ("[]") 和雙引號 ("") 之間的字元被視為單一欄位。

Writing filter patterns that match terms in a space-delimited log event

若要建立用來比對以空格分隔的日誌事件中詞彙的篩選條件模式，須將篩選條件模式括在中括號 ("[]") 中，並指定逗點 (",") 分隔名稱的欄位。下列篩選條件模式會分析七個欄位。

```
[ip=%127\.0\.0\.[1-9]%, user, username, timestamp, request =*.html*, status_code =
 4*, bytes]
```

您可以使用數字運算子 (>、<、=、!=、>= 或 <=) 和星號 (*) 作為萬用字元或 regex，來建立篩選條件模式。在範例篩選條件模式中，ip 使用可比對 IP 地址範圍 127.0.0.1 - 127.0.0.9 的 regex，其中 request 包含萬用字元，說明必須擷取包含 .html 的值，status_code 包含萬用字元，說明必須擷取以 4 開頭的值。

如果您不知道在空格分隔的日誌事件中剖析的欄位數量，則可以使用刪節號 (...) 來引用任何未命名的欄位。刪節號可以用來引用所需數量的欄位。下列範例為帶刪節號的篩選條件模式，代表先前範例篩選條件模式中的前四個未命名欄位。

```
[..., request =*.html*, status_code = 4*, bytes]
```

您也可以使用邏輯運算子，邏輯與 (&&) 和邏輯或 (||) 來建立複合表達式。下列篩選條件模式包含一個複合表達式，該表達式表明 status_code 的值必須是 404 或是 410。

```
[ip, user, username, timestamp, request =*.html*, status_code = 404 || status_code =
 410, bytes]
```

使用模式比對來比對以空格分隔的日誌事件中的詞彙

您可以使用模式比對，來建立按照特定順序比對詞彙的以空格分隔的篩選條件模式。使用指示器指定詞彙的順序。使用 w1 來表示第一個詞彙，並使用 w2 等來表示後續詞彙的順序。在詞彙之間插入逗號(",")。下列範例中的程式碼片段示範了如何使用模式，來比對以空格分隔的篩選條件模式。

> **Note**
>
> 建立篩選條件模式來比對以空格分隔的日誌事件中的詞彙時，您可以使用任何條件式規則運算式。如需支援的規則運算式清單，請參閱支援的規則運算式。

以空格分隔的日誌事件

```
INFO 09/25/2014 12:00:00 GET /service/resource/67 1200
INFO 09/25/2014 12:00:01 POST /service/resource/67/part/111 1310
```

```
WARNING 09/25/2014 12:00:02 Invalid user request
ERROR 09/25/2014 12:00:02 Failed to process request
```

Example: Match terms in order

下列以空格分隔的篩選條件模式會傳回第一個單字為 ERROR (錯誤) 的日誌事件。

```
[w1=ERROR, w2]
```

> **Note**
>
> 建立使用模式比對的以空格分隔的篩選條件模式時，必須在指定詞彙的順序後加上一個空白指標。例如：如果您建立了一個篩選條件模式以傳回第一個單字為 ERROR 的日誌事件，請在 w1 詞彙後加上一個空白 w2 指標。

Example: Match terms with AND (&&) and OR (||)

您可以使用邏輯運算子邏輯與 ("&&") 和邏輯或 ("||") 建立包含條件的以空格分隔的篩選條件模式。下列篩選條件模式會傳回第一個單字為 ERROR (錯誤) 或 WARNING (警告) 的日誌事件。

```
[w1=ERROR || w1=WARNING, w2]
```

Example: Exclude terms from matches

您可以建立以空格分隔的篩選條件模式，以傳回排除一個或多個詞彙的日誌事件。將不等於符號 ("!=") 放在想要排除的一個或多個詞彙之前。下列程式碼片段為篩選條件模式範例，其會傳回第一個單字不是 ERROR (錯誤) 和 WARNING (警告) 的日誌事件。

```
[w1!=ERROR && w1!=WARNING, w2]
```

Example: Match the top level item in a resource URI

下列程式碼片段為篩選條件模式的範例，其使用 regex 比對資源 URI 中的上層詞彙。

```
[logLevel, date, time, method, url=%/service/resource/[0-9]+$%, response_time]
```

Example: Match the child level item in a resource URI

下列程式碼片段為篩選條件模式的範例，其使用 regex 比對資源 URI 中的子層詞彙。

```
[logLevel, date, time, method, url=%/service/resource/[0-9]+/part/[0-9]+$%,
 response_time]
```

啟用從 AWS 服務記錄

雖然許多服務只會將日誌發佈到 CloudWatch 日誌，但有些 AWS 服務可以將日誌直接發佈到 Amazon 簡單儲存服務或 Amazon 資料 Firehose。如果您對日誌的主要需求是在其中一項服務中進行儲存或處理，則可以輕鬆地讓生成日誌的服務將日誌直接傳送到 Amazon S3 或 Firehose，而無需進行其他設置。

即使日誌直接發佈到 Amazon S3 或 Firehose，也需要支付費用。如需詳細資訊，請參閱 Amazon CloudWatch 定價中日誌索引標籤上的付費日誌。

某些 AWS 服務會使用通用基礎結構來傳送記錄檔。若要啟用從這些服務記錄日誌，您必須以具有特定許可的使用者身分登入。此外，您必須授與權限，才 AWS 能啟用要傳送的記錄。

對於需要這些許可的服務，需要兩種許可版本。在資料表中會將需要這些額外許可的服務標註為支援的 [V1 許可] 和 支援的 [V2 許可] 。如需有關這些必要許可的詳細資訊，請參閱資料表後面的章節。

日誌類型	CloudWatch Logs	Amazon S3	Firehose
Amazon API Gateway 存取日誌	支援的 [V1 許可]		
AWS AppSync logs	支援		
Amazon Aurora MySQL 日誌	支援		
Amazon Bedrock 知識庫記錄	支援的 [V2 許可]	支援的 [V2 許可]	支援的 [V2 許可]
Amazon Chime 媒體品質指標日誌和 SIP 訊息日誌	支援的 [V1 許可]		
CloudFront：訪問日誌		支援的 [V1 許可]	
AWS CloudHSM 稽核記錄	支援		
CloudWatch 顯然評估事件日誌	支援的 [V1 許可]	支援的 [V1 許可]	

日誌類型	CloudWatch Logs	Amazon S3	Firehose
CloudWatch 互聯網監控日誌		支援的 [V1 許可]	
CloudTrail 日誌	支援		
AWS CodeBuild logs	支援		
Amazon CodeWhisperer 事件記錄	支援的 [V2 許可]	支援的 [V2 許可]	支援的 [V2 許可]
Amazon Cognito logs	支援的 [V1 許可]		
Amazon Connect 日誌	支援		
AWS DataSync logs	支援		
Amazon ElastiCache 的 Redis 日誌	支援的 [V1 許可]		支援的 [V1 許可]
AWS Elastic Beanstalk logs	支援		
Amazon Elastic Container Service 日誌	支援		
Amazon Elastic Kubernetes Service 控制平面日誌	支援		
Amazon EventBridge 管道記錄	支援的 [V1 許可]	支援的 [V1 許可]	支援的 [V1 許可]
AWS Fargate logs	支援		
AWS Fault Injection Service 實驗日誌		支援的 [V1 許可]	
Amazon FinSpace	支援的 [V1 許可]	支援的 [V1 許可]	支援的 [V1 許可]

日誌類型	CloudWatch Logs	Amazon S3	Firehose
AWS Global Accelerator 流程記錄		支援的 [V1 許可]	
AWS Glue 工作記錄	支援		
IAM 身分識別中心錯誤記錄	支援的 [V2 許可]	支援的 [V2 許可]	支援的 [V2 許可]
Amazon Interactive Video Service 聊天日誌	支援的 [V1 許可]	支援的 [V1 許可]	支援的 [V1 許可]
AWS IoT logs	支援		
AWS IoT FleetWise logs	支援的 [V1 許可]	支援的 [V1 許可]	支援的 [V1 許可]
AWS Lambda logs	支援		
Amazon Macie 日誌	支援		
AWS Mainframe Modernization	支援的 [V1 許可]	支援的 [V1 許可]	支援的 [V1 許可]
Amazon Managed Service for Prometheus	支援的 [V1 許可]		
Amazon MSK 代理程式日誌	支援的 [V1 許可]	支援的 [V1 許可]	支援的 [V1 許可]
Amazon MSK Connect 日誌	支援的 [V1 許可]	支援的 [V1 許可]	支援的 [V1 許可]
Amazon MQ 一般和稽核日誌	支援		
AWS Network Firewall 記錄	支援的 [V1 許可]	支援的 [V1 許可]	支援的 [V1 許可]

日誌類型	CloudWatch Logs	Amazon S3	Firehose
Network Load Balancer 存取日誌		支援的 [V1 許可]	
OpenSearch 日誌	支援		
Amazon OpenSearch 服務擷取日誌	支援的 [V1 許可]	支援的 [V1 許可]	支援的 [V1 許可]
AWS OpsWorks logs	支援		
Amazon 關係數據庫 ServicePostgre SQL 日誌	支援		
AWS RoboMaker 日誌	支援		
Amazon Route 53 公有 DNS 查詢日誌	支援		
Amazon Route 53 Resolver 查詢日誌	支援的 [V1 許可]	支援的 [V1 許可]	
Amazon SageMaker 活動	支援的 [V1 許可]		
Amazon SageMaker 工人活動	支援的 [V1 許可]		
AWS 網站到網站 VPN 記錄檔	支援的 [V1 許可]	支援的 [V1 許可]	支援的 [V1 許可]
Amazon Simple Notification Service 日誌	支援		
Amazon Simple Notification Service 資料保護政策日誌	支援		
EC2 Spot 執行個體資料摘要檔案		支援的 [V1 許可]	

日誌類型	CloudWatch Logs	Amazon S3	Firehose
AWS Step Functions 快速工作流程和標準工作流程	支援的 [V1 許可]		
Storage Gateway 稽核日誌和運作狀態日誌	支援的 [V1 許可]		
AWS Transfer Family logs	支援的 [V1 許可]	支援的 [V1 許可]	支援的 [V1 許可]
AWS Verified Access logs	支援的 [V1 許可]	支援的 [V1 許可]	支援的 [V1 許可]
Amazon Virtual Private Cloud 流程日誌	支援	支援的 [V1 許可]	支援的 [V1 許可]
Amazon VPC Lattice 存取日誌	支援的 [V1 許可]	支援的 [V1 許可]	支援的 [V1 許可]
AWS WAF logs	支援的 [V1 許可]	支援的 [V1 許可]	支援
Amazon WorkMail 日誌	支援的 [V2 許可]	支援的 [V2 許可]	支援的 [V2 許可]

需要額外許可 [V1] 的日誌記錄

某些 AWS 服務使用通用基礎設施將日誌傳送到 CloudWatch 日誌、Amazon S3 或 Firehose。若要讓下表列出的 AWS 服務將日誌傳送到這些目的地，您必須以具有特定許可的使用者身分登入。

此外，必須授與權限才能傳送記錄。 AWS AWS 可以在設定記錄時自動建立這些權限，或者您可以在設定記錄之前先自行建立這些權限。對於跨帳戶傳遞，您必須自行手動建立權限原則。

如果您選擇在您或組織中的某人首次設定記錄檔傳送時 AWS 自動設定必要的權限和資源策略，則設定記錄檔傳送的使用者必須具有特定權限，如本節稍後所述。或者，您可以自行建立資源政策，所以設定傳送日誌的使用者就不需要這麼多許可。

下表摘要說明本節中的資訊適用於哪些日誌類型及哪些日誌目的地。

下列各節提供各個目的地的詳細資訊。

傳送至記錄 CloudWatch 檔的記錄

> **Important**
>
> 當您在下列清單中設定要傳送至記錄檔的 CloudWatch 記錄檔類型時，請視需要 AWS 建立或變更與接收記錄檔之記錄群組相關聯的資源策略。繼續閱讀本節以查看詳細資訊。

當上一節中表格中列出的記錄檔類型傳送至「 CloudWatch 記錄檔」時，本節適用：

使用者許可

若要設定第一次將這些類型的記錄檔傳送至 CloudWatch 記錄檔，您必須登入具有下列權限的帳戶。

- `logs:CreateLogDelivery`
- `logs:PutResourcePolicy`
- `logs:DescribeResourcePolicies`
- `logs:DescribeLogGroups`

> **Note**
>
> 當您指定`logs:DescribeLogGroups`、或`logs:PutResourcePolicy`權限時`logs:DescribeResourcePolicies`，請務必將其`Resource`行的 ARN 設定為使用`*`萬用字元，而不是僅指定單一記錄群組名稱。例如：`"Resource": "arn:aws:logs:us-east-1:111122223333:log-group:*"`

如果這些類型的記錄檔已經傳送至 CloudWatch 記錄檔中的記錄群組，則若要設定將另一種記錄檔傳送到相同的記錄群組，您只需要該`logs:CreateLogDelivery`權限即可。

日誌群組和資源政策

日誌送往的日誌群組必須具有包含特定許可的資源政策。如果記錄群組目前沒有資源原則，而且設定記錄的使用者具有記錄群組的`logs:PutResourcePolicy`、`logs:DescribeResourcePolicies`、和`logs:DescribeLogGroups`權限，則當您開始將記錄檔傳送至記 CloudWatch 錄檔時，會 AWS 自動為其建立下列原則。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSLogDeliveryWrite20150319",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "delivery.logs.amazonaws.com"
        ]
      },
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": [
        "arn:aws:logs:us-east-1:0123456789:log-group:my-log-group:log-stream:*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": ["0123456789"]
        },
        "ArnLike": {
          "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:*"]
        }
      }
    }
  ]
}
```

如果日誌群組有資源政策，但該政策未包含前一個政策中出現的陳述式，且設定記錄的使用者具有日誌群組的 `logs:PutResourcePolicy`、`logs:DescribeResourcePolicies` 及 `logs:DescribeLogGroups` 許可，則該陳述式會附加至日誌群組的資源政策。

日誌群組資源政策大小限制考量

這些服務必須在資源原則中列出要傳送記錄檔的每個記錄群組，而且 CloudWatch 記錄檔資源策略的長度限制為 5120 個字元。將記錄檔傳送至大量記錄群組的服務可能會遇到此限制。

為了減輕此問題， CloudWatch Logs 會監視傳送記錄檔的服務所使用的資源原則大小，以及偵測到原則達到 5120 個字元的大小限制時， CloudWatch 記錄檔會自動`/aws/vendedlogs/*`在該服務的資源

原則中啟用。然後，您就可以開始使用名稱開頭為 /aws/vendedlogs/ 的日誌群組，作為這些服務的日誌目的地。

傳送至 Amazon S3 的日誌

將日誌設定為傳送至 Amazon S3 時，必要時 AWS 建立或變更與接收日誌的 S3 儲存貯體相關聯的資源政策。

直接發佈至 Amazon S3 的日誌會發佈至您指定的現有儲存貯體。在指定的儲存貯體中，每五分鐘會建立一或多個日誌檔案。

當您第一次將日誌傳送到 Amazon S3 儲存貯體時，傳送日誌的服務會記錄儲存貯體的擁有者，以確保日誌僅傳送到屬於此帳戶的儲存貯體。因此，若要變更 Amazon S3 儲存貯體擁有者，您必須在原始服務中重新建立或更新日誌訂閱。

> **Note**
>
> CloudFront 使用與將付費日誌傳送到 S3 的其他服務不同的許可模型。如需詳細資訊，請參閱設定標準記錄和存取日誌檔案所需的許可。
> 此外，如果您使用相同的 S3 儲存貯體來 CloudFront 存取日誌和另一個記錄來源，啟用儲存貯體上的 ACL CloudFront 也會授與使用此儲存貯體的所有其他日誌來源的權限。

使用者許可

您必須以具有下列許可的帳戶登入，才能第一次設定將任何這些類型的日誌傳送到 Amazon S3。

- `logs:CreateLogDelivery`
- `S3:GetBucketPolicy`
- `S3:PutBucketPolicy`

如果任何這些類型的日誌已傳送到某個 Amazon S3 儲存貯體，則若要設定將另一種類型的日誌傳送到同一個儲存貯體，您只需要有 `logs:CreateLogDelivery` 許可。

S3 儲存貯體資源政策

日誌送往的 S3 儲存貯體必須具有包含特定許可的資源政策。如果儲存貯體目前沒有資源政策，且設定記錄的使用者具有儲存貯體的`S3:GetBucketPolicy`和`S3:PutBucketPolicy`許可，則當您開始將日誌傳送到 Amazon S3 時， AWS 自動為其建立下列政策。

```
{
    "Version": "2012-10-17",
    "Id": "AWSLogDeliveryWrite20150319",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryAclCheck",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
                },
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::my-bucket",
            "Condition": {
                "StringEquals": {
                "aws:SourceAccount": ["0123456789"]
                },
                "ArnLike": {
                "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:*"]
                }
            }
        },
        {
            "Sid": "AWSLogDeliveryWrite",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-bucket/AWSLogs/account-ID/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceAccount": ["0123456789"]
                },
                "ArnLike": {
                    "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:*"]
                }
            }
        }
    ]
}
```

在先前的政策中，對於 aws:SourceAccount，指定要將日誌交付至此儲存貯體的帳戶 IDS 清單。對於 aws:SourceArn，指定產生日誌之資源的 ARN 清單，格式為 arn:aws:logs:*source-region*:*source-account-id*:*。

如果儲存貯體具有資源政策，但該政策未包含前一個政策中出現的陳述式，且設定記錄的使用者具有儲存貯體的 S3:GetBucketPolicy 和 S3:PutBucketPolicy 許可，則該陳述式會附加至儲存貯體的資源政策。

> **Note**
>
> 在某些情況下， AWS CloudTrail 如果未授予s3:ListBucket權限，您可能會在中看到AccessDenied錯誤訊息delivery.logs.amazonaws.com。若要避免 CloudTrail 記錄檔中出現這些錯誤，您必須授與s3:ListBucket權限，delivery.logs.amazonaws.com且必須包含上述儲存貯體政策中所設定之s3:GetBucketAcl權限所顯示的Condition參數。為簡化此操作而不用建立一個新的 Statement，你可以直接將 AWSLogDeliveryAclCheck 更新為 “Action”: [“s3:GetBucketAcl”, “s3:ListBucket”]

Amazon S3 儲存貯體伺服器端加密

您可以使用 Amazon S3 受管金鑰 (SSE-S3) 啟用伺服器端加密，或使用存放在 (SSE-KMS) 的伺服器端加密來保護 Amazon S3 儲存貯體中的資 AWS Key Management Service 料。 AWS KMS 如需詳細資訊，請參閱使用伺服器端加密保護資料。

如果您選擇 SSE-S3，則不需要其他組態。Amazon S3 會處理加密金鑰。

> **Warning**
>
> 如果您選擇 SSE-KMS，則必須使用客戶受管金鑰，因為此案例不支援使用 AWS 受管金鑰。如果您使用 AWS 受管理金鑰設定加密，記錄檔將會以無法讀取的格式傳遞。

使用客戶受管 AWS KMS 金鑰時，您可以在啟用儲存貯體加密時指定客戶受管金鑰的 Amazon 資源名稱 (ARN)。您必須將以下內容新增至客戶受管金鑰的金鑰政策 (而不是 S3 儲存貯體的儲存貯體政策)，以便日誌傳遞帳戶可以寫入您的 S3 儲存貯體。

如果您選擇 SSE-KMS，則必須使用客戶受管金鑰，因為此案例不支援使用 AWS 受管金鑰。使用客戶受管 AWS KMS 金鑰時，您可以在啟用儲存貯體加密時指定客戶受管金鑰的 Amazon 資源名稱

(ARN)。您必須將以下內容新增至客戶受管金鑰的金鑰政策 (而不是 S3 儲存貯體的儲存貯體政策)，以便日誌傳遞帳戶可以寫入您的 S3 儲存貯體。

```
{
    "Sid": "Allow Logs Delivery to use the key",
    "Effect": "Allow",
    "Principal": {
        "Service": [ "delivery.logs.amazonaws.com" ]
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:SourceAccount": ["0123456789"]
        },
        "ArnLike": {
            "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:*"]
        }
        }
}
```

對於 aws:SourceAccount，指定要將日誌交付至此儲存貯體的帳戶 IDS 清單。對於 aws:SourceArn，指定產生日誌之資源的 ARN 清單，格式為 arn:aws:logs:*source-region*:*source-account-id*:*。

原木已傳送至 Firehose

本節適用於上一節表格中列出的記錄類型傳送至 Firehose 時：

使用者許可

若要設定第一次將這些類型的記錄傳送至 Firehose，您必須登入具有下列權限的帳戶。

- logs:CreateLogDelivery
- firehose:TagDeliveryStream
- iam:CreateServiceLinkedRole

如果這些類型的記錄檔已經傳送到 Firehose，則若要設定另一種類型的記錄檔傳送至 Firehose，您只需要擁有logs:CreateLogDelivery和firehose:TagDeliveryStream權限即可。

用於許可的 IAM 角色

由於 Firehose 不使用資源政策， AWS 因此在設定這些記錄檔以傳送至 Firehose 時，會使用 IAM 角色。 AWS 會建立名為AWSServiceRoleForLogDelivery的服務連結角色。此服務連結角色包含下列許可。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "firehose:PutRecord",
                "firehose:PutRecordBatch",
                "firehose:ListTagsForDeliveryStream"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/LogDeliveryEnabled": "true"
                }
            },
            "Effect": "Allow"
        }
    ]
}
```

此服務連結角色會授予LogDeliveryEnabled標籤設定為的所有 Firehose 傳遞串流的權限。true AWS 當您設定記錄時，將此標記提供給目的地傳遞串流。

此服務連結角色也有信任政策，以允許 delivery.logs.amazonaws.com 服務委託人擔任所需的服務連結角色。該信任政策如下：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "delivery.logs.amazonaws.com"
      },
```

```
      "Action": "sts:AssumeRole"
    }
  ]
}
```

需要額外許可 [V2] 的日誌記錄

有些 AWS 服務會使用新方法來傳送記錄檔。這是一種靈活的方法，可讓您設定從這些服務到下列一或多個目的地的 CloudWatch 日誌傳遞：日誌、Amazon S3 或 Firehose。

工作記錄傳送包含三個元素：

- ADeliverySource，這是一個邏輯對象，表示實際發送日誌的資源（S）。
- ADeliveryDestination，這是代表實際傳遞目的地的邏輯物件。
- ADelivery，將傳送來源連接至傳送目的地

若要設定支援的 AWS 服務與目的地之間的記錄傳遞，您必須執行下列動作：

- 使用建立傳送來源PutDeliverySource。
- 使用建立傳送目的地PutDeliveryDestination。
- 如果您要跨帳戶傳送記錄檔，則必須 PutDeliveryDestinationPolicy在目標帳戶中使用，將 IAM 策略指派給目的地。此原則授權從帳戶 A 中的傳遞來源建立傳遞至帳戶 B 中的傳遞目的地。對於跨帳戶傳遞，您必須自行手動建立權限原則。
- 使用將一個傳送來源和一個傳送目的地完全配對，以建立傳送 CreateDelivery。

以下各節提供您在登入後使用 V2 處理程序設定每種目的地的日誌傳遞所需之許可的詳細資訊。可將這些許可授予您登入時具有的 IAM 角色。

> **Important**
>
> 您有責任在刪除記錄產生資源之後移除記錄傳遞資源。為此，請按照下列步驟操作。
>
> 1. 使用Delivery作DeleteDelivery業刪除。
> 2. 使用DeliverySource作DeleteDeliverySource業刪除。
> 3. 如果DeliveryDestination與您剛剛刪除的DeliverySource相關聯僅用於此特定內容DeliverySource，則可以使用該DeleteDeliveryDestinations操作將其刪除。

內容

傳送至記錄 CloudWatch 檔的記錄

使用者許可

若要啟用將記錄檔傳送至 CloudWatch 記錄檔，您必須使用下列權限登入。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ReadWriteAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:GetDelivery",
                "logs:GetDeliverySource",
                "logs:PutDeliveryDestination",
                "logs:GetDeliveryDestinationPolicy",
                "logs:DeleteDeliverySource",
                "logs:PutDeliveryDestinationPolicy",
                "logs:CreateDelivery",
                "logs:GetDeliveryDestination",
                "logs:PutDeliverySource",
                "logs:DeleteDeliveryDestination",
                "logs:DeleteDeliveryDestinationPolicy",
                "logs:DeleteDelivery"
            ],
            "Resource": [
                "arn:aws:logs:region:account-id:delivery:*",
                "arn:aws:logs:region:account-id:delivery-source:*",
                "arn:aws:logs:region:account-id:delivery-destination:*"
            ]
        },
```

```
        {
            "Sid": "ListAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeDeliveryDestinations",
                "logs:DescribeDeliverySources",
                "logs:DescribeDeliveries"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowUpdatesToResourcePolicyCWL",
            "Effect": "Allow",
            "Action": [
                "logs:PutResourcePolicy",
                "logs:DescribeResourcePolicies",
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "arn:aws:logs:region:account-id:*"
            ]
        }
    ]
}
```

日誌群組和資源政策

日誌送往的日誌群組必須具有包含特定許可的資源政策。如果記錄群組目前沒有資源原則，而且設定記錄的使用者具有記錄群組的`logs:PutResourcePolicy`、`logs:DescribeResourcePolicies`和`logs:DescribeLogGroups`權限，則當您開始將記錄檔傳送至記 CloudWatch 錄檔時，會 AWS 自動為其建立下列原則。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSLogDeliveryWrite20150319",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "delivery.logs.amazonaws.com"
        ]
      },
```

```
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": [
        "arn:aws:logs:us-east-1:0123456789:log-group:my-log-group:log-stream:*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": ["0123456789"]
        },
        "ArnLike": {
          "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:*"]
        }
      }
    }
  ]
}
```

日誌群組資源政策大小限制考量

這些服務必須在資源原則中列出要傳送記錄檔的每個記錄群組，而且 CloudWatch 記錄檔資源策略的長度限制為 5120 個字元。將日誌傳送至大量日誌群組的服務可能會受到此限制。

為了減輕此問題， CloudWatch Logs 會監視傳送記錄檔的服務所使用的資源原則大小，以及偵測到原則達到 5120 個字元的大小限制時， CloudWatch 記錄檔會自動/aws/vendedlogs/*在該服務的資源原則中啟用。然後，您就可以開始使用名稱開頭為 /aws/vendedlogs/ 的日誌群組，作為這些服務的日誌目的地。

傳送至 Amazon S3 的日誌

使用者許可

若要啟用傳送日誌至 Amazon S3，您登入時必須具有以下許可。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ReadWriteAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:GetDelivery",
```

```
                "logs:GetDeliverySource",
                "logs:PutDeliveryDestination",
                "logs:GetDeliveryDestinationPolicy",
                "logs:DeleteDeliverySource",
                "logs:PutDeliveryDestinationPolicy",
                "logs:CreateDelivery",
                "logs:GetDeliveryDestination",
                "logs:PutDeliverySource",
                "logs:DeleteDeliveryDestination",
                "logs:DeleteDeliveryDestinationPolicy",
                "logs:DeleteDelivery"
            ],
            "Resource": [
                "arn:aws:logs:region:account-id:delivery:*",
                "arn:aws:logs:region:account-id:delivery-source:*",
                "arn:aws:logs:region:account-id:delivery-destination:*"
            ]
        },
        {
            "Sid": "ListAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeDeliveryDestinations",
                "logs:DescribeDeliverySources",
                "logs:DescribeDeliveries"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowUpdatesToResourcePolicyS3",
            "Effect": "Allow",
            "Action": [
                "s3:PutBucketPolicy",
                "s3:GetBucketPolicy"
            ],
            "Resource": "arn:aws:s3:::bucket-name"
        }
    ]
}
```

日誌送往的 S3 儲存貯體必須具有包含特定許可的資源政策。如果儲存貯體目前沒有資源政策，且設定記錄的使用者具有儲存貯體的`S3:GetBucketPolicy`和`S3:PutBucketPolicy`許可，則當您開始將日誌傳送到 Amazon S3 時， AWS 自動為其建立下列政策。

```
{
    "Version": "2012-10-17",
    "Id": "AWSLogDeliveryWrite20150319",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryAclCheck",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
                },
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::my-bucket",
            "Condition": {
                "StringEquals": {
                "aws:SourceAccount": ["0123456789"]
                },
                "ArnLike": {
                "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:delivery-source*"]
                }
            }
        },
        {
            "Sid": "AWSLogDeliveryWrite",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-bucket/AWSLogs/account-ID/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceAccount": ["0123456789"]
                },
                "ArnLike": {
                    "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:delivery-
source:*"]
                }
            }
        }
    ]
}
```

在先前的政策中，對於 aws:SourceAccount，指定要將日誌交付至此儲存貯體的帳戶 IDS 清單。對於 aws:SourceArn，指定產生日誌之資源的 ARN 清單，格式為 arn:aws:logs:*source-region*:*source-account-id*:*。

如果儲存貯體具有資源政策，但該政策未包含前一個政策中出現的陳述式，且設定記錄的使用者具有儲存貯體的 S3:GetBucketPolicy 和 S3:PutBucketPolicy 許可，則該陳述式會附加至儲存貯體的資源政策。

> **Note**
>
> 在某些情況下， AWS CloudTrail 如果未授予s3:ListBucket權限，您可能會在中看到AccessDenied錯誤訊息delivery.logs.amazonaws.com。若要避免 CloudTrail 記錄檔中出現這些錯誤，您必須授與s3:ListBucket權限，delivery.logs.amazonaws.com且必須包含上述儲存貯體政策中所設定之s3:GetBucketAcl權限所顯示的Condition參數。為簡化此操作而不用建立一個新的 Statement，你可以直接將 AWSLogDeliveryAclCheck 更新為 “Action”: [“s3:GetBucketAcl”, “s3:ListBucket”]

Amazon S3 儲存貯體伺服器端加密

您可以使用 Amazon S3 受管金鑰 (SSE-S3) 啟用伺服器端加密，或使用存放在 (SSE-KMS) 的伺服器端加密來保護 Amazon S3 儲存貯體中的資 AWS Key Management Service 料。 AWS KMS 如需詳細資訊，請參閱使用伺服器端加密保護資料。

如果您選擇 SSE-S3，則不需要其他組態。Amazon S3 會處理加密金鑰。

> **Warning**
>
> 如果您選擇 SSE-KMS，則必須使用客戶受管金鑰，因為此案例不支援使用 AWS 受管金鑰。如果您使用 AWS 受管理金鑰設定加密，記錄檔將會以無法讀取的格式傳遞。

使用客戶受管 AWS KMS 金鑰時，您可以在啟用儲存貯體加密時指定客戶受管金鑰的 Amazon 資源名稱 (ARN)。您必須將以下內容新增至客戶受管金鑰的金鑰政策 (而不是 S3 儲存貯體的儲存貯體政策)，以便日誌傳遞帳戶可以寫入您的 S3 儲存貯體。

如果您選擇 SSE-KMS，則必須使用客戶受管金鑰，因為此案例不支援使用 AWS 受管金鑰。使用客戶受管 AWS KMS 金鑰時，您可以在啟用儲存貯體加密時指定客戶受管金鑰的 Amazon 資源名稱

(ARN)。您必須將以下內容新增至客戶受管金鑰的金鑰政策 (而不是 S3 儲存貯體的儲存貯體政策)，以便日誌傳遞帳戶可以寫入您的 S3 儲存貯體。

```
{
    "Sid": "Allow Logs Delivery to use the key",
    "Effect": "Allow",
    "Principal": {
        "Service": [ "delivery.logs.amazonaws.com" ]
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:SourceAccount": ["0123456789"]
        },
        "ArnLike": {
            "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:delivery-source:*"]
        }
        }
}
```

對於 aws:SourceAccount，指定要將日誌交付至此儲存貯體的帳戶 IDS 清單。對於 aws:SourceArn，指定產生日誌之資源的 ARN 清單，格式為 arn:aws:logs:*source-region*:*source-account-id*:*。

原木已傳送至 Firehose

使用者許可

若要啟用將記錄檔傳送至 Firehose，您必須使用下列權限登入。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ReadWriteAccessForLogDeliveryActions",
            "Effect": "Allow",
```

```
            "Action": [
                "logs:GetDelivery",
                "logs:GetDeliverySource",
                "logs:PutDeliveryDestination",
                "logs:GetDeliveryDestinationPolicy",
                "logs:DeleteDeliverySource",
                "logs:PutDeliveryDestinationPolicy",
                "logs:CreateDelivery",
                "logs:GetDeliveryDestination",
                "logs:PutDeliverySource",
                "logs:DeleteDeliveryDestination",
                "logs:DeleteDeliveryDestinationPolicy",
                "logs:DeleteDelivery"
            ],
            "Resource": [
                "arn:aws:logs:region:account-id:delivery:*",
                "arn:aws:logs:region:account-id:delivery-source:*",
                "arn:aws:logs:region:account-id:delivery-destination:*"
            ]
        },
        {
            "Sid": "ListAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeDeliveryDestinations",
                "logs:DescribeDeliverySources",
                "logs:DescribeDeliveries"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowUpdatesToResourcePolicyFH",
            "Effect": "Allow",
            "Action": [
                "firehose:TagDeliveryStream"
            ],
            "Resource": [
                "arn:aws:firehose:region:account-id:deliverystream/*"
            ]
        },
        {
            "Sid": "CreateServiceLinkedRole",
            "Effect": "Allow",
            "Action": [
```

```
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "arn:aws:iam::account-id:role/aws-service-role/
delivery.logs.amazonaws.com/AWSServiceRoleForLogDelivery"
        }
    ]
}
```

用於資源許可的 IAM 角色

由於 Firehose 不使用資源政策， AWS 因此在設定這些記錄檔以傳送至 Firehose 時，會使用 IAM 角色。 AWS 會建立名為AWSServiceRoleForLogDelivery的服務連結角色。此服務連結角色包含下列許可。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "firehose:PutRecord",
                "firehose:PutRecordBatch",
                "firehose:ListTagsForDeliveryStream"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/LogDeliveryEnabled": "true"
                }
            },
            "Effect": "Allow"
        }
    ]
}
```

此服務連結角色會授予LogDeliveryEnabled標籤設定為的所有 Firehose 傳遞串流的權限。`true` AWS 當您設定記錄時，將此標記提供給目的地傳遞串流。

此服務連結角色也有信任政策，以允許 `delivery.logs.amazonaws.com` 服務委託人擔任所需的服務連結角色。該信任政策如下：

```
{
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "delivery.logs.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

服務特定權限

除了前幾節中列出的特定目的地權限之外，某些服務還需要明確授權，允許客戶從其資源傳送記錄檔，作為額外的安全層。它會針對在該服務AllowVendedLogDeliveryForResource中出現記錄的資源授權動作。對於這些服務，請使用下列政策，並以適當的值取代*##*和*####*。如需這些欄位的服務特定值，請參閱這些服務的說明文件頁面以取得付費記錄檔。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ServiceLevelAccessForLogDelivery",
            "Effect": "Allow",
            "Action": [
                "service:AllowVendedLogDeliveryForResource"
            ],
            "Resource": "arn:aws:service:region:account-id:resource-type/*"
        }
    ]
}
```

主機特定權限

除了前幾節所列的權限之外，如果您要使用主控台而非 API 來設定記錄傳遞，您還需要下列其他權限：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Sid": "AllowLogDeliveryActionsConsoleCWL",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "arn:aws:logs:us-east-1:111122223333:log-group:*"
            ]
        },
        {
            "Sid": "AllowLogDeliveryActionsConsoleS3",
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::*"
            ]
        },
        {
            "Sid": "AllowLogDeliveryActionsConsoleFH",
            "Effect": "Allow",
            "Action": [
                "firehose:ListDeliveryStreams",
                "firehose:DescribeDeliveryStream"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

預防跨服務混淆代理人

混淆代理人問題屬於安全性問題，其中沒有執行動作許可的實體可以強制具有更多許可的實體執行該動作。在中 AWS，跨服務模擬可能會導致混淆的副問題。在某個服務 (呼叫服務) 呼叫另一個服務 (被呼叫服務) 時，可能會發生跨服務模擬。可以操縱呼叫服務來使用其許可，以其不應有存取許可的方式對其他客戶的資源採取動作。為了預防這種情況， AWS 提供的工具可協助您保護所有服務的資料，而這些服務主體已獲得您帳戶中資源的存取權。

我們建議在資源策略中使用aws:SourceArnaws:SourceAccountaws:SourceOrgID、、和aws:SourceOrgPaths全域條件內容索引鍵，以限制 CloudWatch Logs 將其他服務提供給資源的權限。用於僅aws:SourceArn將一個資源與跨服務存取相關聯。用於aws:SourceAccount讓該帳號中的任何資源與跨服務使用相關聯。用於aws:SourceOrgID允許組織內任何帳號的任何資源與跨服務使用相關聯。用於aws:SourceOrgPaths將 AWS Organizations 路徑中帳號的任何資源與跨服務使用相關聯。如需有關使用和瞭解路徑的詳細資訊，請參閱瞭解 AWS Organizations 實體路徑。

防範混淆代理人問題的最有效方法是使用 aws:SourceArn 全域條件內容索引鍵，以及資源的完整 ARN。如果不知道資源的完整 ARN，或者如果您指定了多個資源，請使用 aws:SourceArn 全域內容條件索引鍵搭配萬用字元 (*) 來表示 ARN 的未知部分。例如 arn:aws:*servicename*:*:*123456789012*:*。

如果 aws:SourceArn 值不包含帳戶 ID (例如 Amazon S3 儲存貯體 ARN)，您必須同時使用 aws:SourceAccount 和 aws:SourceArn 來限制許可。

若要大規模防範混淆代理人問題，請在資源型政策中使用 aws:SourceOrgID 或 aws:SourceOrgPaths 全域條件內容鍵和資源的組織 ID 或組織路徑。當您新增、移除或移動組織中的帳戶時，包含 aws:SourceOrgID 或 aws:SourceOrgPaths 鍵的政策將會自動包含正確的帳戶，您無需手動更新政策。

本頁先前章節中的政策說明如何使用 aws:SourceArn 和 aws:SourceAccount 全域條件內容金鑰來預防混淆代理人問題。

CloudWatch 記錄 AWS 受管策略的更新

檢視有關 CloudWatch 記錄檔 AWS 受管理策略更新的詳細資料，因為此服務開始追蹤這些變更。如需有關此頁面變更的自動警示，請訂閱 CloudWatch 記錄文件歷史記錄頁面上的 RSS 摘要。

變更	描述	日期
AWSServiceRoleForLogDelivery 服務連結角色策略 — 現有策略的更新	CloudWatch 記錄檔變更了與AWSServiceRoleForLogDelivery服務連結角色相關聯的 IAM 政策中的許可。變更如下： • firehose:ResourceTag/LogDeliveryEnab	2021 年 7 月 15 日

變更	描述	日期
	led": "true" 條件金鑰已變更為 aws:ResourceTag/LogDeliveryEnabled": "true" 。	
CloudWatch 記錄檔開始追蹤變更	CloudWatch 記錄檔開始追蹤其 AWS 受管理策略的變更。	2021 年 6 月 10 日

將日誌資料匯出至 Amazon S3

將日誌資料從日誌群組匯出至 Amazon S3 儲存貯體，並將此資料用於自訂處理和分析，或載入至其他系統。您可以匯出到相同帳戶或其他帳戶中的儲存貯體。

您可以執行下列作業：

- 將日誌資料匯出到由 SSE-KMS 在 AWS Key Management Service () 中加密的 S3 儲存貯體AWS KMS
- 將日誌資料匯出至啟用 S3 Object Lock 且具有保留期的 S3 儲存貯體。

> **Note**
>
> 只有標準日誌類別中的日誌群組才支援匯出到 Amazon S3。如需記錄類別的詳細資訊，請參閱日誌類。

若要開始匯出程序，您必須建立一個 S3 儲存貯體來存放匯出的日誌資料。您可以將匯出的檔案存放在 S3 儲存貯體，並定義 Amazon S3 生命週期規則以自動封存或刪除匯出的檔案。

您可以匯出至使用 AES-256 或 SSE-KMS 加密的 S3 儲存貯體。不支援匯出至使用 DSSE-KMS 加密的儲存貯體。

您可以從多個日誌群組或多個時間範圍，將日誌匯出至相同的 S3 儲存貯體。若要為各個匯出任務區隔日誌資料，您可以指定字首做為所有匯出物件的 Amazon S3 金鑰字首 (key prefix)。

> **Note**
>
> 無法保證將匯出檔案內的日誌資料塊依時間排序。您可以使用 Linux 公用程式對匯出的日誌欄位資料進行排序。例如，以下公用程式命令會對單一資料夾內所有 .gz 檔案中的事件進行排序。
>
> ```
> find . -exec zcat {} + | sed -r 's/^[0-9]+/\x0&/' | sort -z
> ```
>
> 以下公用程式命令對多個子資料夾內的 .gz 檔案進行排序。
>
> ```
> find ./*/ -type f -exec zcat {} + | sed -r 's/^[0-9]+/\x0&/' | sort -z
> ```

此外，您可以使用另一個 stdout 命令將排序後的輸出傳送到另一個檔案儲存。

日誌資料最長需要 12 個小時才能匯出。匯出任務會在 24 小時後逾時。如果匯出任務逾時，請縮短建立匯出任務時的時間範圍。

如需接近即時的日誌資料分析，請改為參閱使用日誌見解分析 CloudWatch 日誌資料或使用訂閱即時處理日誌資料。

目錄

概念

您開始之前，請熟悉以下匯出概念：

日誌群組名稱

與匯出任務關聯的日誌群組名稱。此日誌群組中的日誌資料將匯出到指定的 S3 儲存貯體。

從 (時間戳記)

所需的時間戳記以從 1970 年 1 月 1 日 00:00:00 UTC 開始的毫秒數表示。將匯出在此時間或之後擷取之記錄群組中的所有記錄事件。

至 (時間戳記)

所需的時間戳記以從 1970 年 1 月 1 日 00:00:00 UTC 開始的毫秒數表示。此日誌群組中在此時間之前擷取的所有日誌事件都會匯出。

目的地儲存貯體

與匯出任務關聯的 S3 儲存貯體名稱。此儲存貯體用於從指定的日誌群組匯出日誌資料。

目的地前綴

選用的屬性，作為所有匯出物件的 Amazon S3 金鑰前綴。這有助於在您的儲存貯體建立類似資料夾的整理方式。

使用主控台將日誌資料匯出至 Amazon S3

在下列範例中，您可以使用 Amazon CloudWatch 主控台將所有資料從名為的 Amazon S3 CloudWatch 儲存貯體匯出`my-log-group`到名為的 Amazon S3 儲存貯體`my-exported-logs`。

支援將日誌資料匯出至由 SSE-KMS 加密的 S3 儲存貯體。不支援匯出至使用 DSSE-KMS 加密的儲存貯體。

如何設定匯出作業的詳細方法，取決於您要存放匯出資料的 Amazon S3 儲存貯體是否與要匯出的日誌位於同一帳戶。

主題

同帳戶匯出

如果 Amazon S3 儲存貯體與要匯出的日誌位於同一帳戶，請參閱本區段的說明。

主題

步驟 1：建立 Amazon S3 儲存貯體

我們建議您使用專門為 CloudWatch Logs 建立的值區。不過，如果您想要使用現有的儲存貯體，您可以跳到步驟 2。

> **Note**
>
> S3 儲存貯體必須與要匯出的日誌資料位於相同的區域。 CloudWatch 日誌不支援將資料匯出到不同區域中的 S3 儲存貯體。

建立 S3 儲存貯體

1. 前往 https://console.aws.amazon.com/s3/ 開啟的 Amazon Simple Storage Service (Amazon S3) 主控台。
2. 如有必要請變更區域。在導覽列中，選擇記 CloudWatch 錄所在的區域。
3. 選擇 Create Bucket (建立儲存貯體)。
4. 針對 Bucket Name (儲存貯體名稱)，輸入儲存貯體的名稱。
5. 針對「區域」，選取 CloudWatch 記錄資料所在的區域。
6. 選擇建立。

步驟 2：設置存取許可

若要在步驟 5 中建立匯出任務，您將需要使用 `AmazonS3ReadOnlyAccess` IAM 角色登入且要具有以下許可：

- `logs:CreateExportTask`
- `logs:CancelExportTask`
- `logs:DescribeExportTasks`
- `logs:DescribeLogStreams`
- `logs:DescribeLogGroups`

若要提供存取權，請新增權限至您的使用者、群組或角色：

- 使用者和群組位於 AWS IAM Identity Center：

  建立權限合集。請按照 AWS IAM Identity Center 使用者指南 中的 建立權限合集 說明進行操作。
- 透過身分提供者在 IAM 中管理的使用者：

  建立聯合身分的角色。請按照 IAM 使用者指南 的 為第三方身分提供者 (聯合) 建立角色 中的指示進行操作。

- IAM 使用者：
  - 建立您的使用者可擔任的角色。請按照 IAM 使用者指南 的 為 IAM 使用者建立角色 中的指示進行操作。
  - (不建議) 將政策直接附加至使用者，或將使用者新增至使用者群組。請遵循 IAM 使用者指南的新增許可到使用者 (主控台)中的指示。

步驟 3：設定 S3 儲存貯體的許可

依據預設，所有 S3 儲存貯體與物件皆為私有。只有資源擁有者、建立儲存貯體的 AWS 帳戶 ，才能存取儲存貯體及其包含的任何物件。不過，資源擁有者可藉由編寫存取政策，選擇將存取許可授予其他資源或使用者。

當您設定政策時，我們建議您包含隨機產生的字串做為儲存貯體的前綴，如此一來，只有適用的日誌串流才會匯出到儲存貯體。

> **Important**
>
> 為了使匯出至 S3 儲存貯體更加安全，現在要求您指定允許將日誌資料匯出至 S3 儲存貯體的來源帳戶清單。
> 在下列範例中，`aws:SourceAccount`金鑰中的帳戶 ID 清單將是使用者可以從中將日誌資料匯出到 S3 儲存貯體的帳戶。`aws:SourceArn` 金鑰會是正在採取行動的資源。您可以將其限制為具體的日誌群組，或使用萬用字元，如本範例所示。
> 我們建議您也包含建立 S3 儲存貯體之帳戶的帳戶 ID，以允許在同一帳戶內匯出。

設定 Amazon S3 儲存貯體的許可

1. 在 Amazon S3 主控台中，選擇您在步驟 1 建立的儲存貯體。
2. 選擇 Permissions (許可)、Bucket policy (儲存貯體政策)。
3. 在 Bucket Policy Editor (儲存貯體政策編輯器) 中，輸入下列政策。將 `my-exported-logs` 變更為 Amazon S3 儲存貯體的名稱。務必為主體指定正確的區域端點，例如 `us-west-1`。

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
```

```
            "Resource": "arn:aws:s3:::my-exported-logs",
            "Principal": { "Service": "logs.Region.amazonaws.com" },
            "Condition": {
              "StringEquals": {
                  "aws:SourceAccount": [
                      "AccountId1",
                      "AccountId2",
                      ...
                  ]
              },
              "ArnLike": {
                      "aws:SourceArn": [
                          "arn:aws:logs:Region:AccountId1:log-group:*",
                          "arn:aws:logs:Region:AccountId2:log-group:*",
                          ...
                       ]
              }
            }
        },
        {
            "Action": "s3:PutObject" ,
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::my-exported-logs/*",
            "Principal": { "Service": "logs.Region.amazonaws.com" },
            "Condition": {
              "StringEquals": {
                  "s3:x-amz-acl": "bucket-owner-full-control",
                  "aws:SourceAccount": [
                      "AccountId1",
                      "AccountId2",
                      ...
                  ]
              },
              "ArnLike": {
                      "aws:SourceArn": [
                          "arn:aws:logs:Region:AccountId1:log-group:*",
                          "arn:aws:logs:Region:AccountId2:log-group:*",
                          ...
                      ]
              }
            }
        }
    ]
```

```
}
```

4. 選擇 Save (儲存)，將您剛才新增的政策設定為您儲存貯體上的存取政策。此政策可讓 CloudWatch 日誌將日誌資料匯出到 S3 儲存貯體。儲存貯體擁有者擁有所有匯出物件的完整許可。

> **Warning**
>
> 如果現有值區已附加一或多個政策，請新增該政策或政策的 CloudWatch 記錄存取權限的陳述式。我們建議您評估所產生的一組許可，以確保它們適用於將存取儲存貯體的使用者。

(選用) 步驟 4：匯出至使用 SSE-KMS 加密的儲存貯體

只有在匯出到使用伺服器端加密的 S3 儲存貯體時，才需要執行此步驟 AWS KMS keys。這種加密稱為 SSE-KMS。

匯出至使用 SSE-KMS 加密的儲存貯體

1. 請在以下位置開啟 AWS KMS 主控台。 https://console.aws.amazon.com/kms
2. 若要變更 AWS 區域，請使用頁面右上角的「地區」選取器。
3. 在左側導覽列中，選擇 Customer managed keys (客戶受管金鑰)。

   選擇 Create Key (建立金鑰)。
4. 針對 Key type (金鑰類型)，請選擇 Symmetric (對稱)。
5. 在 Key usage (金鑰用途) 中，選擇 Encrypt and decrypt (加密與解密)，然後選擇 Next (下一步)。
6. 在 Add labels (加入標示) 下，輸入金鑰的別名，並選擇是否新增說明或標籤。然後選擇下一步。
7. 在 Key administrators (金鑰管理員) 下，選取可管理此金鑰的人員，然後選擇 Next (下一步)。
8. 在 Define key usage permissions (定義金鑰用途許可) 下，不進行變更，然後選擇 Next (下一步)。
9. 檢閱設定，然後選擇 Finish (完成)。
10. 返回 Customer managed keys (客戶受管金鑰) 頁面，選擇您剛建立的金鑰名稱。
11. 選擇 Key policy (金鑰政策) 索引標籤，並選擇 Switch to policy view (切換至政策檢視)。
12. 在 Key policy (金鑰政策) 區段中，選擇 Edit (編輯)。

13. 將下列陳述式新增至金鑰政策陳述式清單。執行時，請將 *Region* 替換為日誌區域，並以所擁有 KMS 金鑰的帳戶 ARN 替換 *account-ARN*。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow CWL Service Principal usage",
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.Region.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "account-ARN"
            },
            "Action": [
                "kms:GetKeyPolicy*",
                "kms:PutKeyPolicy*",
                "kms:DescribeKey*",
                "kms:CreateAlias*",
                "kms:ScheduleKeyDeletion*",
                "kms:Decrypt"
            ],
            "Resource": "*"
        }
    ]
}
```

14. 選擇儲存變更。
15. 前往 https://console.aws.amazon.com/s3/ 開啟的 Amazon Simple Storage Service (Amazon S3) 主控台。
16. 尋找您在 步驟 1：建立 S3 儲存貯體 中建立的儲存貯體，選擇儲存貯體名稱。
17. 選擇屬性索引標籤。在 Default encryption (預設加密) 下，選擇 Edit (編輯)。

18. 在 Server-side Encryption (伺服器端加密) 下，選擇 Enable (啟用)。
19. 在 Encryption type (加密類型) 下，選擇 AWS Key Management Service key (SSE-KMS) (金鑰 (SSE-KMS))。
20. 從您的 AWS KMS 金鑰中選擇「選擇」，然後尋找您建立的金鑰。
21. 在 Bucket Key (儲存貯體金鑰) 下，選擇 Enable (啟用)。
22. 選擇儲存變更。

步驟 5：建立匯出任務

在此步驟中，您會建立從日誌群組匯出日誌的匯出任務。

若要使用 CloudWatch 主控台將資料匯出到 Amazon S3

1. 如 步驟 2：設置存取許可 中所示，以足夠的許可登入。
2. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
3. 在導覽窗格中，選擇 Log groups (日誌群組)。
4. 在 Log Groups (日誌群組) 畫面上，選擇日誌群組的名稱。
5. 選擇 Actions (動作)、Export data to Amazon S3 (匯出資料至 Amazon S3)。
6. 在 Export data to Amazon S3 (匯出資料至 Amazon S3) 畫面的 Define data to export (定義資料匯出) 下方，使用 From (從) 和 To (至) 設定所要匯出資料的時間範圍。
7. 如果您的日誌群組有多個日誌串流，您可以提供日誌串流前綴，將日誌群組資料限制於特定串流。選擇 Advanced (進階)，然後針對 Stream prefix (串流前綴)，輸入日誌串流前綴。
8. 在 Choose S3 bucket (選擇 S3 儲存貯體) 下，選擇與 S3 儲存貯體關聯的帳戶。
9. 在 S3 bucket name (S3 儲存貯體名稱) 中，選擇一個 S3 儲存貯體。
10. 針對 S3 Bucket prefix (S3 儲存貯體前綴)，輸入您在儲存貯體政策中指定的隨機產生字串。
11. 選擇 Export (匯出) 將您的日誌資料匯出至 Amazon S3。
12. 若要檢視您匯出至 Amazon S3 的日誌資料的狀態，請選擇 Actions (動作，然後選擇 View all exports to Amazon S3 (檢視所有匯出至 Amazon S3 的項目)。

跨帳戶匯出

如果 Amazon S3 儲存貯體與要匯出的日誌位於不同帳戶，請參閱本區段的說明。

主題

步驟 1：建立 Amazon S3 儲存貯體

我們建議您使用專門為 CloudWatch Logs 建立的值區。不過，如果您想要使用現有的儲存貯體，您可以跳到步驟 2。

> **Note**
>
> S3 儲存貯體必須與要匯出的日誌資料位於相同的區域。 CloudWatch 日誌不支援將資料匯出到不同區域中的 S3 儲存貯體。

建立 S3 儲存貯體

1. 前往 https://console.aws.amazon.com/s3/ 開啟的 Amazon Simple Storage Service (Amazon S3) 主控台。
2. 如有必要請變更區域。在導覽列中，選擇記 CloudWatch 錄所在的區域。
3. 選擇 Create Bucket (建立儲存貯體)。
4. 針對 Bucket Name (儲存貯體名稱)，輸入儲存貯體的名稱。
5. 針對「區域」，選取 CloudWatch 記錄資料所在的區域。
6. 選擇建立。

步驟 2：設置存取許可

首先，您必須建立新的 IAM 政策，讓 CloudWatch 日誌擁有目的地帳戶中目的地 Amazon S3 儲存貯體的`s3:PutObject`許可。

您建立的政策取決於目的地儲存貯體是否使用 AWS KMS 加密。

若要建立 IAM 政策將日誌匯出至 Amazon S3 儲存貯體

1. 前往 https://console.aws.amazon.com/iam/ 開啟 IAM 主控台。
2. 在左側的導覽窗格中，選擇 Policies (政策)。
3. 選擇 Create policy (建立政策)。
4. 在政策編輯器區段中，選擇 JSON。
5. 如果目的地值區未使用 AWS KMS 加密，請將下列原則貼到編輯器中。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-exported-logs/*"
        }
    ]
}
```

如果目的地儲存貯體確實使用 AWS KMS 加密，請將下列原則貼到編輯器中。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::my-exported-logs/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource": "ARN_OF_KMS_KEY"
    }
  ]
}
```

6. 選擇下一步。

7. 輸入政策名稱。您會使用此名稱為您的 IAM 角色附加政策。
8. 選擇建立政策，儲存新政策。

若要在步驟 5 中建立匯出任務，您將需要使用 AmazonS3ReadOnlyAccess IAM 角色登入。您也必須使用剛才建立的 IAM 政策登入，並具有以下許可：

- logs:CreateExportTask
- logs:CancelExportTask
- logs:DescribeExportTasks
- logs:DescribeLogStreams
- logs:DescribeLogGroups

若要提供存取權，請新增權限至您的使用者、群組或角色：

- 使用者和群組位於 AWS IAM Identity Center：

  建立權限合集。請按照 AWS IAM Identity Center 使用者指南 中的 建立權限合集 說明進行操作。
- 透過身分提供者在 IAM 中管理的使用者：

  建立聯合身分的角色。請按照 IAM 使用者指南 的 為第三方身分提供者 (聯合) 建立角色 中的指示進行操作。
- IAM 使用者：
  - 建立您的使用者可擔任的角色。請按照 IAM 使用者指南 的 為 IAM 使用者建立角色 中的指示進行操作。
  - (不建議) 將政策直接附加至使用者，或將使用者新增至使用者群組。請遵循 IAM 使用者指南的新增許可到使用者 (主控台)中的指示。

步驟 3：設定 S3 儲存貯體的許可

依據預設，所有 S3 儲存貯體與物件皆為私有。只有資源擁有者、建立儲存貯體的 AWS 帳戶 ，才能存取儲存貯體及其包含的任何物件。不過，資源擁有者可藉由編寫存取政策，選擇將存取許可授予其他資源或使用者。

當您設定政策時，我們建議您包含隨機產生的字串做為儲存貯體的前綴，如此一來，只有適用的日誌串流才會匯出到儲存貯體。

> **Important**
>
> 為了使匯出至 S3 儲存貯體更加安全，現在要求您指定允許將日誌資料匯出至 S3 儲存貯體的來源帳戶清單。
> 在下列範例中，aws:SourceAccount金鑰中的帳戶 ID 清單將是使用者可以從中將日誌資料匯出到 S3 儲存貯體的帳戶。aws:SourceArn 金鑰會是正在採取行動的資源。您可以將其限制為具體的日誌群組，或使用萬用字元，如本範例所示。
> 我們建議您也包含建立 S3 儲存貯體之帳戶的帳戶 ID，以允許在同一帳戶內匯出。

設定 Amazon S3 儲存貯體的許可

1. 在 Amazon S3 主控台中，選擇您在步驟 1 建立的儲存貯體。
2. 選擇 Permissions (許可)、Bucket policy (儲存貯體政策)。
3. 在 Bucket Policy Editor (儲存貯體政策編輯器) 中，輸入下列政策。將 my-exported-logs 變更為 Amazon S3 儲存貯體的名稱。務必為主體指定正確的區域端點，例如 us-west-1。

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs",
          "Principal": { "Service": "logs.Region.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "aws:SourceAccount": [
                    "AccountId1",
                    "AccountId2",
                    ...
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:Region:AccountId1:log-group:*",
                        "arn:aws:logs:Region:AccountId2:log-group:*",
                        ...
                     ]
            }
```

```
            }
        },
        {
            "Action": "s3:PutObject" ,
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::my-exported-logs/*",
            "Principal": { "Service": "logs.Region.amazonaws.com" },
            "Condition": {
              "StringEquals": {
                  "s3:x-amz-acl": "bucket-owner-full-control",
                  "aws:SourceAccount": [
                      "AccountId1",
                      "AccountId2",
                      ...
                  ]
              },
              "ArnLike": {
                      "aws:SourceArn": [
                          "arn:aws:logs:Region:AccountId1:log-group:*",
                          "arn:aws:logs:Region:AccountId2:log-group:*",
                          ...
                      ]
              }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
              "AWS": "arn:aws:iam::create_export_task_caller_account:role/role_name"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-exported-logs/*",
            "Condition": {
              "StringEquals": {
                  "s3:x-amz-acl": "bucket-owner-full-control"
              }
            }
        }
    ]
}
```

4. 選擇 Save (儲存)，將您剛才新增的政策設定為您儲存貯體上的存取政策。此政策可讓 CloudWatch 日誌將日誌資料匯出到 S3 儲存貯體。儲存貯體擁有者擁有所有匯出物件的完整許可。

> **Warning**
>
> 如果現有值區已附加一或多個政策，請新增該政策或政策的 CloudWatch 記錄存取權限的陳述式。我們建議您評估所產生的一組許可，以確保它們適用於將存取儲存貯體的使用者。

(選用) 步驟 4：匯出至使用 SSE-KMS 加密的儲存貯體

只有在匯出到使用伺服器端加密的 S3 儲存貯體時，才需要執行此步驟 AWS KMS keys。這種加密稱為 SSE-KMS。

匯出至使用 SSE-KMS 加密的儲存貯體

1. 請在以下位置開啟 AWS KMS 主控台。 https://console.aws.amazon.com/kms
2. 若要變更 AWS 區域，請使用頁面右上角的「地區」選取器。
3. 在左側導覽列中，選擇 Customer managed keys (客戶受管金鑰)。

   選擇 Create Key (建立金鑰)。
4. 針對 Key type (金鑰類型)，請選擇 Symmetric (對稱)。
5. 在 Key usage (金鑰用途) 中，選擇 Encrypt and decrypt (加密與解密)，然後選擇 Next (下一步)。
6. 在 Add labels (加入標示) 下，輸入金鑰的別名，並選擇是否新增說明或標籤。然後選擇下一步。
7. 在 Key administrators (金鑰管理員) 下，選取可管理此金鑰的人員，然後選擇 Next (下一步)。
8. 在 Define key usage permissions (定義金鑰用途許可) 下，不進行變更，然後選擇 Next (下一步)。
9. 檢閱設定，然後選擇 Finish (完成)。
10. 返回 Customer managed keys (客戶受管金鑰) 頁面，選擇您剛建立的金鑰名稱。
11. 選擇 Key policy (金鑰政策) 索引標籤，並選擇 Switch to policy view (切換至政策檢視)。
12. 在 Key policy (金鑰政策) 區段中，選擇 Edit (編輯)。
13. 將下列陳述式新增至金鑰政策陳述式清單。執行時，請將 *Region* 替換為日誌區域，並以所擁有 KMS 金鑰的帳戶 ARN 替換 *account-ARN*。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow CWL Service Principal usage",
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.Region.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "account-ARN"
            },
            "Action": [
                "kms:GetKeyPolicy*",
                "kms:PutKeyPolicy*",
                "kms:DescribeKey*",
                "kms:CreateAlias*",
                "kms:ScheduleKeyDeletion*",
                "kms:Decrypt"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM Role Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS":
 "arn:aws:iam::create_export_task_caller_account:role/role_name"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "ARN_OF_KMS_KEY"
```

```
            }
        ]
}
```

14. 選擇儲存變更。
15. 前往 https://console.aws.amazon.com/s3/ 開啟的 Amazon Simple Storage Service (Amazon S3) 主控台。
16. 尋找您在 步驟 1：建立 S3 儲存貯體 中建立的儲存貯體，選擇儲存貯體名稱。
17. 選擇屬性索引標籤。在 Default encryption (預設加密) 下，選擇 Edit (編輯)。
18. 在 Server-side Encryption (伺服器端加密) 下，選擇 Enable (啟用)。
19. 在 Encryption type (加密類型) 下，選擇 AWS Key Management Service key (SSE-KMS) (金鑰 (SSE-KMS))。
20. 從您的 AWS KMS 金鑰中選擇「選擇」，然後尋找您建立的金鑰。
21. 在 Bucket Key (儲存貯體金鑰) 下，選擇 Enable (啟用)。
22. 選擇儲存變更。

步驟 5：建立匯出任務

在此步驟中，您會建立從日誌群組匯出日誌的匯出任務。

若要使用 CloudWatch 主控台將資料匯出到 Amazon S3

1. 如 步驟 2：設置存取許可 中所示，以足夠的許可登入。
2. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
3. 在導覽窗格中，選擇 Log groups (日誌群組)。
4. 在 Log Groups (日誌群組) 畫面上，選擇日誌群組的名稱。
5. 選擇 Actions (動作)、Export data to Amazon S3 (匯出資料至 Amazon S3)。
6. 在 Export data to Amazon S3 (匯出資料至 Amazon S3) 畫面的 Define data to export (定義資料匯出) 下方，使用 From (從) 和 To (至) 設定所要匯出資料的時間範圍。
7. 如果您的日誌群組有多個日誌串流，您可以提供日誌串流前綴，將日誌群組資料限制於特定串流。選擇 Advanced (進階)，然後針對 Stream prefix (串流前綴)，輸入日誌串流前綴。
8. 在 Choose S3 bucket (選擇 S3 儲存貯體) 下，選擇與 S3 儲存貯體關聯的帳戶。
9. 在 S3 bucket name (S3 儲存貯體名稱) 中，選擇一個 S3 儲存貯體。
10. 針對 S3 Bucket prefix (S3 儲存貯體前綴)，輸入您在儲存貯體政策中指定的隨機產生字串。

11. 選擇 Export (匯出) 將您的日誌資料匯出至 Amazon S3。
12. 若要檢視您匯出至 Amazon S3 的日誌資料的狀態，請選擇 Actions (動作，然後選擇 View all exports to Amazon S3 (檢視所有匯出至 Amazon S3 的項目)。

使用將日誌資料匯出到 Amazon S3 AWS CLI

在下列範例中，您使用匯出任務將 CloudWatch 日誌日誌群組中的所有資料匯出`my-log-group`到名為的 Amazon S3 儲存貯體`my-exported-logs`。此範例假設您已建立一個名為 `my-log-group` 的日誌群組。

支援將日誌資料匯出至由 AWS KMS 加密的 S3 儲存貯體。不支援匯出至使用 DSSE-KMS 加密的儲存貯體。

如何設定匯出作業的詳細方法，取決於您要存放匯出資料的 Amazon S3 儲存貯體是否與要匯出的日誌位於同一帳戶。

主題

同帳戶匯出

如果 Amazon S3 儲存貯體與要匯出的日誌位於同一帳戶，請參閱本區段的說明。

主題

步驟 1：建立 S3 儲存貯體

我們建議您使用專門為 CloudWatch Logs 建立的值區。不過，如果您想要使用現有的儲存貯體，您可以跳到步驟 2。

> **Note**
>
> S3 儲存貯體必須與要匯出的日誌資料位於相同的區域。 CloudWatch 日誌不支援將資料匯出到不同區域中的 S3 儲存貯體。

若要使用建立 S3 儲存貯體 AWS CLI

在命令提示中執行以下 create-bucket 命令，其中的 `LocationConstraint` 是您要匯出日誌資料的區域。

```
aws s3api create-bucket --bucket my-exported-logs --create-bucket-configuration
 LocationConstraint=us-east-2
```

下列為範例輸出。

```
{
    "Location": "/my-exported-logs"
}
```

步驟 2：設置存取許可

若要在步驟 5 中建立匯出任務，您將需要使用 `AmazonS3ReadOnlyAccess` IAM 角色登入且要具有以下許可：

- `logs:CreateExportTask`
- `logs:CancelExportTask`
- `logs:DescribeExportTasks`
- `logs:DescribeLogStreams`
- `logs:DescribeLogGroups`

若要提供存取權，請新增權限至您的使用者、群組或角色：

- 使用者和群組位於 AWS IAM Identity Center：

  建立權限合集。請按照 AWS IAM Identity Center 使用者指南 中的 建立權限合集 說明進行操作。
- 透過身分提供者在 IAM 中管理的使用者：

建立聯合身分的角色。請按照 IAM 使用者指南 的 為第三方身分提供者 (聯合) 建立角色 中的指示進行操作。

- IAM 使用者：
  - 建立您的使用者可擔任的角色。請按照 IAM 使用者指南 的 為 IAM 使用者建立角色 中的指示進行操作。
  - (不建議) 將政策直接附加至使用者，或將使用者新增至使用者群組。請遵循 IAM 使用者指南的新增許可到使用者 (主控台)中的指示。

步驟 3：設定 S3 儲存貯體的許可

依據預設，所有 S3 儲存貯體與物件皆為私有。只有資源擁有者、建立儲存貯體的帳戶，才能存取儲存貯體及其包含的任何物件。不過，資源擁有者可藉由編寫存取政策，選擇將存取許可授予其他資源或使用者。

> **Important**
>
> 為了使匯出至 S3 儲存貯體更加安全，現在要求您指定允許將日誌資料匯出至 S3 儲存貯體的來源帳戶清單。
> 在下列範例中，aws:SourceAccount金鑰中的帳戶 ID 清單將是使用者可以從中將日誌資料匯出到 S3 儲存貯體的帳戶。aws:SourceArn 金鑰會是正在採取行動的資源。您可以將其限制為具體的日誌群組，或使用萬用字元，如本範例所示。
> 我們建議您也包含建立 S3 儲存貯體之帳戶的帳戶 ID，以允許在同一帳戶內匯出。

設定 S3 儲存貯體的許可

1. 建立名為 policy.json 的檔案，然後新增以下存取政策，將 my-exported-logs 變更為您的 S3 儲存貯體名稱，然後將 Principal 變更為您匯出日誌資料的區域端點，例如us-west-1 。請使用文字編輯器來建立此政策檔案。請勿使用 IAM 主控台。

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs",
```

```
            "Principal": { "Service": "logs.Region.amazonaws.com" },
            "Condition": {
              "StringEquals": {
                  "aws:SourceAccount": [
                      "AccountId1",
                      "AccountId2",
                      ...
                  ]
              },
              "ArnLike": {
                      "aws:SourceArn": [
                          "arn:aws:logs:Region:AccountId1:log-group:*",
                          "arn:aws:logs:Region:AccountId2:log-group:*",
                          ...
                       ]
              }
            }
        },
        {
            "Action": "s3:PutObject" ,
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::my-exported-logs/*",
            "Principal": { "Service": "logs.Region.amazonaws.com" },
            "Condition": {
              "StringEquals": {
                  "s3:x-amz-acl": "bucket-owner-full-control",
                  "aws:SourceAccount": [
                      "AccountId1",
                      "AccountId2",
                      ...
                  ]
              },
              "ArnLike": {
                      "aws:SourceArn": [
                          "arn:aws:logs:Region:AccountId1:log-group:*",
                          "arn:aws:logs:Region:AccountId2:log-group:*",
                          ...
                      ]
              }
            }
        }
    ]
}
```

2. 使用put-bucket-policy指令設定剛新增為值區存取原則的原則。此政策可讓 CloudWatch 日誌將日誌資料匯出到 S3 儲存貯體。儲存貯體擁有者將擁有所有匯出物件的完整許可。

```
aws s3api put-bucket-policy --bucket my-exported-logs --policy file://policy.json
```

> **Warning**
>
> 如果現有值區已附加一或多個政策，請新增該政策或政策的 CloudWatch 記錄存取權限的陳述式。我們建議您評估所產生的一組許可，以確保它們適用於將存取儲存貯體的使用者。

(選用) 步驟 4：匯出至使用 SSE-KMS 加密的儲存貯體

只有在匯出到使用伺服器端加密的 S3 儲存貯體時，才需要執行此步驟 AWS KMS keys。這種加密稱為 SSE-KMS。

匯出至使用 SSE-KMS 加密的儲存貯體

1. 使用文字編輯器建立名為 `key_policy.json` 的檔案，並新增下列存取政策。新增政策時，進行下列變更：

   - 將 *Region* 替換成您的日誌區域。
   - 將 *account-ARN* 替換成所擁有 KMS 金鑰的帳戶 ARN。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow CWL Service Principal usage",
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.Region.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*"
```

```
        },
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "account-ARN"
            },
            "Action": [
                "kms:GetKeyPolicy*",
                "kms:PutKeyPolicy*",
                "kms:DescribeKey*",
                "kms:CreateAlias*",
                "kms:ScheduleKeyDeletion*",
                "kms:Decrypt"
            ],
            "Resource": "*"
        }
    ]
}
```

2. 輸入以下命令：

```
aws kms create-key --policy file://key_policy.json
```

以下為此命令的範例輸出：

```
{
    "KeyMetadata": {
        "AWSAccountId": "account_id",
        "KeyId": "key_id",
        "Arn": "arn:aws:kms:us-east-2:account_id:key/key_id",
        "CreationDate": "time",
        "Enabled": true,
        "Description": "",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "Origin": "AWS_KMS",
        "KeyManager": "CUSTOMER",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
```

```
        "MultiRegion": false
    }
```

3. 使用文字編輯器建立名為 bucketencryption.json 的檔案，包含下列內容。

```
{
  "Rules": [
    {
      "ApplyServerSideEncryptionByDefault": {
        "SSEAlgorithm": "aws:kms",
        "KMSMasterKeyID": "{KMS Key ARN}"
      },
      "BucketKeyEnabled": true
    }
  ]
}
```

4. 輸入下列命令，將 *bucket-NAME* 替換為日誌匯出所至的儲存貯體名稱。

```
aws s3api put-bucket-encryption --bucket bucket-name --server-side-encryption-
configuration file://bucketencryption.json
```

如果命令沒有傳回錯誤，則表示該流程成功。

步驟 5：建立匯出任務

使用以下命令建立匯出任務。在您建立匯出任務後，此任務可能需花費幾秒到幾小時的時間，視匯出資料的大小而定。

若要使用將資料匯出至 Amazon S3 AWS CLI

1. 如 步驟 2：設置存取許可 中所示，以足夠的許可登入。
2. 在命令提示字元中，使用下列create-export-task命令建立匯出工作。

```
aws logs create-export-task --profile CWLExportUser --task-name "my-log-
group-09-10-2015" --log-group-name "my-log-group" --from 1441490400000 --
to 1441494000000 --destination "my-exported-logs" --destination-prefix "export-
task-output"
```

下列為範例輸出。

```
{
    "taskId": "cda45419-90ea-4db5-9833-aade86253e66"
}
```

跨帳戶匯出

如果 Amazon S3 儲存貯體與要匯出的日誌位於不同帳戶，請參閱本區段的說明。

主題

- 步驟 1：建立 S3 儲存貯體
- 步驟 2：設置存取許可
- 步驟 3：設定 S3 儲存貯體的許可
- (選用) 步驟 4：匯出至使用 SSE-KMS 加密的儲存貯體
- 步驟 5：建立匯出任務

步驟 1：建立 S3 儲存貯體

我們建議您使用專門為 CloudWatch Logs 建立的值區。不過，如果您想要使用現有的儲存貯體，您可以跳到步驟 2。

> **Note**
>
> S3 儲存貯體必須與要匯出的日誌資料位於相同的區域。 CloudWatch 日誌不支援將資料匯出到不同區域中的 S3 儲存貯體。

若要使用建立 S3 儲存貯體 AWS CLI

在命令提示中執行以下 create-bucket 命令，其中的 `LocationConstraint` 是您要匯出日誌資料的區域。

```
aws s3api create-bucket --bucket my-exported-logs --create-bucket-configuration
 LocationConstraint=us-east-2
```

下列為範例輸出。

```
{
    "Location": "/my-exported-logs"
}
```

步驟 2：設置存取許可

首先，您必須建立新的 IAM 政策，以使 CloudWatch 日誌擁有目的地 Amazon S3 儲存貯體的`s3:PutObject`許可。

若要在步驟 5 中建立匯出任務，您將需要使用 `AmazonS3ReadOnlyAccess` IAM 角色登入，並具有其他特定許可。您可以建立包含其他部分必要許可的政策。

您建立的政策取決於目的地儲存貯體是否使用 AWS KMS 加密。如果未使用 AWS KMS 加密，請建立包含下列內容的策略。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-exported-logs/*"
        }
    ]
}
```

如果目的地儲存貯體使用 AWS KMS 加密，請建立包含下列內容的政策。

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-exported-logs/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
```

```
            "Resource": "ARN_OF_KMS_KEY"
        }
    ]
}
```

若要在步驟 5 中建立匯出任務，您必須使用 AmazonS3ReadOnlyAccess IAM 角色登入、執行剛才建立的 IAM 政策，並具有下列許可：

- logs:CreateExportTask
- logs:CancelExportTask
- logs:DescribeExportTasks
- logs:DescribeLogStreams
- logs:DescribeLogGroups

若要提供存取權，請新增權限至您的使用者、群組或角色：

- 使用者和群組位於 AWS IAM Identity Center：

  建立權限合集。請按照 AWS IAM Identity Center 使用者指南 中的 建立權限合集 說明進行操作。

- 透過身分提供者在 IAM 中管理的使用者：

  建立聯合身分的角色。請按照 IAM 使用者指南 的 為第三方身分提供者 (聯合) 建立角色 中的指示進行操作。

- IAM 使用者：
  - 建立您的使用者可擔任的角色。請按照 IAM 使用者指南 的 為 IAM 使用者建立角色 中的指示進行操作。
  - (不建議) 將政策直接附加至使用者，或將使用者新增至使用者群組。請遵循 IAM 使用者指南的新增許可到使用者 (主控台)中的指示。

步驟 3：設定 S3 儲存貯體的許可

依據預設，所有 S3 儲存貯體與物件皆為私有。只有資源擁有者、建立儲存貯體的帳戶，才能存取儲存貯體及其包含的任何物件。不過，資源擁有者可藉由編寫存取政策，選擇將存取許可授予其他資源或使用者。

> **Important**
>
> 為了使匯出至 S3 儲存貯體更加安全，現在要求您指定允許將日誌資料匯出至 S3 儲存貯體的來源帳戶清單。
> 在下列範例中，aws:SourceAccount金鑰中的帳戶 ID 清單將是使用者可以從中將日誌資料匯出到 S3 儲存貯體的帳戶。aws:SourceArn 金鑰會是正在採取行動的資源。您可以將其限制為具體的日誌群組，或使用萬用字元，如本範例所示。
> 我們建議您也包含建立 S3 儲存貯體之帳戶的帳戶 ID，以允許在同一帳戶內匯出。

設定 S3 儲存貯體的許可

1. 建立名為 policy.json 的檔案，然後新增以下存取政策，將 my-exported-logs 變更為您的 S3 儲存貯體名稱，然後將 Principal 變更為您匯出日誌資料的區域端點，例如us-west-1 。請使用文字編輯器來建立此政策檔案。請勿使用 IAM 主控台。

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs",
          "Principal": { "Service": "logs.Region.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "aws:SourceAccount": [
                    "AccountId1",
                    "AccountId2",
                    ...
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:Region:AccountId1:log-group:*",
                        "arn:aws:logs:Region:AccountId2:log-group:*",
                        ...
                     ]
            }
          }
```

```
      },
      {
          "Action": "s3:PutObject" ,
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs/*",
          "Principal": { "Service": "logs.Region.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "s3:x-amz-acl": "bucket-owner-full-control",
                "aws:SourceAccount": [
                    "AccountId1",
                    "AccountId2",
                    ...
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:Region:AccountId1:log-group:*",
                        "arn:aws:logs:Region:AccountId2:log-group:*",
                        ...
                    ]
            }
          }
      },
      {
          "Effect": "Allow",
          "Principal": {
            "AWS": "arn:aws:iam::create_export_task_caller_account:role/role_name"
          },
          "Action": "s3:PutObject",
          "Resource": "arn:aws:s3:::my-exported-logs/*",
          "Condition": {
            "StringEquals": {
                "s3:x-amz-acl": "bucket-owner-full-control"
            }
          }
       }
    ]
}
```

2. 使用put-bucket-policy指令設定剛新增為值區存取原則的原則。此政策可讓 CloudWatch 日誌將日誌資料匯出到 S3 儲存貯體。儲存貯體擁有者將擁有所有匯出物件的完整許可。

```
aws s3api put-bucket-policy --bucket my-exported-logs --policy file://policy.json
```

> **Warning**
>
> 如果現有值區已附加一或多個政策，請新增該政策或政策的 CloudWatch 記錄存取權限的陳述式。我們建議您評估所產生的一組許可，以確保它們適用於將存取儲存貯體的使用者。

(選用) 步驟 4：匯出至使用 SSE-KMS 加密的儲存貯體

只有在匯出到使用伺服器端加密的 S3 儲存貯體時，才需要執行此步驟 AWS KMS keys。這種加密稱為 SSE-KMS。

匯出至使用 SSE-KMS 加密的儲存貯體

1. 使用文字編輯器建立名為 `key_policy.json` 的檔案，並新增下列存取政策。新增政策時，進行下列變更：

   - 將 *Region* 替換成您的日誌區域。
   - 將 *account-ARN* 替換成所擁有 KMS 金鑰的帳戶 ARN。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow CWL Service Principal usage",
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.Region.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM User Permissions",
```

```
            "Effect": "Allow",
            "Principal": {
                "AWS": "account-ARN"
            },
            "Action": [
                "kms:GetKeyPolicy*",
                "kms:PutKeyPolicy*",
                "kms:DescribeKey*",
                "kms:CreateAlias*",
                "kms:ScheduleKeyDeletion*",
                "kms:Decrypt"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM Role Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS":
 "arn:aws:iam::create_export_task_caller_account:role/role_name"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "ARN_OF_KMS_KEY"
        }
    ]
}
```

2. 輸入以下命令：

```
aws kms create-key --policy file://key_policy.json
```

以下為此命令的範例輸出：

```
{
    "KeyMetadata": {
        "AWSAccountId": "account_id",
        "KeyId": "key_id",
        "Arn": "arn:aws:kms:us-east-2:account_id:key/key_id",
        "CreationDate": "time",
        "Enabled": true,
```

```
        "Description": "",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "Origin": "AWS_KMS",
        "KeyManager": "CUSTOMER",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "MultiRegion": false
    }
```

3. 使用文字編輯器建立名為 bucketencryption.json 的檔案，包含下列內容。

```
{
  "Rules": [
    {
      "ApplyServerSideEncryptionByDefault": {
        "SSEAlgorithm": "aws:kms",
        "KMSMasterKeyID": "{KMS Key ARN}"
      },
      "BucketKeyEnabled": true
    }
  ]
}
```

4. 輸入下列命令，將 *bucket-NAME* 替換為日誌匯出所至的儲存貯體名稱。

```
aws s3api put-bucket-encryption --bucket bucket-name --server-side-encryption-
configuration file://bucketencryption.json
```

如果命令沒有傳回錯誤，則表示該流程成功。

步驟 5：建立匯出任務

使用以下命令建立匯出任務。在您建立匯出任務後，此任務可能需花費幾秒到幾小時的時間，視匯出資料的大小而定。

若要使用將資料匯出至 Amazon S3 AWS CLI

1. 如 步驟 2：設置存取許可 中所示，以足夠的許可登入。

2. 在命令提示字元中，使用下列create-export-task命令建立匯出工作。

```
aws logs create-export-task --profile CWLExportUser --task-name "my-log-group-09-10-2015" --log-group-name "my-log-group" --from 1441490400000 --to 1441494000000 --destination "my-exported-logs" --destination-prefix "export-task-output"
```

下列為範例輸出。

```
{
    "taskId": "cda45419-90ea-4db5-9833-aade86253e66"
}
```

描述匯出任務

在您建立匯出任務之後，您可以取得任務的目前狀態。

若要使用說明匯出工作 AWS CLI

在命令提示字元中，使用下列describe-export-tasks命令。

```
aws logs --profile CWLExportUser describe-export-tasks --task-id
 "cda45419-90ea-4db5-9833-aade86253e66"
```

下列為範例輸出。

```
{
   "exportTasks": [
   {
      "destination": "my-exported-logs",
      "destinationPrefix": "export-task-output",
      "executionInfo": {
         "creationTime": 1441495400000
      },
      "from": 1441490400000,
      "logGroupName": "my-log-group",
      "status": {
         "code": "RUNNING",
         "message": "Started Successfully"
      },
```

```
      "taskId": "cda45419-90ea-4db5-9833-aade86253e66",
      "taskName": "my-log-group-09-10-2015",
      "tTo": 1441494000000
   }]
}
```

您有三種使用 describe-export-tasks 命令的方法：

- 無任何篩選條件 – 以建立順序相反的順序列出您所有的匯出任務。
- 任務 ID 篩選 – 列出指定 ID 的匯出任務 (如果有的話)。
- 任務狀態篩選 – 列出指定狀態的匯出任務。

例如，使用以下命令來篩選 FAILED 狀態。

```
aws logs --profile CWLExportUser describe-export-tasks --status-code "FAILED"
```

下列為範例輸出。

```
{
   "exportTasks": [
   {
      "destination": "my-exported-logs",
      "destinationPrefix": "export-task-output",
      "executionInfo": {
         "completionTime": 1441498600000
         "creationTime": 1441495400000
      },
      "from": 1441490400000,
      "logGroupName": "my-log-group",
      "status": {
         "code": "FAILED",
         "message": "FAILED"
      },
      "taskId": "cda45419-90ea-4db5-9833-aade86253e66",
      "taskName": "my-log-group-09-10-2015",
      "to": 1441494000000
   }]
}
```

取消匯出任務

如果匯出任務處於 PENDING 或 RUNNING 狀態，可以取消匯出任務。

若要使用取消匯出任務 AWS CLI

在命令提示字元中，使用下列cancel-export-task命令：

```
aws logs --profile CWLExportUser cancel-export-task --task-id "cda45419-90ea-4db5-9833-aade86253e66"
```

您可以使用describe-export-tasks指令來確認工作已成功取消。

將 CloudWatch 日誌資料串流至 Amazon OpenSearch 服務

您可以設定 CloudWatch 日誌記錄群組，透過日 CloudWatch 誌訂閱以近乎即時的方式將其接收的資料串流到 Amazon Ser OpenSearch vice 叢集。如需詳細資訊，請參閱 使用訂閱即時處理日誌資料。

> **Note**
>
> 只有標準記錄類別中的記錄群組才支援「串流至 OpenSearch 服務」。如需記錄類別的詳細資訊，請參閱日誌類。

根據串流的日誌資料數量，您可能會希望在函數上設定函數層級的同時執行限制。如需詳細資訊，請參閱 Lambda 函數擴展。

> **Note**
>
> 將大量 CloudWatch 記錄資料串流至 OpenSearch 服務可能會導致高額使用費用。我們建議您在 AWS Billing and Cost Management 主控台中建立預算。如需詳細資訊，請參閱使用 AWS Budgets 管理您的成本。

必要條件

開始之前，請先建立 OpenSearch 服務網域。網域可以有公有存取或 VPC 存取，但您不能在建立網域之後再修改存取類型。您稍後可能想要檢閱 OpenSearch Service 網域設定，並根據叢集將要處理的資料量修改叢集配置。如需建立網域的指示，請參閱建立 OpenSearch 服務網域。

如需有關 OpenSearch 服務的詳細資訊，請參閱 Amazon OpenSearch 服務開發人員指南。

將記錄群組訂閱至 OpenSearch 服務

您可以使用主 CloudWatch 控台來訂閱 OpenSearch 服務的記錄群組。

若要將記錄群組訂閱至 OpenSearch 服務

1. 請在以下位置開啟 CloudWatch 主控台。 https://console.aws.amazon.com/cloudwatch/
2. 在導覽窗格中，選擇 Log groups (日誌群組)。

3. 選取日誌群組的名稱。
4. 選擇操作，訂閱過濾器，創建 Amazon OpenSearch 服務訂閱過濾器。
5. 選擇您是否要串流至此帳戶或其他帳戶中的叢集。

   - 如果您選擇此帳戶，請選取您在前一個步驟所建立的網域。
   - 如果您選擇其他帳戶，請提供網域 ARN 和端點。
6. 對於 Lambda IAM 執行角色，請選擇 Lambda 在執行呼叫時應使用的 IAM 角色 OpenSearch。

   您選擇的 IAM 角色必須符合這些要求：

   - 它必須擁有 `lambda.amazonaws.com` 的信任關係。
   - 它必須包含以下政策：

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
           {
               "Action": [
                   "es:*"
               ],
               "Effect": "Allow",
               "Resource": "arn:aws:es:region:account-id:domain/target-domain-name/
   *"
           }
       ]
   }
   ```

   - 如果目標 OpenSearch 服務網域使用 VPC 存取，則該角色必須附加 AWSLambdaVPCAccessExecutionRole原則。此亞馬遜管理政策授予 Lambda 對客戶 VPC 的存取權，讓 Lambda 能夠寫入 VPC 中的 OpenSearch 端點。
7. 針對 Log format (日誌格式)，選擇日誌格式。
8. 針對 Subscription filter pattern (訂閱篩選條件模式)，輸入要在您的日誌事件中尋找的詞彙或模式。這可確保您只將感興趣的資料傳送至 OpenSearch 叢集。如需詳細資訊，請參閱 使用篩選條件從日誌事件建立指標。
9. (選用) 針對 Select log data to test (選取要測試的日誌資料)，選擇一個日誌串流，然後選擇 Test pattern (測試模式)，以驗證您的搜尋篩選條件是否會傳回您預期的結果。
10. 選擇 Start streaming (開始串流)。

使用 AWS SDK 的 CloudWatch 記錄檔的程式碼範例

下列程式碼範例說明如何搭配 AWS 軟體開發套件 (SDK) 使用 CloudWatch Logs。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境和跨服務範例中查看內容中的動作。

Scenarios (案例) 是向您展示如何呼叫相同服務中的多個函數來完成特定任務的程式碼範例。

Cross-service examples (跨服務範例) 是跨多個 AWS 服務執行的應用程式範例。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱搭配 AWS SDK 使用 CloudWatch 記錄檔。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

程式碼範例

- 使用 AWS SDK 的記 CloudWatch 錄處理行動
  - 搭AssociateKmsKey配 AWS 開發套件或 CLI 使用
  - 搭CancelExportTask配 AWS 開發套件或 CLI 使用
  - 搭CreateExportTask配 AWS 開發套件或 CLI 使用
  - 搭CreateLogGroup配 AWS 開發套件或 CLI 使用
  - 搭CreateLogStream配 AWS 開發套件或 CLI 使用
  - 搭DeleteLogGroup配 AWS 開發套件或 CLI 使用
  - 搭DeleteSubscriptionFilter配 AWS 開發套件或 CLI 使用
  - 搭DescribeExportTasks配 AWS 開發套件或 CLI 使用
  - 搭DescribeLogGroups配 AWS 開發套件或 CLI 使用
  - 搭DescribeSubscriptionFilters配 AWS 開發套件或 CLI 使用
  - 搭GetQueryResults配 AWS 開發套件或 CLI 使用
  - 搭PutSubscriptionFilter配 AWS 開發套件或 CLI 使用
  - 搭StartLiveTail配 AWS 開發套件或 CLI 使用
  - 搭StartQuery配 AWS 開發套件或 CLI 使用
- 使用 AWS SDK 的 CloudWatch 記錄檔案例
  - 使用 CloudWatch 記錄檔執行大型查詢
- 使 AWS 用 SDK 的 CloudWatch 記錄跨服務範例

- 使用排程事件來調用 Lambda 函數

使用 AWS SDK 的記 CloudWatch 錄處理行動

下列程式碼範例示範如何使用 AWS SDK 執行個別 CloudWatch 記錄動作。這些摘錄會呼叫 CloudWatch 記錄 API，是來自必須在內容中執行的大型程式碼摘錄。每個範例都包含一個連結 GitHub，您可以在其中找到設定和執行程式碼的指示。

下列範例僅包含最常使用的動作。如需完整清單，請參閱 Amazon CloudWatch 日誌 API 參考。

範例

- 搭AssociateKmsKey配 AWS 開發套件或 CLI 使用
- 搭CancelExportTask配 AWS 開發套件或 CLI 使用
- 搭CreateExportTask配 AWS 開發套件或 CLI 使用
- 搭CreateLogGroup配 AWS 開發套件或 CLI 使用
- 搭CreateLogStream配 AWS 開發套件或 CLI 使用
- 搭DeleteLogGroup配 AWS 開發套件或 CLI 使用
- 搭DeleteSubscriptionFilter配 AWS 開發套件或 CLI 使用
- 搭DescribeExportTasks配 AWS 開發套件或 CLI 使用
- 搭DescribeLogGroups配 AWS 開發套件或 CLI 使用
- 搭DescribeSubscriptionFilters配 AWS 開發套件或 CLI 使用
- 搭GetQueryResults配 AWS 開發套件或 CLI 使用
- 搭PutSubscriptionFilter配 AWS 開發套件或 CLI 使用
- 搭StartLiveTail配 AWS 開發套件或 CLI 使用
- 搭StartQuery配 AWS 開發套件或 CLI 使用

搭`AssociateKmsKey`配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用`AssociateKmsKey`。

.NET

AWS SDK for .NET

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

```
    using System;
    using System.Threading.Tasks;
    using Amazon.CloudWatchLogs;
    using Amazon.CloudWatchLogs.Model;

    /// <summary>
    /// Shows how to associate an AWS Key Management Service (AWS KMS) key with
    /// an Amazon CloudWatch Logs log group.
    /// </summary>
    public class AssociateKmsKey
    {
        public static async Task Main()
        {
            // This client object will be associated with the same AWS Region
            // as the default user on this system. If you need to use a
            // different AWS Region, pass it as a parameter to the client
            // constructor.
            var client = new AmazonCloudWatchLogsClient();

            string kmsKeyId = "arn:aws:kms:us-west-2:<account-
number>:key/7c9eccc2-38cb-4c4f-9db3-766ee8dd3ad4";
            string groupName = "cloudwatchlogs-example-loggroup";

            var request = new AssociateKmsKeyRequest
            {
                KmsKeyId = kmsKeyId,
                LogGroupName = groupName,
            };

            var response = await client.AssociateKmsKeyAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
```

```
            {
                Console.WriteLine($"Successfully associated KMS key ID:
{kmsKeyId} with log group: {groupName}.");
            }
            else
            {
                Console.WriteLine("Could not make the association between:
{kmsKeyId} and {groupName}.");
            }
        }
    }
```

- 如需 API 詳細資訊，請參閱 AWS SDK for .NET API 參考AssociateKmsKey中的。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱搭配 AWS SDK 使用 CloudWatch 記錄檔。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭CancelExportTask配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用CancelExportTask。

.NET

AWS SDK for .NET

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

```
using System;
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;

/// <summary>
/// Shows how to cancel an Amazon CloudWatch Logs export task.
/// </summary>
```

```
public class CancelExportTask
{
    public static async Task Main()
    {
        // This client object will be associated with the same AWS Region
        // as the default user on this system. If you need to use a
        // different AWS Region, pass it as a parameter to the client
        // constructor.
        var client = new AmazonCloudWatchLogsClient();
        string taskId = "exampleTaskId";

        var request = new CancelExportTaskRequest
        {
            TaskId = taskId,
        };

        var response = await client.CancelExportTaskAsync(request);

        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
            Console.WriteLine($"{taskId} successfully canceled.");
        }
        else
        {
            Console.WriteLine($"{taskId} could not be canceled.");
        }
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for .NET API 參考CancelExportTask中的。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱搭配 AWS SDK 使用 CloudWatch 記錄檔。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭CreateExportTask配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用CreateExportTask。

.NET

AWS SDK for .NET

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

```
    using System;
    using System.Threading.Tasks;
    using Amazon.CloudWatchLogs;
    using Amazon.CloudWatchLogs.Model;

    /// <summary>
    /// Shows how to create an Export Task to export the contents of the Amazon
    /// CloudWatch Logs to the specified Amazon Simple Storage Service (Amazon
S3)
    /// bucket.
    /// </summary>
    public class CreateExportTask
    {
        public static async Task Main()
        {
            // This client object will be associated with the same AWS Region
            // as the default user on this system. If you need to use a
            // different AWS Region, pass it as a parameter to the client
            // constructor.
            var client = new AmazonCloudWatchLogsClient();
            string taskName = "export-task-example";
            string logGroupName = "cloudwatchlogs-example-loggroup";
            string destination = "doc-example-bucket";
            var fromTime = 1437584472382;
            var toTime = 1437584472833;

            var request = new CreateExportTaskRequest
            {
                From = fromTime,
                To = toTime,
                TaskName = taskName,
                LogGroupName = logGroupName,
```

```
                Destination = destination,
            };

            var response = await client.CreateExportTaskAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"The task, {taskName} with ID: " +
                                  $"{response.TaskId} has been created
successfully.");
            }
        }
    }
```

- 如需 API 詳細資訊，請參閱 AWS SDK for .NET API 參考CreateExportTask中的。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱搭配 AWS SDK 使用 CloudWatch 記錄檔。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭CreateLogGroup配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用CreateLogGroup。

.NET

AWS SDK for .NET

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

```
using System;
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;

/// <summary>
```

```
    /// Shows how to create an Amazon CloudWatch Logs log group.
    /// </summary>
    public class CreateLogGroup
    {
        public static async Task Main()
        {
            // This client object will be associated with the same AWS Region
            // as the default user on this system. If you need to use a
            // different AWS Region, pass it as a parameter to the client
            // constructor.
            var client = new AmazonCloudWatchLogsClient();

            string logGroupName = "cloudwatchlogs-example-loggroup";

            var request = new CreateLogGroupRequest
            {
                LogGroupName = logGroupName,
            };

            var response = await client.CreateLogGroupAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"Successfully create log group with ID:
{logGroupName}.");
            }
            else
            {
                Console.WriteLine("Could not create log group.");
            }
        }
    }
```

- 如需 API 詳細資訊，請參閱 AWS SDK for .NET API 參考CreateLogGroup中的。

CLI

AWS CLI

下列命令會建立名為的記錄群組my-logs：

```
aws logs create-log-group --log-group-name my-logs
```

- 如需 API 詳細資訊，請參閱AWS CLI 命令參考[CreateLogGroup]中的。

JavaScript

適用於 JavaScript (v3) 的開發套件

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫]。

```
import { CreateLogGroupCommand } from "@aws-sdk/client-cloudwatch-logs";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new CreateLogGroupCommand({
    // The name of the log group.
    logGroupName: process.env.CLOUDWATCH_LOGS_LOG_GROUP,
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

- 如需 API 詳細資訊，請參閱 AWS SDK for JavaScript API 參考[CreateLogGroup]中的。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用 CloudWatch 記錄檔]。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭CreateLogStream配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用CreateLogStream。

.NET

AWS SDK for .NET

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

```
using System;
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;

/// <summary>
/// Shows how to create an Amazon CloudWatch Logs stream for a CloudWatch
/// log group.
/// </summary>
public class CreateLogStream
{
    public static async Task Main()
    {
        // This client object will be associated with the same AWS Region
        // as the default user on this system. If you need to use a
        // different AWS Region, pass it as a parameter to the client
        // constructor.
        var client = new AmazonCloudWatchLogsClient();
        string logGroupName = "cloudwatchlogs-example-loggroup";
        string logStreamName = "cloudwatchlogs-example-logstream";

        var request = new CreateLogStreamRequest
        {
            LogGroupName = logGroupName,
            LogStreamName = logStreamName,
        };

        var response = await client.CreateLogStreamAsync(request);
```

```
            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"{logStreamName} successfully created for
{logGroupName}.");
            }
            else
            {
                Console.WriteLine("Could not create stream.");
            }
        }
    }
```

- 如需 API 詳細資訊，請參閱 AWS SDK for .NET API 參考CreateLogStream中的。

CLI

AWS CLI

下列命令會建立記錄群組20150601中名為的記錄資料流my-logs：

```
aws logs create-log-stream --log-group-name my-logs --log-stream-name 20150601
```

- 如需 API 詳細資訊，請參閱AWS CLI 命令參考CreateLogStream中的。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱搭配 AWS SDK 使用 CloudWatch 記錄檔。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭DeleteLogGroup配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用DeleteLogGroup。

.NET

AWS SDK for .NET

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

```
using System;
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;

/// <summary>
/// Uses the Amazon CloudWatch Logs Service to delete an existing
/// CloudWatch Logs log group.
/// </summary>
public class DeleteLogGroup
{
    public static async Task Main()
    {
        var client = new AmazonCloudWatchLogsClient();
        string logGroupName = "cloudwatchlogs-example-loggroup";

        var request = new DeleteLogGroupRequest
        {
            LogGroupName = logGroupName,
        };

        var response = await client.DeleteLogGroupAsync(request);

        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
            Console.WriteLine($"Successfully deleted CloudWatch log group,
{logGroupName}.");
        }
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for .NET API 參考DeleteLogGroup中的。

CLI

AWS CLI

下列命令會刪除名為的記錄群組my-logs：

```
aws logs delete-log-group --log-group-name my-logs
```

- 如需 API 詳細資訊，請參閱AWS CLI 命令參考DeleteLogGroup中的。

JavaScript

適用於 JavaScript (v3) 的開發套件

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

```
import { DeleteLogGroupCommand } from "@aws-sdk/client-cloudwatch-logs";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new DeleteLogGroupCommand({
    // The name of the log group.
    logGroupName: process.env.CLOUDWATCH_LOGS_LOG_GROUP,
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

- 如需 API 詳細資訊，請參閱 AWS SDK for JavaScript API 參考DeleteLogGroup中的。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱搭配 AWS SDK 使用 CloudWatch 記錄檔。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭DeleteSubscriptionFilter配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用DeleteSubscriptionFilter。

C++

適用於 C++ 的 SDK

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

包括必需的檔案。

```
#include <aws/core/Aws.h>
#include <aws/core/utils/Outcome.h>
#include <aws/logs/CloudWatchLogsClient.h>
#include <aws/logs/model/DeleteSubscriptionFilterRequest.h>
#include <iostream>
```

刪除訂閱篩選條件。

```
        Aws::CloudWatchLogs::CloudWatchLogsClient cwl;
        Aws::CloudWatchLogs::Model::DeleteSubscriptionFilterRequest request;
        request.SetFilterName(filter_name);
        request.SetLogGroupName(log_group);

        auto outcome = cwl.DeleteSubscriptionFilter(request);
        if (!outcome.IsSuccess()) {
            std::cout << "Failed to delete CloudWatch log subscription filter "
                << filter_name << ": " << outcome.GetError().GetMessage() <<
```

```
                std::endl;
        } else {
            std::cout << "Successfully deleted CloudWatch logs subscription " <<
                "filter " << filter_name << std::endl;
        }
```

- 如需 API 詳細資訊，請參閱 AWS SDK for C++ API 參考DeleteSubscriptionFilter中的。

Java

適用於 Java 2.x 的 SDK

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

```
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsClient;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.DeleteSubscriptionFilterRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DeleteSubscriptionFilter {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <filter> <logGroup>

                Where:
```

```
                  filter - The name of the subscription filter (for example,
 MyFilter).
                  logGroup - The name of the log group. (for example, testgroup).
                """";

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String filter = args[0];
        String logGroup = args[1];
        CloudWatchLogsClient logs = CloudWatchLogsClient.builder()
                .build();

        deleteSubFilter(logs, filter, logGroup);
        logs.close();
    }

    public static void deleteSubFilter(CloudWatchLogsClient logs, String filter,
 String logGroup) {
        try {
            DeleteSubscriptionFilterRequest request =
 DeleteSubscriptionFilterRequest.builder()
                    .filterName(filter)
                    .logGroupName(logGroup)
                    .build();

            logs.deleteSubscriptionFilter(request);
            System.out.printf("Successfully deleted CloudWatch logs subscription
 filter %s", filter);

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考DeleteSubscriptionFilter中的。

JavaScript

適用於 JavaScript (v3) 的開發套件

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

```
import { DeleteSubscriptionFilterCommand } from "@aws-sdk/client-cloudwatch-
logs";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new DeleteSubscriptionFilterCommand({
    // The name of the filter.
    filterName: process.env.CLOUDWATCH_LOGS_FILTER_NAME,
    // The name of the log group.
    logGroupName: process.env.CLOUDWATCH_LOGS_LOG_GROUP,
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

- 如需 API 詳細資訊，請參閱 AWS SDK for JavaScript API 參考DeleteSubscriptionFilter中的。

適用於 JavaScript (v2) 的開發套件

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the CloudWatchLogs service object
var cwl = new AWS.CloudWatchLogs({ apiVersion: "2014-03-28" });

var params = {
  filterName: "FILTER",
  logGroupName: "LOG_GROUP",
};

cwl.deleteSubscriptionFilter(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- 如需詳細資訊，請參閱《AWS SDK for JavaScript 開發人員指南》。
- 如需 API 詳細資訊，請參閱 AWS SDK for JavaScript API 參考DeleteSubscriptionFilter中的。

Kotlin

適用於 Kotlin 的 SDK

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

```
suspend fun deleteSubFilter(
    filter: String?,
    logGroup: String?,
) {
```

```
    val request =
        DeleteSubscriptionFilterRequest {
            filterName = filter
            logGroupName = logGroup
        }

    CloudWatchLogsClient { region = "us-west-2" }.use { logs ->
        logs.deleteSubscriptionFilter(request)
        println("Successfully deleted CloudWatch logs subscription filter named
 $filter")
    }
}
```

- 有關 API 的詳細信息，請參閱 AWS SDK DeleteSubscriptionFilter中的 Kotlin API 參考。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱搭配 AWS SDK 使用 CloudWatch 記錄檔。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭配 AWS 開發套件或 CLI 使用**DescribeExportTasks**

下列程式碼範例會示範如何使用DescribeExportTasks。

.NET

AWS SDK for .NET

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

```
    using System;
    using System.Threading.Tasks;
    using Amazon.CloudWatchLogs;
    using Amazon.CloudWatchLogs.Model;

    /// <summary>
    /// Shows how to retrieve a list of information about Amazon CloudWatch
    /// Logs export tasks.
```

```
    /// </summary>
    public class DescribeExportTasks
    {
        public static async Task Main()
        {
            // This client object will be associated with the same AWS Region
            // as the default user on this system. If you need to use a
            // different AWS Region, pass it as a parameter to the client
            // constructor.
            var client = new AmazonCloudWatchLogsClient();

            var request = new DescribeExportTasksRequest
            {
                Limit = 5,
            };

            var response = new DescribeExportTasksResponse();

            do
            {
                response = await client.DescribeExportTasksAsync(request);
                response.ExportTasks.ForEach(t =>
                {
                    Console.WriteLine($"{t.TaskName} with ID: {t.TaskId} has
status: {t.Status}");
                });
            }
            while (response.NextToken is not null);
        }
    }
```

- 如需 API 詳細資訊，請參閱 AWS SDK for .NET API 參考DescribeExportTasks中的。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱搭配 AWS SDK 使用 CloudWatch 記錄檔。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭DescribeLogGroups配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用DescribeLogGroups。

.NET

AWS SDK for .NET

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

```
using System;
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;

/// <summary>
/// Retrieves information about existing Amazon CloudWatch Logs log groups
/// and displays the information on the console.
/// </summary>
public class DescribeLogGroups
{
    public static async Task Main()
    {
        // Creates a CloudWatch Logs client using the default
        // user. If you need to work with resources in another
        // AWS Region than the one defined for the default user,
        // pass the AWS Region as a parameter to the client constructor.
        var client = new AmazonCloudWatchLogsClient();

        bool done = false;
        string newToken = null;

        var request = new DescribeLogGroupsRequest
        {
            Limit = 5,
        };

        DescribeLogGroupsResponse response;

        do
        {
            if (newToken is not null)
```

```
                {
                    request.NextToken = newToken;
                }

                response = await client.DescribeLogGroupsAsync(request);

                response.LogGroups.ForEach(lg =>
                {
                    Console.WriteLine($"{lg.LogGroupName} is associated with the
key: {lg.KmsKeyId}.");
                    Console.WriteLine($"Created on:
{lg.CreationTime.Date.Date}");
                    Console.WriteLine($"Date for this group will be stored for:
{lg.RetentionInDays} days.\n");
                });

                if (response.NextToken is null)
                {
                    done = true;
                }
                else
                {
                    newToken = response.NextToken;
                }
            }
            while (!done);
        }
    }
```

- 如需 API 詳細資訊，請參閱 AWS SDK for .NET API 參考[DescribeLogGroups](https://docs.aws.amazon.com/goto/DotNetSDKV3/logs-2014-03-28/DescribeLogGroups)中的。

CLI

AWS CLI

下列命令描述名為的記錄群組`my-logs`：

```
aws logs describe-log-groups --log-group-name-prefix my-logs
```

輸出：

```
{
    "logGroups": [
        {
            "storedBytes": 0,
            "metricFilterCount": 0,
            "creationTime": 1433189500783,
            "logGroupName": "my-logs",
            "retentionInDays": 5,
            "arn": "arn:aws:logs:us-west-2:0123456789012:log-group:my-logs:*"
        }
    ]
}
```

- 如需 API 詳細資訊，請參閱AWS CLI 命令參考DescribeLogGroups中的。

JavaScript

適用於 JavaScript (v3) 的開發套件

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

```
import {
  paginateDescribeLogGroups,
  CloudWatchLogsClient,
} from "@aws-sdk/client-cloudwatch-logs";

const client = new CloudWatchLogsClient({});

export const main = async () => {
  const paginatedLogGroups = paginateDescribeLogGroups({ client }, {});
  const logGroups = [];

  for await (const page of paginatedLogGroups) {
    if (page.logGroups && page.logGroups.every((lg) => !!lg)) {
      logGroups.push(...page.logGroups);
    }
  }
```

```
  console.log(logGroups);
  return logGroups;
};
```

- 如需 API 詳細資訊，請參閱 AWS SDK for JavaScript API 參考DescribeLogGroups中的。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱搭配 AWS SDK 使用 CloudWatch 記錄檔。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭DescribeSubscriptionFilters配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用DescribeSubscriptionFilters。

C++

適用於 C++ 的 SDK

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

包括必需的檔案。

```
#include <aws/core/Aws.h>
#include <aws/core/utils/Outcome.h>
#include <aws/logs/CloudWatchLogsClient.h>
#include <aws/logs/model/DescribeSubscriptionFiltersRequest.h>
#include <aws/logs/model/DescribeSubscriptionFiltersResult.h>
#include <iostream>
#include <iomanip>
```

列出訂閱篩選條件。

```
        Aws::CloudWatchLogs::CloudWatchLogsClient cwl;
        Aws::CloudWatchLogs::Model::DescribeSubscriptionFiltersRequest request;
```

```
        request.SetLogGroupName(log_group);
        request.SetLimit(1);

        bool done = false;
        bool header = false;
        while (!done) {
            auto outcome = cwl.DescribeSubscriptionFilters(
                    request);
            if (!outcome.IsSuccess()) {
                std::cout << "Failed to describe CloudWatch subscription filters
"
                    << "for log group " << log_group << ": " <<
                    outcome.GetError().GetMessage() << std::endl;
                break;
            }

            if (!header) {
                std::cout << std::left << std::setw(32) << "Name" <<
                    std::setw(64) << "FilterPattern" << std::setw(64) <<
                    "DestinationArn" << std::endl;
                header = true;
            }

            const auto &filters = outcome.GetResult().GetSubscriptionFilters();
            for (const auto &filter : filters) {
                std::cout << std::left << std::setw(32) <<
                    filter.GetFilterName() << std::setw(64) <<
                    filter.GetFilterPattern() << std::setw(64) <<
                    filter.GetDestinationArn() << std::endl;
            }

            const auto &next_token = outcome.GetResult().GetNextToken();
            request.SetNextToken(next_token);
            done = next_token.empty();
        }
```

- 如需 API 詳細資訊，請參閱 AWS SDK for C++ API 參考DescribeSubscriptionFilters中的。

Java

適用於 Java 2.x 的 SDK

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

```
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsClient;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.DescribeSubscriptionFiltersRequest;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.DescribeSubscriptionFiltersResponse
import software.amazon.awssdk.services.cloudwatchlogs.model.SubscriptionFilter;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DescribeSubscriptionFilters {
    public static void main(String[] args) {

        final String usage = """

                Usage:
                  <logGroup>

                Where:
                  logGroup - A log group name (for example, myloggroup).
                """;

        if (args.length != 1) {
            System.out.println(usage);
```

```
            System.exit(1);
        }

        String logGroup = args[0];
        CloudWatchLogsClient logs = CloudWatchLogsClient.builder()
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        describeFilters(logs, logGroup);
        logs.close();
    }

    public static void describeFilters(CloudWatchLogsClient logs, String
logGroup) {
        try {
            boolean done = false;
            String newToken = null;

            while (!done) {
                DescribeSubscriptionFiltersResponse response;
                if (newToken == null) {
                    DescribeSubscriptionFiltersRequest request =
DescribeSubscriptionFiltersRequest.builder()
                            .logGroupName(logGroup)
                            .limit(1).build();

                    response = logs.describeSubscriptionFilters(request);
                } else {
                    DescribeSubscriptionFiltersRequest request =
DescribeSubscriptionFiltersRequest.builder()
                            .nextToken(newToken)
                            .logGroupName(logGroup)
                            .limit(1).build();
                    response = logs.describeSubscriptionFilters(request);
                }

                for (SubscriptionFilter filter : response.subscriptionFilters())
{
                    System.out.printf("Retrieved filter with name %s, " +
"pattern %s " + "and destination arn %s",
                            filter.filterName(),
                            filter.filterPattern(),
                            filter.destinationArn());
                }
```

```
                if (response.nextToken() == null) {
                    done = true;
                } else {
                    newToken = response.nextToken();
                }
            }

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        System.out.printf("Done");
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考DescribeSubscriptionFilters中的。

JavaScript

適用於 JavaScript (v3) 的開發套件

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

```
import { DescribeSubscriptionFiltersCommand } from "@aws-sdk/client-cloudwatch-
logs";
import { client } from "../libs/client.js";

const run = async () => {
  // This will return a list of all subscription filters in your account
  // matching the log group name.
  const command = new DescribeSubscriptionFiltersCommand({
    logGroupName: process.env.CLOUDWATCH_LOGS_LOG_GROUP,
    limit: 1,
  });
```

```
  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

- 如需 API 詳細資訊，請參閱 AWS SDK for JavaScript API 參考DescribeSubscriptionFilters中的。

適用於 JavaScript (v2) 的開發套件

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the CloudWatchLogs service object
var cwl = new AWS.CloudWatchLogs({ apiVersion: "2014-03-28" });

var params = {
  logGroupName: "GROUP_NAME",
  limit: 5,
};

cwl.describeSubscriptionFilters(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data.subscriptionFilters);
  }
});
```

- 如需詳細資訊，請參閱《AWS SDK for JavaScript 開發人員指南》。
- 如需 API 詳細資訊，請參閱 AWS SDK for JavaScript API 參考DescribeSubscriptionFilters中的。

Kotlin

適用於 Kotlin 的 SDK

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

```
suspend fun describeFilters(logGroup: String) {
    val request =
        DescribeSubscriptionFiltersRequest {
            logGroupName = logGroup
            limit = 1
        }

    CloudWatchLogsClient { region = "us-west-2" }.use { cwlClient ->
        val response = cwlClient.describeSubscriptionFilters(request)
        response.subscriptionFilters?.forEach { filter ->
            println("Retrieved filter with name  ${filter.filterName} pattern
 ${filter.filterPattern} and destination ${filter.destinationArn}")
        }
    }
}
```

- 有關 API 的詳細信息，請參閱 AWS SDK DescribeSubscriptionFilters中的 Kotlin API 參考。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱搭配 AWS SDK 使用 CloudWatch 記錄檔。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭GetQueryResults配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用GetQueryResults。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- 執行大型查詢

JavaScript

適用於 JavaScript (v3) 的開發套件

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

```
/**
 * Simple wrapper for the GetQueryResultsCommand.
 * @param {string} queryId
 */
_getQueryResults(queryId) {
  return this.client.send(new GetQueryResultsCommand({ queryId }));
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for JavaScript API 參考GetQueryResults中的。

Python

適用於 Python (Boto3) 的 SDK

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

```
    def _wait_for_query_results(self, client, query_id):
        """
        Waits for the query to complete and retrieves the results.

        :param query_id: The ID of the initiated query.
        :type query_id: str
        :return: A list containing the results of the query.
        :rtype: list
        """
        while True:
            time.sleep(1)
            results = client.get_query_results(queryId=query_id)
            if results["status"] in [
                "Complete",
                "Failed",
                "Cancelled",
                "Timeout",
                "Unknown",
            ]:
                return results.get("results", [])
```

- 如需 API 的詳細資訊，請參閱AWS 開發套件GetQueryResults中的 Python (博托 3) API 參考。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱搭配 AWS SDK 使用 CloudWatch 記錄檔。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭PutSubscriptionFilter配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用PutSubscriptionFilter。

C++

適用於 C++ 的 SDK

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

包括必需的檔案。

```
#include <aws/core/Aws.h>
#include <aws/logs/CloudWatchLogsClient.h>
#include <aws/logs/model/PutSubscriptionFilterRequest.h>
#include <aws/core/utils/Outcome.h>
#include <iostream>
```

建立訂閱篩選條件。

```
        Aws::CloudWatchLogs::CloudWatchLogsClient cwl;
        Aws::CloudWatchLogs::Model::PutSubscriptionFilterRequest request;
        request.SetFilterName(filter_name);
        request.SetFilterPattern(filter_pattern);
        request.SetLogGroupName(log_group);
        request.SetDestinationArn(dest_arn);
        auto outcome = cwl.PutSubscriptionFilter(request);
        if (!outcome.IsSuccess())
        {
            std::cout << "Failed to create CloudWatch logs subscription filter "
                << filter_name << ": " << outcome.GetError().GetMessage() <<
                std::endl;
        }
        else
        {
            std::cout << "Successfully created CloudWatch logs subscription " <<
                "filter " << filter_name << std::endl;
        }
```

- 如需 API 詳細資訊，請參閱 AWS SDK for C++ API 參考PutSubscriptionFilter中的。

Java

適用於 Java 2.x 的 SDK

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsClient;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.CloudWatchLogsException;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.PutSubscriptionFilterRequest;

/**
 * Before running this code example, you need to grant permission to CloudWatch
 * Logs the right to execute your Lambda function.
 * To perform this task, you can use this CLI command:
 *
 * aws lambda add-permission --function-name "lamda1" --statement-id "lamda1"
 * --principal "logs.us-west-2.amazonaws.com" --action "lambda:InvokeFunction"
 * --source-arn "arn:aws:logs:us-west-2:111111111111:log-group:testgroup:*"
 * --source-account "111111111111"
 *
 * Make sure you replace the function name with your function name and replace
 * '111111111111' with your account details.
 * For more information, see "Subscription Filters with AWS Lambda" in the
 * Amazon CloudWatch Logs Guide.
 *
 *
 * Also, before running this Java V2 code example,set up your development
 * environment,including your credentials.
 *
 * For more information,see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 *
 */
```

```
public class PutSubscriptionFilter {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <filter> <pattern> <logGroup> <functionArn>\s

                Where:
                  filter - A filter name (for example, myfilter).
                  pattern - A filter pattern (for example, ERROR).
                  logGroup - A log group name (testgroup).
                  functionArn - An AWS Lambda function ARN (for example,
 arn:aws:lambda:us-west-2:111111111111:function:lambda1) .
                """;

        if (args.length != 4) {
            System.out.println(usage);
            System.exit(1);
        }

        String filter = args[0];
        String pattern = args[1];
        String logGroup = args[2];
        String functionArn = args[3];
        Region region = Region.US_WEST_2;
        CloudWatchLogsClient cwl = CloudWatchLogsClient.builder()
                .region(region)
                .build();

        putSubFilters(cwl, filter, pattern, logGroup, functionArn);
        cwl.close();
    }

    public static void putSubFilters(CloudWatchLogsClient cwl,
            String filter,
            String pattern,
            String logGroup,
            String functionArn) {

        try {
            PutSubscriptionFilterRequest request =
 PutSubscriptionFilterRequest.builder()
                    .filterName(filter)
```

```
                    .filterPattern(pattern)
                    .logGroupName(logGroup)
                    .destinationArn(functionArn)
                    .build();

            cwl.putSubscriptionFilter(request);
            System.out.printf(
                    "Successfully created CloudWatch logs subscription filter
 %s",
                    filter);

        } catch (CloudWatchLogsException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考PutSubscriptionFilter中的。

JavaScript

適用於 JavaScript (v3) 的開發套件

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

```
import { PutSubscriptionFilterCommand } from "@aws-sdk/client-cloudwatch-logs";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new PutSubscriptionFilterCommand({
    // An ARN of a same-account Kinesis stream, Kinesis Firehose
    // delivery stream, or Lambda function.
    // https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/
SubscriptionFilters.html
    destinationArn: process.env.CLOUDWATCH_LOGS_DESTINATION_ARN,
```

```
    // A name for the filter.
    filterName: process.env.CLOUDWATCH_LOGS_FILTER_NAME,

    // A filter pattern for subscribing to a filtered stream of log events.
    // https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/
FilterAndPatternSyntax.html
    filterPattern: process.env.CLOUDWATCH_LOGS_FILTER_PATTERN,

    // The name of the log group. Messages in this group matching the filter
 pattern
    // will be sent to the destination ARN.
    logGroupName: process.env.CLOUDWATCH_LOGS_LOG_GROUP,
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

- 如需 API 詳細資訊，請參閱 AWS SDK for JavaScript API 參考PutSubscriptionFilter中的。

適用於 JavaScript (v2) 的開發套件

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the CloudWatchLogs service object
var cwl = new AWS.CloudWatchLogs({ apiVersion: "2014-03-28" });

var params = {
```

```
  destinationArn: "LAMBDA_FUNCTION_ARN",
  filterName: "FILTER_NAME",
  filterPattern: "ERROR",
  logGroupName: "LOG_GROUP",
};

cwl.putSubscriptionFilter(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- 如需詳細資訊，請參閱《AWS SDK for JavaScript 開發人員指南》。
- 如需 API 詳細資訊，請參閱 AWS SDK for JavaScript API 參考PutSubscriptionFilter中的。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱搭配 AWS SDK 使用 CloudWatch 記錄檔。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭StartLiveTail配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用StartLiveTail。

.NET

AWS SDK for .NET

包括必需的檔案。

```
using Amazon;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;
```

啟動「即時尾巴」工作階段。

```
            var client = new AmazonCloudWatchLogsClient();
            var request = new StartLiveTailRequest
            {
```

```
                LogGroupIdentifiers = logGroupIdentifiers,
                LogStreamNames = logStreamNames,
                LogEventFilterPattern = filterPattern,
            };

            var response = await client.StartLiveTailAsync(request);

            // Catch if request fails
            if (response.HttpStatusCode != System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine("Failed to start live tail session");
                return;
            }
```

您可以使用兩種方式處理來自 Live Tail 工作階段的事件：

```
            /* Method 1
            * 1). Asynchronously loop through the event stream
            * 2). Set a timer to dispose the stream and stop the Live Tail
session at the end.
            */
            var eventStream = response.ResponseStream;
            var task = Task.Run(() =>
            {
                foreach (var item in eventStream)
                {
                    if (item is LiveTailSessionUpdate liveTailSessionUpdate)
                    {
                        foreach (var sessionResult in
liveTailSessionUpdate.SessionResults)
                        {
                            Console.WriteLine("Message : {0}",
sessionResult.Message);
                        }
                    }
                    if (item is LiveTailSessionStart)
                    {
                        Console.WriteLine("Live Tail session started");
                    }
                    // On-stream exceptions are processed here
                    if (item is CloudWatchLogsEventStreamException)
                    {
```

```
                    Console.WriteLine($"ERROR: {item}");
                }
            }
        });
        // Close the stream to stop the session after a timeout
        if (!task.Wait(TimeSpan.FromSeconds(10))){
            eventStream.Dispose();
            Console.WriteLine("End of line");
        }
```

```
        /* Method 2
        * 1). Add event handlers to each event variable
        * 2). Start processing the stream and wait for a timeout using
AutoResetEvent
        */
        AutoResetEvent endEvent = new AutoResetEvent(false);
        var eventStream = response.ResponseStream;
        using (eventStream) // automatically disposes the stream to stop the
session after execution finishes
        {
            eventStream.SessionStartReceived += (sender, e) =>
            {
                Console.WriteLine("LiveTail session started");
            };
            eventStream.SessionUpdateReceived += (sender, e) =>
            {
                foreach (LiveTailSessionLogEvent logEvent in
e.EventStreamEvent.SessionResults){
                    Console.WriteLine("Message: {0}", logEvent.Message);
                }
            };
            // On-stream exceptions are captured here
            eventStream.ExceptionReceived += (sender, e) =>
            {
                Console.WriteLine($"ERROR:
{e.EventStreamException.Message}");
            };

            eventStream.StartProcessing();
            // Stream events for this amount of time.
            endEvent.WaitOne(TimeSpan.FromSeconds(10));
            Console.WriteLine("End of line");
```

```
            }
```

- 如需 API 詳細資訊，請參閱 AWS SDK for .NET API 參考StartLiveTail中的。

Go

SDK for Go V2

包括必需的檔案。

```
import (
 "context"
 "log"
 "time"

 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/service/cloudwatchlogs"
 "github.com/aws/aws-sdk-go-v2/service/cloudwatchlogs/types"
)
```

處理來自即時尾巴工作階段的事件。

```
func handleEventStreamAsync(stream *cloudwatchlogs.StartLiveTailEventStream) {
 eventsChan := stream.Events()
 for {
  event := <-eventsChan
  switch e := event.(type) {
  case *types.StartLiveTailResponseStreamMemberSessionStart:
   log.Println("Received SessionStart event")
  case *types.StartLiveTailResponseStreamMemberSessionUpdate:
   for _, logEvent := range e.Value.SessionResults {
    log.Println(*logEvent.Message)
   }
  default:
   // Handle on-stream exceptions
   if err := stream.Err(); err != nil {
    log.Fatalf("Error occured during streaming: %v", err)
   } else if event == nil {
    log.Println("Stream is Closed")
    return
```

```
   } else {
    log.Fatalf("Unknown event type: %T", e)
   }
  }
 }
}
```

啟動「即時尾巴」工作階段。

```
cfg, err := config.LoadDefaultConfig(context.TODO())
if err != nil {
 panic("configuration error, " + err.Error())
}
client := cloudwatchlogs.NewFromConfig(cfg)

request := &cloudwatchlogs.StartLiveTailInput{
 LogGroupIdentifiers:   logGroupIdentifiers,
 LogStreamNames:        logStreamNames,
 LogEventFilterPattern: logEventFilterPattern,
}

response, err := client.StartLiveTail(context.TODO(), request)
// Handle pre-stream Exceptions
if err != nil {
 log.Fatalf("Failed to start streaming: %v", err)
}

// Start a Goroutine to handle events over stream
stream := response.GetStream()
go handleEventStreamAsync(stream)
```

經過一段時間後，停止「即時尾端」工作階段。

```
// Close the stream (which ends the session) after a timeout
time.Sleep(10 * time.Second)
stream.Close()
log.Println("Event stream closed")
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Go API 參考StartLiveTail中的。

Java

適用於 Java 2.x 的 SDK

包括必需的檔案。

```
import io.reactivex.FlowableSubscriber;
import io.reactivex.annotations.NonNull;
import org.reactivestreams.Subscription;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsAsyncClient;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.LiveTailSessionLogEvent;
import software.amazon.awssdk.services.cloudwatchlogs.model.LiveTailSessionStart;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.LiveTailSessionUpdate;
import software.amazon.awssdk.services.cloudwatchlogs.model.StartLiveTailRequest;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.StartLiveTailResponseHandler;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.CloudWatchLogsException;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.StartLiveTailResponseStream;

import java.util.Date;
import java.util.List;
import java.util.concurrent.atomic.AtomicReference;
```

處理來自即時尾巴工作階段的事件。

```
    private static StartLiveTailResponseHandler
 getStartLiveTailResponseStreamHandler(
            AtomicReference<Subscription> subscriptionAtomicReference) {
        return StartLiveTailResponseHandler.builder()
            .onResponse(r -> System.out.println("Received initial response"))
            .onError(throwable -> {
                CloudWatchLogsException e = (CloudWatchLogsException)
 throwable.getCause();
                System.err.println(e.awsErrorDetails().errorMessage());
                System.exit(1);
            })
            .subscriber(() -> new FlowableSubscriber<>() {
```

```
                @Override
                public void onSubscribe(@NonNull Subscription s) {
                    subscriptionAtomicReference.set(s);
                    s.request(Long.MAX_VALUE);
                }

                @Override
                public void onNext(StartLiveTailResponseStream event) {
                    if (event instanceof LiveTailSessionStart) {
                        LiveTailSessionStart sessionStart =
(LiveTailSessionStart) event;
                        System.out.println(sessionStart);
                    } else if (event instanceof LiveTailSessionUpdate) {
                        LiveTailSessionUpdate sessionUpdate =
(LiveTailSessionUpdate) event;
                        List<LiveTailSessionLogEvent> logEvents =
sessionUpdate.sessionResults();
                        logEvents.forEach(e -> {
                            long timestamp = e.timestamp();
                            Date date = new Date(timestamp);
                            System.out.println("[" + date + "] " + e.message());
                        });
                    } else {
                        throw CloudWatchLogsException.builder().message("Unknown
event type").build();
                    }
                }

                @Override
                public void onError(Throwable throwable) {
                    System.out.println(throwable.getMessage());
                    System.exit(1);
                }

                @Override
                public void onComplete() {
                    System.out.println("Completed Streaming Session");
                }
            })
            .build();
    }
```

啟動「即時尾巴」工作階段。

```
        CloudWatchLogsAsyncClient cloudWatchLogsAsyncClient =
                CloudWatchLogsAsyncClient.builder()
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();

        StartLiveTailRequest request =
                StartLiveTailRequest.builder()
                    .logGroupIdentifiers(logGroupIdentifiers)
                    .logStreamNames(logStreamNames)
                    .logEventFilterPattern(logEventFilterPattern)
                    .build();

        /* Create a reference to store the subscription */
        final AtomicReference<Subscription> subscriptionAtomicReference = new
AtomicReference<>(null);

        cloudWatchLogsAsyncClient.startLiveTail(request,
getStartLiveTailResponseStreamHandler(subscriptionAtomicReference));
```

經過一段時間後，停止「即時尾端」工作階段。

```
        /* Set a timeout for the session and cancel the subscription. This will:
         * 1). Close the stream
         * 2). Stop the Live Tail session
         */
        try {
            Thread.sleep(10000);
        } catch (InterruptedException e) {
            throw new RuntimeException(e);
        }
        if (subscriptionAtomicReference.get() != null) {
            subscriptionAtomicReference.get().cancel();
            System.out.println("Subscription to stream closed");
        }
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考StartLiveTail中的。

JavaScript

適用於 JavaScript (v3) 的開發套件

包括必需的檔案。

```
import { CloudWatchLogsClient, StartLiveTailCommand } from "@aws-sdk/client-
cloudwatch-logs";
```

處理來自即時尾巴工作階段的事件。

```
async function handleResponseAsync(response) {
    try {
      for await (const event of response.responseStream) {
        if (event.sessionStart !== undefined) {
          console.log(event.sessionStart);
        } else if (event.sessionUpdate !== undefined) {
          for (const logEvent of event.sessionUpdate.sessionResults) {
            const timestamp = logEvent.timestamp;
            const date = new Date(timestamp);
            console.log("[" + date + "] " + logEvent.message);
          }
        } else {
            console.error("Unknown event type");
        }
      }
    } catch (err) {
        // On-stream exceptions are captured here
        console.error(err)
    }
}
```

啟動「即時尾巴」工作階段。

```
    const client = new CloudWatchLogsClient();

    const command = new StartLiveTailCommand({
        logGroupIdentifiers: logGroupIdentifiers,
        logStreamNames: logStreamNames,
        logEventFilterPattern: filterPattern
    });
```

```
    try{
        const response = await client.send(command);
        handleResponseAsync(response);
    } catch (err){
        // Pre-stream exceptions are captured here
        console.log(err);
    }
```

經過一段時間後，停止「即時尾端」工作階段。

```
    /* Set a timeout to close the client. This will stop the Live Tail session.
 */
    setTimeout(function() {
        console.log("Client timeout");
        client.destroy();
      }, 10000);
```

- 如需 API 詳細資訊，請參閱 AWS SDK for JavaScript API 參考StartLiveTail中的。

Kotlin

適用於 Kotlin 的 SDK

包括必需的檔案。

```
import aws.sdk.kotlin.services.cloudwatchlogs.CloudWatchLogsClient
import aws.sdk.kotlin.services.cloudwatchlogs.model.StartLiveTailRequest
import aws.sdk.kotlin.services.cloudwatchlogs.model.StartLiveTailResponseStream
import kotlinx.coroutines.flow.takeWhile
```

啟動「即時尾巴」工作階段。

```
    val client = CloudWatchLogsClient.fromEnvironment()

    val request = StartLiveTailRequest {
        logGroupIdentifiers = logGroupIdentifiersVal
        logStreamNames = logStreamNamesVal
        logEventFilterPattern = logEventFilterPatternVal
    }
```

```
    val startTime = System.currentTimeMillis()

    try {
        client.startLiveTail(request) { response ->
            val stream = response.responseStream
            if (stream != null) {
                /* Set a timeout to unsubcribe from the flow. This will:
                * 1). Close the stream
                * 2). Stop the Live Tail session
                */
                stream.takeWhile { System.currentTimeMillis() - startTime <
10000 }.collect { value ->
                    if (value is StartLiveTailResponseStream.SessionStart) {
                        println(value.asSessionStart())
                    } else if (value is
StartLiveTailResponseStream.SessionUpdate) {
                        for (e in value.asSessionUpdate().sessionResults!!) {
                            println(e)
                        }
                    } else {
                        throw IllegalArgumentException("Unknown event type")
                    }
                }
            } else {
                throw IllegalArgumentException("No response stream")
            }
        }
    } catch (e: Exception) {
        println("Exception occurred during StartLiveTail: $e")
        System.exit(1)
    }
```

- 有關 API 的詳細信息，請參閱 AWS SDK StartLiveTail中的 Kotlin API 參考。

Python

適用於 Python (Boto3) 的 SDK

包括必需的檔案。

```
import boto3
```

```
import time
from datetime import datetime
```

啟動「即時尾巴」工作階段。

```
    # Initialize the client
    client = boto3.client('logs')

    start_time = time.time()

    try:
        response = client.start_live_tail(
            logGroupIdentifiers=log_group_identifiers,
            logStreamNames=log_streams,
            logEventFilterPattern=filter_pattern
        )
        event_stream = response['responseStream']
        # Handle the events streamed back in the response
        for event in event_stream:
            # Set a timeout to close the stream.
            # This will end the Live Tail session.
            if (time.time() - start_time >= 10):
                event_stream.close()
                break
            # Handle when session is started
            if 'sessionStart' in event:
                session_start_event = event['sessionStart']
                print(session_start_event)
            # Handle when log event is given in a session update
            elif 'sessionUpdate' in event:
                log_events = event['sessionUpdate']['sessionResults']
                for log_event in log_events:
                    print('[{date}]
{log}'.format(date=datetime.fromtimestamp(log_event['timestamp']/1000),log=log_event['me
            else:
                # On-stream exceptions are captured here
                raise RuntimeError(str(event))
    except Exception as e:
        print(e)
```

- 如需 API 的詳細資訊，請參閱AWS 開發套件StartLiveTail中的 Python (博托 3) API 參考。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱搭配 AWS SDK 使用 CloudWatch 記錄檔。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

搭StartQuery配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用StartQuery。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- 執行大型查詢

JavaScript

適用於 JavaScript (v3) 的開發套件

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

```
/**
 * Wrapper for the StartQueryCommand. Uses a static query string
 * for consistency.
 * @param {[Date, Date]} dateRange
 * @param {number} maxLogs
 * @returns {Promise<{ queryId: string }>}
 */
async _startQuery([startDate, endDate], maxLogs = 10000) {
  try {
    return await this.client.send(
      new StartQueryCommand({
        logGroupNames: this.logGroupNames,
        queryString: "fields @timestamp, @message | sort @timestamp asc",
        startTime: startDate.valueOf(),
        endTime: endDate.valueOf(),
        limit: maxLogs,
      }),
    );
  } catch (err) {
```

```
      /** @type {string} */
      const message = err.message;
      if (message.startsWith("Query's end date and time")) {
        // This error indicates that the query's start or end date occur
        // before the log group was created.
        throw new DateOutOfBoundsError(message);
      }

      throw err;
    }
  }
```

- 如需 API 詳細資訊，請參閱 AWS SDK for JavaScript API 參考StartQuery中的。

Python

適用於 Python (Boto3) 的 SDK

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

```
    def perform_query(self, date_range):
        """
        Performs the actual CloudWatch log query.

        :param date_range: A tuple representing the start and end datetime for
the query.
        :type date_range: tuple
        :return: A list containing the query results.
        :rtype: list
        """
        client = boto3.client("logs")
        try:
            try:
                start_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[0])
                )
```

```
                end_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[1])
                )
                response = client.start_query(
                    logGroupName=self.log_groups,
                    startTime=start_time,
                    endTime=end_time,
                    queryString="fields @timestamp, @message | sort @timestamp
asc",
                    limit=self.limit,
                )
                query_id = response["queryId"]
            except client.exceptions.ResourceNotFoundException as e:
                raise DateOutOfBoundsError(f"Resource not found: {e}")
            while True:
                time.sleep(1)
                results = client.get_query_results(queryId=query_id)
                if results["status"] in [
                    "Complete",
                    "Failed",
                    "Cancelled",
                    "Timeout",
                    "Unknown",
                ]:
                    return results.get("results", [])
        except DateOutOfBoundsError:
            return []

    def _initiate_query(self, client, date_range, max_logs):
        """
        Initiates the CloudWatch logs query.

        :param date_range: A tuple representing the start and end datetime for
the query.
        :type date_range: tuple
        :param max_logs: The maximum number of logs to retrieve.
        :type max_logs: int
        :return: The query ID as a string.
        :rtype: str
        """
        try:
            start_time = round(
```

```
self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[0])
            )
            end_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[1])
            )
            response = client.start_query(
                logGroupName=self.log_groups,
                startTime=start_time,
                endTime=end_time,
                queryString="fields @timestamp, @message | sort @timestamp asc",
                limit=max_logs,
            )
            return response["queryId"]
        except client.exceptions.ResourceNotFoundException as e:
            raise DateOutOfBoundsError(f"Resource not found: {e}")
```

- 如需 API 的詳細資訊，請參閱AWS 開發套件StartQuery中的 Python (博托 3) API 參考。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱搭配 AWS SDK 使用 CloudWatch 記錄檔。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

使用 AWS SDK 的 CloudWatch 記錄檔案例

下列程式碼範例說明如何在使用 AWS SDK 的 CloudWatch 記錄中實作常見案例。這些案例說明如何透過在 CloudWatch Logs 中呼叫多個函數來完成特定工作。每個案例都包含一個連結 GitHub，您可以在其中找到如何設定和執行程式碼的指示。

範例

- 使用 CloudWatch 記錄檔執行大型查詢

使用 CloudWatch 記錄檔執行大型查詢

下列程式碼範例顯示如何使用記 CloudWatch 錄來查詢 10,000 筆以上的記錄。

JavaScript

適用於 JavaScript (v3) 的開發套件

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

這是入口點。

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import { CloudWatchLogsClient } from "@aws-sdk/client-cloudwatch-logs";
import { CloudWatchQuery } from "./cloud-watch-query.js";

console.log("Starting a recursive query...");

if (!process.env.QUERY_START_DATE || !process.env.QUERY_END_DATE) {
  throw new Error(
    "QUERY_START_DATE and QUERY_END_DATE environment variables are required.",
  );
}

const cloudWatchQuery = new CloudWatchQuery(new CloudWatchLogsClient({}), {
  logGroupNames: ["/workflows/cloudwatch-logs/large-query"],
  dateRange: [
    new Date(parseInt(process.env.QUERY_START_DATE)),
    new Date(parseInt(process.env.QUERY_END_DATE)),
  ],
});

await cloudWatchQuery.run();

console.log(
  `Queries finished in ${cloudWatchQuery.secondsElapsed} seconds.\nTotal logs
 found: ${cloudWatchQuery.results.length}`,
);
```

這是一個如果必要的話，將查詢拆分為多個步驟的類。

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import {
  StartQueryCommand,
  GetQueryResultsCommand,
} from "@aws-sdk/client-cloudwatch-logs";
import { splitDateRange } from "@aws-doc-sdk-examples/lib/utils/util-date.js";
import { retry } from "@aws-doc-sdk-examples/lib/utils/util-timers.js";

class DateOutOfBoundsError extends Error {}

export class CloudWatchQuery {
  /**
   * Run a query for all CloudWatch Logs within a certain date range.
   * CloudWatch logs return a max of 10,000 results. This class
   * performs a binary search across all of the logs in the provided
   * date range if a query returns the maximum number of results.
   *
   * @param {import('@aws-sdk/client-cloudwatch-logs').CloudWatchLogsClient}
 client
   * @param {{ logGroupNames: string[], dateRange: [Date, Date], queryConfig:
 { limit: number } }} config
   */
  constructor(client, { logGroupNames, dateRange, queryConfig }) {
    this.client = client;
    /**
     * All log groups are queried.
     */
    this.logGroupNames = logGroupNames;

    /**
     * The inclusive date range that is queried.
     */
    this.dateRange = dateRange;

    /**
     * CloudWatch Logs never returns more than 10,000 logs.
     */
    this.limit = queryConfig?.limit ?? 10000;

    /**
     * @type {import("@aws-sdk/client-cloudwatch-logs").ResultField[][]}
     */
```

```
    this.results = [];
  }

  /**
   * Run the query.
   */
  async run() {
    this.secondsElapsed = 0;
    const start = new Date();
    this.results = await this._largeQuery(this.dateRange);
    const end = new Date();
    this.secondsElapsed = (end - start) / 1000;
    return this.results;
  }

  /**
   * Recursively query for logs.
   * @param {[Date, Date]} dateRange
   * @returns {Promise<import("@aws-sdk/client-cloudwatch-logs").ResultField[]
[]>}
   */
  async _largeQuery(dateRange) {
    const logs = await this._query(dateRange, this.limit);

    console.log(
      `Query date range: ${dateRange
        .map((d) => d.toISOString())
        .join(" to ")}. Found ${logs.length} logs.`,
    );

    if (logs.length < this.limit) {
      return logs;
    }

    const lastLogDate = this._getLastLogDate(logs);
    const offsetLastLogDate = new Date(lastLogDate);
    offsetLastLogDate.setMilliseconds(lastLogDate.getMilliseconds() + 1);
    const subDateRange = [offsetLastLogDate, dateRange[1]];
    const [r1, r2] = splitDateRange(subDateRange);
    const results = await Promise.all([
      this._largeQuery(r1),
      this._largeQuery(r2),
    ]);
    return [logs, ...results].flat();
```

```
}

/**
 * Find the most recent log in a list of logs.
 * @param {import("@aws-sdk/client-cloudwatch-logs").ResultField[][]} logs
 */
_getLastLogDate(logs) {
  const timestamps = logs
    .map(
      (log) =>
        log.find((fieldMeta) => fieldMeta.field === "@timestamp")?.value,
    )
    .filter((t) => !!t)
    .map((t) => `${t}Z`)
    .sort();

  if (!timestamps.length) {
    throw new Error("No timestamp found in logs.");
  }

  return new Date(timestamps[timestamps.length - 1]);
}

// snippet-start:[javascript.v3.cloudwatch-logs.actions.GetQueryResults]
/**
 * Simple wrapper for the GetQueryResultsCommand.
 * @param {string} queryId
 */
_getQueryResults(queryId) {
  return this.client.send(new GetQueryResultsCommand({ queryId }));
}
// snippet-end:[javascript.v3.cloudwatch-logs.actions.GetQueryResults]

/**
 * Starts a query and waits for it to complete.
 * @param {[Date, Date]} dateRange
 * @param {number} maxLogs
 */
async _query(dateRange, maxLogs) {
  try {
    const { queryId } = await this._startQuery(dateRange, maxLogs);
    const { results } = await this._waitUntilQueryDone(queryId);
    return results ?? [];
  } catch (err) {
```

```
      /**
       * This error is thrown when StartQuery returns an error indicating
       * that the query's start or end date occur before the log group was
       * created.
       */
      if (err instanceof DateOutOfBoundsError) {
        return [];
      } else {
        throw err;
      }
    }
  }

  // snippet-start:[javascript.v3.cloudwatch-logs.actions.StartQuery]
  /**
   * Wrapper for the StartQueryCommand. Uses a static query string
   * for consistency.
   * @param {[Date, Date]} dateRange
   * @param {number} maxLogs
   * @returns {Promise<{ queryId: string }>}
   */
  async _startQuery([startDate, endDate], maxLogs = 10000) {
    try {
      return await this.client.send(
        new StartQueryCommand({
          logGroupNames: this.logGroupNames,
          queryString: "fields @timestamp, @message | sort @timestamp asc",
          startTime: startDate.valueOf(),
          endTime: endDate.valueOf(),
          limit: maxLogs,
        }),
      );
    } catch (err) {
      /** @type {string} */
      const message = err.message;
      if (message.startsWith("Query's end date and time")) {
        // This error indicates that the query's start or end date occur
        // before the log group was created.
        throw new DateOutOfBoundsError(message);
      }

      throw err;
    }
  }
```

```
  // snippet-end:[javascript.v3.cloudwatch-logs.actions.StartQuery]

  /**
   * Call GetQueryResultsCommand until the query is done.
   * @param {string} queryId
   */
  _waitUntilQueryDone(queryId) {
    const getResults = async () => {
      const results = await this._getQueryResults(queryId);
      const queryDone = [
        "Complete",
        "Failed",
        "Cancelled",
        "Timeout",
        "Unknown",
      ].includes(results.status);

      return { queryDone, results };
    };

    return retry(
      { intervalInMs: 1000, maxRetries: 60, quiet: true },
      async () => {
        const { queryDone, results } = await getResults();
        if (!queryDone) {
          throw new Error("Query not done.");
        }

        return results;
      },
    );
  }
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for JavaScript API 參考》中的下列主題。
  - GetQueryResults
  - StartQuery

Python

適用於 Python (Boto3) 的 SDK

> **Note**
>
> 還有更多關於 GitHub。尋找完整範例，並了解如何在AWS 設定和執行程式碼範例儲存庫。

此檔案會叫用範例模組來管理超過 10,000 個結果的 CloudWatch 查詢。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
import logging
import os
import sys

import boto3
from botocore.config import Config

from cloudwatch_query import CloudWatchQuery
from date_utilities import DateUtilities

# Configure logging at the module level.
logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s - %(levelname)s - %(filename)s:%(lineno)d - %(message)s",
)


class CloudWatchLogsQueryRunner:
    def __init__(self):
        """
        Initializes the CloudWatchLogsQueryRunner class by setting up date
 utilities
        and creating a CloudWatch Logs client with retry configuration.
        """
        self.date_utilities = DateUtilities()
        self.cloudwatch_logs_client = self.create_cloudwatch_logs_client()

    def create_cloudwatch_logs_client(self):
        """
```

```
        Creates and returns a CloudWatch Logs client with a specified retry
configuration.

        :return: A CloudWatch Logs client instance.
        :rtype: boto3.client
        """
        try:
            return boto3.client("logs", config=Config(retries={"max_attempts":
10}))
        except Exception as e:
            logging.error(f"Failed to create CloudWatch Logs client: {e}")
            sys.exit(1)

    def fetch_environment_variables(self):
        """
        Fetches and validates required environment variables for query start and
end dates.

        :return: Tuple of query start date and end date as integers.
        :rtype: tuple
        :raises SystemExit: If required environment variables are missing or
invalid.
        """
        try:
            query_start_date = int(os.environ["QUERY_START_DATE"])
            query_end_date = int(os.environ["QUERY_END_DATE"])
        except KeyError:
            logging.error(
                "Both QUERY_START_DATE and QUERY_END_DATE environment variables
are required."
            )
            sys.exit(1)
        except ValueError as e:
            logging.error(f"Error parsing date environment variables: {e}")
            sys.exit(1)

        return query_start_date, query_end_date

    def convert_dates_to_iso8601(self, start_date, end_date):
        """
        Converts UNIX timestamp dates to ISO 8601 format using DateUtilities.

        :param start_date: The start date in UNIX timestamp.
        :type start_date: int
```

```
        :param end_date: The end date in UNIX timestamp.
        :type end_date: int
        :return: Start and end dates in ISO 8601 format.
        :rtype: tuple
        """
        start_date_iso8601 =
 self.date_utilities.convert_unix_timestamp_to_iso8601(
            start_date
        )
        end_date_iso8601 = self.date_utilities.convert_unix_timestamp_to_iso8601(
            end_date
        )
        return start_date_iso8601, end_date_iso8601

    def execute_query(
        self,
        start_date_iso8601,
        end_date_iso8601,
        log_group="/workflows/cloudwatch-logs/large-query",
    ):
        """
        Creates a CloudWatchQuery instance and executes the query with provided
 date range.

        :param start_date_iso8601: The start date in ISO 8601 format.
        :type start_date_iso8601: str
        :param end_date_iso8601: The end date in ISO 8601 format.
        :type end_date_iso8601: str
        :param log_group: Log group to search: "/workflows/cloudwatch-logs/large-
query"
        :type log_group: str
        """
        cloudwatch_query = CloudWatchQuery(
            [start_date_iso8601, end_date_iso8601],
        )
        cloudwatch_query.query_logs((start_date_iso8601, end_date_iso8601))
        logging.info("Query executed successfully.")
        logging.info(
            f"Queries completed in {cloudwatch_query.query_duration} seconds.
 Total logs found: {len(cloudwatch_query.query_results)}"
        )


def main():
```

```
    """
    Main function to start a recursive CloudWatch logs query.
    Fetches required environment variables, converts dates, and executes the
 query.
    """
    logging.info("Starting a recursive CloudWatch logs query...")
    runner = CloudWatchLogsQueryRunner()
    query_start_date, query_end_date = runner.fetch_environment_variables()
    start_date_iso8601 = DateUtilities.convert_unix_timestamp_to_iso8601(
        query_start_date
    )
    end_date_iso8601 =
 DateUtilities.convert_unix_timestamp_to_iso8601(query_end_date)
    runner.execute_query(start_date_iso8601, end_date_iso8601)


if __name__ == "__main__":
    main()
```

此模組會處理超過 10,000 個結果的 CloudWatch 查詢。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
import logging
import time
from datetime import datetime
import threading
import boto3

from date_utilities import DateUtilities


class DateOutOfBoundsError(Exception):
    """Exception raised when the date range for a query is out of bounds."""

    pass


class CloudWatchQuery:
    """
    A class to query AWS CloudWatch logs within a specified date range.
```

```
    :ivar date_range: Start and end datetime for the query.
    :vartype date_range: tuple
    :ivar limit: Maximum number of log entries to return.
    :vartype limit: int
    """

    def __init__(self, date_range):
        self.lock = threading.Lock()
        self.log_groups = "/workflows/cloudwatch-logs/large-query"
        self.query_results = []
        self.date_range = date_range
        self.query_duration = None
        self.datetime_format = "%Y-%m-%d %H:%M:%S.%f"
        self.date_utilities = DateUtilities()
        self.limit = 10000

    def query_logs(self, date_range):
        """
        Executes a CloudWatch logs query for a specified date range and
calculates the execution time of the query.

        :return: A batch of logs retrieved from the CloudWatch logs query.
        :rtype: list
        """
        start_time = datetime.now()

        start_date, end_date = self.date_utilities.normalize_date_range_format(
            date_range, from_format="unix_timestamp", to_format="datetime"
        )

        logging.info(
            f"Original query:"
            f"\n       START:    {start_date}"
            f"\n       END:      {end_date}"
        )
        self.recursive_query((start_date, end_date))
        end_time = datetime.now()
        self.query_duration = (end_time - start_time).total_seconds()

    def recursive_query(self, date_range):
        """
        Processes logs within a given date range, fetching batches of logs
recursively if necessary.
```

```
        :param date_range: The date range to fetch logs for, specified as a tuple
(start_timestamp, end_timestamp).
        :type date_range: tuple
        :return: None if the recursive fetching is continued or stops when the
final batch of logs is processed.
                 Although it doesn't explicitly return the query results, this
method accumulates all fetched logs
                 in the `self.query_results` attribute.
        :rtype: None
        """
        batch_of_logs = self.perform_query(date_range)
        # Add the batch to the accumulated logs
        with self.lock:
            self.query_results.extend(batch_of_logs)
        if len(batch_of_logs) == self.limit:
            logging.info(f"Fetched {self.limit}, checking for more...")
            most_recent_log = self.find_most_recent_log(batch_of_logs)
            most_recent_log_timestamp = next(
                item["value"]
                for item in most_recent_log
                if item["field"] == "@timestamp"
            )
            new_range = (most_recent_log_timestamp, date_range[1])
            midpoint = self.date_utilities.find_middle_time(new_range)

            first_half_thread = threading.Thread(
                target=self.recursive_query,
                args=((most_recent_log_timestamp, midpoint),),
            )
            second_half_thread = threading.Thread(
                target=self.recursive_query, args=((midpoint, date_range[1]),)
            )

            first_half_thread.start()
            second_half_thread.start()

            first_half_thread.join()
            second_half_thread.join()

    def find_most_recent_log(self, logs):
        """
        Search a list of log items and return most recent log entry.
        :param logs: A list of logs to analyze.
        :return: log
```

```
        :type :return List containing log item details
        """
        most_recent_log = None
        most_recent_date = "1970-01-01 00:00:00.000"

        for log in logs:
            for item in log:
                if item["field"] == "@timestamp":
                    logging.debug(f"Compared: {item['value']} to
{most_recent_date}")
                    if (
                        self.date_utilities.compare_dates(
                            item["value"], most_recent_date
                        )
                        == item["value"]
                    ):
                        logging.debug(f"New most recent: {item['value']}")
                        most_recent_date = item["value"]
                        most_recent_log = log
        logging.info(f"Most recent log date of batch: {most_recent_date}")
        return most_recent_log

    # snippet-start:[python.example_code.cloudwatch_logs.start_query]
    def perform_query(self, date_range):
        """
        Performs the actual CloudWatch log query.

        :param date_range: A tuple representing the start and end datetime for
the query.
        :type date_range: tuple
        :return: A list containing the query results.
        :rtype: list
        """
        client = boto3.client("logs")
        try:
            try:
                start_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[0])
                )
                end_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[1])
                )
```

```
                response = client.start_query(
                    logGroupName=self.log_groups,
                    startTime=start_time,
                    endTime=end_time,
                    queryString="fields @timestamp, @message | sort @timestamp
asc",
                    limit=self.limit,
                )
                query_id = response["queryId"]
            except client.exceptions.ResourceNotFoundException as e:
                raise DateOutOfBoundsError(f"Resource not found: {e}")
            while True:
                time.sleep(1)
                results = client.get_query_results(queryId=query_id)
                if results["status"] in [
                    "Complete",
                    "Failed",
                    "Cancelled",
                    "Timeout",
                    "Unknown",
                ]:
                    return results.get("results", [])
        except DateOutOfBoundsError:
            return []

    def _initiate_query(self, client, date_range, max_logs):
        """
        Initiates the CloudWatch logs query.

        :param date_range: A tuple representing the start and end datetime for
the query.
        :type date_range: tuple
        :param max_logs: The maximum number of logs to retrieve.
        :type max_logs: int
        :return: The query ID as a string.
        :rtype: str
        """
        try:
            start_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[0])
            )
            end_time = round(
```

```
self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[1])
            )
            response = client.start_query(
                logGroupName=self.log_groups,
                startTime=start_time,
                endTime=end_time,
                queryString="fields @timestamp, @message | sort @timestamp asc",
                limit=max_logs,
            )
            return response["queryId"]
        except client.exceptions.ResourceNotFoundException as e:
            raise DateOutOfBoundsError(f"Resource not found: {e}")

    # snippet-end:[python.example_code.cloudwatch_logs.start_query]

    # snippet-start:[python.example_code.cloudwatch_logs.get_query_results]
    def _wait_for_query_results(self, client, query_id):
        """
        Waits for the query to complete and retrieves the results.

        :param query_id: The ID of the initiated query.
        :type query_id: str
        :return: A list containing the results of the query.
        :rtype: list
        """
        while True:
            time.sleep(1)
            results = client.get_query_results(queryId=query_id)
            if results["status"] in [
                "Complete",
                "Failed",
                "Cancelled",
                "Timeout",
                "Unknown",
            ]:
                return results.get("results", [])

    # snippet-end:[python.example_code.cloudwatch_logs.get_query_results]
```

- 如需 API 的詳細資訊，請參閱《適用於 Python (Boto3) 的AWS SDK API 參考資料》中的下列主題。

- GetQueryResults
- StartQuery

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱搭配 AWS SDK 使用 CloudWatch 記錄檔。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

使 AWS 用 SDK 的 CloudWatch 記錄跨服務範例

下列範例應用程式使用 AWS SDK 來結合 CloudWatch 記錄與其他 AWS 服務應用程式。每個範例都包含一個連結 GitHub，您可以在其中找到如何設定和執行應用程式的指示。

範例

- 使用排程事件來調用 Lambda 函數

使用排程事件來調用 Lambda 函數

下列程式碼範例說明如何建立 Amazon EventBridge 排程事件所叫用的 AWS Lambda 函數。

Python

適用於 Python (Boto3) 的 SDK

此範例顯示如何將 AWS Lambda 函數註冊為已排程 Amazon EventBridge 事件的目標。Lambda 處理常式會將友善的訊息和完整事件資料寫入 Amazon CloudWatch 日誌，以供日後擷取。

- 部署 Lambda 函式。
- 建立 EventBridge 排程的事件，並使 Lambda 函數成為目標。
- 授予允許 EventBridge 叫用 Lambda 函數的權限。
- 列印記 CloudWatch 錄檔中的最新資料，以顯示排定呼叫的結果。
- 清理示範期間建立的所有資源。

此範例最佳檢視時，請參閱 GitHub。有關如何設置和運行的完整源代碼和說明，請參閱中的完整示例GitHub。

此範例中使用的服務

- CloudWatch 日誌

- EventBridge
- Lambda

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱搭配 AWS SDK 使用 CloudWatch 記錄檔。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

Amazon CloudWatch 日誌中的安全

雲安全 AWS 是最高的優先級。身為 AWS 客戶，您可以從資料中心和網路架構中獲益，該架構專為滿足對安全性最敏感的組織的需求而打造。

安全是 AWS 與您之間共同承擔的責任。共同責任模型將其描述為雲端的安全性和雲端中的安全性：

- 雲端的安全性 — AWS 負責保護在 AWS 雲端中執行 AWS 服務的基礎架構。 AWS 還為您提供可以安全使用的服務。若要深入瞭解適用於的規範遵循計劃 WorkSpaces，請參閱合規計劃的AWS 服務範圍範圍)。
- 雲端中的安全性 — 您的責任取決於您使用的 AWS 服務。您也必須對其他因素負責，包括資料的機密性、您公司的要求和適用法律和法規

本文件可協助您了解如何在使用 Amazon CloudWatch 日誌時套用共同的責任模型。它說明如何設定 Amazon CloudWatch 日誌以符合您的安全和合規目標。您也會學到如何使用其他可 AWS 協助您監控和保護 CloudWatch Logs 資源的服務。

目錄

Amazon CloudWatch 日誌中的數據保護

> **Note**
>
> 除了下列有關中一般資料保護的資訊之外 AWS， CloudWatch Logs 也可讓您透過遮罩來保護記錄事件中的機密資料。如需詳細資訊，請參閱 使用遮罩功能協助保護敏感日誌資料。

AWS 共同責任模型適用於 Amazon CloudWatch Logs 中的資料保護。如此模型中所述， AWS 負責保護執行所有 AWS 雲端. 您負責維護在此基礎設施上託管內容的控制權。您也同時負責所使用 AWS

服務 的安全組態和管理任務。如需資料隱私權的詳細資訊，請參閱資料隱私權常見問答集。如需有關歐洲資料保護的相關資訊，請參閱 AWS 安全性部落格上的 AWS 共同的責任模型和 GDPR 部落格文章。

基於資料保護目的，我們建議您使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 保護 AWS 帳戶 登入資料並設定個別使用者。如此一來，每個使用者都只會獲得授與完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 每個帳戶均要使用多重要素驗證 (MFA)。
- 使用 SSL/TLS 與 AWS 資源進行通訊。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 使用設定 API 和使用者活動記錄 AWS CloudTrail。
- 使用 AWS 加密解決方案，以及其中的所有默認安全控制 AWS 服務。
- 使用進階的受管安全服務 (例如 Amazon Macie)，協助探索和保護儲存在 Amazon S3 的敏感資料。
- 如果您在透過命令列介面或 API 存取時需要經 AWS 過 FIPS 140-2 驗證的加密模組，請使用 FIPS 端點。如需有關 FIPS 和 FIPS 端點的更多相關資訊，請參閱聯邦資訊處理標準 (FIPS) 140-2 概觀。

我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊，放在標籤或自由格式的文字欄位中，例如名稱欄位。這包括當您使用控制台，API 或 AWS SDK AWS 服務 使用 CloudWatch 日誌或其他日誌時。 AWS CLI您在標籤或自由格式文字欄位中輸入的任何資料都可能用於計費或診斷日誌。如果您提供外部伺服器的 URL，我們強烈建議請勿在驗證您對該伺服器請求的 URL 中包含憑證資訊。

靜態加密

CloudWatch 記錄檔會使用加密來保護靜態資料。所有日誌群組都會經過加密。根據預設，CloudWatch Logs 服務會管理伺服器端加密金鑰。

如果您想要管理用於加密和解密記錄的金鑰，請使 AWS KMS 用金鑰。如需詳細資訊，請參閱 使用加密記 CloudWatch 錄檔中的記錄資料 AWS Key Management Service。

傳輸中加密

CloudWatch 記錄檔會使用傳輸中的資料 end-to-end 加密。Lo CloudWatch gs 服務會管理伺服器端加密金鑰。

Amazon CloudWatch 日誌的身分和存取管理

存取 Amazon CloudWatch 日誌需要 AWS 可用來驗證請求的登入資料。這些認證必須具有存取 AWS 資源的權限，例如擷取有關雲端資源的 CloudWatch 記錄檔資料。以下各節詳細說明如何使用 AWS

Identity and Access Management (IAM) 和 CloudWatch 記錄，透過控制可存取資源的人員來協助保護資源安全：

- 身分驗證
- 存取控制

身分驗證

若要提供存取權，請新增權限至您的使用者、群組或角色：

- 使用者和群組位於 AWS IAM Identity Center：

  建立權限合集。請按照 AWS IAM Identity Center 使用者指南 中的 建立權限合集 說明進行操作。
- 透過身分提供者在 IAM 中管理的使用者：

  建立聯合身分的角色。請按照 IAM 使用者指南 的 為第三方身分提供者 (聯合) 建立角色 中的指示進行操作。
- IAM 使用者：
  - 建立您的使用者可擔任的角色。請按照 IAM 使用者指南 的 為 IAM 使用者建立角色 中的指示進行操作。
  - (不建議) 將政策直接附加至使用者，或將使用者新增至使用者群組。請遵循 IAM 使用者指南的新增許可到使用者 (主控台)中的指示。

存取控制

您可以擁有有效的認證來驗證您的請求，但除非您有權限，否則您無法創建或訪問 CloudWatch 日誌資源。例如，您必須有建立日誌串流、建立日誌群組等的許可。

下列各節說明如何管理 CloudWatch 記錄檔的權限。我們建議您先閱讀概觀。

- 管理 CloudWatch Logs 資源存取權限的概觀
- 針對記錄使用身分型政策 (IAM 政策) CloudWatch
- CloudWatch 記錄檔權限參考

管理 CloudWatch Logs 資源存取權限的概觀

若要提供存取權，請新增權限至您的使用者、群組或角色：

- 使用者和群組位於 AWS IAM Identity Center：

  建立權限合集。請按照 AWS IAM Identity Center 使用者指南 中的 建立權限合集 說明進行操作。
- 透過身分提供者在 IAM 中管理的使用者：

  建立聯合身分的角色。請按照 IAM 使用者指南 的 為第三方身分提供者 (聯合) 建立角色 中的指示進行操作。
- IAM 使用者：
  - 建立您的使用者可擔任的角色。請按照 IAM 使用者指南 的 為 IAM 使用者建立角色 中的指示進行操作。
  - (不建議) 將政策直接附加至使用者，或將使用者新增至使用者群組。請遵循 IAM 使用者指南的新增許可到使用者 (主控台)中的指示。

主題

CloudWatch 記錄資源和作業

在 CloudWatch 記錄檔中，主要資源包括記錄群組、記錄串流和目的地。 CloudWatch 記錄檔不支援子資源 (與主要資源搭配使用的其他資源)。

這些資源和子資源都有獨一無二的 Amazon Resource Name (ARN) 與其相關聯，如下表所示。

資源類型	ARN 格式
日誌群組	以下兩種方式皆可使用。第二個，:*最後，是 `describe-log-groups` CLI 命令和 DescribeLogGroupsAPI 返回的內容。 arn:aws:logs:*region*:*account-id* :log-group:*log_group_name*

資源類型	ARN 格式
	arn:aws:logs:*region*:*account-id* :log-group:*log_group_name* :* 在下列情況下，請使用第一個版本 (不含尾端:*)： • 在許多 CloudWatch Logs API 的`logGroupIdentifier` 輸入欄位中。 • 在標記 API 的`resourceArn` 欄位中 • 在 IAM 策略中，指定TagResource、UntagResource和的權限時ListTagsForResource。 在 IAM 政策中為所有其他 API 動作指定許可時:*，請使用第二個版本 (尾隨) 參照 ARN。
日誌串流	*ARN: AW: ##:##:## ID: ###:####:###:log-stream-name*
目的地	arn:aws:logs:*region*:*account-id* :destination:*destination_name*

如需 ARN 的詳細資訊，請參閱《IAM 使用者指南》中的 ARN。如需 CloudWatch 日誌 ARN 的相關資訊，請參閱中的 Amazon 資源名稱 (ARN)。Amazon Web Services 一般參考如需涵蓋CloudWatch 記錄檔的策略範例，請參閱針對記錄使用身分型政策 (IAM 政策) CloudWatch 。

CloudWatch 記錄檔提供了一組作業來處理記 CloudWatch 錄資源。如需可用操作的清單，請參閱 CloudWatch 記錄檔權限參考。

了解資源所有權

AWS 帳號擁有在帳號中建立的資源，無論是誰建立資源。具體而言，資源擁有者是驗證資源建立請求的主體實體 (即根帳戶、使用者或 IAM 角色) 的帳戶。 AWS 下列範例說明其如何運作：

- 如果您使用帳戶的根帳戶認證來建立記錄群組，則您的 AWS 帳戶就是 CloudWatch Logs 資源的擁有者。 AWS

- 如果您在 AWS 帳戶中建立使用者，並將建立 CloudWatch 記錄資源的權限授與該使用者，則該使用者可以建立 CloudWatch 記錄資源。不過，使用者所屬的 AWS 帳戶擁有 Lo CloudWatch gs 資源。
- 如果您在具有建立 CloudWatch 記錄資源權限的 AWS 帳戶中建立 IAM 角色，則任何可以擔任該角色的人都可以建立 CloudWatch 記錄資源。您的 AWS 帳戶 (角色所屬) 擁有記 CloudWatch 錄資源。

管理資源存取

許可政策描述誰可以存取哪些資源。下一節說明可用來建立許可政策的選項。

> **Note**
>
> 本節討論在日誌環境中使用 CloudWatch IAM。它不提供 IAM 服務的詳細資訊。如需完整的 IAM 文件，請參閱《IAM 使用者指南》中的什麼是 IAM？。如需有關 IAM 政策語法和說明的資訊，請參閱《IAM 使用者指南》中的 IAM 政策參考。

附加至 IAM 身分的政策稱為身分型政策 (IAM 政策)，而附加至資源的政策則稱為以資源為基礎的政策。 CloudWatch 記錄支援以身分識別為基礎的政策，以及用於啟用跨帳戶訂閱的目的地的資源型政策。如需詳細資訊，請參閱 跨帳戶跨區域訂閱。

主題

- 日誌群組許可和 Contributor Insights
- 資源型政策

日誌群組許可和 Contributor Insights

參與者見解是一項功能，可 CloudWatch 讓您分析記錄群組中的資料，並建立顯示參與者資料的時間序列。您可以查看與前 N 個參與者有關的指標、唯一參與者的總數及其用量。如需詳細資訊，請參閱使用 Contributor Insights 來分析高基數資料。

當您授與使用者`cloudwatch:PutInsightRule`和`cloudwatch:GetInsightRuleReport`權限時，該使用者可以建立規則來評估記 CloudWatch 錄檔中的任何記錄群組，然後查看結果。結果可能包含這些日誌群組的參與者資料。請務必將這些許可只授予允許檢視此資料的使用者。

資源型政策

CloudWatch 記錄支援以資源為基礎的目的地政策，您可以使用這些策略來啟用跨帳戶訂閱。如需詳細資訊，請參閱 步驟 1：建立目的地。您可以使用 PutDestinationAPI 建立目的地，也可以使用 PutDestinationPolicyAPI 將資源策略新增至目的地。下列範例允許其他 AWS 帳戶識別碼為 111122223333 的帳戶將其記錄群組訂閱至目的地。`arn:aws:logs:us-east-1:123456789012:destination:testDestination`

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111122223333"
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" : "arn:aws:logs:us-east-1:123456789012:destination:testDestination"
    }
  ]
}
```

指定政策元素：動作、效果和委託人

服務會針對每個 CloudWatch 記錄檔資源定義一組 API 作業。若要授與這些 API 作業的權限，CloudWatch 記錄會定義一組您可以在策略中指定的動作。為了執行 API 操作，某些 API 操作可能需要多個動作的許可。如需資源與 API 操作的詳細資訊，請參閱 CloudWatch 記錄資源和作業 與 CloudWatch 記錄檔權限參考.

以下是基本的政策元素：

- 資源 - 您使用 Amazon Resource Name (ARN) 識別欲套用政策的資源。如需詳細資訊，請參閱 CloudWatch 記錄資源和作業。
- 動作：使用動作關鍵字識別您要允許或拒絕的資源操作。例如，`logs.DescribeLogGroups` 許可允許使用者執行 `DescribeLogGroups` 操作。
- 效果 – 您可以指定使用者請求特定動作時會有什麼效果 (允許或拒絕)。如果您未明確授予存取 (允許) 資源，則隱含地拒絕存取。您也可以明確拒絕資源存取，這樣做可確保使用者無法存取資源，即使不同政策授予存取也是一樣。

- 委託人：在身分識別型政策 (IAM 政策) 中，政策所連接的使用者就是隱含委託人。對於以資源為基礎的策略，您可以指定要接收權限的使用者、帳戶、服務或其他實體 (僅適用於以資源為基礎的策略)。 CloudWatch 記錄檔支援目標的資源型政策。

如需進一步了解有關 IAM 政策語法和說明的詳細資訊，請參閱《IAM 使用者指南》 中的 AWS IAM 政策參考。

如需顯示所有 CloudWatch 記錄 API 動作及其套用之資源的表格，請參閱CloudWatch 記錄檔權限參考。

在政策中指定條件

當您授予許可時，可以使用存取政策語言來指定政策應該何時生效的條件。例如，建議只在特定日期之後套用政策。如需使用政策語言指定條件的詳細資訊，請參閱IAM 使用者指南中的條件。

欲表示條件，您可以使用預先定義的條件金鑰。如需每個 AWS 服務支援的內容金鑰清單，以及 AWS 全域原則金鑰清單，請參閱服AWS 務和AWS 全域條件內容金鑰的動作、資源和條件索引鍵。

> **Note**
>
> 您可以使用標籤來控制對 CloudWatch 記錄檔資源的存取，包括記錄群組和目的地。由於日誌群組與日誌串流之間的階層關係，系統會在日誌群組層級控制對日誌串流的存取。如需有關使用標籤來控制存取的詳細資訊，請參閱使用標籤控制對 Amazon Web Services 資源的存取。

針對記錄使用身分型政策 (IAM 政策) CloudWatch

這個主題提供以身分為基礎的政策範例，在該政策中帳戶管理員可以將許可政策連接至 IAM 身分 (即使用者、群組和角色)。

> **Important**
>
> 我們建議您先檢閱介紹性主題，其中說明可用來管理 CloudWatch 記錄資源存取權的基本概念和選項。如需詳細資訊，請參閱 管理 CloudWatch Logs 資源存取權限的概觀。

本主題涵蓋下列項目：

- 使用 CloudWatch 主控台所需的權限

- AWS CloudWatch 記錄檔的受管理 (預先定義) 策略
- 客戶受管政策範例

以下是許可政策的範例：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
    ],
      "Resource": [
        "arn:aws:logs:*:*:*"
    ]
  }
 ]
}
```

此政策有一個陳述式，將授予許可來建立日誌群組和日誌串流，以便將日誌事件上傳至日誌串流，以及列出日誌串流的詳細資訊。

Resource 值結尾的萬用字元 (*) 表示該陳述式允許對任何日誌群組執行 logs:CreateLogGroup、logs:CreateLogStream、logs:PutLogEvents 及 logs:DescribeLogStreams 動作的許可。若要限制此許可只提供給特定日誌群組，請將資源 ARN 中的萬用字元 (*) 更換為特定日誌群組 ARN。如需 IAM 政策陳述式中各區段的詳細資訊，請參閱《IAM 使用者指南》中的 IAM 政策元素參考。如需顯示所有「 CloudWatch 記錄檔」動作的清單，請參閱CloudWatch 記錄檔權限參考。

使用 CloudWatch 主控台所需的權限

若要讓使用者在 CloudWatch 主控台中使用 CloudWatch 記錄檔，該使用者必須擁有一組最低權限，才能讓使用者描述其 AWS 帳戶中的其他 AWS 資源。若要在 CloudWatch 主控台中使用 CloudWatch Logs，您必須擁有下列服務的權限：

- CloudWatch

- CloudWatch 日誌
- OpenSearch 服務
- IAM
- Kinesis
- Lambda
- Amazon S3

如果您建立比最基本必要許可更嚴格的 IAM 政策，則對於採取該 IAM 政策的使用者而言， 主控台就無法如預期運作。若要確保這些使用者仍然可以使用 CloudWatch 主控台，請同時將受`CloudWatchReadOnlyAccess`管理的策略附加至使用者，如中所述AWS CloudWatch 記錄檔的受管理 (預先定義) 策略。

您不需要為僅對 AWS CLI 或 CloudWatch Logs API 進行呼叫的使用者允許最低主控台權限。

對於不使用 CloudWatch 控制台來管理記錄訂閱的使用者使用主控台所需的完整權限集如下：

- 雲觀察:GetMetricData
- 雲觀察:ListMetrics
- 日誌：CancelExportTask
- 日誌：CreateExportTask
- 日誌：CreateLogGroup
- 日誌：CreateLogStream
- 日誌：DeleteLogGroup
- 日誌：DeleteLogStream
- 日誌：DeleteMetricFilter
- 日誌：DeleteQueryDefinition
- 日誌：DeleteRetentionPolicy
- 日誌：DeleteSubscriptionFilter
- 日誌：DescribeExportTasks
- 日誌：DescribeLogGroups
- 日誌：DescribeLogStreams
- 日誌：DescribeMetricFilters

- 日誌：DescribeQueryDefinitions
- 日誌：DescribeQueries
- 日誌：DescribeSubscriptionFilters
- 日誌：FilterLogEvents
- 日誌：GetLogEvents
- 日誌：GetLogGroupFields
- 日誌：GetLogRecord
- 日誌：GetQueryResults
- 日誌：PutMetricFilter
- 日誌：PutQueryDefinition
- 日誌：PutRetentionPolicy
- 日誌：StartQuery
- 日誌：StopQuery
- 日誌：PutSubscriptionFilter
- 日誌：TestMetricFilter

對於也將使用主控台來管理日誌訂閱的使用者而言，也需要以下許可：

- 是:DescribeElasticsearchDomain
- 是:ListDomainNames
- IAM：AttachRolePolicy
- IAM：CreateRole
- IAM：GetPolicy
- IAM：GetPolicyVersion
- IAM：GetRole
- IAM：ListAttachedRolePolicies
- IAM：ListRoles
- 室壁運動:DescribeStreams
- 室壁運動:ListStreams
- 拉姆達：AddPermission

- 拉姆達：CreateFunction
- 拉姆達：GetFunctionConfiguration
- 拉姆達：ListAliases
- 拉姆達：ListFunctions
- 拉姆達：ListVersionsByFunction
- 拉姆達：RemovePermission
- S3：ListBuckets

AWS CloudWatch 記錄檔的受管理 (預先定義) 策略

AWS 透過提供由建立和管理的獨立 IAM 政策來解決許多常見使用案例 AWS。受管政策授與常見使用案例中必要的許可，讓您免於查詢需要哪些許可。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 受管理策略 (您可以附加至帳戶中的使用者和角色) 是 CloudWatch 記錄檔專用的：

- CloudWatchLogsFullAccess— 授與 CloudWatch 記錄檔的完整存取權限。
- CloudWatchLogsReadOnlyAccess— 授與CloudWatch 記錄檔的唯讀存取權。

CloudWatchLogsFullAccess

此原CloudWatchLogsFullAccess則會授與 CloudWatch 記錄檔的完整存取權。此原則包含`cloudwatch:GenerateQuery`權限，因此具有此原則的使用者可以從自然語言提示中產生 CloudWatch Logs Insights 查詢字串。其內容如下：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "logs:*",
                "cloudwatch:GenerateQuery"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

CloudWatchLogsReadOnlyAccess

此原CloudWatchLogsReadOnlyAccess則會授與 CloudWatch 記錄檔的唯讀存取權。它包含`cloudwatch:GenerateQuery`權限，因此具有此原則的使用者可以從自然語言提示中產生 CloudWatch Logs Insights 查詢字串。其內容如下：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:Describe*",
                "logs:Get*",
                "logs:List*",
                "logs:StartQuery",
                "logs:StopQuery",
                "logs:TestMetricFilter",
                "logs:FilterLogEvents",
                "logs:StartLiveTail",
                "logs:StopLiveTail",
                "cloudwatch:GenerateQuery"
            ],
            "Resource": "*"
        }
    ]
}
```

CloudWatchLogsCrossAccountSharingConfiguration

此CloudWatchLogsCrossAccountSharingConfiguration原則會授與建立、管理和檢視可觀察性存取管理員連結的存取權，以便在帳號之間共用 CloudWatch 記錄資源。如需詳細資訊，請參閱CloudWatch 跨帳戶可觀察性。

其內容如下：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:Link",
```

```
                "oam:ListLinks"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:DeleteLink",
                "oam:GetLink",
                "oam:TagResource"
            ],
            "Resource": "arn:aws:oam:*:*:link/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:CreateLink",
                "oam:UpdateLink"
            ],
            "Resource": [
                "arn:aws:oam:*:*:link/*",
                "arn:aws:oam:*:*:sink/*"
            ]
        }
    ]
}
```

CloudWatch 記錄 AWS 受管策略的更新

檢視有關 CloudWatch 記錄檔 AWS 受管理策略更新的詳細資料，因為此服務開始追蹤這些變更。如需有關此頁面變更的自動警示，請訂閱 CloudWatch 記錄文件歷史記錄頁面上的 RSS 摘要。

變更	描述	日期
CloudWatchLogsFullAccess – 更新現有政策。	CloudWatch 記錄檔已新增權限CloudWatchLogsFullAccess。 已新增cloudwatch:GenerateQuery 權	2023 年 11 月 27 日

變更	描述	日期
	限，以便具有此原則的使用者可以從自然語言提示中產生 CloudWatch Logs Insights 查詢字串。	
CloudWatchLogsReadOnlyAccess – 更新現有政策。	CloudWatch 已將權限新增至 CloudWatchLogsReadOnlyAccess. 已新增`cloudwatch:GenerateQuery` 權限，以便具有此原則的使用者可以從自然語言提示中產生 CloudWatch Logs Insights 查詢字串。	2023 年 11 月 27 日
CloudWatchLogsReadOnlyAccess – 更新現有政策	CloudWatch 記錄檔已將權限新增至 CloudWatchLogsReadOnlyAccess. `logs:StartLiveTail` 和`logs:StopLiveTail` 權限已新增，讓具有此原則的使用者可以使用主控台來啟動和停止CloudWatch 記錄即時尾端工作階段。如需詳細資訊，請參閱使用 Live Tail 以近乎即時的方式檢視日誌。	2023 年 6 月 6 日

變更	描述	日期
CloudWatchLogsCrossAccountSharingConfiguration – 新政策	CloudWatch 記錄新增了一項新政策，可讓您管理共用 CloudWatch 記錄檔記錄群組的 CloudWatch 跨帳戶觀察性連結。 如需詳細資訊，請參閱CloudWatch 跨帳戶可觀察性	2022 年 11 月 27 日
CloudWatchLogsReadOnlyAccess – 更新現有政策	CloudWatch 記錄檔已將權限新增至 CloudWatchLogsReadOnlyAccess. 已新增`oam:ListSinks` 和`oam:ListAttachedLinks` 權限，以便具有此原則的使用者可以使用主控台，以 CloudWatch 跨帳戶觀察性檢視從來源帳戶共用的資料。	2022 年 11 月 27 日

客戶受管政策範例

您可以建立自己的自訂 IAM 政策，以允許 CloudWatch 記錄動作和資源的許可。您可以將這些自訂政策連接至需要這些許可的使用者或群組。

在本節中，您可以找到授與各種「CloudWatch 記錄檔」動作權限的範例使用者策略。當您使用記 CloudWatch 錄 API、 AWS SDK 或. AWS CLI

範例

- 範例 1：允許完整存取記 CloudWatch 錄檔
- 範例 2：允許唯讀存取記 CloudWatch 錄檔
- 範例 3：允許存取一個日誌群組

範例 1：允許完整存取記 CloudWatch 錄檔

下列原則可讓使用者存取所有「 CloudWatch 記錄檔」動作。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

範例 2：允許唯讀存取記 CloudWatch 錄檔

AWS 提供CloudWatchLogsReadOnlyAccess原則以唯讀方式存取 CloudWatch 記錄檔資料。此政策包含以下許可。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "logs:Describe*",
                "logs:Get*",
                "logs:List*",
                "logs:StartQuery",
                "logs:StopQuery",
                "logs:TestMetricFilter",
                "logs:FilterLogEvents",
                "logs:StartLiveTail",
                "logs:StopLiveTail",
                "cloudwatch:GenerateQuery"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

範例 3：允許存取一個日誌群組

以下政策允許使用者在一個指定的日誌群組中讀取和寫入日誌事件。

> **Important**
>
> 在 Resource 行中，需要日誌群組名稱末尾的 :* 來表示該政策適用於此日誌群組中的所有日誌串流。如果省略 :*，將不會強制執行此政策。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
      "Action": [
        "logs:CreateLogStream",
        "logs:DescribeLogStreams",
        "logs:PutLogEvents",
        "logs:GetLogEvents"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:logs:us-west-2:123456789012:log-group:SampleLogGroupName:*"
      }
   ]
}
```

使用標記和 IAM 政策在日誌群組層級進行控制

您可以授與使用者存取特定日誌群組，同時防止他們存取其他日誌群組。若要這樣做，請標記日誌群組，並使用 IAM 政策來參考這些標籤。若要將標籤套用至日誌群組，您必須擁有 logs:TagResource 或 logs:TagLogGroup 許可。無論您是在建立日誌群組時將標籤指派給日誌群組，或稍後將標籤指派給日誌群組，此許可要求均適用。

如需有關標籤日誌群組的詳細資訊，請參閱 在 Amazon CloudWatch 日誌中標記日誌群組。

當您標記日誌群組時，您就可以授予 IAM 政策給使用者，以只允許存取包含特定標籤的日誌群組。例如，以下政策陳述式只會授予標籤鍵 Team 值為 Green 日誌群組的存取。

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Action": [
                "logs:*"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aws:ResourceTag/Team": "Green"
                }
            }
        }
    ]
}
```

在傳統意義上，StopQuery和 StopLiveTailAPI 操作不會與 AWS 資源互動。其不會傳回任何資料，放置任何資料，或以任何方式修改資源。相反地，它們只會在指定的即時尾端工作階段或指定的 CloudWatch 日誌見解查詢上運作，這些查詢未歸類為資源。因此，當您在 IAM 政策中針對這些操作指定 Resource 欄位時，必須將 Resource 欄位的值設定為 *，如下列範例所示。

```
{
    "Version": "2012-10-17",
    "Statement":
        [ {
            "Effect": "Allow",
            "Action": [
                "logs:StopQuery",
                "logs:StopLiveTail"
            ],
            "Resource": "*"
            }
        ]
}
```

如需有關使用 IAM 政策陳述式的詳細資訊，請參閱《IAM 使用者指南》中的使用政策控制存取。

CloudWatch 記錄檔權限參考

當您在設定 存取控制 並撰寫可連接到 IAM 身分 (以身分為基礎的政策) 的許可政策時，可以使用下列資料表作為參考。此表格會列出每個 CloudWatch 記錄 API 作業，以及您可授與執行動作之權限的對應動作。您可以在政策的 Action 欄位中指定動作。對於此Resource欄位，您可以指定記錄群組或記錄資料流的 ARN，或指定*代表所有 CloudWatch 記錄資源。

您可以在 CloudWatch 記錄檔政策中使用 AWS寬條件金鑰來表示條件。如需全金鑰的 AWS完整清單，請參閱 IAM 使用者指南中的AWS 全域和 IAM 條件內容金鑰。

> **Note**
>
> 若要指定動作，請使用 `logs:` 前綴，後面接著 API 操作名稱。例如：`logs:CreateLogGrouplogs:CreateLogStream`、或 `logs:*` (適用於所有「CloudWatch 記錄檔」動作)。

CloudWatch 記錄 API 操作和動作所需的權限

CloudWatch 記錄 API 作業	所需許可 (API 動作)
CancelExportTask	`logs:CancelExportTask` 取消待處理或執行匯出任務時為必要。
CreateExportTask	`logs:CreateExportTask` 從日誌群組將資料匯出至 Simple Storage Service (Amazon S3) 儲存貯體時為必要。
CreateLogGroup	`logs:CreateLogGroup` 建立新日誌群組時為必要。
CreateLogStream	`logs:CreateLogStream` 在日誌群組中建立新日誌串流時為必要。
DeleteDestination	`logs:DeleteDestination` 刪除日誌目的地及停用其任何訂閱篩選條件時為必要。
DeleteLogGroup	`logs:DeleteLogGroup` 刪除日誌群組及任何相關的存檔日誌事件時為必要。
DeleteLogStream	`logs:DeleteLogStream`

CloudWatch 記錄 API 作業	所需許可 (API 動作)
	刪除日誌串流及任何相關的存檔日誌事件時為必要。
DeleteMetricFilter	`logs:DeleteMetricFilter` 刪除與日誌群組相關聯的指標篩選條件時為必要。
DeleteQueryDefinition	`logs:DeleteQueryDefinition` 刪除 CloudWatch 記錄深入解析中儲存的查詢定義所需。
DeleteResourcePolicy	`logs:DeleteResourcePolicy` 刪除 CloudWatch 記錄檔資源策略所需。
DeleteRetentionPolicy	`logs:DeleteRetentionPolicy` 刪除日誌群組的保留政策時為必要。
DeleteSubscriptionFilter	`logs:DeleteSubscriptionFilter` 刪除與日誌群組相關聯的訂閱篩選條件時為必要。
DescribeDestinations	`logs:DescribeDestinations` 檢視與帳戶相關的所有目的地時為必要。
DescribeExportTasks	`logs:DescribeExportTasks` 檢視與帳戶相關的所有匯出任務時為必要。
DescribeLogGroups	`logs:DescribeLogGroups` 檢視與帳戶相關的所有日誌群組時為必要。

CloudWatch 記錄 API 作業	所需許可 (API 動作)
DescribeLogStreams	`logs:DescribeLogStreams` 檢視與日誌群組相關的所有日誌串流時為必要。
DescribeMetricFilters	`logs:DescribeMetricFilters` 檢視與日誌群組相關的所有指標時為必要。
DescribeQueryDefinitions	`logs:DescribeQueryDefinitions` 需要在 CloudWatch 日誌見解中查看已儲存的查詢定義清單。
DescribeQueries	`logs:DescribeQueries` 需要查看已排程、執行或最近執行的 CloudWatch 記錄見解查詢清單。
DescribeResourcePolicies	`logs:DescribeResourcePolicies` 檢視記 CloudWatch 錄檔資源策略清單所需。
DescribeSubscriptionFilters	`logs:DescribeSubscriptionFilters` 檢視與日誌群組相關聯的所有訂閱篩選條件時為必要。
FilterLogEvents	`logs:FilterLogEvents` 依據日誌群組篩選條件模式排序日誌事件時為必要。
GetLogEvents	`logs:GetLogEvents` 從日誌串流擷取日誌事件時為必要。

CloudWatch 記錄 API 作業	所需許可 (API 動作)
GetLogGroupFields	`logs:GetLogGroupFields` 擷取日誌群組內日誌事件中包含的欄位清單時為必要。
GetLogRecord	`logs:GetLogRecord` 從單一日誌事件擷取詳細資訊時為必要。
GetQueryResults	`logs:GetQueryResults` 擷取 CloudWatch 日誌見解查詢的結果所需。
ListTagsLogGroup	`logs:ListTagsLogGroup` 列出與日誌群組相關的標籤時為必要。
PutDestination	`logs:PutDestination` 需要建立或更新目的地日誌串流 (例如 Kinesis 串流) 時為必要。
PutDestinationPolicy	`logs:PutDestinationPolicy` 建立或更新與現有日誌目的地相關的存取政策時為必要。
PutLogEvents	`logs:PutLogEvents` 將日誌事件批次上傳至日誌串流時為必要。
PutMetricFilter	`logs:PutMetricFilter` 建立或更新指標篩選條件並將其與日誌群組建立關聯時為必要。
PutQueryDefinition	`logs:PutQueryDefinition` 必須在 CloudWatch 記錄檔見解中儲存查詢。

CloudWatch 記錄 API 作業	所需許可 (API 動作)
PutResourcePolicy	`logs:PutResourcePolicy` 建立記 CloudWatch 錄檔資源策略所需。
PutRetentionPolicy	`logs:PutRetentionPolicy` 設定將日誌事件保持 (保留) 在日誌群組中的天數時為必要。
PutSubscriptionFilter	`logs:PutSubscriptionFilter` 建立或更新訂閱篩選條件並將其與日誌群組建立關聯時為必要。
StartQuery	`logs:StartQuery` 需要啟動 CloudWatch 日誌見解查詢。
StopQuery	`logs:StopQuery` 需要停止正在進行的 CloudWatch 日誌見解查詢。
TagLogGroup	`logs:TagLogGroup` 新增或更新日誌群組標籤時為必要。
TestMetricFilter	`logs:TestMetricFilter` 針對日誌事件訊息的取樣來測試篩選條件模式時為必要。

針 CloudWatch 對記錄檔使用服務連結角色

Amazon CloudWatch 日誌使用 AWS Identity and Access Management (IAM) 服務連結角色。服務連結角色是直接連結至 CloudWatch Logs 的唯一 IAM 角色類型。服務連結角色由 CloudWatch Logs 預先定義，並包含服務代表您呼叫其他服 AWS 務所需的所有權限。

服務連結角色可讓設定 CloudWatch 記錄更有效率，因為您不需要手動新增必要的權限。 CloudWatch 記錄檔會定義其服務連結角色的權限，除非另有定義，否則只有 CloudWatch 記錄檔可以擔任這些角色。已定義的許可包括信任政策和許可政策。該許可政策無法連接至其他任何 IAM 實體。

關於支援服務連結角色的其他服務，如需相關資訊，請參閱與 IAM 搭配運作的AWS 服務。尋找服務連結角色欄中顯示 Yes (是) 的服務。選擇具有連結的是，以檢視該服務的服務連結角色文件。

記錄檔的 CloudWatch 服務連結角色權限

CloudWatch 記錄檔會使用名為AWSServiceRoleForLogDelivery的服務連結角色。CloudWatch 記錄檔會使用此服務連結角色，將記錄檔直接寫入 Firehose。如需詳細資訊，請參閱 啟用從 AWS 服務記錄。

AWSServiceRoleForLogDelivery 服務連結角色信任下列服務擔任角色：

- `logs.amazonaws.com`

角色權限原則允許 CloudWatch Logs 對指定的資源完成下列動作：

- 動作：以`firehose:PutRecord`及`firehose:PutRecordBatch`在所有 Firehose 串流上有一個標籤的`LogDeliveryEnabled`索引鍵值為。`True`當您建立訂閱以將記錄傳送至 Firehose 時，此標籤會自動附加至 Firehose 串流。

您必須設定許可來允許 IAM 實體建立、編輯或刪除服務連結角色。此實體可以是使用者、群組或角色。如需詳細資訊，請參閱《IAM 使用者指南》中的服務連結角色許可。

建立記錄檔的 CloudWatch 服務連結角色

您不需要手動建立服務連結角色。當您將記錄設定為直接傳送至 AWS Management Console、或 AWS API 中的 Firehose 串流時 AWS CLI， CloudWatch Logs 會為您建立服務連結角色。

若您刪除此服務連結角色，之後需要再次建立，您可以在帳戶中使用相同程序重新建立角色。當您再次將日誌設置為直接發送到 Firehose 流時， CloudWatch Logs 會再次為您創建服務鏈接的角色。

編輯記錄檔的 CloudWatch 服務連結角色

CloudWatch 記錄檔不允許您在建立後編輯AWSServiceRoleForLogDelivery或編輯任何其他服務連結角色。因為各種實體可能會參考角色，所以您無法變更角色的名稱。然而，您可使用 IAM 來編輯角色描述。如需更多資訊，請參閱 IAM 使用者指南中的編輯服務連結角色。

刪除記錄檔的 CloudWatch 服務連結角色

若您不再使用需要服務連結角色的功能或服務，我們建議您刪除該角色。如此一來，您就沒有未主動監控或維護的未使用實體。然而，在手動刪除服務連結角色之前，您必須先清除資源。

> **Note**
>
> 當您嘗試刪除資源時，如果 CloudWatch 記錄檔服務正在使用此角色，則刪除作業可能會失敗。若此情況發生，請等待數分鐘後並再次嘗試操作。

刪除AWSServiceRoleForLogDelivery服務連結角色所使用的 CloudWatch 記錄檔資源

- 停止將記錄檔直接傳送至 Firehose 串流。

使用 IAM 手動刪除服務連結角色

使用 IAM 主控台或 AWS API 刪除AWSServiceRoleForLogDelivery服務連結角色。 AWS CLI如需詳細資訊，請參閱刪除服務連結角色

支援 CloudWatch 記錄檔服務連結角色的區域

CloudWatch 記錄檔支援在服務提供服務的所有 AWS 區域中使用服務連結角色。如需詳細資訊，請參閱CloudWatch 記錄區域和端點。

Amazon CloudWatch 日誌的合規驗證

第三方稽核員會評估 Amazon CloudWatch Logs 的安全性和合規性，做為多個 AWS 合規計劃的一部分。這些計劃包括 SOC、PCI、FedRAMP、HIPAA 等等。

如需特定規範計劃範圍內的 AWS 服務清單，請參閱合規計劃AWS 服務範圍方案)。如需一般資訊，請參閱 AWS 合規計劃 。

您可以使用下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱下載中的報告中的 AWS Artifact。

使用 Amazon CloudWatch Logs 時的合規責任取決於資料的敏感度、公司的合規目標以及適用的法律和法規。 AWS 提供下列資源以協助遵循法規：

- 安全與合規快速入門指南：這些部署指南討論架構考量，並提供在 AWS上部署以安全及合規為重心之基準環境的步驟。

- 在 Amazon Web Services 上架構 HIPAA 安全性與合規性 — 本白皮書說明公司如何使用建立符合 HIPAA 標準的應 AWS 用程式。
- AWS 合規資源AWS — 此工作簿和指南集合可能適用於您的產業和所在地。
- 使用AWS Config 開發人員指南中的規則評估資源 — AWS Config；評估您的資源配置如何符合內部實踐，業界準則和法規。
- AWS Security Hub— 此 AWS 服務提供安全狀態的全面檢視，協助您檢查您 AWS 是否符合安全性產業標準和最佳做法。

Amazon CloudWatch Logs 中的復原功能

AWS 全球基礎設施是以 AWS 區域與可用區域為中心建置的。區域提供多個分開且隔離的實際可用區域，並以低延遲、高輸送量和高度備援網路連線相互連結。透過可用區域，您可以設計與操作的應用程式和資料庫，在可用區域之間自動容錯移轉而不會發生中斷。可用區域的可用性、容錯能力和擴充能力，均較單一或多個資料中心的傳統基礎設施還高。

如需 AWS 區域與可用區域的詳細資訊，請參閱 AWS 全球基礎設施。

Amazon CloudWatch 日誌中的基礎設施安全

作為受管服務，Amazon CloudWatch Logs 受到 AWS 全球網路安全的保護。有關 AWS 安全服務以及如何 AWS 保護基礎結構的詳細資訊，請參閱AWS 雲端安全 若要使用基礎架構安全性的最佳做法來設計您的 AWS 環境，請參閱安全性支柱架構良 AWS 好的架構中的基礎結構保護。

您可以使用 AWS 已發佈的 API 呼叫透過網路存取 CloudWatch 記錄檔。用戶端必須支援下列項目：

- Transport Layer Security (TLS)。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 具備完美轉送私密(PFS)的密碼套件，例如 DHE (Ephemeral Diffie-Hellman)或 ECDHE (Elliptic Curve Ephemeral Diffie-Hellman)。現代系統(如 Java 7 和更新版本)大多會支援這些模式。

此外，請求必須使用存取索引鍵 ID 和與 IAM 主體相關聯的私密存取金鑰來簽署。或者，您可以使用 AWS Security Token Service (AWS STS) 以產生暫時安全憑證以簽署請求。

將記 CloudWatch 錄檔與介面 VPC 端點搭配使用

如果您使用 Amazon Virtual Private Cloud (Amazon VPC) 託管資 AWS 源，則可以在 VPC 和 CloudWatch 日誌之間建立私有連接。您可以使用此連接將日誌發送到日 CloudWatch 誌，而無需通過互聯網發送日誌。

Amazon VPC 是一項 AWS 服務，可用於在您定義的虛擬網路中啟動 AWS 資源。您可利用 VPC 來控制您的網路設定，例如 IP 地址範圍、子網路、路由表和網路閘道。若要將 VPC 連線到 CloudWatch 記錄檔，請 CloudWatch 為記錄定義介面 VPC 端點。這類端點可讓您將 VPC 連線到 AWS 服務。端點為 CloudWatch 記錄檔提供可靠、可擴充的連線，而不需要網際網路閘道、網路位址轉譯 (NAT) 執行個體或 VPN 連線。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的什麼是 Amazon VPC。

介面 VPC 私人雲端端點的支援是一種 AWS 技術 AWS PrivateLink，可使用具有私有 IP 位址的 elastic network interface，在 AWS 服務之間進行私人通訊。如需詳細資訊，請參閱新增 — AWS PrivateLink 適用於 AWS 服務。

下列步驟適用於 Amazon VPC 的使用者。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的入門。

可用性

CloudWatch 記錄檔目前支援所有 AWS 區域 (包括區域) 的 AWS GovCloud (US) VPC 端點。

建立記錄檔的 CloudWatch VPC 端點

若要開始將 CloudWatch 記錄與 VPC 搭配使用，請 CloudWatch 為記錄檔建立介面 VPC 端點。服務選擇為 com.amazonaws.***Region***.logs。您不需要變更 CloudWatch 記錄檔的任何設定。如需詳細資訊，請參閱 Amazon VPC 使用者指南中的建立界面端點。

測試 VPC 和 CloudWatch 記錄檔之間的連線

建立端點後，您可以測試連線。

測試您的 VPC 和 CloudWatch Logs 端點之間的連線

1. 連線至位於 VPC 中的 Amazon EC2 執行個體。如需連線的詳細資訊，請參閱 Amazon EC2 文件中的連線至 Linux 執行個體或連線至 Windows 執行個體。
2. 在執行個體中，使 AWS CLI 用在其中一個現有的記錄群組中建立記錄項目。

   首先，以日誌事件建立一個 JSON 檔案。時間戳記必須以從 1970 年 1 月 1 日 00:00:00 UTC 之後的毫秒數指定。

```
[
  {
```

```
    "timestamp": 1533854071310,
    "message": "VPC Connection Test"
  }
]
```

然後使用 put-log-events 命令來建立日誌項目：

```
aws logs put-log-events --log-group-name LogGroupName --log-stream-
name LogStreamName --log-events file://JSONFileName
```

如果回應命令包含 nextSequenceToken，則該命令已成功而且您的 VPC 端點是正常運作的。

控制對 CloudWatch 日誌 VPC 端點的訪問

當您建立或修改端點時，VPC 端點政策是您連接至端點的 IAM 資源政策。如果您未在建立端點時連接政策，我們會以預設政策連接以允許完整存取服務。端點政策不會覆寫或取代 IAM 政策或服務特定的政策。這個另行區分的政策會控制從端點到所指定之服務的存取。

端點政策必須以 JSON 格式撰寫。

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的使用 VPC 端點控制服務的存取。

以下是 CloudWatch 記錄檔的端點策略範例。此原則可讓使用者透過 VPC 連線至 CloudWatch 記錄檔，建立記錄串流並將記錄檔傳送至 CloudWatch 記錄檔，並防止他們執行其他 CloudWatch 記錄動作。

```
{
  "Statement": [
    {
      "Sid": "PutOnly",
      "Principal": "*",
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

修改記錄檔的 VPC 端點策略 CloudWatch

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格中選擇 Endpoints (端點)。
3. 如果尚未為 CloudWatch 記錄檔建立端點，請選擇「建立端點」。然後選取 com.amazonaws.***Region***.logs，然後選擇 Create endpoint (建立端點)。
4. 選取 com.amazonaws.***Region***.logs 端點，然後選擇螢幕下半部的 Policy (政策) 標籤。
5. 選擇 Edit Policy (編輯政策)，並對政策做出變更。

VPC 內容金鑰支援

CloudWatch 記錄檔支援可限制存取特定 VPC 或特定 VPC 端點的`aws:SourceVpce`內容金鑰`aws:SourceVpc`和內容金鑰。這些金鑰只有在使用者使用 VPC 端點時才會運作。如需詳細資訊，請參閱《IAM 使用者指南》中的可用於部分服務的金鑰。

記錄 CloudWatch 日誌 API 和控制台操作 AWS CloudTrail

Amazon CloudWatch Logs 與這項服務整合在一起 AWS CloudTrail，可提供使用者、角色或服務在日誌中所採取的動作記 CloudWatch 錄的 AWS 服務。 CloudTrail 擷取您帳戶或代表您 AWS 帳戶進行的 API 呼叫。擷取的呼叫包括來自 CloudWatch 主控台的呼叫，以及對 CloudWatch 記錄 API 作業的程式碼呼叫。如果您建立追蹤，您可以啟用持續交付 CloudTrail事件到 Amazon S3 儲存貯體，包括 CloudWatch 日誌的事件。如果您未設定追蹤，您仍然可以在 [事件歷程記錄] 中檢視 CloudTrail 主控台中最近的事件。使用收集的資訊 CloudTrail，您可以判斷對 CloudWatch Logs 提出的要求、提出要求的 IP 位址、提出要求的人員、提出要求的時間，以及其他詳細資訊。

若要進一步了解 CloudTrail，包括如何設定和啟用它，請參閱AWS CloudTrail 使用者指南。

主題

- CloudWatch 記錄資訊 CloudTrail
- 查詢產生資訊 CloudTrail
- 了解 日誌檔案項目

CloudWatch 記錄資訊 CloudTrail

CloudTrail 在您創建 AWS 帳戶時，您的帳戶已啟用。當支援的事件活動發生在記錄 CloudWatch 檔中時，該活動會與 CloudTrail 事件歷史記錄中的其他 AWS 服務事件一起記錄在事件中。您可以在帳戶中查看，搜索和下載最近的事 AWS 件。如需詳細資訊，請參閱檢視具有事 CloudTrail 件記錄的事件。

如需 AWS 帳戶中持續的事件記錄 (包括記 CloudWatch 錄的事件)，請建立追蹤。追蹤可 CloudTrail 將日誌檔交付到 Amazon S3 儲存貯體。根據預設，當您在主控台中建立追蹤時，追蹤會套用至所有 AWS 區域。追蹤記錄來自 AWS 分區中所有區域的事件，並將日誌檔傳送到您指定的 Amazon S3 儲存貯體。此外，您還可以設定其他 AWS 服務，以進一步分析 CloudTrail 記錄中收集的事件資料並採取行動。如需詳細資訊，請參閱下列內容：

- 建立追蹤的概觀
- CloudTrail 支援的服務與整合
- 設定的 Amazon SNS 通知 CloudTrail
- 從多個區域接收 CloudTrail 記錄檔並從多個帳戶接收 CloudTrail記錄檔

CloudWatch 記錄檔支援將下列動作記錄為記 CloudTrail 錄檔中的事件：

- CancelExportTask
- CreateExportTask
- CreateLogGroup
- CreateLogStream
- DeleteDestination
- DeleteLogGroup
- DeleteLogStream
- DeleteMetricFilter
- DeleteRetentionPolicy
- DeleteSubscriptionFilter
- PutDestination
- PutDestinationPolicy
- PutMetricFilter
- PutResourcePolicy
- PutRetentionPolicy
- PutSubscriptionFilter
- StartQuery
- StopQuery
- TestMetricFilter

只會 CloudTrail 針對下列 CloudWatch 記錄 API 動作登入要求元素：

- DescribeDestinations
- DescribeExportTasks
- DescribeLogGroups
- DescribeLogStreams
- DescribeMetricFilters
- DescribeQueries
- DescribeResourcePolicies
- DescribeSubscriptionFilters
- FilterLogEvents

- GetLogEvents
- GetLogGroupFields
- GetLogRecord
- GetQueryResults

每一筆事件或日誌專案都會包含產生請求者的資訊。身分資訊可協助您判斷下列事項：

- 該請求是否使用根或 IAM 使用者憑證提出。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 請求是否由其他 AWS 服務提出。

如需詳細資訊，請參閱CloudTrail 使 userIdentity 元素。

查詢產生資訊 CloudTrail

CloudTrail 也支援查詢產生器主控台事件的記錄。 CloudWatch 日誌見解和 CloudWatch 指標見解目前支援查詢產生器。在這些 CloudTrail 事件中，`eventSource`是`monitoring.amazonaws.com`.

下列範例顯示示範日 CloudTrail 誌深入解析中GenerateQuery動作的 CloudWatch 記錄項目。

```
{
    "eventVersion": "1.09",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:assumed-role/role_name",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::111222333444:role/Administrator",
                "accountId": "123456789012",
                "userName": "SAMPLE_NAME"
            },
            "attributes": {
                "creationDate": "2020-04-08T21:43:24Z",
                "mfaAuthenticated": "false"
```

```
            }
        }
    },
    "eventTime": "2020-04-08T23:06:30Z",
    "eventSource": "monitoring.amazonaws.com",
    "eventName": "GenerateQuery",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "exampleUserAgent",
    "requestParameters": {
        "query_ask": "***",
        "query_type": "LogsInsights",
        "logs_insights": {
            "fields": "***",
            "log_group_names": ["yourloggroup"]
        },
        "include_description": true
    },
    "responseElements": null,
    "requestID": "2f56318c-cfbd-4b60-9d93-1234567890",
    "eventID": "52723fd9-4a54-478c-ac55-1234567890",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

了解 日誌檔案項目

追蹤是一種組態，可讓事件以日誌檔的形式傳遞到您指定的 Amazon S3 儲存貯體。 CloudTrail 記錄檔包含一或多個記錄項目。事件代表來自任何來源的單一請求，包括有關請求的操作，動作的日期和時間，請求參數等信息。 CloudTrail 日誌文件不是公共 API 調用的有序堆棧跟踪，因此它們不會以任何特定順序顯示。

下列記錄檔項目顯示使用者呼叫「 CloudWatch 記錄檔」CreateExportTask動作。

```
{
        "eventVersion": "1.03",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
```

```
            "arn": "arn:aws:iam::123456789012:user/someuser",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "userName": "someuser"
        },
        "eventTime": "2016-02-08T06:35:14Z",
        "eventSource": "logs.amazonaws.com",
        "eventName": "CreateExportTask",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "aws-sdk-ruby2/2.0.0.rc4 ruby/1.9.3 x86_64-linux Seahorse/0.1.0",
        "requestParameters": {
            "destination": "yourdestination",
            "logGroupName": "yourloggroup",
            "to": 123456789012,
            "from": 0,
            "taskName": "yourtask"
        },
        "responseElements": {
            "taskId": "15e5e534-9548-44ab-a221-64d9d2b27b9b"
        },
        "requestID": "1cd74c1c-ce2e-12e6-99a9-8dbb26bd06c9",
        "eventID": "fd072859-bd7c-4865-9e76-8e364e89307c",
        "eventType": "AwsApiCall",
        "apiVersion": "20140328",
        "recipientAccountId": "123456789012"
}
```

CloudWatch 記錄用戶端參考

> **Important**
>
> 此參考適用於舊版已停用的 CloudWatch Logs 代理程式。如果您使用執行個體中繼資料服務版本 2 (IMDSv2)，則必須使用新的整合 CloudWatch 代理程式。即使您未使用 IMDSv2，我們強烈建議您使用較新的整合 CloudWatch 代理程式，而不是舊版的記錄代理程式。如需更新整合代理程式的詳細資訊，請參閱使用代理程式從 Amazon EC2 執行個體和現場部署伺服器收集指標和日誌。 CloudWatch
> 如需從舊版 CloudWatch Logs 代理程式移轉至整合代理程式的相關資訊，請參閱使用精靈建立 CloudWatch 代理程式組態檔。

CloudWatch 日誌代理程式提供從 Amazon EC2 執行個體將日誌資料傳送到 CloudWatch 日誌的自動化方式。代理程式包含下列元件：

- 將記錄資料推送至 CloudWatch 記錄檔的 AWS CLI 外掛程式。
- 啟動程序以將資料推送至 CloudWatch 記錄檔的程序檔 (協助程式)。
- 確保協助程式持續執行的 Cron 工作。

代理程式組態檔案

記 CloudWatch 錄代理程式組態檔描述 CloudWatch 記錄代理程式所需的資訊。代理程式組態檔案的 [general] 區段定義了通用組態，而這些組態會套用到所有日誌串流。[logstream] 區段定義了將本機檔案傳送到遠端日誌串流時所需的資訊。您可以擁有多個 [logstream] 區段，但每個區段在組態檔案中都必須有唯一的名稱，例如，[logstream1]、[logstream2]，以此類推。[logstream] 值與日誌檔中的第一行資料，用於定義日誌檔的身分。

```
[general]
state_file = value
logging_config_file = value
use_gzip_http_content_encoding = [true | false]

[logstream1]
log_group_name = value
log_stream_name = value
```

```
datetime_format = value
time_zone = [LOCAL|UTC]
file = value
file_fingerprint_lines = integer | integer-integer
multi_line_start_pattern = regex | {datetime_format}
initial_position = [start_of_file | end_of_file]
encoding = [ascii|utf_8|..]
buffer_duration = integer
batch_count = integer
batch_size = integer

[logstream2]
...
```

state_file

指定狀態檔案的存放位置。

logging_config_file

(選用) 指定代理程式日誌組態檔案的位置。如果您未在此指定代理程式日誌組態檔案，將使用預設檔案 awslogs.conf。如果您以指令碼安裝代理程式，預設的檔案位置是 `/var/awslogs/etc/awslogs.conf`，如果以 rpm 安裝代理程式，則是 `/etc/awslogs/awslogs.conf`。該文件是 Python 配置文件格式（ https://docs.python.org/2/library/logging.config.html ＃logging-config-fileformat ）。您可以自訂具有以下名稱的記錄器。

```
cwlogs.push
cwlogs.push.reader
cwlogs.push.publisher
cwlogs.push.event
cwlogs.push.batch
cwlogs.push.stream
cwlogs.push.watcher
```

以下範例會將讀取者和發佈者的層級變更為 WARNING，而預設值為 INFO。

```
[loggers]
keys=root,cwlogs,reader,publisher

[handlers]
keys=consoleHandler
```

```
[formatters]
keys=simpleFormatter

[logger_root]
level=INFO
handlers=consoleHandler

[logger_cwlogs]
level=INFO
handlers=consoleHandler
qualname=cwlogs.push
propagate=0

[logger_reader]
level=WARNING
handlers=consoleHandler
qualname=cwlogs.push.reader
propagate=0

[logger_publisher]
level=WARNING
handlers=consoleHandler
qualname=cwlogs.push.publisher
propagate=0

[handler_consoleHandler]
class=logging.StreamHandler
level=INFO
formatter=simpleFormatter
args=(sys.stderr,)

[formatter_simpleFormatter]
format=%(asctime)s - %(name)s - %(levelname)s - %(process)d - %(threadName)s -
 %(message)s
```

use_gzip_http_content_encoding

當設定為 true (預設值) 時，啟用 gzip http 內容編碼，將壓縮的承載傳送至 CloudWatch 記錄檔。這會降低 CPU 使用率 NetworkOut、降低並減少置入延遲。若要停用此功能，請將 use_gzip_http_content_encoding = 假新增至 CloudWatch 記錄檔代理程式組態檔的 [一般] 區段，然後重新啟動代理程式。

> **Note**
>
> 此設定僅適用於 awscli-cwlogs 1.3.3 及更新版本。

log_group_name

指定目的地日誌群組。如果日誌群組尚未存在，將會自動建立。日誌群組的名稱長度可介於 1 到 512 個字元之間。可用的字元為 a-z、A-Z、0-9、'_' (底線)、'-' (連字號)、'/' (正斜線) 和 '.' (句點)。

log_stream_name

指定目的地日誌串流。您可以使用常值字串、預先定義的變數 ({instance_id}、{hostname}、{ip_address}) 或這兩者的組合，以定義日誌串流名稱。如果日誌串流尚未存在，將會自動建立。

datetime_format

指定從日誌擷取時間戳記的方式。此時間戳記用於擷取日誌事件及產生指標。若未提供 datetime_format，目前時間將用於每個日誌事件。如果提供的 datetime_format 值對於指定的日誌訊息而言是無效的，這時將使用最近一次所含時間戳記成功剖析之日誌事件的時間戳記。如果沒有之前的日誌事件，將使用目前時間。

以下列出常見的 datetime_format 代碼。您也可以使用任何 Python 支援的 datetime_format 代碼、datetime.strptime()。亦支援時區位移 (%z)，雖然 Python 3.2 之前版本並不支援，[+-] HHMM 無需冒號 (:)。如需詳細資訊，請參閱 strftime () 和 strptime () 行為。

%y：年，不包含以填充零之十進位表示的世紀數字。00、01、…、99

%Y：年，包含以十進位表示的世紀數字。1970、1988、2001、2013

%b：月，當地的縮寫名稱。Jan、Feb、…、Dec (en_US)；

%B：月，當地的完整名稱。January、February、…、December (en_US)；

%m：月，填充零的十進位數字。01、02、…、12

%d：日，填充零的十進位數字。01、02、…、31

%H：小時 (24 小時制)，填充零的十進位數字。00、01、…、23

%I：小時 (12 小時制)，填充零的十進位數字。01、02、…、12

%p：相當於當地的 AM 或 PM。

%M：分鐘，填充零的十進位數字。00、01、…、59

%S：秒鐘，填充零的十進位數字。00、01、...、59

%f：微秒，十進位數字，左側填充零。000000、...、999999

%z：UTC 位移，格式為 +HHMM 或 -HHMM。+0000、-0400、+1030

範例格式：

```
Syslog: '%b %d %H:%M:%S', e.g. Jan 23 20:59:29

Log4j: '%d %b %Y %H:%M:%S', e.g. 24 Jan 2014 05:00:00

ISO8601: '%Y-%m-%dT%H:%M:%S%z', e.g. 2014-02-20T05:20:20+0000
```

time_zone

指定日誌事件時間戳記的時區。支援的兩個值為 UTC 和 LOCAL。如果無法依據 datetime_format 推斷時區，預設值為 LOCAL。

file

指定您要推送至記錄檔的記 CloudWatch 錄檔。檔案可指向特定檔案或多個檔案 (使用萬用字元，例如 /var/log/system.log*)。只有最新的檔案會根據檔案修改時間推送至 CloudWatch 記錄檔。我們建議您使用萬用字元來指定一系列的相同類型的檔案，例如 access_log.2014-06-01-01、access_log.2014-06-01-02，以此類推，但不適用於多種種類的檔案，例如 access_log_80 和 access_log_443。若要指定多種種類的檔案，可將另一個日誌串流新增至組態檔案，讓每個種類的日誌檔進入不同的日誌串流。不支援壓縮檔案。

file_fingerprint_lines

指定用於識別檔案的行範圍。有效值是一個數字或兩個以破折號分隔的數字，例如「1」、「2-5」。預設值為「1」，因此第一行用於計算指紋。除非所有指定的行都可用，否則指紋線路不會傳送至 CloudWatch 記錄檔。

multi_line_start_pattern

指定用於識別日誌訊息開始處的模式。日誌訊息是由符合模式的一列及不符合模式的任何幾列所組成。有效值為規則表達式或 {datetime_format}。使用 {datetime_format} 時，應指定 datetime_format 選項。預設值為「^ [^\s]」，因此開頭使用非空白字元的任何列皆可結束之前的日誌訊息，並開始新的日誌訊息。

initial_position

指定開始讀取資料 (start_of_file 或 end_of_file) 的位置。預設值為 start_of_file。只有在日誌串流沒有狀態時才會使用它。

編碼

指定日誌檔的編碼，以便正確讀取檔案。預設值為 utf_8。這裡可以使用 Python codecs.decode () 支援的編碼。

> **Warning**
> 指定不正確的編碼可能導致資料遺失，因為無法解碼的字元會被替換為一些其他的字元。

以下是一些常見的編碼：

```
ascii, big5, big5hkscs, cp037, cp424, cp437, cp500, cp720, cp737,
cp775, cp850, cp852, cp855, cp856, cp857, cp858, cp860, cp861, cp862,
cp863, cp864, cp865, cp866, cp869, cp874, cp875, cp932, cp949, cp950,
cp1006, cp1026, cp1140, cp1250, cp1251, cp1252, cp1253, cp1254, cp1255,
cp1256, cp1257, cp1258, euc_jp, euc_jis_2004, euc_jisx0213, euc_kr,
gb2312, gbk, gb18030, hz, iso2022_jp, iso2022_jp_1, iso2022_jp_2,
iso2022_jp_2004, iso2022_jp_3, iso2022_jp_ext, iso2022_kr, latin_1,
iso8859_2, iso8859_3, iso8859_4, iso8859_5, iso8859_6, iso8859_7,
iso8859_8, iso8859_9, iso8859_10, iso8859_13, iso8859_14, iso8859_15,
iso8859_16, johab, koi8_r, koi8_u, mac_cyrillic, mac_greek, mac_iceland,
mac_latin2, mac_roman, mac_turkish, ptcp154, shift_jis, shift_jis_2004,
shift_jisx0213, utf_32, utf_32_be, utf_32_le, utf_16, utf_16_be,
utf_16_le, utf_7, utf_8, utf_8_sig
```

buffer_duration

指定日誌事件的批次處理的持續時間。最小值為 5000ms，預設值為 5000ms。

batch_count

指定批次中的日誌事件最大數量，最多可達 10,000。預設值為 10000。

batch_size

指定批次中的日誌事件最大大小，以位元組為單位，最多可達 1048576 位元組。預設值為 1048576 位元組。這個大小的計算方式是以所有 UTF-8 事件訊息，加上每個記錄事件 26 個位元組。

搭配 HTTP 代理伺服器使用 CloudWatch 記錄代理程式

您可以將 CloudWatch 記錄代理程式與 HTTP 代理伺服器搭配使用。

> **Note**
>
> 在 awslogs-agent-setup .py 版本 1.3.8 或更高版本中支持 HTTP 代理服務器。

將 CloudWatch 記錄代理程式與 HTTP 代理伺服器搭配使用

1. 執行以下任意一項：

   a. 對於 Lo CloudWatch gs 代理程式的新安裝，請執行下列命令：

   ```
   curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-setup.py -O
   ```

   ```
   sudo python awslogs-agent-setup.py --region us-east-1 --http-proxy http://your/proxy --https-proxy http://your/proxy --no-proxy 169.254.169.254
   ```

   為了維持存取 EC2 執行個體上的 Amazon EC2 中繼資料服務，請使用 --no-proxy 169.254.169.254 (建議)。如需詳細資訊，請參閱 Amazon EC2 使用者指南中的執行個體中繼資料和使用者資料。

   在 `http-proxy` 和 `https-proxy` 的數值中，您必須指定整個 URL。

   b. 對於 CloudWatch 日誌代理程式的現有安裝，請編輯 /var/awslogs/etc/proxy.conf，然後新增您的代理伺服器：

   ```
   HTTP_PROXY=
   HTTPS_PROXY=
   NO_PROXY=
   ```

2. 重新啟動代理程式，讓變更生效：

   ```
   sudo service awslogs restart
   ```

   如果您使用的是 Amazon Linux 2，請使用下列命令來重新啟動代理程式：

```
sudo service awslogsd restart
```

分隔記錄檔代理程式組 CloudWatch 態檔

如果您在 awscli-cwlogs 1.3.3 或更新版本中使用 awslogs-agent-setup .py 版本 1.3.8 或更新版本，您可以在 /var/awslogs/etc/config/ 目錄中建立額外的配置檔案，為不同的元件匯入不同的串流設定。CloudWatch Logs 代理程式啟動時，會在這些額外的設定檔中包含任何串流組態。[general] 區段中的組態屬性必須在主要組態檔案 (/var/awslogs/etc/awslogs.conf) 中定義，並且在 /var/awslogs/etc/config/ 中的任何額外的組態檔案中被忽略。

如果您因為使用 rpm 安裝代理程式，因此沒有 /var/awslogs/etc/config/ 目錄，您可以改為使用 /etc/awslogs/config/ 目錄。

重新啟動代理程式，讓變更生效：

```
sudo service awslogs restart
```

如果您使用的是 Amazon Linux 2，請使用下列命令來重新啟動代理程式：

```
sudo service awslogsd restart
```

CloudWatch 記錄檔代理程式

支援哪些種類的檔案輪換？

支援以下檔案輪換機制：

- 以數值尾碼重新命名現有的日誌檔，然後重新建立原始的空日誌檔。例如，/var/log/syslog.log 重新命名為 /var/log/syslog.log.1. 如果 /var/log/syslog.log.1 從之前的輪換就已存在，它將會重新命名為 /var/log/syslog.log.2。
- 在建立副本之後，截斷已備妥的原始日誌檔。例如，/var/log/syslog.log 將會複製到 /var/log/syslog.log.1，/var/log/syslog.log 將被截斷。在這種情況下，資料可能會遺失，因此請留意使用此檔案輪換機制。
- 使用與舊檔案相同的通用模式建立新檔案。例如，保留 /var/log/syslog.log.2014-01-01，並建立 /var/log/syslog.log.2014-01-02。

檔案的指紋 (來源 ID) 的計算方式是雜湊日誌串流金鑰與檔案的第一行內容。若要覆寫此行為，可使用 file_fingerprint_lines 選項。發生檔案輪換時，新的檔案應該會有新的內容，舊的檔案不應該有附加的內容；代理程式會在完成讀取舊檔案之後推送新的檔案。

我如何判斷我使用的代理程式是哪個版本？

如果您使用安裝程式指令碼來安裝 CloudWatch 記錄代理程式，您可以使用 /var/awslogs/bin/awslogs-version.sh 來檢查您所使用的代理程式版本。它會列印出代理程式的版本及其主要相依性。如果你使用 yum 來安裝 CloudWatch 日誌代理程式，你可以使用 「yum info awslog」 和「yum info aws-cli-plugin-cloudwatch-日誌」來檢查日 CloudWatch 誌代理程式和外掛程式的版本。

日誌項目如何轉換為日誌事件？

日誌事件包含兩個屬性：事件發生時的時間戳記，以及原始日誌訊息。依據預設，開頭使用非空白字元的任何列皆可結束之前的日誌訊息 (如果有的話)，並開始新的日誌訊息。若要覆寫此行為，可以使用 multi_line_start_pattern，符合此模式的任何列都會開始新的日誌訊息。此模式可以是任何 regex 或「{datetime_format}」。例如，如果每個日誌訊息的第一行包含時間戳記，例如 '2014-01-02T13:13:01Z'，則 multi_line_start_pattern 可設定為 '\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}Z'。為了簡化組態，如果已指定 datetime_format option，您可以使用「{datetime_format}」變數。以相同的範例而言，如果 datetime_format 設為 '%Y-%m-%dT%H:%M:%S%z'，則 multi_line_start_pattern 可以只是「{datetime_format}」。

若未提供 datetime_format，目前時間將用於每個日誌事件。如果提供的 datetime_format 對於指定的日誌訊息而言是無效的，將使用最後一個成功剖析時間戳記的日誌事件的時間戳記。如果沒有之前的日誌事件，將使用目前時間。當日誌事件回退至目前時間或之前的日誌事件時間，將會記錄警告訊息。

時間戳記用於擷取日誌事件並產生指標，因此如果您指定錯誤的格式，可能會導致無法擷取日誌事件，並且會產生錯誤的指標。

日誌事件會如何進行批次處理？

當下列任何一個條件成立時，批次將變滿並予以發佈：

1. 從新增第一個日誌事件以來，已經過 buffer_duration 的時間長度。
2. 已累積小於 batch_size 的日誌事件，但新增超過 batch_size 的新日誌事件。
3. 達到 batch_count 的日誌事件數量。
4. 此批次的日誌事件未持續超過 24 小時，但新增新日誌事件過程超過 24 小時限制。

什麼會導致日誌項目、日誌事件，或批次遭到略過或截斷？

若要遵循 `PutLogEvents` 操作的限制，以下問題可能會導致日誌事件或批次進行被略過。

> **Note**
>
> 當略過資料時， CloudWatch Logs 代理程式會在其記錄檔中寫入警告。

1. 如果日誌事件的大小超過 256 KB，將會完全略過該日誌事件。
2. 如果日誌事件的時間戳記超過未來 2 小時，將會略過該日誌事件。
3. 如果日誌事件的時間戳記超過過去 14 天，將會略過該日誌事件。
4. 如有任何日誌事件超過日誌群組的保留期間，將會略過整個批次。
5. 如果在單一 `PutLogEvents` 請求中的日誌事件批次持續超過 24 小時，則 `PutLogEvents` 操作會失敗。

停用代理程式是否會造成資料遺失/重複？

只要有狀態檔案，而且從上次執行之後未發生檔案輪換，就不會造成資料遺失/重複。Lo CloudWatch gs 代理程式可以從停止的位置開始，並繼續推送記錄檔資料。

我是否可以從相同或不同的主機，將不同的日誌檔指向相同的日誌串流？

不支援設定多個日誌來源，將資料傳送到單個日誌串流。

代理程式發出哪些 API 呼叫 (或我應該將哪些動作新增至 IAM 政策)？

記 CloudWatch 錄代理程式需要`CreateLogGroup`、`CreateLogStream`、`DescribeLogStreams`、和`PutLogEvents`作業。如果您使用的是最新的代理程式，則不需要 `DescribeLogStreams`。請參閱以下 IAM 政策範例。

```
{
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "logs:CreateLogGroup",
      "logs:CreateLogStream",
      "logs:PutLogEvents",
      "logs:DescribeLogStreams"
    ],
    "Resource": [
      "arn:aws:logs:*:*:*"
    ]
  }
```

```
 ]
}
```

我不想讓 CloudWatch Logs 代理程式自動建立記錄群組或記錄資料流。我要如何防止代理程式重新建立日誌群組與日誌串流？

您可以在 IAM 政策中限制代理程式只能執行以下操作：DescribeLogStreams、PutLogEvents。

從代理程式撤銷 CreateLogGroup 和 CreateLogStream 許可前，請務必建立您要讓代理程式使用的日誌群組和日誌串流。日誌代理程式無法在您已建立的日誌群組中建立日誌串流，除非其擁有 CreateLogGroup 和 CreateLogStream 許可。

進行故障排除時，我應該查看哪些日誌？

代理程式安裝日誌位於 /var/log/awslogs-agent-setup.log，代理程式日誌位於 /var/log/awslogs.log。

使用 CloudWatch 指標監控

CloudWatch 日誌會 CloudWatch 每分鐘將指標傳送到 Amazon。

CloudWatch 記錄指標

AWS/Logs 命名空間包含下列指標。

指標	描述
CallCount	在您的帳戶中執行的特定 API 操作數目。 CallCount 是 CloudWatch 記錄檔服務使用狀況測量結果。如需詳細資訊，請參閱 CloudWatch 記錄服務使用量度。 有效維度：Class, Resource, Service, Type 有效統計資訊：總和 單位：無
DeliveryErrors	將資料轉送至訂閱目的地時， CloudWatch 記錄檔收到錯誤的記錄事件數目。如果目的地服務傳回可重試的錯誤，例如節流例外狀況或可重試的服務例外狀況 (例如 HTTP 5xx)，則 CloudWatch 記錄檔會持續重試傳遞最多 24 小時。 CloudWatch 如果錯誤是無法重試的錯誤，例如或，記錄檔不會嘗試重新傳送。AccessDeniedException ResourceNotFoundException 有效尺寸: LogGroupName, DestinationType, FilterName, PolicyLevel 有效統計資訊：總和 單位：無
DeliveryThrottling	將資料轉送至訂閱目的地時，已限制 CloudWatch 記錄的記錄事件數目。 如果目的地服務傳回可重試的錯誤，例如節流例外狀況或可重試的服務例外狀況 (例如 HTTP 5xx)，則 CloudWatch 記錄檔會持續重試傳遞最多 24 小時。 CloudWatch 如果錯誤是無法重試的錯誤，例如或，記錄檔不會嘗

指標	描述
	試重新傳送。AccessDeniedException ResourceNotFoundException 有效尺寸: LogGroupName, DestinationType, FilterName, PolicyLevel 有效統計資訊：總和 單位：無
EMFParsingErrors	處理內嵌指標格式日誌時遇到的剖析錯誤數量。如果日誌識別為內嵌指標格式，但未遵循正確格式，就會發生這類錯誤。如需有關內嵌指標格式的詳細資訊，請參閱規格：內嵌指標格式。 有效維度：LogGroupName 有效統計資訊：總和 單位：無
EMFValidationErrors	處理內嵌指標格式日誌時遇到的驗證錯誤數量。如果內嵌指標格式日誌中的指標定義不符合內嵌指標格式和 MetricDatum 規格，就會發生這類錯誤。如需 CloudWatch 內嵌量度格式的相關資訊，請參閱規格：內嵌量度格式。如需有關資料類型的資訊MetricDatum ，請參閱 Amazon CloudWatch API 參考MetricDatum中的。 **Note** 某些驗證錯誤可能會導致 EMF 日誌中的多個指標無法發佈。例如，將捨棄具有無效命名空間的所有指標集。 有效維度：LogGroupName 有效統計資訊：總和 單位：無

指標	描述
ErrorCount	在您的帳戶中執行而導致錯誤的 API 操作數目。 ErrorCount 是 CloudWatch 記錄檔服務使用狀況測量結果。如需詳細資訊，請參閱 CloudWatch 記錄服務使用量度。 有效維度：Class, Resource, Service, Type 有效統計資訊：總和 單位：無
ForwardedBytes	轉送至訂閱目的地的壓縮位元組中的日誌事件量。 有效尺寸: LogGroupName, DestinationType, FilterName 有效統計資訊：總和 單位：位元組
ForwardedLogEvents	轉送至訂閱目的地的日誌事件數量。 有效尺寸: LogGroupName, DestinationType, FilterName, PolicyLevel 有效統計資訊：總和 單位：無
IncomingBytes	上傳至 CloudWatch 記錄檔的未壓縮位元組記錄事件數量。與 LogGroupName 維度搭配使用時，此為已上傳至日誌群組的未壓縮位元組中的日誌事件量。 有效尺寸: LogGroupName 有效統計資訊：總和 單位：位元組

指標	描述
IncomingLogEvents	上傳至 CloudWatch 記錄檔的記錄事件數目。與 LogGroupName 維度搭配使用時，此為已上傳至日誌群組的未壓縮位元組中的日誌事件數量。 有效尺寸: LogGroupName 有效統計資訊：總和 單位：無
LogEventsWithFindings	使用記錄檔資料保護功能與您正在稽核之資料字串相符的 CloudWatch 記錄事件數目。如需詳細資訊，請參閱 使用遮罩功能協助保護敏感日誌資料。 有效維度：無 有效統計資訊：總和 單位：無
ThrottleCount	在您的帳戶中因用量配額而調節執行的 API 操作數目。 ThrottleCount 是 CloudWatch 記錄檔服務使用狀況測量結果。如需詳細資訊，請參閱 CloudWatch 記錄服務使用量度。 有效維度：Class, Resource, Service, Type 有效統計資訊：總和 單位：無

CloudWatch 記錄量度的維度

下表列出可搭配「 CloudWatch 記錄」量度使用的維度。

維度	描述
LogGroupName	要顯示測量結果的 CloudWatch 記錄日誌群組名稱。

維度	描述
DestinationType	CloudWatch 日誌資料的訂閱目的地 AWS Lambda，可以是 Amazon Kinesis Data Streams 或 Amazon 資料 Firehose。
FilterName	將資料從日誌群組轉送至目的地的訂閱篩選器名稱。訂閱篩選器名稱會自動轉換 CloudWatch 為 ASCII，且任何不受支援的字元都會取代為問號 (?)。

下表列出與帳戶層級訂閱篩選條件相關的指標維度。

維度	描述
PolicyLevel	套用策略的層級。目前，此維度的唯一有效值為 AccountPolicy
DestinationType	CloudWatch 日誌資料的訂閱目的地 AWS Lambda，可以是 Amazon Kinesis Data Streams 或 Amazon 資料 Firehose。
FilterName	將資料從日誌群組轉送至目的地的訂閱篩選器名稱。訂閱篩選器名稱會自動轉換 CloudWatch 為 ASCII，且任何不受支援的字元都會取代為問號 (?)。

CloudWatch 記錄服務使用量度

CloudWatch 記錄檔會傳送指標 CloudWatch ，以追蹤使用量 CloudWatch 記錄 API 作業。這些量度對應於 AWS 服務配額。追蹤這些指標可協助您主動管理配額。如需詳細資訊，請參閱 Service Quotas 整合與用量指標。

例如，您可以追蹤 ThrottleCount 指標，或在該指標設定警示。如果此指標的值上升，您應該考慮對遭調節的 API 操作要求增加配額。如需 CloudWatch 記錄檔服務配額的詳細資訊，請參閱CloudWatch 記錄配額。

CloudWatch 記錄檔會每分鐘在AWS/Usage和AWS/Logs命名空間中發佈服務配額使用量度。

下表列出「 CloudWatch 記錄檔」所發佈的服務使用狀況測量結果。這些指標沒有規定的單位。這些指標最實用的統計數字是 SUM，代表 1 分鐘期間的總操作計數。

每個指標會連同 Service、Class、Type 及Resource 全部維度的值一起發佈。也會連同稱為 Account Metrics 的單一維度一起發佈。使用 Account Metrics 維度查看帳戶中所有 API 操作的指標總和。使用其他維度，並指定 API 操作的名稱給 Resource 維度，以尋找該特定 API 的指標。

指標

指標	描述
CallCount	在您的帳戶中執行的指定操作數目。 CallCount 會發佈在 AWS/Usage 和 AWS/Logs 命名空間。
ErrorCount	在您的帳戶中執行而導致錯誤的 API 操作數目。 ErrorCount 只會發佈在 AWS/Logs。
ThrottleCount	在您的帳戶中因用量配額而調節執行的 API 操作數目。 ThrottleCount 只會發佈在 AWS/Logs。

Dimensions (尺寸)

維度	描述
Account metrics	使用此維度可取得所有 CloudWatch 記錄 API 的量度總和。 如果您想要查看某個特定 API 的指標，請使用此表格中列出的其他維度，並指定 API 名稱作為 Resource 的值。
Service	包含資源的 AWS 服務名稱。對於「 CloudWatch 記錄檔」使用量度，此維度的值為Logs。
Class	正在追蹤的資源類別。 CloudWatch 記錄 API 使用量度使用此維度的值為None。
Type	正在追蹤的資源類型。目前，當 Service 維度為 Logs，Type 的唯一有效值為 API。

維度	描述
Resource	API 操作的名稱。有效值包括所有列在動作中的 API 操作名稱。例如，PutLogEvents

CloudWatch 記錄配額

下表提供 AWS 帳戶的 CloudWatch 記錄檔預設服務配額 (也稱為限制)。這些 Service Quotas 中的大多數 (但不是全部) 都列在服務配額主控台的 Amazon CloudWatch Logs 命名空間下。若要請求提高這些配額，請參閱本節稍後的程序。

資源	預設配額
帳戶層級政策	每個帳號都有一個帳戶層級的訂閱過濾政策。 每個帳號都有一個帳戶層級的資料保護政策。 這些配額無法變更。
異常探測器	每個帳戶 10 個異常偵測器。此配額無法變更。
批次大小	批次大小上限為 1,048,576 位元組。這個大小的計算方式是以所有 UTF-8 事件訊息，加上每個記錄事件 26 個位元組。此配額無法變更。
資料存檔	高達 5 GB 的免費資料存檔。此配額無法變更。
CreateLogGroup	每秒 10 筆交易（TPS/ 帳戶/區域），之後將限制交易。您可以要求增加配額。
CreateLogStream	每秒 50 次交易 (TPS/帳戶/區域)，之後交易會受到調節。您可以要求增加配額。
自訂資料識別符	每個資料保護政策最多可包含 10 個自訂資料識別碼。您可以要求增加配額。 定義自訂資料識別碼的每個規則運算式最多可包含 200 個字元。此配額無法變更。
DeleteLogGroup	每秒 10 筆交易（TPS/ 帳戶/區域），之後將限制交易。您可以要求增加配額。
DeleteLogStream	每秒 15 筆交易（TPS/ 帳戶/區域），之後將限制交易。您可以要求增加配額。

資源	預設配額
DescribeLogGroups	每秒 10 筆交易 (TPS/ 帳戶/區域)。您可以要求增加配額。
DescribeLogStreams	每秒 25 筆交易 (TPS/ 帳戶/區域)。您可以要求增加配額。
探索的日誌欄位	CloudWatch 日誌見解可以探索記錄群組中最多 1000 個記錄事件欄位。此配額無法變更。 如需詳細資訊，請參閱 支援的日誌和探索的欄位。
擷取 JSON 日誌中的日誌欄位	CloudWatch 日誌深入解析最多可以從 JSON 記錄擷取 200 個記錄事件欄位。此配額無法變更。 如需詳細資訊，請參閱 支援的日誌和探索的欄位。
匯出任務	每個帳戶一次會有一個作用中 (正在執行或擱置中) 的匯出任務。此配額無法變更。
FilterLogEvents	美國東部 (維吉尼亞北部) 為每秒 25 個請求。 以下區域每秒有 5 個要求： • 亞太區域 (雅加達) • 亞太區域 (大阪) • 歐洲 (法蘭克福) • 加拿大西部 (卡加利) • 以色列 (特拉維夫) 其他地區每秒 10 個請求。 此配額無法變更。

資源	預設配額
GetLogEvents	歐洲 (巴黎) 為每秒 30 個請求。 以下區域為每秒 10 個請求： • 美國西部 (奧勒岡) • 亞太區域 (雅加達) • 亞太區域 (大阪) • 加拿大西部 (卡加利) • 歐洲 (愛爾蘭) • 歐洲 (法蘭克福) • 以色列 (特拉維夫) 所有其他區域每秒 25 個請求。 此配額無法變更。 如果您繼續處理新資料，建議您進行訂閱。如果您需要歷史資料，建議將資料匯出到 Amazon S3。
傳入資料	高達 5 GB 的免費傳入資料。此配額無法變更。
Live Tail 並發工作階段。	15 個並發工作階段。您可以要求增加配額。
Live Tail：在一個工作階段中搜尋的日誌群組。	在一個 Live Tail 工作階段中最多掃描 10 個日誌群組。此配額無法變更。
日誌事件大小	256 KB (上限)。此配額無法變更。
日誌群組	每個區域每個帳戶 100 萬個日誌群組。您可以要求增加配額。 可以屬於一個日誌群組的日誌串流數量並沒有配額。
指標篩選條件	每個日誌群組 100 個。此配額無法變更。

資源	預設配額
內嵌指標格式的指標	每個日誌事件 100 個指標，每個指標 30 個維度。如需有關內嵌指標格式的詳細資訊，請參閱 Amazon CloudWatch 使用者指南中的規格：內嵌指標格式。
PutLogEvents	PutLogEvents 請求的最大批次大小為 1MB。這個大小的計算方式是以所有 UTF-8 事件訊息，加上每個記錄事件 26 個位元組。 每個區域每個帳戶每秒 5000 筆交易您可以使用服務要求增加每秒節流配額。 Service Quotas
查詢執行逾時	CloudWatch 記錄檔見解中的查詢會在 60 分鐘後逾時。此時間限制無法變更。
查詢的日誌群組	在單一日誌深入解析查詢中，最多可查詢 50 個 CloudWatch 記錄群組。此配額無法變更。
查詢並行	對於標準類別記錄群組，最多 30 個並行 CloudWatch 記錄見解查詢，包括已新增至儀表板的查詢。 對於不常存取類別記錄群組，最多 5 個並行 CloudWatch 記錄見解查詢，包括已新增至儀表板的查詢。 這些配額無法變更。
從自然語言產生的查詢	多達五個並發自然語言生成的查詢請求。
查詢可用性	透過歷史記錄命令，在主控台中建構的查詢可使用 30 天。此可用性時間無法變更。 使用建立的查詢定義PutQueryDefinition不會過期。
查詢結果可用性	查詢中的結果可供擷取 7 天。此可用時間無法變更。

資源	預設配額
在主控台中顯示的查詢結果	根據預設，主控台上最多會顯示 1000 個資料列的查詢結果。您可以在查詢中使用 **`limit`** 命令將其增加至最多 10,000 個資料列。如需詳細資訊，請參閱 CloudWatch 日誌見解查詢語法。
常規表達式	建立指標篩選條件或訂閱篩選條件時，每個日誌群組最多有 5 個包含規則運算式的篩選條件模式。此配額無法變更。 針對指標篩選條件和訂閱篩選條件建立分隔或 JSON 篩選條件模式，或篩選條件日誌事件時，每個篩選條件模式最多有 2 個規則運算式。
資源政策	每個區域每個帳號最多 10 個 CloudWatch 記錄資源策略。此配額無法變更。
已儲存的查詢	每個帳戶每個區域最多可以儲存 1000 個 CloudWatch 日誌見解查詢。此配額無法變更。
訂閱篩選條件	每個日誌群組 2 個。此配額無法變更。

管理您的 CloudWatch 記錄檔服務配額

CloudWatch 記錄檔已與 Service Quotas 整合，這項 AWS 服務可讓您從中央位置檢視及管理配額。如需詳細資訊，請參閱Service Quotas 使用者指南中的什麼是 Service Quotas？。

Service Quotas 可讓您輕鬆查詢 CloudWatch 記錄服務配額的值。

AWS Management Console

使用主控台檢視 CloudWatch 記錄檔服務配額

1. 開啟 Service Quotas 主控台，網址為 https://console.aws.amazon.com/servicequotas/。
2. 在導覽窗格中，選擇 AWS services (AWS 服務)。
3. 從AWS 服務清單中搜尋並選取 Amazon CloudWatch 日誌。

在 Service quotas (服務配額) 清單中，您可以看到服務配額名稱、套用的值 (如果有的話)、AWS 預設配額，以及配額值是否可調整。

4. 若要檢視服務配額的其他資訊 (例如說明)，請選擇配額名稱。
5. (選用) 若要請求增加配額，請選取您要增加的配額、選取 Request quota increase (請求增加配額)、輸入或選取必要資訊，然後選取 Request (請求)。

若要使用主控台來進一步處理服務配額，請參閱《Service Quotas 使用者指南》。若要請求提高配額，請參閱《Service Quotas 使用者指南》中的請求提高配額。

AWS CLI

檢視 CloudWatch 記錄檔服務配額 AWS CLI

執行下列命令以檢視預設的 CloudWatch 記錄配額。

```
aws service-quotas list-aws-default-service-quotas \
    --query 'Quotas[*].
{Adjustable:Adjustable,Name:QuotaName,Value:Value,Code:QuotaCode}' \
    --service-code logs \
    --output table
```

若要使用更多使用 Service Quotas AWS CLI，請參閱服務配額 AWS CLI 命令參考。若要請求提高配額，請參閱《AWS CLI 命令參考》中的 `request-service-quota-increase` 命令。

文件歷史紀錄

下表說明從 2018 年 6 月開始，每個「CloudWatch 記錄檔使用手冊」版本中的重要變更。如需有關此文件更新的通知，您可以訂閱 RSS 訂閱源。

變更	描述	日期
CloudWatch 正式提供自然語言查詢產生的日誌見解支援	CloudWatch 日誌洞見支援自然語言來產生和更新查詢。如需詳細資訊，請參閱使用自然語言產生和更新 CloudWatch 記錄見解查詢。	2024年6月20日
CloudWatchLogsReadOnlyAccess政策已更新	CloudWatch 記錄檔已新增`cloudwatch:GenerateQuery` 權限 CloudWatchLogsReadOnlyAccess，以便具有此原則的使用者可以從自然語言提示中產生 CloudWatch Logs Insights 查詢字串。	2023 年 11 月 26 日
CloudWatchLogsFullAccess政策已更新	CloudWatch 記錄檔已新增`cloudwatch:GenerateQuery` 權限 CloudWatchLogsFullAccess，以便具有此原則的使用者可以從自然語言提示中產生 CloudWatch Logs Insights 查詢字串。	2023 年 11 月 26 日
CloudWatch 日誌添加日誌模式分析	CloudWatch 記錄檔現在會在您每次執行記 CloudWatch 錄檔見解查詢時掃描記錄事件中的模式。如需詳細資訊，請參閱陣列分析。	2023 年 11 月 26 日
CloudWatch 記錄新增記錄異常偵測	您可以為日誌群組建立日誌異常偵測器。異常偵測器會掃	2023 年 11 月 26 日

	描擷取到記錄群組中的記錄事件，並在記錄資料中尋找異常。如需詳細資訊，請參閱日誌異常偵測。	
CloudWatch 日誌添加了比較功能	您現在可以使用 CloudWatch 日誌深入解析來比較一段時間內記錄事件中的變更。如需詳細資訊，請參閱比較 (diff) 與先前時間範圍。	2023 年 11 月 26 日
CloudWatch 日誌添加了一個新的日誌類	CloudWatch 記錄檔支援兩種類別的記錄群組，因此您可以針對不常存取的記錄提供符合成本效益的選項，而且您也可以針對需要即時監控或其他功能的記錄提供完整功能選項。如需詳細資訊，請參閱日誌類別。	2023 年 11 月 26 日
CloudWatch 日誌見解支援自然語言查詢產生	CloudWatch 日誌洞見支援自然語言來產生和更新查詢。如需詳細資訊，請參閱使用自然語言產生和更新 CloudWatch 記錄見解查詢。	2023 年 11 月 26 日
CloudWatch 記錄檔新增 Live Tail 的規則運算式篩選器模式語法支援	現在，您可以在 Live Tail 篩選條件模式中使用靈活的規則表達式，進一步自訂搜尋和比對操作，以滿足您的需求。如需詳細資訊，請參閱 Amazon CloudWatch 日誌使用者指南中的篩選器模式語法。	2023 年 11 月 13 日

CloudWatch 記錄新增量度篩選器、訂閱篩選器和篩選記錄事件的規則運算式篩選器模式語法支援	現在，您可以在篩選條件模式中使用靈活的規則運算式，進一步自訂搜尋和比對操作，以滿足您的需求。如需詳細資訊，請參閱 Amazon CloudWatch 日誌使用者指南中的篩選器模式語法。	2023 年 9 月 5 日
CloudWatch 日誌洞察新增模式命令	您現在可以在 CloudWatch Logs Insights 查詢中使用模式，將日誌資料自動叢集到模式中。模式是指日誌欄位之間反覆出現的共同文字結構。如需詳細資訊，請參閱 Amazon CloudWatch 日誌使用者指南中的模式。	2023 年 7 月 17 日
CloudWatch 日誌見解添加了一個刪除命令	您現在可以在 CloudWatch 日誌見解查詢中使用 dedup，根據您指定的欄位中的特定值移除重複的結果。如需詳細資訊，請參閱 Amazon CloudWatch 日誌使用者指南中的資料刪除。	2023 年 6 月 20 日
帳戶層級資料保護政策	您現在可以在帳戶層級設定資料保護政策。這些帳戶層級政策可以稽核和遮罩帳戶中所有日誌群組中日誌事件中的敏感資訊。如需詳細資訊，請參閱 Amazon CloudWatch 日誌使用者指南中的使用遮罩協助保護敏感日誌資料。	2023 年 6 月 8 日

新增了 Live Tail 功能	CloudWatch 日誌添加了 Live Tail 能力，因此您可以掃描日誌，因為它們被攝入，以幫助進行故障排除。您可以選擇性地根據指定的詞彙來篩選顯示的日誌事件串流，也可以反白顯示包含指定詞彙的日誌事件。如需詳細資訊，請參閱使用 Live Tail 以近乎實時的方式檢視日誌。	2023 年 6 月 6 日
CloudWatchLogsReadOnlyAccess政策已更新	CloudWatch 記錄檔已將權限新增至 CloudWatchLogsReadOnlyAccess. `logs:StartLiveTail` 和`logs:StopLiveTail` 權限已新增，讓具有此原則的使用者可以使用主控台來啟動和停止 CloudWatch 記錄即時尾端工作階段。如需詳細資訊，請參閱使用 Live Tail 以近乎即時的方式檢視日誌。	2023 年 6 月 6 日
CloudWatch 日誌洞察發布	您可以使用 CloudWatch 日誌深入解析以互動方式搜尋和分析記錄資料。如需詳細資訊，請參閱 Amazon CloudWatch 日誌使用者指南中的利用日誌洞察分析 CloudWatch 日誌資料	2018 年 11 月 27 日
支援 Amazon VPC 端點	您現在可以在 VPC 和 CloudWatch 記錄檔之間建立私人連線。如需詳細資訊，請參閱 Amazon CloudWatch 日誌使用指南中的將日 CloudWatch 誌與界面 VPC 端點搭配使用。	2018 年 6 月 28 日

下表說明 Amazon CloudWatch 日誌使用者指南的重要變更。

變更	描述	發行日期
介面 VPC 端點	在某些區域中，您可以使用界面 VPC 端點來防止 Amazon VPC 和 CloudWatch 日誌之間的流量離開 Amazon 網路。如需詳細資訊，請參閱 將記 CloudWatch 錄檔與介面 VPC 端點搭配使用。	2018 年 3 月 7 日
Route 53 DNS 查詢日誌	您可以使用 CloudWatch 記錄檔來儲存有關 Route 53 接收之 DNS 查詢的記錄。如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的什麼是 Amazon CloudWatch 日誌？或記錄 DNS 查詢。	2017 年 9 月 7 日
標記日誌群組	您可以使用標籤來分類您的日誌群組。如需詳細資訊，請參閱 在 Amazon CloudWatch 日誌中標記日誌群組。	2016 年 12 月 13 日
主控台改進	您可以從指標圖表導覽到關聯的日誌群組。如需詳細資訊，請參閱 從指標轉換到日誌。	2016 年 11 月 7 日
主控台可用性改善	提升經驗，讓您更輕鬆地搜尋、篩選及進行故障診斷。例如，現在您可以篩選特定日期和時間範圍的日誌資料。如需詳細資訊，請參閱 檢視傳送至 CloudWatch 記錄的記錄檔資料。	2016 年 8 月 29 日
增加了 AWS CloudTrail 對 Amazon CloudWatch 日誌和新日 CloudWatch 誌指標的支持	增加了對 CloudWatch 日誌的 AWS CloudTrail 支持。如需詳細資訊，請參閱 記錄 CloudWatch 日誌 API 和控制台操作 AWS CloudTrail。	2016 年 3 月 10 日
增加了對 CloudWatch 日誌導出到	增加了對將 CloudWatch 日誌數據導出到 Amazon S3 的支持。如需詳細資訊，請參閱 將日誌資料匯出至 Amazon S3。	2015 年 12 月 7 日

變更	描述	發行日期
Amazon S3 的支持		
增加了對 Amazon CloudWatch 日誌中記 AWS CloudTrail 錄事件的支持	您可以在中創建警報 CloudWatch 並接收捕獲的特定 API 活動的通知， CloudTrail 並使用該通知執行故障排除。	2014 年 11 月 10 日
增加了對 Amazon CloudWatch 日誌支持	您可以使用 Amazon CloudWatch 日誌從 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體或其他來源監控、存放和存取系統、應用程式和自訂日誌檔。然後，您可以使用 Amazon CloudWatch 主控台、中的 CloudWatch 日誌命令或日誌開發套件，從 CloudWatch 日誌擷取關聯的 CloudWatch 日誌資料。 AWS CLI如需詳細資訊，請參閱 什麼是 Amazon CloudWatch 日誌？。	2014 年 7 月 10 日

AWS 詞彙表

有關最新 AWS 術語，請參閱AWS 詞彙表 參考文獻中的AWS 詞彙表。

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。