



開發人員指南

Amazon Data Firehose



Amazon Data Firehose: 開發人員指南

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能附屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

Table of Contents

.....	x
什麼是 Amazon Data Firehose	1
了解關鍵概念	1
了解 Amazon Data Firehose 中的資料流程	2
使用 AWS SDKs	3
完成設定 Firehose 的先決條件	5
註冊 AWS	5
(選用) 下載程式庫和工具	5
教學課程：建立 Firehose 串流	7
選擇 Firehose 串流的來源和目的地	7
設定來源設定	9
設定 Amazon MSK 的來源設定	9
設定 Amazon Kinesis Data Streams 的來源設定	10
(選用) 設定記錄轉換和格式轉換	12
設定目的地設定	13
設定 Amazon S3 的目的地設定	14
設定 Apache Iceberg 資料表的目的地設定	17
設定 Amazon Redshift 的目的地設定	17
設定 OpenSearch Service 的目的地設定	22
設定 OpenSearch Serverless 的目的地設定	24
設定 HTTP 端點的目的地設定	25
設定 Datadog 的目的地設定	27
設定 Honeycomb 的目的地設定	29
設定 Coralogix 的目的地設定	30
設定 Dynatrace 的目的地設定	32
設定 LogicMonitor 的目的地設定	34
設定 Logz.io 的目的地設定	35
設定 MongoDB Atlas 的目的地設定	37
設定新複本的目的地設定	38
設定 Snowflake 的目的地設定	40
設定 Splunk 的目的地設定	43
設定 Splunk 可觀測性雲端的目的地設定	45
設定 Sumo Logic 的目的地設定	46
設定彈性的目的地設定	47

設定備份設定	49
設定緩衝提示	50
配置進階設定	52
測試您的 Firehose 串流	54
先決條件	54
使用 Amazon S3 進行測試	54
使用 Amazon Redshift 進行測試	54
使用 OpenSearch Service 進行測試	55
使用 Splunk 進行測試	56
使用 Apache Iceberg 資料表進行測試	56
將資料傳送至 Firehose 串流	57
設定 Kinesis 代理程式以傳送資料	57
先決條件	58
管理 AWS 登入資料	58
建立自訂登入資料提供者	58
下載並安裝 代理程式	59
設定和啟動 代理程式	61
指定代理程式組態設定	62
設定多個檔案目錄和串流	65
使用 代理程式預先處理資料	66
使用常見的 Agent CLI 命令	70
從 Kinesis Agent 傳送時的問題故障診斷	71
使用 AWS SDK 傳送資料	72
使用 PutRecord 的單一寫入操作	73
使用 PutRecordBatch 的批次寫入操作	73
將 CloudWatch Logs 傳送至 Firehose	74
解壓縮 CloudWatch Logs	74
解壓縮 CloudWatch Logs 後擷取訊息	74
從主控台對新的 Firehose 串流啟用解壓縮	76
在現有的 Firehose 串流上啟用解壓縮	76
在 Firehose 串流上停用解壓縮	77
針對 Firehose 中的解壓縮進行故障診斷	78
將 CloudWatch 事件傳送至 Firehose	79
設定 AWS IoT 將資料傳送至 Firehose	79
轉換來源資料	80
了解資料轉換流程	80

Lambda 調用持續時間	80
資料轉換所需的參數	80
支援的 Lambda 藍圖	82
處理資料轉換中的失敗	83
備份來源記錄	84
分割區串流資料	85
啟用動態分割	85
了解分割金鑰	86
使用內嵌剖析建立分割金鑰	86
使用 AWS Lambda 函數建立分割金鑰	88
使用 Amazon S3 儲存貯體字首交付資料	91
將資料交付至 Amazon S3 時新增行分隔符號	92
將動態分割套用至彙總資料	92
疑難排解動態分割錯誤	93
動態分割的緩衝區資料	93
轉換輸入資料格式	95
Deserializer	95
結構描述	96
Serializer	97
啟用記錄格式轉換	97
從主控台啟用記錄格式轉換	97
從 Firehose API 管理記錄格式轉換	98
處理資料格式轉換的錯誤	98
了解資料交付	100
了解跨 AWS 帳戶和區域的交付	102
了解 HTTP 端點交付請求和回應規格	102
要求格式	102
回應格式	106
範例	108
處理資料交付失敗	109
Amazon S3	109
Amazon Redshift	110
Amazon OpenSearch Service 和 OpenSearch Serverless	110
Splunk	111
HTTP 端點目的地	112
Snowflake	113

設定 Amazon S3 物件名稱格式	113
了解 Amazon S3 物件的自訂字首	122
設定 OpenSearch Service 的索引輪換	126
暫停和繼續資料交付	127
暫停 Firehose 串流	127
繼續 Firehose 串流	128
將資料交付至 Apache Iceberg 資料表	129
考量和限制	129
先決條件	132
在 Amazon S3 中交付至 Iceberg 資料表的先決條件	132
交付至 Amazon S3 資料表的先決條件	133
設定 Firehose 串流	133
設定來源和目的地	133
設定資料轉換	134
連接資料目錄	134
設定 JQ 表達式	135
設定唯一金鑰	135
指定重試持續時間	137
處理失敗的交付或處理	137
錯誤處理	137
設定緩衝提示	137
配置進階設定	138
將傳入的記錄路由到單一 Iceberg 資料表	138
將傳入的記錄路由到不同的 Iceberg 資料表	138
使用 JSONQuery 表達式將路由資訊提供給 Firehose	139
使用 AWS Lambda 函數提供路由資訊	140
指標指標	144
了解支援的資料類型	145
資料類型範例	145
Resources	149
標記 Firehose 串流	151
了解標籤基本概念	151
使用標記追蹤成本	152
了解標籤限制	152
安全	154
資料保護	154

使用 Kinesis Data Streams 進行伺服器端加密	155
使用 Direct PUT 或其他資料來源的伺服器端加密	155
控制存取	156
授予 Firehose 資源的存取權	157
授予 Firehose 存取您的私有 Amazon MSK 叢集	158
允許 Firehose 擔任 IAM 角色	158
授予 Firehose 對 的存取權 AWS Glue 以進行資料格式轉換	159
授予 Firehose 對 Amazon S3 目的地的存取權	160
授予 Firehose 對 Amazon S3 資料表的存取權	163
授予 Firehose 存取 Apache Iceberg 資料表目的地的權限	170
授予 Firehose 對 Amazon Redshift 目的地的存取權	171
授予 Firehose 對公有 OpenSearch Service 目的地的存取權	175
授予 Firehose 對 VPC 中 OpenSearch Service 目的地的存取權	176
授予 Firehose 對公有 OpenSearch Serverless 目的地的存取權	177
授予 Firehose 對 VPC 中 OpenSearch Serverless 目的地的存取權	180
授予 Firehose 存取 Splunk 目的地的權限	181
在 VPC 中存取 Splunk	183
教學課程：使用 Amazon Data Firehose 將 VPC 流程日誌擷取至 Splunk	185
存取 Snowflake 或 HTTP 端點	185
授予 Firehose 存取 Snowflake 目的地的權限	186
在 VPC 中存取 Snowflake	188
授予 Firehose 對 HTTP 端點目的地的存取權	192
從 Amazon MSK 跨帳戶交付	192
跨帳戶交付至 Amazon S3 目的地	195
跨帳戶交付至 OpenSearch Service 目的地	197
使用標籤控制存取	198
使用 驗證 AWS Secrets Manager	200
了解秘密	201
建立秘密	202
使用秘密	202
輪換秘密	204
透過主控台管理 IAM 角色	204
選擇現有的 IAM 角色	205
從主控台建立新的 IAM 角色	205
從主控台編輯 IAM 角色	207
法規遵循驗證	208

恢復能力	208
災難復原	208
了解基礎設施安全性	209
將 Firehose 與 AWS PrivateLink 搭配使用	209
實作安全最佳實務	214
實作最低權限存取	214
使用 IAM 角色	214
在相依資源中實作伺服器端加密	214
使用 CloudTrail 監控 API 呼叫	215
監控 Amazon Data Firehose	216
使用 CloudWatch 警示實作最佳實務	216
使用 CloudWatch 指標監控使用量	217
動態分割的 CloudWatch 指標	217
資料交付的 CloudWatch 指標	219
資料擷取指標	232
API 層級 CloudWatch 指標	240
資料轉換 CloudWatch 指標	244
CloudWatch Logs 解壓縮指標	244
格式轉換 CloudWatch 指標	245
伺服器端加密 (SSE) CloudWatch 指標	246
Amazon Data Firehose 的維度	247
Amazon Data Firehose 用量指標	247
存取 Amazon Data Firehose 的 CloudWatch 指標	248
使用 CloudWatch Logs 監控	249
資料交付錯誤	250
存取 Amazon Data Firehose 的 CloudWatch 日誌	283
監控代理程式運作狀態	284
使用 CloudWatch 監控	284
Log Firehose API 呼叫	285
CloudTrail 中的 Firehose 資訊	285
範例：Firehose 日誌檔案項目	286
程式碼範例	292
基本概念	292
動作	292
案例	304
將記錄放入 Firehose	304

故障診斷錯誤	318
常見問題	318
Firehose 串流無法使用	318
目的地沒有資料	319
資料新鮮度指標增加或未發出	319
記錄格式轉換為 Apache Parquet 失敗	320
Lambda 轉換物件缺少欄位	321
故障診斷 Amazon S3	321
疑難排解 Amazon Redshift	322
Amazon OpenSearch Service 疑難排解	323
疑難排解 Splunk	323
Snowflake 故障診斷	325
Firehose 串流建立失敗	325
針對 Firehose 端點連線能力進行故障診斷	326
疑難排解 HTTP 端點	327
CloudWatch Logs	327
疑難排解 MSK As A Source	330
Hose 建立失敗	330
Hose 暫停	330
Hose 預壓	331
資料新鮮度錯誤	331
MSK 叢集連線問題	331
配額	334
文件歷史紀錄	337

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。

什麼是 Amazon Data Firehose ？

Amazon Data Firehose 是一項全受管服務，可將即時[串流資料](#)交付至目的地，例如 Amazon Simple Storage Service (Amazon S3)、Amazon Redshift、Amazon OpenSearch Service、Amazon OpenSearch Serverless、Splunk、Apache Iceberg Tables，以及支援的第三方服務提供者擁有的任何自訂 HTTP 端點或 HTTP 端點，包括 Datadog、Dynatrace、LogicMonitor、MongoDB、New Relic、Coralogix 和 Elastic。使用 Amazon Data Firehose，您將不再需要編寫應用程式或管理資源。您可以將資料生產者設定為將資料傳送至 Amazon Data Firehose，並自動將資料交付至您指定的目的地。您也可以設定 Amazon Data Firehose 在交付資料之前轉換資料。

如需 AWS 大數據解決方案的詳細資訊，請參閱 [大數據 AWS](#)。如需 AWS 串流資料解決方案的詳細資訊，請參閱 [什麼是串流資料？](#)

了解關鍵概念

當您開始使用 Amazon Data Firehose 時，您可以從了解下列概念中獲益。

Firehose 串流

Amazon Data Firehose 的基礎實體。您可以透過建立 Firehose 串流，然後將資料傳送到其中，來使用 Amazon Data Firehose。如需詳細資訊，請參閱[教學課程：從主控台建立 Firehose 串流及將資料傳送至 Firehose 串流](#)。

記錄

您的資料生產者傳送至 Firehose 串流的感興趣的資料。記錄的大小可以高達 1,000 KB。

資料生產者

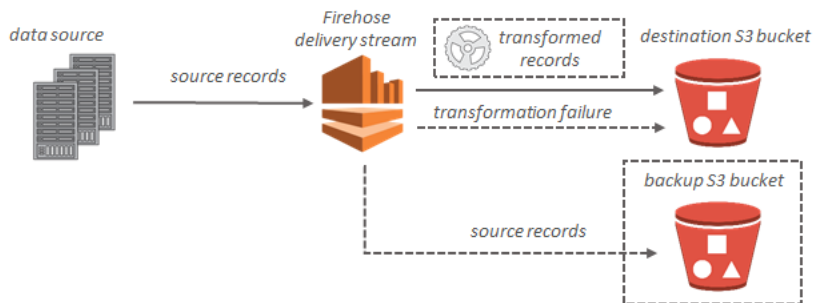
生產者將記錄傳送至 Firehose 串流。例如，將日誌資料傳送至 Firehose 串流的 Web 伺服器是資料生產者。您也可以設定 Firehose 串流自動從現有的 Kinesis 資料串流讀取資料，並將其載入目的地。如需詳細資訊，請參閱[將資料傳送至 Firehose 串流](#)。

緩衝區大小和緩衝區間隔

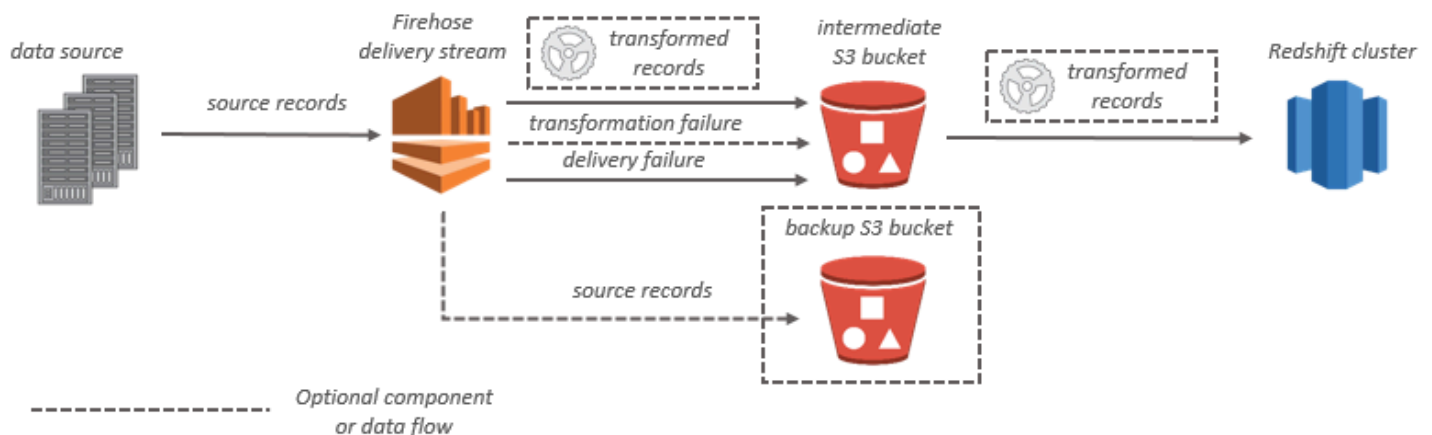
Amazon Data Firehose 會將傳入串流資料緩衝至特定大小或一段時間，再將其交付至目的地。Buffer Size 以 MBs 為單位，以秒 Buffer Interval 為單位。

了解 Amazon Data Firehose 中的資料流程

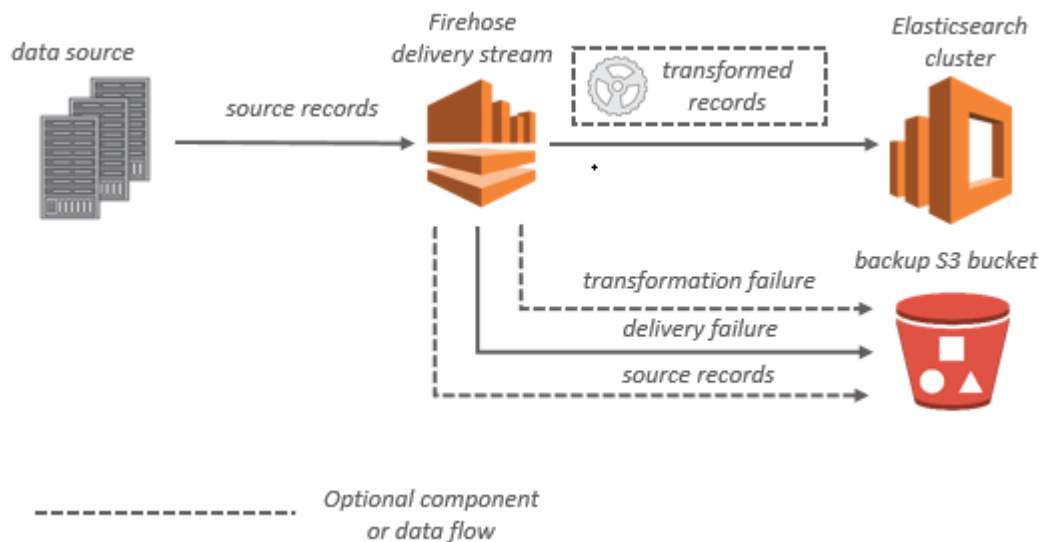
以 Amazon S3 目的地而言，串流資料都交付至您的 S3 儲存貯體。如果已啟用資料轉換，您可以選擇性將原始資料備份到另一個 Amazon S3 儲存貯體。



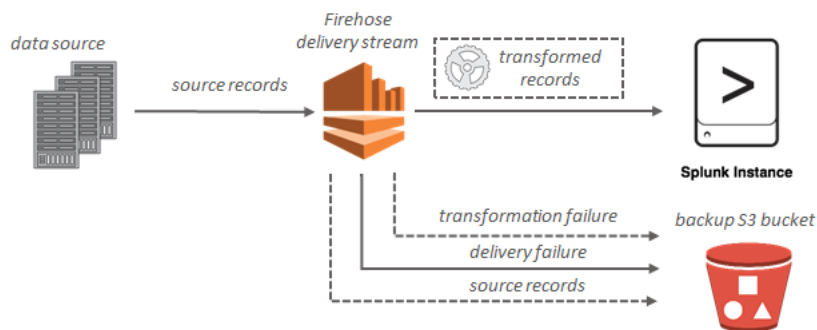
以 Amazon Redshift 目的地而言，串流資料會先交付至您的 S3 儲存貯體。然後，Amazon Data Firehose 會發出 Amazon Redshift COPY 命令，將資料從 S3 儲存貯體載入 Amazon Redshift 叢集。如果已啟用資料轉換，您可以選擇性將原始資料備份到另一個 Amazon S3 儲存貯體。



若目的地為 OpenSearch Service，串流資料會傳送至您的 OpenSearch Service 叢集，然後可以選擇同時備份至 S3 儲存貯體。



以 Splunk 目的地而言，串流資料會交付到 Splunk，然後可以選擇性同時備份到 S3 儲存貯體。



搭配 AWS SDK 使用 Firehose

AWS 軟體開發套件 (SDKs) 適用於許多熱門的程式設計語言。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
適用於 C++ 的 AWS SDK	適用於 C++ 的 AWS SDK 程式碼範例
AWS CLI	AWS CLI 程式碼範例
適用於 Go 的 AWS SDK	適用於 Go 的 AWS SDK 程式碼範例
適用於 Java 的 AWS SDK	適用於 Java 的 AWS SDK 程式碼範例

SDK 文件	代碼範例
適用於 JavaScript 的 AWS SDK	適用於 JavaScript 的 AWS SDK 程式碼範例
適用於 Kotlin 的 AWS SDK	適用於 Kotlin 的 AWS SDK 程式碼範例
適用於 .NET 的 AWS SDK	適用於 .NET 的 AWS SDK 程式碼範例
適用於 PHP 的 AWS SDK	適用於 PHP 的 AWS SDK 程式碼範例
AWS Tools for PowerShell	AWS Tools for PowerShell 程式碼範例
適用於 Python (Boto3) 的 AWS SDK	適用於 Python (Boto3) 的 AWS SDK 程式碼範例
適用於 Ruby 的 AWS SDK	適用於 Ruby 的 AWS SDK 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

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。

完成設定 Amazon Data Firehose 的先決條件

第一次使用 Amazon Data Firehose 之前，請先完成下列任務。

任務

- [註冊 AWS](#)
- [\(選用\) 下載程式庫和工具](#)

註冊 AWS

當您註冊 Amazon Web Services (AWS) 時，AWS 您的帳戶會自動註冊中的所有服務 AWS，包括 Amazon Data Firehose。您只需支付實際使用服務的費用。

如果您已經有 AWS 帳戶，請跳到下一個任務。若您尚未擁有 AWS 帳戶，請使用下列程序建立帳戶。

註冊 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或簡訊，並在電話鍵盤輸入驗證碼。

當您註冊時 AWS 帳戶，AWS 帳戶根使用者會建立。根使用者有權存取該帳戶中的所有 AWS 服務和資源。作為安全最佳實務，請將管理存取權指派給使用者，並且僅使用根使用者來執行[需要根使用者存取權的任務](#)。

(選用) 下載程式庫和工具

下列程式庫和工具可協助您以程式設計方式從命令列使用 Amazon Data Firehose：

- [Firehose API Operations](#) 是 Amazon Data Firehose 支援的基本操作集。
- Go <https://docs.aws.amazon.com/sdk-for-go/api/service/firehose/>、[Java](#)、[.NET](#)、[Node.js](#)、[Python](#) 和 [Ruby](#) AWS SDKs 包含 Amazon Data Firehose 支援和範例。

如果您的版本適用於 Java 的 AWS SDK 不包含 Amazon Data Firehose 的範例，您也可以從 [GitHub](#) 下載最新的 AWS SDK。

- [AWS Command Line Interface](#) 支援 Amazon Data Firehose。AWS CLI 可讓您從命令列控制多項 AWS 服務，並透過指令碼將其自動化。

教學課程：從主控台建立 Firehose 串流

您可以使用 AWS 管理主控台 或 AWS SDK 來建立 Firehose 串流到您選擇的目的地。

建立 Firehose 串流之後，您可以隨時使用 Amazon Data Firehose 主控台或 [UpdateDestination](#) 更新 Firehose 串流的組態。當您的組態更新時，Firehose 串流會保持 Active 狀態，而且您可以繼續傳送資料。更新的組態通常會在幾分鐘內生效。更新組態後，Firehose 串流的版本編號會增加的值。並出現在交付的 Amazon S3 物件名稱中。如需詳細資訊，請參閱[設定 Amazon S3 物件名稱格式](#)。

執行下列主題中的步驟來建立 Firehose 串流。

主題

- [選擇 Firehose 串流的來源和目的地](#)
- [設定來源設定](#)
- [\(選用\) 設定記錄轉換和格式轉換](#)
- [設定目的地設定](#)
- [設定備份設定](#)
- [配置進階設定](#)

選擇 Firehose 串流的來源和目的地

1. 開啟位於 <https://console.aws.amazon.com/firehose/> 的 Firehose 主控台。
2. 選擇建立 Firehose 串流。
3. 在建立 Firehose 串流頁面上，從下列其中一個選項中選擇 Firehose 串流的來源。
 - 直接 PUT – 選擇此選項可建立生產者應用程式直接寫入的 Firehose 串流。以下是與 Amazon Data Firehose 中的 Direct PUT 整合的 AWS 服務和代理程式和開放原始碼服務清單。此清單並不詳盡，可能還有其他服務可用來直接將資料傳送至 Firehose。
 - AWS 開發套件
 - AWS Lambda
 - AWS CloudWatch Logs
 - AWS CloudWatch 事件
 - AWS 雲端指標串流

- AWS IoT
 - AWS Eventbridge
 - Amazon Simple Email Service
 - Amazon SNS
 - AWS WAF Web ACL 日誌
 - Amazon API Gateway - 存取日誌
 - Amazon Pinpoint
 - Amazon MSK 代理程式日誌
 - Amazon Route 53 Resolver 查詢日誌
 - AWS 網路防火牆警示日誌
 - AWS 網路防火牆流程日誌
 - Amazon ElastiCache Redis SLOWLOG
 - Kinesis 代理程式 (linux)
 - Kinesis Tap (windows)
 - Fluentbit
 - Fluentd
 - Apache Nifi
 - Snowflake
 - Amazon Kinesis Data Streams – 選擇此選項可設定使用 Kinesis 資料串流做為資料來源的 Firehose 串流。然後，您可以使用 Firehose 輕鬆從現有的 Kinesis 資料串流讀取資料，並將其載入目的地。如需使用 Kinesis Data Streams 做為資料來源的詳細資訊，請參閱[使用 Kinesis Data Streams 將資料傳送至 Firehose 串流](#)。
 - Amazon MSK – 選擇此選項可設定使用 Amazon MSK 做為資料來源的 Firehose 串流。然後，您可以使用 Firehose 輕鬆從現有的 Amazon MSK 叢集讀取資料，並將其載入指定的 S3 儲存貯體。如需詳細資訊，請參閱[使用 Amazon MSK 將資料傳送至 Firehose 串流](#)。
4. 從下列其中一個 Firehose 支援的目的地中，為您的 Firehose 串流選擇目的地。
- Amazon OpenSearch Service
 - Amazon OpenSearch Serverless
 - Amazon Redshift
 - Amazon S3
 - Apache Iceberg 資料表

- Coralogix
- Datadog
- Dynatrace
- 彈性
- HTTP 端點
- Honeycomb
- Logic Monitor
- Logz.io
- MongoDB Cloud
- New Relic
- Splunk
- Splunk Observability Cloud
- Sumo Logic
- Snowflake

5. 對於 Firehose 串流名稱，您可以使用主控台為您產生的名稱，或新增您選擇的 Firehose 串流。

設定來源設定

您可以根據您選擇從主控台將資訊傳送至 Firehose 串流的來源來設定來源設定。您可以設定 Amazon MSK 和 Amazon Kinesis Data Streams 的來源設定做為來源。沒有可用於 Direct PUT 做為來源的來源設定。

設定 Amazon MSK 的來源設定

當您選擇 Amazon MSK 將資訊傳送至 Firehose 串流時，您可以選擇 MSK 佈建和 MSK-Serverless 叢集。然後，您可以使用 Firehose 輕鬆從特定 Amazon MSK 叢集和主題讀取資料，並將其載入指定的 S3 目的地。

在頁面的來源設定區段中，提供下列欄位的值。

Amazon MSK 叢集連線功能

根據您的叢集組態，選擇私有引導代理程式 (建議使用) 或公有引導代理程式選項。引導代理程式由 ~~Apache Kafka 用戶端用作連線至叢集的起點。公有引導代理程式適用於從外部公開存取 AWS，而~~
設定來源設定

私有引導代理程式則適用於從內存取 AWS。如需有關 Amazon MSK 的詳細資訊，請參閱 [Amazon Managed Streaming for Apache Kafka](#)。

若要透過私有引導代理程式連線至佈建或無伺服器 Amazon MSK 叢集，叢集必須符合以下所有要求。

- 叢集必須為作用中。
- 叢集必須使用 IAM 作為其存取控制方法之一。
- IAM 存取控制方法必須啟用多 VPC 私有連線功能。
- 您必須將資源型政策新增至此叢集，以授予 Firehose 服務委託人調用 Amazon MSK CreateVpcConnection API 操作的許可。

若要透過公有引導代理程式連線至佈建的 Amazon MSK 叢集，叢集必須符合以下所有要求。

- 叢集必須為作用中。
- 叢集必須使用 IAM 作為其存取控制方法之一。
- 叢集必須為可公開存取。

MSK 叢集帳戶

您可以選擇 Amazon MSK 叢集所在的帳戶。這可以是下列其中一項。

- 目前帳戶 – 可讓您從目前 AWS 帳戶中的 MSK 叢集擷取資料。因此，您必須指定 Amazon MSK 叢集的 ARN，Firehose 串流會從中讀取資料。
- 跨帳戶 – 可讓您從另一個 AWS 帳戶中的 MSK 叢集擷取資料。如需詳細資訊，請參閱 [從 Amazon MSK 跨帳戶交付](#)。

主題

指定您希望 Firehose 串流從中擷取資料的 Apache Kafka 主題。Firehose 串流建立完成後，您無法更新此主題。

Note

Firehose 會自動解壓縮 Apache Kafka 訊息。

設定 Amazon Kinesis Data Streams 的來源設定

設定 Amazon Kinesis Data Streams 的來源設定，將資訊傳送至 Firehose 串流，如下所示。

⚠ Important

如果您使用 Kinesis Producer Library (KPL) 來寫入資料到 Kinesis 資料串流，則可使用彙整來合併您寫入至該 Kinesis 資料串流的記錄。如果您接著使用該資料串流做為 Firehose 串流的來源，Amazon Data Firehose 會在記錄交付至目的地之前將其取消彙總。如果您將 Firehose 串流設定為轉換資料，Amazon Data Firehose 會在記錄交付至之前，先將記錄取消彙總 AWS Lambda。如需詳細資訊，請參閱[使用 Kinesis Producer Library 開發 Amazon Kinesis Data Streams 生產者及彙整](#)。

在來源設定下，選擇 Kinesis 資料串流清單中的現有串流，或以格式輸入資料串流 ARNarn:aws:kinesis:[Region]:[AccountId]:stream/[StreamName]。

如果您沒有現有的資料串流，請選擇建立，從 Amazon Kinesis 主控台建立新的資料串流。您可能需要具有 Kinesis 串流必要許可的 IAM 角色。如需詳細資訊，請參閱[???](#)。建立新的串流後，請選擇重新整理圖示以更新 Kinesis 串流清單。如果您有大量串流，使用 Filter by name (依名稱篩選) 來篩選清單。

i Note

當您將 Kinesis 資料串流設定為 Firehose 串流的來源時，Amazon Data Firehose PutRecord和 PutRecordBatch操作會停用。若要在此情況下將資料新增至 Firehose 串流，請使用 Kinesis Data Streams PutRecord和 PutRecords操作。

Amazon Data Firehose 會開始從 Kinesis 串流LATEST的位置讀取資料。如需更多關於 Kinesis Data Streams 位置的詳細資訊，請參閱[GetShardIterator](#)。

Amazon Data Firehose 會呼叫每個碎片的 Kinesis Data Streams [GetRecords](#) 操作，每秒一次。不過，當啟用完整備份時，Firehose 會呼叫每個碎片的 Kinesis Data Streams GetRecords操作，每秒兩次，一個用於主要交付目的地，另一個用於完整備份。

可從相同的 Kinesis 串流讀取多個 Firehose 串流。其他 Kinesis 應用程式 (取用者) 也可以讀取相同串流。來自任何 Firehose 串流或其他消費者應用程式的每次呼叫都會計入碎片的整體限流限制。為了避免受到調節限制，請小心規劃您的應用程式。如需有關 Kinesis Data Streams 限制的詳細資訊，請參閱[Amazon Kinesis Data Streams 限制](#)。

繼續下一個步驟，以設定記錄轉換和格式轉換。

(選用) 設定記錄轉換和格式轉換

設定 Amazon Data Firehose 來轉換和轉換您的記錄資料。

如果您選擇 Amazon MSK 作為 Firehose 串流的來源。

在使用 AWS Lambda 轉換來源記錄區段中，提供下列欄位的值。

1. 資料轉換

若要建立不會轉換傳入資料的 Firehose 串流，請勿勾選啟用資料轉換核取方塊。

若要指定 Lambda 函數讓 Firehose 叫用並使用來轉換傳入的資料，請核取啟用資料轉換核取方塊。您可以使用其中一個 Lambda 藍圖，或選擇現有的 Lambda 函數，進而設定新的 Lambda 函數。您的 Lambda 函數必須包含 Firehose 所需的狀態模型。如需詳細資訊，請參閱 [轉換 Amazon Data Firehose 中的來源資料](#)。

2. 在 Convert record format (轉換記錄格式) 區段中，提供以下欄位的值：

記錄格式轉換

若要建立不會轉換傳入資料記錄格式的 Firehose 串流，請選擇已停用。

若要轉換傳入記錄的格式，請選擇 Enabled (已啟用)，然後指定輸出格式。您需要指定 AWS Glue 資料表，其中包含您希望 Firehose 用來轉換記錄格式的結構描述。如需詳細資訊，請參閱 [轉換輸入資料格式](#)。

如需如何使用設定記錄格式轉換的範例 CloudFormation，請參閱 [AWS :: KinesisFirehose :: DeliveryStream](#)。

如果您選擇 Amazon Kinesis Data Streams 或 Direct PUT 作為 Firehose 串流的來源

在來源設定區段中，提供下列欄位。

1. 在轉換記錄下，選擇下列其中一項：
 - a. 如果您的目的地是 Amazon S3 或 Splunk，請在解壓縮來源記錄 Amazon CloudWatch Logs 區段中，選擇開啟解壓縮。
 - b. 在使用 AWS Lambda 轉換來源記錄區段中，提供下列欄位的值：

資料轉換

若要建立不會轉換傳入資料的 Firehose 串流，請勿勾選啟用資料轉換核取方塊。

若要為 Amazon Data Firehose 指定 Lambda 函數來叫用並使用來轉換傳入的資料，請核取啟用資料轉換核取方塊。您可以使用其中一個 Lambda 藍圖，或選擇現有的 Lambda 函數，進而設定新的 Lambda 函數。您的 Lambda 函數必須包含 Amazon Data Firehose 所需的狀態模型。如需詳細資訊，請參閱[轉換 Amazon Data Firehose 中的來源資料](#)。

2. 在 Convert record format (轉換記錄格式) 區段中，提供以下欄位的值：

記錄格式轉換

若要建立不會轉換傳入資料記錄格式的 Firehose 串流，請選擇已停用。

若要轉換傳入記錄的格式，請選擇 Enabled (已啟用)，然後指定輸出格式。您需要指定 AWS Glue 資料表，其中包含您希望 Amazon Data Firehose 用來轉換記錄格式的結構描述。如需詳細資訊，請參閱[轉換輸入資料格式](#)。

如需如何使用設定記錄格式轉換的範例 CloudFormation，請參閱 [AWS :: KinesisFirehose :: DeliveryStream](#)。

設定目的地設定

本節說明您必須根據所選目的地為 Firehose 串流設定的設定。

主題

- [設定 Amazon S3 的目的地設定](#)
- [設定 Apache Iceberg 資料表的目的地設定](#)
- [設定 Amazon Redshift 的目的地設定](#)
- [設定 OpenSearch Service 的目的地設定](#)
- [設定 OpenSearch Serverless 的目的地設定](#)
- [設定 HTTP 端點的目的地設定](#)
- [設定 Datadog 的目的地設定](#)
- [設定 Honeycomb 的目的地設定](#)
- [設定 Coralogix 的目的地設定](#)
- [設定 Dynatrace 的目的地設定](#)

- [設定 LogicMonitor 的目的地設定](#)
- [設定 Logz.io 的目的地設定](#)
- [設定 MongoDB Atlas 的目的地設定](#)
- [設定新複本的目的地設定](#)
- [設定 Snowflake 的目的地設定](#)
- [設定 Splunk 的目的地設定](#)
- [設定 Splunk 可觀測性雲端的目的地設定](#)
- [設定 Sumo Logic 的目的地設定](#)
- [設定彈性的目的地設定](#)

設定 Amazon S3 的目的地設定

您必須指定以下設定，才能使用 Amazon S3 作為 Firehose 串流目的地。

- 請輸入下列欄位的值。

S3 bucket (S3 儲存貯體)

選擇要交付串流資料您所擁有的 S3 儲存貯體。您可以建立新的或選擇現有的 S3 儲存貯體。

新行分隔符

您可以設定 Firehose 串流，在交付至 Amazon S3 的物件中的記錄之間新增行分隔符號。若要這麼做，請選擇已啟用。若要在交付至 Amazon S3 的物件中的記錄之間新增行分隔符號，請選擇停用。如果您計劃使用 Athena 查詢具有彙總記錄的 S3 物件，請啟用此選項。

動態分割

選擇已啟用以啟用並設定動態分割。

多筆記錄取消彙總

這是剖析 Firehose 串流中記錄的程序，並根據有效的 JSON 或指定的新行分隔符號來分隔它們。

如果您將多個事件、日誌或記錄彙整到單一 PutRecord 和 PutRecordBatch API 呼叫中，您仍然可以啟用和設定動態分割。使用彙總資料時，當您啟用動態分割時，Amazon Data Firehose 會剖析記錄，並在每個 API 呼叫中尋找多個有效的 JSON 物件。當 Firehose 串流設定為 Kinesis Data Stream 做為來源時，您也可以使用 Kinesis Producer Library (KPL) 中使用內建彙

總。資料分割功能在資料取消彙整後執行。因此，每個 API 呼叫中的每個記錄都可以交付到不同的 Amazon S3 字首。您也可以利用 Lambda 函數整合，在資料分割功能之前執行任何其他取消彙總或任何其他轉換。

Important

如果您的資料已彙整，則只有在執行資料取消彙整之後才能套用動態分割。因此，如果您對彙整資料啟用動態分割，則必須選擇已啟用以啟用多記錄取消彙整。

Firehose 串流會依下列順序執行下列處理步驟：KPL (protobuf) 取消彙總、JSON 或分隔符號取消彙總、Lambda 處理、資料分割、資料格式轉換和 Amazon S3 交付。

多筆記錄取消彙總類型

如果您啟用了多筆記錄取消彙總，則必須指定 Firehose 將資料取消彙總的方法。使用下拉式功能表選擇 JSON 或分隔符號。

內嵌剖析

這是其中一種支援的機制，可動態分割繫結到 Amazon S3 的資料。若要使用內嵌剖析來動態分割資料，您必須指定要用作分割索引鍵的資料記錄參數，並為每個指定的分割索引鍵提供一個值。選擇已啟用以啟用並設定內嵌剖析。

Important

如果您在上述步驟中指定 AWS Lambda 函數來轉換來源記錄，您可以使用此函數動態分割繫結至 S3 的資料，而且仍然可以使用內嵌剖析建立分割金鑰。透過動態分割，您可以使用內嵌剖析或 AWS Lambda 函數來建立分割金鑰。或者，您可以同時使用內嵌剖析和 AWS Lambda 函數來建立分割金鑰。

動態分割索引鍵

您可以使用索引鍵和值欄位來指定要用作動態分割索引鍵的資料記錄參數，以及用來產生動態分割索引鍵值的 jq 查詢。Firehose 僅支援 jq 1.6。您可以指定最多 50 個動態分割索引鍵。您必須輸入動態分割索引鍵值的有效 jq 表達式，才能成功設定 Firehose 串流的動態分割。

S3 儲存貯體的字首

當您啟用和設定動態分割時，必須指定 Amazon Data Firehose 交付分割資料的 S3 儲存貯體字首。

為了正確設定動態分割，S3 儲存貯體字首的數目必須與指定的分割索引鍵數目相同。

您可以使用內嵌剖析或指定的 AWS Lambda 函數來分割來源資料。如果您指定 AWS Lambda 函數來建立來源資料的分割索引鍵，則必須使用下列格式手動輸入 S3 儲存貯體字首值："partitionKeyFromLambda:keyID"。如果使用內嵌剖析來指定來源資料的分割索引鍵，則您可以使用下列格式手動輸入 S3 儲存貯體預覽值："partitionKeyFromQuery:keyID"，或者您可以選擇套用動態分割索引鍵按鈕，使用動態分割索引鍵/值對來自動產生 S3 儲存貯體字首。使用內嵌剖析或 AWS Lambda 分割資料時，您也可以使用 S3 儲存貯體字首中使用下列表達式表單：! {namespace : value}，其中命名空間可以是 partitionKeyFromQuery 或 partitionKeyFromLambda。

S3 儲存貯體和 S3 錯誤輸出字首時區

在 [Amazon S3 物件的自訂字首](#) 中，選擇您要用於日期和時間的時區。根據預設，Firehose 會以 UTC 新增時間字首。如果您想要使用不同的時區，可以變更 S3 字首中使用的時區。

緩衝提示

Firehose 會先緩衝傳入的資料，再將其交付至指定的目的地。建議的目的地緩衝區大小因服務提供者而異。

S3 壓縮

選擇 GZIP、Snappy、Zip 或 Hadoopp 相容的 Snappy 資料壓縮，或不壓縮資料。Snappy、zip 檔和 Hadoop 相容的 Snappy 壓縮不適用於以 Amazon Redshift 作為目的地的 Firehose 串流。

S3 副檔名格式 (選用)

為交付至 Amazon S3 目的地儲存貯體的物件指定副檔名格式。如果您啟用此功能，指定的副檔名會覆寫資料格式轉換或 S3 壓縮功能附加的預設副檔名，例如 .parquet 或 .gz。當您搭配資料格式轉換或 S3 壓縮使用此功能時，請確定是否已設定正確的副檔名。副檔名必須以句號 (.) 開頭，且可包含允許的字元：0-9a-z ! -_.*'()。副檔名不能超過 128 個字元。

S3 加密

Firehose 支援使用 AWS Key Management Service (SSE-KMS) 加密 Amazon S3 中交付資料的 Amazon S3 伺服器端加密。您可以選擇使用目的地 S3 儲存貯體中指定的預設加密類型，

或使用您擁有的金鑰清單中的 AWS KMS 金鑰進行加密。如果您使用 AWS KMS 金鑰加密資料，您可以使用預設 AWS 受管金鑰 (aws/s3) 或客戶受管金鑰。如需詳細資訊，請參閱[使用伺服器端加密搭配 AWS KMS 受管金鑰保護資料 \(SSE-KMS\)](#)。

設定 Apache Iceberg 資料表的目的地設定

Firehose 支援 Apache Iceberg Tables 作為除中國區域、亞太區域（台北）AWS GovCloud (US) Regions、亞太區域（馬來西亞）、亞太區域（紐西蘭）和墨西哥（中部）[AWS 區域](#)以外所有的目的地。

如需 Apache Iceberg 資料表做為目的地的詳細資訊，請參閱[使用 Amazon Data Firehose 將資料交付至 Apache Iceberg 資料表](#)。

設定 Amazon Redshift 的目的地設定

本節說明使用 Amazon Redshift 做為 Firehose 串流目的地的設定。

根據您是擁有 Amazon Redshift 佈建的叢集，還是 Amazon Redshift Serverless 工作群組，選擇下列其中一個程序。

- [Amazon Redshift 佈建叢集](#)
- [設定 Amazon Redshift Serverless 工作群組的目的地設定](#)

Note

Firehose 無法寫入使用增強型 VPC 路由的 Amazon Redshift 叢集。

Amazon Redshift 佈建叢集

本節說明使用 Amazon Redshift 佈建叢集做為 Firehose 串流目的地的設定。

- 為下列欄位輸入值：

叢集

要放入複製的 S3 儲存貯體的 Amazon Redshift 叢集。將 Amazon Redshift 叢集設定為可公開存取並解除封鎖 Amazon Data Firehose IP 地址。如需詳細資訊，請參閱[授予 Firehose 對 Amazon Redshift 目的地的存取權](#)。

身分驗證

您可以選擇直接輸入使用者名稱/密碼，或從擷取秘密 AWS Secrets Manager 以存取 Amazon Redshift 叢集。

- 使用者名稱

指定具有存取 Amazon Redshift 叢集許可的 Amazon Redshift 使用者。該使用者必須具備 Amazon Redshift INSERT 許可，才能夠將 S3 儲存貯體中的資料複製到 Amazon Redshift 叢集。

- 密碼

為具有存取叢集許可的使用者指定密碼。

- Secret

從 選取包含 AWS Secrets Manager Amazon Redshift 叢集登入資料的秘密。如果您在下拉式清單中看不到秘密，請在 中 AWS Secrets Manager 為 Amazon Redshift 登入資料建立一個秘密。如需詳細資訊，請參閱 [在 Amazon Data Firehose AWS Secrets Manager 中使用 驗證](#)。

資料庫

要放入複製資料的 Amazon Redshift 資料庫。

資料表

要放入複製資料的 Amazon Redshift 資料表。

資料欄

(選用) 要放入複製資料的表格特定資料欄。如果您的 Amazon S3 物件中定義的資料欄數量少於 Amazon Redshift 資料表中的欄位數量，請使用此選項。

中繼 S3 目的地

Firehose 會先將資料交付至 S3 儲存貯體，然後發出 Amazon Redshift COPY 命令將資料載入 Amazon Redshift 叢集。指定要交付串流資料您所擁有的 S3 儲存貯體。建立新的 S3 儲存貯體，或是選擇您目前擁有的儲存貯體。

Firehose 在將資料載入 Amazon Redshift 叢集之後，不會從 S3 儲存貯體刪除資料。您可以使用生命週期配置管理 S3 儲存貯體中的資料。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的 [物件生命週期管理](#)。

中繼 S3 字首

(選用) 要使用 Amazon S3 物件的預設字首，請將此選項保留空白。Firehose 會自動為交付的 Amazon S3 物件使用 "YYYY/MM/dd/HH" UTC 時間格式的字首。您可以新增到此字首前方。如需詳細資訊，請參閱[設定 Amazon S3 物件名稱格式](#)。

COPY options (COPY 選項)

此處指的是可以在 Amazon Redshift COPY 命令中指定的參數。根據您的組態而定，這些參數可能為必要。例如，如果已啟用 Amazon S3 資料壓縮，則需要 "GZIP"。如果您的 S3 儲存貯體不在與 Amazon Redshift 叢集相同的 AWS 區域中，則需要 "REGION"。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 [COPY](#)。

COPY command (COPY 命令)

Amazon Redshift COPY 命令。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 [COPY](#)。

Retry duration (重試持續時間)

如果 COPY Amazon Redshift 叢集的資料失敗，Firehose 重試的持續時間 (0–7200 秒)。Firehose 每 5 分鐘重試一次，直到重試持續時間結束為止。如果您將重試持續時間設定為 0 (零) 秒，Firehose 不會在 COPY 命令失敗時重試。

緩衝提示

Firehose 會先緩衝傳入的資料，再將其交付至指定的目的地。建議的目的地緩衝區大小因服務提供者而異。

S3 壓縮

選擇 GZIP、Snappy、Zip 或 Hadoopp 相容的 Snappy 資料壓縮，或不壓縮資料。Snappy、zip 檔和 Hadoop 相容的 Snappy 壓縮不適用於以 Amazon Redshift 作為目的地的 Firehose 串流。

S3 副檔名格式 (選用)

S3 副檔名格式 (選用) – 為交付至 Amazon S3 目的地儲存貯體的物件指定副檔名格式。如果您啟用此功能，指定的副檔名會覆寫資料格式轉換或 S3 壓縮功能附加的預設副檔名，例如 .parquet 或 .gz。當您搭配資料格式轉換或 S3 壓縮使用此功能時，請確定是否已設定正確的副檔名。副檔名必須以句號 (.) 開頭，且可包含允許的字元：0-9a-z ! _ . * ()。副檔名不能超過 128 個字元。

S3 加密

Firehose 支援使用 AWS Key Management Service (SSE-KMS) 加密 Amazon S3 中交付資料的 Amazon S3 伺服器端加密。您可以選擇使用目的地 S3 儲存貯體中指定的預設加密類型，或使用您擁有的金鑰清單中的 AWS KMS 金鑰進行加密。如果您使用 AWS KMS 金鑰加密資料，您可以使用預設 AWS 受管金鑰 (aws/s3) 或客戶受管金鑰。如需詳細資訊，請參閱[使用伺服器端加密搭配 AWS KMS 受管金鑰保護資料 \(SSE-KMS\)](#)。

設定 Amazon Redshift Serverless 工作群組的目的地設定

本節說明使用 Amazon Redshift Serverless 工作群組做為 Firehose 串流目的地的設定。

- 為下列欄位輸入值：

Workgroup name (工作群組名稱)

要將 S3 儲存貯體資料複製到的 Amazon Redshift Serverless 工作群組。將 Amazon Redshift Serverless 工作群組設定為可公開存取，並解除封鎖 Firehose IP 地址。如需詳細資訊，請參閱[連線至 Amazon Redshift Serverless](#) 中「連線至可公開存取的 Amazon Redshift Serverless 執行個體」一節以及 [授予 Firehose 對 Amazon Redshift 目的地的存取權](#)。

身分驗證

您可以選擇直接輸入使用者名稱/密碼，或從擷取秘密 AWS Secrets Manager 以存取 Amazon Redshift Serverless 工作群組。

- 使用者名稱

指定具有存取 Amazon Redshift Serverless 工作群組許可的 Amazon Redshift 使用者。該使用者必須具備 Amazon Redshift INSERT 許可，才可以將 S3 儲存貯體中的資料複製至 Amazon Redshift Serverless 工作群組。

- 密碼

為具有存取 Amazon Redshift Serverless 工作群組許可的使用者指定密碼。

- Secret

從選取包含 AWS Secrets Manager Amazon Redshift Serverless 工作群組登入資料的秘密。如果您在下拉式清單中看不到秘密，請在 AWS Secrets Manager 為 Amazon

Redshift 登入資料建立一個秘密。如需詳細資訊，請參閱 [在 Amazon Data Firehose AWS Secrets Manager 中使用 驗證](#)。

資料庫

要放入複製資料的 Amazon Redshift 資料庫。

資料表

要放入複製資料的 Amazon Redshift 資料表。

資料欄

(選用) 要放入複製資料的表格特定資料欄。如果您的 Amazon S3 物件中定義的資料欄數量少於 Amazon Redshift 資料表中的欄位數量，請使用此選項。

中繼 S3 目的地

Amazon Data Firehose 會先將資料交付至 S3 儲存貯體，然後發出 Amazon Redshift COPY 命令將資料載入 Amazon Redshift Serverless 工作群組。指定要交付串流資料您所擁有的 S3 儲存貯體。建立新的 S3 儲存貯體，或是選擇您目前擁有的儲存貯體。

Firehose 在將資料載入至 Amazon Redshift Serverless 工作群組後，不會從 S3 儲存貯體刪除資料。您可以使用生命週期配置管理 S3 儲存貯體中的資料。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的 [物件生命週期管理](#)。

中繼 S3 字首

(選用) 要使用 Amazon S3 物件的預設字首，請將此選項保留空白。Firehose 會自動為交付的 Amazon S3 物件使用 "YYYY/MM/dd/HH" UTC 時間格式的字首。您可以新增到此字首前方。如需詳細資訊，請參閱 [設定 Amazon S3 物件名稱格式](#)。

COPY options (COPY 選項)

此處指的是可以在 Amazon Redshift COPY 命令中指定的參數。根據您的組態而定，這些參數可能為必要。例如，如果啟用 Amazon S3 資料壓縮，則需要 "GZIP"。如果您的 S3 儲存貯體不在與 Amazon Redshift Serverless 工作群組相同的 AWS 區域中，則需要 "REGION"。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 [COPY](#)。

COPY command (COPY 命令)

Amazon Redshift COPY 命令。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 [COPY](#)。

Retry duration (重試持續時間)

如果 COPY Amazon Redshift Serverless 工作群組的資料失敗，Firehose 會重試的持續時間 (0–7200 秒)。Firehose 每 5 分鐘重試一次，直到重試持續時間結束為止。如果您將重試持續時間設定為 0 (零) 秒，Firehose 不會在 COPY 命令失敗時重試。

緩衝提示

Firehose 會先緩衝傳入的資料，再將其交付至指定的目的地。建議的目的地緩衝區大小因服務提供者而異。

S3 壓縮

選擇 GZIP、Snappy、Zip 或 Hadoopp 相容的 Snappy 資料壓縮，或不壓縮資料。Snappy、zip 檔和 Hadoop 相容的 Snappy 壓縮不適用於以 Amazon Redshift 作為目的地的 Firehose 串流。

S3 副檔名格式 (選用)

S3 副檔名格式 (選用) – 為交付至 Amazon S3 目的地儲存貯體的物件指定副檔名格式。如果您啟用此功能，指定的副檔名會覆寫資料格式轉換或 S3 壓縮功能附加的預設副檔名，例如 .parquet 或 .gz。當您搭配資料格式轉換或 S3 壓縮使用此功能時，請確定是否已設定正確的副檔名。副檔名必須以句號 (.) 開頭，且可包含允許的字元：0-9a-z ! -_.*()。副檔名不能超過 128 個字元。

S3 加密

Firehose 支援使用 AWS Key Management Service (SSE-KMS) 加密 Amazon S3 中交付資料的 Amazon S3 伺服器端加密。您可以選擇使用目的地 S3 儲存貯體中指定的預設加密類型，或使用您擁有的金鑰清單中的 AWS KMS 金鑰進行加密。如果您使用 AWS KMS 金鑰加密資料，您可以使用預設 AWS 受管金鑰 (aws/s3) 或客戶受管金鑰。如需詳細資訊，請參閱[使用伺服器端加密搭配 AWS KMS 受管金鑰保護資料 \(SSE-KMS\)](#)。

設定 OpenSearch Service 的目的地設定

Firehose 支援 Elasticsearch 版本 – 1.5、2.3、5.1、5.3、5.5、5.6，以及所有 6.*、7.* 和 8.* 版本。Firehose 支援 Amazon OpenSearch Service 2.x 和 3.x。

本節會描述使用 OpenSearch Service 作為目的地的選項。

- 為下列欄位輸入值：

OpenSearch Service 域

要交付資料的 OpenSearch Service 域。

索引

將資料索引至您的 OpenSearch Service 叢集時，所使用的 OpenSearch Service 索引名稱。

Index rotation (索引輪換)

選擇 OpenSearch Service 索引輪換是否啟用及頻率。若啟用索引輪換，Amazon Data Firehose 便會將對應的時間戳記附加到指定的索引名稱，並進行輪換。如需詳細資訊，請參閱[設定 OpenSearch Service 的索引輪換](#)。

類型

將資料索引至您的 OpenSearch Service 叢集時，所使用的 OpenSearch Service 類型名稱。若是 Elasticsearch 7.x 和 OpenSearch 1.x，每個索引僅能使用一個類型。如果您嘗試為已具備其他類型的現有索引指定新類型，則 Firehose 會在執行時期傳回錯誤。

針對 Elasticsearch 7.x，將此欄位保留空白。

Retry duration (重試持續時間)

如果對 OpenSearch 的索引請求失敗，Firehose 重試的持續時間。對於重試持續時間，您可以設定介於 0-7200 秒之間的任何值。預設的重試持續時間為 300 秒。Firehose 將重試多次並指數退避，直到重試持續時間過期為止。

重試持續時間到期後，Firehose 會將資料交付至已設定的 S3 錯誤儲存貯體無效字母佇列 (DLQ)。對於交付至 DLQ 的資料，您必須將資料從設定的 S3 錯誤儲存貯體重新驅動回 OpenSearch 目的地。

如果因為 OpenSearch 叢集的停機時間或維護而想要封鎖 Firehose 串流將資料交付至 DLQ，您可以將重試持續時間設定為較高的值，以秒為單位。您可以聯絡[AWS 支援](#)，將超過 7200 秒的重試持續時間值增加到。

DocumentID 類型

表示設定文件 ID 的方法。支援的方法為 Firehose 產生的文件 ID 和 OpenSearch Service 產生的文件 ID。未設定文件 ID 值時，Firehose 產生的文件 ID 是預設選項。OpenSearch Service 產生的文件 ID 是建議選項，因為其支援大量寫入操作，包括日誌分析和可觀測性，在 OpenSearch Service 域取用較少的 CPU 資源，因此可改善效能。

目的地 VPC 連線能力

如果您的 OpenSearch Service 域位於私有 VPC 中，則請使用此區段來指定該 VPC。同時，請指定 Amazon Data Firehose 將資料傳送至您的 OpenSearch Service 網域時，要使用的子網路和子群組。您可以使用 OpenSearch Service 域所使用的相同安全群組。如果您指定不同的安全群組，請確認這些群組可以輸出 HTTPS 流量到 OpenSearch Service 域的安全群組。此外，請確定 OpenSearch Service 網域的安全群組允許來自您在設定 Firehose 串流時所指定安全群組的 HTTPS 流量。如果您針對 Firehose 串流和 OpenSearch Service 網域使用相同的安全群組，請確定安全群組的傳入規則允許 HTTPS 流量。如需安全群組規則的詳細資訊，請參閱 Amazon VPC 文件中 OpenSearch 的[安全群組規則](#)。

Important

當您指定子網路以將資料交付至私有 VPC 中的目的地時，請確定您選擇的子網路中有足夠的可用 IP 地址。如果指定的子網路中沒有可用的可用 IP 地址，Firehose 無法為私有 VPC 中的資料交付建立或新增 ENIs，且交付將會降級或失敗。

緩衝提示

Amazon Data Firehose 在將傳入資料傳遞至指定目的地之前會先緩衝資料。建議的目的地緩衝區大小因服務提供者而異。

設定 OpenSearch Serverless 的目的地設定

本節會描述使用 OpenSearch Serviceless 作為目的地的選項。

- 為下列欄位輸入值：

OpenSearch Serverless 集合

資料要交付至的 OpenSearch Serverless 索引群組的端點。

索引

將資料索引至您的 OpenSearch Serviceless 集合時，所使用的 OpenSearch Serviceless 索引名稱。

目的地 VPC 連線能力

如果您的 OpenSearch Serviceless 集合位於私有 VPC 中，請使用此區段來指定該 VPC。同時，請指定 Amazon Data Firehose 將資料傳送至您的 OpenSearch Serverless 集合時，要使用的子網路和子群組。

Important

當您指定子網路以將資料交付至私有 VPC 中的目的地時，請確定您選擇的子網路中有足夠的可用 IP 地址。如果指定的子網路中沒有可用的可用 IP 地址，Firehose 無法為私有 VPC 中的資料交付建立或新增 ENIs，且交付將會降級或失敗。

Retry duration (重試持續時間)

如果對 OpenSearch Serverless 的索引請求失敗，Firehose 重試的持續時間。對於重試持續時間，您可以設定介於 0-7200 秒之間的任何值。預設的重試持續時間為 300 秒。Firehose 將重試多次並指數退避，直到重試持續時間過期為止。

重試持續時間到期後，Firehose 會將資料交付至已設定的 S3 錯誤儲存貯體無效字母佇列 (DLQ)。對於交付至 DLQ 的資料，您必須將資料從設定的 S3 錯誤儲存貯體重新驅動回 OpenSearch Serverless 目的地。

如果因為 OpenSearch Serverless 叢集的停機時間或維護而想要封鎖 Firehose 串流將資料交付至 DLQ，您可以將重試持續時間設定為較高的值，以秒為單位。您可以聯絡 [AWS 支援](#)，將超過 7200 秒的重試持續時間值增加到。

緩衝提示

Amazon Data Firehose 在將傳入資料傳遞至指定目的地之前會先緩衝資料。建議的目的地緩衝區大小因服務提供者而異。

設定 HTTP 端點的目的地設定

本節會描述使用 HTTP 端點作為目的地的選項。

⚠ Important

如果您選擇 HTTP 端點作為目的地，則請檢閱並遵循 [了解 HTTP 端點交付請求和回應規格](#) 中的指示。

- 為下列欄位提供值：

HTTP 端點名稱 - 選用

指定 HTTP 端點的使用者易記名稱。例如 My HTTP Endpoint Destination。

HTTP 端點 URL

請使用下列格式指定 HTTP 端點的 URL：https://xyz.httpendpoint.com。該 URL 必須為 HTTPS URL。

身分驗證

您可以選擇直接輸入存取金鑰，或從擷取秘密 AWS Secrets Manager 以存取 HTTP 端點。

- (選用) 存取金鑰

如果您需要取得存取金鑰，以便從 Firehose 將資料交付至其端點，請聯絡端點擁有者。

- Secret

從 選取 AWS Secrets Manager 包含 HTTP 端點存取金鑰的秘密。如果您在下拉式清單中看不到秘密，請在 中 AWS Secrets Manager 為存取金鑰建立一個秘密。如需詳細資訊，請參閱 [在 Amazon Data Firehose AWS Secrets Manager 中使用 驗證](#)。

內容編碼

Amazon Data Firehose 會使用內容編碼來壓縮請求的內文，然後再將其傳送至目的地。選擇 GZIP 或已停用以啟用/停用請求的內容編碼。

Retry duration (重試持續時間)

指定 Amazon Data Firehose 重試將資料傳送至所選 HTTP 端點的時間長度。

傳送資料後，Amazon Data Firehose 會先等待來自 HTTP 端點的確認。如果發生錯誤，或是在逾時期間內未收到確認訊息，則 Amazon Data Firehose 會開始計算重試持續時間。在重試持續時間結束之前，該服務會持續重試。在此之後，Amazon Data Firehose 會將其視為資料傳遞失敗，然後將資料備份到 Amazon S3 儲存貯體。

每次 Amazon Data Firehose 將資料傳送至 HTTP 端點（初始嘗試或重試）時，都會重新啟動確認逾時計數器，並等待來自 HTTP 端點的確認。

即使重試持續時間過期，Amazon Data Firehose 仍會等待確認，直到收到確認或達到確認逾時期間為止。如果確認逾時，Amazon Data Firehose 會判斷重試計數器中是否剩餘時間。如果還有時間，將再次重試並重複執行邏輯，直到收到確認或判斷已無重試時間為止。

如果您不希望 Amazon Data Firehose 重試傳送資料，請將此值設為 0。

參數 - 選用

Amazon Data Firehose 會在每個 HTTP 呼叫中包含這些金鑰值對。這些參數可協助您識別和整理目的地。

緩衝提示

Amazon Data Firehose 在將傳入資料傳遞至指定目的地之前會先緩衝資料。建議的目的地緩衝區大小因服務提供者而異。

Important

對於 HTTP 端點目的地，如果您在 CloudWatch Logs 中看到目的地端點的 413 個回應代碼，請降低 Firehose 串流上的緩衝提示大小，然後再試一次。

設定 Datadog 的目的地設定

本節說明使用 Datadog 作為目的地的選項。[如需有關 Datadog 的詳細資訊，請參閱 https://docs.datadoghq.com/integrations/amazon_web_services/](https://docs.datadoghq.com/integrations/amazon_web_services/)。

- 提供下列欄位的值。

HTTP 端點 URL

在下拉式功能表中選擇您要從下列其中一個選項傳送資料的位置。

- Datadog 日誌 - US1
- Datadog 日誌 - US3
- Datadog 日誌 - US5
- Datadog 日誌 - AP1

- Datadog 日誌 - EU
- Datadog 日誌 - GOV
- Datadog 指標 - US
- Datadog 指標 - US5
- Datadog 指標 - AP1
- Datadog 指標 - EU
- Datadog 組態 - US1
- Datadog 組態 - US3
- Datadog 組態 - US5
- Datadog 組態 - AP1
- Datadog 組態 - 歐洲
- Datadog 組態 - US GOV

身分驗證

您可以選擇直接輸入 API 金鑰，或從擷取秘密 AWS Secrets Manager 以存取 Datadog。

- API 金鑰

請聯絡 Datadog 取得您需要的 API 金鑰，以便從 Firehose 將資料交付至此端點。

- Secret

從 選取 AWS Secrets Manager 包含 Datadog API 金鑰的秘密。如果您在下拉式清單中看不到秘密，請在 中建立一個秘密 AWS Secrets Manager。如需詳細資訊，請參閱 [在 Amazon Data Firehose AWS Secrets Manager 中使用 驗證](#)。

內容編碼

Amazon Data Firehose 會使用內容編碼來壓縮請求的內文，然後再將其傳送至目的地。選擇 GZIP 或已停用以啟用/停用請求的內容編碼。

Retry duration (重試持續時間)

指定 Amazon Data Firehose 重試將資料傳送至所選 HTTP 端點的時間長度。

傳送資料後，Amazon Data Firehose 會先等待來自 HTTP 端點的確認。如果發生錯誤，或是在逾時期間內未收到確認訊息，則 Amazon Data Firehose 會開始計算重試持續時間。在重試持續時間結束之前，該服務會持續重試。在此之後，Amazon Data Firehose 會將其視為資料傳遞失敗，然後將資料備份到 Amazon S3 儲存貯體。

每次 Amazon Data Firehose 將資料傳送至 HTTP 端點（初始嘗試或重試）時，都會重新啟動確認逾時計數器，並等待來自 HTTP 端點的確認。

即使重試持續時間過期，Amazon Data Firehose 仍會等待確認，直到收到確認或達到確認逾時期間為止。如果確認逾時，Amazon Data Firehose 會判斷重試計數器中是否剩餘時間。如果還有時間，將再次重試並重複執行邏輯，直到收到確認或判斷已無重試時間為止。

如果您不希望 Amazon Data Firehose 重試傳送資料，請將此值設為 0。

參數 - 選用

Amazon Data Firehose 會在每個 HTTP 呼叫中包含這些金鑰值對。這些參數可協助您識別和整理目的地。

緩衝提示

Amazon Data Firehose 在將傳入資料傳遞至指定目的地之前會先緩衝資料。建議的目的地緩衝區大小因服務提供者而異。

設定 Honeycomb 的目的地設定

本節說明使用 Honeycomb 作為目的地的選項。如需有關 Honeycomb 的詳細資訊，請參閱 <https://docs.honeycomb.io/getting-data-in/metrics/aws-cloudwatch-metrics/>。

- 為下列欄位提供值：

Honeycomb Kinesis 端點

請使用下列格式指定 HTTP 端點的 URL：`https://api.honeycomb.io/1/kinesis_events/{{dataset}}`

身分驗證

您可以選擇直接輸入 API 金鑰，或從擷取秘密 AWS Secrets Manager 以存取 Honeycomb。

- API 金鑰

請聯絡 Honeycomb 以取得您需要的 API 金鑰，以便從 Firehose 將資料交付至此端點。

- Secret

從 選取 AWS Secrets Manager 包含 Honeycomb API 金鑰的秘密。如果您在下拉式清單中看不到秘密，請在 中建立一個秘密 AWS Secrets Manager。如需詳細資訊，請參閱在 [Amazon Data Firehose AWS Secrets Manager 中使用 驗證](#)。

內容編碼

Amazon Data Firehose 會使用內容編碼來壓縮請求的內文，然後再將其傳送至目的地。選擇 GZIP 以啟用要求的內容編碼。這是 Honeycomb 目的地的建議選項。

Retry duration (重試持續時間)

指定 Amazon Data Firehose 重試將資料傳送至所選 HTTP 端點的時間長度。

傳送資料後，Amazon Data Firehose 會先等待來自 HTTP 端點的確認。如果發生錯誤，或是在逾時期間內未收到確認訊息，則 Amazon Data Firehose 會開始計算重試持續時間。在重試持續時間結束之前，該服務會持續重試。在此之後，Amazon Data Firehose 會將其視為資料傳遞失敗，然後將資料備份到 Amazon S3 儲存貯體。

每次 Amazon Data Firehose 將資料傳送至 HTTP 端點（初始嘗試或重試）時，都會重新啟動確認逾時計數器，並等待來自 HTTP 端點的確認。

即使重試持續時間過期，Amazon Data Firehose 仍會等待確認，直到收到確認或達到確認逾時期間為止。如果確認逾時，Amazon Data Firehose 會判斷重試計數器中是否剩餘時間。如果還有時間，將再次重試並重複執行邏輯，直到收到確認或判斷已無重試時間為止。

如果您不希望 Amazon Data Firehose 重試傳送資料，請將此值設為 0。

參數 - 選用

Amazon Data Firehose 會在每個 HTTP 呼叫中包含這些金鑰值對。這些參數可協助您識別和整理目的地。

緩衝提示

Amazon Data Firehose 在將傳入資料傳遞至指定目的地之前會先緩衝資料。建議的目的地緩衝區大小因服務提供者而異。

設定 Coralogix 的目的地設定

本節會描述使用 Coralogix 作為目的地的選項。如需 Coralogix 的詳細資訊，請參閱 [Coralogix 入門](#)。

- 為下列欄位提供值：

HTTP 端點 URL

從下拉式功能表的下列選項中選擇 HTTP 端點 URL：

- Coralogix - US
- Coralogix - SINGAPORE
- Coralogix - IRELAND
- Coralogix - INDIA
- Coralogix - STOCKHOLM

身分驗證

您可以選擇直接輸入私有金鑰，或從擷取秘密 AWS Secrets Manager 以存取 Coralogix。

- 私有金鑰

請聯絡 Coralogix 取得您需要的私有金鑰，以便從 Firehose 將資料交付至此端點。

- Secret

從 選取 AWS Secrets Manager 包含 Coralogix 私有金鑰的秘密。如果您在下拉式清單中看不到秘密，請在 中建立一個秘密 AWS Secrets Manager。如需詳細資訊，請參閱[在 Amazon Data Firehose AWS Secrets Manager 中使用 驗證](#)。

內容編碼

Amazon Data Firehose 會使用內容編碼來壓縮請求的內文，然後再將其傳送至目的地。選擇 GZIP 以啟用要求的內容編碼。這是 Coralogix 目的地的建議選項。

Retry duration (重試持續時間)

指定 Amazon Data Firehose 重試將資料傳送至所選 HTTP 端點的時間長度。

傳送資料後，Amazon Data Firehose 會先等待來自 HTTP 端點的確認。如果發生錯誤，或是在逾時期間內未收到確認訊息，則 Amazon Data Firehose 會開始計算重試持續時間。在重試持續時間結束之前，該服務會持續重試。在此之後，Amazon Data Firehose 會將其視為資料傳遞失敗，然後將資料備份到 Amazon S3 儲存貯體。

每次 Amazon Data Firehose 將資料傳送至 HTTP 端點（初始嘗試或重試）時，都會重新啟動確認逾時計數器，並等待來自 HTTP 端點的確認。

即使重試持續時間過期，Amazon Data Firehose 仍會等待確認，直到收到確認或達到確認逾時期間為止。如果確認逾時，Amazon Data Firehose 會判斷重試計數器中是否剩餘時間。如果還有時間，將再次重試並重複執行邏輯，直到收到確認或判斷已無重試時間為止。

如果您不希望 Amazon Data Firehose 重試傳送資料，請將此值設為 0。

參數 - 選用

Amazon Data Firehose 會在每個 HTTP 呼叫中包含這些金鑰值對。這些參數可協助您識別和整理目的地。

- `applicationName`：執行 Data Firehose 的環境
- `subsystemName`：資料 Firehose 整合的名稱
- `computerName`：使用中的 Firehose 串流名稱

緩衝提示

Amazon Data Firehose 在將傳入資料傳遞至指定目的地之前會先緩衝資料。建議的目的地緩衝區大小會根據服務供應商而有所不同。

設定 Dynatrace 的目的地設定

本節會描述使用 Dynatrace 作為目的地的選項。如需詳細資訊，請參閱 <https://www.dynatrace.com/support/help/technology-support/cloud-platforms/amazon-web-services/integrations/cloudwatch-metric-streams/>。

- 選擇選項以使用 Dynatrace 做為 Firehose 串流的目的地。

擷取類型

選擇是否要在 Dynatrace 中交付指標或日誌（預設），以進行進一步分析和處理。

HTTP 端點 URL

從下拉式選單中選擇 HTTP 端點 URL (Dynatrace US、Dynatrace EU 或 Dynatrace Global)。

身分驗證

您可以選擇直接輸入 API 字符，或從擷取秘密 AWS Secrets Manager 以存取 Dynatrace。

- API 記號

產生您需要的 Dynatrace API 字符，以便從 Firehose 將資料交付至此端點。如需詳細資訊，請參閱 [Dynatrace API - 權杖和身分驗證](#)。

- Secret

從 選取 AWS Secrets Manager 包含 Dynatrace API 字符的秘密。如果您在下拉式清單中看不到秘密，請在 中建立一個秘密 AWS Secrets Manager。如需詳細資訊，請參閱在 [Amazon Data Firehose AWS Secrets Manager 中使用 驗證](#)。

API URL

提供您的 Dynatrace 環境的 API URL。

內容編碼

選擇是否要啟用內容編碼來壓縮請求的內文。Amazon Data Firehose 使用內容編碼來壓縮請求的內文，然後再將其傳送至目的地。啟用時，會以 GZIP 格式壓縮的內容。

Retry duration (重試持續時間)

指定 Firehose 重試將資料傳送至所選 HTTP 端點的時間長度。

傳送資料後，Firehose 會先等待來自 HTTP 端點的確認。如果發生錯誤，或是在逾時期間內未收到確認訊息，則 Firehose 會開始計算重試持續時間。在重試持續時間結束之前，該服務會持續重試。在此之後，Firehose 會將其視為資料傳遞失敗，然後將資料備份到 Amazon S3 儲存貯體。

每次 Firehose 在初次嘗試期間或重試後將資料傳送至 HTTP 端點時，都會重新啟動確認逾時計數器，並等待來自 HTTP 端點的確認。

即使重試持續時間過期，Firehose 仍會等待確認，直到收到確認或達到確認逾時期間為止。如果確認逾時，Firehose 會判斷重試計數器中是否剩餘時間。如果還有時間，將再次重試並重複執行邏輯，直到收到確認或判斷已無重試時間為止。

如果您不希望 Firehose 重試傳送資料，請將此值設為 0。

參數 - 選用

Amazon Data Firehose 會在每個 HTTP 呼叫中包含這些金鑰值對。這些參數可協助您識別和整理目的地。

緩衝提示

Amazon Data Firehose 在將傳入資料傳遞至指定目的地之前會先緩衝資料。緩衝提示包含串流的緩衝區大小和間隔。建議的目的地緩衝區大小會根據服務供應商而有所不同。

設定 LogicMonitor 的目的地設定

本節會描述使用 LogicMonitor 作為目的地的選項。如需詳細資訊，請參閱 <https://www.logicmonitor.com>。

- 為下列欄位提供值：

HTTP 端點 URL

以下列格式指定 HTTP 端點的 URL。

```
https://ACCOUNT.logicmonitor.com
```

身分驗證

您可以選擇直接輸入 API 金鑰，或從擷取秘密 AWS Secrets Manager 以存取 LogicMonitor。

- API 金鑰

請聯絡 LogicMonitor，以取得從 Firehose 啟用資料交付至此端點所需的 API 金鑰。

- Secret

從選取 AWS Secrets Manager 包含 LogicMonitor API 金鑰的秘密。如果您在下拉式清單中看不到秘密，請在 中建立一個秘密 AWS Secrets Manager。如需詳細資訊，請參閱 [在 Amazon Data Firehose AWS Secrets Manager 中使用 驗證](#)。

內容編碼

Amazon Data Firehose 使用內容編碼來壓縮請求的內文，然後再將其傳送至目的地。選擇 GZIP 或已停用以啟用/停用請求的內容編碼。

Retry duration (重試持續時間)

指定 Amazon Data Firehose 重試將資料傳送至所選 HTTP 端點的時間長度。

傳送資料後，Amazon Data Firehose 會先等待來自 HTTP 端點的確認。如果發生錯誤，或是在逾時期間內未收到確認訊息，則 Amazon Data Firehose 會開始計算重試持續時間。在重試持續時間結束之前，該服務會持續重試。在此之後，Amazon Data Firehose 會將其視為資料傳遞失敗，然後將資料備份到 Amazon S3 儲存貯體。

每次 Amazon Data Firehose 將資料傳送至 HTTP 端點（初始嘗試或重試）時，都會重新啟動確認逾時計數器，並等待來自 HTTP 端點的確認。

即使重試持續時間過期，Amazon Data Firehose 仍會等待確認，直到收到確認或達到確認逾時期間為止。如果確認逾時，Amazon Data Firehose 會判斷重試計數器中是否剩餘時間。如果還有時間，將再次重試並重複執行邏輯，直到收到確認或判斷已無重試時間為止。

如果您不希望 Amazon Data Firehose 重試傳送資料，請將此值設為 0。

參數 - 選用

Amazon Data Firehose 會在每個 HTTP 呼叫中包含這些金鑰值對。這些參數可協助您識別和整理目的地。

緩衝提示

Amazon Data Firehose 在將傳入資料傳遞至指定目的地之前會先緩衝資料。建議的目的地緩衝區大小因服務提供者而異。

設定 Logz.io 的目的地設定

本節說明使用 Logz.io 作為目的地的選項。如需詳細資訊，請參閱 <https://logz.io/>。

Note

在歐洲 (米蘭) 區域中，Logz.io 並非支援的 Amazon Data Firehose 目的地。

- 為下列欄位提供值：

HTTP 端點 URL

以下列格式指定 HTTP 端點的 URL。URL 必須是 HTTPS URL。

```
https://listener-aws-metrics-stream-<region>.logz.io/
```

例如

```
https://listener-aws-metrics-stream-us.logz.io/
```

身分驗證

您可以選擇直接輸入運送字符，或從擷取秘密 AWS Secrets Manager 以存取 Logz.io。

- 運送字符

請聯絡 Logz.io : //。

- Secret

從 選取 AWS Secrets Manager 包含 Logz.io 運送字符的秘密。如果您在下拉式清單中看不到秘密，請在 中建立一個秘密 AWS Secrets Manager。如需詳細資訊，請參閱 [在 Amazon Data Firehose AWS Secrets Manager 中使用 驗證](#)。

Retry duration (重試持續時間)

指定 Amazon Data Firehose 重試將資料傳送至 Logz.io 的時間長度。

傳送資料後，Amazon Data Firehose 會先等待來自 HTTP 端點的確認。如果發生錯誤，或是在逾時期間內未收到確認訊息，則 Amazon Data Firehose 會開始計算重試持續時間。在重試持續時間結束之前，該服務會持續重試。在此之後，Amazon Data Firehose 會將其視為資料傳遞失敗，然後將資料備份到 Amazon S3 儲存貯體。

每次 Amazon Data Firehose 將資料傳送至 HTTP 端點（初始嘗試或重試）時，都會重新啟動確認逾時計數器，並等待來自 HTTP 端點的確認。

即使重試持續時間過期，Amazon Data Firehose 仍會等待確認，直到收到確認或達到確認逾時期間為止。如果確認逾時，Amazon Data Firehose 會判斷重試計數器中是否剩餘時間。如果還有時間，將再次重試並重複執行邏輯，直到收到確認或判斷已無重試時間為止。

如果您不希望 Amazon Data Firehose 重試傳送資料，請將此值設為 0。

參數 - 選用

Amazon Data Firehose 會在每個 HTTP 呼叫中包含這些金鑰值對。這些參數可協助您識別和整理目的地。

緩衝提示

Amazon Data Firehose 在將傳入資料傳遞至指定目的地之前會先緩衝資料。建議的目的地緩衝區大小因服務提供者而異。

設定 MongoDB Atlas 的目的地設定

本節說明針對目的地使用 MongoDB Atlas 的選項。如需詳細資訊，請參閱 [Amazon Web Services 上的 MongoDB Atlas](#)。

- 為下列欄位提供值：

API Gateway URL

以下列格式指定 HTTP 端點的 URL。

```
https://xxxxx.execute-api.region.amazonaws.com/stage
```

URL 必須是 HTTPS URL。

身分驗證

您可以選擇直接輸入 API 金鑰，或從擷取秘密 AWS Secrets Manager 以存取 MongoDB Atlas。

- API 金鑰

請遵循 [Amazon Web Services 上的 MongoDB Atlas](#) 中的指示，取得APIKeyValue從 Firehose 啟用資料交付至此端點所需的。

- Secret

從 選取秘密 AWS Secrets Manager ，其中包含由 Lambda 與 MongoDB Atlas 互動所支援的 API Gateway API 金鑰。如果您在下拉式清單中看不到秘密，請在 中建立一個秘密 AWS Secrets Manager。如需詳細資訊，請參閱 [在 Amazon Data Firehose AWS Secrets Manager 中使用 驗證](#)。

內容編碼

Amazon Data Firehose 使用內容編碼來壓縮請求的內文，然後再將其傳送至目的地。選擇 GZIP 或已停用以啟用/停用請求的內容編碼。

Retry duration (重試持續時間)

指定 Amazon Data Firehose 重試將資料傳送至所選第三方供應商的時間長度。

傳送資料後，Amazon Data Firehose 會先等待來自 HTTP 端點的確認。如果發生錯誤，或是在逾時期間內未收到確認訊息，則 Amazon Data Firehose 會開始計算重試持續時間。在重試持續時間結束之前，該服務會持續重試。在此之後，Amazon Data Firehose 會將其視為資料傳遞失敗，然後將資料備份到 Amazon S3 儲存貯體。

每次 Amazon Data Firehose 將資料傳送至 HTTP 端點（初始嘗試或重試）時，都會重新啟動確認逾時計數器，並等待來自 HTTP 端點的確認。

即使重試持續時間過期，Amazon Data Firehose 仍會等待確認，直到收到確認或達到確認逾時期間為止。如果確認逾時，Amazon Data Firehose 會判斷重試計數器中是否剩餘時間。如果還有時間，將再次重試並重複執行邏輯，直到收到確認或判斷已無重試時間為止。

如果您不希望 Amazon Data Firehose 重試傳送資料，請將此值設為 0。

緩衝提示

Amazon Data Firehose 在將傳入資料傳遞至指定目的地之前會先緩衝資料。建議的目的地緩衝區大小因服務提供者而異。

參數 - 選用

Amazon Data Firehose 會在每個 HTTP 呼叫中包含這些金鑰值對。這些參數可協助您識別和整理目的地。

設定新複本的目的地設定

本節會描述使用 New Relic 作為目的地的選項。如需詳細資訊，請參閱 <https://newrelic.com>。

- 為下列欄位提供值：

HTTP 端點 URL

從下拉式清單的下列選項中選擇 HTTP 端點 URL。

- New Relic 日誌 - US
- New Relic 指標 - US
- New Relic 指標 - EU

身分驗證

您可以選擇直接輸入 API 金鑰，或從擷取秘密 AWS Secrets Manager 以存取 New Relic。

- API 金鑰

從 New Relic One 帳戶設定輸入您的授權金鑰，這是 40 個字元的十六進位字串。您需要此 API 金鑰，才能從 Firehose 將資料交付至此端點。

- Secret

從 選取 AWS Secrets Manager 包含新複本 API 金鑰的秘密。如果您在下拉式清單中看不到秘密，請在 中建立一個秘密 AWS Secrets Manager。如需詳細資訊，請參閱 [在 Amazon Data Firehose AWS Secrets Manager 中使用 驗證](#)。

內容編碼

Amazon Data Firehose 使用內容編碼來壓縮請求的內文，然後再將其傳送至目的地。選擇 GZIP 或已停用以啟用/停用請求的內容編碼。

Retry duration (重試持續時間)

指定 Amazon Data Firehose 重試將資料傳送至新複本 HTTP 端點的時間長度。

傳送資料後，Amazon Data Firehose 會先等待來自 HTTP 端點的確認。如果發生錯誤，或是在逾時期間內未收到確認訊息，則 Amazon Data Firehose 會開始計算重試持續時間。在重試持續時間結束之前，該服務會持續重試。在此之後，Amazon Data Firehose 會將其視為資料傳遞失敗，然後將資料備份到 Amazon S3 儲存貯體。

每次 Amazon Data Firehose 將資料傳送至 HTTP 端點（初始嘗試或重試）時，都會重新啟動確認逾時計數器，並等待來自 HTTP 端點的確認。

即使重試持續時間過期，Amazon Data Firehose 仍會等待確認，直到收到確認或達到確認逾時期間為止。如果確認逾時，Amazon Data Firehose 會判斷重試計數器中是否剩餘時間。如果還有時間，將再次重試並重複執行邏輯，直到收到確認或判斷已無重試時間為止。

如果您不希望 Amazon Data Firehose 重試傳送資料，請將此值設為 0。

參數 - 選用

Amazon Data Firehose 會在每個 HTTP 呼叫中包含這些金鑰值對。這些參數可協助您識別和整理目的地。

緩衝提示

Amazon Data Firehose 在將傳入資料傳遞至指定目的地之前會先緩衝資料。建議的目的地緩衝區大小因服務提供者而異。

設定 Snowflake 的目的地設定

本節說明將 Snowflake 用於目的地的選項。

Note

Firehose 與 Snowflake 的整合已在美國東部（維吉尼亞北部）、美國西部（奧勒岡）、歐洲（愛爾蘭）、美國東部（俄亥俄）、亞太區域（東京）、歐洲（法蘭克福）、亞太區域（新加坡）、亞太區域（首爾）和亞太區域（雪梨）、亞太區域（孟買）、歐洲（倫敦）、南美洲（聖保羅）、加拿大（中部）、歐洲（巴黎）、亞太區域（大阪）、歐洲（斯德哥爾摩）、亞太區域（雅加達）AWS 區域提供。

連線設定

- 為下列欄位提供值：

Snowflake 帳戶 URL

指定 Snowflake 帳戶 URL。例如：`xy12345.us-east-1.aws.snowflakecomputing.com`。請參閱 [Snowflake 文件](#)，了解如何判斷您的帳戶 URL。請注意，您不可以指定連接埠號碼，而通訊協定 (`https://`) 是選用的。

身分驗證

您可以選擇手動輸入 `userlogin`、私有金鑰和密碼短語，或從擷取秘密 AWS Secrets Manager 以存取 Snowflake。

- 使用者登入

指定要用於載入資料的 Snowflake 使用者。確認使用者具有存取權，可將資料插入 Snowflake 資料表中。

- 私有金鑰

指定要以 PKCS8 格式使用 Snowflake 進行身分驗證的私有金鑰。此外，請勿將 PEM 標頭和頁尾包含在私有金鑰中。如果金鑰分成多行，請移除換行符號。以下是私有金鑰必須看起來的範例。

```
-----BEGIN PRIVATE KEY-----  
KEY_CONTENT  
-----END PRIVATE KEY-----
```

移除 中的空間，KEY_CONTENT 並將其提供給 Firehose。不需要標頭/頁尾或換行字元。

- Passphrase (密碼短語)

指定密碼來解密經加密的私密金鑰。如果私密金鑰未加密，您可以將此欄位保留空白。如需詳細資訊，請參閱 [使用金鑰對身分驗證和金鑰輪換](#)。

- Secret

從 選取 AWS Secrets Manager 包含 Snowflake 登入資料的秘密。如果您在下拉式清單中看不到秘密，請在 中建立一個秘密 AWS Secrets Manager。如需詳細資訊，請參閱 [在 Amazon Data Firehose AWS Secrets Manager 中使用 驗證](#)。

角色組態

使用預設 Snowflake 角色 - 如果選取此選項，Firehose 就不會將任何角色傳遞至 Snowflake。此時會擔任預設角色以載入資料。請確認預設角色具有將資料插入 Snowflake 資料表的許可。

使用自訂 Snowflake 角色 - 輸入將資料載入 Snowflake 資料表時由 Firehose 擔任的非預設 Snowflake 角色。

Snowflake 連線能力

選項為私有或公有。

私有 VPCE ID (選用)

VPCE ID 可讓 Firehose 透過私有方式與 Snowflake 連線。ID 格式為 com.amazonaws.vpce.[region].vpce-svc-[id]。如需詳細資訊，請參閱 [AWS PrivateLink & Snowflake](#)。

Note

如果您的 Snowflake 叢集已啟用私有連結，請使用 AwsVpceIds 型網路政策來允許 Amazon Data Firehose 資料。Firehose 不需要您在 Snowflake 帳戶中設定 IP 型網路政策。啟用 IP 型網路政策可能會干擾 Firehose 連線。如果您有需要 IP 型政策的邊緣

案例，請提交[支援票證](#)來聯絡 Firehose 團隊。如需您可以使用的 VPCE IDs 清單，請參閱 [在 VPC 中存取 Snowflake](#)。

資料庫組態

- 您必須指定以下設定，才能使用 Snowflake 作為 Firehose 串流目的地。
 - Snowflake 資料庫 - Snowflake 中的所有資料都保留在資料庫中。
 - Snowflake 結構描述 - 每個資料庫由一或多個結構描述組成，這些是資料庫物件的邏輯分組，例如資料表和視圖
 - Snowflake 資料表 - Snowflake 中的所有資料都儲存在資料庫資料表中，邏輯結構化為欄和列的集合。

Snowflake 資料表的資料載入選項

- 使用 JSON 金鑰做為資料欄名稱
- 使用 VARIANT 資料欄
 - 內容欄名稱 – 在資料表中指定欄名稱，其中必須載入原始資料。
 - 中繼資料資料欄名稱（選用） – 在資料表中指定資料欄名稱，其中必須載入中繼資料資訊。當您啟用此欄位時，會根據來源類型，在 Snowflake 資料表中看到下列資料欄。

對於做為來源的直接 PUT

```
{
  "firehoseDeliveryStreamName" : "streamname",
  "IngestionTime" : "timestamp"
}
```

對於 Kinesis Data Stream 作為來源

```
{
  "kinesisStreamName" : "streamname",
  "kinesisShardId" : "Id",
  "kinesisPartitionKey" : "key",
  "kinesisSequenceNumber" : "1234",
  "subsequenceNumber" : "2334",
  "IngestionTime" : "timestamp"
}
```

```
}
```

Retry duration (重試持續時間)

如果因 Snowflake 服務問題而開啟頻道或交付至 Snowflake 失敗，Firehose 重試的持續時間 (0–7200 秒)。Firehose 會以指數退避重試，直到重試持續時間結束為止。如果您將重試持續時間設定為 0 (零) 秒，Firehose 不會在 Snowflake 失敗時重試，並將資料路由到 Amazon S3 錯誤儲存貯體。

緩衝提示

Amazon Data Firehose 在將傳入資料傳遞至指定目的地之前會先緩衝資料。建議的目的地緩衝區大小因服務提供者而異。如需詳細資訊，請參閱[設定緩衝提示](#)。

設定 Splunk 的目的地設定

本節說明使用 Splunk 做為目的地的選項。

Note

Firehose 會將資料交付至使用 Classic Load Balancer 或 Application Load Balancer 設定的 Splunk 叢集。

- 為下列欄位提供值：

Splunk cluster endpoint (Splunk 叢集端點)

若要判斷端點，請參閱 [Splunk 文件中的設定 Amazon Data Firehose 將資料傳送至 Splunk 平台](#)。

Splunk endpoint type (Splunk 端點類型)

在大多數情況下，請選擇 Raw endpoint。選擇 Event endpoint 您是否使用 預先處理資料 AWS Lambda，以依事件類型將資料傳送至不同的索引。如需判斷該使用何種端點的相關資訊，請參閱 Splunk 文件中的 [設定 Amazon Data Firehose 以傳送資料至 Splunk 平台](#)。

身分驗證

您可以選擇直接輸入身分驗證字符，或從 擷取秘密 AWS Secrets Manager 以存取 Splunk。

- Authentication token (身分驗證字符)

若要設定可從 Amazon Data Firehose 接收資料的 Splunk 端點，請參閱 Splunk 文件中[適用於 Amazon Data Firehose 的 Splunk 附加元件的安裝和組態概觀](#)。儲存您在設定此 Firehose 串流的端點時從 Splunk 取得的字符，並將其新增至此處。

- Secret

從 選取 AWS Secrets Manager 包含 Splunk 身分驗證字符的秘密。如果您在下拉式清單中看不到秘密，請在 中建立一個秘密 AWS Secrets Manager。如需詳細資訊，請參閱在[Amazon Data Firehose AWS Secrets Manager 中使用 驗證](#)。

HEC acknowledgement timeout (HEC 確認訊息逾時)

指定 Amazon Data Firehose 等待 Splunk 確認索引的時間長度。如果 Splunk 未在達到逾時之前傳送確認，Amazon Data Firehose 會將其視為資料交付失敗。接著 Amazon Data Firehose 會重試或將資料備份到 Amazon S3 儲存貯體，依據您設定的重試持續時間值而定。

Retry duration (重試持續時間)

指定 Amazon Data Firehose 重試傳送資料至 Splunk 的時間長度。

傳送資料後，Amazon Data Firehose 會先等待來自 Splunk 的確認。如果發生錯誤，或是在逾時期間內未收到確認訊息，則 Amazon Data Firehose 會開始計算重試持續時間。在重試持續時間結束之前，該服務會持續重試。在此之後，Amazon Data Firehose 會將其視為資料傳遞失敗，然後將資料備份到 Amazon S3 儲存貯體。

每次 Amazon Data Firehose 將資料傳送至 Splunk（初始嘗試或重試）時，都會重新啟動確認逾時計數器，並等待 Splunk 的確認。

即使重試持續時間過期，Amazon Data Firehose 仍會等待確認，直到收到確認或達到確認逾時期間為止。如果確認逾時，Amazon Data Firehose 會判斷重試計數器中是否剩餘時間。如果還有時間，將再次重試並重複執行邏輯，直到收到確認或判斷已無重試時間為止。

如果您不希望 Amazon Data Firehose 重試傳送資料，請將此值設為 0。

緩衝提示

Amazon Data Firehose 在將傳入資料傳遞至指定目的地之前會先緩衝資料。建議的目的地緩衝區大小會根據服務供應商而有所不同。

設定 Splunk 可觀測性雲端的目的地設定

本節會描述使用 Splunk Observability Cloud 作為目的地的選項。如需詳細資訊，請參閱 <https://docs.splunk.com/observability/en/gdi/get-data-in/connect/aws/aws-apiconfig.html#connect-to-aws-using-the-splunk-observability-cloud-api>。

- 為下列欄位提供值：

雲端擷取端點 URL

您可以在 Splunk 可觀測性主控台的設定檔 > 組織 > 即時資料擷取端點中找到 Splunk 可觀測性雲端的即時資料擷取 URL。

身分驗證

您可以選擇直接輸入存取字符，或從擷取秘密 AWS Secrets Manager 以存取 Splunk 可觀測性雲端。

- 存取記號

在 Splunk Observability 主控台中，從設定下的存取權杖複製包含 INGEST 授權範圍在內的 Splunk Observability 存取權杖。

- Secret

從選取 AWS Secrets Manager 包含 Splunk 可觀測性雲端存取字符的秘密。如果您在下拉式清單中看不到秘密，請在中建立一個秘密 AWS Secrets Manager。如需詳細資訊，請參閱 [在 Amazon Data Firehose AWS Secrets Manager 中使用 驗證](#)。

內容編碼

Amazon Data Firehose 使用內容編碼來壓縮請求的內文，然後再將其傳送至目的地。選擇 GZIP 或已停用以啟用/停用請求的內容編碼。

Retry duration (重試持續時間)

指定 Amazon Data Firehose 重試將資料傳送至所選 HTTP 端點的時間長度。

傳送資料後，Amazon Data Firehose 會先等待來自 HTTP 端點的確認。如果發生錯誤，或是在逾時期間內未收到確認訊息，則 Amazon Data Firehose 會開始計算重試持續時間。在重試持續時間結束之前，該服務會持續重試。在此之後，Amazon Data Firehose 會將其視為資料傳遞失敗，然後將資料備份到 Amazon S3 儲存貯體。

每次 Amazon Data Firehose 將資料傳送至 HTTP 端點（初始嘗試或重試）時，都會重新啟動確認逾時計數器，並等待來自 HTTP 端點的確認。

即使重試持續時間過期，Amazon Data Firehose 仍會等待確認，直到收到確認或達到確認逾時期間為止。如果確認逾時，Amazon Data Firehose 會判斷重試計數器中是否剩餘時間。如果還有時間，將再次重試並重複執行邏輯，直到收到確認或判斷已無重試時間為止。

如果您不希望 Amazon Data Firehose 重試傳送資料，請將此值設為 0。

參數 - 選用

Amazon Data Firehose 會在每個 HTTP 呼叫中包含這些金鑰值對。這些參數可協助您識別和整理目的地。

緩衝提示

Amazon Data Firehose 在將傳入資料傳遞至指定目的地之前會先緩衝資料。建議的目的地緩衝區大小因服務提供者而異。

設定 Sumo Logic 的目的地設定

本節會描述使用 Sumo Logic 作為目的地的選項。如需詳細資訊，請參閱 <https://www.sumologic.com>。

- 為下列欄位提供值：

HTTP 端點 URL

請使用下列格式指定 HTTP 端點的 URL：`https://deployment name.sumologic.net/receiver/v1/kinesis/dataType/access token`。該 URL 必須為 HTTPS URL。

內容編碼

Amazon Data Firehose 使用內容編碼來壓縮請求的內文，然後再將其傳送至目的地。選擇 GZIP 或已停用以啟用/停用請求的內容編碼。

Retry duration (重試持續時間)

指定 Amazon Data Firehose 重試傳送資料到 Sumo Logic 的時間長度。

傳送資料後，Amazon Data Firehose 會先等待來自 HTTP 端點的確認。如果發生錯誤，或是在逾時期間內未收到確認訊息，則 Amazon Data Firehose 會開始計算重試持續時間。在重試持續時間結束之前，該服務會持續重試。在此之後，Amazon Data Firehose 會將其視為資料傳遞失敗，然後將資料備份到 Amazon S3 儲存貯體。

每次 Amazon Data Firehose 將資料傳送至 HTTP 端點（初始嘗試或重試）時，都會重新啟動確認逾時計數器，並等待來自 HTTP 端點的確認。

即使重試持續時間過期，Amazon Data Firehose 仍會等待確認，直到收到確認或達到確認逾時期間為止。如果確認逾時，Amazon Data Firehose 會判斷重試計數器中是否剩餘時間。如果還有時間，將再次重試並重複執行邏輯，直到收到確認或判斷已無重試時間為止。

如果您不希望 Amazon Data Firehose 重試傳送資料，請將此值設為 0。

參數 - 選用

Amazon Data Firehose 會在每個 HTTP 呼叫中包含這些金鑰值對。這些參數可協助您識別和整理目的地。

緩衝提示

Amazon Data Firehose 在將傳入資料傳遞至指定目的地之前會先緩衝資料。建議的 Elastic 目的地緩衝區大小因服務提供者而異。

設定彈性的目的地設定

本節會描述使用 Elastic 作為目的地的選項。

- 為下列欄位提供值：

Elastic 端點 URL

請使用下列格式指定 HTTP 端點的 URL：`https://<cluster-id>.es.<region>.aws.elastic-cloud.com`。該 URL 必須為 HTTPS URL。

身分驗證

您可以選擇直接輸入 API 金鑰，或從擷取秘密 AWS Secrets Manager 以存取 Elastic。

- API 金鑰

請聯絡 Elastic 取得您需要的 API 金鑰，以便從 Firehose 將資料交付至其服務。

- Secret

從 選取 AWS Secrets Manager 包含 Elastic API 金鑰的秘密。如果您在下拉式清單中看不到秘密，請在 中建立一個秘密 AWS Secrets Manager。如需詳細資訊，請參閱 [在 Amazon Data Firehose AWS Secrets Manager 中使用 驗證](#)。

內容編碼

Amazon Data Firehose 使用內容編碼來壓縮請求的內文，然後再將其傳送至目的地。選擇 GZIP (預設選擇) 或已停用以啟用/停用請求的內容編碼。

Retry duration (重試持續時間)

指定 Amazon Data Firehose 重試傳送資料到 Elastic 的時間長度。

傳送資料後，Amazon Data Firehose 會先等待來自 HTTP 端點的確認。如果發生錯誤，或是在逾時期間內未收到確認訊息，則 Amazon Data Firehose 會開始計算重試持續時間。在重試持續時間結束之前，該服務會持續重試。在此之後，Amazon Data Firehose 會將其視為資料傳遞失敗，然後將資料備份到 Amazon S3 儲存貯體。

每次 Amazon Data Firehose 將資料傳送至 HTTP 端點（初始嘗試或重試）時，都會重新啟動確認逾時計數器，並等待來自 HTTP 端點的確認。

即使重試持續時間過期，Amazon Data Firehose 仍會等待確認，直到收到確認或達到確認逾時期間為止。如果確認逾時，Amazon Data Firehose 會判斷重試計數器中是否剩餘時間。如果還有時間，將再次重試並重複執行邏輯，直到收到確認或判斷已無重試時間為止。

如果您不希望 Amazon Data Firehose 重試傳送資料，請將此值設為 0。

參數 - 選用

Amazon Data Firehose 會在每個 HTTP 呼叫中包含這些金鑰值對。這些參數可協助您識別和整理目的地。

緩衝提示

Amazon Data Firehose 在將傳入資料傳遞至指定目的地之前會先緩衝資料。彈性目的地的建議緩衝區大小為 1 MiB。

設定備份設定

Amazon Data Firehose 使用 Amazon S3 來備份其嘗試交付至所選目的地的所有或僅失敗的資料。

Important

- 只有在 Firehose 串流的來源為 Direct PUT 或 Kinesis Data Streams 時，才支援備份設定。
- 零緩衝功能僅適用於應用程式目的地，不適用於 Amazon S3 備份目的地。

如果您做出下列其中一個選擇，您可以指定 Firehose 串流的 S3 備份設定。

- 如果您將 Amazon S3 設定為 Firehose 串流的目的地，並選擇指定 AWS Lambda 函數來轉換資料記錄，或選擇轉換 Firehose 串流的資料記錄格式。
- 如果您將 Amazon Redshift 設定為 Firehose 串流的目的地，並選擇指定 AWS Lambda 函數來轉換資料記錄。
- 如果您將下列任何服務設定為 Firehose 串流的目的地：Amazon OpenSearch Service、Datadog、Dynatrace、HTTP Endpoint、LogicMonitor、MongoDB Cloud、New Relic、Splunk 或 Sumo Logic、Snowflake、Apache Iceberg Tables。

以下是 Firehose 串流的備份設定。

- Amazon S3 中的來源記錄備份 - 如果 S3 或 Amazon Redshift 是您選取的目的地，此設定會指出您是要啟用來源資料備份，還是保持停用狀態。如果將任何其他支援的服務 (S3 或 Amazon Redshift 除外) 設為您選取的目的地，則此設定會指示您是要備份所有來源資料，還是僅備份失敗的資料。
- S3 備份儲存貯體 - 這是 Amazon Data Firehose 備份資料的 S3 儲存貯體。
- S3 備份儲存貯體字首 - 這是 Amazon Data Firehose 備份資料的字首。
- S3 備份儲存貯體錯誤輸出字首 - 所有失敗的資料都會備份在此 S3 儲存貯體錯誤輸出字首中。
- 緩衝提示、壓縮和加密以進行備份 - Amazon Data Firehose 使用 Amazon S3 來備份其嘗試交付至所選目的地的所有或失敗資料。Amazon Data Firehose 會先緩衝傳入的資料，再將其交付（備份）至 Amazon S3。您可以選擇 1-128 MiBs 的緩衝區大小和 60-900 秒的緩衝區間隔。滿足的條件會先觸發資料交付至 Amazon S3。如果您啟用資料轉換，緩衝間隔會從 Amazon Data Firehose 收到轉換資料的時間套用到資料交付到 Amazon S3。如果交付至目的地的資料落後於寫入 Firehose 串流的資料，Amazon Data Firehose 會動態提高緩衝區大小以追上進度。此動作有助於確保所有資料都交付到目的地。

- S3 壓縮 - 選擇 GZIP、Snappy、Zip 或 Hadoop 相容 Snappy 資料壓縮，或沒有資料壓縮。Snappy、Zip 和 Hadoop 相容 Snappy 壓縮不適用於 Amazon Redshift 做為目的地的 Firehose 串流。
- S3 副檔名格式（選用） - 為交付至 Amazon S3 目的地儲存貯體的物件指定副檔名格式。如果您啟用此功能，指定的副檔名會覆寫資料格式轉換或 S3 壓縮功能附加的預設副檔名，例如 .parquet 或 .gz。當您將此功能與資料格式轉換或 S3 壓縮搭配使用時，請確定是否已設定正確的副檔名。副檔名必須以句號 (.) 開頭，且可包含允許的字元：0-9a-z ! - _ . * ()。副檔名不能超過 128 個字元。
- Firehose 支援使用 AWS Key Management Service (SSE-KMS) 的 Amazon S3 伺服器端加密，以加密 Amazon S3 中的交付資料。您可以選擇使用目的地 S3 儲存貯體中指定的預設加密類型，或使用您擁有的金鑰清單中的 AWS KMS 金鑰進行加密。如果您使用 AWS KMS 金鑰加密資料，您可以使用預設 AWS 受管金鑰 (aws/s3) 或客戶受管金鑰。如需詳細資訊，請參閱[使用伺服器端加密搭配 AWS KMS 受管金鑰保護資料 \(SSE-KMS\)](#)。

設定緩衝提示

Amazon Data Firehose 會將記憶體中的傳入串流資料緩衝至特定大小（緩衝大小）和一段特定時間（緩衝間隔），再交付至指定的目的地。當您想要將大小最佳的檔案交付至 Amazon S3，並從資料處理應用程式獲得更好的效能，或調整 Firehose 交付速率以符合目的地速度時，您會使用緩衝提示。

您可以在建立新的 Firehose 串流時設定緩衝大小和緩衝間隔，或更新現有 Firehose 串流上的緩衝大小和緩衝間隔。緩衝大小以 MBs 為單位，緩衝間隔以秒為單位。不過，如果指定它們其中一個值，您也必須提供其他的值。第一個滿足的緩衝條件會觸發 Firehose 交付資料。如果您未設定緩衝值，則會使用預設值。

您可以透過 AWS 管理主控台 AWS Command Line Interface、或 AWS SDKs 設定 Firehose 緩衝提示。對於現有的串流，您可以使用主控台內的編輯選項或使用 [UpdateDestination](#) API，以適合您的使用案例的值重新設定緩衝提示。對於新串流，您可以使用主控台或使用 [CreateDeliveryStream](#) API 將緩衝提示設定為新串流建立的一部分。若要調整緩衝大小，請在 [CreateDeliveryStream](#) 或 [UpdateDestination](#) API 的目的地特定 DestinationConfiguration 參數 IntervalInSeconds 中設定 SizeInMBs 和。

Note

- 緩衝提示會套用至碎片或分割區層級，而動態分割緩衝提示則會套用至串流或主題層級。
- 若要滿足較低的即時使用案例延遲，您可以使用零緩衝間隔提示。當您將緩衝間隔設定為零秒時，Firehose 不會緩衝資料，並在幾秒內交付資料。將緩衝提示變更為較低的值之前，請向廠商確認 Firehose 對其目的地的建議緩衝提示。

- 零緩衝功能僅適用於應用程式目的地，不適用於 Amazon S3 備份目的地。
- 零緩衝功能不適用於動態分割。
- 當您設定緩衝時間間隔少於 60 秒以提供較低的延遲時，Firehose 會對 S3 目的地使用分段上傳。由於 S3 目的地的分段上傳，如果您選擇緩衝時間間隔小於 60 秒，則 S3 PUT API 成本會稍微增加。

如需目的地特定的緩衝提示範圍和預設值，請參閱下表：

目標	緩衝大小，以 MB 為單位（括號中的預設值）	緩衝間隔，以秒為單位（括號中的預設值）
Amazon S3	1-128 (5)	0-900 (300)
Apache Iceberg 資料表	1-128 (5)	0-900 (300)
Amazon Redshift	1-128 (5)	0-900 (300)
OpenSearch Serverless	1-100 (5)	0-900 (300)
OpenSearch	1-100 (5)	0-900 (300)
Splunk	1-5 (5)	0-60 (60)
Datadog	1-4 (4)	0-900 (60)
Coralogix	1-64 (6)	0-900 (60)
Dynatrace	1-64 (5)	0-900 (60)
彈性	1	0-900 (60)
Honeycomb	1-64 (15)	0-900 (60)
HTTP 端點	1-64 (5)	0-900 (60)

目標	緩衝大小，以 MB 為單位（括號中的預設值）	緩衝間隔，以秒為單位（括號中的預設值）
LogicMonitor	1-64 (5)	0-900 (60)
Logzio	1-64 (5)	0-900 (60)
mongoDB	1-16 (5)	0-900 (60)
newRelic	1-64 (5)	0-900 (60)
sumoLogic	1-64 (1)	0-900 (60)
Splunk Observability Cloud	1-64 (1)	0-900 (60)
Snowflake	1 - 128 (1)	0 - 900 (0)

配置進階設定

下一節包含 Firehose 串流進階設定的詳細資訊。

- 伺服器端加密 - Amazon Data Firehose 支援使用 AWS Key Management Service (AWS KMS) 的 Amazon S3 伺服器端加密，以加密 Amazon S3 中的交付資料。如需詳細資訊，請參閱[使用伺服器端加密搭配 AWS KMS 受管金鑰保護資料 \(SSE-KMS\)](#)。
- 錯誤記錄 - Amazon Data Firehose 會記錄與處理和交付相關的錯誤。此外，啟用資料轉換後，它可以記錄 Lambda 調用，並將資料傳送錯誤傳送至 CloudWatch Logs。如需詳細資訊，請參閱[使用 CloudWatch Logs 監控 Amazon Data Firehose](#)。

Important

選用時，強烈建議在 Firehose 串流建立期間啟用 Amazon Data Firehose 錯誤記錄。此作法可確保在記錄處理或交付失敗時，您可以存取錯誤詳細資料。

- 許可 - Amazon Data Firehose 會將 IAM 角色用於 Firehose 串流所需的所有許可。您可以選擇在自動指派必要許可的情況下建立新的角色，或選擇為 Amazon Data Firehose 建立的現有角色。此角色用於授予 Firehose 對各種服務的存取權，包括 S3 儲存貯體、AWS KMS 金鑰（如果啟用資料加

密) 和 Lambda 函數 (如果啟用資料轉換)。主控台可能會建立具有預留位置的角色。如需詳細資訊，請參閱[什麼是 IAM?](#)。

Note

IAM 角色 (包括預留位置) 是根據您在建立 Firehose 串流時選擇的組態所建立。如果您對 Firehose 串流來源或目的地進行任何變更，您必須手動更新 IAM 角色。

- 標籤 - 您可以新增標籤來組織 AWS 資源、追蹤成本和控制存取。

如果您在 `CreateDeliveryStream` 動作中指定標籤，Amazon Data Firehose 會對 `firehose:TagDeliveryStream` 動作執行額外的授權，以驗證使用者是否具有建立標籤的許可。如果您未提供此許可，使用 IAM 資源標籤建立新 Firehose 串流的請求會失敗，`AccessDeniedException` 如下所示。

```
AccessDeniedException
User: arn:aws:sts::x:assumed-role/x/x is not authorized to perform:
  firehose:TagDeliveryStream on resource: arn:aws:firehose:us-east-1:x:deliverystream/
x with an explicit deny in an identity-based policy.
```

下列範例示範的政策可讓使用者建立 Firehose 串流並套用標籤。

選擇備份和進階設定後，請檢閱您的選擇，然後選擇建立 Firehose 串流。

新的 Firehose 串流在可用之前，需要幾分鐘處於建立狀態。您的 Firehose 串流處於作用中狀態後，您就可以開始從生產者傳送資料到該串流。

使用範例資料測試 Firehose 串流

您可以使用 AWS 管理主控台 來擷取模擬股票代號資料。主控台會在瀏覽器中執行指令碼，將範例記錄放入 Firehose 串流。這可讓您測試 Firehose 串流的組態，而不必產生自己的測試資料。

以下為取自模擬資料的範例：

```
{"TICKER_SYMBOL":"QXZ","SECTOR":"HEALTHCARE","CHANGE":-0.05,"PRICE":84.51}
```

請注意，當您的 Firehose 串流傳輸資料時，會收取標準 Amazon Data Firehose 費用，但在產生資料時不會收費。若要停止產生這些費用，您可以隨時從主控台停止範例串流。

先決條件

開始之前，請先建立 Firehose 串流。如需詳細資訊，請參閱[教學課程：從主控台建立 Firehose 串流](#)。

使用 Amazon S3 進行測試

使用下列程序，以 Amazon Simple Storage Service (Amazon S3) 做為目的地來測試 Firehose 串流。

使用 Amazon S3 測試 Firehose 串流

1. 開啟位於 <https://console.aws.amazon.com/firehose/> 的 Firehose 主控台。
2. 選擇作用中的 Firehose 串流。Firehose 串流必須處於作用中狀態，才能開始傳送資料。
3. 在 Test with demo data (以示範資料測試) 底下，選擇 Start sending demo data (開始傳送示範資料) 以產生股票行情資料範例。
4. 按照螢幕上的指示，確認資料正交付到 S3 儲存貯體。請注意，根據您在儲存貯體的緩衝設定，新物件可能需要幾分鐘才會出現於您的儲存貯體中。
5. 測試完成後，選擇 Stop sending demo data (停止傳送示範資料) 以停止產生使用費。

使用 Amazon Redshift 進行測試

使用下列程序，以 Amazon Redshift 做為目的地來測試 Firehose 串流。

使用 Amazon Redshift 測試 Firehose 串流

1. 您的 Firehose 串流預期資料表會出現在 Amazon Redshift 叢集中。[透過 SQL 介面連線至 Amazon Redshift](#)，並執行下列陳述式，建立接收範例資料的資料表。

```
create table firehose_test_table
(
  TICKER_SYMBOL varchar(4),
  SECTOR varchar(16),
  CHANGE float,
  PRICE float
);
```

2. 開啟位於 <https://console.aws.amazon.com/firehose/> 的 Firehose 主控台。
3. 選擇作用中的 Firehose 串流。Firehose 串流必須處於作用中狀態，才能開始傳送資料。
4. 編輯 Firehose 串流的目的地詳細資訊，以指向新建立的 firehose_test_table 資料表。
5. 在 Test with demo data (以示範資料測試) 底下，選擇 Start sending demo data (開始傳送示範資料) 以產生股票行情資料範例。
6. 按照螢幕上的指示，確認資料交付到表格中。請注意，根據緩衝設定，新的列可能需要幾分鐘才會出現於您的表格中。
7. 測試完成後，選擇 Stop sending demo data (停止傳送示範資料) 以停止產生使用費。
8. 編輯 Firehose 串流的目的地詳細資訊，以指向另一個資料表。
9. (選用) 刪除 firehose_test_table 表格。

使用 OpenSearch Service 進行測試

使用下列程序，使用 Amazon OpenSearch Service 做為目的地來測試 Firehose 串流。

使用 OpenSearch Service 測試 Firehose 串流

1. 開啟位於 <https://console.aws.amazon.com/firehose/> 的 Firehose 主控台。
2. 選擇作用中的 Firehose 串流。Firehose 串流必須處於作用中狀態，才能開始傳送資料。
3. 在 Test with demo data (以示範資料測試) 底下，選擇 Start sending demo data (開始傳送示範資料) 以產生股票行情資料範例。
4. 按照螢幕上的指示，確認資料交付到您的 OpenSearch Service 域。如需詳細資訊，請參閱《Amazon OpenSearch Service 開發人員指南》中的[在 OpenSearch Service 域中搜尋文件](#)。

5. 測試完成後，選擇 Stop sending demo data (停止傳送示範資料) 以停止產生使用費。

使用 Splunk 進行測試

使用下列程序，使用 Splunk 做為目的地來測試 Firehose 串流。

使用 Splunk 測試 Firehose 串流

1. 開啟位於 <https://console.aws.amazon.com/firehose/> 的 Firehose 主控台。
2. 選擇作用中的 Firehose 串流。Firehose 串流必須處於作用中狀態，才能開始傳送資料。
3. 在 Test with demo data (以示範資料測試) 底下，選擇 Start sending demo data (開始傳送示範資料) 以產生股票行情資料範例。
4. 確認資料交付到您的 Splunk 索引。Splunk 中的搜尋範例為 `sourcetype="aws:firehose:json"` 和 `index="name-of-your-splunk-index"`。如需進一步了解如何在 Splunk 中搜尋事件，請參閱 Splunk 文件中的 [Search Manual](#)。

如果測試資料未出現於您的 Splunk 索引，則請檢查您的 Amazon S3 儲存貯體是否有失敗事件。另請參閱[資料未傳送至 Splunk](#)。

5. 測試結束後，選擇 Stop sending demo data (停止傳送示範資料) 以停止產生使用費。

使用 Apache Iceberg 資料表進行測試

使用下列程序，以 Apache Iceberg Tables 做為目的地來測試 Firehose 串流。

使用 Apache Iceberg 資料表測試 Firehose 串流

1. 開啟位於 <https://console.aws.amazon.com/firehose/> 的 Firehose 主控台。
2. 選擇作用中的 Firehose 串流。Firehose 串流必須處於作用中狀態，才能開始傳送資料。
3. 在 Test with demo data (以示範資料測試) 底下，選擇 Start sending demo data (開始傳送示範資料) 以產生股票行情資料範例。
4. 遵循畫面上的指示，確認資料正在交付至您的 Apache Iceberg 資料表。請注意，根據新物件的緩衝組態，新物件可能需要幾分鐘才會出現在您的儲存貯體中。
5. 如果測試資料未出現在您的 Apache Iceberg 資料表中，請檢查您的 Amazon S3 儲存貯體是否有失敗的事件。
6. 測試結束後，選擇 Stop sending demo data (停止傳送示範資料) 以停止產生使用費。

將資料傳送至 Firehose 串流

本節說明如何使用不同的資料來源將資料傳送至 Firehose 串流。如果您是初次使用 Amazon Data Firehose，請花一些時間熟悉 [中介紹的概念和術語](#) [什麼是 Amazon Data Firehose?](#)。

Note

有些 AWS 服務只能傳送訊息和事件到位於相同區域的 Firehose 串流。當您為 Amazon CloudWatch Logs、CloudWatch Events 設定目標時，如果您的 Firehose 串流未顯示為選項 AWS IoT，或確認您的 Firehose 串流與其他服務位於相同的區域。如需每個區域服務端點的資訊，請參閱 [Amazon Data Firehose 端點](#)。

您可以從下列資料來源將資料傳送至 Firehose 串流。

主題

- [設定 Kinesis 代理程式以傳送資料](#)
- [使用 AWS SDK 傳送資料](#)
- [將 CloudWatch Logs 傳送至 Firehose](#)
- [將 CloudWatch 事件傳送至 Firehose](#)
- [設定 AWS IoT 將資料傳送至 Firehose](#)

設定 Kinesis 代理程式以傳送資料

Amazon Kinesis 代理程式是一種獨立的 Java 軟體應用程式，可做為參考實作，示範如何收集資料並將其傳送至 Firehose。代理程式會持續監控一組檔案，並將新資料傳送至 Firehose 串流。代理程式會顯示如何處理檔案輪換、檢查點和失敗時的重試。它顯示如何以可靠、及時且簡單的方式交付資料。它還顯示如何發出 CloudWatch 指標，以更好地監控和疑難排解串流程序。若要進一步了解，請參閱 [awslabs/amazon-kinesis-agent](#)。

根據預設，記錄會從各個檔案根據換行符號 ('\n') 字元進行剖析。不過，代理程式也可以設定為剖析多行記錄 (請參閱 [指定代理程式組態設定](#))。

您可以在以 Linux 為基礎的伺服器環境安裝代理程式，例如 Web 伺服器、日誌伺服器，及資料庫伺服器。安裝代理程式後，透過指定要監控的檔案和資料的 Firehose 串流來設定代理程式。設定代理程式之後，它會持久地從檔案收集資料，並可靠地將其傳送至 Firehose 串流。

先決條件

開始使用 Kinesis Agent 之前，請確定您符合下列先決條件。

- 您的作業系統必須是 Amazon Linux，或 Red Hat Enterprise Linux 版本 7 或更新版本。
- 2.0.0 版或更新版本代理程式執行時使用的是 JRE 1.8 版或更新版本。1.1.x 版代理程式執行時使用的是 JRE 1.7 或更新版本。
- 如果您使用 Amazon EC2 執行您的代理程式，則請啟動您的 EC2 執行個體。
- 您指定的 IAM 角色或 AWS 登入資料必須具有執行 Amazon Data Firehose [PutRecordBatch](#) 操作的許可，代理程式才能將資料傳送至 Firehose 串流。若您啟用 CloudWatch 監控代理程式，則另需具備執行 CloudWatch [PutMetricData](#) 操作的許可。如需詳細資訊，請參閱 [使用 Amazon Data Firehose 控制存取](#)、[監控 Kinesis 代理程式運作狀態](#) 和 [Amazon CloudWatch 身分驗證與存取控制](#)。

管理 AWS 登入資料

使用下列其中一種方法來管理您的 AWS 登入資料：

- 建立自訂登入資料提供者。如需詳細資訊，請參閱 [the section called “建立自訂登入資料提供者”](#)。
- 當您啟動 EC2 執行個體時，指定 IAM 角色。
- 設定代理程式時指定 AWS 登入資料（請參閱 下組態資料表 `awsSecretAccessKey` 中的 `awsAccessKeyId` 和 項目 [the section called “指定代理程式組態設定”](#)）。
- 編輯 `/etc/sysconfig/aws-kinesis-agent` 以指定您的 AWS 區域和 AWS 存取金鑰。
- 如果您的 EC2 執行個體位於不同的 AWS 帳戶中，請建立 IAM 角色以提供 Amazon Data Firehose 服務的存取權。並在設定代理程式時指定該角色（請參閱 [assumeRoleARN](#) 和 [assumeRoleExternalId](#)）。使用上述其中一種方法來指定其他帳戶中具有擔任此角色許可之使用者的 AWS 登入資料。

建立自訂登入資料提供者

您可以在下列組態設定中建立自訂憑證提供者，並將其類別名稱和 jar 路徑指定給 Kinesis 代理程式：`userDefinedCredentialsProvider.classname` 和 `userDefinedCredentialsProvider.location`。如需這兩個組態設定的說明，請參閱 [the section called “指定代理程式組態設定”](#)。

若要建立自訂憑證提供者，請定義實作 AWS `CredentialsProvider` 介面的類別，如下列範例所示。

```
import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.AWSCredentialsProvider;
import com.amazonaws.auth.BasicAWSCredentials;

public class YourClassName implements AWSCredentialsProvider {
    public YourClassName() {
    }

    public AWSCredentials getCredentials() {
        return new BasicAWSCredentials("key1", "key2");
    }

    public void refresh() {
    }
}
```

您的類別必須有一個不帶引數的建構函數。

AWS 會定期叫用重新整理方法，以取得更新的登入資料。如果您希望憑證提供者在其整個生命週期內提供不同的憑證，則請在此方法中包含重新整理憑證的程式碼。或者，如果您希望憑證提供者提供靜態(不變更)憑證，則可以將此方法保留為空。

下載並安裝 代理程式

首先，連接至您的執行個體。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的[連線至您的執行個體](#)。如果您在連線時遇到問題，請參閱《Amazon EC2 使用者指南》中的[連線至執行個體的故障診斷](#)。

然後，使用以下方法之一安裝代理程式。

- 從 Amazon Linux 儲存器安裝代理程式

此方法只適用於 Amazon Linux 執行個體。使用下列命令：

```
sudo yum install -y aws-kinesis-agent
```

2.0.0 版或更新版本代理程式安裝在具有 Amazon Linux 2 (AL2) 作業系統的電腦上。此代理程式版本需要 Java 1.8 或更新版本。如果所需的 Java 版本尚不存在，則代理程式安裝程序會安裝該版本。如需 Amazon Linux 2 的詳細資訊，請參閱 <https://aws.amazon.com/amazon-linux-2/>。

- 從 Amazon S3 儲存器安裝代理程式

此方法適用於 Red Hat Enterprise Linux 以及 Amazon Linux 2 執行個體，因為其會從公開的儲存器安裝代理程式。使用下列命令來下載並安裝最新版的代理程式版本 2.x.x：

```
sudo yum install -y https://s3.amazonaws.com/streaming-data-agent/aws-kinesis-agent-latest.amzn2.noarch.rpm
```

若要安裝特定版本的代理程式，請在命令中指定版本號碼。例如，下列命令會安裝代理程式 v 2.0.1。

```
sudo yum install -y https://streaming-data-agent.s3.amazonaws.com/aws-kinesis-agent-2.0.1-1.amzn1.noarch.rpm
```

如果您有 Java 1.7 且不想將其升級，則可下載與 Java 1.7 相容的 1.x.x 版代理程式。例如，若要下載 1.1.6 版代理程式，您可以使用以下命令：

```
sudo yum install -y https://s3.amazonaws.com/streaming-data-agent/aws-kinesis-agent-1.1.6-1.amzn1.noarch.rpm
```

您可以使用下列命令下載最新的代理程式

```
sudo yum install -y https://s3.amazonaws.com/streaming-data-agent/aws-kinesis-agent-latest.amzn2.noarch.rpm
```

- 從 GitHub 儲存器安裝代理程式

1. 首先，請確定您已安裝所需的 Java 版本，具體取決於代理程式版本。
2. 從 [awslabs/amazon-kinesis-agent](https://github.com/aws-labs/amazon-kinesis-agent) GitHub 儲存器下載代理程式。

3. 瀏覽到下載目錄並執行下列命令以安裝代理程式：

```
sudo ./setup --install
```

- 若要在 Docker 容器中設定代理程式

Kinesis 代理程式也可以透過 [amazonlinux](#) 容器基礎在容器中執行。使用以下 Dockerfile，然後執行 `docker build`。

```
FROM amazonlinux

RUN yum install -y aws-kinesis-agent which findutils
COPY agent.json /etc/aws-kinesis/agent.json

CMD ["start-aws-kinesis-agent"]
```

設定和啟動 代理程式

設定和啟動代理程式

1. 開啟並編輯組態檔案 (如果使用預設檔案存取許可，即以超級使用者身分執行)：`/etc/aws-kinesis/agent.json`

在此組態檔案中，指定代理程式收集資料的檔案 `"filePattern"` ()，以及代理程式傳送資料的 Firehose 串流名稱 `"deliveryStream"` ()。檔案名稱具備一種模式，可讓代理程式辨識檔案輪換。您可以輪換檔案或建立新的檔案，每秒不超過一次。代理程式會使用檔案建立時間戳記，來判斷要追蹤哪些檔案並結尾至 Firehose 串流。如果建立新檔案或輪換檔案的頻率超過每秒一次，將導致代理程式無法正確區分這些檔案。

```
{
  "flows": [
    {
      "filePattern": "/tmp/app.log*",
      "deliveryStream": "yourdeliverystream"
    }
  ]
}
```

預設 AWS 區域為 us-east-1。如果您使用不同的區域，請將 `firehose.endpoint` 設定新增至組態檔案，藉此指定您區域的端點。如需詳細資訊，請參閱[指定代理程式組態設定](#)。

2. 手動啟動代理程式：

```
sudo service aws-kinesis-agent start
```

3. (選用) 設定代理程式在系統啟動時開始執行：

```
sudo chkconfig aws-kinesis-agent on
```

代理程式現在已做為系統服務在背景執行。它會持續監控指定的檔案，並將資料傳送至指定的 Firehose 串流。代理程式的活動記錄於 `/var/log/aws-kinesis-agent/aws-kinesis-agent.log`。

指定代理程式組態設定

代理程式支援兩種必要的組態設定 `filePattern` 和 `deliveryStream`，以及用於其他功能的選用組態設定。您可以在 `/etc/aws-kinesis/agent.json` 指定必要及選用的組態設定。

當您變更組態檔案時，必須使用下列命令停止及啟動代理程式：

```
sudo service aws-kinesis-agent stop
sudo service aws-kinesis-agent start
```

或者，您可以使用下列命令：

```
sudo service aws-kinesis-agent restart
```

以下是一般組態設定。

組態設定	Description
<code>assumeRoleARN</code>	使用者欲擔任角色的 Amazon Resource Name (ARN)。如需詳細資訊，請參閱 《IAM 使用者指南》 中的使用 IAM 角色跨 AWS 帳戶委派存取權。

組態設定	Description
<code>assumeRoleExternalId</code>	選用的識別符決定誰可以擔任此角色。如需詳細資訊，請參閱《IAM 使用者指南》中的 如何使用外部 ID 。
<code>awsAccessKeyId</code>	AWS 會覆寫預設登入資料的存取金鑰 ID。此設定優先於所有其他登入資料供應商。
<code>awsSecretAccessKey</code>	AWS 覆寫預設登入資料的私密金鑰。此設定優先於所有其他登入資料供應商。
<code>cloudwatch.emitMetrics</code>	如設定為 (true)，將啟用代理程式發出指標至 CloudWatch。 預設：true
<code>cloudwatch.endpoint</code>	適用於 CloudWatch 的區域端點。 預設：monitoring.us-east-1.amazonaws.com
<code>firehose.endpoint</code>	Amazon Data Firehose 的區域端點。 預設：firehose.us-east-1.amazonaws.com
<code>sts.endpoint</code>	Security AWS Token Service 的區域端點。 預設：https://sts.amazonaws.com
<code>userDefinedCredentialsProvider.classname</code>	如果您定義自訂登入資料提供者，請使用此設定提供其完整類別名稱。不要在類別名稱的末尾包含 <code>.class</code> 。
<code>userDefinedCredentialsProvider.location</code>	如果您定義自訂登入資料提供者，請使用此設定來指定包含自訂登入資料提供者之 jar 的絕對路徑。代理程式也會在下列位置尋找 jar 檔案： <code>/usr/share/aws-kinesis-agent/lib/</code> 。

以下是流程組態設定。

組態設定	Description
aggregateRecordSizeBytes	<p>若要建立代理程式彙總記錄，然後將其放入 Firehose 串流，請指定此設定。將其設定為您希望彙總記錄在代理程式將其放入 Firehose 串流之前擁有的大小。</p> <p>預設值：0 (無彙總)</p>
dataProcessingOptions	<p>在傳送至 Firehose 串流之前，套用至每個剖析記錄的處理選項清單。此處理選項會在指定的資料夾執行。如需詳細資訊，請參閱使用代理程式預先處理資料。</p>
deliveryStream	<p>【必要】 Firehose 串流的名稱。</p>
filePattern	<p>[必要] 需要代理程式監控的檔案部分。符合此模式的任何檔案將由代理程式自動挑選及監控。對於符合此模式的所有檔案，請將讀取許可授與 <code>aws-kinesis-agent-user</code>。對於包含檔案的目錄，請將讀取和執行許可授與 <code>aws-kinesis-agent-user</code>。</p> <div style="border: 1px solid #f00; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p>⚠ Important</p> <p>代理程式會挑選符合此模式的任何檔案。若要確保代理程式不會挑選意外的記錄，請小心選擇此模式。</p> </div>
initialPosition	<p>檔案開始進行剖析的初始位置。有效值為 <code>START_OF_FILE</code> 和 <code>END_OF_FILE</code>。</p> <p>預設：END_OF_FILE</p>
maxBufferAgeMillis	<p>代理程式在將資料傳送到 Firehose 串流之前緩衝資料的最長時間，以毫秒為單位。</p> <p>數值範圍：1,000–900,000 (1 秒到 15 分鐘)</p> <p>預設：60,000 (1 分鐘)</p>
maxBufferSizeBytes	<p>代理程式在將資料傳送到 Firehose 串流之前緩衝資料的大小上限，以位元組為單位。</p>

組態設定	Description
	數值範圍：1–4,194,304 (4 MB) 預設：4,194,304 (4 MB)
maxBuffer SizeRecords	代理程式在將資料傳送到 Firehose 串流之前緩衝資料的記錄數目上限。 數值範圍：1–500 預設：500
minTimeBe tweenFile PollsMillis	代理程式輪詢和剖析檔案以找出新資料的時間間隔 (以毫秒為單位)。 數值範圍：1 或以上 預設：100
multiLine StartPattern	用於識別記錄開始處的模式。記錄是由符合模式的一列及不符合模式的任何幾列所組成。有效值為常規運算式。根據預設，每個新日誌檔中的新列會剖析為一筆記錄。
skipHeaderLines	代理程式剖析監控檔案開頭部分時略過的列數。 數值範圍：0 或以上 預設：0 (零)
truncated RecordTer minator	當記錄大小超過 Amazon Data Firehose 記錄大小限制時，代理程式用來截斷已剖析記錄的字串。(1,000 KB) 預設：'\n' (換行符號)

設定多個檔案目錄和串流

透過指定多個流程組態設定，您可以設定代理程式來監控多個檔案目錄，然後將資料傳送到多個串流。在下列組態範例中，代理程式會監控兩個檔案目錄，並分別將資料傳送至 Kinesis 資料串流和 Firehose 串流。您可以為 Kinesis Data Streams 和 Amazon Data Firehose 指定不同的端點，以便您的資料串流和 Firehose 串流不需要位於相同的區域。

```
{
```

```
"cloudwatch.emitMetrics": true,  
"kinesis.endpoint": "https://your/kinesis/endpoint",  
"firehose.endpoint": "https://your/firehose/endpoint",  
"flows": [  
  {  
    "filePattern": "/tmp/app1.log*",  
    "kinesisStream": "yourkinesisstream"  
  },  
  {  
    "filePattern": "/tmp/app2.log*",  
    "deliveryStream": "yourfirehosedeliverystream"  
  }  
]
```

如需將代理程式與 Amazon Kinesis Data Streams 搭配使用的詳細資訊，請參閱[使用 Kinesis 代理程式寫入 Amazon Kinesis Data Streams](#)。

使用 代理程式預先處理資料

代理程式可以預先處理從受監控檔案剖析的記錄，然後再將其傳送到 Firehose 串流。您可以將 `dataProcessingOptions` 組態設定新增到您的檔案流程以啟用此功能。可新增一個或多個處理選項，這些選項將依照指定的順序執行。

代理程式支援以下處理選項。由於代理程式是開放原始碼，因此您可進一步開發和擴展其處理選項。您可以從 [Kinesis 代理程式](#) 下載代理程式。

處理選項

SINGLELINE

移除換行字元、前方空格及結尾空格，藉此將多列記錄轉換為單列記錄。

```
{  
  "optionName": "SINGLELINE"  
}
```

CSVTOJSON

將記錄從分隔符號區隔格式轉換為 JSON 格式。

```
{
```

```

    "optionName": "CSVTOJSON",
    "customFieldNames": [ "field1", "field2", ... ],
    "delimiter": "yourdelimiter"
}

```

customFieldNames

[必要] 欄位名稱在每個 JSON 鍵值對中做為鍵。例如，如果您指定 ["f1", "f2"]，記錄「v1、v2」將轉換為 {"f1":"v1","f2":"v2"}。

delimiter

在記錄做為分隔符號的字串。預設為逗號 (,)。

LOGTOJSON

將記錄從日誌格式轉換為 JSON 格式。支援的日誌格式為 Apache Common Log、Apache Combined Log、Apache Error Log、以及 RFC3164 Syslog。

```

{
  "optionName": "LOGTOJSON",
  "logFormat": "logformat",
  "matchPattern": "yourregexpattern",
  "customFieldNames": [ "field1", "field2", ... ]
}

```

logFormat

[必要] 日誌項目格式。以下是可能的值：

- COMMONAPACHELOG – Apache Common Log 格式。根據預設，每個日誌項目皆有以下模式：「%{host} %{ident} %{authuser} [%{datetime}] \"%{request}\" %{response} %{bytes}」。
- COMBINEDAPACHELOG – Apache Combined Log 格式。根據預設，每個日誌項目皆有以下模式：「%{host} %{ident} %{authuser} [%{datetime}] \"%{request}\" %{response} %{bytes} %{referrer} %{agent}」。
- APACHEERRORLOG – Apache Error Log 格式。根據預設，每個日誌項目皆有以下模式：「[%{timestamp}] [%{module}:%{severity}] [pid %{processid}:tid %{threadid}] [client: %{client}] %{message}」。
- SYSLOG – RFC3164 Syslog 格式。根據預設，每個日誌項目皆有以下模式：「%{timestamp} %{hostname} %{program}[%{processid}]: %{message}」。

matchPattern

覆寫指定日誌格式的預設模式。如果日誌項目使用自訂格式，則使用此設定從日誌項目擷取值。若您指定 `matchPattern`，您也必須指定 `customFieldNames`。

customFieldNames

自訂欄位名稱在每個 JSON 鍵值對中做為鍵。您可以使用此設定來定義從 `matchPattern` 擷取值的欄位名稱，或覆寫預先定義的日誌格式的預設欄位名稱。

Example : LOGTOJSON 組態

這裡提供一個 Apache Common Log 項目轉換為 JSON 格式的 LOGTOJSON 組態範例：

```
{
  "optionName": "LOGTOJSON",
  "logFormat": "COMMONAPACHELOG"
}
```

轉換前：

```
64.242.88.10 - - [07/Mar/2004:16:10:02 -0800] "GET /mailman/listinfo/hsdivision
HTTP/1.1" 200 6291
```

轉換後：

```
{"host":"64.242.88.10","ident":null,"authuser":null,"datetime":"07/
Mar/2004:16:10:02 -0800","request":"GET /mailman/listinfo/hsdivision
HTTP/1.1","response":"200","bytes":"6291"}
```

Example : 使用自訂欄位的 LOGTOJSON 組態

以下是另一個 LOGTOJSON 組態範例：

```
{
  "optionName": "LOGTOJSON",
  "logFormat": "COMMONAPACHELOG",
  "customFieldNames": ["f1", "f2", "f3", "f4", "f5", "f6", "f7"]
}
```

使用此組態設定，前一個範例的相同 Apache Common Log 項目轉換為 JSON 格式如下：

```
{"f1":"64.242.88.10","f2":null,"f3":null,"f4":"07/Mar/2004:16:10:02 -0800","f5":"GET /mailman/listinfo/hsdivision HTTP/1.1","f6":"200","f7":"6291"}
```

Example：轉換 Apache Common Log 項目

下列流程組態將 Apache Common Log 項目轉換為 JSON 格式的單列記錄：

```
{
  "flows": [
    {
      "filePattern": "/tmp/app.log*",
      "deliveryStream": "my-delivery-stream",
      "dataProcessingOptions": [
        {
          "optionName": "LOGTOJSON",
          "logFormat": "COMMONAPACHELOG"
        }
      ]
    }
  ]
}
```

Example：轉換多列記錄

以下流程組態剖析第一行從「[SEQUENCE=」開始的多列記錄。每筆記錄都會先轉換為單列記錄。然後，根據定位鍵分隔符號從記錄中擷取值。擷取的值會對應到指定的 `customFieldNames` 值以形成 JSON 格式的單列記錄。

```
{
  "flows": [
    {
      "filePattern": "/tmp/app.log*",
      "deliveryStream": "my-delivery-stream",
      "multilineStartPattern": "\\[[SEQUENCE=",
      "dataProcessingOptions": [
        {
          "optionName": "SINGLELINE"
        },
        {
          "optionName": "CSVTOJSON",

```

```

        "customFieldNames": [ "field1", "field2", "field3" ],
        "delimiter": "\\t"
    }
}
]
}

```

Example：使用匹配模式的 LOGTOJSON 組態

以下是 Apache Common Log 項目轉換為 JSON 格式的 LOGTOJSON 組態範例，省略最後欄位 (位元組)：

```

{
  "optionName": "LOGTOJSON",
  "logFormat": "COMMONAPACHELOG",
  "matchPattern": "^(\\d\\.\\d\\.\\d) (\\S+) (\\S+) \\[[([\\w:/]+\\s[+\\-]\\d{4})\\] \\\"(.+?)\\\" (\\d{3})",
  "customFieldNames": ["host", "ident", "authuser", "datetime", "request", "response"]
}

```

轉換前：

```
123.45.67.89 - - [27/Oct/2000:09:27:09 -0400] "GET /java/javaResources.html HTTP/1.0"
200
```

轉換後：

```
{"host":"123.45.67.89","ident":null,"authuser":null,"datetime":"27/Oct/2000:09:27:09
-0400","request":"GET /java/javaResources.html HTTP/1.0","response":"200"}
```

使用常見的 Agent CLI 命令

下表提供一組常用案例和對應的命令，用於使用 AWS Kinesis 代理程式。

使用案例	命令
在系統啟動時自動啟動代理程式	<code>sudo chkconfig aws-kinesis-agent on</code>

使用案例	命令
檢查客服人員的狀態	<code>sudo service aws-kinesis-agent status</code>
停止代理程式	<code>sudo service aws-kinesis-agent stop</code>
從此位置讀取代理程式的日誌檔案	<code>/var/log/aws-kinesis-agent/aws-kinesis-agent.log</code>
解除安裝代理程式	<code>sudo yum remove aws-kinesis-agent</code>

從 Kinesis Agent 傳送時的問題故障診斷

此表格針對使用 Amazon Kinesis Agent 時遇到的常見問題，提供疑難排解資訊和解決方案。

問題	解決方案
為什麼 Kinesis 代理程式無法在 Windows 上運作？	適用於 Windows 的 Kinesis 代理程式 是不同於適用於 Linux 平台的 Kinesis 代理程式的軟體。
為什麼 Kinesis 代理程式會減速和/或 RecordSendErrors 增加？	這通常是由於來自 Kinesis 的限流。檢查 Kinesis Data Streams 的 WriteProvisionedThroughputExceeded 指標或 Firehose 串流的 ThrottledRecords 指標。這些指標中從 0 開始的任何增量，均表示需要提升串流限制。如需詳細資訊，請參閱 Kinesis Data Stream 限制 和 Firehose 串流 。 排除限流之後，請查看 Kinesis 代理程式是否設定為追蹤大量小型檔案。Kinesis 代理程式追蹤新檔案時會有延遲，因此 Kinesis 代理程式應追蹤少量較大的檔案。嘗試將日誌檔案合併至較大的檔案中。
如何解決 java.lang.OutOfMemoryError 例外狀況？	當 Kinesis Agent 沒有足夠的記憶體來處理其目前的工作負載時，就會發生這種情況。嘗試增加 /usr/bin/start-

問題	解決方案
	<code>aws-kinesis-agent</code> 中的 <code>JAVA_START_HEAP</code> 和 <code>JAVA_MAX_HEAP</code> 並重新啟動代理程式。
如何解決 <code>IllegalStateException : connection pool shut down</code> 例外狀況？	Kinesis 代理程式沒有足夠的連線可以處理其目前的工作負載。嘗試在位於 <code>/etc/aws-kinesis/agent.json</code> 的一般代理程式組態設定中增加 <code>maxConnections</code> 和 <code>maxSendingThreads</code> 。這些欄位的預設值是可用執行期處理器的 12 倍。如需進階代理程式組態設定的詳細資訊，請參閱 AgentConfiguration.java 。
如何使用 Kinesis 代理程式對另一個問題進行偵錯？	可以在 <code>/etc/aws-kinesis/log4j.xml</code> 中啟用 <code>DEBUG</code> 層級日誌。
我應該如何對 Kinesis Agent 進行設定？	<code>maxBufferSizeBytes</code> 越小，Kinesis 代理程式傳送資料的頻率就越高。這可能很好，因為這樣會減少記錄的交付時間，但也增加了 Kinesis 的每秒請求。
為什麼 Kinesis 代理程式傳送重複的日誌？	發生這種情況是由於檔案追蹤組態錯誤。請確保每個 <code>fileFlow's filePattern</code> 僅與一個檔案相符。如果正在 <code>copytruncate</code> 模式中使用 <code>logrotate</code> 模式下，也可能發生這種情況。嘗試將模式變更為預設模式，或建立模式以避免重複。如需有關處理重複記錄的詳細資訊，請參閱 處理重複記錄 。

使用 AWS SDK 傳送資料

您可以使用 [Amazon Data Firehose API](#)，使用 [AWS 適用於 Java](#)、[.NET](#)、[Node.js](#)、[Python](#) 或 [Ruby](#) 的 SDK 將資料傳送至 Firehose 串流。如果您是初次使用 Amazon Data Firehose，請花一些時間熟悉中介紹的概念和術語 [什麼是 Amazon Data Firehose ?](#)。如需詳細資訊，請參閱 [Amazon Web Services 開發功能入門](#)。

這些範例不代表可立即生產的程式碼，無法檢查出所有可能的例外狀況，也不可視為任何潛在安全或效能疑慮的原因。

Amazon Data Firehose API 提供兩個將資料傳送至 Firehose 串流的操作：[PutRecord](#) 和 [PutRecordBatch](#)。會在一次呼叫中 `PutRecord()` 傳送一個資料記錄，而且 `PutRecordBatch()` 可以在一次呼叫中傳送多個資料記錄。

使用 PutRecord 的單一寫入操作

放置資料只需要 Firehose 串流名稱和位元組緩衝區 (≤ 1000 KB)。由於 Amazon Data Firehose 會在將檔案載入 Amazon S3 之前批次處理多個記錄，因此您可能想要新增記錄分隔符號。若要將資料一次一個記錄放入 Firehose 串流，請使用下列程式碼：

```
PutRecordRequest putRecordRequest = new PutRecordRequest();
putRecordRequest.setDeliveryStreamName(deliveryStreamName);

String data = line + "\n";

Record record = new Record().withData(ByteBuffer.wrap(data.getBytes()));
putRecordRequest.setRecord(record);

// Put record into the DeliveryStream
firehoseClient.putRecord(putRecordRequest);
```

如需更多程式碼內容，請參閱 AWS SDK 中包含的範例程式碼。如需有關請求和回應語法的資訊，請參閱 [Firehose API Operations](#) 中的相關主題。

使用 PutRecordBatch 的批次寫入操作

放置資料只需要 Firehose 串流名稱和記錄清單。由於 Amazon Data Firehose 會在將檔案載入 Amazon S3 之前批次處理多個記錄，因此您可能想要新增記錄分隔符號。若要將批次資料記錄放入 Firehose 串流，請使用下列程式碼：

```
PutRecordBatchRequest putRecordBatchRequest = new PutRecordBatchRequest();
putRecordBatchRequest.setDeliveryStreamName(deliveryStreamName);
putRecordBatchRequest.setRecords(recordList);

// Put Record Batch records. Max No.Of Records we can put in a
// single put record batch request is 500
firehoseClient.putRecordBatch(putRecordBatchRequest);

recordList.clear();
```

如需更多程式碼內容，請參閱 AWS SDK 中包含的範例程式碼。如需請求和回應語法的相關資訊，請參閱 [Firehose API Operations](#) 中的相關主題。

將 CloudWatch Logs 傳送至 Firehose

CloudWatch Logs 事件可以使用 CloudWatch 訂閱篩選條件傳送至 Firehose。如需詳細資訊，請參閱 [使用 Amazon Data Firehose 訂閱篩選條件](#)。

CloudWatch Logs 事件會以壓縮的 gzip 格式傳送至 Firehose。如果您想要將解壓縮的日誌事件交付至 Firehose 目的地，您可以使用 Firehose 中的解壓縮功能自動解壓縮 CloudWatch Logs。

Important

目前，Firehose 不支援將 CloudWatch Logs 交付至 Amazon OpenSearch Service 目的地，因為 Amazon CloudWatch 將多個日誌事件合併為一個 Firehose 記錄，Amazon OpenSearch Service 無法接受一個記錄中的多個日誌事件。作為替代方案，您可以考慮在 [CloudWatch Logs 中使用 Amazon OpenSearch Service 的訂閱篩選條件](#)。

解壓縮 CloudWatch Logs

如果您使用 Firehose 交付 CloudWatch Logs 並想要將解壓縮的資料交付至 Firehose 串流目的地，請使用 Firehose [Data Format Conversion](#) (Parquet、ORC) 或 [動態分割](#)。您必須為 Firehose 串流啟用解壓縮。

您可以使用 AWS 管理主控台 AWS Command Line Interface 或 AWS SDKs 啟用解壓縮。

Note

如果您在串流上啟用解壓縮功能，請僅將該串流用於 CloudWatch Logs 訂閱篩選條件，而不是用於已取代的日誌。如果您在用來同時擷取 CloudWatch Logs 和 Vended Logs 的串流上啟用解壓縮功能，則 Vended Logs 擷取至 Firehose 失敗。此解壓縮功能僅適用於 CloudWatch Logs。

解壓縮 CloudWatch Logs 後擷取訊息

當您啟用解壓縮時，您也可以選擇啟用訊息擷取。使用訊息擷取時，Firehose 會從解壓縮的 CloudWatch Logs 記錄中篩選掉所有中繼資料，例如擁有者、日誌群組、日誌串流和其他中繼資料，

並僅傳遞訊息欄位中的內容。如果您要將資料交付至 Splunk 目的地，您必須開啟訊息擷取，Splunk 才能剖析資料。以下是使用和不使用訊息擷取解壓縮後的範例輸出。

圖 1：解壓縮後未擷取訊息的範例輸出：

```
{
  "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\":\"Root1\"}"
    },
    {
      "id": "31953106606966983378809025079804211143289615424298221569",
      "timestamp": 1432826855000,
      "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root2\"}"
    },
    {
      "id": "31953106606966983378809025079804211143289615424298221570",
      "timestamp": 1432826855000,
      "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root3\"}"
    }
  ]
}
```

圖 2：使用訊息擷取解壓縮後的範例輸出：

```
{"eventVersion":"1.03","userIdentity":{"type":"Root1"}
{"eventVersion":"1.03","userIdentity":{"type":"Root2"}
{"eventVersion":"1.03","userIdentity":{"type":"Root3"}
```

從主控台對新的 Firehose 串流啟用解壓縮

使用 在新的 Firehose 串流上啟用解壓縮 AWS 管理主控台

1. 登入 AWS 管理主控台 並開啟位於 <https://console.aws.amazon.com/kinesis> 的 Kinesis 主控台。
2. 在導覽窗格中選擇 Amazon Data Firehose。
3. 選擇建立 Firehose 串流。
4. 在選擇來源和目的地下

來源

Firehose 串流的來源。選擇下列其中一個來源：

- 直接 PUT – 選擇此選項可建立生產者應用程式直接寫入的 Firehose 串流。如需與 Firehose 中的 Direct PUT 整合的服務和代理程式和開放原始碼服務清單 AWS，請參閱[本節](#)。
- Kinesis 串流：選擇此選項可設定使用 Kinesis 資料串流做為資料來源的 Firehose 串流。然後，您可以使用 Firehose 輕鬆從現有的 Kinesis 資料串流讀取資料，並將其載入目的地。如需詳細資訊，請參閱[使用 Kinesis Data Streams 寫入 Firehose](#)

目的地

Firehose 串流的目的地。選擇下列其中一項：

- Amazon S3
 - Splunk
5. 在 Firehose 串流名稱下，輸入串流的名稱。
 6. (選用) 在轉換記錄下：
 - 在從 Amazon CloudWatch Logs 解壓縮來源記錄區段中，選擇開啟解壓縮。
 - 如果您想要在解壓縮後使用訊息擷取，請選擇開啟訊息擷取。

在現有的 Firehose 串流上啟用解壓縮

本節提供在現有 Firehose 串流上啟用解壓縮的指示。它涵蓋兩個案例：已停用 Lambda 處理的串流，以及已啟用 Lambda 處理的串流。以下各節概述每個案例的 step-by-step 程序，包括 Lambda 函數的建立或修改、更新 Firehose 設定，以及監控 CloudWatch 指標，以確保成功實作內建的 Firehose 解壓縮功能。

停用 Lambda 處理時啟用解壓縮

若要在停用 Lambda 處理的現有 Firehose 串流上啟用解壓縮，您必須先啟用 Lambda 處理。此條件僅適用於現有的串流。下列步驟顯示如何在未啟用 Lambda 處理的現有串流上啟用解壓縮。

1. 建立 Lambda 函式。您可以建立傳遞的虛擬記錄，也可以使用此[藍圖](#)建立新的 Lambda 函數。
2. 更新您目前的 Firehose 串流以啟用 Lambda 處理，並使用您建立用於處理的 Lambda 函數。
3. 使用新的 Lambda 函數更新串流後，請返回 Firehose 主控台並啟用解壓縮。
4. 停用您在步驟 1 中啟用的 Lambda 處理。您現在可以刪除您在步驟 1 中建立的函數。

啟用 Lambda 處理時啟用解壓縮

如果您已有具有 Lambda 函數的 Firehose 串流，若要執行解壓縮，您可以將其取代為 Firehose 解壓縮功能。在繼續之前，請檢閱您的 Lambda 函數程式碼，以確認其僅執行解壓縮或訊息擷取。Lambda 函數的輸出看起來應該類似於[圖 1 或圖 2](#)所示的範例。如果輸出看起來類似，您可以使用下列步驟取代 Lambda 函數。

1. 使用此[藍圖](#)取代您目前的 Lambda 函數。新的藍圖 Lambda 函數會自動偵測傳入的資料是壓縮還是解壓縮。只有在壓縮其輸入資料時，才會執行解壓縮。
2. 使用內建的 Firehose 選項開啟解壓縮以解壓縮。
3. 如果尚未啟用，請為您的 Firehose 串流啟用 CloudWatch 指標。監控指標 CloudWatchProcessorLambda_IncomingCompressedData 並等到此指標變更為零。這會確認傳送至 Lambda 函數的所有輸入資料都已解壓縮，而且不再需要 Lambda 函數。
4. 移除 Lambda 資料轉換，因為您不再需要它來解壓縮串流。

在 Firehose 串流上停用解壓縮

使用 在資料串流上停用解壓縮 AWS 管理主控台

1. 登入 AWS 管理主控台 並開啟位於 <https://console.aws.amazon.com/kinesis> 的 Kinesis 主控台。
2. 在導覽窗格中選擇 Amazon Data Firehose。
3. 選擇您要編輯的 Firehose 串流。
4. 在 Firehose 串流詳細資訊頁面上，選擇組態索引標籤。
5. 在轉換和轉換記錄區段中，選擇編輯。

6. 在從 Amazon CloudWatch Logs 解壓縮來源記錄下，清除開啟解壓縮，然後選擇儲存變更。

針對 Firehose 中的解壓縮進行故障診斷

下表顯示 Firehose 在資料解壓縮和處理期間如何處理錯誤，包括將記錄交付至錯誤 S3 儲存貯體、記錄錯誤和發出指標。它還說明針對未經授權的資料放置操作傳回的錯誤訊息。

問題	解決方案
如果在解壓縮期間發生錯誤，來源資料會發生什麼情況？	如果 Amazon Data Firehose 無法解壓縮記錄，記錄會依原狀（壓縮格式）交付，以便在 Firehose 串流建立時間期間指定的 S3 儲存貯體發生錯誤。除了記錄之外，交付的物件也包含錯誤碼和錯誤訊息，而且這些物件會交付到名為的 S3 儲存貯體字首decompression-failed。Firehose 會在記錄解壓縮失敗後繼續處理其他記錄。
如果成功解壓縮後處理管道發生錯誤，來源資料會發生什麼情況？	如果在解壓縮如動態分割和資料格式轉換等處理步驟中，Amazon Data Firehose 發生錯誤，記錄會以壓縮格式交付到您 Firehose 串流建立時間期間指定的錯誤 S3 儲存貯體。除了記錄之外，交付的物件也包含錯誤碼和錯誤訊息。
如果發生錯誤或例外狀況，您會如何收到通知？	如果在解壓縮期間發生錯誤或例外狀況，如果您設定 CloudWatch Logs，Firehose 會將錯誤訊息記錄到 CloudWatch Logs。此外，Firehose 會將指標傳送至您可以監控的 CloudWatch 指標。您也可以根據 Firehose 發出的指標選擇性地建立警示。
當put操作不是來自 CloudWatch Logs 時會發生什麼情況？	當客戶puts不是來自 CloudWatch Logs 時，會傳回下列錯誤訊息： <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;"> <pre>Put to Firehose failed for AccountId: <accountID>, FirehoseName: <firehosename> because the request is not originating from allowed source types.</pre> </div>
Firehose 針對解壓縮功能發出哪些指標？	Firehose 會發出每個記錄的解壓縮指標。您應該選取期間 (1 分鐘)、統計資料 (總和)、日期範圍，以取

問題	解決方案
	得DecompressedRecords 失敗、成功或DecompressedBytes 失敗或成功的數量。如需詳細資訊，請參閱 CloudWatch Logs 解壓縮指標 。

將 CloudWatch 事件傳送至 Firehose

您可以將目標新增至 Amazon CloudWatch CloudWatch 將事件傳送至 Firehose 串流。

為將事件傳送至現有 Firehose 串流的 CloudWatch Events 規則建立目標

1. 登入 AWS 管理主控台，並在 <https://console.aws.amazon.com/cloudwatch/> 開啟 CloudWatch 主控台。
2. 選擇建立規則。
3. 在步驟 1：建立規則頁面上，針對目標選擇新增目標，然後選擇 Firehose 串流。
4. 選擇現有的 Firehose 串流。

如需建立 CloudWatch Events 規則的詳細資訊，請參閱 [Amazon CloudWatch Events 入門](#)。

設定 AWS IoT 將資料傳送至 Firehose

您可以藉由新增 動作 AWS IoT，將設定為將資訊傳送至 Firehose 串流。

建立將事件傳送至現有 Firehose 串流的動作

1. 在 AWS IoT 主控台中建立規則時，請在建立規則頁面的設定一或多個動作下，選擇新增動作。
2. 選擇將訊息傳送至 Amazon Kinesis Firehose 串流。
3. 選擇 Configure action (設定動作)。
4. 針對串流名稱，選擇現有的 Firehose 串流。
5. 在 Separator (分隔符號) 的部分，選擇欲插入記錄間的分隔符號字元。
6. 對於 IAM 角色名稱，選擇現有 IAM 角色或選擇建立新角色。
7. 選擇新增動作。

如需建立 AWS IoT 規則的詳細資訊，請參閱 [AWS IoT 規則教學課程](#)。

轉換 Amazon Data Firehose 中的來源資料

Amazon Data Firehose 可以調用 Lambda 函數來轉換傳入的來源資料，並將轉換的資料傳遞至目的地。您可以在建立 Firehose 串流時啟用 Amazon Data Firehose 資料轉換。

了解資料轉換流程

當您啟用 Firehose 資料轉換時，Firehose 會緩衝傳入的資料。緩衝大小提示範圍介於 0.2 MB 到 3MB 之間。除了 Splunk 和 Snowflake 之外，所有目的地的預設 Lambda 緩衝大小提示皆為 1 MB。對於 Splunk 和 Snowflake，預設緩衝提示為 256 KB。Lambda 緩衝間隔提示範圍介於 0 到 900 秒之間。除了 Snowflake 之外，所有目的地的預設 Lambda 緩衝間隔提示都是 60 秒。對於 Snowflake，預設緩衝提示間隔為 30 秒。若要調整緩衝大小，請使用名為 `BufferSizeInMBs` 和的 [ProcessorParameter](#)，設定 [CreateDeliveryStream](#) 或 [UpdateDestination](#) API 的 [ProcessingConfiguration](#) 參數 `IntervalInSeconds`。接著 Firehose 會使用同步調用模式，與每個緩衝批次同步調用指定的 Lambda AWS Lambda 函數。轉換後的資料會從 Lambda 傳送至 Firehose。然後，Firehose 會在達到指定的目的地緩衝大小或緩衝間隔時將其傳送至目的地，以先發生者為準。

Important

在請求和回應方面，Lambda 同步調用模式的承載大小上限為 6 MB。因此，用來傳送請求至函式的緩衝區大小必須小於或等於 6 MB，且函式所傳回的回應也不得超過 6 MB。

Lambda 調用持續時間

Amazon Data Firehose 支援最長 5 分鐘的 Lambda 調用時間。如果您的 Lambda 函數需要超過 5 分鐘才能完成，您會收到下列錯誤：Firehose 在呼叫 AWS Lambda 時遇到逾時錯誤。支援的函數逾時上限為 5 分鐘。

如需發生此類錯誤時 Amazon Data Firehose 如何處理的資訊，請參閱 [the section called “處理資料轉換中的失敗”](#)。

資料轉換所需的參數

從 Lambda 轉換的所有記錄都必須包含下列參數，否則 Amazon Data Firehose 會拒絕這些參數，並將其視為資料轉換失敗。

For Kinesis Data Streams and Direct PUT

從 Lambda 轉換的所有記錄都需要下列參數。

- `recordId` – 在調用期間，記錄 ID 會從 Amazon Data Firehose 傳遞至 Lambda。轉換記錄必須包含相同的記錄 ID。原始記錄的 ID 與轉換記錄的 ID 若有任何不符，就會視為資料轉換失敗。
- `result` – 記錄的資料轉換狀態。可能的值包括：Ok (記錄轉換成功)、Dropped (處理邏輯刻意捨棄記錄)，以及 ProcessingFailed (記錄無法轉換)。如果記錄的狀態為 Ok 或 Dropped，Amazon Data Firehose 會將其視為已成功處理。否則，Amazon Data Firehose 會將其視為未成功處理。
- `data` – 轉換後的資料承載，在 base64 編碼之後。

以下是一個示例 Lambda 結果輸出：

```
{
  "recordId": "<recordId from the Lambda input>",
  "result": "Ok",
  "data": "<Base64 encoded Transformed data>"
}
```

For Amazon MSK

從 Lambda 轉換的所有記錄都需要下列參數。

- `recordId` – 在調用期間，記錄 ID 會從 Firehose 傳遞至 Lambda。轉換記錄必須包含相同的記錄 ID。原始記錄的 ID 與轉換記錄的 ID 若有任何不符，就會視為資料轉換失敗。
- `result` – 記錄的資料轉換狀態。可能的值包括：Ok (記錄轉換成功)、Dropped (處理邏輯刻意捨棄記錄)，以及 ProcessingFailed (記錄無法轉換)。如果記錄的狀態為 Ok 或 Dropped，Firehose 會將其視為已成功處理。否則，Firehose 會將其視為未成功處理。
- `KafkaRecordValue` – 轉換後的資料承載，在 base64 編碼之後。

以下是一個示例 Lambda 結果輸出：

```
{
  "recordId": "<recordId from the Lambda input>",
  "result": "Ok",
  "kafkaRecordValue": "<Base64 encoded Transformed data>"
}
```

支援的 Lambda 藍圖

這些藍圖示範如何建立和使用 AWS Lambda 函數來轉換 Amazon Data Firehose 資料串流中的資料。

查看 AWS Lambda 主控台中可用的藍圖

1. 登入 AWS 管理主控台，並在 <https://console.aws.amazon.com/lambda/> 開啟 AWS Lambda 主控台。
2. 選擇 Create function (建立函式)，接著選擇 Use a blueprint (使用藍圖)。
3. 在藍圖欄位中，搜尋關鍵字 firehose 以尋找 Amazon Data Firehose Lambda 藍圖。

藍圖清單：

- 傳送至 Amazon Data Firehose 串流的程序記錄 (Node.js、Python)

此藍圖顯示如何使用 AWS Lambda 處理 Firehose 資料串流中資料的基本範例。

最新發佈日期：2016 年 11 月。

發行說明：無。

- 處理傳送至 Firehose 的 CloudWatch Logs

此藍圖已棄用。請勿使用此藍圖。當解壓縮的 CloudWatch Logs 資料超過 6MB (Lambda 限制) 時，可能會產生高費用。如需有關處理傳送至 Firehose 的 CloudWatch Logs 的資訊，請參閱 [使用 CloudWatch Logs 寫入 Firehose](#)。

- 將 syslog 格式的 Amazon Data Firehose 串流記錄轉換為 JSON (Node.js)

此藍圖顯示如何將 RFC3164 Syslog 格式的輸入記錄轉換為 JSON。

最新發佈日期：2016 年 11 月。

發行說明：無。

若要查看 中可用的藍圖 AWS Serverless Application Repository

1. 前往 [AWS Serverless Application Repository](#)。
2. 選擇瀏覽所有應用程式。
3. 在 Applications (應用程式) 欄位中，搜尋關鍵字 firehose。

您也可以在不使用藍圖的情況下建立 Lambda 函數。請參閱 [AWS Lambda 入門](#)。

處理資料轉換中的失敗

如果您的 Lambda 函數調用因為網路逾時或因為您已達到 Lambda 調用限制而失敗，Amazon Data Firehose 預設會重試調用三次。如果調用不成功，Amazon Data Firehose 會略過該批次的記錄。略過的記錄會視為未處理成功。您可以使用 [CreateDeliveryStream](#) 或 [UpdateDestination](#) API 來指定或覆寫重試選項。若發生這類失敗，您可以將調用錯誤記錄至 Amazon CloudWatch Logs。如需詳細資訊，請參閱 [使用 CloudWatch Logs 監控 Amazon Data Firehose](#)。

如果記錄的資料轉換狀態為 `ProcessingFailed`，Amazon Data Firehose 會將記錄視為未成功處理。若發生此類失敗，您可透過 Lambda 函數將錯誤日誌發送到 Amazon CloudWatch Logs。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 [存取 AWS Lambda 的 Amazon CloudWatch Logs](#)。

如果資料轉換失敗，未成功處理的記錄會傳送到 `processing-failed` 資料夾中的 S3 儲存貯體。記錄的格式如下：

```
{
  "attemptsMade": "count",
  "arrivalTimestamp": "timestamp",
  "errorCode": "code",
  "errorMessage": "message",
  "attemptEndingTimestamp": "timestamp",
  "rawData": "data",
  "lambdaArn": "arn"
}
```

attemptsMade

嘗試叫用請求的次數。

arrivalTimestamp

Amazon Data Firehose 收到記錄的時間。

errorCode

Lambda 傳回 HTTP 錯誤代碼。

errorMessage

Lambda 傳回錯誤訊息。

attemptEndingTimestamp

Amazon Data Firehose 停止嘗試 Lambda 調用的時間。

rawData

Base64 編碼記錄資料。

lambdaArn

Lambda 函數的 Amazon Resource Name (ARN)。

備份來源記錄

Amazon Data Firehose 可以同時將所有未轉換的記錄備份到 S3 儲存貯體，同時將轉換的記錄交付到目的地。您可以在建立或更新 Firehose 串流時啟用來源記錄備份。啟用來源記錄備份後，便不能停用。

Amazon Data Firehose 中的分割區串流資料

動態分割可讓您使用資料中的金鑰 (例如, `customer_id` 或 `transaction_id`) 持續分割 Firehose 中的串流資料, 然後將這些金鑰分組的資料交付至對應的 Amazon Simple Storage Service (Amazon S3) 字首。這可讓您使用各種服務 (Amazon Athena、Amazon EMR、Amazon Redshift Spectrum 和 Amazon QuickSight) 對 Amazon S3 中的串流資料執行高效能且具成本效益的分析。此外, 在需要額外處理的使用案例中, AWS Glue 可以在動態分割串流資料交付至 Amazon S3 之後執行更複雜的擷取、轉換和載入 (ETL) 任務。

分割資料可最大限度地減少掃描的資料量、最佳化性能, 並降低 Amazon S3 上的分析查詢成本。它還可以增加對資料的精細存取。Firehose 串流傳統上用於擷取資料並將其載入 Amazon S3。若要分割 Amazon S3 分析的串流資料集, 您需要在 Amazon S3 儲存貯體之間執行分割應用程式, 然後才能將資料提供給分析, 這可能會變得複雜或昂貴。

透過動態分割, Firehose 會持續使用動態或靜態定義的資料金鑰將傳輸中資料分組, 並將資料依金鑰交付至個別 Amazon S3 字首。這可以縮短幾分鐘或數小時的洞察時間。它還可以降低成本並簡化架構。

主題

- [在 Amazon Data Firehose 中啟用動態分割](#)
- [了解分割金鑰](#)
- [使用 Amazon S3 儲存貯體字首交付資料](#)
- [將動態分割套用至彙總資料](#)
- [疑難排解動態分割錯誤](#)
- [動態分割的緩衝區資料](#)

在 Amazon Data Firehose 中啟用動態分割

您可以透過 Amazon Data Firehose 管理主控台、CLI 或 APIs, 為您的 Firehose 串流設定動態分割。

Important

您只能在建立新的 Firehose 串流時啟用動態分割。您無法為尚未啟用動態分割的現有 Firehose 串流啟用動態分割。

如需如何在建立新的 Firehose 串流時透過 Firehose 管理主控台啟用和設定動態分割的詳細步驟，請參閱[建立 Amazon Firehose 串流](#)。當您前往指定 Firehose 串流目的地的任務時，請務必遵循[設定目的地設定](#)區段中的步驟，因為目前，只有使用 Amazon S3 做為目的地的 Firehose 串流才支援動態分割。

啟用作用中 Firehose 串流上的動態分割後，您可以透過新增或移除或更新現有的分割金鑰和 S3 字首表達式來更新組態。更新後，Firehose 會開始使用新的金鑰和新的 S3 字首表達式。

Important

在 Firehose 串流上啟用動態分割後，就無法在此 Firehose 串流上停用。

了解分割金鑰

透過動態分割，您可以根據分割索引鍵分割資料，從串流 S3 資料建立目標資料集。分割索引鍵可讓您根據特定值篩選串流資料。例如，如果您需要根據客戶 ID 和國家/地區篩選資料，可以將 `customer_id` 的資料欄位指定為一個分割索引鍵，而將 `country` 的資料欄位指定為另一個分割索引鍵。然後，您可以指定運算式 (使用支援的格式) 來定義要傳送動態分割資料記錄的 S3 儲存貯體字首。

您可以使用下列方法建立分割金鑰。

- 內嵌剖析 – 此方法使用 Firehose 內建支援機制 [jq 剖析器](#)，從 JSON 格式的資料記錄擷取分割金鑰。目前，我們僅支援 jq 1.6 版本。
- AWS Lambda 函數 – 此方法使用指定的 AWS Lambda 函數來擷取和傳回分割所需的資料欄位。

Important

啟用動態分割時，您必須至少設定下列一種方法，以分割您的資料。您可以設定其中一種方法來同時指定您的分割索引鍵，或同時指定兩個分割索引鍵。

使用內嵌剖析建立分割金鑰

若要將內嵌剖析設定為串流資料的動態分割方法，您必須選擇要用作分割索引鍵的資料記錄參數，並為每個指定的分割索引鍵提供一個值。

下列範例資料記錄顯示如何使用內嵌剖析為其定義分割索引鍵。請注意，資料應以 Base64 格式編碼。您也可以參考 [CLI 範例](#)。

```
{
  "type": {
    "device": "mobile",
    "event": "user_clicked_submit_button"
  },
  "customer_id": "1234567890",
  "event_timestamp": 1565382027,    #epoch timestamp
  "region": "sample_region"
}
```

例如，您可以選擇根據 `customer_id` 參數或 `event_timestamp` 參數對資料進行分割。這表示您希望在決定要交付記錄的 S3 字首時，使用每個記錄中的 `customer_id` 參數或 `event_timestamp` 參數值。您也可以選擇巢狀參數，就像使用運算式 `.type.device` 的 `device` 一樣。您的動態分割邏輯可以依賴於多個參數。

選取分割索引鍵的資料參數之後，您可以將每個參數映射至有效的 jq 運算式。下表顯示了參數到 jq 運算式的映射：

參數	jq 運算式
<code>customer_id</code>	<code>.customer_id</code>
<code>device</code>	<code>.type.device</code>
<code>year</code>	<code>.event_timestamp strftime("%Y")</code>
<code>month</code>	<code>.event_timestamp strftime("%m")</code>
<code>day</code>	<code>.event_timestamp strftime("%d")</code>
<code>hour</code>	<code>.event_timestamp strftime("%H")</code>

在執行時間，Firehose 會使用上面的右欄，根據每個記錄中的資料來評估參數。

使用 AWS Lambda 函數建立分割金鑰

對於壓縮或加密的資料記錄，或 JSON 以外的任何檔案格式的資料，您可以使用整合式 AWS Lambda 函數搭配您自己的自訂程式碼來解壓縮、解密或轉換記錄，以擷取和傳回分割所需的資料欄位。這是目前 Firehose 提供的現有轉換 Lambda 函數的擴展。您可以轉換、剖析和傳回資料欄位，然後使用相同的 Lambda 函數來進行動態分割。

以下是 Python 中的 Firehose 串流處理 Lambda 函數範例，該函數會將每個讀取記錄從輸入重播至輸出，並從記錄擷取分割金鑰。

```
from __future__ import print_function
import base64
import json
import datetime

# Signature for all Lambda functions that user must implement
def lambda_handler(firehose_records_input, context):
    print("Received records for processing from DeliveryStream: " +
          firehose_records_input['deliveryStreamArn']
          + ", Region: " + firehose_records_input['region']
          + ", and InvocationId: " + firehose_records_input['invocationId'])

    # Create return value.
    firehose_records_output = {'records': []}

    # Create result object.
    # Go through records and process them

    for firehose_record_input in firehose_records_input['records']:
        # Get user payload
        payload = base64.b64decode(firehose_record_input['data'])
        json_value = json.loads(payload)

        print("Record that was received")
        print(json_value)
        print("\n")
        # Create output Firehose record and add modified payload and record ID to it.
        firehose_record_output = {}
        event_timestamp = datetime.datetime.fromtimestamp(json_value['eventTimestamp'])
        partition_keys = {"customerId": json_value['customerId'],
                          "year": event_timestamp.strftime('%Y'),
                          "month": event_timestamp.strftime('%m'),
```

```
        "day": event_timestamp.strftime('%d'),
        "hour": event_timestamp.strftime('%H'),
        "minute": event_timestamp.strftime('%M')
    }

    # Create output Firehose record and add modified payload and record ID to it.
    firehose_record_output = {'recordId': firehose_record_input['recordId'],
                              'data': firehose_record_input['data'],
                              'result': 'Ok',
                              'metadata': { 'partitionKeys': partition_keys }}

    # Must set proper record ID
    # Add the record to the list of output records.

    firehose_records_output['records'].append(firehose_record_output)

# At the end return processed records
return firehose_records_output
```

以下是 Go 中 Firehose 串流處理 Lambda 函數的範例，該函數會將每個讀取記錄從輸入重播至輸出，並從記錄擷取分割金鑰。

```
package main

import (
    "fmt"
    "encoding/json"
    "time"
    "strconv"

    "github.com/aws/aws-lambda-go/events"
    "github.com/aws/aws-lambda-go/lambda"
)

type DataFirehoseEventRecordData struct {
    CustomerId string `json:"customerId"`
}

func handleRequest(evnt events.DataFirehoseEvent) (events.DataFirehoseResponse, error) {
    fmt.Printf("InvocationID: %s\n", evnt.InvocationID)
```

```
fmt.Printf("DeliveryStreamArn: %s\n", evnt.DeliveryStreamArn)
fmt.Printf("Region: %s\n", evnt.Region)

var response events.DataFirehoseResponse

for _, record := range evnt.Records {
    fmt.Printf("RecordID: %s\n", record.RecordID)
    fmt.Printf("ApproximateArrivalTimestamp: %s\n", record.ApproximateArrivalTimestamp)

    var transformedRecord events.DataFirehoseResponseRecord
    transformedRecord.RecordID = record.RecordID
    transformedRecord.Result = events.DataFirehoseTransformedStateOk
    transformedRecord.Data = record.Data

    var metaData events.DataFirehoseResponseRecordMetadata
    var recordData DataFirehoseEventRecordData
    partitionKeys := make(map[string]string)

    currentTime := time.Now()
    json.Unmarshal(record.Data, &recordData)
    partitionKeys["customerId"] = recordData.CustomerId
    partitionKeys["year"] = strconv.Itoa(currentTime.Year())
    partitionKeys["month"] = strconv.Itoa(int(currentTime.Month()))
    partitionKeys["date"] = strconv.Itoa(currentTime.Day())
    partitionKeys["hour"] = strconv.Itoa(currentTime.Hour())
    partitionKeys["minute"] = strconv.Itoa(currentTime.Minute())
    metaData.PartitionKeys = partitionKeys
    transformedRecord.Metadata = metaData

    response.Records = append(response.Records, transformedRecord)
}

return response, nil
}

func main() {
    lambda.Start(handleRequest)
}
```

使用 Amazon S3 儲存貯體字首交付資料

當您建立使用 Amazon S3 做為目的地的 Firehose 串流時，您必須指定 Amazon S3 儲存貯體，Firehose 會在其中交付您的資料。Amazon S3 儲存貯體字首用於整理儲存在 Amazon S3 儲存貯體中的資料。Amazon S3 儲存貯體字首類似於目錄，可讓您在儲存貯體中對類似物件進行分組。

透過動態分割，您的分割資料會交付到指定的 Amazon S3 字首。如果您未啟用動態分割，則為 Firehose 串流指定 S3 儲存貯體字首是選用的。不過，如果您選擇啟用動態分割，則必須指定 Firehose 交付分割資料的 S3 儲存貯體字首。

在您啟用動態分割的每個 Firehose 串流中，S3 儲存貯體字首值包含以該 Firehose 串流的指定分割索引鍵為基礎的運算式。再次使用上述資料記錄範例，您可以建立下列 S3 字首值，該值由以上定義的分割索引鍵為基礎的運算式組成：

```
"ExtendedS3DestinationConfiguration": {
  "BucketARN": "arn:aws:s3:::my-logs-prod",
  "Prefix": "customer_id={!{partitionKeyFromQuery:customer_id}}/
    device={!{partitionKeyFromQuery:device}}/
    year={!{partitionKeyFromQuery:year}}/
    month={!{partitionKeyFromQuery:month}}/
    day={!{partitionKeyFromQuery:day}}/
    hour={!{partitionKeyFromQuery:hour}}/"
}
```

Firehose 會在執行時間評估上述表達式。它會將符合相同評估後 S3 字首運算式的記錄分組到單一資料集中。Firehose 接著會將每個資料集交付至評估的 S3 字首。資料集交付至 S3 的頻率取決於 Firehose 串流緩衝區設定。因此，本範例中的記錄會交付至下列 S3 物件金鑰：

```
s3://my-logs-prod/customer_id=1234567890/device=mobile/year=2019/month=08/day=09/
hour=20/my-delivery-stream-2019-08-09-23-55-09-a9fa96af-e4e4-409f-bac3-1f804714faaa
```

對於動態分割，您必須在 S3 儲存貯體字首中使用下列運算式格式：`!{namespace:value}`，其中命名空間可以是 `partitionKeyFromQuery` 和/或 `partitionKeyFromLambda`。如果您使用內嵌剖析來建立來源資料的分割索引鍵，則必須指定由以下格式指定之運算式所組成的 S3 儲存貯體字首值：`"partitionKeyFromQuery:keyID"`。如果您使用 AWS Lambda 函數

為來源資料建立分割索引鍵，則必須指定由以下格式指定之運算式所組成的 S3 儲存貯體字首
值："partitionKeyFromLambda:keyID"。

Note

您也可以使用 hive 樣式格式指定 S3 儲存貯體字首值，例如 `customer_id={partitionKeyFromQuery:customer_id}`。

如需詳細資訊，請參閱建立 Amazon S3 [Firehose 串流](#) 中的「為您的目的地選擇 [Amazon S3](#)」和 [Amazon S3 物件的自訂字首](#)。

將資料交付至 Amazon S3 時新增行分隔符號

您可以啟用新行分隔符號，在交付至 Amazon S3 的物件中的記錄之間新增行分隔符號。這對於在 Amazon S3 中剖析物件很有幫助。這在將動態分割套用至彙總資料時也特別有用，因為多筆記錄取消彙總（必須先套用至彙總資料，才能動態分割）會在剖析程序中從記錄移除新行。

將動態分割套用至彙總資料

您可以將動態分割套用至彙整資料（例如，將多個事件、日誌或彙整成單一 PutRecord 和 PutRecordBatch API 呼叫的記錄），但必須先將此資料取消彙整。您可以啟用多筆記錄取消彙總來取消彙總資料 - 剖析 Firehose 串流中記錄並將其分隔的程序。

多筆記錄取消彙總可以是 JSON 類型，這表示記錄的分隔是以連續的 JSON 物件為基礎。取消彙總也可以是 類型 Delimited，這表示記錄的分隔是根據指定的自訂分隔符號執行。此自訂分隔符號必須是 base-64 編碼字串。例如，如果您想要使用下列字串做為自訂分隔符號 #####，您必須以 base-64 編碼格式指定它，將其轉譯為 IyMjIw==。依 JSON 或依分隔符號進行記錄取消彙總上限為每筆記錄 500。

Note

取消彙總 JSON 記錄時，請確定您的輸入仍以支援的 JSON 格式顯示。JSON 物件必須位於不帶分隔符號或新行分隔符號 (JSONL) 的單一行上。JSON 物件的陣列不是有效的輸入。
以下是正確輸入的範例：`{"a":1}{ "a":2}` and `{"a":1}\n{"a":2}`
這是不正確輸入的範例：`[{"a":1}, {"a":2}]`

使用彙總資料時，當您啟用動態分割時，Firehose 會剖析記錄，並根據指定的多筆記錄取消彙總類型，在每個 API 呼叫中尋找有效的 JSON 物件或分隔記錄。

Important

如果您的資料已彙整，則只有在您的資料第一次取消彙整時，才能套用動態分割。

Important

當您在 Firehose 中使用資料轉換功能時，取消彙總會在資料轉換之前套用。進入 Firehose 的資料將依下列順序處理：取消彙總 → 透過 Lambda 的資料轉換 → 分割區索引鍵。

疑難排解動態分割錯誤

如果 Amazon Data Firehose 無法剖析 Firehose 串流中的資料記錄，或無法擷取指定的分割金鑰，或無法評估 S3 字首值中包含的表達式，這些資料記錄會傳送到您在建立 Firehose 串流時啟用動態分割時必須指定的 S3 錯誤儲存貯體字首。S3 錯誤儲存貯體字首包含 Firehose 無法交付至指定 S3 目的地的所有記錄。這些記錄根據錯誤類型進行組織。除了記錄之外，傳送的物件還包括錯誤的相關資訊，以協助了解並解決錯誤。

如果您想要為此 Firehose 串流啟用動態分割，您必須指定 Firehose 串流的 S3 錯誤儲存貯體字首。如果您不想為 Firehose 串流啟用動態分割，則指定 S3 錯誤儲存貯體字首是選用的。

動態分割的緩衝區資料

Amazon Data Firehose 會將傳入串流資料緩衝至特定大小，並在一段時間內將其交付至指定的目的地。您可以在建立新的 Firehose 串流時設定緩衝區大小和緩衝區間隔，或更新現有 Firehose 串流上的緩衝區大小和緩衝區間隔。緩衝區大小以 MB 為測量單位，緩衝間隔以秒為測量單位。

Note

零緩衝功能不適用於動態分割。

啟用動態分割時，Firehose 會根據設定的緩衝提示（大小和時間），在內部緩衝屬於指定分割區的記錄，再將這些記錄交付到您的 Amazon S3 儲存貯體。為了提供最大大小的物件，Firehose 會在內部使

用多階段緩衝。因此，一批記錄的端到端延遲可能是設定緩衝提示時間的 1.5 倍。這會影響 Firehose 串流的資料新鮮度。

作用中分割區計數是交付緩衝區內的作用中分割區總數。例如，如果動態分割查詢每秒建構 3 個分割區，而且您的緩衝區提示組態會每 60 秒觸發交付，則平均而言，您就會有 180 個作用中分割區。如果 Firehose 無法將分割區中的資料交付至目的地，此分割區會在交付緩衝區中計為作用中，直到可以交付為止。

根據記錄資料欄位和 S3 字首運算式將 S3 字首評估為新值時，會建立新的分割。會為每個作用中的分割建立新的緩衝區。具有相同評估 S3 字首的每個後續記錄都會交付到該緩衝區。

一旦緩衝區符合緩衝區大小限制或緩衝區時間間隔，Firehose 會使用緩衝區資料建立物件，並將其交付至指定的 Amazon S3 字首。交付物件後，該分割區的緩衝區和分割區本身會遭到刪除，並從作用中分割區計數中移除。

一旦分別符合每個分割區的緩衝區大小或間隔，Firehose 會將每個緩衝區資料做為單一物件傳遞。一旦作用中分割區的數量達到每個 Firehose 串流 500 個的限制，Firehose 串流中的其餘記錄就會交付至指定的 S3 錯誤儲存貯體字首 (activePartitionExceeded)。您可以使用 [Amazon Data Firehose 限制表單](#)，請求增加此配額，每個指定 Firehose 串流最多 5000 個作用中分割區。如果您需要更多分割區，您可以建立更多 Firehose 串流，並將作用中分割區分散到其中。

在 Amazon Data Firehose 中轉換輸入資料格式

Amazon Data Firehose 可以將輸入資料的格式從 JSON 轉換為 [Apache Parquet](#) 或 [Apache ORC](#)，然後再將資料儲存在 Amazon S3 中。Parquet 和 ORC 是單欄式資料格式，相較於 JSON 等資料列導向格式，可節省空間並啟用更快的查詢。如果您想要轉換 JSON 以外的輸入格式，例如逗號分隔值 (CSV) 或結構化文字，您可以使用將其先 AWS Lambda 轉換為 JSON。如需詳細資訊，請參閱[轉換來源資料](#)。

即使您在將記錄傳送到 Amazon Data Firehose 之前彙總記錄，也可以轉換資料的格式。

Amazon Data Firehose 需要以下三個元素來轉換記錄資料的格式：

Deserializer

Amazon Data Firehose 需要還原序列化程式來讀取輸入資料的 JSON。您可以選擇下列兩種還原序列化程式類型之一。

將多個 JSON 文件合併到同一筆記錄時，請確保您的輸入仍以支援的 JSON 格式顯示。JSON 文件陣列不是有效的輸入。

例如，這是正確的輸入：`{"a": 1}{ "b": 1}`，而這是不正確的輸入：`[{"a":1}, {"a":2}]`。

- [Apache Hive JSON SerDe](#)
- [OpenX JSON SerDe](#)

選擇 JSON 還原序列化程式

若您的輸入 JSON 內含下列格式的時間戳記，請選擇 [OpenX JSON SerDe](#)：

- `yyyy-MM-dd'T'HH:mm:ss[.S]'Z'`，其中小數點最多可有 9 位數 – 例如：`2017-02-07T15:13:01.39256Z`。
- `yyyy-[M]M-[d]d HH:mm:ss[.S]`，其中小數點最多可有 9 位數 – 例如：`2017-02-07 15:13:01.14`。
- Epoch 秒 – 例如：`1518033528`。
- Epoch 毫秒 – 例如：`1518033528123`。
- 浮點 epoch 秒 – 例如：`1518033528.123`。

OpenX JSON SerDe 可將小數點 (.) 轉換為底線 (_), 也可將 JSON 金鑰轉換為小寫, 再加以還原序列。如需透過 Amazon Data Firehose 與此還原序列化程式搭配使用之選項的詳細資訊, 請參閱 [OpenXJsonSerDe](#)。

若您不確定應選擇何種還原序列化程式, 請使用 OpenX JSON SerDe, 除非其中具有不支援的時間戳記。

若您時間戳記的格式並未列於上述, 請使用 [Apache Hive JSON SerDe](#)。選擇此還原序列化程式後, 您可指定欲使用的時間戳記格式, 方法是遵循 Joda-Time DateTimeFormat 格式字串的模式語法。如需詳細資訊, 請參閱 [Class DateTimeFormat](#)。

您亦可使用特殊值 `millis` 來剖析時間戳記 (epoch 毫秒)。如果您未指定格式, Amazon Data Firehose `java.sql.Timestamp::valueOf` 預設會使用。

Hive JSON SerDe 不允許下列事項：

- 欄位名稱內的句點 (.)。
- 類型為 `uniontype` 的欄位。
- 結構描述內有數值類型但屬於 JSON 中字串的欄位, 例如, 若結構描述為整數 (int) 且 JSON 為 `{"a":"123"}`, 則 Hive SerDe 會出現錯誤。

Hive SerDe 不會將巢狀 JSON 轉換為字串。例如, 若其中有 `{"a":{"inner":1}}`, 則 `{"inner":1}` 不會視為字串。

結構描述

Amazon Data Firehose 需要結構描述來判斷如何解譯該資料。使用 [AWS Glue](#) 在中建立結構描述 AWS Glue Data Catalog。然後, Amazon Data Firehose 會參考該結構描述, 並使用它來解譯您的輸入資料。您可以使用相同的結構描述來設定 Amazon Data Firehose 和分析軟體。如需詳細資訊, 請參閱《AWS Glue 開發人員指南》中的 [填入 AWS Glue Data Catalog](#)。

Note

在 AWS Glue Data Catalog 中建立的結構描述應與輸入資料結構相符。否則, 已轉換資料將不會包含結構描述中未指定的屬性。如果您使用巢狀 JSON, 則請在鏡像 JSON 資料結構的結構描述中使用 `STRUCT` 類型。有關如何使用 `STRUCT` 類型處理巢狀 JSON, 請參閱 [此範例](#)。

⚠ Important

對於未指定大小限制的資料類型，單一資料列中所有資料的實際限制為 32 MBs。
如果您指定 CHAR 或 的長度 VARCHAR，Firehose 會在讀取輸入資料時截斷指定長度的字串。如果基礎資料字串較長，則保持不變。

Serializer

Firehose 需要序列化程式將資料轉換為目標單欄式儲存格式 (Parquet 或 ORC) – 您可以選擇下列兩種類型的序列化程式之一。

- [ORC SerDe](#)
- [Parquet SerDe](#)

選擇序列化程式

您選擇的序列化程式取決於您的商業需求。欲進一步了解序列化程式的兩種選項，請參閱 [ORC SerDe](#) 和 [Parquet SerDe](#)。

啟用記錄格式轉換

如果您啟用記錄格式轉換，則無法將 Amazon Data Firehose 目的地設定為 Amazon OpenSearch Service、Amazon Redshift 或 Splunk。啟用格式轉換後，Amazon S3 是您可用於 Firehose 串流的唯一目的地。下一節說明如何從主控台和 Firehose API 操作啟用記錄格式轉換。如需如何使用 設定記錄格式轉換的範例 CloudFormation，請參閱 [AWS :: DataFirehose :: DeliveryStream](#)。

從主控台啟用記錄格式轉換

您可以在建立或更新 Firehose 串流時，在主控台上啟用資料格式轉換。啟用資料格式轉換後，Amazon S3 是您為 Firehose 串流設定的唯一目的地。此外，啟用格式轉換將停用 Amazon S3 壓縮。然而，轉換程序中會自動出現 Snappy 壓縮。Amazon Data Firehose 在此案例中使用的 Snappy 訊框格式與 Hadoop 相容。這表示您可使用 Snappy 壓縮的結果，並在 Athena 中查詢這些資料。如需 Hadoop 可使用的 Snappy 影格格式，請參閱 [BlockCompressorStream.java](#)。

啟用資料 Firehose 串流的資料格式轉換

1. 登入 AWS 管理主控台，然後開啟位於 <https://console.aws.amazon.com/firehose/> 的 Amazon Data Firehose 主控台。
2. 選擇要更新的 Firehose 串流，或依照中的步驟建立新的 Firehose 串流 [教學課程：從主控台建立 Firehose 串流](#)。
3. 在 Convert record format (轉換記錄格式) 底下，將 Record format conversion (記錄格式轉換) 設定為 Enabled (已啟用)。
4. 選擇您想要的輸出格式。如需兩個選項的詳細資訊，請參閱 [Apache Parquet](#) 和 [Apache ORC](#)。
5. 選擇 AWS Glue 資料表來指定來源記錄的結構描述。設定區域、資料庫、表格與表格版本。

從 Firehose API 管理記錄格式轉換

如果您希望 Amazon Data Firehose 將輸入資料的格式從 JSON 轉換為 Parquet 或 ORC，請在 `ExtendedS3DestinationConfiguration` 或 `ExtendedS3DestinationUpdate` 中指定選用的 [DataFormatConversionConfiguration](#) 元素。 [ExtendedS3DestinationConfiguration](#) [ExtendedS3DestinationUpdate](#) 如果您指定 [DataFormatConversionConfiguration](#)，則適用下列限制。

- 在 [BufferingHints](#) 中，若您啟用記錄格式轉換，`SizeInMBs` 的值將無法設定為小於 64。格式轉換未啟用時，該值預設為 5，啟用後則變成 128。
- 您必須在 [ExtendedS3DestinationConfiguration](#) 或在 [ExtendedS3DestinationUpdate](#) 中將 `CompressionFormat` 設定為 `UNCOMPRESSED`。`CompressionFormat` 的預設值為 `UNCOMPRESSED`。因此，[ExtendedS3DestinationConfiguration](#) 也可不指定該值。序列化程序仍會壓縮這些資料，預設使用 Snappy 壓縮。Amazon Data Firehose 在此案例中使用的 Snappy 訊框格式與 Hadoop 相容。這表示您可使用 Snappy 壓縮的結果，並在 Athena 中查詢這些資料。如需 Hadoop 可使用的 Snappy 影格格式，請參閱 [BlockCompressorStream.java](#)。設定序列化程式時，您可選擇其他壓縮類型。

處理資料格式轉換的錯誤

當 Amazon Data Firehose 無法剖析或還原序列化記錄時（例如，當資料不符合結構描述時），它會將它寫入具有錯誤字首的 Amazon S3。如果此寫入失敗，Amazon Data Firehose 會永久重試，封鎖進一步交付。對於每個失敗的記錄，Amazon Data Firehose 會使用下列結構描述寫入 JSON 文件：

```
{
  "attemptsMade": long,
```

```
"arrivalTimestamp": long,  
"errorCode": string,  
"errorMessage": string,  
"attemptEndingTimestamp": long,  
"rawData": string,  
"sequenceNumber": string,  
"subSequenceNumber": long,  
"dataCatalogTable": {  
  "catalogId": string,  
  "databaseName": string,  
  "tableName": string,  
  "region": string,  
  "versionId": string,  
  "catalogArn": string  
}  
}
```

了解 Amazon Data Firehose 中的資料交付

當您將資料傳送至 Firehose 串流時，資料會自動傳送至您選擇的目的地。下表說明資料交付至不同目的地的方式。

目標	詳細資訊
Amazon S3	為了將資料交付至 Amazon S3，Firehose 會根據 Firehose 串流的緩衝組態來串連多個傳入記錄。接著，該服務會將這些記錄作為 Amazon S3 物件，並交付至 Amazon S3。根據預設，Firehose 會串連資料，而沒有任何分隔符號。如果您想要在記錄之間有新的行分隔符號，您可以透過在 Firehose 主控台組態 或 API 參數 中啟用 功能來新增行分隔符號。Firehose 和 Amazon S3 目的地之間的資料交付會使用 TLS (HTTPS) 加密。
Amazon Redshift	若要將資料交付至 Amazon Redshift，Firehose 會先以前述格式將傳入資料交付至 S3 儲存貯體。接著 Firehose 會發出 Amazon Redshift COPY 命令，將資料從 S3 儲存貯體載入 Amazon Redshift 佈建叢集或 Amazon Redshift Serverless 工作群組。確保 Amazon Data Firehose 將多個傳入記錄串連至 Amazon S3 物件後，Amazon S3 物件可以複製到您的 Amazon Redshift 佈建叢集或 Amazon Redshift Serverless 工作群組。如需詳細資訊，請參閱 Amazon Redshift COPY 命令資料格式參數 。
OpenSearch Service 和 OpenSearch Serverless	對於交付至 OpenSearch Service 和 OpenSearch Serverless 的資料，Amazon Data Firehose 會根據 Firehose 串流的緩衝組態來緩衝傳入的記錄。然後，它會產生 OpenSearch Service 或 OpenSearch Serverless 大量請求，在您的 OpenSearch Service 叢集或 OpenSearch Serverless 集合為多個記錄建立索引。將單行 JSON 物件傳送到 Amazon Data Firehose 之前，請確定您的記錄已進行 UTF-8 編碼並扁平化。此外，您還必須將 OpenSearch Service 叢集的 <code>rest.action.multi.allow_explicit_index</code> 選項設定為 true (預設值)，才能夠根據每筆記錄設定的明確索引接受大量請求。如需更多資訊，請參閱《Amazon OpenSearch Service 開發人員指南》中的 Amazon OpenSearch Service 進階選項 。

目標	詳細資訊
Splunk	對於傳送到 Splunk 的資料，Amazon Data Firehose 會串連您傳送的位元組。如果您的資料需要定界符號 (例如換行字元)，您必須自行插入。請確認將 Splunk 設定為剖析這類定界符號。若要將交付至 S3 錯誤儲存貯體 (S3 備份) 的資料重新驅動回 Splunk，請遵循 Splunk 文件中 所述的步驟。
HTTP 端點	若要將資料交付至受支援的第三方服務提供者所擁有的 HTTP 端點，您可以使用整合的 Amazon Lambda 服務來建立函數，將傳入記錄轉換為符合服務提供者整合預期的格式 (格式)。請聯絡您為目的地選擇 HTTP 端點的第三方服務供應商，以進一步了解其接受的記錄格式。
Snowflake	為了將資料交付至 Snowflake，Amazon Data Firehose 會在內部緩衝資料一秒鐘，並使用 Snowflake 串流 API 操作將資料插入 Snowflake。根據預設，您插入的記錄會每秒排清並遞交至 Snowflake 資料表。在您進行插入呼叫後，Firehose 會發出 CloudWatch 指標，測量資料遞交至 Snowflake 所需的時間。Firehose 目前僅支援單一 JSON 項目做為記錄承載，且不支援 JSON 陣列。請確定您的輸入承載是有效的 JSON 物件，而且格式良好，沒有任何額外的雙引號、引號或逸出字元。

每個 Firehose 目的地都有自己的資料交付頻率。如需詳細資訊，請參閱[設定緩衝提示](#)。

複製記錄

Amazon Data Firehose 使用 at-least-once 的語意進行資料交付。在某些情況下，例如資料交付逾時時，如果原始資料交付請求最終通過，Amazon Data Firehose 的交付重試可能會引入重複項目。這適用於 Amazon Data Firehose 支援的所有目的地類型，但 Amazon S3 目的地、Apache Iceberg Tables 和 Snowflake 目的地除外。

主題

- [了解跨 AWS 帳戶和區域的交付](#)
- [了解 HTTP 端點交付請求和回應規格](#)
- [處理資料交付失敗](#)
- [設定 Amazon S3 物件名稱格式](#)

- [設定 OpenSearch Service 的索引輪換](#)
- [暫停和繼續資料交付](#)

了解跨 AWS 帳戶和區域的交付

Amazon Data Firehose 支援跨 AWS 帳戶將資料交付至 HTTP 端點目的地。您選擇做為目的地的 Firehose 串流和 HTTP 端點可以屬於不同的 AWS 帳戶。

Amazon Data Firehose 也支援跨 AWS 區域將資料交付至 HTTP 端點目的地。您可以將資料從一個 AWS 區域中的 Firehose 串流傳遞到另一個區域中的 HTTP 端點 AWS。您也可以將 HTTP 端點 URL 設定為所需的目的地，將資料從 Firehose 串流交付到 AWS 區域外的 HTTP 端點目的地，例如您自己的內部部署伺服器。在這些情況下，額外的資料傳輸費用會新增到您的交付成本中。如需詳細資訊，請參閱「隨需定價」頁面上的[資料傳輸](#)一節。

了解 HTTP 端點交付請求和回應規格

若要讓 Amazon Data Firehose 成功將資料交付至自訂 HTTP 端點，這些端點必須接受請求，並使用特定 Amazon Data Firehose 請求和回應格式傳送回應。本節說明 Amazon Data Firehose 服務傳送至自訂 HTTP 端點的 HTTP 請求格式規格，以及 Amazon Data Firehose 服務預期的 HTTP 回應格式規格。HTTP 端點有 3 分鐘的時間在 Amazon Data Firehose 逾時該請求之前回應請求。Amazon Data Firehose 會將不符合適當格式的回應視為交付失敗。

要求格式

路徑和 URL 參數

這些是由您作為單一 URL 欄位的一部分直接進行設定。Amazon Data Firehose 會依設定傳送它們，無需修改。僅 https 目的地受支援。系統會在交付串流組態期間套用 URL 限制。

Note

目前，HTTP 端點資料交付僅支援連接埠 443。

HTTP 標頭 – X-Amz-Firehose-Protocol-Version

此標頭用於指示請求/回應格式的版本。目前僅支援 1.0 版本。

HTTP 標頭 – X-Amz-Firehose-Request-Id

此標頭的值是不透明的 GUID，可用於偵錯和重複資料刪除目的。如果可能，端點實作應該記錄此標頭的值，以滿足成功和不成功的請求。在相同請求的多次嘗試之間，請求 ID 保持不變。

HTTP 標頭 – Content-Type

Content-Type 標頭的值永遠是 application/json。

HTTP 標頭 – Content-Encoding

Firehose 串流可設定為在傳送請求時使用 GZIP 壓縮內文。啟用此壓縮時，根據標準實務，Content-Encoding 標頭的值會設定為 gzip。如果未啟用壓縮，則 Content-Encoding 標頭完全不存在。

HTTP 標頭 – Content-Length

以標準方式對其進行使用。

HTTP 標頭 – X-Amz-Firehose-Source-Arn :

以 ASCII 字串格式表示的 Firehose 串流 ARN。ARN 會編碼區域、AWS 帳戶 ID 和串流名稱。例如 arn:aws:firehose:us-east-1:123456789:deliverystream/testStream。

HTTP 標頭 – X-Amz-Firehose-Access-Key

此標頭帶有 API 金鑰或其他憑證。您可以在建立或更新交付串流時建立或更新 API 金鑰 (也稱為授權記號)。Amazon Data Firehose 會將存取金鑰的大小限制為 4096 個位元組。Amazon Data Firehose 不會嘗試以任何方式解譯此金鑰。設定的金鑰會逐字複製到此標頭的值中。不過，如果您使用 Secrets Manager 來設定金鑰，則秘密必須遵循特定的 JSON 物件格式：{"api_key": "..."}。

內容可以是任意的，並且可能代表 JWT 記號或 ACCESS_KEY。如果端點需要多欄位憑證 (例如，使用者名稱和密碼)，則所有欄位的值都應以端點理解的格式 (JSON 或 CSV) 儲存在單一存取金鑰中。如果原始內容是二進位，則此欄位可以是 base-64 編碼。Amazon Data Firehose 不會修改和/或編碼設定的值，並照原樣使用內容。

HTTP 標頭 – X-Amz-Firehose-Common-Attributes

該標頭包含與整個請求和/或請求中的所有記錄有關的共同屬性 (中繼資料)。這些是由您在建立 Firehose 串流時直接設定。系統會將此屬性的值編碼為具有下列結構描述的 JSON 物件：

```
"$schema": http://json-schema.org/draft-07/schema#
```

```

properties:
  commonAttributes:
    type: object
    minProperties: 0
    maxProperties: 50
    patternProperties:
      "^.{1,256}$":
        type: string
        minLength: 0
        maxLength: 1024

```

範例如下：

```

"commonAttributes": {
  "deployment -context": "pre-prod-gamma",
  "device-types": ""
}

```

主體 – 大小上限

壓縮之前，主體大小上限由您設定，最大可達 64 MiB。

主體 – 結構描述

主體包含具有下列 JSON 結構描述 (以 YAML 寫入) 的單一 JSON 文件：

```

"$schema": http://json-schema.org/draft-07/schema#

title: FirehoseCustomHttpsEndpointRequest
description: >
  The request body that the Firehose service sends to
  custom HTTPS endpoints.
type: object
properties:
  requestId:
    description: >
      Same as the value in the X-Amz-Firehose-Request-Id header,
      duplicated here for convenience.
    type: string
  timestamp:

```

```

description: >
  The timestamp (milliseconds since epoch) at which the Firehose
  server generated this request.
type: integer
records:
  description: >
    The actual records of the Firehose stream, carrying
    the customer data.
  type: array
  minItems: 1
  maxItems: 10000
  items:
    type: object
    properties:
      data:
        description: >
          The data of this record, in Base64. Note that empty
          records are permitted in Firehose. The maximum allowed
          size of the data, before Base64 encoding, is 1024000
          bytes; the maximum length of this field is therefore
          1365336 chars.
        type: string
        minLength: 0
        maxLength: 1365336

required:
  - requestId
  - records

```

範例如下：

```

{
  "requestId": "ed4acda5-034f-9f42-bba1-f29aea6d7d8f",
  "timestamp": 1578090901599
  "records": [
    {
      "data": "aGVsbG8="
    },
    {
      "data": "aGVsbG8gd29ybGQ="
    }
  ]
}

```

```
]
}
```

回應格式

出現錯誤時的預設行為

如果回應不符合下列要求，Firehose 伺服器會將其視為具有 500 狀態碼且沒有內文。

狀態碼

HTTP 狀態碼必須在 2XX、4XX 或 5XX 的範圍內。

Amazon Data Firehose 伺服器不遵循重新導向 (3XX 狀態碼)。只將回應碼 200 視為成功將記錄交付至 HTTP/EP。會將回應碼 413 (大小已超出) 視為永久失效，如果已設定，則不會將記錄批次傳送至錯誤儲存貯體。會將所有其他回應碼視為可重試的錯誤，並受限於稍後解釋的輪詢重試演算法。

標頭 – 內容類型

唯一可接受的內容類型是應用程式/json。

HTTP 標頭 – Content-Encoding

不能使用 Content-Encoding。必須對主體進行解壓縮。

HTTP 標頭 – Content-Length

如果回應有主體，則 Content-Length 標頭必須存在。

主體 – 大小上限

回應主體的大小必須為小於或等於 1 MiB。

```
"$schema": http://json-schema.org/draft-07/schema#
title: FirehoseCustomHttpsEndpointResponse
description: >
  The response body that the Firehose service sends to
  custom HTTPS endpoints.
```

```
type: object
properties:
  requestId:
    description: >
      Must match the requestId in the request.
    type: string

  timestamp:
    description: >
      The timestamp (milliseconds since epoch) at which the
      server processed this request.
    type: integer

  errorMessage:
    description: >
      For failed requests, a message explaining the failure.
      If a request fails after exhausting all retries, the last
      Instance of the error message is copied to error output
      S3 bucket if configured.
    type: string
    minLength: 0
    maxLength: 8192
required:
  - requestId
  - timestamp
```

範例如下：

```
Failure Case (HTTP Response Code 4xx or 5xx)
{
  "requestId": "ed4acda5-034f-9f42-bba1-f29aea6d7d8f",
  "timestamp": "1578090903599",
  "errorMessage": "Unable to deliver records due to unknown error."
}
Success case (HTTP Response Code 200)
{
  "requestId": "ed4acda5-034f-9f42-bba1-f29aea6d7d8f",
  "timestamp": 1578090903599
}
```

錯誤回應處理

在所有錯誤情況下，Amazon Data Firehose 伺服器會使用指數退避演算法重新嘗試交付相同批次的記錄。重試會使用初始退避時間 (1 秒) 和抖動係數 (15%) 進行退避，並且每次後續重試都會使用新增抖動的公式 ($\text{initial-backoff-time} * (\text{multiplier}(2) ^ \text{retry_count})$) 來退避。退避時間受限於 2 分鐘的間隔上限。例如，在第 n 次重試時，退避時間為 $= \text{MAX}(120, 2^n) * \text{random}(0.85, 1.15)$ 。

在上一個方程式中所指定的參數可能會發生變更。如需確切的初始退避時間、最大退避時間、倍數和指數退避演算法中使用的抖動百分比，請參閱 AWS Firehose 文件。

每次後續重試嘗試時，交付記錄的存取金鑰和/或目的地可能會根據 Firehose 串流的更新組態而變更。Amazon Data Firehose 服務會盡最大努力跨重試使用相同的 request-id。可將這最後一個功能用於 HTTP 端點伺服器的重複資料刪除目的。如果請求在允許的最長時間（根據 Firehose 串流組態）之後仍未交付，批次記錄可以根據串流組態選擇性地交付到錯誤儲存貯體。

範例

CWLog 來源請求的範例。

```
{
  "requestId": "ed4acda5-034f-9f42-bba1-f29aea6d7d8f",
  "timestamp": 1578090901599,
  "records": [
    {
      "data": {
        "messageType": "DATA_MESSAGE",
        "owner": "123456789012",
        "logGroup": "log_group_name",
        "logStream": "log_stream_name",
        "subscriptionFilters": [
          "subscription_filter_name"
        ],
        "logEvents": [
          {
            "id": "01234567890123456789012345678901234567890123456789012345",
            "timestamp": 1510109208016,
            "message": "log message 1"
          },
          {
            "id": "01234567890123456789012345678901234567890123456789012345",
            "timestamp": 1510109208017,
```

```
        "message": "log message 2"
      }
    ]
  }
}
]
```

處理資料交付失敗

每個 Amazon Data Firehose 目的地都有自己的資料交付失敗處理。

當您設定 Firehose 串流時，對於許多目的地，例如 OpenSearch、Splunk 和 HTTP 端點，您也可以設定 S3 儲存貯體，其中可以備份無法交付的資料。如需 Firehose 在交付失敗時如何備份資料的詳細資訊，請參閱此頁面的相關目的地區段。如需如何授予無法交付資料之 S3 儲存貯體存取權的詳細資訊，請參閱[授予 Firehose 存取 Amazon S3 目的地](#)。當 Firehose (a) 無法將資料交付至串流目的地，以及 (b) 無法將資料寫入備份 S3 儲存貯體以取得失敗的交付時，它會有效地暫停串流交付，直到資料可以交付至目的地或寫入備份 S3 位置為止。

Amazon S3

資料交付至 S3 儲存貯體時，可能會因各種問題導致失敗。例如，儲存貯體可能不再存在，Amazon Data Firehose 擔任的 IAM 角色可能無法存取儲存貯體、網路失敗或類似事件。在這些情況下，Amazon Data Firehose 會持續重試最多 24 小時，直到交付成功為止。Amazon Data Firehose 的資料儲存時間上限為 24 小時。如果超過 24 小時仍無法交付，資料將會遺失。

資料交付至 S3 儲存貯體可能會因為各種原因而失敗，例如：

- 儲存貯體不再存在。
- Amazon Data Firehose 擔任的 IAM 角色無法存取儲存貯體。
- 網路問題。
- S3 錯誤，例如 HTTP 500s 或其他 API 失敗。

在這些情況下，Amazon Data Firehose 將重試交付：

- DirectPut 來源：重試會持續長達 24 小時。
- Kinesis Data Streams 或 Amazon MSK 來源：重試會無限期繼續，直到串流上定義的保留政策為止。

只有在 Lambda 處理或 parquet 轉換失敗時，Amazon Data Firehose 才會將失敗的記錄傳送至 S3 錯誤儲存貯體。其他失敗案例將導致 S3 持續重試嘗試，直到達到保留期為止。當 Firehose 成功將記錄交付至 S3 時，它會建立 S3 物件檔案，並在部分記錄失敗時，自動重試交付，並使用成功處理的記錄更新相同的 S3 物件檔案。

Amazon Redshift

對於 Amazon Redshift 目的地，您可以在建立 Firehose 串流時指定重試持續時間 (0–7200 秒)。

將資料交付到您的 Amazon Redshift 佈建的叢集或 Amazon Redshift Serverless 工作群組可能會因多種原因而失敗。例如，您的 Firehose 串流的叢集組態可能不正確、叢集或工作群組正在進行維護，或網路故障。在這些情況下，Amazon Data Firehose 會在指定的持續時間內重試，並略過該特定批次的 Amazon S3 物件。略過物件的資訊會傳送至 S3 儲存貯體，成為 errors/資料夾中的資訊清單檔案，以供手動回填使用。如需進一步了解如何使用資訊清單檔案來手動 COPY 資料，請參閱[使用資訊清單來指定資料檔案](#)。

Amazon OpenSearch Service 和 OpenSearch Serverless

對於 OpenSearch Service 和 OpenSearch Serverless 目的地，您可以在 Firehose 串流建立期間指定重試持續時間 (0–7200 秒)。

交付至您的 OpenSearch Service 叢集或 OpenSearch Serverless 集合的資料可能會因為幾個原因而失敗。例如，您可能有不正確的 OpenSearch Service 叢集或 Firehose 串流的 OpenSearch Serverless 集合組態、OpenSearch Service 叢集或維護中的 OpenSearch Serverless 集合、網路故障或類似事件。在這些情況下，Amazon Data Firehose 會在指定的時間內重試，然後略過該特定索引請求。略過的文件會傳輸至 S3 儲存貯體的 AmazonOpenSearchService_failed/資料夾，以供手動回填使用。

對於 OpenSearch Server，每份文件具有下列 JSON 格式：

```
{
  "attemptsMade": "(number of index requests attempted)",
  "arrivalTimestamp": "(the time when the document was received by Firehose)",
  "errorCode": "(http error code returned by OpenSearch Service)",
  "errorMessage": "(error message returned by OpenSearch Service)",
  "attemptEndingTimestamp": "(the time when Firehose stopped attempting index request)",
  "esDocumentId": "(intended OpenSearch Service document ID)",
  "esIndexName": "(intended OpenSearch Service index name)",
  "esTypeName": "(intended OpenSearch Service type name)",
```

```
"rawData": "(base64-encoded document data)"
}
```

對於 OpenSearch Serverless，每份文件具有下列 JSON 格式：

```
{
  "attemptsMade": "(number of index requests attempted)",
  "arrivalTimestamp": "(the time when the document was received by Firehose)",
  "errorCode": "(http error code returned by OpenSearch Serverless)",
  "errorMessage": "(error message returned by OpenSearch Serverless)",
  "attemptEndingTimestamp": "(the time when Firehose stopped attempting index request)",
  "osDocumentId": "(intended OpenSearch Serverless document ID)",
  "osIndexName": "(intended OpenSearch Serverless index name)",
  "rawData": "(base64-encoded document data)"
}
```

Splunk

Amazon Data Firehose 將資料傳送至 Splunk 時，會等待 Splunk 的確認。如果發生錯誤，或確認未在確認逾時期間內送達，Amazon Data Firehose 會啟動重試持續時間計數器。它會持續重試，直到重試持續時間過期為止。之後，Amazon Data Firehose 會將其視為資料交付失敗，並將資料備份到您的 Amazon S3 儲存貯體。

每次 Amazon Data Firehose 將資料傳送至 Splunk 時，無論是初次嘗試還是重試，都會重新啟動確認逾時計數器。並等待 Splunk 傳回確認訊息。即使重試持續時間過期，Amazon Data Firehose 仍會等待確認，直到收到確認或確認逾時為止。如果確認逾時，Amazon Data Firehose 會檢查以確定重試計數器中是否還有時間。如果還有時間，將再次重試並重複執行邏輯，直到收到確認或判斷已無重試時間為止。

未收到確認訊息，並非唯一的資料交付錯誤類型。如需其他資料交付錯誤類型的相關資訊，請參閱 [Splunk 資料交付錯誤](#)。如果您的重試期間大於 0，則任何資料交付錯誤都會觸發重試邏輯。

錯誤記錄例示如下。

```
{
  "attemptsMade": 0,
  "arrivalTimestamp": 1506035354675,
```


Snowflake

對於 Snowflake 目的地，當您建立 Firehose 串流時，您可以指定選用的重試持續時間 (0-7200 秒)。重試持續時間的預設值為 60 秒。

資料交付至 Snowflake 資料表可能會失敗，原因包括 Snowflake 目的地組態不正確、Snowflake 中斷、網路故障等。重試政策不適用於不可重試的錯誤。例如，如果 Snowflake 拒絕您的 JSON 承載，因為資料表中缺少額外的資料欄，Firehose 不會再次嘗試交付。相反地，它會為因對 S3 錯誤儲存貯體的 JSON 承載問題而導致的所有插入失敗建立備份。

同樣地，如果交付因不正確的角色、資料表或資料庫而失敗，Firehose 不會重試並將資料寫入您的 S3 儲存貯體。重試持續時間僅適用於由於 Snowflake 服務問題、暫時性網路故障等造成的失敗。在這些情況下，Firehose 會在指定的時間內重試，然後再將其交付至 S3。失敗的記錄會以 snowflake-failed/資料夾交付，您可以用來手動回填。

以下是您交付給 S3 的每個記錄的範例 JSON。

```
{
  "attemptsMade": 3,
  "arrivalTimestamp": 1594265943615,
  "errorCode": "Snowflake.InvalidColumns",
  "errorMessage": "Snowpipe Streaming does not support columns of type AUTOINCREMENT,
IDENTITY, GEO, or columns with a default value or collation",
  "attemptEndingTimestamp": 1712937865543,
  "rawData": "c2FtcGx1IHJhdYBkYXRh"
}
```

設定 Amazon S3 物件名稱格式

當 Firehose 將資料交付至 Amazon S3 時，S3 物件金鑰名稱會遵循 <evaluated prefix><suffix> 格式，其中尾碼的格式為 <Firehose stream name>-<Firehose stream version>-<year>-<month>-<day>-<hour>-<minute>-<second>-<uuid><file extension> <Firehose stream version>，且每次 Firehose 串流的組態變更時都會增加 1。您可以變更 Firehose 串流組態（例如，S3 儲存貯體的名稱、緩衝提示、壓縮和加密）。您可以使用 Firehose 主控台或 [UpdateDestination](#) API 操作來執行此操作。

對於 <評估字首>，Firehose 會新增格式為的預設時間字首 YYYY/MM/dd/HH。此字首會在儲存貯體中建立邏輯階層，其中每個正斜線 (/) 會在階層中建立層級。您可以指定自訂字首來修改此結構，其中包含在執行時間評估的表達式。如需如何指定自訂字首的資訊，請參閱 [Amazon Simple Storage Service 物件的自訂字首](#)。

根據預設，用於時間字首和尾碼的時區為 UTC，但您可以將其變更為您偏好的時區。例如，若要使用日本標準時間而非 UTC，您可以在 AWS 管理主控台 或 [API 參數設定 \(CustomTimeZone\)](#) 中將時區設定為亞洲/東京。下列清單包含 Firehose 針對 S3 字首組態支援的時區。

支援的時區

以下是 Firehose 針對 S3 字首組態支援的時區清單。

Africa

```
Africa/Abidjan
Africa/Accra
Africa/Addis_Ababa
Africa/Algiers
Africa/Asmera
Africa/Bangui
Africa/Banjul
Africa/Bissau
Africa/Blantyre
Africa/Bujumbura
Africa/Cairo
Africa/Casablanca
Africa/Conakry
Africa/Dakar
Africa/Dar_es_Salaam
Africa/Djibouti
Africa/Douala
Africa/Freetown
Africa/Gaborone
Africa/Harare
Africa/Johannesburg
Africa/Kampala
Africa/Khartoum
Africa/Kigali
Africa/Kinshasa
Africa/Lagos
Africa/Libreville
Africa/Lome
Africa/Luanda
Africa/Lubumbashi
Africa/Lusaka
Africa/Malabo
Africa/Maputo
```

Africa/Maseru
Africa/Mbabane
Africa/Mogadishu
Africa/Monrovia
Africa/Nairobi
Africa/Ndjamena
Africa/Niamey
Africa/Nouakchott
Africa/Ouagadougou
Africa/Porto-Novo
Africa/Sao_Tome
Africa/Timbuktu
Africa/Tripoli
Africa/Tunis
Africa/Windhoek

America

America/Adak
America/Anchorage
America/Anguilla
America/Antigua
America/Aruba
America/Asuncion
America/Barbados
America/Belize
America/Bogota
America/Buenos_Aires
America/Caracas
America/Cayenne
America/Cayman
America/Chicago
America/Costa_Rica
America/Cuiaba
America/Curacao
America/Dawson_Creek
America/Denver
America/Dominica
America/Edmonton
America/El_Salvador
America/Fortaleza
America/Godthab
America/Grand_Turk

America/Grenada
America/Guadeloupe
America/Guatemala
America/Guayaquil
America/Guyana
America/Halifax
America/Havana
America/Indianapolis
America/Jamaica
America/La_Paz
America/Lima
America/Los_Angeles
America/Managua
America/Manaus
America/Martinique
America/Mazatlan
America/Mexico_City
America/Miquelon
America/Montevideo
America/Montreal
America/Montserrat
America/Nassau
America/New_York
America/Noronha
America/Panama
America/Paramaribo
America/Phoenix
America/Port_of_Spain
America/Port-au-Prince
America/Porto_Acre
America/Puerto_Rico
America/Regina
America/Rio_Branco
America/Santiago
America/Santo_Domingo
America/Sao_Paulo
America/Scoresbysund
America/St_Johns
America/St_Kitts
America/St_Lucia
America/St_Thomas
America/St_Vincent
America/Tegucigalpa
America/Thule

America/Tijuana
America/Tortola
America/Vancouver
America/Winnipeg

Antarctica

Antarctica/Casey
Antarctica/DumontDUrville
Antarctica/Mawson
Antarctica/McMurdo
Antarctica/Palmer

Asia

Asia/Aden
Asia/Almaty
Asia/Amman
Asia/Anadyr
Asia/Aqtau
Asia/Aqtobe
Asia/Ashgabat
Asia/Ashkhabad
Asia/Baghdad
Asia/Bahrain
Asia/Baku
Asia/Bangkok
Asia/Beirut
Asia/Bishkek
Asia/Brunei
Asia/Calcutta
Asia/Colombo
Asia/Dacca
Asia/Damascus
Asia/Dhaka
Asia/Dubai
Asia/Dushanbe
Asia/Hong_Kong
Asia/Irkutsk
Asia/Jakarta
Asia/Jayapura
Asia/Jerusalem
Asia/Kabul

Asia/Kamchatka
Asia/Karachi
Asia/Katmandu
Asia/Krasnoyarsk
Asia/Kuala_Lumpur
Asia/Kuwait
Asia/Macao
Asia/Magadan
Asia/Manila
Asia/Muscat
Asia/Nicosia
Asia/Novosibirsk
Asia/Phnom_Penh
Asia/Pyongyang
Asia/Qatar
Asia/Rangoon
Asia/Riyadh
Asia/Saigon
Asia/Seoul
Asia/Shanghai
Asia/Singapore
Asia/Taipei
Asia/Tashkent
Asia/Tbilisi
Asia/Tehran
Asia/Thimbu
Asia/Thimphu
Asia/Tokyo
Asia/Ujung_Pandang
Asia/Ulaanbaatar
Asia/Ulan_Bator
Asia/Vientiane
Asia/Vladivostok
Asia/Yakutsk
Asia/Yekaterinburg
Asia/Yerevan

Atlantic

Atlantic/Azores
Atlantic/Bermuda
Atlantic/Canary
Atlantic/Cape_Verde

Atlantic/Faeroe
Atlantic/Jan_Mayen
Atlantic/Reykjavik
Atlantic/South_Georgia
Atlantic/St_Helena
Atlantic/Stanley

Australia

Australia/Adelaide
Australia/Brisbane
Australia/Broken_Hill
Australia/Darwin
Australia/Hobart
Australia/Lord_Howe
Australia/Perth
Australia/Sydney

Europe

Europe/Amsterdam
Europe/Andorra
Europe/Athens
Europe/Belgrade
Europe/Berlin
Europe/Brussels
Europe/Bucharest
Europe/Budapest
Europe/Chisinau
Europe/Copenhagen
Europe/Dublin
Europe/Gibraltar
Europe/Helsinki
Europe/Istanbul
Europe/Kaliningrad
Europe/Kiev
Europe/Lisbon
Europe/London
Europe/Luxembourg
Europe/Madrid
Europe/Malta
Europe/Minsk
Europe/Monaco

Europe/Moscow
Europe/Oslo
Europe/Paris
Europe/Prague
Europe/Riga
Europe/Rome
Europe/Samara
Europe/Simferopol
Europe/Sofia
Europe/Stockholm
Europe/Tallinn
Europe/Tirane
Europe/Vaduz
Europe/Vienna
Europe/Vilnius
Europe/Warsaw
Europe/Zurich

Indian

Indian/Antananarivo
Indian/Chagos
Indian/Christmas
Indian/Cocos
Indian/Comoro
Indian/Kerguelen
Indian/Mahe
Indian/Maldives
Indian/Mauritius
Indian/Mayotte
Indian/Reunion

Pacific

Pacific/Apia
Pacific/Auckland
Pacific/Chatham
Pacific/Easter
Pacific/Efate
Pacific/Enderbury
Pacific/Fakaofu
Pacific/Fiji
Pacific/Funafuti

Pacific/Galapagos
 Pacific/Gambier
 Pacific/Guadalcanal
 Pacific/Guam
 Pacific/Honolulu
 Pacific/Kiritimati
 Pacific/Kosrae
 Pacific/Majuro
 Pacific/Marquesas
 Pacific/Nauru
 Pacific/Niue
 Pacific/Norfolk
 Pacific/Noumea
 Pacific/Pago_Pago
 Pacific/Palau
 Pacific/Pitcairn
 Pacific/Ponape
 Pacific/Port_Moresby
 Pacific/Rarotonga
 Pacific/Saipan
 Pacific/Tahiti
 Pacific/Tarawa
 Pacific/Tongatapu
 Pacific/Truk
 Pacific/Wake
 Pacific/Wallis

除了 <file extension> 之外，您無法變更尾碼欄位。當您啟用資料格式轉換或壓縮時，Firehose 會根據組態附加副檔名。下表說明 Firehose 附加的預設副檔名：

Configuration	副檔名
資料格式轉換：Parquet	.parquet
資料格式轉換：ORC	.orc
壓縮：Gzip	.gz
壓縮：Zip	.zip
壓縮：Snappy	.snappy

Configuration	副檔名
壓縮 : Hadoop-Snappy	.hsnappy

您也可以在此 Firehose 主控台或 API 中指定您偏好的副檔名。副檔名必須以句號 (.) 開頭，且可包含允許的字元：0-9a-z ! -_.*()。副檔名不能超過 128 個字元。

Note

當您指定副檔名時，它會覆寫 Firehose 在啟用[資料格式轉換](#)或壓縮時新增的預設副檔名。

了解 Amazon S3 物件的自訂字首

交付至 Amazon S3 的物件會遵循 `<evaluated prefix><suffix>` 的**名稱格式**。您可以指定自訂字首，其中包含在執行時間評估的表達式。您指定的自訂字首會覆寫的預設字首 `yyyy/MM/dd/HH`。

自訂字首可使用下列形式的運算式：`!{namespace:value}`，其中 `namespace` 可為以下項目之一，如下列部分所述。

- `firehose`
- `timestamp`
- `partitionKeyFromQuery`
- `partitionKeyFromLambda`

如果字首結尾為斜線，看起來會像是 Amazon S3 儲存貯體內的資料夾。如需詳細資訊，請參閱 [《Amazon Data Firehose Developer 指南》](#) 中的 [Amazon S3 物件名稱格式](#)。Firehose Developer

timestamp 命名空間

此命名空間的有效值為 [Java DateTimeFormatter](#) 的有效字串。例如，在 2018 年，表達式 `{timestamp:yyyy}` 的評估結果為 2018。

評估時間戳記時，Firehose 會使用所寫入 Amazon S3 物件中包含之最舊記錄的大致到達時間戳記。

根據預設，時間戳記為 UTC。但是，您可以指定您偏好的時區。例如，如果您想要使用日本標準時間而非 UTC，您可以在 或 API 參數設定 ([CustomTimeZone](#)) AWS 管理主控台 中將時區設定為亞洲/東京。若要查看支援的時區清單，請參閱 [Amazon S3 物件名稱格式](#)。

若您在同一字首表達式內多次使用 timestamp 命名空間，每個執行個體的評估結果都是相同的時間。

firehose 命名空間

此命名空間可使用兩個值：error-output-type 和 random-string。下表說明如何使用這兩個值。

firehose 命名空間值

Conversion (轉換)	Description	範例輸入	範例輸出	備註
error-output-type	<p>根據 Firehose 串流的組態以及失敗原因，評估下列其中一個字串：{processing-failed, AmazonOpenSearchService-failed, splunk-failed, format-conversion-failed, http-endpoint-failed}。</p> <p>若您在同一表達式內多次使用此值，每個執行個體的評估結果都是相同的錯誤字串。</p>	myPrefix/ result={! firehose: error-out put-type} /!{timest amp:yyyy/ MM/dd}	myPrefix/ result=pr ocessing- failed/20 18/08/03	error-output-type 值僅能用於 ErrorOutputPrefix 欄位中。
random-string	<p>評估結果為 11 個字元的隨機字串。若您在同一表達式內多次使用此值，每個執</p>	myPrefix/ !{firehos e:random- string}/	myPrefix/ 046b6c7f- 0b/	<p>可用於這兩種字首類型。</p> <p>您可將其放在格式字串的開頭，</p>

Conversion (轉換)	Description	範例輸入	範例輸出	備註
	行個體的評估結果為新的隨機字串。			以取得 Amazon S3 達到極高輸送量有時需要的隨機字首。

partitionKeyFromLambda 和 partitionKeyFromQuery 命名空間

對於[動態分割](#)，您必須在 S3 儲存貯體字首中使用下列運算式格式：`!{namespace:value}`，其中命名空間可以是 `partitionKeyFromQuery` 或 `partitionKeyFromLambda`，也可以是兩者。如果您使用內嵌剖析來建立來源資料的分割索引鍵，則必須指定由以下格式指定之運算式所組成的 S3 儲存貯體字首值：`"partitionKeyFromQuery:keyID"`。如果您使用 AWS Lambda 函數為來源資料建立分割索引鍵，則必須指定由以下格式指定之運算式所組成的 S3 儲存貯體字首值：`"partitionKeyFromLambda:keyID"`。如需詳細資訊，請參閱建立 Amazon Firehose 串流中的「為您的目的地選擇 Amazon S3」。 [???](#)

語義規則

下列規則適用於 Prefix 及 ErrorOutputPrefix 表達式。

- 以 timestamp 命名空間而言，不在單引號內的任何字元都將納入評估。換言之，值欄位中任何以單引號逸出的字串都將依照字面意思處理。
- 如果您指定的字首不包含時間戳記命名空間表達式，Firehose 會將表達式附加 `{timestamp:yyyy/MM/dd/HH/}` 到 Prefix 欄位中的值。
- 序列 `!{` 僅出現於 `!{namespace:value}` 表達式。
- 僅在 Prefix 不具表達式時，ErrorOutputPrefix 才可為零。在此情況下，Prefix 評估為 `<specified-prefix>yyyy/MM/DDD/HH/`，而 ErrorOutputPrefix 評估為 `<specified-prefix><error-output-type>yyyy/MM/DDD/HH/`。DDD 代表某年某日。
- 若您指定 ErrorOutputPrefix 的表達式，務必納入至少一個 `!{firehose:error-output-type}` 執行個體。
- Prefix 無法納入 `!{firehose:error-output-type}`。
- Prefix 或 ErrorOutputPrefix 評估後都不能超過 512 個字元。
- 若目的地為 Amazon Redshift，Prefix 必定不能具備運算式，ErrorOutputPrefix 必須為零。

- 當目的地為 Amazon OpenSearch Service 或 Splunk 且未指定 `ErrorOutputPrefix` 任何時，Firehose 會使用 `Prefix` 欄位來記錄失敗。
- 當目的地為 Amazon S3 時，Amazon S3 目的地組態中的 `Prefix` 和 `ErrorOutputPrefix` 會分別用於成功的記錄和失敗的記錄。如果您使用 AWS CLI 或 API，則可以透過其自己的 `Prefix` 和 `ErrorOutputPrefix` 使用 `ExtendedS3DestinationConfiguration` 來指定 Amazon S3 備份組態。
- 當您使用 AWS 管理主控台 並將目的地設定為 Amazon S3 時，Firehose 會在目的地組態 `ErrorOutputPrefix` 中分別使用 `Prefix` 和 來成功記錄和失敗記錄。如果您使用表達式指定字首，則必須指定包含的錯誤字首 `!{firehose:error-output-type}`。
- 當您 `ExtendedS3DestinationConfiguration` 搭配 AWS CLI、API 或 使用 CloudFormation 時，如果您指定 `S3BackupConfiguration`，Firehose 不會提供預設的 `ErrorOutputPrefix`。
- 建立 `ErrorOutputPrefix` 運算式時，您無法使用 `partitionKeyFromLambda` 和 `partitionKeyFromQuery` 命名空間。

範例字首

Prefix 及 ErrorOutputPrefix 範例

Input	評估的字首 (於 2018 年 8 月 27 日上午 10:30 UTC)
Prefix : 未指定	Prefix: 2018/08/27/10
ErrorOutputPrefix : myFirehoseFailures/!{firehose:error-output-type}/	ErrorOutputPrefix : myFirehoseFailures/processing-failed/
Prefix: !{timestamp:yyyy/MM/dd}	無效輸入 : Prefix 具有表達式時，ErrorOutputPrefix 不可為零
ErrorOutputPrefix : 未指定	
Prefix: myFirehose/DeliveredYear=!{timestamp:yyyy}/anyMonth/rand=!{firehose:random-string}	Prefix: myFirehose/DeliveredYear=2018/anyMonth/rand=5abf82daaa5
ErrorOutputPrefix : myFirehoseFailures/!{firehose:error-	ErrorOutputPrefix : myFirehoseFailures/processing-failed/2018/anyMonth/10

Input	評估的字首 (於 2018 年 8 月 27 日上午 10:30 UTC)
output-type}/{!{timestamp:yyyy}/anyMonth/{!{timestamp:dd}	
Prefix: myPrefix/year={!{timestamp:yyyy}/month={!{timestamp:MM}/day={!{timestamp:dd}/hour={!{timestamp:HH}/	Prefix: myPrefix/year=2018/month=07/day=06/hour=23/
ErrorOutputPrefix : myErrorPrefix/year={!{timestamp:yyyy}/month={!{timestamp:MM}/day={!{timestamp:dd}/hour={!{timestamp:HH}/!{firehose:error-output-type}	ErrorOutputPrefix : myErrorPrefix/year=2018/month=07/day=06/hour=23/processing-failed
Prefix: myFirehosePrefix/	Prefix: myFirehosePrefix/2
ErrorOutputPrefix : 未指定	018/08/27/
	ErrorOutputPrefix : myFirehosePrefix/processing-failed/2
	018/08/27/

設定 OpenSearch Service 的索引輪換

您能夠針對 OpenSearch Service 目的地指定一個以時間為基礎的索引輪換選項，其中包括下列五種選擇：NoRotation、OneHour、OneDay、OneWeek 或 OneMonth。

根據您選擇的輪換選項，Amazon Data Firehose 會將 UTC 到達時間戳記的一部分附加至您指定的索引名稱。並依此輪換該時間戳記。下方範例是 OpenSearch Service 中每個索引輪換選項產生的索引名稱，其中指定的索引名稱為 myindex，抵達時間戳記則是 2016-02-25T13:00:00Z。

RotationPeriod	IndexName
NoRotation	myindex
OneHour	myindex-2016-02-25-13

RotationPeriod	IndexName
OneDay	myindex-2016-02-25
OneWeek	myindex-2016-w08
OneMonth	myindex-2016-02

Note

透過 OneWeek 選項，Data Firehose 會使用 <YEAR>-w<WEEK NUMBER> 格式自動建立索引 (例如，2020-w33)，其中週數是使用 UTC 時間計算，並根據下列美國慣例來計算：

- 一週的開始日為週日
- 一年中的第一週是指當年包含週六的第一週

暫停和繼續資料交付

設定 Firehose 串流後，串流來源中可用的資料會持續交付至目的地。如果您遇到串流目的地暫時無法使用的情況 (例如，在規劃的維護操作期間)，您可能會想要暫停資料交付，並在目的地再次可用時繼續執行。

Important

當您使用下述方法來暫停和繼續串流時，在繼續串流之後，您會看到很少記錄會交付到 Amazon S3 中的錯誤儲存貯體，而其餘的串流會繼續交付到目的地。這是方法的已知限制，這是因為在多次重試被追蹤為失敗之後，少量記錄先前無法交付到目的地。

暫停 Firehose 串流

若要暫停 Firehose 中的串流交付，請先移除 Firehose 寫入失敗交付之 S3 備份位置的許可。例如，如果您想要使用 OpenSearch 目的地暫停 Firehose 串流，您可以透過更新許可來執行此操作。如需詳細資訊，請參閱[授予 Firehose 存取公有 OpenSearch Service 目的地](#)。

移除動作 s3:PutObject 的 "Effect": "Allow" 許可，並明確新增陳述式，該陳述式會針對用於備份失敗交付的 S3 儲存貯體的動作 s3:PutObject 套用 "Effect": "Deny" 許可。接著，關閉串

流目的地（例如，關閉目的地 OpenSearch 網域），或移除 Firehose 寫入目的地的許可。若要更新其他目的地的許可，請在[使用 Amazon Data Firehose 控制存取](#)中檢查目的地的區段。完成這兩個動作後，Firehose 將停止交付串流，而且您可以使用 [Firehose 的 CloudWatch 指標](#)來監控此動作。

Important

當您在 Firehose 中暫停串流交付時，您需要確保串流的來源（例如 Kinesis Data Streams 或 Managed Service for Kafka）已設定為保留資料，直到串流交付恢復且資料交付至目的地為止。如果來源是 DirectPUT，Firehose 會保留資料 24 小時。如果您未繼續串流，並在資料保留期限到期之前交付資料，則可能會發生資料遺失。

繼續 Firehose 串流

若要繼續交付，請先開啟目的地，並確保 Firehose 具有將串流交付至目的地的許可，將先前所做的變更還原至串流目的地。接下來，還原先前套用至 S3 儲存貯體的許可變更，以備份失敗的交付。也就是說，套用動作 `s3:PutObject` 的 `"Effect": "Allow"` 許可，並對用於備份失敗交付的 S3 儲存貯體的動作 `s3:PutObject` 移除 `"Effect": "Deny"` 許可。最後，使用 [Firehose 的 CloudWatch 指標](#)進行監控，以確認串流已交付至目的地。若要檢視和疑難排解錯誤，請使用 [Firehose 的 Amazon CloudWatch Logs 監控](#)。

使用 Amazon Data Firehose 將資料交付至 Apache Iceberg 資料表

Apache Iceberg 是一種高效能的開放原始碼資料表格式，用於執行大數據分析。Apache Iceberg 將 SQL 資料表的可靠性和簡單性帶入 Amazon S3 資料湖，並讓 Spark、Flink、Trino、Hive 和 Impala 等開放原始碼分析引擎能夠同時使用相同的資料。如需詳細資訊，請參閱 [Apache Iceberg](#) 和 [考量和限制](#)。

您可以使用 Firehose 將串流資料交付至 Amazon S3 中的 Apache Iceberg 資料表。您的 Apache Iceberg 資料表可以在 Amazon S3 中自我管理，或在 Amazon S3 資料表中託管。在自我管理的 Iceberg 資料表中，您可以管理所有資料表最佳化，例如壓縮和快照過期。Amazon S3 Tables 提供針對大規模分析工作負載最佳化的儲存體，其功能可持續改善查詢效能並降低表格式資料的儲存成本。如需 Amazon S3 資料表的詳細資訊，請參閱 [Amazon S3 資料表](#)。

此功能可讓您將記錄從單一串流路由到不同的 Apache Iceberg 資料表。您可以將插入、更新和刪除操作自動套用至這些資料表中的記錄。它還支援對 Amazon S3 中的 Apache Iceberg 資料表進行精細的資料存取控制 AWS Lake Formation。您可以在 [中](#)集中指定存取控制 AWS Lake Formation，並為 Firehose 提供更精細的資料表層級和資料欄層級許可。

考量和限制

Note

Firehose 支援 Apache Iceberg Tables 作為除中國區域、亞太區域（台北）AWS GovCloud (US) Regions、亞太區域（馬來西亞）、亞太區域（紐西蘭）和墨西哥（中部）[AWS 區域](#)以外所有的目的地。

Firehose 支援 Apache Iceberg 資料表有下列考量和限制。

- 輸送量 – 如果您使用 Direct PUT 作為將資料交付至 Apache Iceberg 資料表的來源，則每個串流的最大輸送量在美國東部（維吉尼亞北部）、美國西部（奧勒岡）和歐洲（愛爾蘭）區域為 5 MiB/秒，所有其他區域為 1 MiB/秒 AWS 區域。如果您想要在沒有更新和刪除的情況下將資料插入 Iceberg 資料表，並且想要提高串流的輸送量，則可以使用 [Firehose 限制表單](#) 請求提高輸送量限制。

True 如果您只想要插入資料而不執行更新和刪除，也可以將 AppendOnly 旗標設定為 True。透過將 AppendOnly 旗標設定為 True，Firehose 會自動擴展以符合您的輸送量。目前，您只能使用 [CreateDeliveryStream](#) API 操作來設定此旗標。

如果直接 PUT 串流因為較高的資料擷取磁碟區超過 Firehose 串流的輸送量容量而遇到限流，則 Firehose 會自動增加串流的輸送量限制，直到包含限流為止。根據增加的輸送量和限流，Firehose 可能需要更長的時間，才能將串流的輸送量增加到所需的層級。因此，請繼續重試失敗的資料擷取記錄。如果您預期資料磁碟區在突然的大爆量中增加，或如果您的新串流需要高於預設輸送量限制的輸送量，請請求增加輸送量限制。

- S3 每秒交易數 (TPS) – 若要最佳化 S3 效能，如果您使用 Kinesis Data Streams 或 Amazon MSK 做為來源，建議您使用適當的分割區索引鍵來分割來源記錄。如此一來，路由至相同 Iceberg 資料表的資料記錄會映射至一或多個來源分割區，稱為碎片。如果可能，請將屬於不同目標 Iceberg 資料表的資料記錄分散到不同的分割區/碎片，以便您可以使用來源主題/串流的所有分割區/碎片中可用的所有彙總輸送量。
- 資料欄 – 對於資料欄名稱和值，Firehose 只會接受多層巢狀 JSON 中第一個層級的節點。例如，Firehose 會選取第一個層級中可用的節點，包括位置欄位。來源資料的欄名稱和資料類型必須與 Firehose 目標資料表的資料類型完全相符，才能成功交付。在此情況下，Firehose 預期您在 Iceberg 資料表中有結構或映射資料類型資料欄，以符合位置欄位。Firehose 支援 16 個巢狀層級。以下是巢狀 JSON 的範例。

```
{
  "version": "2016-04-01",
  "deviceId": "<solution_unique_device_id>",
  "sensorId": "<device_sensor_id>",
  "timestamp": "2024-01-11T20:42:45.000Z",
  "value": "<actual_value>",
  "position": {
    "x": 143.595901,
    "y": 476.399628,
    "z": 0.24234876
  }
}
```

如果資料欄名稱或資料類型不相符，則 Firehose 會擲回錯誤，並將資料傳送到 S3 錯誤儲存貯體。如果 Apache Iceberg 資料表中的所有資料欄名稱和資料類型相符，但來源記錄中存在其他欄位，Firehose 會略過新欄位。

- 每筆記錄一個 JSON 物件 – 一個 Firehose 記錄中只能傳送一個 JSON 物件。如果您彙總並傳送記錄內的多個 JSON 物件，Firehose 會擲回錯誤，並將資料交付至 S3 錯誤儲存貯體。如果您使用 [KPL](#) 彙總記錄，並使用 Amazon Kinesis Data Streams 將資料擷取至 Firehose 做為來源，則 Firehose 會自動取消彙總每個記錄並使用一個 JSON 物件。
- 壓縮和儲存最佳化 – 每次使用 Firehose 寫入 Iceberg Tables 時，都會遞交和產生快照、資料檔案和刪除檔案。擁有許多資料檔案會增加中繼資料額外負荷，並影響讀取效能。為了獲得有效率的查詢效能，建議您考慮一個解決方案，該解決方案會定期擷取小型資料檔案，並將其重寫為較小的大型資料檔案。此程序稱為 compaction。AWS Glue Data Catalog 支援自動壓縮 Apache Iceberg Tables。如需詳細資訊，請參閱《AWS Glue 使用者指南》中的 [壓縮管理](#)。如需詳細資訊，請參閱 [自動壓縮 Apache Iceberg 資料表](#)。或者，您可以執行 Athena Optimize 命令來手動執行壓縮。如需 Optimize 命令的詳細資訊，請參閱 [Athena Optimize](#)。

除了壓縮資料檔案之外，您還可以使用對 Apache Iceberg 資料表執行資料表維護的 [VACUUM](#) 陳述式來最佳化儲存體使用量，例如快照過期和孤立檔案移除。或者，您也可以使用 AWS Glue Data Catalog 來支援 Apache Iceberg 資料表的受管資料表最佳化，方法是自動移除資料檔案、孤立檔案，並使不再需要的快照過期。如需詳細資訊，請參閱有關 [Apache Iceberg Tables 儲存最佳化的](#) 部落格文章。

- 我們不支援 Apache Iceberg Tables 的 Amazon MSK Serverless 來源做為目的地。
- 對於更新操作，Firehose 會放置刪除檔案，後面接著插入操作。放置刪除檔案會產生 Amazon S3 費用。
- Firehose 不建議使用多個 Firehose 串流將資料寫入相同的 Apache Iceberg 資料表。這是因為 Apache Iceberg 依賴 [樂觀並行控制 \(OCC\)](#)。如果多個 Firehose 串流嘗試同時寫入單一 Iceberg 資料表，則只有一個串流成功在指定時間遞交資料。其他串流無法遞交退避，並重試遞交操作，直到設定的重試持續時間過期為止。耗盡重試持續時間後，資料和刪除檔案金鑰 (Amazon S3 路徑) 會傳送至設定的 Amazon S3 錯誤字首。
- Firehose 支援的目前 Iceberg Library 版本是 1.5.2 版。
- 若要將加密資料交付至 Amazon S3 Tables，您應該在 Amazon S3 Tables 中設定 AWS Key Management Service 參數，而不是在 Firehose 組態中設定參數。如果您在 Firehose 中設定 AWS Key Management Service 參數以將加密的資料交付至 Amazon S3 Tables，則 Firehose 無法使用這些參數進行加密。如需詳細資訊，請參閱 [搭配 AWS KMS 金鑰使用伺服器端加密](#)。
- Firehose 串流僅支援交付至透過 Iceberg 的 GlueCatalog API 建立的資料庫和資料表。不支援交付至透過 Glue SDK 建立的資料庫和資料表。請注意，連字號 (-) 不是資料庫和 Iceberg 程式庫中的資料表名稱支援的字元。如需詳細資訊，請參閱 Iceberg 程式庫支援的 [Glue Database Regex](#) 和 [Glue Table Regex](#)。

- Firehose 寫入的所有檔案都是使用記錄中存在的分割區來計算。這也適用於已刪除的檔案。不支援全域刪除，例如寫入分割資料表的未分割刪除檔案。

使用 Apache Iceberg 資料表做為目的地的先決條件

從下列選項中選擇，以完成必要的先決條件。

主題

- [在 Amazon S3 中交付至 Iceberg 資料表的先決條件](#)
- [交付至 Amazon S3 資料表的先決條件](#)

在 Amazon S3 中交付至 Iceberg 資料表的先決條件

開始之前，請先完成下列先決條件。

- 建立 Amazon S3 儲存貯體 – 您必須建立 Amazon S3 儲存貯體，以在建立資料表期間新增中繼資料檔案路徑。如需詳細資訊，請參閱[建立 S3 儲存貯體](#)。
- 建立具有必要許可的 IAM 角色 – Firehose 需要具有特定許可的 IAM 角色，才能存取 AWS Glue 資料表並將資料寫入 Amazon S3。相同角色用於授予對 Amazon S3 儲存貯體的 AWS Glue 存取權。當您建立 Iceberg 資料表和 Firehose 串流時，需要此 IAM 角色。如需詳細資訊，請參閱[授予 Firehose 對 Amazon S3 資料表的存取權](#)。
- 建立 Apache Iceberg 資料表 – 如果您要在 Firehose 串流中設定唯一金鑰以進行更新和刪除，Firehose 會驗證資料表和唯一金鑰是否存在做為串流建立的一部分。在此案例中，您必須先建立資料表，才能建立 Firehose 串流。您可以使用 AWS Glue 建立 Apache Iceberg 資料表。如需詳細資訊，請參閱[建立 Apache Iceberg 資料表](#)。如果您未在 Firehose 串流中設定唯一金鑰，則不需要在建立 Firehose 串流之前建立 Iceberg 資料表。

Note

Firehose 支援 Apache Iceberg 資料表的下列資料表版本和格式。

- 資料表格式版本 – Firehose 僅支援 [V2 資料表格式](#)。請勿以 V1 格式建立資料表，否則您會收到錯誤，而資料會改為傳送到 S3 錯誤儲存貯體。
- 資料儲存格式 – Firehose 以 Parquet 格式將資料寫入 Apache Iceberg 資料表。
- 資料列層級操作 – Firehose 支援將資料寫入 Apache Iceberg Tables Merge-on-Read (MOR) 模式。

交付至 Amazon S3 資料表的先決條件

若要將資料交付至 Amazon S3 資料表儲存貯體，請完成下列先決條件。

- 建立 S3 Table 儲存貯體、命名空間、資料表儲存貯體中的資料表，以及 [Amazon S3 Tables 入門](#) 中概述的其他整合步驟。資料欄名稱必須小寫，因為 S3 Tables 目錄整合所施加的限制，如 [S3 Tables 目錄整合限制](#) 中所指定。
- 建立具有必要許可的 IAM 角色 – Firehose 需要具有特定許可的 IAM 角色，才能存取 AWS Glue 資料表並將資料寫入 Amazon S3 資料表儲存貯體中的資料表。若要寫入 Amazon S3 資料表儲存貯體中的資料表，您還必須提供 IAM 角色所需的許可。Amazon S3 Tables 目錄所需的許可取決於您使用的存取控制模式：
 - IAM 存取控制 – Firehose 交付角色需要直接在 Amazon S3 Tables 資源上的 IAM 許可。
 - Lake Formation 存取控制 – Firehose 交付角色需要 AWS Lake Formation 許可才能管理對資料表資源的存取。AWS Lake Formation 會使用自己的許可模型，為 Data Catalog 資源啟用精細存取控制。

您可以在建立 Firehose 串流時設定此 IAM 角色。如需詳細資訊，請參閱 [授予 Firehose 對 Amazon S3 Tables 的存取權](#)。

如需 step-by-step 整合，請參閱部落格 [建置資料湖以使用 Amazon S3 Tables 和 Amazon Data Firehose 串流資料](#)。如需詳細資訊，請參閱 [搭配使用 Amazon S3 Tables 與 AWS 分析服務](#)。

設定 Firehose 串流

若要以 Apache Iceberg Tables 做為目的地建立 Firehose 串流，您必須設定下列項目。

Note

用於交付至 S3 資料表儲存貯體中資料表的 Firehose 串流設定與 Amazon S3 中的 Apache Iceberg 資料表相同。

設定來源和目的地

若要將資料交付至 Apache Iceberg Tables，請選擇串流的來源。

若要設定串流的來源，請參閱 [設定來源設定](#)。

接著，選擇 Apache Iceberg Tables 做為目的地，並提供 Firehose 串流名稱。

設定資料轉換

若要對資料執行自訂轉換，例如新增或修改傳入串流中的記錄，您可以將 Lambda 函數新增至 Firehose 串流。如需在 Firehose 串流中使用 Lambda 進行資料轉換的詳細資訊，請參閱 [轉換 Amazon Data Firehose 中的來源資料](#)。

對於 Apache Iceberg 資料表，您必須指定如何將傳入記錄路由到不同的目的地資料表，以及您要執行的操作。向 Firehose 提供所需路由資訊的其中一種方法是使用 Lambda 函數。

如需詳細資訊，請參閱將 [記錄路由到不同的 Iceberg 資料表](#)。

連接資料目錄

Apache Iceberg 需要資料目錄才能寫入 Apache Iceberg Tables。Firehose 與 AWS Glue Data Catalog for Apache Iceberg Tables 整合。

您可以在與 Firehose 串流 AWS Glue Data Catalog 相同的帳戶中，或在跨帳戶和與 Firehose 串流相同的區域中（預設），或在不同的區域中使用。

如果您要將交付至 Amazon S3 資料表，而且您正在使用主控台來設定 Firehose 串流，請選取對應至 Amazon S3 資料表目錄的目錄。如果您使用 CLI 來設定 Firehose 串流，請在 CatalogConfiguration 輸入中使用格式 `arn:aws:glue:<region>:<account-id>:catalog/s3tablescatalog/<s3 table bucket name>`。如需詳細資訊，請參閱 [設定 Firehose 串流至 Amazon S3 資料表](#)。

Note

Firehose 支援 Iceberg 資料表的三個操作：插入、更新和刪除。如果沒有指定的操作，Firehose 預設會插入、將每個傳入記錄新增為新資料列，以及保留重複項目。若要改為修改現有的記錄，請指定「更新」操作，該操作使用主索引鍵來尋找和變更現有的資料列。

範例：

- 預設（插入）：多個相同的客戶記錄會建立重複的資料列。
- 指定的更新：新客戶地址會更新現有記錄。

設定 JQ 表達式

對於 Apache Iceberg 資料表，您必須指定如何將傳入記錄路由到不同的目的地資料表，以及要執行的操作，例如插入、更新和刪除。您可以透過設定 Firehose 的 JQ 表達式來剖析並取得必要資訊來執行此操作。如需詳細資訊，請參閱[???](#)。

設定唯一金鑰

具有多個資料表的更新和刪除 – 唯一索引鍵是來源記錄中的一或多個欄位，可唯一識別 Apache Iceberg 資料表中的資料列。如果您只插入具有多個資料表的案例，則不需要設定唯一金鑰。如果您想要對特定資料表執行更新和刪除，則必須為這些必要資料表設定唯一金鑰。請注意，如果資料表中的資料列遺失，更新會自動插入資料列。如果您只有單一資料表，則可以設定唯一金鑰。對於更新操作，Firehose 會先放置刪除檔案，再放置插入。

您可以將每個資料表的唯一索引鍵設定為 Firehose 串流建立的一部分，也可以在[建立資料表](#)或[修改資料表](#)操作期間，在 Iceberg 中原生設定 `identifier-field-ids`。在建立串流期間為每個資料表設定唯一索引鍵是選用的。如果您在建立串流期間未為每個資料表設定唯一索引鍵，Firehose 會檢查 `identifier-field-ids` 所需的資料表，並將其用作唯一索引鍵。如果未設定兩者，則具有更新和刪除操作的資料交付會失敗。

若要設定本節，請提供您要更新或刪除資料的資料表的資料庫名稱、資料表名稱和唯一索引鍵。組態中每個資料表只能有項目。您不需要為僅附加的案例設定本節。如果資料表中的資料無法交付，您也可以選擇提供錯誤儲存貯體字首，如下列範例所示。

```
[
  {
    "DestinationDatabaseName": "MySampleDatabase",
    "DestinationTableName": "MySampleTable",
    "UniqueKeys": [
      "COLUMN_PLACEHOLDER"
    ],
    "S3ErrorOutputPrefix": "OPTIONAL_PREFIX_PLACEHOLDER"
  }
]
```

如果提供的欄名稱在整個資料表中是唯一的，Firehose 支援唯一索引鍵的組態。不過，它不支援將完整資料欄名稱做為唯一索引鍵。例如，如果資料欄名稱 `_id` 也出現在最上層，則名為 `top._id` 的金鑰不會被視為唯一金鑰。如果在整個資料表中 `_id` 是唯一的，則會使用它，無論其在資料表結構中的位置為何。這是最上層資料欄或巢狀資料欄。在下列範例中，`_id` 是結構描述的有效唯一金鑰，因為資料欄名稱在結構描述中是唯一的。

```
[
  "schema": {
    "type": "struct",
    "fields": [
      {
        "name": "top",
        "type": {
          "type": "struct",
          "fields": [
            { "name": "_id", "type": "string" },
            { "name": "name", "type": "string" }
          ]
        }
      },
      { "name": "user", "type": "string" }
    ]
  }
]
```

在下列範例中，`_id`不是結構描述的有效唯一索引鍵，因為它在頂層資料欄和巢狀結構中使用。

```
[
  "schema": {
    "type": "struct",
    "fields": [
      {
        "name": "top",
        "type": {
          "type": "struct",
          "fields": [
            { "name": "_id", "type": "string" },
            { "name": "name", "type": "string" }
          ]
        }
      },
      { "name": "_id", "type": "string" }
    ]
  }
]
```

指定重試持續時間

您可以使用此組態來指定 Firehose 在寫入 Amazon S3 中的 Apache Iceberg Tables 時，應嘗試重試的持續時間，以秒為單位。您可以設定 0 到 7200 秒的任何值來執行重試。根據預設，Firehose 會重試 300 秒。

處理失敗的交付或處理

您必須設定 Firehose 將記錄交付至 S3 備份儲存貯體，以防其在重試持續時間到期後處理或交付串流失敗。為此，請從主控台的主控台設定 S3 備份儲存貯體和 S3 備份儲存貯體錯誤輸出字首。

錯誤處理

Firehose 將所有交付錯誤傳送到 CloudWatch Logs 和 Amazon S3 錯誤儲存貯體。

錯誤清單：

錯誤訊息	Description
Iceberg.NoSuchTable	Firehose 正在寫入不存在的資料表，或資料表不是 V2 格式。Firehose 不支援 V1 格式的資料表。
Iceberg.InvalidTableName	傳遞 null 或空白資料表名稱，或資料表不是 V2 格式。Firehose 不支援 V1 格式的資料表。
S3.AccessDenied	確定在先決條件步驟中建立的 IAM 角色具有必要的許可和信任政策。
Glue.AccessDenied	確定在先決條件步驟中建立的 IAM 角色具有必要的許可和信任政策。

設定緩衝提示

Firehose 會將記憶體中的傳入串流資料緩衝至特定大小（緩衝大小）和一段特定時間（緩衝間隔），再交付至 Apache Iceberg Tables。您可以選擇 1–128 MiBs 緩衝區大小，以及 0–900 秒的緩衝區間隔。較高的緩衝提示會導致較低的 S3 寫入次數、因較大的資料檔案而降低壓縮成本，以及更快的查詢執行時間，但延遲較高。較低的緩衝區提示值會以較低的延遲提供資料。

配置進階設定

您可以設定 Apache Iceberg Tables 的伺服器端加密、錯誤記錄、許可和標籤。如需詳細資訊，請參閱[配置進階設定](#)。您必須新增在中建立的 IAM 角色[???](#)。Firehose 將擔任存取 AWS Glue 資料表和寫入 Amazon S3 儲存貯體的角色。

Firehose 串流建立可能需要幾分鐘的時間才能完成。成功建立 Firehose 串流後，您可以開始將資料擷取到其中，並且可以在 Apache Iceberg 資料表中檢視資料。

將傳入的記錄路由到單一 Iceberg 資料表

如果您希望 Firehose 將資料插入單一 Iceberg 資料表，只需在串流組態中設定單一資料庫和資料表，如下列範例 JSON 所示。對於單一資料表，您不需要 JQ 表達式和 Lambda 函數，即可將路由資訊提供給 Firehose。如果您將這些欄位與 JQ 或 Lambda 一起提供，則 Firehose 將從 JQ 或 Lambda 取得輸入。

```
[
  {
    "DestinationDatabaseName": "UserEvents",
    "DestinationTableName": "customer_id",
    "UniqueKeys": [
      "COLUMN_PLACEHOLDER"
    ],
    "S3ErrorOutputPrefix": "OPTIONAL_PREFIX_PLACEHOLDER"
  }
]
```

在此範例中，Firehose 將所有輸入記錄路由到 UserEvents 資料庫中的 customer_id 資料表。如果您想要在單一資料表上執行更新或刪除操作，則必須使用 [JSONQuery 方法](#) 或 [Lambda 方法](#)，將每個傳入記錄的操作提供給 Firehose。

將傳入的記錄路由到不同的 Iceberg 資料表

Amazon Data Firehose 可以根據記錄的內容，將串流中的傳入記錄路由到不同的 Iceberg 資料表。從 Amazon Data Firehose 交付時，記錄不會依序保留。請考慮以下範例輸入記錄。

```
{
  "deviceId": "Device1234",
  "timestamp": "2024-11-28T11:30:00Z",
  "data": {
```

```
"temperature": 21.5,
"location": {
  "latitude": 37.3324,
  "longitude": -122.0311
},
"powerlevel": 84,
"status": "online"
}
```

```
{
  "deviceId": "Device4567",
  "timestamp": "2023-11-28T10:40:00Z",
  "data": {
    "pressure": 1012.4,
    "location": {
      "zipcode": 24567
    }
  },
  "powerlevel": 82,
  "status": "online"
}
```

在此範例中，**deviceId** 欄位有兩個可能的值 – Device1234和 Device4567。當傳入記錄**deviceId**的欄位為 Device1234，我們希望將記錄寫入名為 Iceberg 資料表Device1234，當傳入記錄**deviceId**的欄位為 Device4567，我們希望將記錄寫入名為 Device4567的資料表。

請注意，具有 Device1234和 Device4567的記錄可能會有一組不同的欄位，對應到對應 Iceberg 資料表中的不同資料欄。傳入的記錄可能有巢狀 JSON 結構，其中 **deviceId**可以在 JSON 記錄中巢狀化。在接下來的章節中，我們會討論如何透過提供適當的路由資訊給 Firehose，將記錄路由到不同的資料表。

使用 JSONQuery 表達式將路由資訊提供給 Firehose

提供記錄路由資訊給 Firehose 最簡單且最具成本效益的方式，就是提供 JSONQuery 表達式。使用此方法，您可以為三個參數提供 JSONQuery 表達式：Database Name、Table Name和 (選用) Operation。Firehose 會使用您提供的表達式，從傳入串流記錄中擷取資訊來路由記錄。

Database Name 參數指定目的地資料庫的名稱。Table Name 參數指定目的地資料表的名稱。Operation 是選用參數，指出是否要將傳入的串流記錄作為新記錄插入目的地資料表，或修改或刪除目的地資料表中的現有記錄。操作欄位必須具有下列其中一個值：insert、update或 delete。

對於這三個參數中的每個參數，您可以提供靜態值或動態表達式，其中值會從傳入記錄擷取。例如，如果您想要將所有傳入串流記錄交付至名為 `iot` 的單一資料庫 `IoTEvents`，資料庫名稱的靜態值為 `"IoTEvents"`。如果目的地資料表名稱必須從傳入記錄中的欄位取得，則資料表名稱是一種動態表達式，可指定傳入記錄中需要從中擷取目的地資料表名稱的欄位。

在下列範例中，我們使用資料庫名稱的靜態值、資料表名稱的動態值，以及操作的靜態值。請注意，指定操作是選用的。如果未指定任何操作，Firehose 預設會將傳入的記錄插入目的地資料表，做為新記錄。

```
Database Name : "IoTEvents"  
Table Name : .deviceId  
Operation : "insert"
```

如果 `deviceId` 欄位在 JSON 記錄中巢狀化，我們會將巢狀欄位資訊的資料表名稱指定為 `.event.deviceId`。

Note

- 當您將操作指定為 `update` 或 `delete`，您必須在設定 Firehose 串流時指定目的地資料表的唯一索引鍵，或在 Iceberg 中執行 [建立資料表](#) 或 [更改資料表](#) 操作時，在 Iceberg 中設定 [identifier-field-ids](#)。如果您無法指定，則 Firehose 會擲回錯誤，並將資料交付至 S3 錯誤儲存貯體。
- Database Name 和 Table Name 值必須與目的地資料庫和資料表名稱完全相符。如果不相符，則 Firehose 會擲回錯誤，並將資料交付至 S3 錯誤儲存貯體。

使用 AWS Lambda 函數提供路由資訊

在某些情況下，您可能有複雜的規則來決定如何將傳入的記錄路由到目的地資料表。例如，如果欄位包含值 `A`、`B` 或 `F`，則您可能有一個定義規則，該值應路由到名為 `TableX` 的目的地資料表，或者您可能想要透過新增其他屬性來擴增傳入的串流記錄。例如，如果記錄包含欄位 `device_id` 為 `1`，您可能想要新增另一個欄位 `device_type` 做為「`modem`」，並將其他欄位寫入目的地資料表欄。在這種情況下，您可以使用 Firehose 中的 AWS Lambda 函數轉換來源串流，並提供路由資訊作為 Lambda 轉換函數輸出的一部分。若要了解如何在 Firehose 中使用 AWS Lambda 函數轉換來源串流，請參閱在 [Amazon Data Firehose 中轉換來源資料](#)。

當您使用 Lambda 在 Firehose 中轉換來源串流時，輸出必須包含 `recordId`、`result` 和 `data` 或 `KafkaRecordValue` 參數。參數 `recordId` 包含輸入串流記錄、`result` 指出轉換是否成功，以及 `data` 包含 Lambda 函數的 Base64-encoded 轉換輸出。如需詳細資訊，請參閱 [???](#)。

```
{
  "recordId": "49655962066601463032522589543535113056108699331451682818000000",
  "result": "Ok",
  "data": "1IiwiI6ICJmYWxsIiwgImdgU21IiwiI6ICJmYWxsIiwg==tcHV0ZXIgdU2NpZW5jZSIzICJzZW1"
}
```

若要指定將串流記錄路由到目的地資料表做為 Lambda 函數一部分的路由資訊給 Firehose，Lambda 函數的輸出必須包含的額外區段 `metadata`。下列範例示範如何將中繼資料區段新增至使用 Kinesis Data Streams 做為資料來源的 Firehose 串流的 Lambda 輸出，以指示 Firehose 必須將記錄作為新記錄插入資料庫 `Device1234` 的資料表 `IoTevents`。

```
{
  "recordId": "49655962066601463032522589543535113056108699331451682818000000",
  "result": "Ok",
  "data":
  "1IiwiI6ICJmYWxsIiwgImdgU21IiwiI6ICJmYWxsIiwg==tcHV0ZXIgdU2NpZW5jZSIzICJzZW1",

  "metadata":{
  "otfMetadata":{
    "destinationTableName":"Device1234",
    "destinationDatabaseName":"IoTevents",
    "operation":"insert"
  }
}
}
```

同樣地，下列範例示範如何將中繼資料區段新增至使用 Amazon Managed Streaming for Apache Kafka 作為資料來源的 Firehose 的 Lambda 輸出，以指示 Firehose 必須將記錄作為新記錄插入資料庫中名為 `Device1234` 的資料表 `IoTevents`。

```
{
  "recordId": "49655962066601463032522589543535113056108699331451682818000000",
  "result": "Ok",
  "kafkaRecordValue":
  "1IiwiI6ICJmYWxsIiwgImdgU21IiwiI6ICJmYWxsIiwg==tcHV0ZXIgdU2NpZW5jZSIzICJzZW1",

  "metadata":{
```

```
"otfMetadata":{
  "destinationTableName":"Device1234",
  "destinationDatabaseName":"IoTevents",
  "operation":"insert"
}
```

在此範例中，

- `destinationDatabaseName` 是指目標資料庫的名稱，並且是必要欄位。
- `destinationTableName` 是指目標資料表的名稱，並且是必要欄位。
- `operation` 是選用欄位，可能的值為 `insert`、`update` 和 `delete`。如果您未指定任何值，則預設操作為 `insert`。

Note

- 當您將操作指定為 `update` 或 `delete`，您必須在設定 Firehose 串流時指定目的地資料表的唯一索引鍵，或在 Iceberg 中執行 [建立資料表](#) 或 [變更資料表](#) 操作時，在 Iceberg 中設定 [identifier-field-ids](#)。如果您無法指定，則 Firehose 會擲回錯誤，並將資料交付至 S3 錯誤儲存貯體。
- Database Name 和 Table Name 值必須與目的地資料庫和資料表名稱完全相符。如果不相符，則 Firehose 會擲回錯誤，並將資料交付至 S3 錯誤儲存貯體。
- 當您的 Firehose 串流同時具有 Lambda 轉換函數和 JSONQuery 表達式時，Firehose 會先檢查 Lambda 輸出中的中繼資料欄位，以判斷如何將記錄路由到適當的目的地資料表，然後查看 JSONQuery 表達式的輸出是否有遺漏的欄位。

如果 Lambda 或 JSONQuery 表達式不提供必要的路由資訊，則 Firehose 會假設這是單一資料表案例，並在唯一金鑰組態中尋找單一資料表資訊。

如需詳細資訊，請參閱將 [傳入記錄路由到單一 Iceberg 資料表](#)。如果 Firehose 無法判斷路由資訊，並將記錄比對至指定的目的地資料表，則會將資料交付至您指定的 S3 錯誤儲存貯體。

Lambda 函數範例

此 Lambda 函數是範例 Python 程式碼，可剖析傳入的串流記錄並新增必要欄位，以指定資料應如何寫入特定資料表。您可以使用此範本程式碼來新增路由資訊的中繼資料區段。

```
import json
import base64

def lambda_handler(firehose_records_input, context):
    print("Received records for processing from DeliveryStream: " +
        firehose_records_input['deliveryStreamArn'])

    firehose_records_output = {}
    firehose_records_output['records'] = []

    for firehose_record_input in firehose_records_input['records']:

        # Get payload from Lambda input, it could be different with different sources
        if 'kafkaRecordValue' in firehose_record_input:
            payload_bytes =
base64.b64decode(firehose_record_input['kafkaRecordValue']).decode('utf-8')
        else
            payload_bytes =
base64.b64decode(firehose_record_input['data']).decode('utf-8')

        # perform data processing on customer payload bytes here

        # Create output with proper record ID, output data (may be different with
different sources), result, and metadata
        firehose_record_output = {}

        if 'kafkaRecordValue' in firehose_record_input:
            firehose_record_output['kafkaRecordValue'] =
base64.b64encode(payload_bytes.encode('utf-8'))
        else
            firehose_record_output['data'] =
base64.b64encode(payload_bytes.encode('utf-8'))

        firehose_record_output['recordId'] = firehose_record_input['recordId']
        firehose_record_output['result'] = 'Ok'
        firehose_record_output['metadata'] = {
```

```

    'otfMetadata': {
        'destinationDatabaseName': 'your_destination_database',
        'destinationTableName': 'your_destination_table',
        'operation': 'insert'
    }
}
firehose_records_output['records'].append(firehose_record_output)
return firehose_records_output

```

指標指標

為了將資料交付至 Apache Iceberg Tables，Firehose 會在串流層級發出下列 CloudWatch 指標。

指標	Description
DeliveryToIceberg.Bytes	在指定期間內交付至 Apache Iceberg 資料表的位元組數。 單位：位元組
DeliveryToIceberg.IncomingRowCount	Firehose 嘗試交付至 Apache Iceberg Tables 的記錄數目。 單位：Count
DeliveryToIceberg.SuccessfulRowCount	傳送至 Apache Iceberg 資料表的成功資料列數。 單位：Count
DeliveryToIceberg.FailedRowCount	傳送至 S3 備份儲存貯體的失敗資料列數。 單位：Count
DeliveryToIceberg.DataFreshness	Firehose 中最早記錄的存留期（從進入 Firehose 到現在）。任何早於此存留期的記錄都已交付至 Apache Iceberg Tables。 單位：秒
DeliveryToIceberg.Success	成功遞交至 Apache Iceberg 資料表的總和。
JQProcessing.Duration	執行 JQ 表達式所需的時間。

指標	Description
	單位：毫秒

了解支援的資料類型

Firehose 支援 Apache Iceberg 支援的所有基本和複雜資料類型。如需詳細資訊，請參閱[結構描述和資料類型](#)。以字串傳送二進位資料時，您必須使用 Firehose 支援的編碼類型 - Basic Base64、MIME Base64、URL 和檔案名稱 safe Base64 和 Hex。對於時間戳記資料類型，您必須一律以微秒傳送。

資料類型範例

下一節顯示不同資料類型的範例。

MapType

```
{
  "destination_column_0":
  {"WP5o0J0kuIQcDPcsvpJJygF1xza0Sq0wUlgTwuIeCEzgVneGxA":"P03ReF3auyDqbfonx9Cd8NTmcQnqnw7JuZ0CWwI
    "destination_column_1": "{\"destination_nested_column_0\\\": \\
    \\\"18:56:14.974\\\"\", \\\"destination_nested_column_1\\\": 241.86246}\\":
    \\\"M07kAvYdHvBh61F7RzfxEd39YQI33LnM2NbGS67D0FFsRUyUUujKT5VnK7Wtfz1mHNeIix6FAY9cYpwTdedgr9XnFwG0
    \",\"{\\\"destination_nested_column_0\\\": \\\"18:56:14.974\\
    \\\", \\\"destination_nested_column_1\\\": 562.56384}\\":
    \\\"9G1xhDCt95LxBo51HybBZihq0qf6EU8jrDu7NMpxtGB2dY6q6kXpvxIrFuMdqHCJKIZIcDikwggLniUm8kgE4d
    \",\"{\\\"destination_nested_column_0\\\": \\\"18:56:14.974\\
    \\\", \\\"destination_nested_column_1\\\": 496.03268}\\":
    \\\"keTJZYLNvLRB50DMKzEI6M0AM4mueyNnA1m2YVnYdDwyxUpPqkb72Q6LiX0B9s8gCjZ6trW6C1PFk9KNBIpxYsj5Tc5Xs
    \",\"{\\\"destination_nested_column_0\\\": \\\"18:56:14.974\\\", \\
    \\\"destination_nested_column_1\\\": 559.0878}\\":
    \\\"mG0ZET84BUF28E312UCIWgmyPyQFSU0DH9NAMAnF3LJEutbooZwcBt97PP5AhaopNvC8pQZ4mGXB9hmVmJUNmuj5Qanyx
    \",\"{\\\"destination_nested_column_0\\\": \\\"18:56:14.974\\
    \\\", \\\"destination_nested_column_1\\\": 106.845245}\\":
    \\\"aidoVYrzu8gcLRkVVUyTKCN9gqTUFYi8uJQsrXEFY11f9oo17JhAtg9QKG5BBu67Ngb95ENsNKQyCHNImsu5x4hMnmHU
    \\\"}"}
}
```

DecimalType

```
{
  "destination_column_0": 9455262425851.1342772,
```

```

"destination_column_1": "9455262425851.1342772",
"destination_column_2": 9455262425852
}

```

BinaryType (base64-default, base64-mime, base64-url-safe, hex)

```

{
  "destination_column_0": "AsYhnHD\Ra54hIT11daNV9g10jtWPEfopH
+PjgUKHYB6K7UcYi4K19b80wD4J\93x5tyh+0y
+k5cM1jVRlmfIkIuLx19ERBiPPLhf4+yoJ2k70VavPnYWmNLs1hLDHlfeEMIfVhrq0GzJMoA
+CBAWXfIuiG420JSQP5iAx5xFG\
m0fkM5zYothje80GX1tdthcCL6WYBiP0S1wXcE0uMerfwc1Ac9fT0Bz6RzdJ1HhUDjoAXg
+4cvly27F82XpuGMNwpUj98A0rgbh2MoU9yvsM9ZrjD0eGVg0ZP8Ky7Za4oE\oK8j
+qABF6XV712iA6pVtTNJFvX6Ey3ssNYvno+LYF5ZsySs2rB5AbVM73Rf0PqdS\c\
r3MEqoEq+nPx6eGam4WSA+0swztt7aLdr1X6yK7xJeIJ0rTlIDBo0ZUaw011ykY
\8Bvy+4byoPlmr4Z5yhN1z3ZT0kx7eDR6xMv+vDVSDbtItVazDwHgDy41r
\hQNeNedPKrozc8TY9k7wZre\6V2lCa3BmT8Uu9b9ydjR9z+fCSdG
+VRv35nz5kdqdKy8YIrynYs4e0cjh8jH3UwVYrYQcnWkBAFF7Xk9CoPvNl3ciHZtyiZ0aTGIj9r00xX\
W5dGe9\4YChs6LbD584kxLTxvHgS14vadaTGNKci3SvNmZnsz8ducxtNXF\Tv2DUub465hzgpaLPur3+MB
+kfdN2YXUfqB
+xJAgxThWfUe151nrH0EPow9lgS1p21rUBGznJAvPR11ExGIAuc7JYAoUrJUKx5Hf16PekPDhqt7+yJwCB8qxhTTryxo
+bjtai4ndRCGcuCaxT8Kk0cXsS37urd3YGSdMinZdMNVc646s25415qK6nBRlqqAY8+EYmcUIVB9XcNdke4zoUfhVQoruwidzDU\
\kFafoulo5DEoM0yaH1N2HCSxG5tZXNQocSZPaY8efZYMCpmDXsPAzkmGskYRDSu\3wUqR0a2tGK5\
pQY24v+Jq0U\jQ99GShlU283nZ85ot2ocbtMAGD\WsrSEh61nt9RaI3HfA7\HcH\
fgr9jsTtxDgZhabTBwwDwX0zjWgX1bCuTLKBN7byxg9ZvAVgqwPS4HERLer5T5UkKf74zn9Eq3HYH1Q5JpyDUx
+im7mte1sprf1+A24kksVU\MD9aP9N8\QDsQ13gkh0n5KwFMz3BC2Vw5gL
+gGNHFKDRL6wGIfhuYcx9LucolZ1yNy9Gbb3ioWSSufyFpyXqtdnDLPi5QS1SjPjM2KDyqcH1SmRLIhd9MNRUC73EAEm
+N05wxPzBRSjhCHZpf8SrYITWJ17K3XzG0fPFh2NgES3jMP9cvSX06yyICcep2HBYGbFf1ni89+Rw==",
  "destination_column_1": "AsYhnHD\Ra54hIT11daNV9g10jtWPEfopH
+PjgUKHYB6K7UcYi4K19b80wD4J\93x5tyh+0y+k5c\r
\nM1jVRlmfIkIuLx19ERBiPPLhf4+yoJ2k70VavPnYWmNLs1hLDHlfeEMIfVhrq0GzJMoA+CBAWXfI\r
\nuiG420JSQP5iAx5xFG\m0fkM5zYothje80GX1tdthcCL6WYBiP0S1wXcE0uMerfwc1Ac9fT0Bz6R\r
\nzdJ1HhUDjoAXg+4cvly27F82XpuGMNwpUj98A0rgbh2MoU9yvsM9ZrjD0eGVg0ZP8Ky7Za4oE\oK\r
\n8j+qABF6XV712iA6pVtTNJFvX6Ey3ssNYvno+LYF5ZsySs2rB5AbVM73Rf0PqdS\c\r3MEqoEq+
\r\nnPx6eGam4WSA+0swztt7aLdr1X6yK7xJeIJ0rTlIDBo0ZUaw011ykY\8Bvy+4byoPlmr4Z5yhN1z
\r\n3ZT0kx7eDR6xMv+vDVSDbtItVazDwHgDy41r\hQNeNedPKrozc8TY9k7wZre\6V2lCa3BmT8Uu9b
\r\n9ydjR9z+fCSdG+VRv35nz5kdqdKy8YIrynYs4e0cjh8jH3UwVYrYQcnWkBAFF7Xk9CoPvNl3ciHZ
\r\nntyZ0aTGIj9r00xX\W5dGe9\4YChs6LbD584kxLTxvHgS14vadaTGNKci3SvNmZnsz8ducxtNXF
\r\nTv2DUub465hzgpaLPur3+MB+kfdN2YXUfqB+xJAgxThWfUe151nrH0EPow9lgS1p21rUBGznJAvP
\r\nR11ExGIAuc7JYAoUrJUKx5Hf16PekPDhqt7+yJwCB8qxhTTryxo+bjtai4ndRCGcuCaxT8Kk0cXs\r
\nS37urd3YGSdMinZdMNVc646s25415qK6nBRlqqAY8+EYmcUIVB9XcNdke4zoUfhVQoruwidzDU\k\r
\nFafoulo5DEoM0yaH1N2HCSxG5tZXNQocSZPaY8efZYMCpmDXsPAzkmGskYRDSu\3wUqR0a2tGK5\r
\n\pQY24v+Jq0U\jQ99GShlU283nZ85ot2ocbtMAGD\WsrSEh61nt9RaI3HfA7\HcH\fgR9jsTtxDg

```

```
\r\nZhabTBwwDwX0zjWgX1bCuTLKBN7byxg9ZvAVgqwPS4HERLer5T5UkKf74zn9Eq3HYH1Q5JpyDUx+\r
\nim7mte1sprf1+A24kksVU\MD9aP9N8\QDsQ13gkh0n5KwFMz3BC2Vw5gL+gGNHFKDRL6wGIIfhuYc
\r\nx9LucolZ1yNy9Gbb3ioWSSufyFpyXqtndDLPI5QS1S5JpJm2KDyqcH1SmRLIhd9MNRUC73EAEm+N0\r
\n5wxPzBRSjhCHZpf8SrYITWJl7K3XzG0fPFh2NgES3jMP9cvSX06yyICcep2HBYGbFflni89+Rw==",
  "destination_column_2": "AsYhnHD_Ra54hITl1daNV9g10jtwPEfopH-
PjgUKHYB6K7UcYi4K19b80wD4J_93x5tyh-0y-k5cMljVRlmfIkIuLx19ERBiPPLhf4-
yoJ2k70VavPnYwmNLs1hLDHlfeEMIfVhrq0GzJMoA-
CBAWXfIuiG420JSQP5iAx5xFG_m0fkM5zYothje80GXltdthcCL6WYBiP0S1wXcE0uMeRfwclAc9fT0Bz6RzdJlHhUDjoAX
qABF6XV712iA6pVtTNJFvX6Ey3ssNYvno-LYF5ZsySs2rB5AbVM73RfOPqdS_c_r3MEqoEq-
nPx6eGam4WSA-0swztt7aLdr1X6yK7xJeIJ0rTlIDBo0ZUaw011ykY_8Bvy-4byoPlmr4Z5yhN1z3ZT0kx7eDR6xMv-
vDVsDBtItVazDwHgDy41r_hQNeNedPKrozc8TY9k7wZre_6V21Ca3BmT8Uu9b9ydjR9z-fCSdG-
VRv35nz5kdqdKy8YIrynYs4e0cjh8jH3UwVYrYQcnWkBAFF7Xk9CoPVnL3ciHZtyiZ0aTGIj9r00xX_W5dGe9_4YChs6LbD
MB-kfdN2YXUfq-
xJAgxThWfUe151nrH0EPow9lgSlp21rUBGznJAvPRl1ExGIAuc7JYAoUrJUkx5Hf16PekPDhqt7-
yJwCB8qxhTTryxo-bjtai4ndRCGcuCaxT8Kk0cXsS37urd3YGSDMinZdMNVc646s25415qK6nBRlqqAY8-
EYmcUIVB9XcNdke4zoUfhVQoruwidzDU_kFafoulo5DEoM0yaH1N2HCSxG5tZXNqocSZPaY8efZYMCpmDXsPAzkmgSkYRDS
Jq0U_jQ99GShlU283nZ85ot2ocbtMAgD_wsrsEh61nt9RaI3HfA7_HcH_fgr9jsTtxDgZhabTBwwDwX0zjWgX1bCuTLKBN7
im7mte1sprf1-A24kksVU_MD9aP9N8_QDsQ13gkh0n5KwFMz3BC2Vw5gL-
gGNHFKDRL6wGIIfhuYcx9LucolZ1yNy9Gbb3ioWSSufyFpyXqtndDLPI5QS1S5JpJm2KDyqcH1SmRLIhd9MNRUC73EAEm-
N05wxPzBRSjhCHZpf8SrYITWJl7K3XzG0fPFh2NgES3jMP9cvSX06yyICcep2HBYGbFflni89-Rw==",
  "destination_column_3":
  "02c6219c70ff45ae788484e5d5d68d57d8253a3b563c47e8a47f8f8e050a1d807a2bb51c622e0ad7d6fcd300f827f
}
```

TimeType (Epoch , 以微秒為單位 , LocalDateTime Java 物件)

```
{
  "destination_column_0": 68175096000,
  "destination_column_1": "18:56:15.096"
}
```

TimestampType.withZone (Epoch , 以微秒為單位、 OffsetDateTime Java 物件、 LocalDateTime Java 物件)

```
{
  "destination_column_0": 1725476175099000,
  "destination_column_1": "2024-09-04T18:56:15.099Z",
  "destination_column_2": "2024-09-04T18:56:15.099"
}
```

DoubleType

```
{
  "destination_column_0": 9.18477568715142,
  "destination_column_1": "9.18477568715142"
}
```

BooleanType

```
{
  "destination_column_0": true,
  "destination_column_1": "false",
  "destination_column_2": 1,
  "destination_column_3": 0
}
```

FloatType

```
{
  "destination_column_0": 0.6242226,
  "destination_column_1": "0.6242226"
}
```

IntegerType

```
{
  "destination_column_0": 7,
  "destination_column_1": "7"
}
```

TimestampType.withoutZone (Epoch , 以微秒為單位、 LocalDateTime Java 物件、 OffsetDateTime Java 物件、 ZonedDateTime Java 物件)

```
{
  "destination_column_0": 1725476175114000,
  "destination_column_1": "2024-09-04T18:56:15.114",
  "destination_column_2": "2024-09-04T18:56:15.114Z",
  "destination_column_3": "2024-09-04T18:56:15.114-07:00"
}
```

DateType

```
{
```

```

"destination_column_0": 19970,
"destination_column_1": "2024-09-04"
}

```

LongType

```

{
  "destination_column_0": 8,
  "destination_column_1": "8"
}

```

UUIDType (UUID Java 物件)

```

{
  "destination_column_0": "21c5521c-a6d4-48d4-b2c8-7f6d842f72c3"
}

```

ListType

```

{
  "destination_column_0":
  ["s1FSrgb0lGDxfn2iYT0Et1P47aHSjwmLZgrdr1JqRs0dmbeCcQoaLr4Xhi2KIVvmus9ppFdpWIc0HnJ0omhAphXH0yns
  "destination_column_1": "[{"destination_nested_column_0": "\bb00f8e6-
db82-4241-a5c5-0d9c0d2f71a4"}, {"destination_nested_column_1": "907.35345"},
{"destination_nested_column_0": "\2c77b702-d405-4fe1-beee-fb541d7ab833"},
{"destination_nested_column_1": "544.0026"}, {"destination_nested_column_0":
"\68389200-d6b1-413d-bcd9-fdb931708395"}, {"destination_nested_column_1": "153.683"},
{"destination_nested_column_0": "\bc31cbaa-39cd-4e2f-b357-9ea9ce75532b"},
{"destination_nested_column_1": "977.5165"}, {"destination_nested_column_0":
"\b7d627f9-0d5b-41b7-903a-525488259fba"}, {"destination_nested_column_1": "434.17215"},
{"destination_nested_column_0": "\06b6ec1e-1952-4582-b285-46aaf40064b8"},
{"destination_nested_column_1": "580.33124"}, {"destination_nested_column_0":
"\f04b3bbf-61ad-4c5c-8740-6f666f57c431"}, {"destination_nested_column_1": "550.75793}]"
}

```

Resources

使用下列資源進一步了解：

- [使用 Amazon Data Firehose 將即時資料串流到 Amazon S3 中的 Apache Iceberg 資料表](#)

- [使用 Apache Iceberg 和 Amazon Data Firehose 簡化 AWS WAF 日誌分析](#)
- [使用 Amazon S3 Tables 和 Amazon Data Firehose 建置用於串流資料的資料湖](#)

標記 Firehose 串流

您可以將自己的中繼資料以標籤形式指派給您在 Amazon Data Firehose 中建立的 Firehose 串流。標籤是您為串流所定義的索引鍵值組。使用標籤是一種簡單但強大的方法來管理 AWS 資源和組織資料，包括帳單資料。

您可以在叫用 [CreateDeliveryStream](#) 建立新的 Firehose 串流時指定標籤。對於現有的 Firehose 串流，您可以使用下列三個操作來新增、列出和移除標籤：

- [TagDeliveryStream](#)
- [ListTagsForDeliveryStream](#)
- [UntagDeliveryStream](#)

了解標籤基本概念

您可以使用 Amazon Data Firehose API 操作來完成下列任務：

- 將標籤新增至 Firehose 串流。
- 列出 Firehose 串流的標籤。
- 從 Firehose 串流移除標籤。

您可以使用標籤來分類 Firehose 串流。例如，您可以依用途、擁有者或環境來分類 Firehose 串流。由於您定義了每個標籤的鍵和值，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，協助您依擁有者和相關聯的應用程式追蹤 Firehose 串流。

下列為數個標籤的範例：

- Project: *Project name*
- Owner: *Name*
- Purpose: Load testing
- Application: *Application name*
- Environment: Production

如果您在 `CreateDeliveryStream` 動作中指定標籤，Amazon Data Firehose 會對 `firehose:TagDeliveryStream` 動作執行額外的授權，以驗證使用者是否具有建立

標籤的許可。如果您未提供此許可，使用 IAM 資源標籤建立新 Firehose 串流的請求會失敗，`AccessDeniedException`如下所示。

```
AccessDeniedException
User: arn:aws:sts::x:assumed-role/x/x is not authorized to perform:
  firehose:TagDeliveryStream on resource: arn:aws:firehose:us-east-1:x:deliverystream/x
  with an explicit deny in an identity-based policy.
```

下列範例示範允許使用者建立 Firehose 串流並套用標籤的政策。

使用標記追蹤成本

您可以使用標籤來分類和追蹤您的 AWS 成本。當您將標籤套用至資源 AWS 時，包括 Firehose 串流，AWS 成本分配報告會包含依標籤彙總的用量和成本。套用代表商業類別的標籤 (例如成本中心、應用程式名稱或擁有者)，即可整理多個服務的成本。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](#)。

了解標籤限制

下列限制適用於 Amazon Data Firehose 中的標籤。

基本限制

- 每項資源 (串流) 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串流的標籤。

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索引鍵的標籤，則新的標籤會覆寫現有鍵值對。
- 標籤金鑰開頭不能為 `aws:`，因為此字首保留供 AWS 使用。AWS 會代表您建立開頭為此字首的標籤，但您無法加以編輯或刪除。
- 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：_ . / = + - @。

Amazon Data Firehose 的安全性

的雲端安全 AWS 是最高優先順序。身為 AWS 客戶，您將受益於資料中心和網路架構，該架構旨在滿足最安全敏感組織的需求。

安全性是 AWS 與您之間共同責任。[共同責任模型](#) 將此描述為雲端的安全和雲端內的安全：

- 雲端的安全性 – AWS 負責保護在 AWS Cloud 中執行 AWS 服務的基礎設施。AWS 也為您提供可安全使用的服務。第三方稽核人員定期檢測及驗證安全的效率也是我們 [AWS 合規計劃](#) 的一部分。若要了解適用於 Data Firehose 的合規計劃，請參閱 [AWS 合規計劃範圍內的服務](#)。
- 雲端的安全性 – 您的責任取決於您使用 AWS 的服務。您也必須對資料敏感度、組織要求，以及適用法律和法規等其他因素負責。

本文件可協助您了解如何在使用 Data Firehose 時套用共同責任模型。下列主題說明如何設定 Data Firehose 以符合您的安全與合規目標。您也將了解如何使用其他 AWS 服務，協助您監控和保護 Data Firehose 資源。

主題

- [Amazon Data Firehose 中的資料保護](#)
- [使用 Amazon Data Firehose 控制存取](#)
- [在 Amazon Data Firehose AWS Secrets Manager 中使用 驗證](#)
- [透過 Amazon Data Firehose 主控台管理 IAM 角色](#)
- [了解 Amazon Data Firehose 的合規性](#)
- [Amazon Data Firehose 中的彈性](#)
- [了解 Amazon Data Firehose 中的基礎設施安全性](#)
- [實作 Amazon Data Firehose 的安全最佳實務](#)

Amazon Data Firehose 中的資料保護

Amazon Data Firehose 會使用 TLS 通訊協定加密傳輸中的所有資料。此外，對於在處理期間存放在臨時儲存中的資料，Amazon Data Firehose 會使用 加密資料 [AWS Key Management Service](#)，並使用檢查總和驗證來驗證資料完整性。

如果您有敏感資料，您可以在使用 Amazon Data Firehose 時啟用伺服器端資料加密。您的做法取決於資料的來源。

Note

如果您在 AWS 透過命令列界面或 API 存取時需要 FIPS 140-2 驗證的密碼編譯模組，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](#)。

使用 Kinesis Data Streams 進行伺服器端加密

當您將資料從資料生產者傳送到資料串流時，Kinesis Data Streams 會使用 AWS Key Management Service (AWS KMS) 金鑰加密您的資料，然後再存放靜態資料。當您的 Firehose 串流從資料串流讀取資料時，Kinesis Data Streams 會先解密資料，然後將其傳送至 Amazon Data Firehose。Amazon Data Firehose 會根據您指定的緩衝提示，緩衝記憶體中的資料。再傳送到您的目的地，而不會將未加密的資料儲存為靜態資料。

如需如何為 Kinesis 資料串流啟用伺服器端加密的相關資訊，請參閱《Amazon Kinesis Data Streams 開發人員指南》中的[使用伺服器端加密](#)。

使用 Direct PUT 或其他資料來源的伺服器端加密

如果您使用 [PutRecord](#) 或 [PutRecordBatch](#) 將資料傳送至 Firehose 串流，或使用 AWS IoT Amazon CloudWatch Logs 或 CloudWatch Events 傳送資料，您可以使用 [StartDeliveryStreamEncryption](#) 操作開啟伺服器端加密。

欲停用伺服器端加密，請使用 [StopDeliveryStreamEncryption](#) 操作。

您也可以在建立 Firehose 串流時啟用 SSE。為此，請在調用 [CreateDeliveryStream](#) 時指定 [DeliveryStreamEncryptionConfigurationInput](#)。

若要成功使用 CUSTOMER_MANAGED_CMK，呼叫者的 IAM 政策和 KMS 金鑰政策都必須允許 kms:GenerateDataKey 和 kms:Decrypt 操作。當您使用 CUSTOMER_MANAGED_CMK 加密呼叫 PutRecord 或 PutRecordBatch 時，Firehose 會驗證這些許可。此外，使用 CUSTOMER_MANAGED_CMK 加密呼叫 CreateDeliveryStream 或 StartDeliveryStreamEncryption 時需要 kms:CreateGrant 許可。

當 CMK 類型為 CUSTOMER_MANAGED_CMK，如果 Amazon Data Firehose 服務因為 KMSNotFoundException、KMSInvalidStateException、KMSDisabledException 或無法解密記錄 KMSAccessDeniedException，服務會等待最多 24 小時（保留期間）讓您解決問題。如果問題持續超過保留期間，服務會略過那些已經過保留期間且無法解密的記錄，然後捨棄資料。Amazon Data Firehose 提供下列四個 CloudWatch 指標，可用來追蹤四個 AWS KMS 例外狀況：

- KMSKeyAccessDenied
- KMSKeyDisabled
- KMSKeyInvalidState
- KMSKeyNotFound

如需這四個指標的詳細資訊，請參閱 [the section called “使用 CloudWatch 指標監控使用量”](#)。

Important

若要加密 Firehose 串流，請使用對稱 CMKs。Amazon Data Firehose 不支援非對稱 CMKs。如需對稱和非對稱 CMKs 的相關資訊，請參閱 AWS Key Management Service 開發人員指南中的 [關於對稱和非對稱 CMKs](#)。

Note

當您使用 [客戶受管金鑰](#) (CUSTOMER_MANAGED_CMK) 為您的 Firehose 串流啟用伺服器端加密 (SSE) 時，Firehose 服務會在使用您的金鑰時設定加密內容。由於此加密內容代表使用您 AWS 帳戶所擁有的金鑰，因此會記錄為 AWS 帳戶 AWS CloudTrail 事件日誌的一部分。此加密內容是由 Firehose 服務產生的系統。您的應用程式不應對 Firehose 服務所設定的加密內容的格式或內容做出任何假設。

使用 Amazon Data Firehose 控制存取

下列各節說明如何控制對 Amazon Data Firehose 資源的存取。他們涵蓋的資訊包括如何授予應用程式存取權，以便將資料傳送到 Firehose 串流。它們也說明如何授予 Amazon Data Firehose 存取 Amazon Simple Storage Service (Amazon S3) 儲存貯體、Amazon Redshift 叢集或 Amazon OpenSearch Service 叢集的權限，以及如果您使用 Datadog、Dynatrace、LogicMonitor、MongoDB、New Relic、Splunk 或 Sumo Logic 做為目的地時所需的存取許可。最後，您可以在本主題指南中找到如何設定 Amazon Data Firehose，以便將資料交付到屬於不同 AWS 帳戶的目的地。管理所有這些存取形式的技術是 AWS Identity and Access Management (IAM)。如需 IAM 的相關資訊，請參閱 [什麼是 IAM?](#)。

目錄

- [授予 Firehose 資源的存取權](#)

- [授予 Firehose 存取您的私有 Amazon MSK 叢集](#)
- [允許 Firehose 擔任 IAM 角色](#)
- [授予 Firehose 對的存取權 AWS Glue 以進行資料格式轉換](#)
- [授予 Firehose 對 Amazon S3 目的地的存取權](#)
- [授予 Firehose 對 Amazon S3 資料表的存取權](#)
- [授予 Firehose 存取 Apache Iceberg 資料表目的地的權限](#)
- [授予 Firehose 對 Amazon Redshift 目的地的存取權](#)
- [授予 Firehose 對公有 OpenSearch Service 目的地的存取權](#)
- [授予 Firehose 對 VPC 中 OpenSearch Service 目的地的存取權](#)
- [授予 Firehose 對公有 OpenSearch Serverless 目的地的存取權](#)
- [授予 Firehose 對 VPC 中 OpenSearch Serverless 目的地的存取權](#)
- [授予 Firehose 存取 Splunk 目的地的權限](#)
- [在 VPC 中存取 Splunk](#)
- [使用 Amazon Data Firehose 將 VPC 流程日誌擷取至 Splunk](#)
- [存取 Snowflake 或 HTTP 端點](#)
- [授予 Firehose 存取 Snowflake 目的地的權限](#)
- [在 VPC 中存取 Snowflake](#)
- [授予 Firehose 對 HTTP 端點目的地的存取權](#)
- [從 Amazon MSK 跨帳戶交付](#)
- [跨帳戶交付至 Amazon S3 目的地](#)
- [跨帳戶交付至 OpenSearch Service 目的地](#)
- [使用標籤控制存取](#)

授予 Firehose 資源的存取權

若要讓您的應用程式存取 Firehose 串流，請使用類似此範例的政策。您可以修改 Action 部分，調整授予存取的個別 API 操作，或搭配 "firehose:*"，將存取權授予給所有操作。

JSON

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "firehose:DeleteDeliveryStream",
      "firehose:PutRecord",
      "firehose:PutRecordBatch",
      "firehose:UpdateDestination"
    ],
    "Resource": [
      "arn:aws:firehose:us-east-1:123456789012:deliverystream/delivery-stream-name"
    ]
  }
]
```

授予 Firehose 存取您的私有 Amazon MSK 叢集

如果您的 Firehose 串流來源是私有 Amazon MSK 叢集，請使用類似此範例的政策。

您必須將此政策新增至叢集的資源型政策，以授予 Firehose 服務主體調用 Amazon MSK CreateVpcConnection API 操作的許可。

允許 Firehose 擔任 IAM 角色

本節說明授予 Amazon Data Firehose 擷取、處理和將資料從來源交付至目的地的許可和政策。

Note

如果您使用主控台來建立 Firehose 串流，並選擇建立新角色的選項，會將所需的信任政策 AWS 連接到該角色。如果您希望 Amazon Data Firehose 使用現有的 IAM 角色，或您自己建立角色，請將下列信任政策連接到該角色，以便 Amazon Data Firehose 可以擔任該角色。編輯政策，將 *account-id* 取代為 AWS 您的帳戶 ID。如需有關如何修改角色之信任關係的資訊，請參閱[修改角色](#)。

Amazon Data Firehose 會將 IAM 角色用於 Firehose 串流處理和交付資料所需的所有許可。請確定下列信任政策已連接至該角色，以便 Amazon Data Firehose 可以擔任該角色。

如果您選擇 Amazon MSK 做為 Firehose 串流的來源，您必須指定另一個 IAM 角色，以授予 Amazon Data Firehose 從指定 Amazon MSK 叢集擷取來源資料的許可。請確定下列信任政策已連接至該角色，以便 Amazon Data Firehose 可以擔任該角色。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Principal": {
        "Service": [
          "firehose.amazonaws.com"
        ]
      },
      "Effect": "Allow",
      "Action": "sts:AssumeRole"
    }
  ]
}
```

請確定此角色授予 Amazon Data Firehose 從指定 Amazon MSK 叢集擷取來源資料的許可，並授予下列許可：

授予 Firehose 對的存取權 AWS Glue 以進行資料格式轉換

如果您的 Firehose 串流執行資料格式轉換，Amazon Data Firehose 會參考存放於 [中的資料表定義 AWS Glue](#)。若要為 Amazon Data Firehose 提供必要的存取權 AWS Glue，請將下列陳述式新增至您的政策。如需如何尋找資料表 ARN 的資訊，請參閱[指定 AWS Glue 資源 ARNs](#)。

```
{
  "Sid": "",
  "Effect": "Allow",
  "Action": [
    "glue:GetTable",
    "glue:GetTableVersion",
    "glue:GetTableVersions"
  ],
```

```
"Resource": [
  "arn:aws:glue:us-east-1:123456789012:catalog",
  "arn:aws:glue:us-east-1:123456789012:database/b",
  "arn:aws:glue:us-east-1:123456789012:table/b/easd"
],
{
  actions: ['glue:GetSchemaVersion'],
  grantee: options.role,
  resourceArns: ['*'],
}
```

從結構描述登錄檔取得結構描述的建議政策沒有資源限制。如需詳細資訊，請參閱《AWS Glue 開發人員指南》中的[還原序列化程式的 IAM 範例](#)。

授予 Firehose 對 Amazon S3 目的地的存取權

當您使用 Amazon S3 目的地時，Amazon Data Firehose 會將資料交付到您的 S3 儲存貯體，並可選擇使用您擁有的 AWS KMS 金鑰進行資料加密。如果啟用錯誤記錄，Amazon Data Firehose 也會將資料交付錯誤傳送至 CloudWatch 日誌群組和串流。建立 Firehose 串流時，您必須擁有 IAM 角色。Amazon Data Firehose 假設 IAM 角色和 可以存取指定的儲存貯體、金鑰和 CloudWatch 日誌群組和串流。

使用以下存取政策，讓 Amazon Data Firehose 存取您的 S3 儲存貯體和 AWS KMS 金鑰。如果您未擁有 S3 儲存貯體，請新增 `s3:PutObjectAcl` 至 Amazon S3 動作清單。這可讓儲存貯體擁有者完整存取 Amazon Data Firehose 交付的物件。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:AbortMultipartUpload",
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
```

```

        "s3:PutObject"
    ],
    "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "kinesis:DescribeStream",
        "kinesis:GetShardIterator",
        "kinesis:GetRecords",
        "kinesis:ListShards"
    ],
    "Resource": "arn:aws:kinesis:us-east-1:123456789012:stream/stream-
name"
},
{
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
    ],
    "Resource": [
        "arn:aws:kms:us-east-1:123456789012:key/key-id"
    ],
    "Condition": {
        "StringEquals": {
            "kms:ViaService": "s3.us-east-1.amazonaws.com"
        },
        "StringLike": {
            "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::amzn-s3-
demo-bucket/prefix*"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "logs:PutLogEvents"
    ],
    "Resource": [

```

```

        "arn:aws:logs:us-east-1:123456789012:log-group:log-group-name:log-
stream:log-stream-name"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "lambda:InvokeFunction",
      "lambda:GetFunctionConfiguration"
    ],
    "Resource": [
      "arn:aws:lambda:us-east-1:123456789012:function:func-
tion-name:func-version"
    ]
  }
]
}

```

上述政策亦有允許存取 Amazon Kinesis Data Streams 的陳述式。如果您不使用 Amazon Kinesis Data Streams 作為資料來源，可移除該陳述式。如果您使用 Amazon MSK 做為來源，則可以使用下列內容取代該陳述式：

```

{
  "Sid": "",
  "Effect": "Allow",
  "Action": [
    "kafka:GetBootstrapBrokers",
    "kafka:DescribeCluster",
    "kafka:DescribeClusterV2",
    "kafka-cluster:Connect"
  ],
  "Resource": "arn:aws:kafka:{{mskClusterRegion}}:{{mskClusterAccount}}:cluster/
{{mskClusterName}}/{{clusterUUID}}"
},
{
  "Sid": "",
  "Effect": "Allow",
  "Action": [
    "kafka-cluster:DescribeTopic",
    "kafka-cluster:DescribeTopicDynamicConfiguration",
    "kafka-cluster:ReadData"
  ],
}

```

```
"Resource": "arn:aws:kafka:{{mskClusterRegion}}:{{mskClusterAccount}}:topic/
{{mskClusterName}}/{{clusterUUID}}/{{mskTopicName}}"
},
{
  "Sid": "",
  "Effect": "Allow",
  "Action": [
    "kafka-cluster:DescribeGroup"
  ],
  "Resource": "arn:aws:kafka:{{mskClusterRegion}}:{{mskClusterAccount}}:group/
{{mskClusterName}}/{{clusterUUID}}/*"
}
```

如需允許其他 AWS 服務存取 資源 AWS 的詳細資訊，請參閱《IAM 使用者指南》中的[建立角色以委派許可給 AWS 服務](#)。

若要了解如何授予 Amazon Data Firehose 對另一個帳戶中 Amazon S3 目的地的存取權，請參閱 [the section called “跨帳戶交付至 Amazon S3 目的地”](#)。

授予 Firehose 對 Amazon S3 資料表的存取權

Firehose 需要具有特定許可的 IAM 角色，才能存取 AWS AWS Glue 資料表並將資料寫入 Amazon S3 資料表儲存貯體中的資料表。若要寫入 Amazon S3 資料表儲存貯體中的資料表，您還必須提供 IAM 角色所需的許可。Amazon S3 Tables 目錄所需的許可取決於您使用的存取控制模式：

- IAM 存取控制 – Firehose 交付角色需要直接在 Amazon S3 Tables 資源上的 IAM 許可。
- Lake Formation 存取控制 – Firehose 交付角色需要 AWS AWS Lake Formation 許可才能管理對資料表資源的存取。AWS Lake Formation 會使用自己的許可模型，為 Data Catalog 資源啟用精細存取控制。

您可以在建立 Firehose 串流時設定此 IAM 角色。選擇與您的存取控制模式對應的標籤。

IAM 存取控制

如果您使用 IAM 存取控制模式（不含 AWS Lake Formation），Firehose 交付角色需要直接在 Amazon S3 Tables 資源和 AWS Glue Data Catalog 物件上取得 IAM 許可。

登入 AWS 管理主控台 並開啟位於 <https://console.aws.amazon.com/iam/> 的 IAM 主控台。

建立政策，然後在政策編輯器中選擇 JSON。新增以下授予必要許可的內嵌政策。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "S3TablesAccessPermission",
      "Effect": "Allow",
      "Action": [
        "s3tables:GetTable",
        "s3tables:GetTableData",
        "s3tables:GetTableMetadataLocation",
        "s3tables:UpdateTableMetadataLocation"
      ],
      "Resource": [
        "arn:aws:s3tables:region:account-id:bucket/*",
        "arn:aws:s3tables:region:account-id:bucket/*/table/*"
      ]
    },
    {
      "Sid": "S3TableBucketAccessPermission",
      "Effect": "Allow",
      "Action": [
        "s3tables:GetTableBucket"
      ],
      "Resource": "arn:aws:s3tables:region:account-id:bucket/*"
    },
    {
      "Sid": "GlueCatalogAccessPermission",
      "Effect": "Allow",
      "Action": [
        "glue:GetDatabase",
        "glue:GetDatabases",
        "glue:GetTable",
        "glue:GetTables",
        "glue:UpdateTable"
      ],
      "Resource": [
        "arn:aws:glue:region:account-id:catalog",
        "arn:aws:glue:region:account-id:catalog/s3tablescatalog",
        "arn:aws:glue:region:account-id:catalog/s3tablescatalog/*",
        "arn:aws:glue:region:account-id:database/*",
        "arn:aws:glue:region:account-id:table/*/*"
      ]
    }
  ],
}
```

```

    {
      "Sid": "S3DeliveryErrorBucketPermission",
      "Effect": "Allow",
      "Action": [
        "s3:AbortMultipartUpload",
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::error-delivery-bucket",
        "arn:aws:s3:::error-delivery-bucket/*"
      ]
    },
    {
      "Sid": "RequiredWhenUsingKinesisDataStreamsAsSource",
      "Effect": "Allow",
      "Action": [
        "kinesis:DescribeStream",
        "kinesis:GetShardIterator",
        "kinesis:GetRecords",
        "kinesis:ListShards"
      ],
      "Resource": "arn:aws:kinesis:region:account-id:stream/stream-name"
    },
    {
      "Sid": "KMSPermissionForS3TablesEncryption",
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
      ],
      "Resource": [
        "arn:aws:kms:region:account-id:key/key-id"
      ],
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "s3.amazonaws.com"
        },
        "StringLike": {
          "kms:EncryptionContext:aws:s3:arn":
            "arn:aws:s3tables:region:account-id:bucket/*/table/*"
        }
      }
    }
  ]
}

```

```
    }
  }
},
{
  "Sid": "RequiredWhenUsingLambdaForDataTransformation",
  "Effect": "Allow",
  "Action": [
    "lambda:InvokeFunction",
    "lambda:GetFunctionConfiguration"
  ],
  "Resource": "arn:aws:lambda:region:account-id:function:function-
name:function-version"
},
{
  "Sid": "CloudWatchLogsPermission",
  "Effect": "Allow",
  "Action": [
    "logs:PutLogEvents"
  ],
  "Resource": "arn:aws:logs:region:account-id:log-group:log-group-name:log-
stream:log-stream-name"
}
]
}
```

此政策具有允許存取 Amazon Kinesis Data Streams、叫用 Lambda 函數和存取 AWS KMS 金鑰的陳述式。如果您不使用任何這些資源，您可以移除個別的陳述式。如果啟用錯誤記錄，Amazon Data Firehose 也會將資料交付錯誤傳送至 CloudWatch 日誌群組和串流。您必須設定日誌群組和日誌串流名稱，才能使用此選項。如需日誌群組和日誌串流名稱，請參閱 [使用 CloudWatch Logs 監控 Amazon Data Firehose](#)。

在內嵌政策中，將預留位置值取代為您的實際資源名稱、AWS 帳戶 數字和區域。

建立政策後，請開啟位於 <https://console.aws.amazon.com/iam/> : // 的 IAM 主控台，並使用 建立 IAM 角色AWS 服務做為信任的實體類型。

針對服務或使用案例，選擇 Kinesis。針對使用案例，選擇 Kinesis Firehose。

在下一頁中，選擇要連接到此角色的上一個步驟中建立的政策。在檢閱頁面上，您會找到已連接到此角色的信任政策，授予 Firehose 服務擔任此角色的許可。當您建立角色時，Amazon Data Firehose 可以擔任它在 AWS Glue 和 Amazon S3 Tables 上執行必要的操作。將 Firehose 服務主體新增至所建立角色的信任政策。如需詳細資訊，請參閱 [允許 Firehose 擔任 IAM 角色](#)。

Lake Formation 存取控制

如果您使用的是 AWS Lake Formation 存取控制模式，除了 IAM 政策之外，Firehose 交付角色還需要憑證販賣的 AWS Lake Formation 許可。

登入 AWS 管理主控台 並開啟位於 <https://console.aws.amazon.com/iam/> 的 IAM 主控台。

建立政策，然後在政策編輯器中選擇 JSON。新增下列內嵌政策，以授予 Amazon S3 許可，例如讀取/寫入許可、更新資料目錄中資料表的許可，以及其他許可。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "S3TableAccessViaGlueFederation",
      "Effect": "Allow",
      "Action": [
        "glue:GetTable",
        "glue:GetDatabase",
        "glue:UpdateTable"
      ],
      "Resource": [
        "arn:aws:glue:us-east-1:123456789012:catalog/s3tablescatalog/*",
        "arn:aws:glue:us-east-1:123456789012:catalog/s3tablescatalog",
        "arn:aws:glue:us-east-1:123456789012:catalog",
        "arn:aws:glue:us-east-1:123456789012:database/*",
        "arn:aws:glue:us-east-1:123456789012:table/*/*"
      ]
    },
    {
      "Sid": "S3DeliveryErrorBucketPermission",
      "Effect": "Allow",
      "Action": [
        "s3:AbortMultipartUpload",
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:PutObject"
      ],
      "Resource": [
```

```

    "arn:aws:s3:::<error delivery bucket>",
    "arn:aws:s3:::<error delivery bucket>/*"
  ]
},
{
  "Sid": "RequiredWhenUsingKinesisDataStreamsAsSource",
  "Effect": "Allow",
  "Action": [
    "kinesis:DescribeStream",
    "kinesis:GetShardIterator",
    "kinesis:GetRecords",
    "kinesis:ListShards"
  ],
  "Resource": "arn:aws:kinesis:us-east-1:123456789012:stream/<stream-name>"
},
{
  "Sid":
"RequiredWhenDoingMetadataReadsANDDataAndMetadataWriteViaLakeformation",
  "Effect": "Allow",
  "Action": [
    "lakeformation:GetDataAccess"
  ],
  "Resource": "*"
},
{
  "Sid": "RequiredWhenUsingKMSEncryptionForS3ErrorBucketDelivery",
  "Effect": "Allow",
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey"
  ],
  "Resource": [
    "arn:aws:kms:us-east-1:123456789012:key/<KMS-key-id>"
  ],
  "Condition": {
    "StringEquals": {
      "kms:ViaService": "s3.us-east-1.amazonaws.com"
    },
    "StringLike": {
      "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::<error delivery
bucket>/prefix*"
    }
  }
},
},

```

```

{
  "Sid": "LoggingInCloudWatch",
  "Effect": "Allow",
  "Action": [
    "logs:PutLogEvents"
  ],
  "Resource": [
    "arn:aws:logs:us-east-1:123456789012:log-group:log-group-name>:log-
stream:<log-stream-name>"
  ]
},
{
  "Sid": "RequiredWhenAttachingLambdaToFirehose",
  "Effect": "Allow",
  "Action": [
    "lambda:InvokeFunction",
    "lambda:GetFunctionConfiguration"
  ],
  "Resource": [
    "arn:aws:lambda:us-east-1:123456789012:function:<function-
name>:<function-version>"
  ]
}
]
}

```

此政策具有允許存取 Amazon Kinesis Data Streams、叫用 Lambda 函數和存取 AWS KMS 金鑰的陳述式。如果您不使用任何這些資源，您可以移除個別的陳述式。如果啟用錯誤記錄，Amazon Data Firehose 也會將資料交付錯誤傳送至 CloudWatch 日誌群組和串流。您必須設定日誌群組和日誌串流名稱，才能使用此選項。如需日誌群組和日誌串流名稱，請參閱 [使用 CloudWatch Logs 監控 Amazon Data Firehose](#)。

在內嵌政策中，將 <error delivery bucket> 取代為您的 Amazon S3 儲存貯體名稱，aws-account-id 並將 區域 取代為資源的有效 AWS 帳戶 數字和區域。

除了 IAM 政策之外，您還必須授予 Firehose 交付角色所需的許可 AWS Lake Formation。如需詳細資訊，請參閱 [授予資料表的許可](#)。

建立政策後，請開啟位於 <https://console.aws.amazon.com/iam/> : // 的 IAM 主控台，並使用 建立 IAM 角色 AWS 服務 做為信任的實體類型。

針對服務或使用案例，選擇 Kinesis。針對使用案例，選擇 Kinesis Firehose。

在下一頁中，選擇要連接到此角色的上一個步驟中建立的政策。在檢閱頁面上，您會找到已連接到此角色的信任政策，授予 Firehose 服務擔任此角色的許可。當您建立角色時，Amazon Data Firehose 可以擔任該角色，以在 AWS Glue 和 S3 儲存貯體上執行必要的操作。將 Firehose 服務主體新增至所建立角色的信任政策。如需詳細資訊，請參閱[允許 Firehose 擔任 IAM 角色](#)。

授予 Firehose 存取 Apache Iceberg 資料表目的地的權限

您必須先擁有 IAM 角色，才能使用 建立 Firehose 串流和 Apache Iceberg 資料表 AWS Glue。使用下列步驟來建立政策和 IAM 角色。Firehose 擔任此 IAM 角色並執行必要的動作。

1. 登入 AWS 管理主控台 並開啟位於 <https://console.aws.amazon.com/iam/> 的 IAM 主控台。
2. 建立政策，然後在政策編輯器中選擇 JSON。
3. 新增下列內嵌政策，以授予 Amazon S3 讀取/寫入許可、更新資料目錄中資料表的許可等許可。

此政策的陳述式允許存取 Amazon Kinesis Data Streams、叫用 Lambda 函數，以及存取 KMS 金鑰。如果您不使用任何這些資源，您可以移除個別的陳述式。

如果啟用錯誤記錄，Firehose 也會一併將資料交付錯誤傳送至 CloudWatch 日誌群組與串流。為此，您必須設定日誌群組和日誌串流名稱。如需日誌群組和日誌串流名稱，請參閱[使用 CloudWatch Logs 監控 Amazon Data Firehose](#)。

4. 在內嵌政策中，將 `amzn-s3-demo-bucket` 取代為您的 Amazon S3 儲存貯體名稱、aws-account-id 和區域，取代為資源的有效 AWS 帳戶 數量和區域。

Note

此角色會授予資料目錄中所有資料庫和資料表的許可。如有需要，您可以只授予特定資料表和資料庫的許可。

5. 建立政策之後，請開啟 [IAM 主控台](#)，並使用 建立 IAM 角色AWS 服務做為信任的實體類型。
6. 針對服務或使用案例，選擇 Kinesis。針對使用案例，選擇 Kinesis Firehose。
7. 在下一頁中，選擇要連接到此角色的上一個步驟中建立的政策。在檢閱頁面上，您會找到已連接到此角色的信任政策，授予 Firehose 服務擔任此角色的許可。當您建立角色時，Amazon Data Firehose 可以擔任該角色，以在 和 S3 儲存貯體上執行 AWS Glue 必要的操作。

授予 Firehose 對 Amazon Redshift 目的地的存取權

使用 Amazon Redshift 目的地時，當您授予 Amazon Data Firehose 的存取權時，請參閱下列內容。

主題

- [IAM 角色和存取政策](#)
- [Amazon Redshift 佈建叢集或 Amazon Redshift Serverless 工作群組的 VPC 存取權](#)

IAM 角色和存取政策

當您使用 Amazon Redshift 目的地時，Amazon Data Firehose 會將資料交付至 S3 儲存貯體做為中繼位置。它可以選擇性地使用您擁有的 AWS KMS 金鑰進行資料加密。然後，Amazon Data Firehose 會將資料從 S3 儲存貯體載入您的 Amazon Redshift 佈建叢集或 Amazon Redshift Serverless 工作群組。如果啟用錯誤記錄，Amazon Data Firehose 也會將資料交付錯誤傳送至 CloudWatch 日誌群組和串流。Amazon Data Firehose 使用指定的 Amazon Redshift 使用者名稱和密碼來存取您佈建的叢集或 Amazon Redshift Serverless 工作群組，並使用 IAM 角色來存取指定的儲存貯體、金鑰、CloudWatch 日誌群組和串流。建立 Firehose 串流時，您必須擁有 IAM 角色。

使用以下存取政策，讓 Amazon Data Firehose 存取您的 S3 儲存貯體和 AWS KMS 金鑰。如果您不擁有 S3 儲存貯體，請將 `s3:PutObjectAcl` 新增至 Amazon S3 動作清單，這會授予儲存貯體擁有者對 Amazon Data Firehose 交付物件的完整存取權。此政策亦有允許存取 Amazon Kinesis Data Streams 的陳述式。如果您不使用 Amazon Kinesis Data Streams 作為資料來源，可移除該陳述式。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:AbortMultipartUpload",
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:PutObject"
      ]
    }
  ],
}
```

```

    "Resource": [
      "arn:aws:s3:::amzn-s3-demo-bucket",
      "arn:aws:s3:::amzn-s3-demo-bucket/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "kms:Decrypt",
      "kms:GenerateDataKey"
    ],
    "Resource": [
      "arn:aws:kms:us-east-1:123456789012:key/key-id"
    ],
    "Condition": {
      "StringEquals": {
        "kms:ViaService": "s3.us-east-1.amazonaws.com"
      },
      "StringLike": {
        "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::amzn-s3-
demo-bucket/prefix*"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "kinesis:DescribeStream",
      "kinesis:GetShardIterator",
      "kinesis:GetRecords",
      "kinesis:ListShards"
    ],
    "Resource": "arn:aws:kinesis:us-east-1:123456789012:stream/stream-
name"
  },
  {
    "Effect": "Allow",
    "Action": [
      "logs:PutLogEvents"
    ],
    "Resource": [
      "arn:aws:logs:us-east-1:123456789012:log-group:log-group-name:log-
stream:log-stream-name"
    ]
  }
]

```

```

    },
    {
      "Effect": "Allow",
      "Action": [
        "lambda:InvokeFunction",
        "lambda:GetFunctionConfiguration"
      ],
      "Resource": [
        "arn:aws:lambda:us-east-1:123456789012:function:function-
name:function-version"
      ]
    }
  ]
}

```

如需允許其他 AWS 服務存取 資源 AWS 的詳細資訊，請參閱《IAM 使用者指南》中的[建立角色以委派許可給 AWS 服務](#)。

Amazon Redshift 佈建叢集或 Amazon Redshift Serverless 工作群組的 VPC 存取權

如果您的 Amazon Redshift 佈建的叢集或 Amazon Redshift Serverless 工作群組位於虛擬私有雲端 (VPC)，則其必須有公有 IP 地址，可供公開存取。此外，透過解除封鎖 Amazon Data Firehose IP 地址，授予 Amazon Data Firehose 存取 Amazon Redshift 佈建叢集或 Amazon Redshift Serverless 工作群組的權限。Amazon Data Firehose 目前針對每個可用區域使用一個 CIDR 區塊。

區域	CIDR 區塊
美國東部 (俄亥俄)	13.58.135.96/27
美國東部 (維吉尼亞北部)	52.70.63.192/27
美國西部 (加利佛尼亞北部)	13.57.135.192/27
美國西部 (奧勒岡)	52.89.255.224/27
AWS GovCloud (美國東部)	18.253.138.96/27

區域	CIDR 區塊
AWS GovCloud (美國西部)	52.61.204.160/27
加拿大 (中部)	35.183.92.128/27
加拿大西部 (卡加利)	40.176.98.192/27
亞太地區 (香港)	18.162.221.32/27
亞太地區 (孟買)	13.232.67.32/27
亞太地區 (海德拉巴)	18.60.192.128/27
亞太地區 (首爾)	13.209.1.64/27
亞太區域 (新加坡)	13.228.64.192/27
亞太地區 (悉尼)	13.210.67.224/27
亞太地區 (雅加達)	108.136.221.64/27
亞太地區 (東京)	13.113.196.224/27
亞太地區 (大阪)	13.208.177.192/27
亞太區域 (泰國)	43.208.112.96/27
亞太區域 (台北)	43.212.53.160/27
中國 (北京)	52.81.151.32/27
中國 (寧夏)	161.189.23.64/27
歐洲 (蘇黎世)	16.62.183.32/27
歐洲 (法蘭克福)	35.158.127.160/27
歐洲 (愛爾蘭)	52.19.239.192/27
歐洲 (倫敦)	18.130.1.96/27

區域	CIDR 區塊
歐洲 (巴黎)	35.180.1.96/27
歐洲 (斯德哥爾摩)	13.53.63.224/27
歐洲 (西班牙)	18.100.71.96/27
Middle East (Bahrain)	15.185.91.0/27
墨西哥 (中部)	78.12.207.32/27
南美洲 (聖保羅)	18.228.1.128/27
歐洲 (米蘭)	15.161.135.128/27
非洲 (開普敦)	13.244.121.224/27
中東 (阿拉伯聯合大公國)	3.28.159.32/27
以色列 (特拉維夫)	51.16.102.0/27
亞太地區 (墨爾本)	16.50.161.128/27
亞太區域 (馬來西亞)	43.216.58.0/27

如需如何解鎖 IP 地址的詳細資訊，請參閱《Amazon Redshift 入門指南》中的[授予叢集存取步驟](#)。

授予 Firehose 對公有 OpenSearch Service 目的地的存取權

當您使用 OpenSearch Service 目的地時，Amazon Data Firehose 會將資料交付到您的 OpenSearch Service 叢集，並同時將失敗或所有文件備份到您的 S3 儲存貯體。如果啟用錯誤記錄，Amazon Data Firehose 也會將資料交付錯誤傳送至 CloudWatch 日誌群組和串流。Amazon Data Firehose 使用 IAM 角色來存取指定的 OpenSearch Service 網域、S3 儲存貯體、AWS KMS 金鑰和 CloudWatch 日誌群組和串流。建立 Firehose 串流時，您必須擁有 IAM 角色。

使用以下存取政策，讓 Amazon Data Firehose 存取您的 S3 儲存貯體、OpenSearch Service 網域和 AWS KMS 金鑰。如果您不擁有 S3 儲存貯體，請將新增至 Amazon S3 動作 `s3:PutObjectAcl` 清單，這會授予儲存貯體擁有者對 Amazon Data Firehose 交付物件的完整存取權。此政策亦有允許存取

Amazon Kinesis Data Streams 的陳述式。如果您不使用 Amazon Kinesis Data Streams 作為資料來源，可移除該陳述式。

如需允許其他 AWS 服務存取 資源 AWS 的詳細資訊，請參閱《IAM 使用者指南》中的[建立角色以委派許可給 AWS 服務](#)。

若要了解如何授予 Amazon Data Firehose 對另一個帳戶中 OpenSearch Service 叢集的存取權，請參閱 [the section called “跨帳戶交付至 OpenSearch Service 目的地”](#)。

授予 Firehose 對 VPC 中 OpenSearch Service 目的地的存取權

如果您的 OpenSearch Service 網域位於 VPC 中，請務必為 Amazon Data Firehose 提供上一節所述的許可。此外，您需要為 Amazon Data Firehose 提供下列許可，使其能夠存取 OpenSearch Service 網域的 VPC。

- ec2:DescribeVpcs
- ec2:DescribeVpcAttribute
- ec2:DescribeSubnets
- ec2:DescribeSecurityGroups
- ec2:DescribeNetworkInterfaces
- ec2:CreateNetworkInterface
- ec2:CreateNetworkInterfacePermission
- ec2>DeleteNetworkInterface

Important

請勿在建立 Firehose 串流後撤銷這些許可。如果您撤銷這些許可，每當服務嘗試查詢或更新 ENIs 時，您的 Firehose 串流將會降級或停止將資料交付到您的 OpenSearch 服務網域。

Important

當您指定子網路以將資料交付至私有 VPC 中的目的地時，請確定您選擇的子網路中有足夠的可用 IP 地址。如果指定的子網路中沒有可用的可用 IP 地址，Firehose 無法為私有 VPC 中的資料交付建立或新增 ENIs，且交付將會降級或失敗。

當您建立或更新 Firehose 串流時，您可以為 Firehose 指定在將資料傳送至 OpenSearch Service 網域時要使用的安全群組。您使用的安全群組可與 OpenSearch Service 域所使用的相同或不同。如果您指定不同的安全群組，請確認其允許輸出 HTTPS 流量到 OpenSearch Service 域的安全群組。此外，請確定 OpenSearch Service 網域的安全群組允許來自您在設定 Firehose 串流時所指定安全群組的 HTTPS 流量。如果您針對 Firehose 串流和 OpenSearch Service 網域使用相同的安全群組，請確定安全群組傳入規則允許 HTTPS 流量。如需安全群組規則的詳細資訊，請參閱 Amazon VPC 文件中 OpenSearch 的[安全群組規則](#)。

授予 Firehose 對公有 OpenSearch Serverless 目的地的存取權

當您使用 OpenSearch Serverless 目的地時，Amazon Data Firehose 會將資料交付到您的 OpenSearch Serverless 集合，並同時將失敗或所有文件備份到您的 S3 儲存貯體。如果啟用錯誤記錄，Amazon Data Firehose 也會將資料交付錯誤傳送至 CloudWatch 日誌群組和串流。Amazon Data Firehose 使用 IAM 角色來存取指定的 OpenSearch Serverless 集合、S3 儲存貯體、AWS KMS 金鑰和 CloudWatch 日誌群組和串流。建立 Firehose 串流時，您必須擁有 IAM 角色。

使用以下存取政策，讓 Amazon Data Firehose 存取您的 S3 儲存貯體、OpenSearch Serverless 網域和 AWS KMS 金鑰。如果您不擁有 S3 儲存貯體，請將新增至 Amazon S3 動作 `s3:PutObjectAcl` 清單，這會授予儲存貯體擁有者對 Amazon Data Firehose 交付物件的完整存取權。此政策亦有允許存取 Amazon Kinesis Data Streams 的陳述式。如果您不使用 Amazon Kinesis Data Streams 作為資料來源，可移除該陳述式。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:AbortMultipartUpload",
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ]
    }
  ]
}
```

```

    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "kms:Decrypt",
      "kms:GenerateDataKey"
    ],
    "Resource": [
      "arn:aws:kms:us-east-1:123456789012:key/key-id"
    ],
    "Condition": {
      "StringEquals": {
        "kms:ViaService": "s3.us-east-1.amazonaws.com"
      },
      "StringLike": {
        "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::amzn-s3-
demo-bucket/prefix*"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "kinesis:DescribeStream",
      "kinesis:GetShardIterator",
      "kinesis:GetRecords",
      "kinesis:ListShards"
    ],
    "Resource": "arn:aws:kinesis:us-east-1:123456789012:stream/stream-name"
  },
  {
    "Effect": "Allow",
    "Action": [
      "logs:PutLogEvents"
    ],
    "Resource": [
      "arn:aws:logs:us-east-1:123456789012:log-group:log-group-name:log-
stream:log-stream-name"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [

```

```

        "lambda:InvokeFunction",
        "lambda:GetFunctionConfiguration"
    ],
    "Resource": [
        "arn:aws:lambda:us-east-1:123456789012:function:function-
name:function-version"
    ]
},
{
    "Effect": "Allow",
    "Action": "aoss:APIAccessAll",
    "Resource": "arn:aws:aoss:us-east-1:123456789012:collection/collection-
id"
}
]
}

```

除了上述政策之外，您還必須將 Amazon Data Firehose 設定為在資料存取政策中指派下列最低許可：

```

[
  {
    "Rules": [
      {
        "ResourceType": "index",
        "Resource": [
          "index/target-collection/target-index"
        ],
        "Permission": [
          "aoss:WriteDocument",
          "aoss:UpdateIndex",
          "aoss:CreateIndex"
        ]
      }
    ],
    "Principal": [
      "arn:aws:sts::123456789012:assumed-role/firehose-delivery-role-name/*"
    ]
  }
]

```

如需允許其他 AWS 服務存取 資源 AWS 的詳細資訊，請參閱《IAM 使用者指南》中的[建立角色以委派許可給 AWS 服務](#)。

授予 Firehose 對 VPC 中 OpenSearch Serverless 目的地的存取權

如果您的 OpenSearch Serverless 集合位於 VPC 中，請務必為 Amazon Data Firehose 提供上一節所述的許可。此外，您需要為 Amazon Data Firehose 提供下列許可，使其能夠存取 OpenSearch Serverless 集合的 VPC。

- ec2:DescribeVpcs
- ec2:DescribeVpcAttribute
- ec2:DescribeSubnets
- ec2:DescribeSecurityGroups
- ec2:DescribeNetworkInterfaces
- ec2:CreateNetworkInterface
- ec2:CreateNetworkInterfacePermission
- ec2>DeleteNetworkInterface

Important

請勿在建立 Firehose 串流後撤銷這些許可。如果您撤銷這些許可，每當服務嘗試查詢或更新 ENIs 時，您的 Firehose 串流將會降級或停止將資料交付到您的 OpenSearch 服務網域。

Important

當您指定子網路以將資料交付至私有 VPC 中的目的地時，請確定您選擇的子網路中有足夠的可用 IP 地址。如果指定的子網路中沒有可用的可用 IP 地址，Firehose 無法為私有 VPC 中的資料交付建立或新增 ENIs，且交付將會降級或失敗。

當您建立或更新 Firehose 串流時，您可以為 Firehose 指定安全群組，以便在將資料傳送至 OpenSearch Serverless 集合時使用。您使用的安全群組可與 OpenSearch Serviceless 集合所使用的相同或不同。如果您指定不同的安全群組，請確認其允許輸出 HTTPS 流量到 OpenSearch Serviceless 集合的安全群組。此外，請確定 OpenSearch Serverless 集合的安全群組允許來自您在設定 Firehose 串流時所指定安全群組的 HTTPS 流量。如果您針對 Firehose 串流和 OpenSearch

Serverless 集合使用相同的安全群組，請確定安全群組傳入規則允許 HTTPS 流量。如需安全群組規則的詳細資訊，請參閱 Amazon VPC 文件中 OpenSearch 的[安全群組規則](#)。

授予 Firehose 存取 Splunk 目的地的權限

當您使用 Splunk 目的地時，Amazon Data Firehose 會將資料交付至 Splunk HTTP 事件收集器 (HEC) 端點。它也會將該資料備份到您指定的 Amazon S3 儲存貯體，而且您可以選擇使用您擁有的 AWS KMS 金鑰進行 Amazon S3 伺服器端加密。如果啟用錯誤記錄，Firehose 會將資料交付錯誤傳送至 CloudWatch 日誌串流。您也可以使用 AWS Lambda 進行資料轉換。

如果您使用 AWS 負載平衡器，請確定它是 Classic Load Balancer 或 Application Load Balancer。此外，針對 Classic Load Balancer 啟用停用 Cookie 過期的持續時間型黏性工作階段，並將過期設定為 Application Load Balancer 的最大 (7 天)。如需如何執行此操作的詳細資訊，請參閱 [Classic Load Balancer](#) 或 [Application Load Balancer](#) 的持續時間型工作階段黏性。

建立 Firehose 串流時，您必須擁有 IAM 角色。Firehose 假設 IAM 角色並存取指定的儲存貯體、金鑰和 CloudWatch 日誌群組和串流。

使用以下存取政策，讓 Amazon Data Firehose 存取您的 S3 儲存貯體。如果您不擁有 S3 儲存貯體，請將新增至 Amazon S3 動作 `s3:PutObjectAcl` 清單，這會授予儲存貯體擁有者對 Amazon Data Firehose 交付物件的完整存取權。此政策也會授予 Amazon Data Firehose 對 CloudWatch 的存取權，以用於錯誤記錄和 AWS Lambda 資料轉換。此政策亦有允許存取 Amazon Kinesis Data Streams 的陳述式。如果您不使用 Amazon Kinesis Data Streams 作為資料來源，可移除該陳述式。Amazon Data Firehose 不會使用 IAM 存取 Splunk。存取 Splunk 時，它會使用您 HEC 符記。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:AbortMultipartUpload",
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:PutObject"
      ],
    },
  ],
}
```

```

    "Resource": [
      "arn:aws:s3:::amzn-s3-demo-bucket",
      "arn:aws:s3:::amzn-s3-demo-bucket/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "kms:Decrypt",
      "kms:GenerateDataKey"
    ],
    "Resource": [
      "arn:aws:kms:us-east-1:123456789012:key/key-id"
    ],
    "Condition": {
      "StringEquals": {
        "kms:ViaService": "s3.us-east-1.amazonaws.com"
      },
      "StringLike": {
        "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::amzn-s3-
demo-bucket/prefix*"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "kinesis:DescribeStream",
      "kinesis:GetShardIterator",
      "kinesis:GetRecords",
      "kinesis:ListShards"
    ],
    "Resource": "arn:aws:kinesis:us-east-1:123456789012:stream/stream-
name"
  },
  {
    "Effect": "Allow",
    "Action": [
      "logs:PutLogEvents"
    ],
    "Resource": [
      "arn:aws:logs:us-east-1:123456789012:log-group:log-group-name:log-
stream:*"
    ]
  }
]

```

```

    },
    {
      "Effect": "Allow",
      "Action": [
        "lambda:InvokeFunction",
        "lambda:GetFunctionConfiguration"
      ],
      "Resource": [
        "arn:aws:lambda:us-east-1:123456789012:function:function-  
name:function-version"
      ]
    }
  ]
}

```

如需允許其他 AWS 服務存取資源 AWS 的詳細資訊，請參閱《IAM 使用者指南》中的[建立角色以委派許可給 AWS 服務](#)。

在 VPC 中存取 Splunk

如果您的 Splunk 平台位在 VPC，則其必須有公有 IP 地址，可供公開存取。此外，透過解除封鎖 Amazon Data Firehose IP 地址，授予 Amazon Data Firehose 存取您的 Splunk 平台。Amazon Data Firehose 目前使用以下 CIDR 區塊。

區域	CIDR 區塊
美國東部 (俄亥俄)	18.216.68.160/27, 18.216.170.64/27, 18.216.170.96/27 \
美國東部 (維吉尼亞北部)	34.238.188.128/26, 34.238.188.192/26, 34.238.195.0/26
美國西部 (加利佛尼亞北部)	13.57.180.0/26
美國西部 (奧勒岡)	34.216.24.32/27, 34.216.24.192/27, 34.216.24.224/27
AWS GovCloud (美國東部)	18.253.138.192/26

區域	CIDR 區塊
AWS GovCloud (美國西部)	52.61.204.192/26
亞太地區 (香港)	18.162.221.64/26
Asia Pacific (Mumbai)	13.232.67.64/26
亞太區域 (首爾)	13.209.71.0/26
亞太區域 (新加坡)	13.229.187.128/26
亞太地區 (雪梨)	13.211.12.0/26
亞太區域 (泰國)	43.208.112.128/26
亞太地區 (東京)	13.230.21.0/27, 13.230.21.32/27
加拿大 (中部)	35.183.92.64/26
加拿大西部 (卡加利)	40.176.98.128/26
歐洲 (法蘭克福)	18.194.95.192/27, 18.194.95.224/27, 18.195.48.0/27
歐洲 (愛爾蘭)	34.241.197.32/27, 34.241.197.64/27, 34.241.197.96/27
歐洲 (倫敦)	18.130.91.0/26
Europe (Paris)	35.180.112.0/26
歐洲 (西班牙)	18.100.194.0/26
歐洲 (斯德哥爾摩)	13.53.191.0/26
Middle East (Bahrain)	15.185.91.64/26
墨西哥 (中部)	78.12.207.64/26

區域	CIDR 區塊
南美洲 (聖保羅)	18.228.1.192/26
歐洲 (米蘭)	15.161.135.192/26
非洲 (開普敦)	13.244.165.128/26
亞太地區 (大阪)	13.208.217.0/26
中國 (北京)	52.81.151.64/26
中國 (寧夏)	161.189.23.128/26
亞太地區 (雅加達)	108.136.221.128/26
中東 (阿拉伯聯合大公國)	3.28.159.64/26
以色列 (特拉維夫)	51.16.102.64/26
歐洲 (蘇黎世)	16.62.183.64/26
亞太地區 (海德拉巴)	18.60.192.192/26
亞太地區 (墨爾本)	16.50.161.192/26
亞太區域 (馬來西亞)	43.216.44.192/26

使用 Amazon Data Firehose 將 VPC 流程日誌擷取至 Splunk

若要進一步了解如何建立 VPC 流程日誌訂閱、發佈至 Firehose，以及將 VPC 流程日誌傳送至支援的目的地，請參閱[使用 Amazon Data Firehose 將 VPC 流程日誌擷取至 Splunk](#)。

存取 Snowflake 或 HTTP 端點

當目的地為 HTTP 端點或 Snowflake 公有叢集時，Amazon Data Firehose 沒有特定的 [AWS IP 地址範圍](#)子集。

若要將 Firehose 新增至公有 Snowflake 叢集或公有 HTTP 或 HTTPS 端點的允許清單，請將所有目前的 [AWS IP 地址範圍](#)新增至您的輸入規則。

Note

通知並非一律來自與其相關聯主題位於相同區域中 AWS 的 IP 地址。您必須包含所有區域的 AWS IP 地址範圍。

授予 Firehose 存取 Snowflake 目的地的權限

當您使用 Snowflake 做為目的地時，Firehose 會使用 Snowflake 帳戶 URL 將資料交付至 Snowflake 帳戶。它也會將錯誤資料備份到您指定的 Amazon Simple Storage Service 儲存貯體，而且您可以選擇使用您擁有的 AWS Key Management Service 金鑰進行 Amazon S3 伺服器端加密。如果啟用錯誤記錄，Firehose 會將資料交付錯誤傳送至 CloudWatch Logs 串流。

您必須先擁有 IAM 角色，才能建立 Firehose 串流。Firehose 假設 IAM 角色並存取指定的儲存貯體、金鑰和 CloudWatch Logs 群組和串流。使用以下存取政策，讓 Firehose 存取您的 S3 儲存貯體。如果您沒有 S3 儲存貯體，請將 `s3:PutObjectAc1` 新增至 Amazon Simple Storage Service 動作清單，這會授予儲存貯體擁有者 Firehose 交付物件的完整存取權。此政策也會授予 Firehose 對 CloudWatch 的存取權，以記錄錯誤。此政策亦有允許存取 Amazon Kinesis Data Streams 的陳述式。如果您不使用 Amazon Kinesis Data Streams 作為資料來源，可移除該陳述式。Firehose 不使用 IAM 存取 Snowflake。為了存取 Snowflake，在私有叢集的情況下，它會使用您的 Snowflake 帳戶 Url 和 PrivateLink Vpce ID。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:AbortMultipartUpload",
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:PutObject"
      ],
      "Resource": [
```

```

        "arn:aws:s3:::amzn-s3-demo-bucket",
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
    ]
},
{
"Effect": "Allow",
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey"
  ],
  "Resource": [
    "arn:aws:kms:us-east-1:123456789012:key/key-id"
  ],
  "Condition": {
"StringEquals": {
"kms:ViaService": "s3.us-east-1.amazonaws.com"
},
  "StringLike": {
"kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::amzn-s3-demo-bucket/prefix*"
}
},
{
"Effect": "Allow",
  "Action": [
    "kinesis:DescribeStream",
    "kinesis:GetShardIterator",
    "kinesis:GetRecords",
    "kinesis:ListShards"
  ],
  "Resource": "arn:aws:kinesis:us-east-1:123456789012:stream/stream-
name"
},
{
"Effect": "Allow",
  "Action": [
    "logs:PutLogEvents"
  ],
  "Resource": [
    "arn:aws:logs:us-east-1:123456789012:log-group:log-group-name:log-
stream:*"
  ]
}
]

```

}

如需允許其他 AWS 服務存取 AWS 資源的詳細資訊，請參閱《IAM 使用者指南》中的[建立角色以委派許可給 AWS 服務](#)。

在 VPC 中存取 Snowflake

如果您的 Snowflake 叢集已啟用私有連結，Firehose 將在建立私有連結時使用以下其中一個 VPC 端點，將資料交付到您的私有叢集，而無需透過公有網際網路。為此，請建立 Snowflake 網路規則，以允許 `AwsVpceIds` AWS 區域 叢集所在的下列傳入。如需詳細資訊，請參閱 Snowflake 使用者指南中的[建立網路規則](#)。

根據叢集所在的區域，要使用的 VPC 端點 ID

AWS 區域	VPCE IDs
美國東部 (俄亥俄)	vpce-0d96cafcd96a50aeb
	vpce-0cec34343d48f537b
美國東部 (維吉尼亞北部)	vpce-0b4d7e8478e141ba8
	vpce-0b75cd681fb507352
	vpce-01c03e63820ec00d8
	vpce-0c2cfc51dc2882422
	vpce-06ca862f019e4e056
	vpce-020cda0cfa63f8d1c
	vpce-0b80504a1a783cd70
	vpce-0289b9ff0b5259a96
	vpce-0d7add8628bd69a12
	vpce-02bfb5966cc59b2af
vpce-09e707674af878bf2	

AWS 區域	VPCE IDs
	vpce-049b52e96cc1a2165
	vpce-0bb6c7b7a8a86cddb
	vpce-03b22d599f51e80f3
	vpce-01d60dc60fc106fe1
	vpce-0186d20a4b24ecbef
	vpce-0533906401a36e416
	vpce-05111fb13d396710e
	vpce-0694613f4fbd6f514
	vpce-09b21cb25fe4cc4f4
	vpce-06029c3550e4d2399
	vpce-00961862a21b033da
	vpce-01620b9ae33273587
	vpce-078cf4ec226880ac9
	vpce-0d711bf076ce56381
	vpce-066b7e13cbfca6f6e
	vpce-0674541252d9ccc26
	vpce-03540b88dedb4b000
	vpce-0b1828e79ad394b95
	vpce-0dc0e6f001fb1a60d
	vpce-0d8f82e71a244098a
	vpce-00e374d9e3f1af5ce

AWS 區域	VPCE IDs
	vpce-0c1e3d6631ddb442f
美國西部 (奧勒岡)	vpce-0f60f72da4cd1e4e7 vpce-0c60d21eb8b1669fd vpce-01c4e3e29afdafbef vpce-0cc6bf2a88da139de vpce-0797e08e169e50662 vpce-033cbe480381b5c0e vpce-00debbdd8f9eb10a5 vpce-08ec2f386c809e889 vpce-0856d14310857b545
歐洲 (法蘭克福)	vpce-068dbb7d71c9460fb vpce-0a7a7f095942d4ec9
歐洲 (愛爾蘭)	vpce-06857e59c005a6276 vpce-04390f4f8778b75f2 vpce-011fd2b1f0a172fd
亞太地區 (東京)	vpce-06369e5258144e68a vpce-0f2363cdb8926fbe8
亞太地區 (新加坡)	vpce-049cd46cce7a12d52 vpce-0e8965a1a4bdb8941
亞太地區 (首爾)	vpce-0aa444d9001e1faa1 vpce-04a49d4dcfd02b884

AWS 區域	VPCE IDs
亞太地區 (悉尼)	vpce-048a60a182c52be63 vpce-03c19949787fd1859
亞太地區 (孟買)	vpce-0d68cb822f6f0db68 vpce-0517d32692ffcbde2
歐洲 (倫敦)	vpce-0fd1874a0ba3b9374 vpce-08091b1a85e206029
南美洲 (聖保羅)	vpce-065169b8144e4f12e vpce-0493699f0e5762d63
加拿大 (中部)	vpce-07e6ed81689d5271f vpce-0f53239730541394c
Europe (Paris)	vpce-09419680077e6488a vpce-0ea81ba2c08140c14
亞太地區 (大阪)	vpce-0a9f003e6a7e38c05 vpce-02886510b897b1c5a
歐洲 (斯德哥爾摩)	vpce-0d96410833219025a vpce-060a32f9a75ba969f
亞太地區 (雅加達)	vpce-00add4b9a25e5c649 vpce-004ae2de34338a856

授予 Firehose 對 HTTP 端點目的地的存取權

您可以使用 Amazon Data Firehose 將資料交付至任何 HTTP 端點目的地。Amazon Data Firehose 也會將該資料備份到您指定的 Amazon S3 儲存貯體，您可以選擇使用 AWS KMS 您擁有的金鑰進行 Amazon S3 伺服器端加密。如果啟用錯誤記錄，Amazon Data Firehose 會將資料交付錯誤傳送至 CloudWatch 日誌串流。您也可以使用 AWS Lambda 進行資料轉換。

建立 Firehose 串流時，您必須擁有 IAM 角色。Amazon Data Firehose 假設 IAM 角色和 可以存取指定的儲存貯體、金鑰和 CloudWatch 日誌群組和串流。

使用以下存取政策，讓 Amazon Data Firehose 存取您為資料備份指定的 S3 儲存貯體。如果您不擁有 S3 儲存貯體，請將新增至 Amazon S3 動作 `s3:PutObjectAcl` 清單，這會授予儲存貯體擁有者對 Amazon Data Firehose 交付物件的完整存取權。此政策也會授予 Amazon Data Firehose 對 CloudWatch 的存取權，以用於錯誤記錄和 AWS Lambda 資料轉換。此政策亦有允許存取 Amazon Kinesis Data Streams 的陳述式。如果您不使用 Amazon Kinesis Data Streams 作為資料來源，可移除該陳述式。

Important

Amazon Data Firehose 不會使用 IAM 存取受支援的第三方服務提供者所擁有的 HTTP 端點目的地，包括 Datadog、Dynatrace、LogicMonitor、MongoDB、New Relic、Splunk 或 Sumo Logic。若要存取受支援的第三方服務提供者擁有的指定 HTTP 端點目的地，請聯絡該服務提供者以取得 API 金鑰或從 Amazon Data Firehose 啟用資料交付至該服務所需的存取金鑰。

如需允許其他 AWS 服務存取 資源 AWS 的詳細資訊，請參閱《IAM 使用者指南》中的 [建立角色以委派許可給 AWS 服務](#)。

Important

目前，Amazon Data Firehose 不支援將資料交付至 VPC 中的 HTTP 端點。

從 Amazon MSK 跨帳戶交付

當您從 Firehose 帳戶（例如帳戶 B）建立 Firehose 串流，且來源是另一個 AWS 帳戶（帳戶 A）中的 MSK 叢集時，您必須具備下列組態。

帳戶 A：

1. 在 Amazon MSK 主控台中，選擇佈建的叢集，然後選擇屬性。
2. 在網路設定下，選擇編輯並開啟多 VPC 連線。
3. 在安全性設定下，選擇編輯叢集。
 - a. 如果叢集尚未設定政策，請勾選包含 Firehose 服務主體和啟用 Firehose 跨帳戶 S3 交付。AWS 管理主控台 會自動產生具有適當許可的政策。
 - b. 如果叢集已設定政策，請將下列許可新增至現有政策：

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::us-east-1:role/mskaasTestDeliveryRole"
  },
  "Action": [
    "kafka:GetBootstrapBrokers",
    "kafka:DescribeCluster",
    "kafka:DescribeClusterV2",
    "kafka-cluster:Connect"
  ],
  "Resource": "arn:aws:kafka:us-east-1:123456789012:cluster/D0-NOT-TOUCH-
mskaas-provisioned-privateLink/xxxxxxxx-2f3a-462a-ba09-xxxxxxxx-20" // ARN
of the cluster
},
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::us-east-1:role/mskaasTestDeliveryRole"
  },
  "Action": [
    "kafka-cluster:DescribeTopic",
    "kafka-cluster:DescribeTopicDynamicConfiguration",
    "kafka-cluster:ReadData"
  ],
  "Resource": "arn:aws:kafka:us-east-1:arn:topic/D0-NOT-TOUCH-mskaas-
provisioned-privateLink/xxxxxxxx-2f3a-462a-ba09-xxxxxxxx-20/*//topic of the
cluster
},
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::us-east-1:role/mskaasTestDeliveryRole"
  },
}
```

```

    "Action": "kafka-cluster:DescribeGroup",
    "Resource": "arn:aws:kafka:us-east-1:arn:group/D0-NOT-TOUCH-mskaas-
provisioned-privateLink/xxxxxxxxx-2f3a-462a-ba09-xxxxxxxxxx-20/*" //topic of
the cluster
  },
}

```

4. 在 AWS 主體下，輸入帳戶 B 的主要 ID。
5. 在主題下，指定您希望 Firehose 串流從中擷取資料的 Apache Kafka 主題。建立 Firehose 串流後，您就無法更新此主題。
6. 選擇 Save changes (儲存變更)

帳戶 B：

1. 在 Firehose 主控台中，選擇使用帳戶 B 建立 Firehose 串流。
2. 在來源下，選擇 Amazon Managed Streaming for Apache Kafka。
3. 在來源設定下，針對 Amazon Managed Streaming for Apache Kafka 叢集，在帳戶 A 中輸入 Amazon MSK 叢集的 ARN。
4. 在主題下，指定您希望 Firehose 串流從中擷取資料的 Apache Kafka 主題。建立 Firehose 串流後，您就無法更新此主題。
5. 在交付串流名稱中，指定 Firehose 串流的名稱。

在帳戶 B 中，當您建立 Firehose 串流時，您必須擁有 IAM 角色（使用時預設建立 AWS 管理主控台），授予 Firehose 串流「讀取」對已設定主題跨帳戶 Amazon MSK 叢集的存取權。

以下是由 AWS 管理主控台設定的內容：

```

{
  "Sid": "",
  "Effect": "Allow",
  "Action": [
    "kafka:GetBootstrapBrokers",
    "kafka:DescribeCluster",
    "kafka:DescribeClusterV2",
    "kafka-cluster:Connect"
  ],
  "Resource": "arn:aws:kafka:us-east-1:arn:aws::cluster/D0-NOT-TOUCH-mskaas-
provisioned-privateLink/xxxxxxxxx-2f3a-462a-ba09-xxxxxxxxxx-20/*" //topic of the
cluster

```

```

    },
    {
      "Sid": "",
      "Effect": "Allow",
      "Action": [
        "kafka-cluster:DescribeTopic",
        "kafka-cluster:DescribeTopicDynamicConfiguration",
        "kafka-cluster:ReadData"
      ],
      "Resource": "arn:aws:kafka:us-east-1:arn:aws::topic/D0-NOT-TOUCH-mskaas-
        provisioned-privateLink/xxxxxxxx-2f3a-462a-ba09-xxxxxxxx-20/mskaas_test_topic" //
        topic of the cluster
    },
    {
      "Sid": "",
      "Effect": "Allow",
      "Action": [
        "kafka-cluster:DescribeGroup"
      ],
      "Resource": "arn:aws:kafka:us-east-1:arn:aws::group/D0-NOT-TOUCH-mskaas-
        provisioned-privateLink/xxxxxxxx-2f3a-462a-ba09-xxxxxxxx-20/*" //topic of the
        cluster
    },
  ]
}

```

接下來，您可以完成設定記錄轉換和記錄格式轉換的選用步驟。如需詳細資訊，請參閱 [\(選用\) 設定記錄轉換和格式轉換](#)。

跨帳戶交付至 Amazon S3 目的地

您可以使用 AWS CLI 或 Amazon Data Firehose APIs，在不同帳戶中具有 Amazon S3 目的地的帳戶中 AWS 建立 Firehose 串流。下列程序顯示設定帳戶 A 擁有的 Firehose 串流，將資料交付至帳戶 B 擁有的 Amazon S3 儲存貯體的範例。

1. 使用 [授予 Firehose 存取 Amazon S3 目的地中所述的步驟](#)，在帳戶 A 下建立 IAM 角色。

Note

存取政策中指定的 Amazon S3 儲存貯體在此範例中為 B 帳戶所有。請務必將 `s3:PutObjectAcl` 新增至存取政策中的 Amazon S3 動作清單，以授予帳戶 B 對

Amazon Data Firehose 交付之物件的完整存取權。跨帳戶交付需要此許可。Amazon Data Firehose 會將請求上的 "x-amz-acl" 標頭設定為 "bucket-owner-full-control"。

- 若要允許先前建立的 IAM 角色存取，請在 B 帳戶下建立 S3 儲存貯體政策。以下程式碼為儲存貯體政策的範例。如需詳細資訊，請參閱[使用儲存貯體政策和使用者政策](#)。

JSON

```
{

  "Version": "2012-10-17",
  "Id": "PolicyID",
  "Statement": [
    {
      "Sid": "StmtID",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/iam-role-name"
      },
      "Action": [
        "s3:AbortMultipartUpload",
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:PutObject",
        "s3:PutObjectAcl"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ]
    }
  ]
}
```

- 使用您在步驟 1 中建立的 IAM 角色，在帳戶 A 下建立 Firehose 串流。

跨帳戶交付至 OpenSearch Service 目的地

您可以使用 AWS CLI 或 Amazon Data Firehose APIs，在不同帳戶中具有 OpenSearch Service 目的地的帳戶中 AWS 建立 Firehose 串流。下列程序顯示如何在帳戶 A 下建立 Firehose 串流，並將其設定為將資料交付至帳戶 B 擁有的 OpenSearch Service 目的地的範例。

1. 使用 [the section called “授予 Firehose 對公有 OpenSearch Service 目的地的存取權”](#) 中所述的步驟，在 A 帳戶下建立 IAM 角色。
2. 若要允許在先前步驟中建立的 IAM 角色存取，請在 B 帳戶下建立 OpenSearch Service 政策。以下 JSON 為範例。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/firehose_delivery_role "
      },
      "Action": "es:ESHttpGet",
      "Resource": [
        "arn:aws:es:us-east-1:123456789012:domain/cross-account-cluster/_all/_settings",
        "arn:aws:es:us-east-1:123456789012:domain/cross-account-cluster/_cluster/stats",
        "arn:aws:es:us-east-1:123456789012:domain/cross-account-cluster/roletest*/_mapping/roletest",
        "arn:aws:es:us-east-1:123456789012:domain/cross-account-cluster/_nodes",
        "arn:aws:es:us-east-1:123456789012:domain/cross-account-cluster/_nodes/stats",
        "arn:aws:es:us-east-1:123456789012:domain/cross-account-cluster/_nodes/*/stats",
        "arn:aws:es:us-east-1:123456789012:domain/cross-account-cluster/_stats",
        "arn:aws:es:us-east-1:123456789012:domain/cross-account-cluster/roletest*/_stats",
        "arn:aws:es:us-east-1:123456789012:domain/cross-account-cluster/"
      ]
    }
  ]
}
```

```
    }  
  ]  
}
```

3. 使用您在步驟 1 中建立的 IAM 角色，在帳戶 A 下建立 Firehose 串流。當您建立 Firehose 串流時，請使用 AWS CLI 或 Amazon Data Firehose APIs，並指定 `ClusterEndpoint` 欄位，而不是 `DomainARN` `OpenSearch Service` 的欄位。

Note

若要在不同 AWS 帳戶中具有 `OpenSearch Service` 目的地的帳戶中建立 Firehose 串流，您必須使用 AWS CLI 或 Amazon Data Firehose APIs。您無法使用 AWS 管理主控台來建立這種跨帳戶組態。

使用標籤控制存取

您可以在 IAM 政策中使用選用 `Condition` 元素（或 `Condition` 區塊），根據標籤索引鍵和值微調對 Amazon Data Firehose 操作的存取。下列子節說明如何針對不同的 Amazon Data Firehose 操作執行此操作。如需使用 `Condition` 元素的詳細資訊，並了解該元素中可利用的運算子，請參閱 [IAM JSON 政策元素：Condition](#)。

CreateDeliveryStream

若是 `CreateDeliveryStream` 操作，請使用 `aws:RequestTag` 條件金鑰。在下方範例中，`MyKey` 與 `MyValue` 分別表示標籤的金鑰和對應值。如需詳細資訊，請參閱 [了解標籤基本概念](#)

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [{  
    "Effect": "Allow",  
    "Action": [  
      "firehose:CreateDeliveryStream",  
      "firehose:TagDeliveryStream"  
    ],  
    "Resource": "*",  
    "Condition": {
```

```
        "StringEquals": {
            "aws:RequestTag/MyKey": "MyValue"
        }
    }
}]
}
```

TagDeliveryStream

若是 TagDeliveryStream 操作，請使用 `aws:TagKeys` 條件金鑰。在下方範例中，MyKey 為標籤金鑰範例。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "firehose:TagDeliveryStream",
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:TagKeys": "MyKey"
        }
      }
    }
  ]
}
```

UntagDeliveryStream

若是 UntagDeliveryStream 操作，請使用 `aws:TagKeys` 條件金鑰。在下方範例中，MyKey 為標籤金鑰範例。

JSON

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

```
"Statement": [
  {
    "Effect": "Allow",
    "Action": "firehose:UntagDeliveryStream",
    "Resource": "*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:TagKeys": "MyKey"
      }
    }
  }
]
```

ListDeliveryStreams

ListDeliveryStreams 無法搭配使用以標籤為基礎的存取控制。

其他操作

對於 CreateDeliveryStream、UntagDeliveryStream、TagDeliveryStream 和 以外的所有 Firehose 操作 ListDeliveryStreams，請使用 aws:RequestTag 條件金鑰。在下方範例中，MyKey 與 MyValue 分別表示標籤的金鑰和對應值。

ListDeliveryStreams，使用 firehose:ResourceTag 條件金鑰根據該 Firehose 串流上的標籤來控制存取。

在下方範例中，MyKey 與 MyValue 分別表示標籤的金鑰和對應值。此政策僅適用於標籤名為 MyKey 且值為 MyValue 的 Data Firehose 串流。如需根據資源標籤控制存取的詳細資訊，請參閱《IAM 使用者指南》中的 [使用標籤控制 AWS 資源的存取](#)。

在 Amazon Data Firehose AWS Secrets Manager 中使用 驗證

Amazon Data Firehose 與 整合 AWS Secrets Manager，以提供秘密的安全存取，並自動輪換登入資料。此整合可讓 Firehose 在執行時間從 Secrets Manager 擷取秘密，以連線至先前提到的串流目的地，並交付您的資料串流。因此，在 AWS 管理主控台 或 API 參數中的串流建立工作流程期間，您的秘密不會以純文字顯示。它提供安全的實務來管理您的秘密，並讓您免於複雜的登入資料管理活動，例如設定自訂 Lambda 函數來管理密碼輪換。

如需詳細資訊，請參閱 [「AWS Secrets Manager 使用者指南」](#)。

主題

- [了解秘密](#)
- [建立秘密](#)
- [使用秘密](#)
- [輪換秘密](#)

了解秘密

秘密可以是以加密形式存放在 Secrets Manager 中的一個密碼、一組憑證，例如使用者名稱和密碼、OAuth 字符或其他秘密資訊。

對於每個目的地，您必須以正確的 JSON 格式指定秘密金鑰值對，如下節所示。如果您的秘密沒有符合目的地的正確 JSON 格式，Amazon Data Firehose 將無法連線至您的目的地。

MySQL 和 PostgreSQL 等資料庫的秘密格式

```
{
  "username": "<username>",
  "password": "<password>"
}
```

Amazon Redshift 佈建叢集和 Amazon Redshift Serverless 工作群組的秘密格式

```
{
  "username": "<username>",
  "password": "<password>"
}
```

Splunk 的秘密格式

```
{
  "hec_token": "<hec token>"
}
```

Snowflake 的秘密格式

```
{
```

```
"user": "<snowflake-username>",
"private_key": "<snowflake-private-key>", // without the beginning and ending
private key, remove all spaces and newlines
"key_passphrase": "<snowflake-private-key-passphrase>" // optional
}
```

HTTP 端

點、Coralogix、Datadog、Dynatrace、Elastic、Honeycomb、LogicMonitor、Logz.io、MongoDB Cloud 和 New Relic 的秘密格式

```
{
  "api_key": "<apikey>"
}
```

建立秘密

若要建立秘密，請遵循AWS Secrets Manager 《使用者指南》中的[建立 AWS Secrets Manager 秘密](#)中的步驟。

使用秘密

我們建議您使用 AWS Secrets Manager 來存放登入資料或金鑰，以連線至串流目的地，例如 Amazon Redshift、HTTP 端點、Snowflake、Splunk、Coralogix、Datadog、Dynatrace、Elastic、Honeycomb、LogicMonitor、Logz.io : //、MongoDB Cloud 和 New Relic。

您可以在建立 Firehose 串流時，AWS 透過 管理主控台設定這些目的地的 Secrets Manager 身分驗證。如需詳細資訊，請參閱[設定目的地設定](#)。或者，您也可以使用 [CreateDeliveryStream](#) 和 [UpdateDestination](#) API 操作來設定 Secrets Manager 的身分驗證。

Firehose 會使用 加密快取秘密，並在每次連線至目的地時使用這些秘密。它會每 10 分鐘重新整理快取一次，以確保使用最新的登入資料。

您可以選擇在串流生命週期內隨時關閉從 Secrets Manager 擷取秘密的功能。如果您不想使用 Secrets Manager 擷取秘密，您可以改用使用者名稱/密碼或 API 金鑰。

Note

雖然 Firehose 中此功能無需額外費用，但您需要支付 Secrets Manager 的存取和維護費用。如需詳細資訊，請參閱 [AWS Secrets Manager 定價頁面](#)。

授予 Firehose 擷取秘密的存取權

若要讓 Firehose 從中擷取秘密 AWS Secrets Manager，您必須向 Firehose 提供存取秘密所需的許可，以及加密秘密的金鑰。

使用 AWS Secrets Manager 存放和擷取秘密時，根據秘密的存放位置和加密方式，有一些不同的組態選項。

- 如果秘密與您的 IAM 角色存放在相同的 AWS 帳戶中，並使用預設 AWS 受管金鑰 (aws/secretsmanager) 加密，Firehose 擔任的 IAM 角色只需要秘密的 `secretsmanager:GetSecretValue` 許可。

```
// secret role policy
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "Secret ARN"
    }
  ]
}
```

如需 IAM 政策的詳細資訊，請參閱 [的許可政策範例 AWS Secrets Manager](#)。

- 如果秘密與角色存放在相同的帳戶中，但使用 [客戶受管金鑰 \(CMK\)](#) 加密，則角色需要 `secretsmanager:GetSecretValue` 和 `kms:Decrypt` 許可。CMK 政策也需要允許 IAM 角色執行 `kms:Decrypt`。
- 如果秘密存放在與您的角色不同的 AWS 帳戶中，並使用預設 AWS 受管金鑰加密，則無法進行此組態，因為當秘密使用 AWS 受管金鑰加密時，Secrets Manager 不允許跨帳戶存取。
- 如果秘密存放在不同的帳戶中並使用 CMK 加密，IAM 角色需要 CMK 秘密和 `kms:Decrypt` 許可的 `secretsmanager:GetSecretValue` 許可。秘密的資源政策和另一個帳戶中的 CMK 政策也需要允許 IAM 角色具有必要的許可。如需詳細資訊，請參閱 [跨帳戶存取](#)。

輪換秘密

當您定期更新秘密時，輪換即為。您可以設定 AWS Secrets Manager，根據您指定的排程自動輪換秘密。如此一來，您可以將長期秘密取代為短期秘密。這有助於降低入侵的風險。如需詳細資訊，請參閱 AWS Secrets Manager 《使用者指南》中的 [輪換 AWS Secrets Manager 秘密](#)。

透過 Amazon Data Firehose 主控台管理 IAM 角色

Amazon Data Firehose 是一項全受管服務，可將即時串流資料交付至目的地。您也可以設定 Firehose 在交付前轉換和轉換資料的格式。若要使用這些功能，您必須先提供 IAM 角色，以便在建立或編輯 Firehose 串流時授予 Firehose 許可。Firehose 會將此 IAM 角色用於 Firehose 串流所需的所有許可。

例如，假設您建立 Firehose 串流將資料交付至 Amazon S3 的案例，而此 Firehose 串流具有啟用 AWS Lambda 功能的轉換來源記錄。在此情況下，您必須提供 IAM 角色，以授予 Firehose 存取 S3 儲存貯體和叫用 Lambda 函數的許可，如下所示。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "lambdaProcessing",
    "Effect": "Allow",
    "Action": ["lambda:InvokeFunction", "lambda:GetFunctionConfiguration"],
    "Resource": "arn:aws:lambda:us-east-1:123456789012:function:<lambda
function name>:<lambda function version>"
  }, {
    "Sid": "s3Permissions",
    "Effect": "Allow",
    "Action": ["s3:AbortMultipartUpload", "s3:GetBucketLocation",
"s3:GetObject", "s3:ListBucket", "s3:ListBucketMultipartUploads",
"s3:PutObject"],
    "Resource": ["arn:aws:s3:::<bucket name>", "arn:aws:s3:::<bucket name>/
*"]
  }
]
```

Firehose 主控台可讓您選擇提供這些角色的方式。您可以選擇下列其中一個選項。

- [選擇現有的 IAM 角色](#)

- [從主控台建立新的 IAM 角色](#)

選擇現有的 IAM 角色

您可以從現有的 IAM 角色中進行選擇。使用此選項，請確定您選擇的 IAM 角色具有來源和目的地所需的適當信任政策和許可。如需詳細資訊，請參閱[使用 Amazon Data Firehose 控制存取](#)。

從主控台建立新的 IAM 角色

或者，您也可以使用 Firehose 主控台代表您建立新的角色。

當 Firehose 代表您建立 IAM 角色時，該角色會自動包含根據 Firehose 串流組態授予所需許可的所有許可和信任政策。

例如，如果您未啟用具有功能的轉換來源記錄 AWS Lambda，則主控台會在許可政策中產生下列陳述式。

```
{
  "Sid": "lambdaProcessing",
  "Effect": "Allow",
  "Action": [
    "lambda:InvokeFunction",
    "lambda:GetFunctionConfiguration"
  ],
  "Resource": "arn:aws:lambda:us-east-1123456789012:function:
%FIREHOSE_POLICY_TEMPLATE_PLACEHOLDER%"
}
```

Note

您可以忽略包含的政策陳述式，%FIREHOSE_POLICY_TEMPLATE_PLACEHOLDER%因為它們不會授予任何資源的許可。

主控台建立和編輯 Firehose 串流工作流程也會建立信任政策，並將其連接到 IAM 角色。信任政策允許 Firehose 擔任 IAM 角色。以下是信任政策的範例。

JSON

```
{
```

```
"Version": "2012-10-17",
"Statement": [{
  "Sid": "firehoseAssume",
  "Effect": "Allow",
  "Principal": {
    "Service": "firehose.amazonaws.com"
  },
  "Action": "sts:AssumeRole"
}]
}
```

Important

- 您應該避免對多個 Firehose 串流使用相同的主控制台受管 IAM 角色。否則，IAM 角色可能會過於寬鬆或導致錯誤。
- 若要從主控制台管理的 IAM 角色使用許可政策中的不同政策陳述式，您可以建立自己的 IAM 角色，並將政策陳述式複製到連接到新角色的許可政策。若要將角色連接至 Firehose 串流，請在服務存取中選取選擇現有的 IAM 角色選項。
- 主控制台會管理在其 ARN 中包含字串服務角色的任何 IAM 角色。當您選擇現有的 IAM 角色選項時，請務必選取在其 ARN 中沒有服務角色字串的 IAM 角色，讓主控制台不會對其進行任何變更。

從主控制台建立 IAM 角色的步驟

1. 開啟位於 <https://console.aws.amazon.com/firehose/> 的 Firehose 主控制台。
2. 選擇建立 Firehose 串流。
3. 選擇來源和目的地。如需詳細資訊，請參閱[教學課程：從主控制台建立 Firehose 串流](#)。
4. 選擇目的地設定。如需詳細資訊，請參閱[設定目的地設定](#)。
5. 在[進階設定](#)下，針對服務存取，選擇建立或更新 IAM 角色。

Note

這是預設選項。若要使用現有角色，請選取選擇現有 IAM 角色選項。Firehose 主控制台不會對您自己的角色進行任何變更。

6. 選擇建立 Firehose 串流。

從主控台編輯 IAM 角色

當您編輯 Firehose 串流時，Firehose 會相應地更新對應的許可政策，以反映組態和許可變更。

例如，當您編輯 Firehose 串流並使用最新版本的 Lambda 函數做為 啟用轉換來源記錄 AWS Lambda 功能時 `exampleLambdaFunction`，您會在 許可政策中取得下列政策陳述式。

```
{
  "Sid": "lambdaProcessing",
  "Effect": "Allow",
  "Action": [
    "lambda:InvokeFunction",
    "lambda:GetFunctionConfiguration"
  ],
  "Resource": "arn:aws:lambda:us-east-1:123456789012:function:exampleLambdaFunction:$LATEST"
}
```

Important

主控台管理的 IAM 角色旨在自主執行。我們不建議您在主控台之外修改許可政策或信任政策。

從主控台編輯 IAM 角色的步驟

1. 開啟位於 <https://console.aws.amazon.com/firehose/> 的 Firehose 主控台。
2. 選擇 Firehose 串流，然後選擇您要更新的 Firehose 串流名稱。
3. 在組態索引標籤的伺服器存取區段中，選擇編輯。
4. 更新 IAM 角色選項。

Note

根據預設，主控台一律會在其 ARN 中使用模式服務角色更新 IAM 角色。當您選擇現有的 IAM 角色選項時，請務必選取 ARN 中沒有服務角色字串的 IAM 角色，讓主控台不會對其進行任何變更。

5. 選擇儲存變更。

了解 Amazon Data Firehose 的合規性

在多個合規計畫中，第三方稽核人員會評估 Amazon Data Firehose 的安全與 AWS 合規。這些計畫包括 SOC、PCI、FedRAMP、HIPAA 等等。

如需特定合規計畫範圍內 AWS 的服務清單，請參閱[AWS 合規計畫範圍內的服務](#)。如需一般資訊，請參閱 [AWS 合規計畫](#)。

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[以 AWS 成品下載報告](#)。

您使用 Data Firehose 時的合規責任取決於資料的機密性、您公司的合規目標，以及適用的法律和法規。如果您使用 Data Firehose 符合 HIPAA、PCI 或 FedRAMP 等標準，AWS 會提供資源來協助：

- [安全與合規快速入門指南](#) – 這些部署指南討論架構考量，並提供在其中部署以安全與合規為重心的基準環境的步驟 AWS。
- [HIPAA 安全與合規架構白皮書](#) – 此白皮書說明公司如何使用 AWS 來建立符合 HIPAA 規範的應用程式。
- [AWS 合規資源](#) – 此工作手冊和指南集合可能適用於您的產業和位置。
- [AWS Config](#) – AWS 此服務會評估資源組態符合內部實務、產業準則和法規的程度。
- [AWS Security Hub CSPM](#) – AWS 此服務提供 內安全狀態的全方位檢視 AWS ，可協助您檢查是否符合安全產業標準和最佳實務。

Amazon Data Firehose 中的彈性

AWS 全球基礎設施是以 AWS 區域和可用區域為基礎建置的。AWS 區域提供多個實體隔離和隔離的可用區域，這些區域以低延遲、高輸送量和高度備援的網路連接。透過可用區域，您所設計與操作的應用程式和資料庫，就能夠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 區域和可用區域的詳細資訊，請參閱 [AWS 全球基礎設施](#)。

除了 AWS 全球基礎設施之外，Data Firehose 還提供數種功能，以協助支援您的資料彈性和備份需求。

災難復原

Amazon Data Firehose 以無伺服器模式執行，並透過執行自動遷移處理主機降級、可用區域可用性和其他基礎設施相關問題。發生這種情況時，Amazon Data Firehose 會確保 Firehose 串流遷移時不會遺失任何資料。

了解 Amazon Data Firehose 中的基礎設施安全性

Amazon Data Firehose 是受管服務，受到 AWS 全球網路安全的保護。如需 AWS 安全服務以及如何 AWS 保護基礎設施的資訊，請參閱[AWS 雲端安全](#)。若要使用基礎設施安全最佳實務來設計您的 AWS 環境，請參閱安全支柱 AWS Well-Architected Framework 中的[基礎設施保護](#)。

您可以使用 AWS 發佈的 API 呼叫，透過網路存取 Firehose。使用者端必須支援下列專案：

- 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。

Note

針對傳出 HTTPS 請求，Amazon Data Firehose 會使用 HTTP 程式庫，自動選取目的地端支援的最高 TLS 通訊協定版本。

搭配 AWS PrivateLink 使用 Amazon Data Firehose

您可以使用介面 VPC 端點 (AWS PrivateLink) 從 VPC 存取 Amazon Data Firehose，而不需要網際網路閘道或 NAT 閘道。介面 VPC 端點不需要網際網路閘道、NAT 裝置、VPN 連線或 Direct Connect 連線。介面 VPC 端點採用 AWS PrivateLink 技術，這項 AWS 技術可讓您在 Amazon VPC 中使用具有私有 IPs 彈性網路介面，在 AWS 服務之間進行私有通訊。如需詳細資訊，請參閱 [Amazon Virtual Private Cloud](#)。

使用 Firehose 的介面 VPC 端點 (AWS PrivateLink)

若要開始使用，請建立介面 VPC 端點，讓來自 Amazon VPC 資源的 Amazon Data Firehose 流量開始流經介面 VPC 端點。建立端點時，您可以將端點政策連接到它，以控制對 Amazon Data Firehose 的存取。如需使用政策控制從 VPC 端點到 Amazon Data Firehose 的存取的詳細資訊，請參閱[使用 VPC 端點控制對服務的存取](#)。

下列範例示範如何在 VPC 中設定 AWS Lambda 函數並建立 VPC 端點，以允許函數與 Amazon Data Firehose 服務安全地通訊。在此範例中，您使用的政策允許 Lambda 函數列出目前區域中的 Firehose 串流，但不描述任何 Firehose 串流。

建立 VPC 端點

1. 登入 AWS 管理主控台 並開啟位於 <https://console.aws.amazon.com/vpc/> 的 Amazon VPC 主控台。
2. 在 VPC 儀表板中選擇 Endpoints (端點)。
3. 選擇建立端點。
4. 在服務名稱清單中，選擇 `com.amazonaws.your_region.kinesis-firehose`。
5. 選擇要在其中建立端點的 VPC 和一或多個子網路。
6. 選擇要與端點關聯的一或多個安全群組。
7. 對於 Policy (政策)，選擇 Custom (自訂) 並貼上以下政策：

```
{
  "Statement": [
    {
      "Sid": "Allow-only-specific-PrivateAPIs",
      "Principal": "*",
      "Action": [
        "firehose:ListDeliveryStreams"
      ],
      "Effect": "Allow",
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "Allow-only-specific-PrivateAPIs",
      "Principal": "*",
      "Action": [
        "firehose:DescribeDeliveryStream"
      ],
      "Effect": "Deny",
      "Resource": [
        "*"
      ]
    }
  ]
}
```

8. 選擇建立端點。

建立要與 Lambda 函數搭配使用的 IAM 角色

1. 前往 <https://console.aws.amazon.com/iam/> 開啟 IAM 主控台。
2. 在左側導覽窗格中，選擇角色，然後選擇建立角色。
3. 在選取可信實體的類型下，保留預設選項 AWS 服務。
4. 在 Choose the service that will use this role (選擇將使用此角色的服務) 下，選擇 Lambda (Lambda)。
5. 選擇 Next: Permissions (下一步：許可)。
6. 在政策清單中，搜尋和新增名為 AWS LambdaVPCAccessExecutionRole 和 AmazonDataFirehoseReadOnlyAccess 的兩個政策。

Important

以下是範例。您可能需要對生產環境採用更嚴格的政策。

7. 選擇下一步：標籤。在本練習中，您不需要新增標籤。選擇下一步：檢閱。
8. 輸入角色名稱，然後選擇建立角色。

在 VPC 中建立 Lambda 函數

1. 在 <https://console.aws.amazon.com/lambda/> 開啟 AWS Lambda 主控台。
2. 選擇 Create function (建立函數)。
3. 選擇從頭開始撰寫。
4. 輸入函數的名稱，然後將執行時間設定為 Python 3.9 或更高。
5. 在 Permissions (許可) 下，展開 Choose or create an execution role (選擇或建立執行角色)。
6. 在 Execution role (執行角色) 清單中，選擇 Use an existing role (使用現有角色)。
7. 在 Existing role (現有角色) 清單中，選擇您以上建立的角色。
8. 選擇建立函數。
9. 在 Function code (函數程式碼) 中貼上以下程式碼。

```
import json
import boto3
import os
from botocore.exceptions import ClientError
```

```
def lambda_handler(event, context):
    REGION = os.environ['AWS_REGION']
    client = boto3.client(
        'firehose',
        REGION

    )
    print("Calling list_delivery_streams with ListDeliveryStreams allowed
policy.")
    delivery_stream_request = client.list_delivery_streams()
    print("Successfully returned list_delivery_streams request %s." % (
        delivery_stream_request
    ))
    describe_access_denied = False
    try:
        print("Calling describe_delivery_stream with DescribeDeliveryStream
denied policy.")
        delivery_stream_info =
client.describe_delivery_stream(DeliveryStreamName='test-describe-denied')
    except ClientError as e:
        error_code = e.response['Error']['Code']
        print ("Caught %s." % (error_code))
        if error_code == 'AccessDeniedException':
            describe_access_denied = True

    if not describe_access_denied:
        raise
    else:
        print("Access denied test succeeded.")
```

10. 在 Basic settings (基本設定) 下，將逾時設定為 1 分鐘。
11. 在 Network (網路) 下，選擇以上端點建立所在的 VPC，然後選擇子網路與您在建立安全群組時與該端點建立關聯的安全群組。
12. 請在頁面頂端附近選擇儲存。
13. 選擇測試。
14. 輸入事件名稱，然後選擇建立。
15. 再次選擇 Test (測試)。這可讓函數得以執行。在執行結果出現後，展開 Details (詳細資訊) 並將日誌輸出與函數程式碼做比較。成功結果會顯示區域中 Firehose 串流的清單，以及下列輸出：

Calling describe_delivery_stream.

AccessDeniedException

Access denied test succeeded.

支援的 AWS 區域

下列區域目前支援介面 VPC 端點。

- 美國東部 (俄亥俄)
- 美國東部 (維吉尼亞北部)
- 美國西部 (加利佛尼亞北部)
- 美國西部 (奧勒岡)
- 亞太區域 (孟買)
- 亞太區域 (首爾)
- 亞太區域 (新加坡)
- 亞太地區 (雪梨)
- 亞太區域 (泰國)
- 亞太地區 (東京)
- 亞太地區 (香港)
- 加拿大 (中部)
- 加拿大西部 (卡加利)
- 中國 (北京)
- 中國 (寧夏)
- 歐洲 (法蘭克福)
- 歐洲 (愛爾蘭)
- 歐洲 (倫敦)
- Europe (Paris)
- 墨西哥 (中部)
- 南美洲 (聖保羅)
- AWS GovCloud (美國東部)
- AWS GovCloud (美國西部)
- 歐洲 (西班牙)

- 中東 (阿拉伯聯合大公國)
- 亞太地區 (雅加達)
- 亞太地區 (大阪)
- 以色列 (特拉維夫)
- 亞太地區 (馬來西亞)

實作 Amazon Data Firehose 的安全最佳實務

Amazon Data Firehose 提供許多安全功能，供您在開發和實作自己的安全政策時考慮。以下最佳實務為一般準則，並不代表完整的安全解決方案。這些最佳實務可能不適用或無法滿足您的環境需求，因此請將其視為實用建議就好，而不要當作是指示。

實作最低權限存取

授予許可時，您可以決定誰取得哪些 Amazon Data Firehose 資源的許可。您還需針對這些資源啟用允許執行的動作，因此，您只應授與執行任務所需的許可。對降低錯誤或惡意意圖所引起的安全風險和影響而言，實作最低權限存取是相當重要的一環。

使用 IAM 角色

生產者和用戶端應用程式必須具有有效的登入資料才能存取 Firehose 串流，而您的 Firehose 串流必須具有有效的登入資料才能存取目的地。您不應將 AWS 登入資料直接存放在用戶端應用程式或 Amazon S3 儲存貯體中。這些是不會自動輪換的長期憑證，如果遭到盜用，可能會對業務造成嚴重的影響。

反之，您應該使用 IAM 角色來管理臨時登入資料，讓您的生產者和用戶端應用程式存取 Firehose 串流。使用角色時，您不必使用長期登入資料 (例如使用者名稱和密碼或存取金鑰) 來存取其他資源。

如需詳細資訊，請參閱《IAM 使用者指南》中的以下主題：

- [IAM 角色](#)
- [常見的角色方案：使用者、應用程式和服務](#)

在相依資源中實作伺服器端加密

靜態資料和傳輸中的資料可以在 Amazon Data Firehose 中加密。如需詳細資訊，請參閱[Amazon Data Firehose 中的資料保護](#)。

使用 CloudTrail 監控 API 呼叫

Amazon Data Firehose 已與 服務整合 AWS CloudTrail，此服務提供 Amazon Data Firehose AWS 中使用者、角色或服務所採取動作的記錄。

您可以使用 CloudTrail 所收集的資訊，判斷向 Amazon Data Firehose 提出的請求、提出請求的 IP 地址、提出請求的人員、提出請求的時間，以及其他詳細資訊。

如需詳細資訊，請參閱[the section called “Log Firehose API 呼叫”](#)。

監控 Amazon Data Firehose

您可以使用下列功能監控 Amazon Data Firehose：

主題

- [使用 CloudWatch 警示實作最佳實務](#)
- [使用 CloudWatch 指標監控 Amazon Data Firehose](#)
- [存取 Amazon Data Firehose 的 CloudWatch 指標](#)
- [使用 CloudWatch Logs 監控 Amazon Data Firehose](#)
- [存取 Amazon Data Firehose 的 CloudWatch 日誌](#)
- [監控 Kinesis 代理程式運作狀態](#)
- [使用 記錄 Amazon Data Firehose API 呼叫 AWS CloudTrail](#)

使用 CloudWatch 警示實作最佳實務

當下列指標超過緩衝限制（最長 15 分鐘）時，新增的 CloudWatch 警示。

- `DeliveryToS3.DataFreshness`
- `DeliveryToIceberg.DataFreshness`
- `DeliveryToSplunk.DataFreshness`
- `DeliveryToAmazonOpenSearchService.DataFreshness`
- `DeliveryToAmazonOpenSearchServerless.DataFreshness`
- `DeliveryToHttpEndpoint.DataFreshness`

此外，根據以下指標數學表達式建立警示。

- $\text{IncomingBytes (Sum per 5 Minutes)} / 300$ 接近 `BytesPerSecondLimit` 的百分比。
- $\text{IncomingRecords (Sum per 5 Minutes)} / 300$ 接近 `RecordsPerSecondLimit` 的百分比。
- $\text{IncomingPutRequests (Sum per 5 Minutes)} / 300$ 接近 `PutRequestsPerSecondLimit` 的百分比。

我們建議為其建立警示的另一個指標是 `ThrottledRecords`。

如需關於警示進入 ALARM 狀態時的故障排除詳細資訊，請參閱[故障診斷錯誤](#)。

使用 CloudWatch 指標監控 Amazon Data Firehose

Important

請務必在屬於您目的地的所有 CloudWatch 指標上啟用警示，以便及時識別錯誤。

Amazon Data Firehose 與 Amazon CloudWatch 指標整合，因此您可以收集、檢視和分析 Firehose 串流的 CloudWatch 指標。例如，您可以監控 IncomingBytes 和 IncomingRecords 指標，以追蹤從資料生產者擷取至 Amazon Data Firehose 的資料。

Amazon Data Firehose 每分鐘收集和發佈 CloudWatch 指標。但是，如果傳入資料的突發僅發生幾秒鐘，則可能無法在一分鐘指標中完全擷取或顯示這些資料。這是因為 CloudWatch 指標會以一分鐘的間隔從 Amazon Data Firehose 彙總。

針對 Firehose 串流收集的指標是免費的。如需 Kinesis 代理程式指標的詳細資訊，請參閱[監控 Kinesis 代理程式運作狀態](#)。

主題

- [動態分割的 CloudWatch 指標](#)
- [資料交付的 CloudWatch 指標](#)
- [資料擷取指標](#)
- [API 層級 CloudWatch 指標](#)
- [資料轉換 CloudWatch 指標](#)
- [CloudWatch Logs 解壓縮指標](#)
- [格式轉換 CloudWatch 指標](#)
- [伺服器端加密 \(SSE\) CloudWatch 指標](#)
- [Amazon Data Firehose 的維度](#)
- [Amazon Data Firehose 用量指標](#)

動態分割的 CloudWatch 指標

如果啟用[動態分割](#)，AWS/Firehose 命名空間會包含下列指標。

指標	Description
ActivePartitionsLimit	<p>Firehose 串流在將資料傳送至錯誤儲存貯體之前處理的作用中分割區數量上限。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
PartitionCount	<p>正在處理的分割區數目，換句話說，使用中的分割區計數。這個數字在 1 和分割區數量上限的 500 (預設值) 之間變化。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
PartitionCountExceeded	<p>此指標表示您是否超過分割數量上限。它根據是否違反限制發出 1 或 0。</p>
JQProcessing.Duration	<p>傳回在 JQ Lambda 函數中執行 JQ 運算式所花費的時間量。</p> <p>單位：毫秒</p>
PerPartitionThroughput	<p>指出每個分割區正在處理的輸送量。此指標可讓您能監控每個分割區的輸送量。</p> <p>單位：StandardUnit.BytesSecond</p>
DeliveryToS3.ObjectCount	<p>表示要交付到 S3 儲存貯體的物件數目。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>

資料交付的 CloudWatch 指標

AWS/Firehose 命名空間包含下列服務層級指標。如果您看到 BackupToS3.Success、DeliveryToS3.Success、DeliveryToSplunk.Success、DeliveryToAmazonS3.Success 或 DeliveryToRedshift.Success 的平均值小幅下降，並不表示資料遺失。Amazon Data Firehose 會重試交付錯誤，在記錄成功交付到設定的目的地或備份 S3 儲存貯體之前，不會繼續。

主題

- [交付至 OpenSearch Service](#)
- [交付至 OpenSearch Serverless](#)
- [交付至 Amazon Redshift](#)
- [交付至 Amazon S3](#)
- [交付至 Snowflake](#)
- [交付至 Splunk](#)
- [交付至 HTTP 端點](#)

交付至 OpenSearch Service

指標	Description
DeliveryToAmazonOpenSearchService.Bytes	<p>在指定期間內，編製索引至 OpenSearch Service 的位元組數目。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：位元組</p>
DeliveryToAmazonOpenSearchService.DataFreshness	<p>Amazon Data Firehose 中最舊記錄的存留期（從進入 Amazon Data Firehose 到現在）。任何比此存留期還要久的記錄都已交付至 OpenSearch Service。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：秒</p>

指標	Description
<code>DeliveryToAmazonOpenSearchService.Records</code>	<p>在指定期間內，編製索引至 OpenSearch Service 的記錄數目。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
<code>DeliveryToAmazonOpenSearchService.Success</code>	成功編製索引的記錄總和。
<code>DeliveryToS3.Bytes</code>	<p>在指定期間內，交付至 Amazon S3 的位元組數目。只有在您為所有文件啟用備份時，Amazon Data Firehose 才會發出此指標。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
<code>DeliveryToS3.DataFreshness</code>	<p>Amazon Data Firehose 中最舊記錄的存留期（從進入 Amazon Data Firehose 到現在）。任何比此存留期還要久的記錄都已交付至 S3 儲存貯體。只有在您為所有文件啟用備份時，Amazon Data Firehose 才會發出此指標。</p> <p>單位：秒</p>
<code>DeliveryToS3.Records</code>	<p>在指定期間內，交付至 Amazon S3 的記錄數目。只有在您為所有文件啟用備份時，Amazon Data Firehose 才會發出此指標。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>

指標	Description
DeliveryToS3.Success	成功的 Amazon S3 put 命令總和。無論僅針對失敗的文件或所有文件啟用備份，Amazon Data Firehose 一律都會發出此指標。
DeliveryToAmazonOpenSearchService.AuthFailure	身分驗證和授權錯誤。驗證 OS/ES 叢集政策和角色許可。 0 表示沒有問題。1 表示驗證失敗。
DeliveryToAmazonOpenSearchService.DeliveryRejected	交付拒絕錯誤。驗證 OS/ES 叢集政策和角色許可。 0 表示沒有問題。1 表示交付失敗。

交付至 OpenSearch Serverless

指標	Description
DeliveryToAmazonOpenSearchServerless.Bytes	在指定期間內，編製索引至 OpenSearch Serviceless 的位元組數目。 統計資訊：Minimum、Maximum、Average、Sum、Samples 單位：位元組
DeliveryToAmazonOpenSearchServerless.DataFreshness	Amazon Data Firehose 中最舊記錄的存留期（從進入 Amazon Data Firehose 到現在）。任何比此存留期還要久的記錄都已交付至 OpenSearch Serviceless。 單位：秒
DeliveryToAmazonOpenSearchServerless.Records	在指定期間內，編製索引至 OpenSearch Serviceless 的記錄數目。 統計資訊：Minimum、Maximum、Average、Sum、Samples

指標	Description
	單位：Count
DeliveryToAmazonOpenSearchServerless.Success	成功編製索引的記錄總和。
DeliveryToS3.Bytes	<p>在指定期間內，交付至 Amazon S3 的位元組數目。只有在您為所有文件啟用備份時，Amazon Data Firehose 才會發出此指標。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
DeliveryToS3.DataFreshness	<p>Amazon Data Firehose 中最舊記錄的存留期（從進入 Amazon Data Firehose 到現在）。任何比此存留期還要久的記錄都已交付至 S3 儲存貯體。只有在您為所有文件啟用備份時，Amazon Data Firehose 才會發出此指標。</p> <p>單位：秒</p>
DeliveryToS3.Records	<p>在指定期間內，交付至 Amazon S3 的記錄數目。只有在您為所有文件啟用備份時，Amazon Data Firehose 才會發出此指標。</p> <p>單位：Count</p>
DeliveryToS3.Success	成功的 Amazon S3 put 命令總和。無論僅針對失敗的文件或所有文件啟用備份，Amazon Data Firehose 一律都會發出此指標。
DeliveryToAmazonOpenSearchServerless.AuthFailure	<p>身分驗證和授權錯誤。驗證 OS/ES 叢集政策和角色許可。</p> <p>0 表示沒有問題。1 表示發生驗證失敗。</p>

指標	Description
DeliveryToAmazonOpenSearchServerless.DeliveryRejected	交付拒絕錯誤。驗證 OS/ES 叢集政策和角色許可。 0 表示沒有問題。1 表示交付失敗。

交付至 Amazon Redshift

指標	Description
DeliveryToRedshift.Bytes	在指定期間內，複製至 Amazon Redshift 的位元組數目。 統計資訊：Minimum、Maximum、Average、Sum、Samples 單位：Count
DeliveryToRedshift.Records	在指定期間內，複製至 Amazon Redshift 的記錄數目。 統計資訊：Minimum、Maximum、Average、Sum、Samples 單位：Count
DeliveryToRedshift.Success	成功的 Amazon Redshift COPY 命令總和。
DeliveryToS3.Bytes	在指定期間內，交付至 Amazon S3 的位元組數目。 統計資訊：Minimum、Maximum、Average、Sum、Samples 單位：位元組
DeliveryToS3.DataFreshness	Amazon Data Firehose 中最舊記錄的存留期（從進入 Amazon Data Firehose 到現在）。任何超過此存留期的記錄都會傳送到 S3 儲存貯體。 單位：秒

指標	Description
DeliveryToS3.Records	<p>在指定期間內，交付至 Amazon S3 的記錄數目。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
DeliveryToS3.Success	成功的 Amazon S3 put 命令總和。
DeliveryToRedshift.DataFreshness	Amazon Data Firehose 中最舊記錄的存留期（從進入 Amazon Data Firehose 到現在）。任何超過此存留期的記錄都會傳送到 Amazon Redshift 叢集。
BackupToS3.Bytes	<p>在指定期間內，交付至 Amazon S3 供備份的位元組數目。啟用備份至 Amazon S3 時，Amazon Data Firehose 會發出此指標。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
BackupToS3.DataFreshness	<p>Amazon Data Firehose 中最舊記錄的存留期（從進入 Amazon Data Firehose 到現在）。任何比此存留期還要久的記錄都已交付至 Amazon S3 儲存貯體供備份。啟用備份至 Amazon S3 時，Amazon Data Firehose 會發出此指標。</p> <p>統計資訊：Minimum、Maximum、Average、Samples</p> <p>單位：秒</p>

指標	Description
BackupToS3.Records	<p>在指定期間內，交付至 Amazon S3 供備份的記錄數目。啟用備份至 Amazon S3 時，Amazon Data Firehose 會發出此指標。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
BackupToS3.Success	<p>用於備份的成功 Amazon S3 put 命令總和。啟用備份至 Amazon S3 時，Amazon Data Firehose 會發出此指標。</p>

交付至 Amazon S3

當 Amazon S3 是 Firehose 串流的主要目的地時，下表中的指標與交付相關。

指標	Description
DeliveryToS3.Bytes	<p>在指定期間內，交付至 Amazon S3 的位元組數目。啟用資料轉換時，此指標會反映轉換前預先處理的位元組大小。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：位元組</p>
DeliveryToS3.DataFreshness	<p>Amazon Data Firehose 中最舊記錄的存留期（從進入 Amazon Data Firehose 到現在）。任何比此存留期還要久的記錄都已交付至 S3 儲存貯體。</p> <p>統計資訊：Minimum、Maximum、Average、Samples</p> <p>單位：秒</p>
DeliveryToS3.Records	<p>在指定期間內，交付至 Amazon S3 的記錄數目。</p>

指標	Description
	<p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
DeliveryToS3.Success	成功的 Amazon S3 put 命令總和。
BackupToS3.Bytes	<p>在指定期間內，交付至 Amazon S3 供備份的位元組數目。啟用備份時，Amazon Data Firehose 會發出此指標（只有在也啟用資料轉換時）。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
BackupToS3.DataFreshness	<p>Amazon Data Firehose 中最舊記錄的存留期（從進入 Amazon Data Firehose 到現在）。任何比此存留期還要久的記錄都已交付至 Amazon S3 儲存貯體供備份。啟用備份時，Amazon Data Firehose 會發出此指標（只有在也啟用資料轉換時）。</p> <p>統計資訊：Minimum、Maximum、Average、Samples</p> <p>單位：秒</p>
BackupToS3.Records	<p>在指定期間內，交付至 Amazon S3 供備份的記錄數目。啟用備份時，Amazon Data Firehose 會發出此指標（只有在也啟用資料轉換時）。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
BackupToS3.Success	用於備份的成功 Amazon S3 put 命令總和。啟用備份時，Amazon Data Firehose 會發出此指標（只有在也啟用資料轉換時）。

交付至 Snowflake

指標	Description
DeliveryToSnowflake.Bytes	<p>在指定期間內交付至 Snowflake 的位元組數。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：位元組</p>
DeliveryToSnowflake.DataFreshness	<p>Firehose 中最舊記錄的年齡（從進入 Firehose 到現在）。任何超過此年齡的記錄都已交付至 Snowflake。請注意，在 Firehose 插入呼叫成功後，可能需要幾秒鐘的時間將資料遞交至 Snowflake。如需將資料遞交至 Snowflake 所需的時間，請參閱 DeliveryToSnowflake.DataCommitLatency 指標。</p> <p>統計資訊：Minimum、Maximum、Average、Samples</p> <p>單位：秒</p>
DeliveryToSnowflake.DataCommitLatency	<p>Firehose 成功插入記錄後，資料遞交至 Snowflake 所需的時間。</p> <p>統計資訊：Minimum、Maximum、Average、Samples</p> <p>單位：秒</p>
DeliveryToSnowflake.Records	<p>在指定期間內交付至 Snowflake 的記錄數目。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
DeliveryToSnowflake.Success	<p>對 Snowflake 成功插入呼叫的總和。</p>

指標	Description
DeliveryToS3.Bytes	<p>在指定期間內，交付至 Amazon S3 的位元組數目。只有在交付至 Snowflake 失敗，且 Firehose 嘗試將失敗的資料備份至 S3 時，才能使用此指標。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：位元組</p>
DeliveryToS3.Records	<p>在指定期間內，交付至 Amazon S3 的記錄數目。只有在交付至 Snowflake 失敗，且 Firehose 嘗試將失敗的資料備份至 S3 時，才能使用此指標。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
DeliveryToS3.Success	<p>成功的 Amazon S3 put 命令總和。只有在交付至 Snowflake 失敗，且 Firehose 嘗試將失敗的資料備份至 S3 時，才能使用此指標。</p>
BackupToS3.DataFreshness	<p>Firehose 中最舊記錄的年齡（從到現在）。任何超過此存留期的記錄都會備份到 Amazon S3 儲存貯體。當 Firehose 串流設定為備份所有資料時，即可使用此指標。</p> <p>統計資訊：Minimum、Maximum、Average、Samples</p> <p>單位：秒</p>
BackupToS3.Records	<p>在指定期間內，交付至 Amazon S3 供備份的記錄數目。當 Firehose 串流設定為備份所有資料時，即可使用此指標。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：計數</p>

指標	Description
BackupToS3.Bytes	<p>在指定期間內，交付至 Amazon S3 供備份的位元組數目。當 Firehose 串流設定為備份所有資料時，即可使用此指標。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：計數</p>
BackupToS3.Success	<p>用於備份的成功 Amazon S3 put 命令總和。Firehose 串流設定為備份所有資料時，Firehose 會發出此指標。</p>

交付至 Splunk

指標	Description
DeliveryToSplunk.Bytes	<p>在指定期間內，交付至 Splunk 的位元組數目。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：位元組</p>
DeliveryToSplunk.DataAckLatency	<p>Amazon Data Firehose 傳送資料後，從 Splunk 接收確認所需的大約持續時間。這項指標的上升趨勢或下降趨勢比絕對約略值更有用。上升趨勢可能表示 Splunk 索引器的索引和確認速度較慢。</p> <p>統計資訊：Minimum、Maximum、Average、Samples</p> <p>單位：秒</p>
DeliveryToSplunk.DataFreshness	<p>Amazon Data Firehose 中最舊記錄的存留期（從進入 Amazon Data Firehose 到現在）。任何比此存留期還要久的記錄都已交付至 Splunk。</p> <p>統計資訊：Minimum、Maximum、Average、Samples</p>

指標	Description
	單位：秒
DeliveryToSplunk.Records	<p>在指定期間內，交付至 Splunk 的記錄數目。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
DeliveryToSplunk.Success	成功編製索引的記錄總和。
DeliveryToS3.Success	成功的 Amazon S3 put 命令總和。啟用備份至 Amazon S3 時，會發出此指標。
BackupToS3.Bytes	<p>在指定期間內，交付至 Amazon S3 供備份的位元組數目。當 Firehose 串流設定為備份所有文件時，Amazon Data Firehose 會發出此指標。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
BackupToS3.DataFreshness	<p>Amazon Data Firehose 中最舊記錄的存留期（從進入 Amazon Data Firehose 到現在）。任何比此存留期還要久的記錄都已交付至 Amazon S3 儲存貯體供備份。當 Firehose 串流設定為備份所有文件時，Amazon Data Firehose 會發出此指標。</p> <p>統計資訊：Minimum、Maximum、Average、Samples</p> <p>單位：秒</p>

指標	Description
BackupToS3.Records	<p>在指定期間內，交付至 Amazon S3 供備份的記錄數目。當 Firehose 串流設定為備份所有文件時，Amazon Data Firehose 會發出此指標。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
BackupToS3.Success	<p>用於備份的成功 Amazon S3 put 命令總和。當 Firehose 串流設定為備份所有文件時，Amazon Data Firehose 會發出此指標。</p>

交付至 HTTP 端點

指標	Description
DeliveryToHttpEndpoint.Bytes	<p>成功交付至 HTTP 端點的位元組數目。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：位元組</p>
DeliveryToHttpEndpoint.Records	<p>成功交付至 HTTP 端點的記錄數目。此指標只會針對成功交付嘗試發出，並且在交付嘗試失敗時不會發出。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：計數</p>
DeliveryToHttpEndpoint.DataFreshness	<p>Amazon Data Firehose 中最舊記錄的存留期。</p> <p>統計資訊：Minimum、Maximum、Average、Samples</p> <p>單位：秒</p>

指標	Description
<code>DeliveryToHttpEndpoint.Success</code>	<p>每次交付嘗試成功交付至 HTTP 端點的記錄數量。與不同 <code>DeliveryToHttpEndpoint.Records</code> ，此指標會在每次交付嘗試時發出。成功時，值等於交付嘗試中的記錄數目。在交付嘗試中的所有記錄失敗時，值為 0。使用 最小統計資料來監控交付失敗。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
<code>DeliveryToHttpEndpoint.ProcessedBytes</code>	嘗試處理的位元組數目，包括重試。
<code>DeliveryToHttpEndpoint.ProcessedRecords</code>	包括重試在內的嘗試記錄數目。

資料擷取指標

主題

- [透過 Kinesis Data Streams 擷取資料](#)
- [透過 Direct PUT 擷取資料](#)
- [從 MSK 擷取資料](#)

透過 Kinesis Data Streams 擷取資料

指標	Description
<code>DataReadFromKinesisStream.Bytes</code>	<p>當資料來源為 Kinesis 資料串流時，此指標指出從該資料串流讀取的位元組數目。此數目包含由於容錯移轉的重新讀取。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p>

指標	Description
	單位：位元組
DataReadFromKinesisStream.Records	<p>當資料來源為 Kinesis 資料串流時，此指標指出從該資料串流讀取的記錄數目。此數目包含由於容錯移轉的重新讀取。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
ThrottledDescribeStream	<p>當資料來源為 Kinesis 資料串流時，調節 DescribeStream 操作的總次數。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
ThrottledGetRecords	<p>當資料來源為 Kinesis 資料串流時，調節 GetRecords 操作的總次數。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
ThrottledGetShardIterator	<p>當資料來源為 Kinesis 資料串流時，調節 GetShardIterator 操作的總次數。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>

指標	Description
KinesisMillisBehindLatest	<p>當資料來源為 Kinesis 資料串流時，這項指標表示上次讀取記錄落後 Kinesis 資料串流中最新記錄的毫秒數。</p> <p>統計資訊：Minimum、Maximum、Average、Samples</p> <p>單位：毫秒</p>

透過 Direct PUT 擷取資料

指標	Description
BackupToS3.Bytes	<p>在指定期間內，交付至 Amazon S3 供備份的位元組數目。當 Amazon S3 或 Amazon Redshift 目的地啟用資料轉換時，Amazon Data Firehose 會發出此指標。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：位元組</p>
BackupToS3.DataFreshness	<p>Amazon Data Firehose 中最舊記錄的存留期（從進入 Amazon Data Firehose 到現在）。任何比此存留期還要久的記錄都已交付至 Amazon S3 儲存貯體供備份。當 Amazon S3 或 Amazon Redshift 目的地啟用資料轉換時，Amazon Data Firehose 會發出此指標。</p> <p>統計資訊：Minimum、Maximum、Average、Samples</p> <p>單位：秒</p>
BackupToS3.Records	<p>在指定期間內，交付至 Amazon S3 供備份的記錄數目。當 Amazon S3 或 Amazon Redshift 目的地啟用資料轉換時，Amazon Data Firehose 會發出此指標。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p>

指標	Description
	單位：Count
BackupToS3.Success	用於備份的成功 Amazon S3 put 命令總和。當 Amazon S3 或 Amazon Redshift 目的地啟用資料轉換時，Amazon Data Firehose 會發出此指標。
BytesPerSecondLimit	Firehose 串流在調節之前可以擷取的目前每秒位元組數上限。若要請求增加此限制，請前往 AWS 支援中心 並選擇建立案例，然後選擇服務配額提升。
DeliveryToAmazonOpenSearchService.Bytes	<p>在指定期間內，編製索引至 OpenSearch Service 的位元組數目。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：位元組</p>
DeliveryToAmazonOpenSearchService.DataFreshness	<p>Amazon Data Firehose 中最舊記錄的存留期（從進入 Amazon Data Firehose 到現在）。任何比此存留期還要久的記錄都已交付至 OpenSearch Service。</p> <p>統計資訊：Minimum、Maximum、Average、Samples</p> <p>單位：秒</p>
DeliveryToAmazonOpenSearchService.Records	<p>在指定期間內，編製索引至 OpenSearch Service 的記錄數目。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
DeliveryToAmazonOpenSearchService.Success	成功編製索引的記錄總和。

指標	Description
DeliveryToRedshift.Bytes	<p>在指定期間內，複製至 Amazon Redshift 的位元組數目。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：位元組</p>
DeliveryToRedshift.Records	<p>在指定期間內，複製至 Amazon Redshift 的記錄數目。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
DeliveryToRedshift.Success	成功的 Amazon Redshift COPY 命令總和。
DeliveryToS3.Bytes	<p>在指定期間內，交付至 Amazon S3 的位元組數目。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：位元組</p>
DeliveryToS3.DataFreshness	<p>Amazon Data Firehose 中最舊記錄的存留期（從進入 Amazon Data Firehose 到現在）。任何比此存留期還要久的記錄都已交付至 S3 儲存貯體。</p> <p>統計資訊：Minimum、Maximum、Average、Samples</p> <p>單位：秒</p>
DeliveryToS3.Records	<p>在指定期間內，交付至 Amazon S3 的記錄數目。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>

指標	Description
<code>DeliveryToS3.Success</code>	成功的 Amazon S3 put 命令總和。
<code>DeliveryToSplunk.Bytes</code>	<p>在指定期間內，交付至 Splunk 的位元組數目。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：位元組</p>
<code>DeliveryToSplunk.DataAckLatency</code>	<p>Amazon Data Firehose 傳送資料後，從 Splunk 接收確認所需的大約持續時間。這項指標的上升趨勢或下降趨勢比絕對約略值更有用。上升趨勢可能表示 Splunk 索引器的索引和確認速度較慢。</p> <p>統計資訊：Minimum、Maximum、Average、Samples</p> <p>單位：秒</p>
<code>DeliveryToSplunk.DataFreshness</code>	<p>Amazon Data Firehose 中最舊記錄的存留期（從進入 Amazon Data Firehose 到現在）。任何比此存留期還要久的記錄都已交付至 Splunk。</p> <p>統計資訊：Minimum、Maximum、Average、Samples</p> <p>單位：秒</p>
<code>DeliveryToSplunk.Records</code>	<p>在指定期間內，交付至 Splunk 的記錄數目。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
<code>DeliveryToSplunk.Success</code>	成功編製索引的記錄總和。

指標	Description
IncomingBytes	<p>在指定期間內成功擷取至 Firehose 串流的位元組數。當資料擷取超過其中一個 Firehose 串流限制時，資料擷取可能會受到調節。限流的資料將不會計算在 IncomingBytes 內。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：位元組</p>
IncomingPutRequests	<p>指定時間段內成功的 PutRecord 和 PutRecordBatch 請求數目。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
IncomingRecords	<p>在指定期間內成功擷取至 Firehose 串流的記錄數目。當資料擷取超過其中一個 Firehose 串流限制時，資料擷取可能會受到調節。限流的資料將不會計算在 IncomingRecords 內。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
RecordsPerSecondLimit	<p>Firehose 串流在調節之前可以擷取的目前每秒記錄數上限。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>

指標	Description
ThrottledRecords	<p>由於資料擷取超過其中一個 Firehose 串流限制而調節的記錄數目。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>

從 MSK 擷取資料

指標	Description
DataReadFromSource.Records	<p>從來源 Kafka 主題讀取的記錄數目。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
DataReadFromSource.Bytes	<p>從來源 Kafka 主題讀取的位元組數目。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：位元組</p>
SourceThrottled.Delay	<p>來源 Kafka 叢集從來源 Kafka 主題傳回記錄的延遲時間。</p> <p>統計資訊：Minimum、Maximum、Average、Samples</p> <p>單位：毫秒</p>
BytesPerSecondLimit	<p>Firehose 將從來源 Kafka 主題的每個分割區讀取的目前輸送量限制。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p>

指標	Description
	單位：位元組/秒
KafkaOffsetLag	<p>Firehose 從來源 Kafka 主題所讀取記錄的最大偏移量與來源 Kafka 主題可用記錄的最大偏移量之間的差異。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
FailedValidation.Records	<p>記錄驗證失敗的記錄數目。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
FailedValidation.Bytes	<p>記錄驗證失敗的位元組數目。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：位元組</p>
DataReadFromSource.Backpressured	<p>表示由於超過每個分割區的 BytesPerSecondLimit 或正常交付流程緩慢或已停止，Firehose 串流延遲從來源分割區讀取記錄</p> <p>單位：布林</p>

API 層級 CloudWatch 指標

AWS/Firehose 命名空間包含下列 API 層級指標。

指標	Description
DescribeDeliveryStream.Latency	每個 DescribeDeliveryStream 操作所需的時間，這是對指定的期間進行測量。

指標	Description
	<p>統計資訊：Minimum、Maximum、Average、Samples</p> <p>單位：毫秒</p>
DescribeDeliveryStream.Requests	<p>DescribeDeliveryStream 請求的總數。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
ListDeliveryStreams.Latency	<p>每個 ListDeliveryStreams 操作所需的時間，這是對指定的期間進行測量。</p> <p>統計資訊：Minimum、Maximum、Average、Samples</p> <p>單位：毫秒</p>
ListDeliveryStreams.Requests	<p>ListFirehose 請求的總數。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
PutRecord.Bytes	<p>在指定期間內使用放入 Firehose 串流 PutRecord 的位元組數。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：位元組</p>
PutRecord.Latency	<p>每個 PutRecord 操作所需的時間，這是對指定的期間進行測量。</p> <p>統計資訊：Minimum、Maximum、Average、Samples</p> <p>單位：毫秒</p>

指標	Description
PutRecord.Requests	<p>PutRecord 請求的總數，它等於來自 PutRecord 操作的總數。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
PutRecordBatch.Bytes	<p>在指定期間內使用放入 Firehose 串流 PutRecordBatch 的位元組數。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：位元組</p>
PutRecordBatch.Latency	<p>每個 PutRecordBatch 操作所需的時間，這是對指定的期間進行測量。</p> <p>統計資訊：Minimum、Maximum、Average、Samples</p> <p>單位：毫秒</p>
PutRecordBatch.Records	<p>來自 PutRecordBatch 操作的記錄總數。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
PutRecordBatch.Requests	<p>PutRecordBatch 請求的總數。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>

指標	Description
PutRequestsPerSecondLimit	<p>Firehose 串流在調節之前每秒可處理的放置請求數量上限。此數字包含 PutRecord 和 PutRecordBatch 請求。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
ThrottledDescribeStream	<p>當資料來源為 Kinesis 資料串流時，調節 DescribeStream 操作的總次數。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
ThrottledGetRecords	<p>當資料來源為 Kinesis 資料串流時，調節 GetRecords 操作的總次數。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
ThrottledGetShardIterator	<p>當資料來源為 Kinesis 資料串流時，調節 GetShardIterator 操作的總次數。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
UpdateDeliveryStream.Latency	<p>每個 UpdateDeliveryStream 操作所需的時間，這是對指定的期間進行測量。</p> <p>統計資訊：Minimum、Maximum、Average、Samples</p> <p>單位：毫秒</p>

指標	Description
UpdateDeliveryStream.Requests	UpdateDeliveryStream 請求的總數。 統計資訊：Minimum、Maximum、Average、Sum、Samples 單位：Count

資料轉換 CloudWatch 指標

如果啟用 Lambda 的資料轉換，AWS/Firehose 命名空間包含下列指標。

指標	Description
ExecuteProcessing.Duration	Firehose 執行的每個 Lambda 函數調用所需的時間。 單位：毫秒
ExecuteProcessing.Success	成功 Lambda 函數調用的總和與 Lambda 函數調用總數之比。
SucceedProcessing.Records	在指定的期間內，成功處理的記錄數目。 單位：Count
SucceedProcessing.Bytes	在指定的期間內，成功處理的位元組記錄數目。 單位：位元組

CloudWatch Logs 解壓縮指標

如果 CloudWatch Logs 交付已啟用解壓縮，AWS/Firehose 命名空間會包含下列指標。

指標	Description
OutputDecompressedBytes.Success	成功解壓縮的資料，以位元組為單位 統計資訊：Minimum、Maximum、Average、Sum、Samples 單位：位元組
OutputDecompressedBytes.Failed	以位元組為單位的解壓縮資料失敗 統計資訊：Minimum、Maximum、Average、Sum、Samples 單位：位元組
OutputDecompressedRecords.Success	成功解壓縮的記錄數量 統計資訊：Minimum、Maximum、Average、Sum、Samples 單位：Count
OutputDecompressedRecords.Failed	解壓縮記錄失敗的數量 統計資訊：Minimum、Maximum、Average、Sum、Samples 單位：Count

格式轉換 CloudWatch 指標

如果啟用格式轉換，AWS/Firehose 命名空間包含下列指標。

指標	Description
SucceedConversion.Records	已成功轉換的記錄數目。 單位：Count

指標	Description
SucceedConversion.Bytes	已成功轉換的記錄大小。 單位：位元組
FailedConversion.Records	無法轉換的記錄數目。 單位：Count
FailedConversion.Bytes	無法轉換的記錄大小。 單位：位元組

伺服器端加密 (SSE) CloudWatch 指標

AWS/Firehose 命名空間包含下列與 SSE 相關的指標。

指標	Description
KMSKeyAccessDenied	服務遇到 Firehose 串流KMSAccessDeniedException 的次數。 統計資訊：Minimum、Maximum、Average、Sum、Samples 單位：Count
KMSKeyDisabled	服務遇到 Firehose 串流KMSDisabledException 的次數。 統計資訊：Minimum、Maximum、Average、Sum、Samples 單位：Count
KMSKeyInvalidState	服務遇到 Firehose 串流KMSInvalidStateException 的次數。

指標	Description
	<p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>
KMSKeyNotFound	<p>服務遇到 Firehose 串流KMSNotFoundException 的次數。</p> <p>統計資訊：Minimum、Maximum、Average、Sum、Samples</p> <p>單位：Count</p>

Amazon Data Firehose 的維度

若要依 Firehose 串流篩選指標，請使用 DeliveryStreamName 維度。

Amazon Data Firehose 用量指標

您可以使用 CloudWatch 用量指標來提供您帳戶的資源用量可見度。使用這些指標，以 CloudWatch 圖表和儀表板視覺化目前的服務使用狀況。

服務配額用量指標位於 AWS/用量命名空間中，每三分鐘收集一次。

目前，CloudWatch 在此命名空間中發佈的唯一指標名稱為 ResourceCount。此指標會與維度 Service、Class、Type 和 Resource 一起發佈。

指標	描述
ResourceCount	<p>您的帳戶中正在執行的特定資源數量。資源由與指標相關聯的維度定義。</p> <p>此指標最有用的統計資料是 MAXIMUM，代表 3 分鐘內使用的資源數量上限。</p>

下列維度用於精簡 Amazon Data Firehose 發佈的用量指標。

維度	Description
Service	包含資源 AWS 的服務名稱。對於 Amazon Data Firehose 用量指標，此維度的值為 Firehose。
Class	正在追蹤的資源類別。Amazon Data Firehose API 用量指標使用此維度，值為 None。
Type	正在追蹤的資源類型。目前，當服務維度為 Firehose 時，類型的唯一有效值為 Resource。
Resource	AWS 資源的名稱。目前，當服務維度為 Firehose 時，資源的唯一有效值為 DeliveryStreams。

存取 Amazon Data Firehose 的 CloudWatch 指標

您可以使用 CloudWatch 主控台、命令列或 CloudWatch API 監控 Amazon Data Firehose 的指標。以下程序將說明如何使用這些不同的方法來存取指標。

使用 CloudWatch 主控台檢視指標

1. 透過 <https://console.aws.amazon.com/cloudwatch/> 開啟 CloudWatch 主控台。
2. 在導覽列上，選擇一個區域。
3. 在導覽窗格中，選擇 指標。
4. 選擇 Firehose (Firehose) 命名空間。
5. 選擇 Firehose 串流指標或 Firehose 指標。
6. 選擇欲新增至圖表的指標。

使用 存取指標 AWS CLI

使用 [list-metrics](#) 和 [get-metric-statistics](#) 命令。

```
aws cloudwatch list-metrics --namespace "AWS/Firehose"
```

```
aws cloudwatch get-metric-statistics --namespace "AWS/Firehose" \  
--metric-name DescribeDeliveryStream.Latency --statistics Average --period 3600 \  

```

```
--start-time 2017-06-01T00:00:00Z --end-time 2017-06-30T00:00:00Z
```

使用 CloudWatch Logs 監控 Amazon Data Firehose

Amazon Data Firehose 與 Amazon CloudWatch Logs 整合，以便在 Lambda 調用資料轉換或資料交付失敗時檢視特定錯誤日誌。您可以在建立 Firehose 串流時啟用 Amazon Data Firehose 錯誤記錄。

如果您在 Amazon Data Firehose 主控台中啟用 Amazon Data Firehose 錯誤記錄，則會代表您為 Firehose 串流建立日誌群組和對應的日誌串流。日誌群組名稱的格式為 `/aws/kinesisfirehose/delivery-stream-name`，其中 *delivery-stream-name* 是對應 Firehose 串流的名稱。DestinationDelivery 是建立的日誌串流，用於記錄與交付至主要目的地相關的任何錯誤。只有在為目的地啟用 S3 備份時，才會建立另一個名為 BackupDelivery 的日誌串流。BackupDelivery 日誌串流可用來記錄與交付至 S3 備份相關的任何錯誤。

例如，如果您使用 Amazon Redshift 建立 Firehose 串流 "MyStream" 做為目的地，並啟用 Amazon Data Firehose 錯誤記錄，則會代表您建立下列項目：名為 `aws/kinesisfirehose/MyStream` 的日誌群組，以及名為 DestinationDelivery 和 BackupDelivery 的兩個日誌串流。在此範例中，DestinationDelivery 將用於記錄與交付到 Amazon Redshift 目的地以及中繼 S3 目的地相關的任何錯誤。如果啟用 S3 備份，BackupDelivery 將用於記錄與交付到 S3 備份儲存貯體相關的任何錯誤。

您可以透過 AWS CLI、API CloudFormation 或使用 CloudWatchLoggingOptions 組態來啟用 Amazon Data Firehose 錯誤記錄。若要這樣做，請事先建立日誌群組和日誌資料流。我們建議僅為 Amazon Data Firehose 錯誤記錄保留該日誌群組和日誌串流。同時，請確認相關 IAM 政策具備 "logs:putLogEvents" 許可。如需詳細資訊，請參閱 [使用 Amazon Data Firehose 控制存取](#)。

請注意，Amazon Data Firehose 不保證所有交付錯誤日誌都會傳送至 CloudWatch Logs。在交付失敗率很高的情況下，Amazon Data Firehose 會在將交付錯誤日誌傳送至 CloudWatch Logs 之前取樣交付錯誤日誌。

傳送錯誤日誌至 CloudWatch Logs 將收取一筆名目費用。如需詳細資訊，請參閱 [Amazon CloudWatch 定價](#)。

目錄

- [資料交付錯誤](#)

資料交付錯誤

以下是每個 Amazon Data Firehose 目的地的資料交付錯誤代碼和訊息清單。每個錯誤訊息亦建議修復問題應採取的適當動作。

錯誤

- [Amazon S3 資料交付錯誤](#)
- [Apache Iceberg 資料表資料交付錯誤](#)
- [Amazon Redshift 資料交付錯誤](#)
- [Snowflake 資料交付錯誤](#)
- [Splunk Data 交付錯誤](#)
- [ElasticSearch 資料交付錯誤](#)
- [HTTPS 端點資料交付錯誤](#)
- [Amazon OpenSearch Service Data 交付錯誤](#)
- [Lambda 調用錯誤](#)
- [Kinesis 叫用錯誤](#)
- [Kinesis DirectPut 調用錯誤](#)
- [AWS Glue 叫用錯誤](#)
- [DataFormatConversion 調用錯誤](#)

Amazon S3 資料交付錯誤

Amazon Data Firehose 可以將下列 S3-related 錯誤傳送至 CloudWatch Logs。

錯誤程式碼	錯誤訊息與資訊
S3.KMS.NotFoundException	「找不到提供的 AWS KMS 金鑰。如果您使用的是您認為是具有正確角色的有效 AWS KMS 金鑰，請檢查 AWS KMS 金鑰連接的帳戶是否有問題。」
S3.KMS.RequestLimitExceeded	「在嘗試加密 S3 物件時，KMS 每秒請求超過限制。請增加每秒請求限制。」 如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的 限制 。

錯誤程式碼	錯誤訊息與資訊
S3.AccessDenied	「存取遭拒。確保所提供 IAM 角色的信任政策允許 Amazon Data Firehose 擔任該角色，且存取政策允許存取 S3 儲存貯體。」
S3.AccountProblem	「AWS 您的帳戶發生問題，導致操作無法成功完成。聯絡 AWS 支援。」
S3.AllAccessDisabled	「所提供的帳戶存取已停用。請聯絡 AWS Support。」
S3.InvalidPayer	「所提供的帳戶存取已停用。請聯絡 AWS Support。」
S3.NotSignedUp	「帳戶未註冊 Amazon S3。請註冊帳戶或使用不同的帳戶。」
S3.NoSuchBucket	「指定的儲存貯體不存在。建立儲存貯體或使用另一現有的儲存貯體。」
S3.MethodNotAllowed	「指定的方法不得使用此資源。請修改儲存貯體的政策，允許正確的 Amazon S3 操作許可。」
InternalError	「嘗試傳輸資料時發生內部錯誤。交付將會重試；如果錯誤仍然存在，則會向回報 AWS 以解決。」
S3.KMS.KeyDisabled	「提供的 KMS 金鑰已停用。啟用金鑰或使用其他金鑰。」
S3.KMS.InvalidStateException	「提供的 KMS 金鑰處於無效狀態。請使用不同的金鑰。」
KMS.InvalidStateException	「提供的 KMS 金鑰處於無效狀態。請使用不同的金鑰。」
KMS.DisabledException	「提供的 KMS 金鑰已停用。請啟用金鑰或使用其他金鑰。」
S3.SlowDown	「指定儲存貯體的 Put 請求率太高。增加 Firehose 串流緩衝區大小或減少來自其他應用程式的放置請求。」

錯誤程式碼	錯誤訊息與資訊
S3.SubscriptionRequired	「呼叫 S3 時存取被拒絕。確保傳入的 IAM 角色和 KMS 金鑰 (如有提供) 具有 Amazon S3 訂閱。」
S3.InvalidToken	「提供的記號格式錯誤或以其他方式無效。請檢查提供的憑證。」
S3.KMS.KeyNotConfigured	「KMS 金鑰未設定。設定您的 KMSPMasterkeyID，或停用 S3 儲存貯體的加密功能。」
S3.KMS.AsymmetricCMKNotSupported	「Amazon S3 支援對稱 CMK。您無法使用非對稱 CMK 來加密 Amazon S3 中的資料。若要取得您的 CMK 類型，請使用 KMS DescribeKey 操作。」
S3.IllegalLocationConstraintException	「Firehose 目前使用 s3 全球端點將資料交付到設定的 s3 儲存貯體。已設定 s3 儲存貯體的區域不支援 s3 全球端點。請在與 s3 儲存貯體相同的區域中建立 Firehose 串流，或在支援全域端點的區域中使用 s3 儲存貯體。」
S3.InvalidPrefixConfigurationException	「用於時間戳記評估的自定義 s3 字首無效。檢查您的 s3 字首包含一年中當前日期和時間的有效運算式。」
DataFormatConversion.MalformedData	「記號之間發現非法字符。」

Apache Iceberg 資料表資料交付錯誤

如需 Apache Iceberg Tables 資料交付錯誤，請參閱 [將資料交付至 Apache Iceberg 資料表](#)。

Amazon Redshift 資料交付錯誤

Amazon Data Firehose 可以將下列 Amazon Redshift 相關錯誤傳送至 CloudWatch Logs。

錯誤程式碼	錯誤訊息與資訊
Redshift. TableNotFound	「找不到欲載入資料的表格。請確認該表格已存在。」 找不到應從 S3 複製至 Amazon Redshift 內的資料表目的地。請注意，如果 Amazon Data Firehose 不存在，則不會建立 Amazon Redshift 資料表。
Redshift. SyntaxError	「COPY 命令包含語法錯誤。請重試命令。」
Redshift. AuthenticationFailed	「所提供的使用者名稱和密碼驗證失敗。請提供有效的使用者名稱和密碼。」
Redshift. AccessDenied	「存取遭拒。確保所提供 IAM 角色的信任政策允許 Amazon Data Firehose 擔任該角色。」
Redshift. S3BucketAccessDenied	「COPY 命令無法存取 S3 儲存貯體。請確認所提供 IAM 角色的存取政策允許存取 S3 儲存貯體。」
Redshift. DataLoadFailed	「載入資料至表格失敗。請檢查 STL_LOAD_ERRORS 系統表格以了解詳細資訊。」
Redshift. ColumnNotFound	「COPY 命令的欄位不存在於表格中。請指定有效的欄位名稱。」
Redshift. DatabaseNotFound	「找不到 Amazon Redshift 目的地設定值中指定的資料庫或 JDBC URL。請指定有效的資料庫名稱。」
Redshift. IncorrectCopyOptions	「所提供的 COPY 選項互相衝突或過多。部分選項不相容於特定組合。請查看 COPY 命令參考資料以取得詳細資訊。」 如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 Amazon Redshift COPY 命令 。

錯誤程式碼	錯誤訊息與資訊
Redshift. MissingColumn	「表格結構描述將欄位定義為無預設值的非空白值，且未包含在欄位清單中。請排除此欄位、確認載入的資料一律提供此欄位的值，或新增此資料表的預設值至 Amazon Redshift 結構描述。」
Redshift. ConnectionFailed	「連線至指定的 Amazon Redshift 叢集失敗。確保安全設定允許 Amazon Data Firehose 連線，Amazon Redshift 目的地組態或 JDBC URL 中指定的叢集或資料庫正確，且叢集可用。」
Redshift. ColumnMismatch	「COPY 命令的 jsonpaths 數量和目的地表格的欄位數量應相符。請重試命令。」
Redshift. IncorrectOrMissingRegion	「Amazon Redshift 嘗試使用錯誤的區域的端點來存取 S3 儲存貯體。請於 COPY 命令選項中指定正確的區域值，或確認 S3 儲存貯體與 Amazon Redshift 資料庫位於相同區域。」
Redshift. IncorrectJsonPathsFile	「所提供的 jsonpaths 檔案未支援 JSON 格式。請重試命令。」
Redshift. MissingS3File	「Amazon Redshift 所需的一個或多個 S3 檔案已自 S3 儲存貯體移除。請檢查 S3 儲存貯體的政策來移除 S3 檔案自動刪除。」
Redshift. InsufficientPrivilege	「使用者不具備將資料載入表格的許可。請檢查 Amazon Redshift INSERT 許可的使用者許可。」
Redshift. ReadOnlyCluster	「無法執行該查詢，因系統正在調整規模。請稍後再嘗試查詢。」
Redshift. DiskFull	「資料無法載入，因為磁碟已滿。請增加 Amazon Redshift 叢集的容量，或刪除未使用的資料以釋放磁碟空間。」
InternalError	「嘗試傳輸資料時發生內部錯誤。交付將重試；如果錯誤仍然存在，則會向 報告 AWS 以解決問題。」

錯誤程式碼	錯誤訊息與資訊
Redshift. ArgumentNotSupported	「COPY 命令包含不支持的選項。」
Redshift. AnalyzeTableAccessDenied	「存取遭拒。從 S3 複製到 Redshift 失敗，因為分析資料表只能由資料表或資料庫擁有者完成。」
Redshift. SchemaNotFound	「找不到 Amazon Redshift 目的地組態的 DataTableName 中指定的結構描述。請指定有效的結構描述名稱。」
Redshift. ColumnSpecifiedMoreThanOnce	「在列列表中指定了一個以上的列。確保刪除重複的列。」
Redshift. ColumnNotNullWithoutDefault	「有一個無預設值的非空白欄位，且未包含在欄位清單中。請確認此類欄位包含在欄位清單中。」
Redshift. IncorrectBucketRegion	「Redshift 嘗試使用與叢集不同區域的儲存貯體。請指定與叢集同一區域的儲存貯體。」
Redshift. S3SlowDown	「S3 的高請求率。降低速率以避免限流。」
Redshift. InvalidCopyOptionForJson	「請將 auto 或有效的 S3 路徑用於 json 複製選項。」

錯誤程式碼	錯誤訊息與資訊
Redshift.InvalidCopyOptionJSONPathFormat	「COPY 失敗，出現錯誤 \"無效的 JSONPath 格式。陣列索引超出範圍\"。請糾正 JSONPath 運算式。」
Redshift.InvalidCopyOptionRBACaclNotAllowed	「COPY 失敗，出現錯誤 \"在未啟用許可傳播時，無法使用 RBAC acl 架構。\"
Redshift.DiskSpaceQuotaExceeded	「由於磁盤空間配額超出，交易中止。釋放磁碟空間或請求提升結構描述的配額。」
Redshift.ConnectionsLimitExceeded	「超出使用者的連線限制。」
Redshift.SslNotSupported	「由於伺服器不支援 SSL，因此與指定的 Amazon Redshift 叢集的連線失敗。請檢查您的叢集設定。」
Redshift.HoseNotFound	「hose 已被刪除。請檢查您的 hose 狀態。」
Redshift.Delimiter	「copyCommand 中的 copyOptions 分隔符號無效。確保它屬於單一字元。」
Redshift.QueryCancelled	「使用者已經取消了 COPY 操作。」
Redshift.CompressionMismatch	「hose 設定為 UNCOMPRESSED，但 copyOption 包括壓縮格式。」

錯誤程式碼	錯誤訊息與資訊
Redshift. EncryptionCredentials	「ENCRYPTED 選項需要以下格式的憑證：'aws_iam_role=...;master_symmetric_key=...' or 'aws_access_key_id=...;aws_secret_access_key=...[;token=...];master_symmetric_key=...'」
Redshift. InvalidCopyOptions	「無效的 COPY 組態選項。」
Redshift. InvalidMessageFormat	「COPY 命令包含無效字符。」
Redshift. TransactionIdLimitReached	「已達交易 ID 限制。」
Redshift. DestinationRemoved	「請確認是否存在 redshift 目的地，並且在 Firehose 組態中已正確設定。」
Redshift. OutOfMemory	「Redshift 叢集的記憶體不足。請確保叢集有足夠的容量。」
Redshift. Cannot Fork Process	「Redshift 叢集的記憶體不足。請確保叢集有足夠的容量。」
Redshift. SslFailure	「握手期間 SSL 連線已關閉。」
Redshift.Resize	「Redshift 叢集正在調整大小。當叢集調整大小時，Firehose 將無法傳送資料。」
Redshift. ImproperQualifiedName	「限定名稱不正確 (帶點的名稱太多)。」

錯誤程式碼	錯誤訊息與資訊
Redshift. InvalidJsonPathFormat	「無效的 JSONPath 格式。」
Redshift. TooManyConnectionsException	「與 Redshift 的連線太多。」
Redshift. PSQLErrorException	「從 Redshift 觀測到的 PSQLErrorException。」
Redshift. DuplicateSecondsSpecification	「以日期/時間格式中的秒數規格重複。」
Redshift. RelationCouldNotBeOpened	「遇到 Redshift 錯誤，無法打開關係。檢查指定資料庫的 Redshift 日誌。」
Redshift. TooManyClients	「從 Redshift 遇到太多客戶端例外狀況。如果有多個生產者同時寫入資料庫，請重新檢視資料庫的連線上限。」

Snowflake 資料交付錯誤

Firehose 可以將下列 Snowflake 相關錯誤傳送至 CloudWatch Logs。

錯誤程式碼	錯誤訊息與資訊
Snowflake .InvalidUrl	「Firehose 無法連線至 Snowflake。請確定已在 Snowflake 目的地組態中正確指定帳戶 URL。」
Snowflake .InvalidUser	「Firehose 無法連線至 Snowflake。請確定已在 Snowflake 目的地組態中正確指定使用者。」

錯誤程式碼	錯誤訊息與資訊
Snowflake .InvalidRole	「指定的 snowflake 角色不存在或未授權。請確定角色已授予指定的使用者」
Snowflake .InvalidTable	「提供的資料表不存在或未授權」
Snowflake .InvalidSchema	「提供的結構描述不存在或未授權」
Snowflake .InvalidDatabase	「提供的資料庫不存在或未授權」
Snowflake .InvalidPrivateKeyOrPassphrase	「指定的私有金鑰或複雜密碼無效。請注意，提供的私有金鑰應該是有效的 PEM RSA 私有金鑰」
Snowflake .MissingColumns	「插入請求會因為輸入承載中缺少資料欄而遭到拒絕。確定為所有不可為空的資料欄指定值」
Snowflake .ExtraColumns	「插入請求會因為額外的資料欄而遭到拒絕。資料表中不存在的資料欄不應指定」
Snowflake .InvalidInput	「由於無效的輸入格式，傳遞失敗。請確定提供的輸入承載為可接受的 JSON 格式」
Snowflake .IncorrectValue	「由於輸入承載中的資料類型不正確，傳遞失敗。確定輸入承載中指定的 JSON 值符合 Snowflake 資料表定義中宣告的資料類型」

Splunk Data 交付錯誤

Amazon Data Firehose 可以將下列 Splunk 相關錯誤傳送至 CloudWatch Logs。

錯誤程式碼	錯誤訊息與資訊
<code>Splunk.ProxyWithoutStickySessions</code>	「如果您在 Amazon Data Firehose 和 HEC 節點之間有代理 (ELB 或其他) , 則必須啟用黏性工作階段以支援 HEC ACKs."
<code>Splunk.DisabledToken</code>	「HEC 符記未啟用。請啟用符記以將資料交付至 Splunk。」
<code>Splunk.InvalidToken</code>	「無效的 HEC 符記。使用有效的 HEC 字符更新 Amazon Data Firehose。」
<code>Splunk.InvalidDataFormat</code>	「資料格式化不正確。欲理解如何正確格式化原生或事件 HEC 端點的資料, 請參閱 Splunk Event Data 。」
<code>Splunk.InvalidIndex</code>	「HEC 符記或輸入的索引無效。請檢查您的索引設定並再試一次。」
<code>Splunk.ServerError</code>	「將資料傳送至 Splunk 失敗, 因為 HEC 節點的伺服器出現錯誤。如果 Amazon Data Firehose 中的重試持續時間大於 0, Amazon Data Firehose 將重試傳送資料。如果所有重試都失敗, Amazon Data Firehose 會將資料備份到 Amazon S3。」
<code>Splunk.DisabledAck</code>	「HEC 符記的 indexer 確認未啟用。請啟用 indexer 確認並再試一次。如需詳細資訊, 請參閱 Enable indexer acknowledgement 。」
<code>Splunk.AckTimeout</code>	「HEC 確認逾時過期之前, 未收到 HEC 的確認。儘管確認逾時, 資料仍可能已成功索引至 Splunk。Amazon Data Firehose 會在確認逾時過期的 Amazon S3 資料中備份。」
<code>Splunk.MaxRetriesFailed</code>	「無法傳送資料至 Splunk, 或無法接收確認。請檢查您的 HEC 運作狀態並再試一次。」
<code>Splunk.ConnectionTimeout</code>	「連線至 Splunk 逾時。這可能是暫時性錯誤, 請求將會重試。如果所有重試都失敗, Amazon Data Firehose 會將資料備份到 Amazon S3。」

錯誤程式碼	錯誤訊息與資訊
Splunk.InvalidEndpoint	「無法連線至 HEC 終端節點。確定 HEC 端點 URL 有效且可從 Amazon Data Firehose 連線。」
Splunk.ConnectionClosed	「無法將資料傳送至 Splunk，因為連線失敗。這可能是暫時性錯誤。增加 Amazon Data Firehose 組態中的重試持續時間可能會防止此類暫時性故障。」
Splunk.SSLUnverified	「無法連線至 HEC 終端節點。主機與同儕提供的憑證不符。請確認該憑證和主機有效。」
Splunk.SSLHandshake	「無法連線至 HEC 終端節點。請確認該憑證和主機有效。」
Splunk.URLNotFound	「在 Splunk 伺服器上找不到請求的 URL。請檢查 Splunk 叢集並確認其設定正確。」
Splunk.ServerError.ContentTooLarge	「傳送至 Splunk 的資料失敗，因為伺服器錯誤顯示 statusCode : 413，訊息：您的使用者端傳送的請求太大。請參閱 SPLUNK 文件以設定 max_content_length。」
Splunk.IndexerBusy	「將資料傳送至 Splunk 失敗，因為 HEC 節點的伺服器出現錯誤。確保 HEC 端點或 Elastic Load Balancer 可連接並且狀態良好。」
Splunk.ConnectionRecycled	「從 Firehose 與 Splunk 的連線已回收。將重試交付。」
Splunk.AcknowledgmentsDisabled	「無法獲得 POST 的確認。確保已在 HEC 端點上啟用了確認。」
Splunk.InvalidHecResponseCharacter	「在 HEC 響應中發現無效字符，請務必檢查服務和 HEC 組態。」

ElasticSearch 資料交付錯誤

Amazon Data Firehose 可以將下列 ElasticSearch 錯誤傳送至 CloudWatch Logs。

錯誤程式碼	錯誤訊息與資訊
ES.AccessDenied	「存取遭拒。確保提供與 Firehose 相關聯的 IAM 角色不會被刪除。」
ES.ResourceNotFound	「指定的 AWS Elasticsearch 網域不存在。」

HTTPS 端點資料交付錯誤

Amazon Data Firehose 可以將下列 HTTP 端點相關錯誤傳送至 CloudWatch Logs。如果這些錯誤都不符合您遇到的問題，則預設錯誤如下：「嘗試傳送資料時發生內部錯誤。交付將重試；如果錯誤仍然存在，則會向 報告 AWS 以解決問題。」

錯誤程式碼	錯誤訊息與資訊
HttpEndpoint.RequestTimeout	在收到回應之前交付逾時，將會重試。如果此錯誤仍然存在，請聯絡 AWS Firehose 服務團隊。
HttpEndpoint.ResponseTooLarge	「從端點收到的回應尺寸過大。請聯絡端點的擁有者以解決此問題。」
HttpEndpoint.InvalidResponseFromDestination	「從指定端點收到的回應無效。請聯絡端點的擁有者以解決此問題。」
HttpEndpoint.DestinationException	「從端點目的地接收到以下回應。」

錯誤程式碼	錯誤訊息與資訊
<code>HttpEndpoint.ConnectionFailed</code>	「無法連線至目的地端點。請聯絡端點的擁有者以解決此問題。」
<code>HttpEndpoint.ConnectionReset</code>	「無法維持與端點的連線。請聯絡端點的擁有者以解決此問題。」
<code>HttpEndpoint.ConnectionReset</code>	「難以維持與端點的連線。請聯絡端點的所有者。」
<code>HttpEndpoint.ResponseReasonPhraseExceededLimit</code>	「從端點收到的回應原因片語超過設定的 64 個字元限制。」
<code>HttpEndpoint.InvalidResponseFromDestination</code>	「從端點收到的回應無效。如需詳細資訊，請參閱 Firehose 文件中的 HTTP 端點疑難排解。原因：「
<code>HttpEndpoint.DestinationException</code>	「交付至端點失敗。如需詳細資訊，請參閱 Firehose 文件中的 HTTP 端點疑難排解。收到狀態碼的回應」
<code>HttpEndpoint.InvalidStatusCode</code>	「收到無效的回應狀態碼。」
<code>HttpEndpoint.SSLHandshakeFailure</code>	「無法與端點完成 SSL 握手。請聯絡端點的擁有者以解決此問題。」

錯誤程式碼	錯誤訊息與資訊
<code>HttpEndpoint.SSLHandshakeFailure</code>	「無法與端點完成 SSL 握手。請聯絡端點的擁有者以解決此問題。」
<code>HttpEndpoint.SSLFailure</code>	「無法與端點完成 TLS 握手。請聯絡端點的擁有者以解決此問題。」
<code>HttpEndpoint.SSLHandshakeCertificatePathFailure</code>	「由於憑證路徑無效，無法與端點完成 SSL 交握。請聯絡端點的擁有者以解決此問題。」
<code>HttpEndpoint.SSLHandshakeCertificatePathValidationFailure</code>	「由於認證路徑驗證失敗，無法與端點完成 SSL 握手。請聯絡端點的擁有者以解決此問題。」
<code>HttpEndpoint.MakeRequestFailure.IllegalUriException</code>	「由於 URI 中輸入無效，HttpEndpoint 請求失敗。請確保輸入 URI 中的所有字符都有效。」
<code>HttpEndpoint.MakeRequestFailure.IllegalCharacterInHeaderValue</code>	「由於非法回應錯誤，HttpEndpoint 請求失敗。標頭值中的非法字元 '\n'。」

錯誤程式碼	錯誤訊息與資訊
HttpEndpoint.IllegalResponseFailure	「由於非法回應錯誤，HttpEndpoint 請求失敗。HTTP 消息不得包含多個內容類型標頭。」
HttpEndpoint.IllegalMessageStart	「由於非法回應錯誤，HttpEndpoint 請求失敗。非法 HTTP 消息啟動。如需詳細資訊，請參閱 Firehose 文件中的 HTTP 端點疑難排解。」

Amazon OpenSearch Service Data 交付錯誤

對於 OpenSearch Service 目的地，Amazon Data Firehose 會在 OpenSearch Service 傳回錯誤時，將錯誤傳送至 CloudWatch Logs。

除了可能從 OpenSearch 叢集傳回的錯誤之外，您可能會遇到下列兩個錯誤：

- 嘗試交付資料至目的地 OpenSearch Service 叢集時，會發生驗證/授權錯誤。這可能是由於 Amazon Data Firehose 目標 OpenSearch Service 網域組態修改時的任何許可問題和/或間歇性發生。請檢查叢集原則和角色許可。
- 由於驗證/授權失敗，資料無法交付至目的地 OpenSearch Service 叢集。這可能是由於 Amazon Data Firehose 目標 OpenSearch Service 網域組態修改時的任何許可問題和/或間歇性發生。請檢查叢集原則和角色許可。

錯誤程式碼	錯誤訊息與資訊
OS.AccessDenied	「存取遭拒。請確認所提供 IAM 角色的信任政策允許 Firehose 擔任此角色，並確認存取政策允許存取 Amazon OpenSearch Service API。」
OS.AccessDenied	「存取遭拒。請確認所提供 IAM 角色的信任政策允許 Firehose 擔任此角色，並確認存取政策允許存取 Amazon OpenSearch Service API。」
OS.AccessDenied	「存取遭拒。確保提供與 Firehose 相關聯的 IAM 角色不會被刪除。」
OS.AccessDenied	「存取遭拒。確保提供與 Firehose 相關聯的 IAM 角色不會被刪除。」

錯誤程式碼	錯誤訊息與資訊
OS.ResourceNotFound	「指定的 Amazon OpenSearch Service 域不存在。」
OS.ResourceNotFound	「指定的 Amazon OpenSearch Service 域不存在。」
OS.AccessDenied	「存取遭拒。請確認所提供 IAM 角色的信任政策允許 Firehose 擔任此角色，並確認存取政策允許存取 Amazon OpenSearch Service API。」
OS.RequestTimeout	「向 Amazon OpenSearch Service 叢集或 OpenSearch Serverless 集合發出的請求逾時。確保叢集或集合具有足夠的容量來容納目前的工作負載。」
OS.ClusterError	「Amazon OpenSearch Service 叢集返回了一個未指定的錯誤。」
OS.RequestTimeout	「向 Amazon OpenSearch Service 叢集發出的請求逾時。確保叢集具有足夠的容量來容納目前的工作負載。」
OS.ConnectionFailed	「無法連線至 Amazon OpenSearch Service 叢集或 OpenSearch Serverless 集合。確保叢集或集合狀態良好且可存取。」
OS.ConnectionReset	「無法保持與 Amazon OpenSearch Service 叢集或 OpenSearch Serverless 集合的連線。請聯絡叢集或集合的擁有者以解決此問題。」
OS.ConnectionReset	「無法維護與 Amazon OpenSearch Service 叢集或 OpenSearch Serverless 集合的連線。確保叢集或集合狀態良好，並且具有足夠的容量來容納目前的工作負載。」
OS.ConnectionReset	「無法維護與 Amazon OpenSearch Service 叢集或 OpenSearch Serverless 集合的連線。確保叢集或集合狀態良好，並且具有足夠的容量來容納目前的工作負載。」
OS.AccessDenied	「存取遭拒。確保 Amazon OpenSearch Service 叢集上的存取政策授予對已設定 IAM 角色的存取權。」

錯誤程式碼	錯誤訊息與資訊
OS.ValidationException	「OpenSearch 叢集傳回 ESServiceException。其中一個原因是叢集已升級到 OS 2.x 或更高版本，但 hose 仍然設定了 TypeName 參數。透過將 TypeName 設定為空白字串，或將端點變更為支援 Type 參數的叢集來更新 hose 組態。」
OS.ValidationException	「成員必須滿足正則運算式模式： <code>[a-z][a-z0-9\\-]+</code> 」
OS.JsonParseException	「Amazon OpenSearch Service 叢集返回 JsonParseException。確保放入的資料是有效的。」
OS.AmazonOpenSearchServiceParseException	「Amazon OpenSearch Service 叢集返回了一個 AmazonOpenSearchServiceParseException。確保放入的資料是有效的。」
OS.ExplicitIndexInBulkNotAllowed	「確保 <code>rest.action.multi.allow_explicit_index</code> 在 Amazon OpenSearch Service 叢集上設定為 <code>true</code> 。」
OS.ClusterError	「Amazon OpenSearch Service 叢集或 OpenSearch Serverless 集合傳回不明錯誤。」
OS.ClusterBlockException	「叢集傳回 ClusterBlockException。它可能會超載。」
OS.InvalidARN	「提供的 Amazon OpenSearch Service ARN 無效。請檢查您的 DeliveryStream 組態。」
OS.MalformedData	「一個或多個記錄格式錯誤。請確保每條記錄都是單個有效的 JSON 對象，並且不包含換行符。」
OS.InternalError	「嘗試交付資料時發生內部錯誤。交付將重試；如果錯誤仍然存在，則會向 報告 AWS 以解決問題。」

錯誤程式碼	錯誤訊息與資訊
OS.AliasWithMultipleIndicesNotAllowed	「別名有一個以上的索引與它相關聯。確保別名只有一個與其關聯的索引。」
OS.UnsupportedVersion	「Amazon Data Firehose 目前不支援 Amazon OpenSearch Service 6.0。如需詳細資訊，請聯絡 AWS Support。」
OS.CharacterConversionException	「一個或多個記錄包含無效字符。」
OS.InvalidDomainNameLength	「域名稱長度不在有效的作業系統限制範圍內。」
OS.VPCDomainNotSupported	「目前不支援虛擬 VPC 內的 Amazon OpenSearch Service 域。」
OS.ConnectionError	「http 伺服器意外關閉了連線，請確認 Amazon OpenSearch Service 叢集或 OpenSearch Serverless 集合的運作狀態。」
OS.LargeFieldData	「Amazon OpenSearch Service 叢集中止了請求，因為它包含的欄位資料大於允許的資料。」
OS.BadGateway	「Amazon OpenSearch Service 叢集或 OpenSearch Serverless 集合中止請求，並回應：502 錯誤的閘道。」
OS.ServiceException	「從 Amazon OpenSearch Service 叢集或 OpenSearch Serverless 集合收到錯誤。如果叢集或集合位於 VPC 後面，請確保網路組態允許連線。」
OS.GatewayTimeout	「Firehose 在連線至 Amazon OpenSearch Service 叢集或 OpenSearch Serverless 集合時遇到逾時錯誤。」

錯誤程式碼	錯誤訊息與資訊
OS.MalformedData	「Amazon Data Firehose 不支援 Firehose 記錄中的 Amazon OpenSearch Service Bulk API 命令。」
OS.ResponseEntryCountMismatch	「來自批量 API 的響應包含的條目多於傳送的記錄數目。確保每條記錄只包含一個 JSON 對象，並且沒有換行符。」

Lambda 調用錯誤

Amazon Data Firehose 可以將下列 Lambda 調用錯誤傳送至 CloudWatch Logs。

錯誤程式碼	錯誤訊息與資訊
Lambda.AssumeRoleAccessDenied	「存取遭拒。確保所提供 IAM 角色的信任政策允許 Amazon Data Firehose 擔任該角色。」
Lambda.InvokeAccessDenied	「存取遭拒。請確認存取政策允許存取 Lambda 函數。」
Lambda.JsonProcessingException	「Lambda 函數回傳記錄時發生錯誤剖析。確保傳回的記錄遵循 Amazon Data Firehose 所需的狀態模型。」 如需詳細資訊，請參閱 資料轉換所需的參數 。
Lambda.InvokeLimitExceeded	「Lambda 並行執行超過限制。請增加並行執行的限制。」 如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 AWS Lambda 限制 。
Lambda.DuplicatedRecordId	「多個回傳記錄具有相同的記錄 ID。請確認 Lambda 函數回傳的每一記錄均有獨特記錄 ID。」 如需詳細資訊，請參閱 資料轉換所需的參數 。

錯誤程式碼	錯誤訊息與資訊
Lambda.MissingRecordId	<p>「一個或多個記錄 ID 未回傳。請確認 Lambda 函數會回傳所有接收到的記錄 ID。」</p> <p>如需詳細資訊，請參閱資料轉換所需的參數。</p>
Lambda.ResourceNotFound	<p>「指定的 Lambda 函數不存在。請使用另一現有的函數。」</p>
Lambda.InvalidSubnetIDException	<p>「Lambda 函數 VPC 設定指定的子網路 ID 無效。請確認子網路 ID 有效。」</p>
Lambda.InvalidSecurityGroupIDException	<p>「Lambda 函數 VPC 設定指定的安全群組 ID 無效。請確認安全群組 ID 有效。」</p>
Lambda.SubnetIPAddressLimitReachedException	<p>「AWS Lambda 無法設定 Lambda 函數的 VPC 存取，因為一或多個設定的子網路沒有可用的 IP 地址。請提高 IP 地址的限制。」</p> <p>如需有關 Amazon VPC 配額的詳細資訊，請參閱《Amazon VPC 使用者指南》中的Amazon VPC 限制 - VPC 和子網路。</p>
Lambda.ENILimitReachedException	<p>「AWS Lambda 無法於 VPC 中建立彈性網路介面 (ENI)，指定為 Lambda 函數組態的一部分，因為已達到網路介面的限制。請提高網路介面的限制。」</p> <p>如需詳細資訊，請參閱《Amazon VPC 使用者指南》https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Appendix_Limits.html#vpc-limits-enis的 Amazon VPC 限制 - 網路介面。</p>
Lambda.FunctionTimedOut	<p>Lambda 函數調用已逾時。增加 Lambda 函數中的「逾時」設定。如需詳細資訊，請參閱設定函數逾時。</p>

錯誤程式碼	錯誤訊息與資訊
Lambda.FunctionError	<p>這可能是由於以下任何錯誤造成的：</p> <ul style="list-style-type: none"> 無效的輸出結構。檢查您的功能，並確保輸出是所需的格式。此外，請確認已處理的記錄包含 Dropped、Ok 或 ProcessingFailed 的有效結果狀態。 已成功調用 Lambda 函數，但傳回錯誤結果。 Lambda 無法解密環境變數，因為 KMS 存取被拒絕。請檢查函數的 KMS 金鑰設定以及金鑰政策。如需詳細資訊，請參閱金鑰存取疑難排解。
Lambda.FunctionRequestTimedOut	<p>叫用 Lambda 時，Amazon Data Firehose 在請求逾時組態錯誤之前遇到請求未完成。重新檢視 Lambda 程式碼，以檢查 Lambda 程式碼是否打算在設定的逾時後執行。如果已就緒，請考慮調整 Lambda 組態設定，包括記憶體、逾時。如需詳細資訊，請參閱設定 Lambda 函數選項。</p>
Lambda.TargetServerFailedToRespond	<p>Amazon Data Firehose 發生錯誤。目標伺服器在呼叫 AWS Lambda 服務時無法回應錯誤。</p>
Lambda.InvalidZipFileException	<p>叫用 Lambda 函數時，Amazon Data Firehose 遇到 InvalidZipFileException。檢查您的 Lambda 函數組態設定和 Lambda 程式碼壓縮檔案。</p>
Lambda.InternalServerError	<p>Amazon Data Firehose 在呼叫 AWS Lambda 服務時遇到 InternalServerError。Amazon Data Firehose 將重試傳送資料固定次數。您可以使用 CreateDeliveryStream 或 UpdateDestination API 來指定或覆寫重試選項。如果錯誤仍然存在，請聯絡 AWS Lambda 支援團隊。</p>
Lambda.ServiceUnavailable	<p>Amazon Data Firehose 在呼叫 AWS Lambda 服務時遇到 ServiceUnavailableException。Amazon Data Firehose 將重試傳送資料固定次數。您可以使用 CreateDeliveryStream 或 UpdateDestination API 來指定或覆寫重試選項。如果錯誤仍然存在，請聯絡 AWS Lambda 支援。</p>

錯誤程式碼	錯誤訊息與資訊
Lambda.InvalidSecurityToken	由於無效的安全記號，無法調用 Lambda 函數。跨分割 Lambda 調用不支援跨分割 Lambda 調用。
Lambda.InvocationFailure	這可能是由於以下任何錯誤造成的： <ul style="list-style-type: none"> Amazon Data Firehose 在呼叫 AWS Lambda 時發生錯誤。操作將重試；若錯誤仍持續出現，將會回報至 AWS 以尋求解決方案。」 Amazon Data Firehose 從 Lambda 遇到 KMSInvalidStateException。Lambda 無法解密環境變數，因為使用的 KMS 金鑰對於解密而言處於無效狀態。請檢查 Lambda 函數的 KMS 金鑰。 Amazon Data Firehose 從 AWS Lambda 遇到 LambdaException。Lambda 無法初始化提供的容器映像。驗證映像。 呼叫 AWS Lambda 時，Amazon Data Firehose 遇到逾時錯誤。支援的函數逾時上限為 5 分鐘。如需詳細資訊，請參閱資料轉換執行持續時間。
Lambda.JsonMappingException	Lambda 函數回傳記錄時發生錯誤剖析。確認資料欄位採用 base-64 編碼。

Kinesis 叫用錯誤

Amazon Data Firehose 可以將下列 Kinesis 調用錯誤傳送至 CloudWatch Logs。

錯誤程式碼	錯誤訊息與資訊
Kinesis.AccessDenied	「呼叫 Kinesis 時存取被拒絕。確保所使用 IAM 角色上的存取政策允許存取適當的 Kinesis API。」
Kinesis.ResourceNotFound	「Firehose 未能從串流中讀取。如果 Firehose 與 Kinesis Stream 相連，則該串流可能不存在，或碎片可能已合併或分割。如果 Firehose 屬於 DirectPut 類型，則 Firehose 可能已經不存在。」

錯誤程式碼	錯誤訊息與資訊
Kinesis.S ubscripti onRequired	「呼叫 Kinesis 時存取被拒絕。確保為 Kinesis 串流存取傳遞的 IAM 角色具有 AWS Kinesis 訂閱。」
Kinesis.T hrottling	「呼叫 Kinesis 時遇到限流錯誤。這可能是由於其他應用程式呼叫與 Firehose 串流相同的 APIs，或因為您建立了太多與來源具有相同 Kinesis 串流的 Firehose 串流。」
Kinesis.T hrottling	「呼叫 Kinesis 時遇到限流錯誤。這可能是由於其他應用程式呼叫與 Firehose 串流相同的 APIs，或因為您建立了太多與來源具有相同 Kinesis 串流的 Firehose 串流。」
Kinesis.A ccessDenied	「呼叫 Kinesis 時存取被拒絕。確保所使用 IAM 角色上的存取政策允許存取適當的 Kinesis API。」
Kinesis.A ccessDenied	「嘗試呼叫基礎 Kinesis Stream 上的 API 操作時，存取遭拒。確保 IAM 角色已傳播且有效。」
Kinesis.K MS.Access DeniedExc eption	「Firehose 無法存取用於加密/解密 Kinesis 串流的 KMS 金鑰。請授予 Firehose 交付角色對金鑰的存取權。」
Kinesis.K MS.KeyDisabled	「Firehose 無法從來源 Kinesis 串流讀取，因為用於加密/解密的 KMS 金鑰已停用。啟用金鑰，以便讀取可以繼續。」
Kinesis.K MS.Invali dStateExc eption	Firehose 無法從來源 Kinesis Streams 讀取，因為用來加密 Kinesis Streams 的 KMS 金鑰處於無效狀態。」
Kinesis.K MS.NotFou ndException	「Firehose 無法從來源 Kinesis 串流讀取，因為找不到用來加密的 KMS 金鑰。」

Kinesis DirectPut 調用錯誤

Amazon Data Firehose 可以將下列 Kinesis DirectPut 調用錯誤傳送至 CloudWatch Logs。

錯誤程式碼	錯誤訊息與資訊
Firehose.KMS.AccessDeniedException	「Firehose 無法存取 KMS 金鑰。請檢查金鑰政策。」
Firehose.KMS.InvalidStateException	「Firehose 無法解密資料，因為用於加密資料的 KMS 金鑰處於無效狀態。」
Firehose.KMS.NotFoundException	「Firehose 無法解密資料，因為找不到用於加密資料的 KMS 金鑰。」
Firehose.KMS.KeyDisabled	「Firehose 無法解密資料，因為用於加密資料的 KMS 金鑰已停用。啟用金鑰，以便可以繼續進行交付。」

AWS Glue 叫用錯誤

Amazon Data Firehose 可以將下列 AWS Glue 調用錯誤傳送至 CloudWatch Logs。

錯誤程式碼	錯誤訊息與資訊
DataFormatConversion.InvalidSchema	「結構描述無效。」
DataFormatConversion	「找不到指定的資料表/資料庫。請確保資料表/資料庫存在，並且結構描述組態中提供的值是正確的，尤其是在大小寫方面。」

錯誤程式碼	錯誤訊息與資訊
on.EntityNotFound	
DataFormatConversion.InvalidInput	「無法從 glue 中找到匹配的模式。請確保具有提供的目錄 ID 的指定資料庫存在。」
DataFormatConversion.InvalidInput	「無法從 glue 中找到匹配的模式。請確保傳遞的 ARN 格式正確。」
DataFormatConversion.InvalidInput	「無法從 glue 中找到匹配的模式。請確保提供的 catalogId 有效。」
DataFormatConversion.InvalidVersionId	「無法從 glue 中找到匹配的模式。請確保資料表的指定版本存在。」
DataFormatConversion.NonExistentColumns	「無法從 glue 中找到匹配的模式。請確保資料表設定了包含目標列的非空儲存描述符。」
DataFormatConversion.AccessDenied	「假設角色時拒絕存取。請確保在資料格式轉換組態中指定的角色已授予 Firehose 服務承擔該角色的許可。」

錯誤程式碼	錯誤訊息與資訊
DataFormatConversion.ThrottledByGlue	「呼叫 Glue 時遇到限流錯誤。可以提高請求速率限制，或降低透過其他應用程式呼叫 Glue 的當前速率。」
DataFormatConversion.AccessDenied	「呼叫 Glue 時存取被拒絕。請確保在資料格式轉換設定中指定的角色具有必要的許可。」
DataFormatConversion.InvalidGlueRole	「無效的角色。請確保在資料格式轉換組態中指定的角色存在。」
DataFormatConversion.InvalidGlueRole	「包含在請求中的安全性記號無效。確保提供與 Firehose 相關聯的 IAM 角色不會被刪除。」
DataFormatConversion.GlueNotAvailableInRegion	「您指定的區域尚未提供AWS Glue；請指定不同的區域。」
DataFormatConversion.GlueEncryptionException	「擷取主金鑰時發生錯誤。確保金鑰存在並具有正確的存取許可。」

錯誤程式碼	錯誤訊息與資訊
DataFormatConversion.SchemaValidationTimeout	「從 Glue 檢索資料表時超時。如果您有大量的 Glue 資料表版本，請新增 'glue:GetTableVersion' 許可 (建議使用) 或刪除未使用的資料表版本。如果您在 Glue 中沒有大量資料表，請聯絡 AWS Support。」
DataFirehose.InternalError	「從 Glue 檢索資料表時超時。如果您有大量的 Glue 資料表版本，請新增 'glue:GetTableVersion' 許可 (建議使用) 或刪除未使用的資料表版本。如果您在 Glue 中沒有大量資料表，請聯絡 AWS Support。」
DataFormatConversion.GlueEncryptionException	「擷取主金鑰時發生錯誤。確保金鑰存在且狀態正確。」

DataFormatConversion 調用錯誤

Amazon Data Firehose 可以將下列 DataFormatConversion 調用錯誤傳送至 CloudWatch Logs。

錯誤程式碼	錯誤訊息與資訊
DataFormatConversion.InvalidSchema	「結構描述無效。」
DataFormatConversion.ValidationException	「欄位名稱和類型必須是非空字符串。」
DataFormatConversion.ParseError	「遇到格式錯誤的 JSON。」

錯誤程式碼	錯誤訊息與資訊
DataFormatConversion.MalformedData	「資料與結構描述不相符。」
DataFormatConversion.MalformedData	「JSON 金鑰的長度不得大於 262144」
DataFormatConversion.MalformedData	「資料無法解碼為 UTF-8。」
DataFormatConversion.MalformedData	「記號之間發現非法字符。」
DataFormatConversion.InvalidTypeFormat	「類型格式無效。檢查類型語法。」
DataFormatConversion.InvalidSchema	「無效的結構描述。請確保資料欄名稱中沒有特殊字元或空格。」
DataFormatConversion.InvalidRecord	「記錄不符合結構描述。一個或多個映射鍵對於映射無效<string, string>。」

錯誤程式碼	錯誤訊息與資訊
DataFormatConversion.MalformedData	「輸入 JSON 在頂層包含一個基本類型。頂層必須是物件或陣列。」
DataFormatConversion.MalformedData	「輸入 JSON 在頂層包含一個基本類型。頂層必須是物件或陣列。」
DataFormatConversion.MalformedData	「記錄為空或僅包含空格。」
DataFormatConversion.MalformedData	「遇到無效字元。」
DataFormatConversion.MalformedData	「遇到無效或不支持的時間戳格式。如需支援的時間戳記格式，請參閱《Firehose 開發人員指南》。」
DataFormatConversion.MalformedData	「在資料中找到了一個標量類型，但在結構描述上指定了一個複雜的類型。」
DataFormatConversion.MalformedData	「資料與結構描述不相符。」

錯誤程式碼	錯誤訊息與資訊
DataFormatConversion.MalformedData	「在資料中找到了一個標量類型，但在結構描述上指定了一個複雜的類型。」
DataFormatConversion.ConversionFailureException	"ConversionFailureException"
DataFormatConversion.DataFormatConversionCustomerErrorException	"DataFormatConversionCustomerErrorException"
DataFormatConversion.DataFormatConversionCustomerErrorException	"DataFormatConversionCustomerErrorException"
DataFormatConversion.MalformedData	「資料與結構描述不相符。」

錯誤程式碼	錯誤訊息與資訊
DataFormatConversion.InvalidSchema	「結構描述無效。」
DataFormatConversion.MalformedData	「資料與結構描述不相符。一個或多個日期的格式無效。」
DataFormatConversion.MalformedData	「資料包含不支援的高度巢狀 JSON 結構。」
DataFormatConversion.EntityNotFound	「找不到指定的資料表/資料庫。請確保資料表/資料庫存在，並且結構描述組態中提供的值是正確的，尤其是在大小寫方面。」
DataFormatConversion.InvalidInput	「無法從 glue 中找到匹配的模式。請確保具有提供的目錄 ID 的指定資料庫存在。」
DataFormatConversion.InvalidInput	「無法從 glue 中找到匹配的模式。請確保傳遞的 ARN 格式正確。」
DataFormatConversion.InvalidInput	「無法從 glue 中找到匹配的模式。請確保提供的 catalogId 有效。」

錯誤程式碼	錯誤訊息與資訊
DataFormatConversion.InvalidVersionId	「無法從 glue 中找到匹配的模式。請確保資料表的指定版本存在。」
DataFormatConversion.NonExistentColumns	「無法從 glue 中找到匹配的模式。請確保資料表設定了包含目標列的非空儲存描述符。」
DataFormatConversion.AccessDenied	「假設角色時拒絕存取。請確保在資料格式轉換組態中指定的角色已授予 Firehose 服務承擔該角色的許可。」
DataFormatConversion.ThrottledByGlue	「呼叫 Glue 時遇到限流錯誤。可以提高請求速率限制，或降低透過其他應用程式呼叫 Glue 的當前速率。」
DataFormatConversion.AccessDenied	「呼叫 Glue 時存取被拒絕。請確保在資料格式轉換設定中指定的角色具有必要的許可。」
DataFormatConversion.InvalidGlueRole	「無效的角色。請確保在資料格式轉換組態中指定的角色存在。」
DataFormatConversion.GlueNotAvailableInRegion	「您指定的區域尚未提供AWS Glue；請指定不同的區域。」

錯誤程式碼	錯誤訊息與資訊
DataFormatConversion.GlueEncryptionException	「擷取主金鑰時發生錯誤。確保金鑰存在並具有正確的存取許可。」
DataFormatConversion.SchemaValidationTimeout	「從 Glue 檢索資料表時超時。如果您有大量的 Glue 資料表版本，請新增 'glue:GetTableVersion' 許可 (建議使用) 或刪除未使用的資料表版本。如果您在 Glue 中沒有大量資料表，請聯絡 AWS Support。」
DataFirehose.InternalError	「從 Glue 檢索資料表時超時。如果您有大量的 Glue 資料表版本，請新增 'glue:GetTableVersion' 許可 (建議使用) 或刪除未使用的資料表版本。如果您在 Glue 中沒有大量資料表，請聯絡 AWS Support。」
DataFormatConversion.MalformedData	「一個或多個欄位的格式不正確。」

存取 Amazon Data Firehose 的 CloudWatch 日誌

您可以使用 Amazon Data Firehose 主控台或 CloudWatch 主控台，檢視與 Amazon Data Firehose 資料交付失敗相關的錯誤日誌。以下程序將說明如何使用下列兩種方法來存取錯誤日誌。

使用 Amazon Data Firehose 主控台存取錯誤日誌

1. 登入 AWS 管理主控台 並開啟 Firehose 主控台，網址為 <https://console.aws.amazon.com/firehose/>
2. 在導覽列上，選擇 AWS 區域。
3. 選擇 Firehose 串流名稱以前往 Firehose 串流詳細資訊頁面。
4. 選擇 Error Log (錯誤日誌)，檢視資料交付失敗相關的錯誤日誌清單。

若使用 CloudWatch 主控台來存取錯誤日誌

1. 透過 <https://console.aws.amazon.com/cloudwatch/> 開啟 CloudWatch 主控台。
2. 在導覽列上，選擇一個區域。
3. 在導覽窗格中，選擇日誌。
4. 選擇日誌群組和日誌串流，以檢視資料交付失敗相關的錯誤日誌清單。

監控 Kinesis 代理程式運作狀態

Kinesis 代理程式會使用 AWS KinesisAgent 命名空間來發佈自訂 CloudWatch 指標。它有助於評估代理程式是否正常運作、依指定將資料提交至 Amazon Data Firehose，以及在資料生產者上耗用適當數量的 CPU 和記憶體資源。

傳送的記錄數量和位元組等指標有助於了解代理程式將資料提交至 Firehose 串流的速率。當這些指標滑落至預期閾值以下特定百分比或滑落至零，可能表示設定有問題、網路出現錯誤或代理程式運作狀態不佳。諸如主機 CPU 和記憶體的消耗量與代理程式錯誤計數器等指標，均顯示資料產生來源的資源使用情況，並提供潛在的設定或主機錯誤等洞見。最後，代理程式亦會記錄服務例外狀況，以協助調查代理程式的問題。

代理程式指標，會在代理程式的 `cloudwatch.endpoint` 的阻態所指定的區域中回報。如需詳細資訊，請參閱[指定代理程式組態設定](#)。

對從多個 Kinesis 代理程式發佈的 Cloudwatch 指標進行彙整或合併。

Kinesis 代理程式發出的指標將收取一筆名目費用，此功能預設為啟用。如需詳細資訊，請參閱[Amazon CloudWatch 定價](#)。

使用 CloudWatch 監控

Kinesis 代理程式會向 CloudWatch 傳送下列指標。

指標	Description
BytesSent	在指定期間內傳送至 Firehose 串流的位元組數。 單位：位元組
RecordSendAttempts	呼叫於指定期間內PutRecordBatch 嘗試的記錄 (不論第一次嘗試或是重試) 數量。

指標	Description
	單位：Count
RecordSendErrors	呼叫於指定期間內PutRecordBatch 回傳失敗狀態的記錄 (包括重試) 數量。 單位：Count
ServiceErrors	呼叫於指定期間內PutRecordBatch 導致服務錯誤 (調節錯誤除外) 數量。 單位：Count

使用 記錄 Amazon Data Firehose API 呼叫 AWS CloudTrail

Amazon Data Firehose 已與 服務整合 AWS CloudTrail，此服務提供 Amazon Data Firehose AWS 中使用者、角色或服務所採取動作的記錄。CloudTrail 會將 Amazon Data Firehose 的所有 API 呼叫擷取為事件。擷取的呼叫包括來自 Amazon Data Firehose 主控台的呼叫，以及對 Amazon Data Firehose API 操作的程式碼呼叫。如果您建立線索，則可以將 CloudTrail 事件持續交付至 Amazon S3 儲存貯體，包括 Amazon Data Firehose 的事件。即使您未設定追蹤，依然可以透過 CloudTrail 主控台的事件歷史記錄檢視最新事件。您可以使用 CloudTrail 所收集的資訊，判斷向 Amazon Data Firehose 提出的請求、提出請求的 IP 地址、提出請求的人員、提出請求的時間，以及其他詳細資訊。

若要進一步了解 CloudTrail，包括如何設定及啟用，請參閱 [《AWS CloudTrail 使用者指南》](#)。

CloudTrail 中的 Firehose 資訊

當您建立 AWS 帳戶時，會在您的帳戶上啟用 CloudTrail。當 Amazon Data Firehose 中發生支援的事件活動時，該活動會與事件歷史記錄中的其他 AWS 服務事件一起記錄在 CloudTrail 事件中。您可以檢視、搜尋和下載 AWS 帳戶的最新事件。如需詳細資訊，請參閱 [使用 CloudTrail 事件歷史記錄檢視事件](#)。

若要持續記錄您 AWS 帳戶中的事件，包括 Amazon Data Firehose 的事件，請建立追蹤。線索能讓 CloudTrail 將日誌檔案交付至 Amazon S3 儲存貯體。根據預設，當您在主控台中建立線索時，線索會套用至所有 AWS 區域。線索會記錄 AWS 分割區中所有區域的事件，並將日誌檔案交付至您指定的 Amazon S3 儲存貯體。此外，您可以設定其他 AWS 服務，以進一步分析和處理 CloudTrail 日誌中所收集的事件資料。如需詳細資訊，請參閱下列內容：

- [建立追蹤的概觀](#)
- [CloudTrail 支援的服務和整合](#)
- [設定 CloudTrail 的 Amazon SNS 通知](#)
- [從多個區域接收 CloudTrail 日誌檔案](#)，以及[從多個帳戶接收 CloudTrail 日誌檔案](#)

Amazon Data Firehose 支援將下列動作記錄為 CloudTrail 日誌檔案中的事件：

- [CreateDeliveryStream](#)
- [DeleteDeliveryStream](#)
- [DescribeDeliveryStream](#)
- [ListDeliveryStreams](#)
- [ListTagsForDeliveryStream](#)
- [TagDeliveryStream](#)
- [StartDeliveryStreamEncryption](#)
- [StopDeliveryStreamEncryption](#)
- [UntagDeliveryStream](#)
- [UpdateDestination](#)

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：

- 請求是否使用根還是 AWS Identity and Access Management (IAM) 使用者登入資料提出。
- 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 userIdentity 元素](#)。

範例：Firehose 日誌檔案項目

追蹤是一種組態，能讓事件以日誌檔案的形式交付到您指定的 Amazon S3 儲存貯體。CloudTrail 日誌檔案包含一或多個日誌專案。一個事件為任何來源提出的單一請求，並包含請求動作、請求的日期和時間、請求參數等資訊。CloudTrail 日誌檔並非依公有 API 呼叫的堆疊追蹤排序，因此不會以任何特定順序出現。

以下範例顯示的是展示

CreateDeliveryStream、DescribeDeliveryStream、ListDeliveryStreams、UpdateDestination及 DeleteDeliveryStream 動作的 CloudTrail 日誌項目。

```
{
  "Records": [
    {
      "eventVersion": "1.02",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "AKIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/CloudTrail_Test_User",
        "accountId": "111122223333",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "userName": "CloudTrail_Test_User"
      },
      "eventTime": "2016-02-24T18:08:22Z",
      "eventSource": "firehose.amazonaws.com",
      "eventName": "CreateDeliveryStream",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "127.0.0.1",
      "userAgent": "aws-internal/3",
      "requestParameters": {
        "deliveryStreamName": "TestRedshiftStream",
        "redshiftDestinationConfiguration": {
          "s3Configuration": {
            "compressionFormat": "GZIP",
            "prefix": "prefix",
            "bucketARN": "arn:aws:s3:::amzn-s3-demo-bucket",
            "roleARN": "arn:aws:iam::111122223333:role/Firehose",
            "bufferingHints": {
              "sizeInMBs": 3,
              "intervalInSeconds": 900
            }
          },
          "encryptionConfiguration": {
            "kMSEncryptionConfig": {
              "aWSKMSKeyARN": "arn:aws:kms:us-east-1:key"
            }
          }
        }
      },
      "clusterJDBCURL": "jdbc:redshift://example.abc123.us-west-2.redshift.amazonaws.com:5439/dev",
      "copyCommand": {
```

```

        "copyOptions":"copyOptions",
        "dataTableName":"dataTable"
    },
    "password":"",
    "username":"",
    "roleARN":"arn:aws:iam::111122223333:role/Firehose"
}
},
"responseElements":{
    "deliveryStreamARN":"arn:aws:firehose:us-
east-1:111122223333:deliverystream/TestRedshiftStream"
},
"requestID":"958abf6a-db21-11e5-bb88-91ae9617edf5",
"eventID":"875d2d68-476c-4ad5-bbc6-d02872cfc884",
"eventType":"AwsApiCall",
"recipientAccountId":"111122223333"
},
{
    "eventVersion":"1.02",
    "userIdentity":{
        "type":"IAMUser",
        "principalId":"AKIAIOSFODNN7EXAMPLE",
        "arn":"arn:aws:iam::111122223333:user/CloudTrail_Test_User",
        "accountId":"111122223333",
        "accessKeyId":"AKIAI44QH8DHBEXAMPLE",
        "userName":"CloudTrail_Test_User"
    },
    "eventTime":"2016-02-24T18:08:54Z",
    "eventSource":"firehose.amazonaws.com",
    "eventName":"DescribeDeliveryStream",
    "awsRegion":"us-east-1",
    "sourceIPAddress":"127.0.0.1",
    "userAgent":"aws-internal/3",
    "requestParameters":{
        "deliveryStreamName":"TestRedshiftStream"
    },
    "responseElements":null,
    "requestID":"aa6ea5ed-db21-11e5-bb88-91ae9617edf5",
    "eventID":"d9b285d8-d690-4d5c-b9fe-d1ad5ab03f14",
    "eventType":"AwsApiCall",
    "recipientAccountId":"111122223333"
},
{
    "eventVersion":"1.02",

```

```

    "userIdentity":{
      "type":"IAMUser",
      "principalId":"AKIAIOSFODNN7EXAMPLE",
      "arn":"arn:aws:iam::111122223333:user/CloudTrail_Test_User",
      "accountId":"111122223333",
      "accessKeyId":"AKIAI44QH8DHBEXAMPLE",
      "userName":"CloudTrail_Test_User"
    },
    "eventTime":"2016-02-24T18:10:00Z",
    "eventSource":"firehose.amazonaws.com",
    "eventName":"ListDeliveryStreams",
    "awsRegion":"us-east-1",
    "sourceIPAddress":"127.0.0.1",
    "userAgent":"aws-internal/3",
    "requestParameters":{
      "limit":10
    },
    "responseElements":null,
    "requestID":"d1bf7f86-db21-11e5-bb88-91ae9617edf5",
    "eventID":"67f63c74-4335-48c0-9004-4ba35ce00128",
    "eventType":"AwsApiCall",
    "recipientAccountId":"111122223333"
  },
  {
    "eventVersion":"1.02",
    "userIdentity":{
      "type":"IAMUser",
      "principalId":"AKIAIOSFODNN7EXAMPLE",
      "arn":"arn:aws:iam::111122223333:user/CloudTrail_Test_User",
      "accountId":"111122223333",
      "accessKeyId":"AKIAI44QH8DHBEXAMPLE",
      "userName":"CloudTrail_Test_User"
    },
    "eventTime":"2016-02-24T18:10:09Z",
    "eventSource":"firehose.amazonaws.com",
    "eventName":"UpdateDestination",
    "awsRegion":"us-east-1",
    "sourceIPAddress":"127.0.0.1",
    "userAgent":"aws-internal/3",
    "requestParameters":{
      "destinationId":"destinationId-000000000001",
      "deliveryStreamName":"TestRedshiftStream",
      "currentDeliveryStreamVersionId":"1",
      "redshiftDestinationUpdate":{

```

```

        "roleARN": "arn:aws:iam::111122223333:role/Firehose",
        "clusterJDBCURL": "jdbc:redshift://example.abc123.us-
west-2.redshift.amazonaws.com:5439/dev",
        "password": "",
        "username": "",
        "copyCommand": {
            "copyOptions": "copyOptions",
            "dataTableName": "dataTable"
        },
        "s3Update": {
            "bucketARN": "arn:aws:s3:::amzn-s3-demo-bucket-update",
            "roleARN": "arn:aws:iam::111122223333:role/Firehose",
            "compressionFormat": "GZIP",
            "bufferingHints": {
                "sizeInMBs": 3,
                "intervalInSeconds": 900
            },
            "encryptionConfiguration": {
                "kMSEncryptionConfig": {
                    "aWSKMSKeyARN": "arn:aws:kms:us-east-1:key"
                }
            },
            "prefix": "arn:aws:s3:::amzn-s3-demo-bucket"
        }
    },
    "responseElements": null,
    "requestID": "d549428d-db21-11e5-bb88-91ae9617edf5",
    "eventID": "1cb21e0b-416a-415d-bbf9-769b152a6585",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
},
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AKIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/CloudTrail_Test_User",
        "accountId": "111122223333",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "userName": "CloudTrail_Test_User"
    },
    "eventTime": "2016-02-24T18:10:12Z",
    "eventSource": "firehose.amazonaws.com",

```

```
    "eventName": "DeleteDeliveryStream",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "aws-internal/3",
    "requestParameters": {
      "deliveryStreamName": "TestRedshiftStream"
    },
    "responseElements": null,
    "requestID": "d85968c1-db21-11e5-bb88-91ae9617edf5",
    "eventID": "dd46bb98-b4e9-42ff-a6af-32d57e636ad1",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
  }
]
}
```

Firehose AWS SDKs的程式碼範例

下列程式碼範例示範如何使用 Firehose 搭配 AWS 軟體開發套件 (SDK)。

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中查看內容中的動作。

案例是向您展示如何呼叫服務中的多個函數或與其他 AWS 服務組合來完成特定任務的程式碼範例。

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 Firehose](#)。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

- [Firehose AWS SDKs的基本範例](#)
 - [Firehose 使用 AWS SDKs的動作](#)
 - [PutRecord 搭配 AWS SDK 或 CLI 使用](#)
 - [PutRecordBatch 搭配 AWS SDK 或 CLI 使用](#)
- [Firehose AWS SDKs的案例](#)
 - [使用 Amazon Data Firehose 處理個別和批次記錄](#)

Firehose AWS SDKs的基本範例

下列程式碼範例示範如何搭配 AWS SDK 使用 Amazon Data Firehose 的基本功能。

範例

- [Firehose 使用 AWS SDKs的動作](#)
 - [PutRecord 搭配 AWS SDK 或 CLI 使用](#)
 - [PutRecordBatch 搭配 AWS SDK 或 CLI 使用](#)

Firehose 使用 AWS SDKs的動作

下列程式碼範例示範如何使用 AWS SDKs 執行個別 Firehose 動作。每個範例均包含 GitHub 的連結，您可以在連結中找到設定和執行程式碼的相關說明。

這些摘錄會呼叫 Firehose API，是必須在內容中執行之大型程式的程式碼摘錄。您可以在 [Firehose AWS SDKs的案例](#) 中查看內容中的動作。

下列範例僅包含最常使用的動作。如需完整清單，請參閱 [《Amazon Data Firehose API 參考》](#)。

範例

- [PutRecord 搭配 AWS SDK 或 CLI 使用](#)
- [PutRecordBatch 搭配 AWS SDK 或 CLI 使用](#)

PutRecord 搭配 AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 PutRecord。

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：

- [將記錄放入 Firehose](#)

CLI

AWS CLI

將記錄寫入串流

下列 put-record 範例會將資料寫入串流。資料以 Base64 格式編碼。

```
aws firehose put-record \  
  --delivery-stream-name my-stream \  
  --record '{"Data": "SGVsbG8gd29ybGQ="}'
```

輸出：

```
{  
  "RecordId": "RjB5K/nnoGFHqwTsZ1Nd/  
TTqvjE8V5dsyXZTQn2JXrdpMT0wssyEb6nfC8fwf1whhwnItt4mvrn+gsqeK5jB7QjuLg283+Ps4Sz/  
j1Xujv31iDhnPdaLw4B0yM9Amv7PcCuB2079RuM0NhoakbyUymlwY8yt20G8X2420wu1j1Fafhci4erAt7QhDEvpw  
  "Encrypted": false  
}
```

如需詳細資訊，請參閱 [《Amazon Kinesis Data Firehose 開發人員指南》](#) 中的 [傳送資料到 Amazon Kinesis Data Firehose 交付串流](#)。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [PutRecord](#)。

Java

SDK for Java 2.x

Note

GitHub 上提供更多範例。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。

```
/**
 * Puts a record to the specified Amazon Kinesis Data Firehose delivery
 * stream.
 *
 * @param record The record to be put to the delivery stream. The record must
 * be a {@link Map} of String keys and Object values.
 * @param deliveryStreamName The name of the Amazon Kinesis Data Firehose
 * delivery stream to which the record should be put.
 * @throws IllegalArgumentException if the input record or delivery stream
 * name is null or empty.
 * @throws RuntimeException if there is an error putting the record to the
 * delivery stream.
 */
public static void putRecord(Map<String, Object> record, String
deliveryStreamName) {
    if (record == null || deliveryStreamName == null ||
deliveryStreamName.isEmpty()) {
        throw new IllegalArgumentException("Invalid input: record or delivery
stream name cannot be null/empty");
    }
    try {
        String jsonRecord = new ObjectMapper().writeValueAsString(record);
        Record firehoseRecord = Record.builder()

.data(SdkBytes.fromByteArray(jsonRecord.getBytes(StandardCharsets.UTF_8)))
        .build();

        PutRecordRequest putRecordRequest = PutRecordRequest.builder()
            .deliveryStreamName(deliveryStreamName)
            .record(firehoseRecord)
```

```
        .build());

        getFirehoseClient().putRecord(putRecordRequest);
        System.out.println("Record sent: " + jsonRecord);
    } catch (Exception e) {
        throw new RuntimeException("Failed to put record: " + e.getMessage(),
e);
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Java 2.x API 參考》中的 [PutRecord](#)。

Python

適用於 Python 的 SDK (Boto3)

Note

GitHub 上提供更多範例。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。

```
class FirehoseClient:
    """
    AWS Firehose client to send records and monitor metrics.

    Attributes:
        config (object): Configuration object with delivery stream name and
region.
        delivery_stream_name (str): Name of the Firehose delivery stream.
        region (str): AWS region for Firehose and CloudWatch clients.
        firehose (boto3.client): Boto3 Firehose client.
        cloudwatch (boto3.client): Boto3 CloudWatch client.
    """

    def __init__(self, config):
        """
        Initialize the FirehoseClient.

        Args:
```

```
        config (object): Configuration object with delivery stream name and
region.
    """
    self.config = config
    self.delivery_stream_name = config.delivery_stream_name
    self.region = config.region
    self.firehose = boto3.client("firehose", region_name=self.region)
    self.cloudwatch = boto3.client("cloudwatch", region_name=self.region)

    @backoff.on_exception(
        backoff.expo, Exception, max_tries=5, jitter=backoff.full_jitter
    )
    def put_record(self, record: dict):
        """
        Put individual records to Firehose with backoff and retry.

        Args:
            record (dict): The data record to be sent to Firehose.

        This method attempts to send an individual record to the Firehose
        delivery stream.
        It retries with exponential backoff in case of exceptions.
        """
        try:
            entry = self._create_record_entry(record)
            response = self.firehose.put_record(
                DeliveryStreamName=self.delivery_stream_name, Record=entry
            )
            self._log_response(response, entry)
        except Exception:
            logger.info(f"Fail record: {record}.")
            raise
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 [PutRecord](#)。

SAP ABAP

適用於 SAP ABAP 的開發套件

Note

GitHub 上提供更多範例。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。

```
TRY.  
  DATA(lo_record) = NEW /aws1/cl_frhrecord( iv_data = iv_data ).  
  
  DATA(lo_result) = lo_frh->putrecord(  
    iv_deliverystreamname = iv_deliv_stream_name  
    io_record              = lo_record ).  
  
  MESSAGE 'Record sent to Firehose delivery stream.' TYPE 'I'.  
CATCH /aws1/cx_frhresourcefoundex.  
  MESSAGE 'Delivery stream not found.' TYPE 'E'.  
CATCH /aws1/cx_frhinvalidargumentex.  
  MESSAGE 'Invalid argument provided.' TYPE 'E'.  
CATCH /aws1/cx_frhserviceunavailex.  
  MESSAGE 'Service temporarily unavailable.' TYPE 'E'.  
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 SAP ABAP 的 AWS SDK API 參考》中的 [PutRecord](#)。

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 Firehose](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

PutRecordBatch 搭配 AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 PutRecordBatch。

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：

- [將記錄放入 Firehose](#)

CLI

AWS CLI

將多筆記錄寫入串流

下列 `put-record-batch` 範例將三筆記錄寫入串流。資料以 Base64 格式編碼。

```
aws firehose put-record-batch \
  --delivery-stream-name my-stream \
  --records file://records.json
```

`myfile.json` 的內容：

```
[
  {"Data": "Rm1yc3QgdGhpbmc="},
  {"Data": "U2Vjb25kIHRoaW5n"},
  {"Data": "VGhpcmQgdGhpbmc="}
]
```

輸出：

```
{
  "FailedPutCount": 0,
  "Encrypted": false,
  "RequestResponses": [
    {
      "RecordId": "9D20J6t2EqCTZTXwGzeSv/EVHxRoRCw89xd+o3+sXg8DhY0aWKPSmZy/
CG1RVEys1u1xbeKh6VofEYKkoeiDrcjrxhQp9iF7sUW7pujiMEQ5LzlrzCkGosxQn
+3boDnURDEaD42V7Giixp0yLJkYZcae1i7HzlCEoy9LJhMr8EjDSi40m/9Vc2uhwwuAtGE0XKpxJ2WD7ZRwtAnY1k"
    },
    {
      "RecordId": "jFirejqxCLlK5xjH/UNmLMVcjkTEN76I7916X9PaZ
+PVa0SXdfU1WG0qEZhxq2js7xcZ552eoeDxsuTU1MSq9nZTbVfb6cQTIXnm/
GsuF37Uhg67GkmR5z9016XKJ+/
+pDl0Fv7Hh9a3oUS6wYm3DcNRLTHHAimANp1PhkQvWpvLRfzbuCukBphR2QVzhP90iHLbzGwy8/
DfH8sqWEUYASNJKS8GXP5s"
    },
    {
      "RecordId":
"oy0amQ40o5Y2YV4vxzufdcM00w6n3EP13tpPJGoYVNKH4APPVqNcbUgefo1stEFRg4hTLrf2k6eliHu/9+YJ5R3
DTBt3qBlmTj7Xq8SKVb01S7YvMTpWkMKA86f8JfmT8BMKomb4XZS/s0kQLe+qh0sYKXWl"
    }
  ]
}
```

```
]
}
```

如需詳細資訊，請參閱《Amazon Kinesis Data Firehose 開發人員指南》中的[傳送資料到 Amazon Kinesis Data Firehose 交付串流](#)。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [PutRecordBatch](#)。

Java

SDK for Java 2.x

Note

GitHub 上提供更多範例。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。

```
/**
 * Puts a batch of records to an Amazon Kinesis Data Firehose delivery
 * stream.
 *
 * @param records          a list of maps representing the records to be
 * sent
 * @param batchSize       the maximum number of records to include in each
 * batch
 * @param deliveryStreamName the name of the Kinesis Data Firehose delivery
 * stream
 * @throws IllegalArgumentException if the input parameters are invalid (null
 * or empty)
 * @throws RuntimeException       if there is an error putting the record
 * batch
 */
public static void putRecordBatch(List<Map<String, Object>> records, int
batchSize, String deliveryStreamName) {
    if (records == null || records.isEmpty() || deliveryStreamName == null ||
deliveryStreamName.isEmpty()) {
        throw new IllegalArgumentException("Invalid input: records or
delivery stream name cannot be null/empty");
    }
    ObjectMapper objectMapper = new ObjectMapper();
```

```
try {
    for (int i = 0; i < records.size(); i += batchSize) {
        List<Map<String, Object>> batch = records.subList(i, Math.min(i +
batchSize, records.size()));

        List<Record> batchRecords = batch.stream().map(record -> {
            try {
                String jsonRecord =
objectMapper.writeValueAsString(record);
                return Record.builder()

.data(SdkBytes.fromByteArray(jsonRecord.getBytes(StandardCharsets.UTF_8)))
                .build();
            } catch (Exception e) {
                throw new RuntimeException("Error creating Firehose
record", e);
            }
        }).collect(Collectors.toList());

        PutRecordBatchRequest request = PutRecordBatchRequest.builder()
            .deliveryStreamName(deliveryStreamName)
            .records(batchRecords)
            .build();

        PutRecordBatchResponse response =
getFirehoseClient().putRecordBatch(request);

        if (response.failedPutCount() > 0) {
            response.requestResponses().stream()
                .filter(r -> r.errorCode() != null)
                .forEach(r -> System.err.println("Failed record: " +
r.errorMessage()));
        }
        System.out.println("Batch sent with size: " +
batchRecords.size());
    }
} catch (Exception e) {
    throw new RuntimeException("Failed to put record batch: " +
e.getMessage(), e);
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Java 2.x API 參考》中的 [PutRecordBatch](#)。

Python

適用於 Python 的 SDK (Boto3)

Note

GitHub 上提供更多範例。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。

```
class FirehoseClient:
    """
    AWS Firehose client to send records and monitor metrics.

    Attributes:
        config (object): Configuration object with delivery stream name and
        region.
        delivery_stream_name (str): Name of the Firehose delivery stream.
        region (str): AWS region for Firehose and CloudWatch clients.
        firehose (boto3.client): Boto3 Firehose client.
        cloudwatch (boto3.client): Boto3 CloudWatch client.
    """

    def __init__(self, config):
        """
        Initialize the FirehoseClient.

        Args:
            config (object): Configuration object with delivery stream name and
            region.
        """
        self.config = config
        self.delivery_stream_name = config.delivery_stream_name
        self.region = config.region
        self.firehose = boto3.client("firehose", region_name=self.region)
        self.cloudwatch = boto3.client("cloudwatch", region_name=self.region)

    @backoff.on_exception(
        backoff.expo, Exception, max_tries=5, jitter=backoff.full_jitter
    )
    def put_record_batch(self, data: list, batch_size: int = 500):
```

```
"""
Put records in batches to Firehose with backoff and retry.

Args:
    data (list): List of data records to be sent to Firehose.
    batch_size (int): Number of records to send in each batch. Default is
500.

This method attempts to send records in batches to the Firehose delivery
stream.
It retries with exponential backoff in case of exceptions.
"""
for i in range(0, len(data), batch_size):
    batch = data[i : i + batch_size]
    record_dicts = [{"Data": json.dumps(record)} for record in batch]
    try:
        response = self.firehose.put_record_batch(
            DeliveryStreamName=self.delivery_stream_name,
Records=record_dicts
        )
        self._log_batch_response(response, len(batch))
    except Exception as e:
        logger.info(f"Failed to send batch of {len(batch)} records.
Error: {e}")
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 [PutRecordBatch](#)。

Rust

適用於 Rust 的 SDK

Note

GitHub 上提供更多範例。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。

```
async fn put_record_batch(
```

```

    client: &Client,
    stream: &str,
    data: Vec<Record>,
) -> Result<PutRecordBatchOutput, SdkError<PutRecordBatchError>> {
    client
        .put_record_batch()
        .delivery_stream_name(stream)
        .set_records(Some(data))
        .send()
        .await
}

```

- 如需 API 詳細資訊，請參閱《AWS SDK for Rust API 參考》中的 [PutRecordBatch](#)。

SAP ABAP

適用於 SAP ABAP 的開發套件

Note

GitHub 上提供更多範例。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。

```

TRY.
  DATA(lo_result) = lo_frh->putrecordbatch(
    iv_deliverystreamname = iv_deliv_stream_name
    it_records             = it_records ).

  DATA(lv_failed_count) = lo_result->get_failedputcount( ).

  IF lv_failed_count > 0.
    MESSAGE |{ lv_failed_count } records failed to send.| TYPE 'I'.
  ELSE.
    MESSAGE 'All records sent successfully to Firehose delivery stream.'
    TYPE 'I'.
  ENDIF.
CATCH /aws1/cx_frhresourcenotfoundex.
  MESSAGE 'Delivery stream not found.' TYPE 'E'.
CATCH /aws1/cx_frhinvalidargumentex.
  MESSAGE 'Invalid argument provided.' TYPE 'E'.

```

```
CATCH /aws1/cx_frhserviceunavailex.  
    MESSAGE 'Service temporarily unavailable.' TYPE 'E'.  
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [PutRecordBatch](#)。

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 Firehose](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

Firehose AWS SDKs 的案例

下列程式碼範例示範如何在 Firehose AWS SDKs 中實作常見案例。這些案例示範如何呼叫 Firehose 中的多個函數，或與其他 AWS 服務結合，藉以完成特定任務。每個案例均包含完整原始碼的連結，您可在連結中找到如何設定和執程式碼的相關指示。

案例的目標是獲得中等水平的經驗，協助您了解內容中的服務動作。

範例

- [使用 Amazon Data Firehose 處理個別和批次記錄](#)

使用 Amazon Data Firehose 處理個別和批次記錄

下列程式碼範例示範如何使用 Firehose 處理個別和批次記錄。

Java

適用於 Java 2.x 的 SDK

Note

GitHub 上提供更多範例。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。

此範例會將個別記錄和批次記錄放入 Firehose。

```
/**
```

```
* Amazon Firehose Scenario example using Java V2 SDK.
*
* Demonstrates individual and batch record processing,
* and monitoring Firehose delivery stream metrics.
*/
public class FirehoseScenario {

    private static FirehoseClient firehoseClient;
    private static CloudWatchClient cloudWatchClient;

    public static void main(String[] args) {
        final String usage = ""
            Usage:
            <deliveryStreamName>
            Where:
            deliveryStreamName - The Firehose delivery stream name.
            """;

        if (args.length != 1) {
            System.out.println(usage);
            return;
        }

        String deliveryStreamName = args[0];

        try {
            // Read and parse sample data.
            String jsonContent = readJsonFile("sample_records.json");
            ObjectMapper objectMapper = new ObjectMapper();
            List<Map<String, Object>> sampleData =
objectMapper.readValue(jsonContent, new TypeReference<>() {});

            // Process individual records.
            System.out.println("Processing individual records...");
            sampleData.subList(0, 100).forEach(record -> {
                try {
                    putRecord(record, deliveryStreamName);
                } catch (Exception e) {
                    System.err.println("Error processing record: " +
e.getMessage());
                }
            });

            // Monitor metrics.
        }
    }
}
```

```
        monitorMetrics(deliveryStreamName);

        // Process batch records.
        System.out.println("Processing batch records...");
        putRecordBatch(sampleData.subList(100, sampleData.size()), 500,
deliveryStreamName);
        monitorMetrics(deliveryStreamName);

    } catch (Exception e) {
        System.err.println("Scenario failed: " + e.getMessage());
    } finally {
        closeClients();
    }
}

private static FirehoseClient getFirehoseClient() {
    if (firehoseClient == null) {
        firehoseClient = FirehoseClient.builder()
            .region(Region.US_EAST_1)
            .build();
    }
    return firehoseClient;
}

private static CloudWatchClient getCloudWatchClient() {
    if (cloudWatchClient == null) {
        cloudWatchClient = CloudWatchClient.builder()
            .region(Region.US_EAST_1)
            .build();
    }
    return cloudWatchClient;
}

/**
 * Puts a record to the specified Amazon Kinesis Data Firehose delivery
stream.
 *
 * @param record The record to be put to the delivery stream. The record must
be a {@link Map} of String keys and Object values.
 * @param deliveryStreamName The name of the Amazon Kinesis Data Firehose
delivery stream to which the record should be put.
 * @throws IllegalArgumentException if the input record or delivery stream
name is null or empty.

```

```
    * @throws RuntimeException if there is an error putting the record to the
    delivery stream.
    */
    public static void putRecord(Map<String, Object> record, String
deliveryStreamName) {
        if (record == null || deliveryStreamName == null ||
deliveryStreamName.isEmpty()) {
            throw new IllegalArgumentException("Invalid input: record or delivery
stream name cannot be null/empty");
        }
        try {
            String jsonRecord = new ObjectMapper().writeValueAsString(record);
            Record firehoseRecord = Record.builder()

.data(SdkBytes.fromByteArray(jsonRecord.getBytes(StandardCharsets.UTF_8)))
            .build();

            PutRecordRequest putRecordRequest = PutRecordRequest.builder()
                .deliveryStreamName(deliveryStreamName)
                .record(firehoseRecord)
                .build();

            getFirehoseClient().putRecord(putRecordRequest);
            System.out.println("Record sent: " + jsonRecord);
        } catch (Exception e) {
            throw new RuntimeException("Failed to put record: " + e.getMessage(),
e);
        }
    }

    /**
     * Puts a batch of records to an Amazon Kinesis Data Firehose delivery
stream.
     *
     * @param records          a list of maps representing the records to be
sent
     * @param batchSize       the maximum number of records to include in each
batch
     * @param deliveryStreamName the name of the Kinesis Data Firehose delivery
stream
     * @throws IllegalArgumentException if the input parameters are invalid (null
or empty)
    */
}
```

```
    * @throws RuntimeException          if there is an error putting the record
batch
    */
    public static void putRecordBatch(List<Map<String, Object>> records, int
batchSize, String deliveryStreamName) {
        if (records == null || records.isEmpty() || deliveryStreamName == null ||
deliveryStreamName.isEmpty()) {
            throw new IllegalArgumentException("Invalid input: records or
delivery stream name cannot be null/empty");
        }
        ObjectMapper objectMapper = new ObjectMapper();

        try {
            for (int i = 0; i < records.size(); i += batchSize) {
                List<Map<String, Object>> batch = records.subList(i, Math.min(i +
batchSize, records.size()));

                List<Record> batchRecords = batch.stream().map(record -> {
                    try {
                        String jsonRecord =
objectMapper.writeValueAsString(record);
                        return Record.builder()

.data(SdkBytes.fromByteArray(jsonRecord.getBytes(StandardCharsets.UTF_8)))
                        .build();
                    } catch (Exception e) {
                        throw new RuntimeException("Error creating Firehose
record", e);
                    }
                }).collect(Collectors.toList());

                PutRecordBatchRequest request = PutRecordBatchRequest.builder()
                    .deliveryStreamName(deliveryStreamName)
                    .records(batchRecords)
                    .build();

                PutRecordBatchResponse response =
getFirehoseClient().putRecordBatch(request);

                if (response.failedPutCount() > 0) {
                    response.requestResponses().stream()
                        .filter(r -> r.errorCode() != null)
                        .forEach(r -> System.err.println("Failed record: " +
r.errorMessage()));
                }
            }
        }
    }
}
```

```
        }
        System.out.println("Batch sent with size: " +
batchRecords.size());
    }
    } catch (Exception e) {
        throw new RuntimeException("Failed to put record batch: " +
e.getMessage(), e);
    }
}

public static void monitorMetrics(String deliveryStreamName) {
    Instant endTime = Instant.now();
    Instant startTime = endTime.minusSeconds(600);

    List<String> metrics = List.of("IncomingBytes", "IncomingRecords",
"FailedPutCount");
    metrics.forEach(metric -> monitorMetric(metric, startTime, endTime,
deliveryStreamName));
}

private static void monitorMetric(String metricName, Instant startTime,
Instant endTime, String deliveryStreamName) {
    try {
        GetMetricStatisticsRequest request =
GetMetricStatisticsRequest.builder()
            .namespace("AWS/Firehose")
            .metricName(metricName)

.dimensions(Dimension.builder().name("DeliveryStreamName").value(deliveryStreamName).bu

            .startTime(startTime)
            .endTime(endTime)
            .period(60)
            .statistics(Statistic.SUM)
            .build();

        GetMetricStatisticsResponse response =
getCloudWatchClient().getMetricStatistics(request);
        double totalSum =
response.datapoints().stream().mapToDouble(Datapoint::sum).sum();
        System.out.println(metricName + ": " + totalSum);
    } catch (Exception e) {
        System.err.println("Failed to monitor metric " + metricName + ": " +
e.getMessage());
    }
}
```

```
    }

    public static String readJsonFile(String fileName) throws IOException {
        try (InputStream inputStream =
            FirehoseScenario.class.getResourceAsStream("/" + fileName);
            Scanner scanner = new Scanner(inputStream, StandardCharsets.UTF_8))
        {
            return scanner.useDelimiter("\\\\A").next();
        } catch (Exception e) {
            throw new RuntimeException("Error reading file: " + fileName, e);
        }
    }

    private static void closeClients() {
        try {
            if (firehoseClient != null) firehoseClient.close();
            if (cloudWatchClient != null) cloudWatchClient.close();
        } catch (Exception e) {
            System.err.println("Error closing clients: " + e.getMessage());
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Java 2.x API 參考》中的下列主題。
 - [PutRecord](#)
 - [PutRecordBatch](#)

Python

適用於 Python 的 SDK (Boto3)

Note

GitHub 上提供更多範例。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。

此指令碼會將個別和批次記錄放入 Firehose。

```
import json
```

```
import logging
import random
from datetime import datetime, timedelta

import backoff
import boto3

from config import get_config

def load_sample_data(path: str) -> dict:
    """
    Load sample data from a JSON file.

    Args:
        path (str): The file path to the JSON file containing sample data.

    Returns:
        dict: The loaded sample data as a dictionary.
    """
    with open(path, "r") as f:
        return json.load(f)

# Configure logging
logging.basicConfig(level=logging.INFO)
logger = logging.getLogger(__name__)

class FirehoseClient:
    """
    AWS Firehose client to send records and monitor metrics.

    Attributes:
        config (object): Configuration object with delivery stream name and
        region.
        delivery_stream_name (str): Name of the Firehose delivery stream.
        region (str): AWS region for Firehose and CloudWatch clients.
        firehose (boto3.client): Boto3 Firehose client.
        cloudwatch (boto3.client): Boto3 CloudWatch client.
    """

    def __init__(self, config):
        """
```

```
Initialize the FirehoseClient.

Args:
    config (object): Configuration object with delivery stream name and
region.
"""
self.config = config
self.delivery_stream_name = config.delivery_stream_name
self.region = config.region
self.firehose = boto3.client("firehose", region_name=self.region)
self.cloudwatch = boto3.client("cloudwatch", region_name=self.region)

@backoff.on_exception(
    backoff.expo, Exception, max_tries=5, jitter=backoff.full_jitter
)
def put_record(self, record: dict):
    """
    Put individual records to Firehose with backoff and retry.

    Args:
        record (dict): The data record to be sent to Firehose.

    This method attempts to send an individual record to the Firehose
    delivery stream.
    It retries with exponential backoff in case of exceptions.
    """
    try:
        entry = self._create_record_entry(record)
        response = self.firehose.put_record(
            DeliveryStreamName=self.delivery_stream_name, Record=entry
        )
        self._log_response(response, entry)
    except Exception:
        logger.info(f"Fail record: {record}.")
        raise

@backoff.on_exception(
    backoff.expo, Exception, max_tries=5, jitter=backoff.full_jitter
)
def put_record_batch(self, data: list, batch_size: int = 500):
    """
    Put records in batches to Firehose with backoff and retry.
```

Args:

- `data (list)`: List of data records to be sent to Firehose.
- `batch_size (int)`: Number of records to send in each batch. Default is 500.

This method attempts to send records in batches to the Firehose delivery stream.

It retries with exponential backoff in case of exceptions.

```

"""
for i in range(0, len(data), batch_size):
    batch = data[i : i + batch_size]
    record_dicts = [{"Data": json.dumps(record)} for record in batch]
    try:
        response = self.firehose.put_record_batch(
            DeliveryStreamName=self.delivery_stream_name,
            Records=record_dicts
        )
        self._log_batch_response(response, len(batch))
    except Exception as e:
        logger.info(f"Failed to send batch of {len(batch)} records.
Error: {e}")

```

```

def get_metric_statistics(
    self,
    metric_name: str,
    start_time: datetime,
    end_time: datetime,
    period: int,
    statistics: list = ["Sum"],
) -> list:
    """

```

Retrieve metric statistics from CloudWatch.

Args:

- `metric_name (str)`: The name of the metric.
- `start_time (datetime)`: The start time for the metric statistics.
- `end_time (datetime)`: The end time for the metric statistics.
- `period (int)`: The granularity, in seconds, of the returned data points.
- `statistics (list)`: A list of statistics to retrieve. Default is ['Sum'].

```
Returns:
    list: List of datapoints containing the metric statistics.
"""
response = self.cloudwatch.get_metric_statistics(
    Namespace="AWS/Firehose",
    MetricName=metric_name,
    Dimensions=[
        {"Name": "DeliveryStreamName", "Value":
self.delivery_stream_name},
    ],
    StartTime=start_time,
    EndTime=end_time,
    Period=period,
    Statistics=statistics,
)
return response["Datapoints"]

def monitor_metrics(self):
    """
    Monitor Firehose metrics for the last 5 minutes.

    This method retrieves and logs the 'IncomingBytes', 'IncomingRecords',
    and 'FailedPutCount' metrics
    from CloudWatch for the last 5 minutes.
    """
    end_time = datetime.utcnow()
    start_time = end_time - timedelta(minutes=10)
    period = int((end_time - start_time).total_seconds())

    metrics = {
        "IncomingBytes": self.get_metric_statistics(
            "IncomingBytes", start_time, end_time, period
        ),
        "IncomingRecords": self.get_metric_statistics(
            "IncomingRecords", start_time, end_time, period
        ),
        "FailedPutCount": self.get_metric_statistics(
            "FailedPutCount", start_time, end_time, period
        ),
    }

    for metric, datapoints in metrics.items():
        if datapoints:
            total_sum = sum(datapoint["Sum"] for datapoint in datapoints)
```

```
        if metric == "IncomingBytes":
            logger.info(
                f"{metric}: {round(total_sum)} ({total_sum / (1024 *
1024):.2f} MB)"
            )
        else:
            logger.info(f"{metric}: {round(total_sum)}")
    else:
        logger.info(f"No data found for {metric} over the last 5
minutes")

def _create_record_entry(self, record: dict) -> dict:
    """
    Create a record entry for Firehose.

    Args:
        record (dict): The data record to be sent.

    Returns:
        dict: The record entry formatted for Firehose.

    Raises:
        Exception: If a simulated network error occurs.
    """
    if random.random() < 0.2:
        raise Exception("Simulated network error")
    elif random.random() < 0.1:
        return {"Data": '{"malformed": "data"}'}
    else:
        return {"Data": json.dumps(record)}

def _log_response(self, response: dict, entry: dict):
    """
    Log the response from Firehose.

    Args:
        response (dict): The response from the Firehose put_record API call.
        entry (dict): The record entry that was sent.
    """
    if response["ResponseMetadata"]["HTTPStatusCode"] == 200:
        logger.info(f"Sent record: {entry}")
    else:
        logger.info(f"Fail record: {entry}")
```

```
def _log_batch_response(self, response: dict, batch_size: int):
    """
    Log the batch response from Firehose.

    Args:
        response (dict): The response from the Firehose put_record_batch API
call.
        batch_size (int): The number of records in the batch.
    """
    if response.get("FailedPutCount", 0) > 0:
        logger.info(
            f'Failed to send {response["FailedPutCount"]} records in batch of
{batch_size}'
        )
    else:
        logger.info(f"Successfully sent batch of {batch_size} records")

if __name__ == "__main__":
    config = get_config()
    data = load_sample_data(config.sample_data_file)
    client = FirehoseClient(config)

    # Process the first 100 sample network records
    for record in data[:100]:
        try:
            client.put_record(record)
        except Exception as e:
            logger.info(f"Put record failed after retries and backoff: {e}")
    client.monitor_metrics()

    # Process remaining records using the batch method
    try:
        client.put_record_batch(data[100:])
    except Exception as e:
        logger.info(f"Put record batch failed after retries and backoff: {e}")
    client.monitor_metrics()
```

此檔案包含上述指令碼的組態。

```
class Config:
```

```
def __init__(self):
    self.delivery_stream_name = "ENTER YOUR DELIVERY STREAM NAME HERE"
    self.region = "us-east-1"
    self.sample_data_file = (
        "../../../../../scenarios/features/firehose/resources/
sample_records.json"
    )

def get_config():
    return Config()
```

- 如需 API 詳細資訊，請參閱《適用於 Python (Boto3) 的 AWS SDK API 參考》中的下列主題。
 - [PutRecord](#)
 - [PutRecordBatch](#)

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 Firehose](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

對 Amazon Data Firehose 中的錯誤進行故障診斷

如果 Firehose 在交付或處理資料時發生錯誤，它會重試，直到設定的重試持續時間過期為止。如果重試持續時間在資料成功交付之前結束，Firehose 會將資料備份到設定的 S3 備份儲存貯體。如果目的地是 Amazon S3 且交付失敗，或交付至備份 S3 儲存貯體失敗，Firehose 會一直重試，直到保留期間結束為止。

如需使用 CloudWatch 追蹤交付錯誤的相關資訊，請參閱 [the section called “使用 CloudWatch Logs 監控”](#)。

Direct PUT

對於 DirectPut Firehose 串流，Firehose 會保留記錄 24 小時。對於資料來源為 Kinesis 資料串流的 Firehose 串流，您可以變更保留期，如[變更資料保留期](#)所述。在此情況下，Firehose 會無限期重試下列操作：GetRecords、DescribeStream 和 GetShardIterator。

如果 Firehose 串流使用 DirectPut，請檢查 IncomingBytes 和 IncomingRecords 指標，以查看是否有傳入流量。如果您使用的是 PutRecord 或 PutRecordBatch，請確定您擷取例外狀況並重試。我們建議搭配指數退避的重試原則，以進行抖動和多次重試。此外，如果您使用 PutRecordBatch API，請確定您的程式碼會在回應中檢查 [FailedPutCount](#) 的值，即使 API 呼叫成功也一樣。

Kinesis Data Stream

如果 Firehose 串流使用 Kinesis 資料串流做為來源，請檢查來源資料串流的 IncomingBytes 和 IncomingRecords 指標。此外，請確保針對 Firehose 串流發出 DataReadFromKinesisStream.Bytes 和 DataReadFromKinesisStream.Records 指標。

常見問題

以下是疑難排解秘訣，可協助您在使用 Firehose 串流時解決常見問題。

Firehose 串流無法使用

Firehose 串流無法做為 CloudWatch Logs、CloudWatch Events 或 AWS IoT 動作的目標，因為某些 AWS 服務只能將訊息和事件傳送到位於相同中的 Firehose 串流 AWS 區域。確認您的 Firehose 串流與其他服務位於相同的區域。

目的地沒有資料

如果沒有資料擷取問題，而且針對 Firehose 串流發出的指標看起來不錯，但您沒有在目的地看到資料，請檢查讀取器邏輯。確認您的讀者正確解析出所有資料。

資料新鮮度指標增加或未發出

資料新鮮度是衡量資料在 Firehose 串流內目前狀態的指標。這是 Firehose 串流中最舊資料記錄的存留期，從 Firehose 擷取資料到目前為止測量。Firehose 提供可用來監控資料新鮮度的指標。若要識別指定目的地的資料新鮮度指標，請參閱 [the section called “使用 CloudWatch 指標監控使用量”](#)。

如果您啟用所有事件或所有文件的備份，請監控兩個不同的資料新鮮度指標：一個針對主要目的地，另一個則針對備份。

如果未發出資料重新整理指標，這表示 Firehose 串流沒有作用中的交付。資料交付完全封鎖時或沒有傳入資料時，就會發生這種情況。

如果資料新鮮度指標不斷增加，這表示資料交付落後。這種情況可能是由於下列其中一個原因而發生的。

- 目的地無法處理交付率。如果 Firehose 因為高流量而遇到暫時性錯誤，則交付可能會落後。除了 Amazon S3 以外的目的地都可能發生這種情況 (OpenSearch Service、Amazon Redshift 或 Splunk 就可能發生)。確定您的目的地有足夠的容量來處理傳入流量。
- 目的地很慢。如果 Firehose 遇到高延遲，資料交付可能會落後。監控目的地的延遲指標。
- Lambda 函數緩慢。這可能會導致資料交付率低於 Firehose 串流的資料擷取率。如果可能的話，請提高 Lambda 函數的效率。例如，如果此函數執行網路 IO，請使用多個執行緒或非同步 IO 來增加並行性。此外，請考慮增加 Lambda 函數的記憶體大小，以便 CPU 配置可以相應地增加。這可能會導致更快的 Lambda 調用。如需設定 Lambda 函數的資訊，請參閱 [設定 AWS Lambda 函數](#)。
- 資料交付期間發生失敗。如需如何使用 Amazon CloudWatch Logs 監控錯誤的相關資訊，請參閱 [the section called “使用 CloudWatch Logs 監控”](#)。
- 如果 Firehose 串流的資料來源是 Kinesis 資料串流，則可能會發生限流。檢查 `ThrottledGetRecords`、`ThrottledGetShardIterator` 和 `ThrottledDescribeStream` 指標。如果有多個取用者附加到 Kinesis 資料串流，請考慮下列情況：
 - 如果 `ThrottledGetRecords` 和 `ThrottledGetShardIterator` 指標很高，建議您增加針對資料串流佈建的碎片數目。
 - 如果 `ThrottledDescribeStream` 很高，我們建議您將 `kinesis:listshards` 許可新增至 [KinesisStreamSourceConfiguration](#) 中設定的角色。

- 目的地的緩衝提示很低。這可能會增加 Firehose 需要對目的地進行的往返次數，這可能會導致交付落後。請考慮增加緩衝提示的值。如需詳細資訊，請參閱[緩衝提示](#)。
- 當錯誤頻繁發生時，高重試持續時間可能會導致交付落後。請考慮減少重試持續時間。此外，監控錯誤並嘗試減少錯誤。如需如何使用 Amazon CloudWatch Logs 監控錯誤的相關資訊，請參閱 [the section called “使用 CloudWatch Logs 監控”](#)。
- 如果目的地是 Splunk，而且 `DeliveryToSplunk.DataFreshness` 很高，但 `DeliveryToSplunk.Success` 看起來不錯，Splunk 叢集可能忙碌中。如果可能的話，釋出 Splunk 叢集。或者，請聯絡 AWS Support 並請求增加 Firehose 用來與 Splunk 叢集通訊的頻道數量。

記錄格式轉換為 Apache Parquet 失敗

如果您取得包含 Set 類型的 DynamoDB 資料、透過 Lambda 將其串流至 Firehose 串流，並使用 AWS Glue Data Catalog 將記錄格式轉換為 Apache Parquet，就會發生這種情況。

當 AWS Glue 爬蟲程式為 DynamoDB 集合資料類型 (StringSet、NumberSet 和 BinarySet) 編製索引時 SET<BINARY>，它會分別將它們存放在資料目錄中，分別為 SET<STRING>、SET<BIGINT> 和 。不過，若要讓 Firehose 將資料記錄轉換為 Apache Parquet 格式，則需要 Apache Hive 資料類型。因為集合類型不是有效的 Apache Hive 資料類型，轉換會失敗。若要讓轉換正常運作，請以 Apache Hive 資料類型更新資料目錄。您可以在資料目錄中將 set 變更為 array 來這麼做。

將資料 AWS Glue 目錄中 **array** 的一或多個資料類型從 **set** 變更為

1. 登入 AWS 管理主控台，並在 <https://console.aws.amazon.com/glue/> 開啟 AWS Glue 主控台。
2. 在左側窗格的 Data catalog (資料目錄) 標題下，選擇 Tables (資料表)。
3. 在資料表清單中，選擇您需要修改一或多個資料類型的資料表名稱。您將會移至該資料表的詳細資訊頁面。
4. 選擇詳細資訊頁面右上角的編輯結構描述按鈕。
5. 在 Data type (資料類型) 欄中，選擇第一個 set 資料類型。
6. 在 Column type (欄類型) 下拉式清單中，將類型從 set 變更為 array。
7. 在 ArraySchema 欄位中，根據您案例的適當資料類型，輸入 `array<string>`、`array<int>` 或 `array<binary>`。
8. 選擇更新。

9. 重複上述步驟，將其他 set 類型轉換為 array 類型。
10. 選擇儲存。

Lambda 轉換物件缺少欄位

當您使用 Lambda 資料轉換將 JSON 資料變更為 Parquet 物件時，某些欄位可能會在轉換後遺失。如果您的 JSON 物件具有大寫字母，且區分大小寫設定為 `false`，這可能會導致資料轉換後 JSON 金鑰不相符，導致 S3 儲存貯體中產生的 Parquet 物件遺失資料。

若要修正此問題，請確定管線組態已將 `deserializationOption: case.insensitive` 設定為 `true` 以便 JSON 金鑰在轉換後相符。

故障診斷 Amazon S3

若資料未交付至您的 Amazon Simple Storage Service (Amazon S3) 儲存貯體，請確認下列事項。

- 檢查 Firehose `IncomingBytes` 和 `IncomingRecords` 指標，以確保資料成功傳送到您的 Firehose 串流。如需詳細資訊，請參閱 [使用 CloudWatch 指標監控 Amazon Data Firehose](#)。
- 如果已啟用 Lambda 的資料轉換，請檢查 Firehose `ExecuteProcessingSuccess` 指標，確認 Firehose 已嘗試調用 Lambda 函數。如需詳細資訊，請參閱 [使用 CloudWatch 指標監控 Amazon Data Firehose](#)。
- 檢查 Firehose `DeliveryToS3.Success` 指標，確認 Firehose 已嘗試將資料放入 Amazon S3 儲存貯體。如需詳細資訊，請參閱 [使用 CloudWatch 指標監控 Amazon Data Firehose](#)。
- 若尚未啟用錯誤記錄，請啟用這項功能並檢查傳送失敗的錯誤記錄。如需詳細資訊，請參閱 [使用 CloudWatch Logs 監控 Amazon Data Firehose](#)。
- 如果您在日誌中看到錯誤訊息，指出「Firehose 在呼叫 Amazon S3 服務時遇到 `InternalServerError`。此操作將重試；如果錯誤仍然存在，請聯絡 S3 進行解決。」，這可能是由於 S3 中單一分割區的請求率大幅提高所致。您可以最佳化 S3 字首設計模式以減輕問題。如需詳細資訊，請參閱 [最佳實務設計模式：最佳化 Amazon S3 效能](#)。如果這無法解決問題，請聯絡 AWS Support 以取得進一步協助。
- 請確定 Firehose 串流中指定的 Amazon S3 儲存貯體仍然存在。
- 如果已啟用 Lambda 的資料轉換，請確定 Firehose 串流中指定的 Lambda 函數仍然存在。
- 請確定 Firehose 串流中指定的 IAM 角色可存取您的 S3 儲存貯體和 Lambda 函數（如果已啟用資料轉換）。此外，請確定 IAM 角色可存取 CloudWatch 日誌群組和日誌串流，以檢查錯誤日誌。如需詳細資訊，請參閱 [授予 Firehose 對 Amazon S3 目的地的存取權](#)。

- 若您正使用資料轉換功能，請確認 Lambda 函數的回應從未回傳大小超過 6 MB 的乘載。如需詳細資訊，請參閱 [Amazon Data FirehoseData 轉換](#)。

疑難排解 Amazon Redshift

如果資料未交付到您的 Amazon Redshift 佈建叢集或 Amazon Redshift Serverless 工作群組，請檢查下列各項。

資料會先交付至您的 S3 儲存貯體，再載入 Amazon Redshift。如果資料未傳送至您的 S3 儲存貯體，請參閱[故障診斷 Amazon S3](#)。

- 檢查 Firehose DeliveryToRedshift.Success 指標，確認 Firehose 已嘗試將資料從 S3 儲存貯體複製到 Amazon Redshift 佈建叢集或 Amazon Redshift Serverless 工作群組。如需詳細資訊，請參閱[使用 CloudWatch 指標監控 Amazon Data Firehose](#)。
- 若尚未啟用錯誤記錄，請啟用這項功能並檢查傳送失敗的錯誤記錄。如需詳細資訊，請參閱[使用 CloudWatch Logs 監控 Amazon Data Firehose](#)。
- 檢查 Amazon Redshift STL_CONNECTION_LOG 資料表，查看 Firehose 是否可以成功進行連線。本表格可供您依使用者名稱檢視連線和狀態。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 [STL_CONNECTION_LOG](#)。
- 如果上述檢查顯示連線已建立，則請查看 Amazon Redshift STL_LOAD_ERRORS 資料表以驗證 COPY 失敗原因。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 [STL_LOAD_ERRORS](#)。
- 請確定 Firehose 串流中的 Amazon Redshift 組態準確且有效。
- 請確定 Firehose 串流中指定的 IAM 角色可以存取 Amazon Redshift 複製資料的 S3 儲存貯體，以及用於資料轉換的 Lambda 函數（如果已啟用資料轉換）。此外，請確定 IAM 角色可存取 CloudWatch 日誌群組和日誌串流，以檢查錯誤日誌。如需詳細資訊，請參閱[授予 Firehose 對 Amazon Redshift 目的地的存取權](#)。
- 如果您的 Amazon Redshift 佈建叢集或 Amazon Redshift Serverless 工作群組位於虛擬私有雲端 (VPC) 中，請確定叢集允許從 Firehose IP 地址存取。如需詳細資訊，請參閱[授予 Firehose 對 Amazon Redshift 目的地的存取權](#)。
- 請確認 Amazon Redshift 佈建的叢集或 Amazon Redshift Serverless 工作群組可公開使用。
- 若您正使用資料轉換功能，請確認 Lambda 函數的回應從未回傳大小超過 6 MB 的乘載。如需詳細資訊，請參閱 [Amazon Data FirehoseData 轉換](#)。

Amazon OpenSearch Service 疑難排解

如果資料未交付至您的 OpenSearch Service 域，請確認以下步驟。

您可以將資料同時備份到 Amazon S3 儲存貯體。如果資料未傳送至您的 S3 儲存貯體，請參閱 [故障診斷 Amazon S3](#)。

- 檢查 Firehose IncomingBytes 和 IncomingRecords 指標，以確保資料成功傳送到您的 Firehose 串流。如需詳細資訊，請參閱 [使用 CloudWatch 指標監控 Amazon Data Firehose](#)。
- 如果已啟用 Lambda 的資料轉換，請檢查 Firehose ExecuteProcessingSuccess 指標，確認 Firehose 已嘗試叫用 Lambda 函數。如需詳細資訊，請參閱 [使用 CloudWatch 指標監控 Amazon Data Firehose](#)。
- 檢查 Firehose DeliveryToAmazonOpenSearchService.Success 指標，確認 Firehose 已嘗試將資料索引至 OpenSearch Service 叢集。如需詳細資訊，請參閱 [使用 CloudWatch 指標監控 Amazon Data Firehose](#)。
- 若尚未啟用錯誤記錄，請啟用這項功能並檢查傳送失敗的錯誤記錄。如需詳細資訊，請參閱 [使用 CloudWatch Logs 監控 Amazon Data Firehose](#)。
- 請確定 Firehose 串流中的 OpenSearch Service 組態準確且有效。
- 如果已啟用 Lambda 的資料轉換，請確定 Firehose 串流中指定的 Lambda 函數仍然存在。此外，請確定 IAM 角色可存取 CloudWatch 日誌群組和日誌串流，以檢查錯誤日誌。如需詳細資訊，請參閱 [將 FirehoseAccess 授予公有 OpenSearch Service 目的地](#)。
- 請確定 Firehose 串流中指定的 IAM 角色可以存取您的 OpenSearch Service 叢集、S3 備份儲存貯體和 Lambda 函數（如果啟用資料轉換）。此外，請確定 IAM 角色可存取 CloudWatch 日誌群組和日誌串流，以檢查錯誤日誌。如需詳細資訊，請參閱 [授予 Firehose 對公有 OpenSearch Service 目的地的存取權](#)。
- 若您正使用資料轉換功能，請確認 Lambda 函數的回應從未回傳大小超過 6 MB 的乘載。如需詳細資訊，請參閱 [Amazon Data FirehoseData 轉換](#)。
- Amazon Data Firehose 目前不支援將 CloudWatch Logs 交付至 Amazon OpenSearch Service 目的地，因為 Amazon CloudWatch 會將多個日誌事件合併為一個 Firehose 記錄，且 Amazon OpenSearch Service 無法接受單一記錄中的多個日誌事件。作為替代方案，您可以考慮在 [CloudWatch Logs 中使用 Amazon OpenSearch Service 的訂閱篩選條件](#)。

疑難排解 Splunk

如果資料未傳送至您的 Splunk 終端節點，請確認以下步驟。

- 如果您的 Splunk 平台位於 VPC 中，請確定 Firehose 可以存取它。如需詳細資訊，請參閱[在 VPC 中存取 Splunk](#)。
- 如果您使用 AWS 負載平衡器，請確定它是 Classic Load Balancer 或 Application Load Balancer。此外，針對 Classic Load Balancer 啟用停用 Cookie 過期的持續時間型黏性工作階段，並將過期設定為 Application Load Balancer 的最大 (7 天)。如需如何執行此操作的詳細資訊，請參閱[Classic Load Balancer](#) 或 [Application Load Balancer](#) 的持續時間型工作階段黏性。
- 檢閱 Splunk 平台的要求。Splunk 附加元件 for Firehose 需要 Splunk 平台 6.6.X 版或更新版本。如需詳細資訊，請參閱[Amazon Kinesis Firehose 的 Splunk 附加元件](#)。
- 如果您在 Firehose 和 HTTP 事件收集器 (HEC) 節點之間有代理 (Elastic Load Balancing 或其他)，請啟用黏性工作階段以支援 HEC 認可 (ACKs)。
- 確認您使用的是有效的 HEC 符記。
- 確認 HEC 符記已啟用。
- 檢查傳送到 Splunk 的資料是否正確格式化。如需詳細資訊，請參閱[HTTP 事件收集器格式化事件](#)。
- 確認 HEC 符記和輸入事件已設定有效索引。
- 如果上傳至 Splunk 失敗原因在於 HEC 節點伺服器錯誤，請求會自動重試。如果重試失敗，資料會備份到 Amazon S3。檢查您的資料是否出現在 Amazon S3，若有則表示重試失敗。
- 確認您已在 HEC 符記上啟用 indexer 確認功能。
- 在 Firehose 串流的 Splunk 目的地組態HECAcknowledgmentTimeoutInSeconds中增加 的值。
- 在 Firehose 串流的 Splunk 目的地組態RetryOptions中增加 DurationInSeconds 下的值。
- 檢查您的 HEC 運作狀態。啟用運作狀態檢查是將資料傳輸到 Splunk 的先決條件。
- 若您正使用資料轉換功能，請確認 Lambda 函數的回應從未回傳大小超過 6 MB 的乘載。如需詳細資訊，請參閱[Amazon Data Firehose 資料轉換](#)。
- 請確認名為 ackIdleCleanup 的 Splunk 參數已設定為 true。預設為 false。欲將此參數設定為 true，請執行下列動作：
 - 以受管 [Splunk 雲端部署](#) 而言，請提交使用 Splunk 支援入口網站的案例。此時，請尋求 Splunk 協助啟用 HTTP 事件收集器、將 inputs.conf 內的 ackIdleCleanup 設定為 true，並建立或修改負載平衡器以使用此附加元件。
 - 以分散式 [Splunk Enterprise 部署](#) 而言，請在 inputs.conf 檔案中將 ackIdleCleanup 參數設定為 true。若是 *nix 使用者，這個檔案位於 \$SPLUNK_HOME/etc/apps/splunk_httpinput/local/ 底下。若是 Windows 使用者，此檔案位於 %SPLUNK_HOME%\etc\apps\splunk_httpinput\local\ 底下。
 - 以單一執行個體 [Splunk Enterprise 部署](#) 而言，請在 inputs.conf 檔案中將 ackIdleCleanup 參數設定為 true。若是 *nix 使用者，這個檔案位於 \$SPLUNK_HOME/etc/apps/

splunk_httpinput/local/ 底下。若是 Windows 使用者，此檔案位於 %SPLUNK_HOME%\etc\apps\splunk_httpinput\local\ 底下。

- 請確定 Firehose 串流中指定的 IAM 角色可以存取 S3 備份儲存貯體和 Lambda 函數以進行資料轉換（如果已啟用資料轉換）。此外，請確定 IAM 角色可存取 CloudWatch Logs 群組和日誌串流，以檢查錯誤日誌。如需詳細資訊，請參閱 [將 FirehoseAccess 授予 Splunk 目的地](#)。
- 若要將交付至 S3 錯誤儲存貯體 (S3 備份) 的資料重新驅動回 Splunk，請遵循 [Splunk 文件](#) 中所述的步驟。
- 請參閱 [Amazon Kinesis Firehose 的 Splunk 附加元件故障診斷](#)。

Snowflake 故障診斷

本節說明使用 Snowflake 做為目的地時的常見疑難排解步驟

Firehose 串流建立失敗

如果將資料交付至啟用 PrivateLink 的 Snowflake 叢集的串流建立失敗，則表示 Firehose 無法存取 VPCE-ID。這可能是下列其中一個原因所造成：

- VPCE-ID 不正確。確認沒有排版錯誤。
- Firehose 在預覽中不支援無區域 Snowflake URLs。使用 Snowflake 帳戶定位器提供 URL。如需詳細資訊，請參閱 [Snowflake 文件](#)。
- 確認 Firehose 串流是在與 Snowflake AWS 區域相同的區域中建立。
- 如果問題仍然存在，請聯絡 AWS 支援。

交付失敗

如果資料未交付到您的 Snowflake 資料表，請檢查下列項目。Snowflake 交付失敗的資料將與錯誤代碼和對應至承載的錯誤訊息一起交付至 S3 錯誤儲存貯體。以下是幾個常見的錯誤案例。如需錯誤代碼的完整清單，請參閱 [Snowflake 資料交付錯誤](#)。

- 錯誤代碼：Snowflake.DefaultRoleMissing：表示在建立 Firehose 串流時未設定 snowflake 角色。如果未設定 Snowflake 角色，請確定您將預設角色設定為指定的 Snowflake 使用者。
- 錯誤碼：Snowflake.ExtraColumns：表示插入 Snowflake 因輸入承載中的額外資料欄而遭到拒絕。不應指定資料表中不存在的資料欄。請注意，Snowflake 資料欄名稱區分大小寫。如果即使資料欄出現在資料表中，交付仍失敗，請確定輸入承載中的資料欄名稱大小寫符合資料表定義中宣告的資料欄名稱。

- 錯誤碼：Snowflake.MissingColumns：表示插入 Snowflake 因輸入承載中缺少資料欄而遭到拒絕。確定為所有不可為空的資料欄指定值。
- 錯誤碼：Snowflake.InvalidInput：當 Firehose 無法將提供的輸入承載剖析為有效的 JSON 格式時，可能會發生這種情況。請確定 json 承載格式良好，沒有額外的雙引號、引號、逸出字元等。目前，Firehose 僅支援單一 JSON 項目做為記錄承載，不支援 JSON 陣列。
- 錯誤代碼：Snowflake.InvalidValue：表示由於輸入承載中的資料類型不正確而導致交付失敗。請確定輸入承載中指定的 JSON 值符合 Snowflake 資料表定義中宣告的資料類型。
- 錯誤代碼：Snowflake.InvalidTableType：表示不支援在 Firehose 串流中設定的資料表類型。請參閱受支援資料表、資料欄和資料類型的 snowpipe 串流[限制](#)。

Note

基於任何原因，如果您的 Snowflake 目的地在建立 Firehose 串流後變更資料表定義或角色許可，Firehose 可能需要幾分鐘的時間來偵測這些變更。如果您因此看到交付錯誤，請嘗試刪除並重新建立 Firehose 串流。

針對 Firehose 端點連線能力進行故障診斷

如果 Firehose API 遇到逾時，請執行下列步驟來測試端點連線能力：

- 檢查 API 請求是否來自 VPC 中的主機。來自 VPC 的所有流量都需要設定 Firehose VPC 端點。如需詳細資訊，請參閱[搭配使用 Firehose 與 AWS PrivateLink](#)。
- 如果流量來自己在特定子網路中設定 Firehose VPC 端點的公有網路或 VPC，請從主機執行下列命令來檢查網路連線。Firehose 端點可在 [Firehose 端點和配額](#) 中找到。
- 使用 traceroute 或 等工具 tcping 來檢查網路設定是否正確。如果失敗，請檢查您的網路設定：

例如：

```
traceroute firehose.us-east-2.amazonaws.com
```

或

```
tcping firehose.us-east-2.amazonaws.com 443
```

- 如果網路設定顯示正確且下列命令失敗，請檢查 [Amazon CA \(憑證授權單位\)](#) 是否位於信任鏈中。

例如：

```
curl firehose.us-east-2.amazonaws.com
```

如果上述命令成功，請再次嘗試 API，以查看 API 是否傳回回應。

疑難排解 HTTP 端點

本節說明處理 Amazon Data Firehose 將資料交付至一般 HTTP 端點目的地和合作夥伴目的地時常見的疑難排解步驟，包括 Datadog、Dynatrace、LogicMonitor、MongoDB、New Relic、Splunk 或 Sumo Logic。就本節而言，所有適用的目的地均稱為 HTTP 端點。請確定 Firehose 串流中指定的 IAM 角色可以存取 S3 備份儲存貯體和 Lambda 函數以進行資料轉換（如果已啟用資料轉換）。此外，請確定 IAM 角色可存取 CloudWatch 日誌群組和日誌串流，以檢查錯誤日誌。如需詳細資訊，請參閱 [授予 Firehose 存取 HTTP 端點目的地](#)。

Note

本節中的資訊不適用於下列目的地：Splunk、OpenSearch Service、S3 和 Redshift。

CloudWatch Logs

強烈建議您為 [啟用 CloudWatch Logging](#)。只有在交付到目的地時發生錯誤時，才會發佈日誌。

目的地例外狀況

ErrorCode: HttpEndpoint.DestinationException

```
{
  "deliveryStreamARN": "arn:aws:firehose:us-east-1:123456789012:deliverystream/ronald-test",
  "destination": "custom.firehose.endpoint.com...",
  "deliveryStreamVersionId": 1,
```

```
"message": "The following response was received from the endpoint destination.  
413: {\"requestId\": \"43b8e724-dbac-4510-adb7-ef211c6044b9\", \"timestamp\":  
1598556019164, \"errorMessage\": \"Payload too large\"},  
  \"errorCode\": \"HttpEndpoint.DestinationException\",  
  \"processor\": \"arn:aws:lambda:us-east-1:379522611494:function:httpLambdaProcessing\"  
}
```

目標例外狀況表示 Firehose 能夠與您的端點建立連線並發出 HTTP 請求，但未收到 200 個回應碼。並非 200 的 2xx 個回應也會導致目的地例外狀況。Amazon Data Firehose 會將從設定端點收到的回應程式碼和截斷的回應承載記錄到 CloudWatch Logs。由於 Amazon Data Firehose 會記錄回應程式碼和承載而不進行修改或解譯，因此取決於端點，以提供拒絕 Amazon Data Firehose HTTP 交付請求的確切原因。以下是針對這些例外狀況的最常見疑難排解建議：

- 400：表示您因為 Amazon Data Firehose 組態錯誤而傳送錯誤請求。請確認您有正確的 [url](#)、[一般屬性](#)、[內容編碼](#)、[存取金鑰](#)，以及目的地 [緩衝提示](#)。請參閱有關所需組態的目的地特定文件。
- 401：表示您為 Firehose 串流設定的存取金鑰不正確或遺失。
- 403：表示您為 Firehose 串流設定的存取金鑰沒有將資料交付至已設定端點的許可。
- 413：表示 Amazon Data Firehose 傳送至端點的請求承載太大，端點無法處理。嘗試將 [緩衝提示降低](#) 到目的地的建議大小。
- 429：表示 Amazon Data Firehose 傳送請求的速率高於目的地可以處理的速率。透過增加緩衝時間和/或增加緩衝大小 (但仍在目的地限制內) 來微調緩衝提示。
- 5xx：表示目的地發生問題。Amazon Data Firehose 服務仍然正常運作。

Important

重要事項：雖然這些是常見的疑難排解建議，但特定端點可能有不同的原因提供回應代碼，並且應先遵循端點特定建議。

無效的回應

ErrorCode : HttpEndpoint.InvalidResponseFromDestination

```
{  
  \"deliveryStreamARN\": \"arn:aws:firehose:us-east-1:123456789012:deliverystream/  
ronald-test\",
```

```
"destination": "custom.firehose.endpoint.com...",
"deliveryStreamVersionId": 1,
"message": "The response received from the specified endpoint is invalid.
Contact the owner of the endpoint to resolve the issue. Response for request
2de9e8e9-7296-47b0-bea6-9f17b133d847 is not recognized as valid JSON or has unexpected
fields. Raw response received: 200 {\"requestId\": null}\",
"errorCode": "HttpEndpoint.InvalidResponseFromDestination",
"processor": "arn:aws:lambda:us-east-1:379522611494:function:httpLambdaProcessing"
}
```

無效回應例外狀況表示 Amazon Data Firehose 收到來自端點目的地的無效回應。回應必須符合[回應規格](#)，否則 Amazon Data Firehose 會將交付嘗試視為失敗，並會重新交付相同的資料，直到超過設定的重試持續時間為止。Amazon Data Firehose 會將不符合回應規格的回應視為失敗，即使回應的狀態為 200。如果您正在開發與 Amazon Data Firehose 相容的端點，請遵循回應規格以確保資料成功交付。

以下是一些常見的無效回覆類型，以及修正這些問題的方法：

- 無效的 JSON 或非預期欄位：表示回應無法正確還原序列化為 JSON，或具有非預期的欄位。請確認回應未進行內容編碼。
- 缺失 RequestId：表示回應不包含 requestId。
- RequestId 項目不符：表示回應中的 requestId 與傳出的 requestId 不符。
- 缺失時間戳記：表示回應不包含時間戳記欄位。時間戳記欄位必須是數字，而不是字串。
- 缺失內容類型標題：表示回應不包含「內容類型：應用程式/json」標頭。沒有接受的其他內容類型。

Important

重要： Amazon Data Firehose 只能將資料交付至遵循 Firehose 請求和[回應規格](#)的端點。如果您要將目的地設定為第三方服務，請確定您使用的是正確的 Amazon Data Firehose 相容端點，這可能與公有擷取端點不同。例如，Datadog 的 Amazon Data Firehose 端點是 <https://aws-kinesis-http-intake.logs.datadoghq.com/>，而其公有端點是 <https://api.datadoghq.com/>。

其他常見錯誤

下面列有其他錯誤代碼和定義。

- 錯誤代碼：HttpEndpoint.RequestTimeout - 表示端點花費超過 3 分鐘的時間來回應。如果您是目的地的擁有者，則請縮短目的地端點的回應時間。如果您不是目的地的擁有者，請聯絡擁有者並詢問是否可以採取任何措施來縮短回應時間 (減少緩衝提示，以減少每個請求處理的資料)。
- 錯誤代碼：HttpEndpoint.ResponseTooLarge - 表示回應太大。回應必須小於 1 MiB (包括標頭)。
- 錯誤代碼：HttpEndpoint.ConnectionFailed - 表示無法用設定的端點建立連線。這可能是由於所設定 url 中的錯別字、Amazon Data Firehose 無法存取端點，或端點回應連線請求的時間太長。
- 錯誤代碼：HttpEndpoint.ConnectionReset - 表示已建立連線，但由端點重設或過早關閉。
- 錯誤代碼：HttpEndpoint.SSLHandshakeFailure - 表示無法使用設定端點成功完成 SSL 握手。

疑難排解 MSK As A Source

本節會描述使用 MSK As Source 時的常見疑難排解步驟

Note

如需疑難排解處理、轉換或 S3 交付問題，請參閱前面的章節

Hose 建立失敗

如果您的 MSK 做為來源的軟管建立失敗，請檢查下列項目：

- 檢查來源 MSK 叢集是否處於「作用中」狀態。
- 如果您使用的是私有連線，請確定[叢集上的私有連結已開啟](#)。

如果您使用的是公有連線，請確定[叢集上的公有存取已開啟](#)。

- 如果您正在使用私有連線，則請確認您新增[可讓 Firehose 建立私有連結的資源型政策](#)。另請參閱：[MSK 跨帳戶許可](#)。
- 確定來源組態中的角色具有[從叢集主題擷取資料的許可](#)。
- 確保您的 VPC 安全群組允許[叢集引導伺服器所使用的連接埠](#)上的傳入流量。

Hose 暫停

如果您的 hose 處於懸掛狀態，請檢查以下內容

- 檢查來源 MSK 叢集是否處於「作用中」狀態。

- 檢查來源主題是否存在。如果刪除並重新建立主題，您也必須刪除並重新建立 Firehose 串流。

Hose 預壓

當超出每個分割區的 BytesPerSecondLimit 或正常傳輸流緩慢或停止時，DataReadFromSource.BackPressured 的值將為 1。

- 如果您達到 BytesPerSecondLimit，則請檢查 DataReadFromSource.Bytes 指標並請求提升限制。
- 檢查 CloudWatch 日誌、目的地指標、資料轉換指標和格式轉換指標，以識別瓶頸。

資料新鮮度錯誤

資料新鮮度似乎有誤

- Firehose 會根據使用記錄的時間戳記來計算資料新鮮度。為了確保在生產者記錄保留在 Kafka 的代理日誌中時正確記錄此時間戳記，請將 Kafka 主題時間戳記類型組態設定為 `message.timestamp.type=LogAppendTime`。

MSK 叢集連線問題

下列程序說明如何驗證 MSK 叢集的連線。如需設定 Amazon MSK 用戶端的詳細資訊，請參閱 [《Amazon Managed Streaming for Apache Kafka 開發人員指南》](#) 中的開始使用 Amazon MSK。

驗證 MSK 叢集的連線

1. 建立以 Unix 為基礎的（最好是 AL2）Amazon EC2 執行個體。如果您的叢集上只啟用 VPC 連線，請確定 EC2 執行個體在相同的 VPC 中執行。一旦執行個體可用，SSH 就會進入執行個體。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的[本教學](#)課程。
2. 執行下列命令，使用 Yum 套件管理員安裝 Java。如需詳細資訊，請參閱《Amazon Corretto 8 使用者指南》中的[安裝說明](#)。

```
sudo yum install java-1.8.0
```

3. 執行下列命令來安裝[AWS 用戶端](#)。

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"  
unzip awscliv2.zip
```

```
sudo ./aws/install
```

4. 執行下列命令，下載 Apache Kafka 用戶端 2.6* 版本。

```
wget https://archive.apache.org/dist/kafka/2.6.2/kafka_2.12-2.6.2.tgz
tar -xzf kafka_2.12-2.6.2.tgz
```

5. 前往 `kafka_2.12-2.6.2/libs` 目錄，然後執行下列命令以下載 Amazon MSK IAM JAR 檔案。

```
wget https://github.com/aws/aws-msk-iam-auth/releases/download/v1.1.3/aws-msk-iam-auth-1.1.3-all.jar
```

6. 在 Kafka bin 資料夾中建立 `client.properties` 檔案。
7. `awsRoleArn` 將取代為您在 Firehose 中使用的角色 ARN，`SourceConfiguration` 並驗證憑證位置。允許 AWS 用戶端使用者擔任角色 `awsRoleArn`。AWS client 使用者將嘗試擔任您在此處指定的角色。

```
[ec2-user@ip-xx-xx-xx-xx bin]$ cat client.properties
security.protocol=SASL_SSL
sasl.mechanism=AWS_MSK_IAM
sasl.jaas.config=software.amazon.msk.auth.iam.IAMLoginModule required
  awsRoleArn="<role arn>" awsStsRegion="<region name>";
sasl.client.callback.handler.class=software.amazon.msk.auth.iam.IAMClientCallbackHandler
awsDebugCreds=true
ssl.truststore.location=/usr/lib/jvm/java-1.8.0-
openjdk-1.8.0.342.b07-1.amzn2.0.1.x86_64/jre/lib/security/cacerts
ssl.truststore.password=changeit
```

8. 執行下列 Kafka 命令來列出主題。如果您的連線是公有連線，請使用公有端點 Bootstrap 伺服器。如果您的連線是私有的，請使用私有端點 Bootstrap 伺服器。

```
bin/kafka-topics.sh --list --bootstrap-server <bootstrap servers> --command-config
bin/client.properties
```

如果請求成功，您應該會看到類似下列範例的輸出。

```
[ec2-user@ip-xx-xx-xx-xx kafka_2.12-2.6.2]$ bin/kafka-topics.sh --list --bootstrap-
server <bootstrap servers> --command-config bin/client.properties
```

```
[xxxx-xx-xx 05:49:50,877] WARN The configuration 'awsDebugCreds' was supplied but
  isn't a known config. (org.apache.kafka.clients.admin.AdminClientConfig)
[xxxx-xx-xx 05:49:50,878] WARN The configuration 'ssl.truststore.location' was
  supplied but isn't a known config.
  (org.apache.kafka.clients.admin.AdminClientConfig)
[xxxx-xx-xx 05:49:50,878] WARN The configuration 'sasl.jaas.config' was supplied
  but isn't a known config. (org.apache.kafka.clients.admin.AdminClientConfig)
[xxxx-xx-xx 05:49:50,878] WARN The configuration
  'sasl.client.callback.handler.class' was supplied but isn't a known config.
  (org.apache.kafka.clients.admin.AdminClientConfig)
[xxxx-xx-xx 05:49:50,878] WARN The configuration 'ssl.truststore.password' was
  supplied but isn't a known config.
  (org.apache.kafka.clients.admin.AdminClientConfig)
[xxxx-xx-xx 05:50:21,629] WARN [AdminClient clientId=adminclient-1] Connection to
  node...
__amazon_msk_canary
__consumer_offsets
```

9. 如果您在執行上一個指令碼時遇到任何問題，請確認您提供的引導伺服器可在指定的連接埠上連線。若要這樣做，您可以下載並使用 telnet 或類似公用程式，如下列命令所示。

```
sudo yum install telnet
telnet <bootstrap servers><port>
```

如果請求成功，您會收到下列輸出。這表示您可以在本機 VPC 中連線至 MSK 叢集，且引導伺服器在指定的連接埠上運作狀態良好。

```
Connected to ..
```

10. 如果請求失敗，請檢查 VPC [安全群組](#)上的傳入規則。例如，您可以在傳入規則上使用下列屬性。

```
Type: All traffic
Port: Port used by the bootstrap server (e.g. 14001)
Source: 0.0.0.0/0
```

重試 telnet 連線，如上一個步驟所示。如果您仍然無法連線或 Firehose 連線仍然失敗，請聯絡 [AWS 支援](#)。

Amazon Data Firehose 配額

本節說明 Amazon Data Firehose 內目前配額，先前稱為限制。各項配額除非另有說明，否則都是區域特定規定。

Service Quotas 主控台是一個集中位置，您可以在其中檢視和管理 AWS 服務的配額，並請求增加您使用的許多資源的配額。使用我們提供的配額資訊來管理您的 AWS 基礎設施。計劃在您需要的時候，先行請求提高配額。

如需詳細資訊，請參閱《》中的 [Amazon Data Firehose 端點和配額](#) Amazon Web Services 一般參考。

下一節顯示 Amazon Data Firehose 具有下列配額。

- 使用 Amazon MSK 作為 Firehose 串流的來源，每個 Firehose 串流的預設配額為每個分割區 10 MB/秒的讀取輸送量和 10MB 的最大記錄大小。
- 使用 Amazon MSK 做為 Firehose 串流的來源時，如果啟用 AWS Lambda，記錄大小上限為 6 MB，如果停用 Lambda，則記錄大小上限為 10 MB。AWS Lambda 將其傳入記錄限制為 6 MB，Amazon Data Firehose 會將 6Mb 以上的記錄轉送至錯誤 S3 儲存貯體。如果 Lambda 已停用，Firehose 將其傳入記錄限制為 10 MB。如果 Amazon Data Firehose 從 Amazon MSK 收到大於 10 MB 的記錄大小，Amazon Data Firehose 會將此記錄交付至 S3 錯誤儲存貯體，並將 Cloudwatch 指標發出至您的帳戶。如需 AWS Lambda 限制的詳細資訊，請參閱 [Lambda 配額](#)。
- 啟用 Firehose 串流上的 [動態分割](#) 時，可以為該 Firehose 串流建立 500 個作用中分割區的預設配額。作用中分割區計數是交付緩衝區內的作用中分割區總數。例如，如果動態分割查詢每秒建構 3 個分割區，而且您的緩衝區提示組態會每 60 秒觸發交付，則平均而言，您就會有 180 個作用中分割區。在分割區中交付資料後，此分割區就不再處於作用中狀態。如果您需要更多分割區，您可以建立更多 Firehose 串流，並將作用中分割區分散到其中。
- 啟用 Firehose 串流上的 [動態分割](#) 時，每個作用中分割區支援每秒 1 GB 的最大輸送量。
- 每個帳戶每個區域的 Firehose 串流數量配額如下：
 - 美國東部（維吉尼亞北部）、美國東部（俄亥俄）、美國西部（奧勒岡）、歐洲（愛爾蘭）、亞太區域（東京）：5,000 個 Firehose 串流
 - 歐洲（法蘭克福）、歐洲（倫敦）、亞太區域（新加坡）、亞太區域（雪梨）、亞太區域（首爾）、亞太區域（孟買）、AWS GovCloud（美國西部）、加拿大（西部）、加拿大（中部）：2,000 Firehose 串流

- 歐洲 (巴黎)、歐洲 (米蘭)、歐洲 (斯德哥爾摩)、亞太區域 (香港)、亞太區域 (大阪)、南美洲 (聖保羅)、中國 (寧夏)、中國 (北京)、中東 (巴林)、AWS GovCloud (美國東部)、非洲 (開普敦)：500 Firehose 串流
- 歐洲 (蘇黎世)、歐洲 (西班牙)、亞太區域 (海德拉巴)、亞太區域 (雅加達)、亞太區域 (墨爾本)、中東 (阿拉伯聯合大公國)、以色列 (特拉維夫)、加拿大西部 (卡加利)、加拿大 (中部)、亞太區域 (馬來西亞)、亞太區域 (泰國)、墨西哥 (中部)：100 Firehose 串流
- 如果您超過此數目，呼叫 [CreateDeliveryStream](#) 會導致 `LimitExceededException` 例外狀況。若要增加此配額，您可以使用 [Service Quotas](#) (如果您的區域提供的話)。如需有關使用 [Service Quotas](#) 的詳細資訊，請參閱[請求配額提升](#)。
- 當 Direct PUT 設定為資料來源時，每個 Firehose 串流都會為 [PutRecord](#) 和 [PutRecordBatch](#) 請求提供下列合併配額：
 - 對於美國東部 (維吉尼亞北部)、美國西部 (奧勒岡) 和歐洲 (愛爾蘭)：500,000 筆記錄/秒、2,000 個請求/秒和 5 MiB/秒。
 - 其他 AWS 區域：100,000 筆記錄/秒、1,000 筆請求/秒，以及 1 MiB/秒。

如果 Direct PUT 串流因為較高的資料擷取磁碟區超過 Firehose 串流的輸送量容量而遇到限流，Amazon Data Firehose 會自動增加串流的輸送量限制，直到包含限流為止。根據增加的輸送量和限流，Firehose 可能需要更長的時間，才能將串流的輸送量增加到所需的層級。因此，請繼續重試失敗的資料擷取記錄。如果您預期資料磁碟區在突然的大爆量中增加，或者如果您的新串流需要高於預設輸送量限制的輸送量，請請求增加輸送量限制。

配額有三個配額比例。例如，如果您將美國東部 (維吉尼亞北部)、美國西部 (奧勒岡) 或歐洲 (愛爾蘭) 中的輸送量配額提升至 10 MiB/秒，則另外兩個配額會提升至 4,000 個請求/秒，以及 1,000,000 筆記錄/秒。

Note

- 請勿使用資源層級限制和配額來控制服務的用量。
- 當 Kinesis Data Streams 設定為資料來源時，此配額不適用，Amazon Data Firehose 會無限制地縱向擴展和縮減。
- 如果提高的配額遠高於執行中的流量，則會導致傳送至目的地的批次變得極小，既缺乏效率且可能使目的地服務產生更多成本。提高配額時，請務必符合目前執行中的流量，如果流量增加，應進一步提高配額。
- 較小的資料記錄可能會導致更高的成本。[Firehose 擷取定價](#)是根據您傳送至服務的資料記錄數量，乘以每個記錄的大小四捨五入到最接近的 5KB (5120 位元組)。因此，對於相同

的傳入資料 (位元組) 量，如果有更多的傳入記錄，則產生的成本會更高。例如，如果傳入資料量總計為 5 MiB，則與使用 1,000 筆記錄傳送相同數量的資料相比，傳送超過 5,000 筆記錄的 5 MiB 資料會花費更高。如需詳細資訊，請參閱 [AWS 計算器](#) 中的 Amazon Data Firehose。

- 如果交付目的地無法使用且來源為 DirectPut，每個 Firehose 串流最多可存放 24 小時的資料記錄。如果來源是 Kinesis Data Streams (KDS) 且目的地無法使用，則會根據您的 KDS 組態保留資料。
- 在 base64 編碼之前，傳送至 Amazon Data Firehose 的記錄大小上限為 1,000 KiB。
- [PutRecordBatch](#) 操作每次呼叫可處理的上限為 500 筆記錄或 4 MiB，以較小者為準。此配額無法變更。
- 下列每個操作每秒最多可提供五個叫用，這是硬性限制。
 - [CreateDeliveryStream](#)
 - [DeleteDeliveryStream](#)
 - [DescribeDeliveryStream](#)
 - [ListDeliveryStreams](#)
 - [UpdateDestination](#)
 - [TagDeliveryStream](#)
 - [UntagDeliveryStream](#)
 - [ListTagsForDeliveryStream](#)
 - [StartDeliveryStreamEncryption](#)
 - [StopDeliveryStreamEncryption](#)
- 緩衝提示間隔介於 60 秒到 900 秒。
- 對於從 Amazon Data Firehose 交付到 Amazon Redshift，僅支援可公開存取的 Amazon Redshift 叢集。
- Amazon Redshift 和 OpenSearch Service 交付的重試持續時間介於 0 秒到 7,200 秒之間。
- 當目的地為 Amazon S3、Amazon Redshift 或 OpenSearch Service 時，Amazon Data Firehose 最多允許每個碎片 5 個未完成的 Lambda 調用。對於 Splunk，配額為每個碎片 10 個未完成的 Lambda 調用。
- 您可以使用 CMK 類型 CUSTOMER_MANAGED_CMK 來加密最多 500 個 Firehose 串流。

文件歷史紀錄

下表說明 Amazon Data Firehose 文件的重要變更。

變更	描述	變更日期
移除資料庫做為來源 (公開預覽)	資料庫作為來源 (公開預覽) 現在已移除。	2025 年 9 月 24 日
新增對 Glue 多目錄階層的支援	這可簡化 Firehose 與 Amazon S3 Tables 的整合，而不需要預設資料目錄和之間的資源連結 S3TablesCatalog。請參閱 設定 Firehose 串流到 Amazon S3 資料表 。	2025 年 5 月 14 日
新增資料庫做為來源 (公開預覽)	您現在可以將資料庫變更複寫至 Amazon S3 中的 Apache Iceberg 資料表。	2024 年 11 月 15 日
新增 Apache Iceberg 資料表做為目的地的一般可用性 (GA) 版本	您可以使用 Apache Iceberg Tables 做為目的地來建立 Firehose 串流。請參閱 將資料交付至 Apache Iceberg 資料表 。	2024 年 9 月 30 日
新增的資料類型範例	新增 Apache Iceberg Tables 支援的資料類型範例。請參閱 了解支援的資料類型 。	2024 年 8 月 22 日
新區域啟動	Amazon Data Firehose 現已在亞太區域 (馬來西亞) 提供。請參閱 Amazon Data Firehose 配額 。	2024 年 8 月 22 日
新增 Apache Iceberg 資料表做為目的地 (公開預覽)	您可以使用 Apache Iceberg Tables 做為目的地來建立 Firehose 串流。請參閱 將資料交付至 Apache Iceberg 資料表 。	2024 年 7 月 25 日
Snowflake 的緩衝提示	Snowflake 現在支援緩衝提示。請參閱 the section called “設定 Snowflake 的目的地設定” 。	2024 年 7 月 25 日

變更	描述	變更日期
Snowflake 作為新區域中的目的地	Snowflake 現在可做為亞太區域（新加坡）、亞太區域（首爾）和亞太區域（雪梨）的目的地。請參閱 the section called “設定 Snowflake 的目的地設定” 。	2024 年 7 月 25 日
重組使用者指南章節	使用者指南中章節的簡化導覽。請參閱 將資料傳送至 Firehose 串流 和 故障診斷錯誤 。	2024 年 7 月 5 日
Amazon Data Firehose 與整合 AWS Secrets Manager	您現在可以使用 Secrets Manager 存取您的秘密，並安全地自動化登入資料輪換。請參閱 the section called “使用 驗證 AWS Secrets Manager” 。	2024 年 6 月 6 日
新增對擷取 Dynatrace 日誌的支援	您現在可以將日誌和事件傳送至 Dynatrace 進行進一步分析。請參閱 the section called “設定 Dynatrace 的目的地設定” 。	2024 年 4 月 18 日
Snowflake 做為目的地的一般可用性 (GA) 版本	Snowflake 現已正式推出做為目的地。請參閱 the section called “設定 Snowflake 的目的地設定” 。	2024 年 4 月 17 日
Amazon Kinesis Data Firehose 現在稱為 Amazon Data Firehose	Amazon Kinesis Data Firehose 已重新命名為 Amazon Data Firehose。請參閱 什麼是 Amazon Data Firehose	2024 年 2 月 9 日
新增 Snowflake 做為目的地（公開預覽）	您可以使用 Snowflake 做為目的地來建立 Firehose 串流。請參閱 the section called “設定 Snowflake 的目的地設定” 。	2024 年 1 月 19 日
新增 CloudWatch Logs 的自動解壓縮	您可以在新的或現有的串流上啟用解壓縮，將解壓縮的 CloudWatch Logs 資料傳送至 Firehose 目的地。請參閱 the section called “將 CloudWatch Logs 傳送至 Firehose” 。	2023 年 12 月 15 日
新增 Splunk Observability Cloud 作為目的地	您可以使用 Splunk 可觀測性雲端作為目的地來建立 Firehose 串流。請參閱 the section called “設定 Splunk 可觀測性雲端的目的地設定” 。	2023 年 10 月 3 日

變更	描述	變更日期
已新增 Amazon Managed Streaming for Apache Kafka 作為資料來源	您現在可以設定 Amazon MSK 將資訊傳送至 Firehose 串流。請參閱 the section called “設定 Amazon MSK 的來源設定” 。	2023 年 9 月 26 日
新增了對 OpenSearch Service 目的地 DocumentId 類型的支援	如果 OpenSearch Service 是 Firehose 串流的目的地，DocumentID 類型會指出設定文件 ID 的方法。支援的方法為 Firehose 產生的文件 ID 和 OpenSearch Service 產生的文件 ID。請參閱 the section called “設定目的地設定” 。	2023 年 5 月 10 日
已新增支援動態分割	新增對 Amazon Data Firehose 中串流資料持續動態分割的支援。請參閱 分割區串流資料 。	2021 年 8 月 31 日
新增自訂字首的相關主題。	新增相關主題，其中指出為交付至 Amazon S3 的資料建置自訂字首時，能夠使用的運算式。請參閱 the section called “了解 Amazon S3 物件的自訂字首” 。	2018 年 12 月 20 日
新增 Amazon Data Firehose 教學課程	新增教學課程，示範如何透過 Amazon Data Firehose 將 Amazon VPC 流程日誌傳送至 Splunk。請參閱 使用 Amazon Data Firehose 將 VPC 流程日誌擷取至 Splunk 。	2018 年 10 月 30 日
新增四個新的 Amazon Data Firehose 區域	新增巴黎、孟買、聖保羅和倫敦區域。如需詳細資訊，請參閱 Amazon Data Firehose 配額 。	2018 年 6 月 27 日
新增兩個新的 Amazon Data Firehose 區域	新增首爾和蒙特婁區域。如需詳細資訊，請參閱 Amazon Data Firehose 配額 。	2018 年 6 月 13 日
新的 Kinesis Streams 做為來源功能	新增 Kinesis Streams 作為 Firehose 串流記錄的潛在來源。如需詳細資訊，請參閱 選擇 Firehose 串流的來源和目的地 。	2017 年 8 月 18 日

變更	描述	變更日期
更新主控台文件	Firehose 串流建立精靈已更新。如需詳細資訊，請參閱 教學課程：從主控台建立 Firehose 串流 。	2017 年 7 月 19 日
新資料轉換	您可以設定 Amazon Data Firehose 在資料交付之前轉換資料。如需詳細資訊，請參閱 轉換 Amazon Data Firehose 中的來源資料 。	2016 年 12 月 19 日
新 Amazon Redshift COPY 複製重試	您可以設定 Amazon Data Firehose 在 COPY 命令失敗時，對 Amazon Redshift 叢集重試 COPY 命令。如需詳細資訊，請參閱 教學課程：從主控台建立 Firehose 串流 、 了解 Amazon Data Firehose 中的資料交付 及 Amazon Data Firehose 配額 。	2016 年 5 月 18 日
新的 Amazon Data Firehose 目的地，Amazon OpenSearch Service	您可以使用 Amazon OpenSearch Service 做為目的地來建立 Firehose 串流。如需詳細資訊，請參閱 教學課程：從主控台建立 Firehose 串流 、 了解 Amazon Data Firehose 中的資料交付 及 授予 Firehose 對公有 OpenSearch Service 目的地的存取權 。	2016 年 4 月 19 日
全新增強的 CloudWatch 指標和疑難排解功能	更新 監控 Amazon Data Firehose 和 對 Amazon Data Firehose 中的錯誤進行故障診斷 。	2016 年 4 月 19 日
全新增強型 Kinesis 代理程式	已更新 設定 Kinesis 代理程式以傳送資料 。	2016 年 4 月 11 日
全新 Kinesis 代理程式	新增了 設定 Kinesis 代理程式以傳送資料 。	2015 年 10 月 2 日
初始版本	Amazon Data Firehose 開發人員指南的初始版本。	2015 年 10 月 4 日